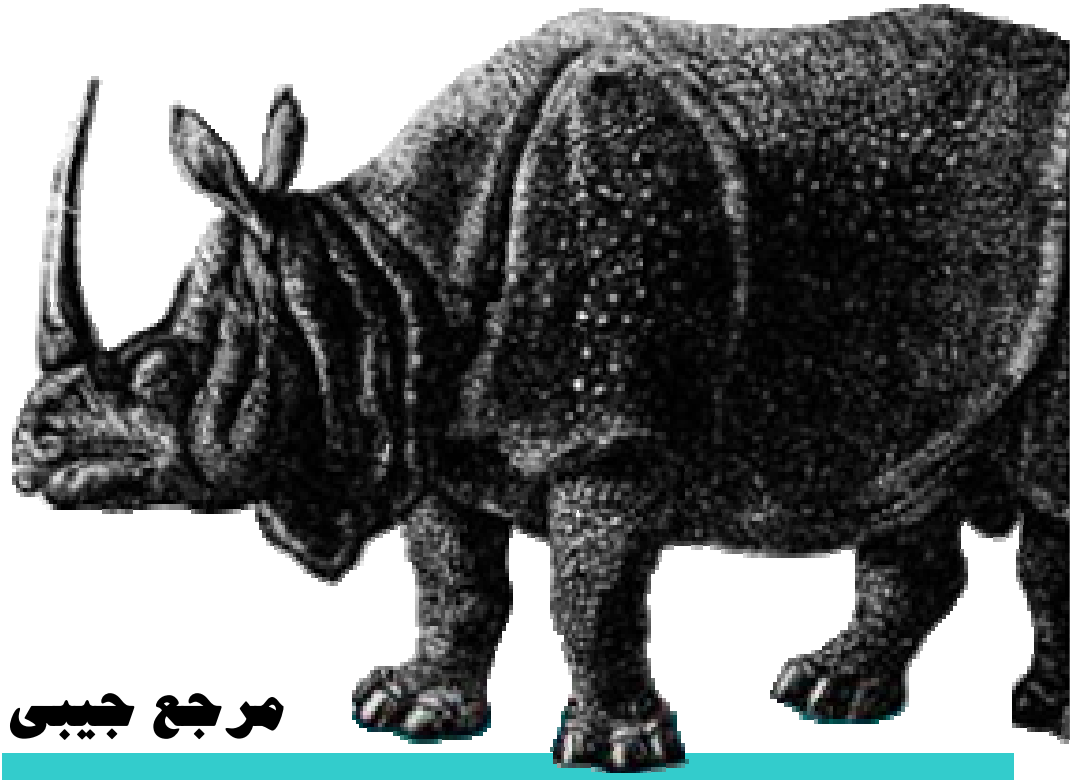

مرجع جیبی

جاوا اسکریپت

دیوید فلاناگان

ترجمه‌ی دکتر قاسم کیانی مقدم

مرجع جیبی

جاوا اسکریپت

دیوید فلاناگان

ترجمه‌ی دکتر قاسم کیانی مقدم
ghasemkiani@gmail.com
http://ghasemkiani.blogspot.com/

توجه:
در صورت تمایل، می‌توانید
در ازای استفاده از این کتاب الکترونیک
مبلغ ۲۰۰۰ تومان
به حساب شماره‌ی
02 00819614 00 6
بانک پارسیان شعبه‌ی ۲۰۳۹
واریز فرمایید.

Version 2.0 2007-03-26

این کتاب ترجمه‌ای است از:

JavaScript Pocket Reference, 2nd Edition
By David Flanagan
Publisher: O'Reilly
Publication Date: October 2002
ISBN: 0-596-00411-7

آثار دیگر از همین مترجم

طعمه اثر مایکل کرایتون
ISBN 964-8605-19-X

سوداگر اثر جان گریشام
ISBN 964-8605-38-6

برای تهیه‌ی کتاب‌های فوق با انتشارات امید مهر (تلفن همراه ۰۹۱۵۱۷۱۰۳۶۰) تماس بگیرید و یا به مراکز پخش مراجعه فرمایید.

فهرست

۱ زبان جاوا اسکریپت

جاوا اسکریپت یک زبان اسکریپت‌نویسی سبک‌وزن مبتنی بر اشیا است که می‌توان آن را در صفحات HTML جا داد. در این کتاب ابتدا جاوا اسکریپت هسته را بررسی می‌کنیم، و سپس مطالبی در باره‌ی جاوا اسکریپت سمت مشتری که در مرورگر استفاده می‌شود، بیان می‌کنیم. قسمت انتهایی کتاب مرجع سریعی برای توابع جاوا اسکریپت هسته و سمت مشتری است.

۱/۱ دستور

دستور زبان جاوا اسکریپت مبتنی بر زبان جاوا است. خود زبان جاوا هم بر مبنای دستور زبان C و C++ بنا شده است. بنا بر این، دستور زبان جاوا اسکریپت برای برنامه‌نویسانی که قبلاً به زبان‌های جاوا، C، و یا C++ برنامه‌نویسی کرده‌اند، بسیار آشنا به نظر خواهد رسید.

۱/۱/۱ حروف کوچک و بزرگ

در جاوا اسکریپت حروف کوچک و بزرگ با هم متفاوت‌اند. تمام کلیدواژه‌ها با حروف کوچک نوشته می‌شوند. نام تمام متغیرها، توابع، و شناسه‌های دیگر باید از نظر حروف کوچک و بزرگ همواره یکسان نوشته شود.

۱/۱/۲ فضای سفید

جاوا اسکریپت از فضای سفید بین نمادها چشمپوشی می‌کند. می‌توانید از فضا، جهش، و سر سطر برای آرایش و خواناسازی متن برنامه‌ی خود استفاده کنید.

۱/۱/۳ ویرگول‌نقطه

دستورات جاوا اسکریپت با ویرگول‌نقطه تمام می‌شود. اما وقتی که بعد از یک دستورالعمل به سطر بعد می‌رویم، می‌توانیم ویرگول‌نقطه‌ی انتهایی را حذف کنیم. دقت کنید که این مسئله باعث می‌شود که از نظر اینکه در متن جاوا اسکریپت

کی می‌توانیم به سطر بعد برویم، محدودیت ایجاد می‌شود. یعنی یک قطعه از دستورالعمل را اگر به تنهایی بتواند دستورالعمل کاملی به شمار رود، نمی‌توان در یک سطر قرار داد، و ادامه‌ی آن را در سطر بعد نوشت.

۱/۱/۴ توضیحات

جاوا اسکریپت هر دو نوع توضیح C و C++ را پشتیبانی می‌کند. متن بین دو علامت */ و /*، هر چند سطر باشد، توضیح به شمار می‌رود، و جاوا اسکریپت از آن صرف نظر می‌کند. در ضمن، هر متنی بین علامت // و پایان سطر باشد، توضیح به شمار می‌رود، و جاوا اسکریپت از آن صرف نظر می‌کند. چند مثال:

```
// This is a single-line, C++-style comment.
/*
 * This is a multi-line, C-style comment.
 * Here is the second line.
 */
/* Another comment. */ // This too.
```

۱/۱/۵ شناسه‌ها

در جاوا اسکریپت، *شناسه‌ها* شامل نام متغیرها، توابع، و برچسب‌ها هستند. شناسه‌ها متشکل از حروف، ارقام، و نویسه‌های _ و $ هستند. با این حال، اولین نویسه‌ی یک شناسه نباید یک رقم باشد. موارد زیر به عنوان شناسه قابل قبول‌اند:

```
i
my_variable_name
v13
$str
```

۱/۱/۶ کلیدواژه‌ها

کلیدواژه‌های زیر بخشی از زبان جاوا اسکریپت هستند، و برای تفسیرگر جاوا اسکریپت معنای خاصی دارند. بنا بر این، از آنها نمی‌توان به عنوان شناسه استفاده کرد:

```
break       do          if
switch      typeof      case
else        in          this
var         catch       false
instanceof  throw       void
continue    finally     new
true        while       default
for         null        try
with        delete      function
return
```

در ضمن، واژه‌های زیر نیز در جاوا اسکریپت برای استفاده‌ی آینده ذخیره شده‌اند. از این کلمات نیز نمی‌توانید به عنوان شناسه استفاده کنید:

```
abstract        enum        int
short           boolean     export
interface       static      byte
extends         long        super
char            final       native
synchronized    class       float
package         throws      const
goto            private     transient
debugger        implements  protected
volatile        double      import
public
```

به علاوه، باید از ایجاد متغیرهایی که نام آنها با خصلت‌ها و روش‌های سراسری یکسان است، اجتناب کنید: مثلاً به صفحات مرجع مربوط به `Global`، `Object`، و `Window` مراجعه کنید. در داخل توابع، از شناسه‌ی `arguments` به عنوان نام آوند یا متغیر محلی استفاده نکنید.

۱/۲ متغیرها

متغیرها با استفاده از دستورالعمل `var` اعلام یا آغازش می‌شوند:

```
var i = 1+2+3;
var x = 3, message = 'hello world';
```

اعلام متغیرها در متن سطح بالای جاوا اسکریپت را می‌توان حذف کرد، ولی اعلام متغیرهای محلی در بدنه‌ی یک تابع لازم است.

متغیرهای جاوا اسکریپت *بدون نوع‌اند*: مقدار یک متغیر از هر نوع داده‌ای می‌تواند باشد.

متغیرهای سراسری به صورت خصلت‌های یک شیء سراسری پیاده‌سازی شده‌اند. متغیرهای محلی در درون تابع‌ها به صورت خصلت‌های شیء آوند تابع پیاده‌سازی می‌شوند. متغیرهای سراسری در سر تا سر برنامه‌ی جاوا اسکریپت قابل دستیابی‌اند. متغیرهایی که در درون یک تابع اعلام شده‌اند، تنها در درون تابع قابل دستیابی‌اند. بر خلاف C، C++، و جاوا، جاوا اسکریپت قلمرو سطح بلوک ندارد: متغیرهایی که در درون قلابهای تابع تعریف می‌شوند، منحصر به آن قطعه نیستند، و از خارج آن هم قابل دستیابی‌اند.

۱/۳ نوع‌های داده‌ای

جاوا اسکریپت سه نوع داده‌ای بدوی را پشتیبانی می‌کند: اعداد، مقادیر بولی، و رشته‌ها؛ و نیز سه نوع داده‌ای مرکب دارد: اشیا و آرایه‌ها. به علاوه، انواع اختصاصی اشیا نیز دارد که نماینده‌ی توابع، عبارت‌های مرتب، و تاریخ هستند.

۱/۳/۱ اعداد

اعداد در جاوا اسکریپت به صورت قالب ۶۴ بیتی شناور نمایش داده می‌شوند. جاوا اسکریپت بین اعداد صحیح و اعداد ممیز شناور افتراقی قایل نمی‌شود. مقادیر عددی در برنامه‌ی جاوا اسکریپت به صورت معمول ظاهر می‌شوند: رشته‌ای از ارقام، با ممیز اعشاری اختیاری و نمای اختیاری. مثال:

```
1
3.14
0001
6.02e23
```

اعداد صحیح ممکن است با نماد شانزدهگانی نیز نمایش داده شوند. اعداد شانزدهگانی با 0x شروع می‌شوند:

```
0xFF // The number 255 in hexadecimal
```

وقتی یک عملیات عددی سرریز می‌شود، مقدار ویژه‌ای بر می‌گرداند که معرف مثبت یا منفی بی‌نهایت است. و وقتی زیرریز می‌شود، صفر بر می‌گرداند. هنگامی که عملیاتی مانند جذر گرفتن از یک عدد منفی یک مقدار خطا یا بی‌معنی بر می‌گرداند، مقدار ویژه‌ی NaN را بر می‌گرداند که معرف مقداری است که یک عدد نیست. برای آزمودن از نظر این مقدار، از تابع isNaN() استفاده کنید.

شیء Number ثابت‌های عددی مفیدی را تعریف می‌کند. شیء Math توابع ریاضی مختلفی را مانند Math.sin()، Math.pow()، Math.random() تعریف می‌نماید.

۱/۳/۲ مقادیر بولی

نوع بولی دو مقدار ممکن دارد، که با کلیدواژه‌های true و false نشان داده می‌شوند. این مقادیر معرف درست یا غلط، روشن یا خاموش، بله یا نه، و یا هر چیز دیگری که بتوان آن را با یک بیت اطلاعات نشان داد، هستند.

۱/۳/۳ رشته‌ها

یک رشته‌ی جاوا اسکریپت عبارت از رشته‌ای از حروف، ارقام، و سایر

نویسه‌ها از مجموعه‌ی نویسه‌های یونیکد[1] ۱۶ بیتی است.

مقادیر رشته‌ای در جاوا اسکریپت با علامت نقل قول منفرد یا دوتایی مشخص می‌شوند. هر نوع نقل قول را می‌توان در داخل نوع دیگر قرار داد:

```
'testing'
"3.14"
'name="myform"'
"Wouldn't you prefer O'Reilly's book?"
```

هرگاه در داخل یک مقدار رشته‌ای نویسه‌ی \ واقع شود، معنای نویسه‌ی بعدی را تغییر می‌دهد. این ترکیبات ویژه در جدول زیر نشان داده شده‌اند:

ترکیب	معنا
\b	فضای وارون
\f	صفحه‌ی جدید
\n	سطر جدید
\r	سر سطر
\t	جهش
\'	آپوستروف یا نقل قول منفرد که پایان دهنده‌ی رشته نیست
\"	نقل قول دوتایی که پایان دهنده‌ی رشته نیست
\\	نویسه‌ی کج‌خط وارون
\xdd	نویسه‌ای که رمز آن در رمزگذاری لاتین ۱ با عدد شانزدهگانی dd مشخص شده است
\udddd	نویسه‌ای که رمز آن در رمزگذاری یونیکد با عدد شانزدهگانی dddd مشخص شده است

کلاس String روش‌های زیادی تعریف می‌کند که می‌توانید برای کار روی رشته‌ها از آنها استفاده کنید. در ضمن، این کلاس خصلت length را تعریف می‌کند که تعداد نویسه‌های موجود در رشته را مشخص می‌نماید.

عملگر جمع (+) رشته‌ها را به هم وصل می‌کند. عملگر تساوی (==) دو رشته را بررسی می‌کند ببیند آیا دقیقاً حاوی رشته‌های نویسه‌ای یکسانی هستند یا نه. (این مقایسه بر روی مقدار رشته‌ها صورت می‌پذیرد و نه بر اساس نشانی آنها، بر خلاف آنچه برنامه‌نویسان C، C++، و یا جاوا ممکن است انتظار داشته باشند.) عملگر عدم تساوی (=!) عکس این کار را انجام می‌دهد. عملگرهای رابطه‌ای (<، =<، >، و =>)، رشته‌ها را با استفاده از ترتیب الفبایی مقایسه می‌کنند.

رشته‌های جاوا اسکریپت تغییرناپذیرند، یعنی به هیچ ترتیبی نمی‌توان محتویات یک رشته را تغییر داد. روش‌هایی که روی رشته‌ها عمل می‌کنند، معمولاً نسخه‌ی تغییر یافته‌ای از رشته را بر می‌گردانند.

[1] Unicode.

۱٫۳٫۴ اشیا

یک شیء یک نوع داده‌ای مرکب است که دارای تعدادی خصلت است. هر خصلت یک نام و یک مقدار دارد. برای دستیابی به خصلت‌های یک شیء از عملگر . استفاده می‌شود. مثلاً می‌توانید مقادیر خصلت‌های یک شیء o را به صورت زیر بخوانید و بنویسید:

```
o.x = 1;
o.y = 2;
o.total = o.x + o.y;
```

بر خلاف C، C++، و جاوا، در جاوا اسکریپت خصلت‌های شیء از قبل تعریف نشده‌اند؛ به هر شیئی می‌توان هر خصلتی را منتسب کرد. اشیای جاوا اسکریپت آرایه‌های ارتباطی هستند: آنها مقادیر داده‌ای دلخواه را با نام‌های دلخواه همراه می‌کنند. بدین سبب، خصلت‌های اشیا را با نماد آرایه می‌توان بیان کرد:

```
o["x"] = 1;
o["y"] = 2;
```

اشیا با عملگر new ساخته می‌شوند. می‌توانید به صورت زیر شیئی بدون خصلت بسازید:

```
var o = new Object();
```

اما به طور معمول از سازنده‌های از پیش تعریف شده برای ساخت اشیایی که عضو کلاسی از اشیا هستند و از قبل تعدادی خصلت و روش مفید برای آنها تعریف شده است، استفاده می‌کنید. مثلاً، با جمله‌ی زیر، می‌توانید یک شیء Date ایجاد کنید که معرف زمان حاضر است:

```
var now = new Date();
```

همچنین، می‌توانید خودتان کلاس‌های اشیای و سازنده‌های ویژه بسازید. انجام این کار را بعداً نشان خواهیم داد.

در جاوا اسکریپت ۱٫۲ و بعد از آن، می‌توانید اشیا را با تعریف مستقیم در متن برنامه ایجاد کنید. تعریف مستقیم شامل لیستی از مقادیر نام و مقدار است که به صورت name:value نوشته می‌شود و با ویرگول از یکدیگر جدا می‌شود، و در داخل قلاب واقع شده است. به عنوان مثال:

```
var o = {x:1, y:2, total:3};
```

به قسمت Object (و Date) در بخش مرجع مراجعه کنید.

۱٫۳٫۵ آرایه‌ها

آرایه نوعی شیء است که، به جای مقادیر نامگذاری شده، مقادیر شماره‌گذاری شده دارد. برای دسترسی به مقادیر شماره‌گذاری شده‌ی یک آرایه از

عملگر [] استفاده می‌شود:

```
a[0] = 1;
a[1] = a[0] + a[0];
```

اولین عنصر یک آرایه‌ی جاوا اسکریپت، عنصر ۰ است. هر آرایه یک خصلت length دارد که تعداد عناصر موجود در آرایه را نشان می‌دهد. آخرین عنصر یک آرایه‌ی جاوا اسکریپت، عنصر length - 1 است. عناصر آرایه می‌توانند حاوی هر نوع مقدار دیگری، از جمله اشیا و آرایه‌های دیگر، باشند، و لازم نیست که مقادیر تمام عناصر یک آرایه از یک نوع باشند.

برای ساختن یک آرایه از سازنده‌ی Array() استفاده می‌کنید:

```
var a = new Array();        // Empty array
var b = new Array(10);      // 10 elements
var c = new Array(1, 2, 3); // Elements 1, 2, 3
```

از جاوا اسکریپت ۱٫۲ به بعد، می‌توانید آرایه‌ها را مستقیماً در متن برنامه ایجاد کنید. در این حالت، مقادیر را در درون کروشه با ویرگول از هم جدا می‌کنید. برای مثال:

```
var a = [1, 2, 3];
var b = [1, true, [1, 2], {x:1, y:2}, "Hello"];
```

برای اطلاع از شماری از روش‌های مفید کار با آرایه‌ها به مبحث آرایه در قسمت مرجع مراجعه کنید.

۱٫۳٫۶ توابع و روش‌ها

تابع قطعه‌ای از متن برنامه‌ی جاوا اسکریپت است که یک بار تعریف می‌شود، و می‌توان آن را به دفعات در برنامه فراخوانی کرد. تعریف تابع به صورت زیر است:

```
function sum(x, y) {
  return x + y;
}
```

توابع با استفاده از عملگر () و با دادن لیستی از مقادیر آوند فراخوانی می‌شوند:

```
var total = sum(1, 2);  // Total is now 3
```

در جاوا اسکریپت ۱٫۱، می‌توانید با استفاده از سازنده‌ی Function() تابع ایجاد کنید:

```
var sum = new Function("x", "y", "return x+y;");
```

در جاوا اسکریپت ۱٫۲ به بعد، توابع با دستور مستقیم تعریف می‌شوند، و لذا سازنده‌ی Function() منسوخ شده است:

```
var sum = function(x, y) { return x+y; }
```

هنگامی که تابعی به خصلت یک شیء اختصاص داده می‌شود، به آن یک *روش* آن شیء می‌گویند. در درون بدنه‌ی یک روش، کلیدواژه‌ی this به شیئی

اشاره می‌کند که تابع، خصلت آن است.

در داخل بدنه‌ی یک تابع، آرایه‌ی `arguments[]` حاوی مجموعه‌ی کامل آوندهایی است که به تابع داده شده است. به مبحث توابع و آوندها در قسمت مرجع مراجعه کنید.

۱/۳/۷ null و undefined

در جاوا اسکریپت دو مقدار وجود دارند که از هیچکدام از انواع فوق‌الذکر نیستند. کلیدواژه‌ی `null` در جاوا اسکریپت مقدار ویژه‌ای است که نشان دهنده‌ی «عدم مقدار» است. اگر متغیری حاوی `null` باشد، می‌فهمید که هیچ مقدار مجازی از نوع دیگر ندارد. دیگر مقدار خاص در جاوا اسکریپت، کلیدواژه‌ی `undefined` است. این مقدار در متغیرهای آغازش نشده وجود دارد، و زمانی نیز که خصلتی را که وجود ندارد، می‌پرسید، بر گردانده می‌شود. در جاوا اسکریپت ۱/۵، متغیر سراسری از پیش تعریف شده‌ای به نام `undefined` وجود دارد که حاوی مقدار ویژه‌ی تعریف نشده است. `null` و `undefined` مقاصد مشابهی دارند، و عملگر == آنها را مساوی می‌انگارد. اگر بخواهید بین آنها تمییز قایل شوید، باید از عملگر === استفاده کنید.

۱/۴ عبارت‌ها و عملگرها

برای تشکیل عبارت‌ها در جاوا اسکریپت، مقادیر را (که ممکن است ثابت‌ها، متغیرها، خصلت‌های شیء، عناصر آرایه، و یا فراخوانی توابع باشند) با استفاده از عملگر با هم ترکیب می‌کنیم. برای تقسیم کردن یک عبارت به چند عبارت فرعی و تغییر دادن ترتیب پیش‌فرض ارزیابی عبارت می‌توان از پرانتز استفاده کرد. چند مثال:

```
1+2
total/n
sum(o.x, a[3])++
```

جاوا اسکریپت مجموعه‌ی کاملی از عملگرها را تعریف کرده است، که اکثر آنها برای برنامه‌نویسان C، C++، و جاوا آشنا به نظر خواهد رسید. لیست عملگرها در جدول زیر آمده است، و بعد مختصراً در باره‌ی عملگرهای استاندارد و غیراستاندارد بحث خواهیم کرد. در ستونی که جهت عملگر را نشان می‌دهد، L نشان دهنده‌ی چپ به راست، و R نشان دهنده‌ی راست به چپ است.

تقدم	جهت	عملگر	عمل مربوطه
۱۵	L	.	دستیابی به خصلت‌های شیء
	L	[]	دستیابی به عناصر آرایه
	L	()	فراخوانی یک تابع
	R	new	ایجاد شیء جدید
۱۴	R	++	افزایش قبلی یا بعدی (عملگر یک‌عملوندی)
	R	--	کاهش قبلی یا بعدی (عملگر یک‌عملوندی)
	R	-	منهای یک‌عملوندی (منفی)
	R	+	به علاوه‌ی یک‌عملوندی (عدم عملگر)
	R	~	متمم بیتی (یک‌عملوندی)
	R	!	متمم منطقی (یک‌عملوندی)
	R	delete	حذف تعریف یک خصلت (یک‌عملوندی) (جاوا اسکریپت ۱٫۲)
	R	typeof	بر گرداندن نوع داده (یک‌عملوندی) (جاوا اسکریپت ۱٫۱)
	R	void	بر گرداندن مقدار تعریف نشده (یک‌عملوندی) (جاوا اسکریپت ۱٫۱)
۱۳	L	*، /، %	ضرب، تقسیم، باقیمانده
۱۲	L	+، -	جمع، تفریق
	L	+	ادغام رشته‌ها
۱۱	L	<<	انتقال چپ عدد صحیح
	L	>>	انتقال راست، بسط علامت
	L	>>>	انتقال راست، بسط صفر
۱۰	L	<، <=	کوچک‌تر، کوچک‌تر یا مساوی
	L	>، >=	بزرگ‌تر، بزرگ‌تر یا مساوی
	L	instanceof	وارسی نوع شیء (جاوا اسکریپت ۱٫۵)
	L	in	وارسی وجود خصلت (جاوا اسکریپت ۱٫۵)
۹	L	==	آزمون تساوی
	L	!=	آزمون عدم تساوی
	L	===	آزمون یکسانی (جاوا اسکریپت ۱٫۳)
	L	!==	آزمون عدم یکسانی (جاوا اسکریپت ۱٫۳)
۸	L	&	AND بیتی صحیح
۷	L	^	XOR بیتی صحیح
۶	L	|	OR بیتی صحیح
۵	L	&&	AND منطقی
۴	L	||	OR منطقی
۳	R	?:	عملگر شرطی (۳ عملوند)
۲	R	=	اختصاص
	R	=*، =+، =-، و غیره	اختصاص با عمل
۱	L	,	ارزیابی چندگانه

عملگرهای جاوا اسکریپت که با زبان‌های C، C++، و جاوا تفاوت دارند، شامل موارد زیر هستند:

- **=== و ==!**

عملگر تساوی جاوا اسکریپت، یعنی ==، تساوی را به صورت آزاد تعریف می‌کند، و امکان تبدیل نوع را نیز می‌دهد. مثلاً، عدد 3 و رشته‌ی "3" را برابر می‌گیرد، false و 0 را برابر می‌گیرد، null و undefined را نیز مساوی فرض می‌کند. عملگر یکسانی، یعنی ===، دقیق‌تر است: فقط زمانی true بر می‌گرداند که دو عملوند یکسان باشند: یعنی نوع یکسانی داشته و مقدارشان برابر باشد. به همین ترتیب، عملگر عدم یکسانی جاوا اسکریپت، ==!، اکیدتر از عملگر =! است.

- **عملگرهای رشته‌ای**

در جاوا اسکریپت، عملگر + علاوه بر جمع زدن آوندهای عددی، آوندهای رشته‌ای را هم ادغام می‌کند. عملگرهای == و === رشته‌ها را بر اساس مقدار مقایسه می‌کنند ببینند آیا دقیقاً حاوی نویسه‌های یکسانی هستند، یا نه. عملگرهای <، =>، >، و =< رشته‌ها را بر اساس ترتیب الفبایی با هم مقایسه می‌کنند.

- **typeof**

نوع عملوند را به صورت رشته بر می‌گرداند. مقادیر "number"، "string"، "boolean"، "object"، "function"، و "undefined" بر می‌گرداند. اگر عملوند null باشد، مقدار "object" بر می‌گرداند.

- **instanceof**

در صورتی مقدار true بر می‌گرداند که شیء طرف چپ با تابع سازنده‌ی طرف راست (مانند Date یا RegExp) ساخته شده باشد.

- **in**

در صورتی مقدار true بر می‌گرداند که شیء طرف راست واجد (یا وارث) خصلتی که نام آن در طرف چپ آمده است، باشد.

- **delete**

یک خصلت شیء را پاک می‌کند. دقت کنید که این معادل آن نیست که به خصلت مقدار null بدهیم. اگر خصلت را نتوان حذف کرد، مقدار false بر می‌گرداند، وگرنه مقدار true بر می‌گرداند.

- **void**

  از عملوند صرف نظر کرده و مقدار undefined بر می‌گرداند.

۱/۵ دستورالعمل‌ها

یک برنامه‌ی جاوا اسکریپت، سلسله‌ای از دستورالعمل‌های جاوا اسکریپت است. اکثر دستورالعمل‌های جاوا اسکریپت مانند زبان‌های C، C++ و جاوا نوشته می‌شوند.

۱/۵/۱ دستورالعمل‌های عبارتی

هر عبارت جاوا اسکریپت می‌تواند به تنهایی به عنوان یک دستورالعمل تلقی شود. اختصاص مقدار، فراخوانی روش، افزایش، و کاهش دستورالعمل‌های عبارتی هستند. برای نمونه:

```
s = "hello world";
x = Math.sqrt(4);
x++;
```

۱/۵/۲ دستورالعمل‌های مرکب

هنگامی که سلسله‌ای از دستورالعمل‌ها در درون قلاب قرار داده شود، به عنوان یک دستورالعمل مرکب تلقی می‌شود. مثلاً بدنه‌ی یک حلقه‌ی while از یک دستورالعمل تشکیل می‌شود. اگر بخواهید که چندین دستورالعمل چند بار انجام شوند، از دستورالعمل مرکب استفاده کنید. این تکنیک به طور شایع برای if، for، و سایر دستورالعمل‌هایی که بعداً خواهیم گفت، مورد استفاده قرار می‌گیرد.

۱/۵/۳ دستورالعمل‌های خالی

دستورالعمل خالی صرفاً متشکل از یک ویرگول نقطه است. گاه می‌توان برای نوشتن حلقه‌ای با بدنه‌ی خالی از دستورالعمل خالی استفاده کرد.

۱/۵/۴ دستورالعمل‌های برچسب‌دار

از جاوا اسکریپت ۱/۲ به بعد، به هر دستورالعمل می‌توان نامی به عنوان برچسب اختصاص داد. با این کار، می‌توان در دستورالعمل‌های break و continue از این برچسب‌ها استفاده کرد.

```
label : statement
```

۱/۵/۵ مرجع الفبایی دستورالعمل‌ها

در بندهای زیر، تمام دستورالعمل‌های جاوا اسکریپت به ترتیب الفبایی ذکر می‌شوند.

- **break**

دستورالعمل break اجرای درونی‌ترین حلقه و یا در مورد جاوا اسکریپت ۱/۲ به بعد، حلقه‌ی نام برده شده را متوقف می‌کند:

```
break ;
break label ;
```

- **case**

این یک دستورالعمل واقعی نیست، بلکه کلیدواژه‌ای برای برچسب زدن دستورالعمل‌ها در درون یک دستورالعمل switch در جاوا اسکریپت ۱/۲ یا بعد از آن است:

```
case constant-expression :
    statements
    [ break ; ]
```

به خاطر ماهیت دستورالعمل switch، گروهی از دستورالعمل‌ها که با برچسب case مشخص شده‌اند، معمولاً باید به دستورالعمل break ختم بشوند.

- **continue**

دستورالعمل continue درونی‌ترین حلقه، و یا در جاوا اسکریپت ۱/۲ یا بعد از آن، حلقه‌ی نام برده شده، را دوباره آغاز می‌کند:

```
continue ;
continue label ;
```

- **default**

این هم مانند case یک دستورالعمل واقعی نیست، بلکه برچسبی است که در درون دستورالعمل switch در جاوا اسکریپت ۱/۲ یا بعد از آن ظاهر می‌شود:

```
default:
    statements
    [ break ; ]
```

- **do/while**

حلقه‌ی do/while تا وقتی که مقدار یک عبارت true باشد، دستورالعملی را به طور مکرر اجرا می‌کند. این دستورالعمل مانند حلقه‌ی

while است، جز اینکه شرط حلقه در پایان حلقه قرار می‌گیرد (و آزموده می‌شود). این بدان معنا است که بدنه‌ی حلقه دستکم یک بار اجرا می‌شود:

```
do
    statement
while (expression) ;
```

این دستورالعمل در جاوا اسکریپت ۱٫۲ اضافه شد. در نت‌اسکیپ[1] ۴، دستورالعمل continue در حلقه‌ی do/while درست کار نمی‌کند.

- **for**

دستورالعمل for یک حلقه‌ی سهل‌الاستفاده است که در آن عبارت‌های آغازش و افزایش با عبارت شرط حلقه ترکیب می‌شوند:

```
for (initialize ; test ; update)
    statement
```

حلقه‌ی for تا وقتی که عبارت *test* مقدار true داشته باشد، *statement* را اجرا می‌کند. این حلقه عبارت *initialize* را یک بار قبل از شروع حلقه اجرا می‌کند، و در پایان هر دور تکرار، عبارت *update* را ارزیابی می‌نماید.

- **for/in**

این عبارت روی خصلت‌های شیء تعیین شده حلقه می‌زند:

```
for (variable in object)
    statement
```

حلقه‌ی for/in برای هر خصلت یک شیء، دستورالعمل را یک بار انجام می‌دهد. قبل از هر بار تکرار، مقدار خصلت را به متغیر تعیین شده اختصاص می‌دهد. بعضی از خواص شیءهای از پیش تعریف شده‌ی جاوا اسکریپت به وسیله‌ی این حلقه احصا نمی‌شود، ولی خصلت‌های تعیین شده توسط کاربر همیشه برشماری می‌شوند.

- **function**

این دستورالعمل تابعی را در برنامه‌ی جاوا اسکریپت تعریف می‌کند:

```
function funcname (args) {
    statements
}
```

دستورالعمل فوق تابعی به نام *funcname* تعریف می‌کند که بدنه‌ی آن متشکل از دستورالعمل‌های ارائه شده است، و آوندهای آن با *args*

[1] Netscape.

مشخص شده است. *args* لیستی متشکل از صفر یا چند نام آوند است که با ویرگول از هم جدا شده‌اند. در بدنه‌ی تابع می‌توان از این آوندها برای اشاره به پارامترهایی که به تابع داده می‌شود، استفاده کرد.

- **if/else**

دستورالعمل if در صورتی که عبارت داده شده true باشد، دستورالعمل مربوطه را اجرا می‌کند:

```
if (expression)
    statement
```

اگر بند else هم وجود داشته باشد، دستورالعمل در صورت false بودن عبارت، دستورالعمل دیگری را اجرا می‌کند:

```
if (expression)
    statement
else
    statement2
```

هر بند else را می‌توان دوباره با یک دستورالعمل if/else ادغام کرد، تا دستورالعمل else if به دست آید:

```
if (expression)
    statement
else if (expression2)
    statement2
else
    statement3
```

- **return**

این دستورالعمل باعث می‌شود که تابعی که فعلاً در حال اجرا است، متوقف شود و به فرا خواننده‌ی خود باز گردد. اگر به دنبال آن عبارتی قرار گرفته باشد، مقدار آن عبارت به عنوان مقدار بازگشتی تابع بر گردانده می‌شود:

```
return ;
return expression ;
```

- **switch**

دستورالعمل switch یک انشعاب چندشاخه‌ای است. این دستورالعمل ابتدا عبارتی را ارزیابی می‌کند، و بعد به دستورالعملی که با برچسب case مشخص شده و منطبق بر مقدار به دست آمده است، می‌رود. اگر مقدار هیچکدام از دستورالعمل‌های مشخص شده با case مطابقت با مقدار مورد نظر نداشته باشد، دستورالعمل switch در صورت وجود به دستورالعملی که با برچسب default مشخص شده است، می‌رود:

```
switch (expression) {
    case constant-expression: statements
    [ case constant-expression: statements ]
    [  . . .  ]
    default: statements
}
```

هر مجموعه از دستورالعمل‌های داخل دستورالعمل switch معمولاً با یک دستورالعمل break یا return تمام می‌شود، تا برنامه از یک مورد وارد مورد دیگر نشود.

- **throw**

دستورالعمل throw نشان دهنده‌ی خطا است، و یا یک استثنا را بر می‌انگیزد. این امر موجب می‌شود که کنترل برنامه بلافاصله به نزدیک‌ترین پردازنده‌ی استثنا منتقل شود (به دستورالعمل try/catch/finally مراجعه کنید). دستورالعمل throw بر اساس ECMA ۳ تعریف شده و در جاوا اسکریپت ۱٫۵ پیاده‌سازی شده است. دستور آن به صورت زیر است:

```
throw expression ;
```

این عبارت *expression* ممکن است به مقداری از هر نوع ارزیابی شود. (به مبحث خطا در قسمت مرجع مراجعه کنید.)

- **try/catch/finally**

دستورالعمل try/catch/finally، ساز و کار پردازش استثناها در جاوا اسکریپت را تشکیل می‌دهد. این دستورالعمل در ECMA ۳ تعریف شده و در جاوا اسکریپت ۱٫۵ پیاده‌سازی شده است. دستور آن به صورت زیر است:

```
try {
    statements
}
catch (argument) {
    statements
}
finally {
    statements
}
```

بند try در این دستورالعمل، قطعه‌ای از متن را مشخص می‌کند که استثناها و خطاهای آن باید مورد پردازش قرار گیرند. اگر خطایی در برنامه صورت پذیرد، و یا استثنایی از درون قطعه‌ی try بر انگیخته شود، کنترل به دستورالعمل‌های پردازش استثنا در بند catch منتقل می‌شود.

این بند شامل یک آوند منفرد و یا یک متغیر محلی است؛ مقداری که توسط استثنا ارائه می‌شود، به این متغیر محلی اختصاص داده می‌شود، تا اینکه بعداً دستورالعمل‌های بند catch بتوانند از آن استفاده کنند. بند finally حاوی دستورالعمل‌هایی است که صرف نظر از اینکه استثنا بر انگیخته شود یا نه، بعد از بندهای try یا catch اجرا می‌شوند. بندهای catch یا finally اختیاری‌اند، ولی هر دوی آنها را نمی‌توان حذف کرد.

- **var**

دستورالعمل var یک یا چند متغیر را اعلام کرده و به طور اختیاری آغازش می‌کند. اعلام متغیر در متن سطح بالا اختیاری است، ولی در داخل بدنه‌ی توابع باید متغیرهای محلی را اعلام کرد.

```
var name [ = value ] [ , name2 [ = value2 ]  . . .  ] ;
```

- **while**

دستورالعمل while یک حلقه‌ی اساسی است. این حلقه تا زمانی که عبارت داده شده true باشد، دستورالعمل درون خود را اجرا می‌کند:

```
while (expression)
    statement ;
```

- **with**

این دستورالعمل، شیئی را به زنجیره‌ی قلمرو اضافه می‌کند، به طوری که یک دستورالعمل در زمینه‌ی آن شیء اجرا می‌شود:

```
with (object)
    statement ;
```

دستورالعمل with اثرات جانبی پیچیده‌ای دارد که به آسانی به ذهن تبادر نمی‌کند؛ قویاً توصیه می‌کنیم که از آن کمتر استفاده شود.

۱/۶ جاوا اسکریپت شیء‌گرا

اشیای جاوا اسکریپت آرایه‌های ارتباطی هستند که مقادیری را با نام‌های خصلت‌ها ارتباط می‌دهند. جاوا اسکریپت ساز و کار اشتقاقی ساده‌ای دارد و می‌توانید برای استفاده در برنامه‌های خود کلاس‌های جدیدی از اشیا تعریف کنید. برای تعریف یک کلاس جدید، ابتدا باید یک تابع سازنده بنویسید. سازنده مثل هر تابع دیگری است، جز اینکه با عملگر new فرا خوانده می‌شود، و برای آغازش شیء جدید و

اشاره به آن از کلیدواژه‌ی this استفاده می‌کند. مثلاً، در اینجا سازنده‌ای است می‌بینید که اشیایی از کلاس جدیدی به نام Point می‌سازد:

```
function Point(x, y) { // Constructor for Point
  this.x = x;  // Initialize X coordinate
  this.y = y;  // Initialize Y coordinate
}
```

هر تابع جاوا اسکریپت که به عنوان سازنده مورد استفاده قرار می‌گیرد، خصلتی به نام prototype دارد. این خصلت به شیء خاصی اشاره دارد که به عنوان سرمشق برای ساخت اشیا توسط سازنده مورد استفاده قرار می‌گیرد. هر گونه خصلتی روی این شیء سرمشق تعریف کنید، برای تمام اشیایی که با سازنده ساخته می‌شوند، به ارث می‌رسد. از شیء سرمشق معمولاً برای قرار دادن روش‌ها در دسترس تمام نمونه‌های یک کلاس استفاده می‌شود. تعریف کردن روش toString امکان می‌دهد که نمونه‌های کلاس قابلیت تبدیل به رشته را داشته باشند. برای مثال:

```
// Define function literals and assign them
// to properties of the prototype object.
Point.prototype.distanceTo = function(that) {
  var dx = this.x - that.x;
  var dy = this.y - that.y;
  return Math.sqrt(dx*dx + dy*dy);
}
Point.prototype.toString = function () {
  return '(' + this.x + ', ' + this.y + ')';
}
```

اگر می‌خواهید روش‌ها یا خصلت‌های ایستا (کلاسی) تعریف کنید، می‌توانید آنها را مستقیماً به تابع سازنده اختصاص دهید و نه به شیء سرمشق. برای مثال:

```
// Define a commonly used Point constant
Point.ORIGIN = new Point(0, 0);
```

در قطعات متنی که مشاهده کردید، یک کلاس Point ساده تعریف شد که می‌توانیم از آن در متنی مانند متن زیر استفاده کنیم:

```
// Call constructor to create a new Point object
var p = new Point(3, 4);
// Invoke a method of the object, using a static
// property as the argument.
var d = p.distanceTo(Point.ORIGIN);
// Adding the object to a string implicitly
// invokes toString().
var msg = "Distance to " + p + " is " + d;
```

۱٫۷ عبارت‌های مرتب

جاوا اسکریپت برای تطابق الگو قابلیت استفاده از عبارت‌های مرتب دارد و

دستور آن همانند زبان برنامه‌نویسی پرل[1] است. جاوا اسکریپت ۱٫۲ قابلیت پشتیبانی از عبارت‌های مرتب پرل ۴ دارد، و در جاوا اسکریپت ۱٫۵ خصوصیات بیشتری که مربوط به عبارت‌های مرتب در پرل ۵ است، به آن اضافه شده است. یک عبارت مرتب در جاوا اسکریپت به صورت سلسله‌ای از نویسه‌ها در داخل نویسه‌های کج‌خط (/) مشخص می‌شود، که به طور اختیاری ممکن است به دنبال آن نویسه‌های شاخص g (جستجوی سراسری)، i (جستجوی غیرحساس به حروف کوچک و بزرگ)، و m (حالت چندسطری؛ این از اضافات جاوا اسکریپت ۱٫۵ است)، قرار گرفته باشند. علاوه بر این دستور تعریف لفظی، می‌توان با استفاده از سازنده‌ی RegExp()، اشیای RegExp ساخت. این سازنده، نویسه‌های الگو و شاخص را به صورت آوندهای رشته‌ای، بدون نویسه‌ی کج‌خط، می‌گیرد.

توضیح کامل دستور عبارت‌های مرتب خارج از حوصله‌ی این کتاب است، ولی جدول‌های موجود در قسمت‌های زیر، خلاصه‌ای از دستور را بیان می‌کنند.

۱٫۷٫۱ نویسه‌های لفظی

حروف، اعداد، و اکثر نویسه‌های دیگر در یک عبارت مرتب لفظی هستند: اینها صرفاً با خودشان تطبیق داده می‌شوند. با این حال، به طوری که در قسمت‌های زیر خواهیم دید، تعدادی از نویسه‌های نقطه‌گذاری و سلسله‌های گریز (که با \ شروع می‌شوند) هستند که معانی خاصی دارند. ساده‌ترین این سلسله‌های گریز روش‌های دیگری برای نمایش نویسه‌های لفظی فراهم می‌کنند:

نویسه	معنا
\n، \r، \t	با مقادیر لفظی نویسه‌های سطر جدید، سر سطر، و جهش منطبق می‌شوند
\\، \/، *، \+، \?، و غیره	با مقدار لفظی نویسه‌ی نقطه‌گذاری منطبق می‌شوند، و مانع از آن می‌شوند که معنای ویژه یا گریز داشته باشد
\xnn	نویسه با رمز شانزدهگانی nn
\uxxxx	نویسه‌ی یونیکد با رمز شانزدهگانی nnnn

۱٫۷٫۲ کلاس‌های نویسه

در دستور عبارت‌های مرتب، از کروشه برای نشان دادن مجموعه‌های نویسه‌ها و یا کلاس‌های موجود در یک الگو استفاده می‌شود. علاوه بر این، سلسله‌های گریز نیز پاره‌ای از کلاس‌های نویسه‌ای شایع را تعریف می‌کنند، که در جدول زیر نشان داده شده‌اند:

[1] Perl.

نویسه	معنا
[...]	با هر نویسه‌ی منفرد در داخل کروشه منطبق می‌شود
[^...]	با هر نویسه‌ی منفرد غیر موجود در داخل کروشه منطبق می‌شود
.	با هر نویسه غیر از سطر جدید منطبق می‌شود
\w، \W	با هر نویسه‌ی کلمه/غیر کلمه منطبق می‌شود
\s، \S	با هر نویسه‌ی فضای سفید/غیر فضای سفید منطبق می‌شود
\d، \D	با هر رقم/غیر رقم منطبق می‌شود

۱/۷/۳ تکرار

جدول زیر دستور عبارت‌های مرتب را برای تعیین اینکه یک انطباق چند بار می‌تواند تکرار شود، نشان می‌دهد.

نویسه	معنا
?	جمله‌ی اختیاری؛ صفر یا یک بار منطبق می‌شود
+	یک یا چند بار با جمله‌ی قبلی منطبق می‌شود
*	صفر یا چند بار با جمله‌ی قبلی منطبق می‌شود
{n}	دقیقاً n بار با جمله‌ی قبلی منطبق می‌شود
{n, }	n بار یا بیشتر با جمله‌ی قبلی منطبق می‌شود
{n, m}	لااقل n بار ولی حداکثر m بار با جمله‌ی قبلی منطبق می‌شود

در جاوا اسکریپت ۱/۵، به دنبال هر کدام از نویسه‌های تکرار می‌توان یک علامت سؤال قرار داد تا حرصشان کمتر شود، یعنی با کمترین تکرار ممکن منطبق شوند تا حدی که الگو بتواند درست در بیاید.

۱/۷/۴ گروه‌بندی و جایگزینی

در عبارت‌های مرتب برای گروه‌بندی عبارت‌های فرعی، درست مانند عبارت‌های ریاضی، از پرانتز استفاده می‌شود. مثلاً، پرانتز برای اینکه یک نویسه‌ی تکرار بر تمام یک زیرعبارت اعمال شود، مفید است. در ضمن می‌توان از آنها به همراه نویسه‌ی | استفاده کرد که برای نشان دادن قابل‌جایگزین بودن به کار می‌رود. گروه‌های قرار گرفته در داخل پرانتز رفتار خاصی دارند: هنگامی که تطابق الگو یافت می‌شود، متنی که با هر گروه منطبق است، ذخیره می‌شود، و می‌توان بر اساس شماره‌ی گروه به آن اشاره کرد. دستور این کار در جدول زیر نشان داده شده است:

نویسه	معنا
a | b	با a یا b منطبق می‌شود
(sub)	زیرعبارت sub به یک جمله‌ی واحد گروه‌بندی می‌شود، و متن منطبق با آن به خاطر سپرده می‌شود
(?:sub)	زیرعبارت sub به یک جمله‌ی واحد گروه‌بندی می‌شود، ولی متن منطبق با آن شماره‌گذاری یا به خاطر سپرده نمی‌شود (جاوا اسکریپت ۱/۵)
\n	دقیقاً با همان نویسه‌هایی که با گروه شماره‌ی n منطبق شده بودند، منطبق می‌شود
$n	در رشته‌های جایگزینی، متنی را که با زیرعبارت شماره‌ی n منطبق شده بود، جایگزین می‌کند

۱/۷/۵ تعیین موقعیت انطباق

یک *لنگر* در عبارت مرتب با موقعیتی در درون رشته (مثلاً اول یا آخر آن) منطبق می‌شود، بدون اینکه با هیچکدام از نویسه‌های آن منطبق گردد. می‌توان از آن برای محدود کردن یا لنگر انداختن انطباق به موقعیت خاصی از رشته استفاده کرد.

نویسه	معنا
^، $	انطباق در اول یا آخر رشته و یا در مورد حالت چند سطری، در اول یا آخر سطر
\b، \B	انطباق در کران کلمه یا در غیر کران کلمه
(?=p)	بررسی پیش‌بینانه: انطباق نویسه‌های بعدی با الگوی داده شده بدون گنجاندن آنها در انطباق (جاوا اسکریپت ۱/۵)
(?!p)	بررسی پیش‌بینانه‌ی منفی: عدم انطباق نویسه‌های بعدی با الگوی داده شده (جاوا اسکریپت ۱/۵)

۱/۸ روایت‌های جاوا اسکریپت

نت‌اسکیپ چند روایت از جاوا اسکریپت را تعریف کرده است، میکروسافت هم روایت‌های کمابیش متناظری به نام "JScript" عرضه کرده است، و نهاد استانداردسازی ECMA سه روایت از جاوا اسکریپت استاندارد تحت عنوان "ECMAScript" عرضه کرده است. در بندهای زیر این روایت‌های مختلف را بررسی کرده و ارتباط آنها را با یکدیگر بررسی می‌کنیم. هر درایه در قسمت مرجع حاوی اطلاعاتی است که نشان می‌دهد آن ویژگی در کدامیک از روایت‌ها ارائه شده است.

- **جاوا اسکریپت ۱/۰**

روایت اولیه‌ی زبان جاوا اسکریپت. این روایت پر از اشکال بود و امروزه تقریباً منسوخ شده است. روایت مذکور در نت‌اسکیپ ۲ پیاده‌سازی شد.

- **جاوا اسکریپت ۱/۱**

شیء آرایه‌ی واقعی را ارائه کرد؛ خطاهای مهم اصلاح شده بود. در

نت‌اسکیپ ۳ پیاده‌سازی شد.

- **جاوا اسکریپت ۱٫۲**
دستورالعمل switch، عبارت‌های مرتب، و شماری از ویژگی‌های دیگر به آن اضافه شد. تقریباً با ECMA ۱ سازگار است، ولی برخی ناهماهنگی‌ها در آن به چشم می‌خورد. در نت‌اسکیپ ۴ پیاده‌سازی شد.

- **جاوا اسکریپت ۱٫۳**
ناهماهنگی‌های جاوا اسکریپت ۱٫۲ اصلاح شده است. با ECMA ۱ سازگار است. در نت‌اسکیپ ۴٫۵ پیاده‌سازی شد.

- **جاوا اسکریپت ۱٫۴**
فقط در محصولات سرور نت‌اسکیپ پیاده‌سازی شد.

- **جاوا اسکریپت ۱٫۵**
پردازش استثنا به آن اضافه شد. با ECMA ۳ سازگار است. در موزیلا[1] و نت‌اسکیپ ۶ پیاده‌سازی شد.

- **جی‌اسکریپت ۱٫۰**
تقریباً معادل جاوا اسکریپت ۱٫۰ است. در نسخه‌های اولیه‌ی اینترنت اکسپلورر[2] ۳ پیاده‌سازی شد.

- **جی‌اسکریپت ۲٫۰**
تقریباً معادل جاوا اسکریپت ۱٫۱ است. در نسخه‌های بعدی اینترنت اکسپلورر ۳ پیاده‌سازی شد.

- **جی‌اسکریپت ۳٫۰**
تقریباً معادل جاوا اسکریپت ۱٫۳ است. با ECMA ۱ سازگار است. در اینترنت اکسپلورر ۴ پیاده‌سازی شد.

- **جی‌اسکریپت ۴٫۰**
در هیچ مرورگر شبکه‌ای پیاده‌سازی نشد.

- **جی‌اسکریپت ۵٫۰**
قابلیت پردازش استثنا دارد؛ به طور نسبی با ECMA ۳ سازگار است. در

[1] Mozilla.
[2] Internet Explorer.

اینترنت اکسپلورر ۵ پیاده‌سازی شد.

- **جی‌اسکریپت ۵٫۵**
تقریباً معادل جاوا اسکریپت ۱٫۵ است. کاملاً با ECMA ۳ سازگار است. در اینترنت اکسپلورر ۵٫۵ و ۶ پیاده‌سازی شده است.

- **ECMA ۱**
اولین ویرایش استاندارد زبان جاوا اسکریپت. ویژگی‌های اساسی جاوا اسکریپت ۱٫۱ را استانداردسازی کرده و چند ویژگی جدید نیز اضافه کرد. دستورالعمل switch و پشتیبانی از عبارت‌های مرتب را استانداردسازی نکرد. پیاده‌سازی‌های سازگار با آن، جاوا اسکریپت ۱٫۳ و جی‌اسکریپت ۳٫۰ هستند.

- **ECMA ۲**
نسخه‌ی نگهدارنده‌ی استاندارد که فقط به توضیح ابهامات بسنده کرده و هیچ ویژگی جدیدی معرفی نکرد.

- **ECMA ۳**
دستورالعمل switch، پشتیبانی از عبارت‌های مرتب، و پردازش استثناها را استانداردسازی کرد. پیاده‌سازی‌های سازگار با آن، جاوا اسکریپت ۱٫۵ و جی‌اسکریپت ۵٫۵ هستند.

۲ جاوا اسکریپت سمت مشتری

منظور از جاوا اسکریپت سمت مشتری، متن جاوا اسکریپتی است که در درون صفحه‌ی HTML جای داده می‌شود و به وسیله‌ی مرورگر شبکه اجرا می‌شود. علاوه بر اشیای هسته که در قسمت قبل گفته شد، متن جاوا اسکریپت سمت مشتری به چند شیء دیگر نیز دسترسی دارد که معرف مرورگر، سند نشان داده شده در مرورگر، و محتوای سند هستند. برنامه‌های جاوا اسکریپت سمت مشتری معمولاً مبتنی بر رویداد هستند، که معنای آن این است که در پاسخ به تعاملات کاربر با مرورگر و سند، *رویدادپردازهای* جاوا اسکریپت اجرا می‌شوند. اسکریپت‌های جاوا اسکریپت سمت مشتری آنقدر قدرتمند است که می‌تواند رخنه‌های امنیتی عمده‌ای را در مرورگر شبکه ایجاد کند. بدین علت، مرورگرهای شبکه معمولاً اعمال اسکریپت‌های سمت مشتری را محدود می‌کنند. در این قسمت ابتدا چگونگی قرار دادن متن جاوا اسکریپت در پرونده‌های HTML را بیان می‌کنیم، آنگاه به ارائه‌ی اشیای جاوا اسکریپت سمت مشتری، رویدادها و رویدادپردازهای جاوا اسکریپت، و محدودیت‌های امنیتی جاوا اسکریپت می‌پردازیم.

۲/۱ جاوا اسکریپت در HTML

متن جاوا اسکریپت را می‌توان به صورت اسکریپت، رویدادپرداز، و نشانی در پرونده‌های HTML قرار داد، که ذیلاً اینها را شرح می‌دهیم.

۲/۱/۱ برگه‌ی <script>

اکثراً متن جاوا اسکریپت در پرونده‌های HTML در درون برگه‌های <script> و </script> واقع می‌شود. برای نمونه:

```
<script>
document.write("The time is: " + new Date());
</script>
```

در جاوا اسکریپت ۱/۱ و بعد از آن، می‌توانید از صفت src برگه‌ی <script> برای مشخص کردن نشانی یک اسکریپت خارجی استفاده کنید. اسکریپت موجود در نشانی مذکور خوانده شده واجرا می‌شود. معمولاً پرونده‌های متن

جاوا اسکریپت با پسوند *.js* مشخص می‌شوند. دقت کنید که در صورت استفاده از این صفت هم برگه‌ی </script> مورد نیاز است:

```
<script src="library.js"></script>
```

در HTML می‌توان اسکریپت‌ها را به زبانی غیر از جاوا اسکریپت نیز نوشت، و مثلاً برخی از مرورگرها، مانند اینترنت اکسپلورر، از زبان‌های دیگری مانند وی‌بی‌اسکریپت[1] نیز پشتیبانی می‌کنند. می‌توان برای تعیین زبان یک اسکریپت از صفت language استفاده کرد. مقدار پیش‌فرض این صفت در تمام مرورگرها "JavaScript" است، لذا معمولاً نیازی به نوشتن آن نیست. علاوه بر این، می‌توان از مقادیری مانند "JavaScript1.3" و "JavaScript1.5" برای تعیین روایت جاوا اسکریپت استفاده کرد. مرورگرهایی که از روایت مشخص شده پشتیبانی نمی‌کنند، از اسکریپت به سادگی چشمپوشی خواهند کرد.

در حقیقت، در HTML ۴ خصلت language برای برگه‌ی <script> وجود ندارد. در عوض، روش رسمی برای تعیین زبان به کار رفته در اسکریپت، استفاده از خصلت type است. برای جاوا اسکریپت، مقدار این خصلت را نوع MIME "text/javascript" قرار دهید.

```
<script src="functions.js"
        language="JavaScript1.5"
        type="text/javascript"></script>
```

۲٫۱٫۲ رویدادپردازها

متن جاوا اسکریپت را می‌توان به عنوان مقدار یک صفت رویدادپرداز برگه‌ی HTML نیز به کار برد. نام صفت‌های رویدادپرداز همیشه با "on" شروع می‌شود. وقتی رویداد مربوطه بروز می‌کند، متن مربوط به رویدادپرداز اجرا می‌شود. مثلاً، در متن HTML زیر، یک دکمه ایجاد می‌شود، و برای خصلت onclick رویدادپردازی تعریف می‌شود که زمانی که کاربر روی دکمه کلیک می‌کند، پیغامی را در پنجره‌ی پیغام نمایش می‌دهد:

```
<input type="button" value="Press Me"
       onclick="alert('Hello World!');">
```

لیست سایر خصلت‌های رویدادپرداز موجود بعداً ارائه خواهد شد.

۲٫۱٫۳ نشانی‌های جاوا اسکریپت

متن جاوا اسکریپت را به صورت نشانی نیز می‌توان ذکر کرد، که برای این منظور، از پروتکل کاذب javascript: استفاده می‌شود. محتوای اینگونه نشانی‌ها

[1] VBScript.

با ارزیابی متن جاوا اسکریپت و تبدیل مقدار حاصله به رشته تعیین می‌شود. اگر می‌خواهید از یک نشانی جاوا اسکریپت استفاده کنید که متن جاوا اسکریپت را اجرا کند، ولی هیچ مقداری بر نگرداند تا جایگزین سند فعلی شود، از عملگر void استفاده کنید:

```
<form action="javascript:void validate()">
```

۲/۲ شیء پنجره

شیء Window معرف یک پنجره‌ی مرورگر شبکه است. در جاوا اسکریپت سمت مشتری، شیء پنجره شیء سراسری تعریف کننده‌ی خصلت‌ها و روش‌های سطح بالا است. بنا بر این، خصلت‌ها و روش‌های Window، خصلت‌های سراسری و توابع سراسری هستند، و می‌توانید بدون پیشوند کردن نام شیء، به آنها اشاره کنید. یکی از خصلت‌های شیء پنجره خصلت window است که دوباره به خود شیء پنجره اشاره می‌کند:

```
window           // The global Window object
window.document  // The document property of the window
document         // Or omit the object prefix
```

برای لیست کامل خصلت‌ها و روش‌های شیء پنجره به مبحث مربوطه در قسمت مرجع مراجعه کنید. در قسمت‌های زیر، موارد مهم‌تر این خصلت‌ها و روش‌ها را بررسی می‌کنیم، و فنون اصلی استفاده از شیء پنجره در برنامه‌نویسی سمت مشتری را نشان می‌دهیم. دقت کنید که مهم‌ترین خصلت شیء پنجره، document است، که به شیء Document اشاره دارد که توصیف کننده‌ی سندی است که در پنجره‌ی مرورگر نمایش داده می‌شود. شیء سند بعداً در مبحث جداگانه‌ای بررسی خواهد شد.

۲/۲/۱ پنجره‌های گفتگوی ساده

برای نمایش پنجره‌های گفتگوی ساده به مشتری سه راه وجود دارد.

()alert به شما امکان می‌دهد که پیغامی را برای مشتری نمایش دهید، ()confirm به شما امکان می‌دهد که سؤال بلی/خیر ساده‌ای بپرسید، و ()prompt به شما امکان می‌دهد که از مشتری بخواهید که یک رشته‌ی یک‌سطری را وارد کند. مثال:

```
alert("Welcome to my home page!");
if (confirm("Do you want to play?")) {
    var n = prompt("Enter your name");
}
```

۲٫۲٫۲ سطر وضعیت

اکثر مرورگرهای شبکه دارای یک سطر وضعیت در پایین پنجره هستند که برای نمایش مقصد پیوندها و اطلاعات دیگر به کار می‌رود. با استفاده از خصلت status می‌توانید کاری کنید که متنی در این سطر وضعیت نمایش داده شود. متنی که به این خصلت بدهید، در ناحیه‌ی وضعیت ظاهر می‌شود، تا زمانی که خود شما یا مرورگر مقدار جدیدی را جایگزین مقدار قبلی کنید. همچنین، می‌توانید با استفاده از خصلت defaultStatus متن پیش‌فرضی را مشخص کنید که مرورگر زمانی که اطلاعات دیگری برای نمایش دادن در سطر وضعیت ندارد، آن را نمایش بدهد. مثلاً، در متن HTML زیر، پیوندی را می‌بینید که متن جاوا اسکریپت موجود در رویدادپرداز آن سبب می‌شود که به جای نشانی پیوند، متن دیگری در سطر وضعیت نمایش داده شود:

```
<a href="help.html"
   onmouseover="window.status='Help'; return true;">Help</a>
```

۲٫۲٫۳ زمان‌سنج‌ها

در جاوا اسکریپت سمت مشتری برای اجرا متن برنامه در زمان بروز یک رویداد از رویدادپرداز استفاده می‌شود. در صورتی که بخواهید متنی پس از گذشتن زمان مشخصی بر حسب میلی‌ثانیه، اجرا شود، می‌توانید از زمان‌سنج استفاده کنید. برای اینکه پس از سپری شدن یک مدت زمان معین، یک رشته‌ی جاوا اسکریپت اجرا شود، از روش setTimeout() استفاده کنید، و رشته‌ی مربوطه و زمان مورد نظر را به عنوان پارامتر به آن بدهید. اگر بخواهید که این متن به صورت مکرر اجرا شود، از روش setInterval() استفاده کنید، و رشته‌ی مربوطه و فاصله‌ی زمانی مورد نظر را به عنوان پارامتر به آن بدهید. هر دو تابع مقداری بر می‌گردانند که می‌توانید آن را به ترتیب به روش clearTimeout() یا clearInterval() بدهید تا اجرای بعدی متن لغو شود. برای مثال:

```
var count = 0;
// Update status line every second
var timer = setInterval("status=++count", 1000);
// But stop updating after 5 seconds;
setTimeout("clearInterval(timer)", 5000);
```

۲٫۲٫۴ اطلاعات سیستم

خصلت‌های navigator و screen از شیء پنجره به اشیای مرورگر و صفحه‌ی نمایش اشاره می‌کنند، که خود خصلت‌هایی را تعریف می‌کنند که حاوی

اطلاعات سیستم، از قبیل نام و روایت مرورگر شبکه، سیستم عاملی که روی آن اجرا می‌شود، و وضوح صفحه‌ی نمایش کاربر هستند. به مباحث مربوط به مرورگر و صفحه‌ی نمایش در قسمت مراجع مراجعه کنید. شیء مرورگر عموماً زمانی استفاده می‌شود که می‌خواهیم متنی را اختصاصاً برای یک مرورگر شبکه‌ی خاص و یا برای روایت خاصی از مرورگر شبکه بنویسیم:

```
if (navigator.appName == "Netscape" &&
    parseInt(navigator.appVersion) == 4) {
  // Code for Netscape 4 goes here.
}
```

۲/۲/۵ ناوبری مرورگر

خصلت location از شیء پنجره به محتوای نوار نشانی مرورگر (جایی که نشانی را در آن تایپ می‌کنید)، اشاره می‌کند. با خواندن مقدار این خصلت می‌توان به نشانی‌ای که در حال حاضر نمایش داده می‌شود، دست یافت. مهم‌تر از همه، با اختصاص مقدار جدید به خصلت location، می‌توان کاری کرد که مرورگر نشانی داده شده را بخواند و آن را نمایش بدهد:

```
// In old browsers, load a different page
if (parseInt(navigator.appVersion) <= 4)
    location = "staticpage.html";
```

دقت کنید که هر اسکریپت یا رویدادپردازی که به خصلت location مربوط به پنجره‌ی خود، نشانی جدیدی بدهد، باعث خواهد شد که با خوانده شدن آن سند، خودش هم از بین برود و اجرایش متوقف شود!

گرچه مقدار خصلت location را مانند یک رشته می‌توان پرسجو یا تعیین کرد، ولی این خصلت در واقع به شیئی از نوع Location اشاره می‌کند. این شیء خصلت‌هایی دارد که به شما امکان می‌دهد که قسمت‌های مختلف نشانی حاضر را بخوانید یا بنویسید:

```
// Get the substring of the URL following ?
var query = location.search.substring(1);
// Scroll to a named portion of the document
location.hash = "#top";
```

به علاوه، روش ()reload باعث می‌شود که مرورگر دوباره نشانی حاضر را بار کند.

خصلت history از شیء پنجره به شیء سابقه در پنجره‌ی مرورگر اشاره می‌کند. این شیء روش‌هایی دارد که به شما امکان می‌دهد که در بین نشانی‌های مرور شده‌ی آن جلو یا عقب بروید، درست همانطور که کاربر با دکمه‌های Back و Forward این کار را انجام می‌دهد:

```
history.back();    // Go back once
```

```
history.forward(); // Go forward
history.go(-3);    // Go back three times
```

۲/۲/۶ کنترل پنجره

شیء پنجره روش‌هایی برای حرکت دادن، تغییر اندازه، و در نوردیدن پنجره دارد، و نیز روش‌هایی برای منتقل کردن کانون صفحه‌کلید به یک پنجره و یا خارج کردن از آن پنجره دارد. برای مثال:

```
// Automatically scroll 10 pixels a second
setInterval("scrollBy(0, 1)", 100);
```

برای اطلاعات بیشتر در خصوص روش‌های moveTo()، moveBy()، resizeTo()، resizeBy()، scrollTo()، scrollBy()، focus()، و blur() به مبحث شیء پنجره در قسمت مرجع، مراجعه کنید.

مهم‌تر از این روش‌ها، که روی پنجره‌های موجود عمل می‌کنند، دو روش دیگر هستند: روش open() که پنجره‌ی مرورگر جدیدی ایجاد می‌کند، و روش close() که یک پنجره‌ی ایجاد شده با اسکریپت را می‌بندد. روش open() سه آوند می‌گیرد. آوند اول نشانی نمایش داده شده در پنجره است. آوند دوم نام اختیاری پنجره است. اگر یک پنجره‌ی قبلی با آن نام وجود داشته باشد، همان پنجره دوباره مورد استفاده قرار می‌گیرد، و پنجره‌ی جدیدی ایجاد نمی‌شود. آوند سوم رشته‌ای اختیاری است که اندازه‌ی پنجره‌ی جدید و مشخصات یا ویژگی‌هایی را که باید نمایش بدهد، تعیین می‌کند. برای مثال:

```
// Open a new window
w = open("new.html", "newwin", // URL and name
         "width=400, height=300, " + // size
         "location, menubar, " +     // chrome
         "resizable, scrollbars, status, toolbar");
// And close that new window
w.close();
```

دقت کنید که اکثر مرورگرها فقط به اسکریپت‌ها اجازه می‌دهند که پنجره‌هایی را که خودشان باز کرده‌اند، ببندند. همچنین، به خاطر رشد اخیر پنجره‌های جهشی آگهی آزار دهنده، بعضی از مرورگرها اصلاً به اسکریپت‌ها اجازه‌ی باز کردن هیچ پنجره‌ای نمی‌دهند.

۲/۲/۷ پنجره‌ها و کادرهای متعدد

به طوری که قبلاً گفتیم، با استفاده از روش open() متعلق به شیء پنجره می‌توان پنجره‌های مرورگر جدیدی ساخت که هر کدام معرف یک شیء پنجره‌ی جدید هستند. پنجره‌ای که یک اسکریپت در آن اجرا می‌شود، شیء سراسری برای آن

اسکریپت است، و می‌توانید از تمام خصلت‌ها و روش‌های آن شیء به طوری که گویی به صورت سراسری تعریف شده‌اند، استفاده کنید. لیکن زمانی که اسکریپتی که در یک پنجره اجرا می‌شود، نیازمند تعامل با پنجره‌ی دیگری است، لازم است که شیء Window به صراحت ذکر گردد:

```
// Create a new window and manipulate it
var w = open("newdoc.html");
w.alert("Hello new window");
w.setInterval("scrollBy(0, 1)", 50);
```

در HTML یک پنجره می‌تواند کادرهای متعدد داشته باشد. بسیاری از طراحان شبکه از کادرها دوری می‌کنند، ولی با این حال، استفاده از آنها نسبتاً شایع است. جاوا اسکریپت با هر کادر به عنوان یک شیء پنجره‌ی جداگانه رفتار می‌کند، و اسکریپت‌های موجود در کادرهای مختلف مستقل از هم اجرا می‌شوند. خصلت frames از شیء پنجره آرایه‌ای از اشیای پنجره است، که معرف زیرکادرهای یک پنجره هستند:

```
// Scripts in framesets refer to frames like this:
frames[0].location = "frame1.html";
frames[1].location = "frame2.html";
// With deeply nested frames, you can use:
frames[1].frames[2].location = "frame2.3.html";
// Code in a frame refers to the top-level window:
top.status = "Hello from the frame";
```

خصلت parent از شیء پنجره به کادر یا پنجره‌ای که پنجره در آن واقع شده است، اشاره دارد. خصلت top به پنجره‌ی سطح بالای مرورگر که ریشه‌ی سلسله‌ی کادرها است، اشاره می‌کند. (اگر شیء پنجره خود یک پنجره‌ی سطح بالا نه یک کادر باشد، خصلت‌های parent و top صرفاً به خود شیء پنجره اشاره می‌کنند.)

هر پنجره و کادر مرورگر، قرینه‌ی متفاوتی برای اجرای جاوا اسکریپت دارد، و در هر قرینه، شیء پنجره شیء سراسری است. این بدان معنا است که هر متغیر اعلام شده یا خصلت تعریف شده تبدیل به خصلت‌های شیء پنجره‌ی مربوطه می‌شوند. به این ترتیب، اسکریپتی در یک پنجره یا کادر می‌تواند از متغیرها و توابع تعریف شده در پنجره یا کادر دیگر استفاده کند. مثلاً، شایع است که توابع را در قسمت <head> پنجره‌ی سطح بالا تعریف می‌کنند، و بعد اسکریپت‌ها و رویدادپردازهای موجود در کادرهای درونی با استفاده از خصلت top به آن توابع دست پیدا می‌کنند:

```
// Code in a frame calls code in the top-level window.
top.stop_scrolling();
```

۲/۳ شیء سند

هر شیء پنجره یک خصلت document دارد که به شیئی از نوع سند اشاره می‌کند. شیء سند را به جهاتی می‌توان مهم‌تر از خود شیء پنجره دانست: در حالی که پنجره معرف پنجره‌ی مرورگر است، سند معرف سندی HTMLای است که در آن پنجره نمایش داده می‌شود. شیء سند خصلت‌های مختلفی دارد که به اشیایی اشاره می‌کنند که امکان دستیابی و تغییر محتویات سند را فراهم می‌نمایند. برای دستیابی به محتویات سند از روشی به نام مدل شیء سند، یا DOM، استفاده می‌شود، و هم‌اینک چند نوع DOM وجود دارند:

- **DOM قدیمی**

نوع اولیه‌ی مدل شیء سند به موازات روایت‌های ابتدایی زبان جاوا اسکریپت توسعه یافت. این نوع مورد حمایت همه‌ی مرورگرها است، ولی تنها امکان دستیابی به برخی قسمت‌های کلیدی سند را، از قبیل فرم‌ها، عناصر فرم، و تصاویر، فراهم می‌کند.

- **W3C DOM**

این مدل شیء سند امکان دستیابی و تغییر تمام محتوای سند را فراهم می‌کند، و به وسیله‌ی کنسرسیوم شبکه‌ی جهانی[1] (W3C) استانداردسازی شده است. این نوع لااقل به طور نسبی به وسیله‌ی نت‌اسکیپ ۶ و بالاتر، اینترنت اکسپلورر ۵ و بالاتر، و سایر مرورگرهای مدرن پشتیبانی می‌شود. W3C DOM به طور دقیق با IE4 DOM سازگار نیست، ولی بسیاری از ویژگی‌های قدیمی DOM اولیه را استانداردسازی کرده است. این کتاب ویژگی‌های هسته‌ای استاندارد را پوشش می‌دهد، و زیرمجموعه‌ی ساده شده‌ای از DOM را که برای برنامه‌نویسان جاوا اسکریپت که با سندهای HTML کار می‌کنند، ارائه می‌نماید. برای توضیحات کامل به کتاب‌های جامع‌تر در باره‌ی جاوا اسکریپت مراجعه کنید.

- **IE4 DOM**

ویرایش ۴ اینترنت اکسپلورر میکروسافت، DOM قدیمی را با ویژگی‌های جدیدی برای دستیابی و امکان تغییر تمام محتوای یک سند گسترش داد. این ویژگی‌ها هرگز استانداردسازی نشدند، ولی بعضی از آنها در

[1] World Wide Web Consortium.

مرورگرهای غیر میکروسافت نیز پشتیبانی می‌شوند.

در قسمت‌های زیر هر کدام از این DOMها را با تفصیل بیشتر بررسی می‌کنیم، و توضیح خواهیم داد که چگونه می‌توانید از آنها برای دستیابی و تغییر دادن محتوای سند استفاده کنید.

۲٫۴ DOM قدیمی

DOM اولیه‌ی جاوا اسکریپت سمت مشتری از طریق خصلت‌های شیء سند امکان دستیابی به محتویات سند را فراهم می‌کند. خصلت‌های فقط-خواندنی متعددی، از قبیل `title`، `URL`، و `lastModified`، اطلاعاتی را در باره‌ی سند به عنوان یک کل در اختیار ما قرار می‌دهند. برای اطلاعات بیشتر در باره‌ی این خصلت‌های شیء سند به مباحث مربوطه در قسمت مرجع مراجعه نمایید. سایر خصلت‌ها، آرایه‌هایی هستند که به انواع خاص محتوای سند اشاره می‌کنند:

- **`forms[]`**
  آرایه‌ای از اشیای فرم که معرف فرم‌های موجود در سند است.
- **`images[]`**
  آرایه‌ای از اشیای تصویر که معرف تصویرهای موجود در سند است.
- **`applets[]`**
  آرایه‌ای از اشیا که معرف برنامک‌های جاوای موجود در سند است. در واقع از جاوا اسکریپت می‌توان برای اسکریپت‌نویسی جاوا و کنترل این برنامک‌ها استفاده کرد، ولی چگونگی انجام این کار فراتر از حد این کتاب است.
- **`links[]`**
  آرایه‌ای از اشیای پیوند که معرف پیوندهای موجود در سند است.
- **`anchors[]`**
  آرایه‌ای از اشیای لنگر که معرف لنگرهای موجود در سند است (لنگر به مکان‌های نامگذاری شده‌ی درون سند می‌گویند که با استفاده از صفت `name` از برگه‌ی HTML `<a>` ایجاد می‌شود).

این آرایه‌ها حاوی اشیای مربوطه به ترتیب بروز در سند هستند. بنا بر این، اولین فرم در یک سند `document.forms[0]` است، و سومین تصویر

document.images[2]. راه دیگر برای اشاره فرم‌ها، تصاویر، و برنامک‌های سند، دادن نام به آنها با استفاده از صفت HTML name است:

```
<form name="address">...</form>
```

هنگامی که به این طریق به یک فرم، تصویر، یا برنامک نامی داده می‌شود، می‌توان با استفاده از آن نام آن را در آرایه پیدا کنید، و یا اینکه مستقیماً آن را به صورت خصلتی از خود سند مورد استفاده قرار دهید:

```
document.forms["address"]  // A named form
document.address           // The same thing
```

شیء فرم خصوصاً جالب است. این شیء یک آرایه‌ی elements[] دارد که حاوی اشیایی است که عناصر فرم را به ترتیب ظهور در فرم نشان می‌دهند. برای جزئیات بیشتر در مورد این عناصر فرم به مباحث Input، Select، و Textarea در قسمت مرجع مراجعه فرمایید.

آرایه‌ی elements[] یک فرم تا حد زیادی مانند آرایه‌ی forms[] یک سند عمل می‌کند؛ این آرایه حاوی عناصر فرم به ترتیب ظهور در فرم است، ولی این هم امکان آن را به ما می‌دهد که عناصر را به نام مورد اشاره قرار دهیم. چکیده‌ی HTML زیر را در نظر بگیرید:

```
<form name='address'><input name='street'></form>
```

برای اشاره به عنصر ورود متن در این فرم از چند راه می‌توانیم استفاده کنیم:

```
document.forms[0].elements[0]
document.address.elements['street']
document.address.street
```

DOM قدیمی هیچ راهی برای اشاره به محتوای سند غیر از فرم‌ها، عناصر فرم، تصاویر، برنامک‌ها، پیوندها، و لنگرها فراهم نمی‌کند. مثلاً، هیچ آرایه‌ای وجود ندارد که حاوی تمام برگه‌های <h1> باشد، و یا به هیچ طریقی نمی‌توان متن معمولی سند را گرفت. این نقصی است که به طوری که بعداً خواهیم دید، در DOMهای W3C و اینترنت اکسپلورر ۴ به آن پرداخته شده است. با این حال، به طوری که در قسمت‌های زیر خواهیم دید، DOM قدیمی با وجود محدودیت‌هایی که دارد، امکان آن را فراهم می‌کند که اسکریپت‌ها محتوای سند را به طور پویا تغییر دهند.

۲/۴/۱ سندهای ایجاد شده به طور پویا

علاوه بر خصلت‌های پیشگفته، شیء سند چندین روش مهم برای تولید پویای محتوای سند نیز دارد. با استفاده از روش write() می‌توانید متنی را در سند در محل برگه‌ی <script> حاوی فراخوانی روش بنویسید. برای مثال:

```
document.write("<p>Today is: " + new Date());
document.write("<p>Document updated: " +
  document.lastModified);
```

دقت کنید که متنی که به این ترتیب نوشته می‌شود، می‌تواند حاوی هر کدام از برگه‌های HTML نیز باشد؛ مرورگر بعد از اجرا کردن اسکریپت، متن نوشته شده توسط آن را تجزیه کرده و نمایش می‌دهد.

از روش write() در برگه‌ی <script> تنها در زمانی که صفحه هنوز در حال خوانده شدن است، می‌توان استفاده کرد. اگر سعی کنید که از آن در داخل رویدادپردازی استفاده کنید که بعد از بار شدن کامل سند فعال می‌شود، باعث پاک شدن سند و خود رویدادپرداز می‌شود. با این حال، می‌توانید از یک رویدادپرداز در یک پنجره یا کادر برای فعال کردن فراخوانی document.write() در یک پنجره‌ی دیگر استفاده کنید. اما برای این کار، باید محتویات کامل یک سند جدید را بنویسید، و یادتان باشد که وقتی کارتان تمام شد، روش document.close() را فراخوانی کنید:

```
var clock = open("", "", "width=400, height=30");
var d = clock.document; // Save typing below
setInterval("d.write(new Date());d.close();",
            1000);
```

۲/۴/۲ فرم‌های پویا

به طوری که دیده‌ایم، آرایه‌ی elements[] از یک شیء فرم حاوی اشیایی متناظر با عناصر موجود در فرم است. بسیاری از این اشیا خصلت‌هایی دارند که می‌توانید از آن برای خواندن یا نوشتن مقدار نمایش داده شده در عنصر فرم استفاده کنید. این راه دیگری برای تغییر پویای محتویات سند است. مثلاً، متن زیر خصلت value در یک شیء Text را به گونه‌ای تعیین می‌کند که زمان محلی فعلی را نمایش دهد:

```
<form><input size=10></form>  // An HTML form
<script>  /* Display a clock in the form */
// The Text element we're working with.
var e = document.forms[0].elements[0];
// Code to display the time in that element
var s="e.value=(new Date()).toLocaleTimeString();"
setInterval(s, 1000); // Run it every second
</script>
```

۲/۴/۳ اعتبارسنجی فرم

برگه‌ی <form> یک رویدادپرداز onsubmit دارد که زمانی که کاربر سعی می‌کند فرمی را تحویل دهد، فعال می‌شود. می‌توانید از این رویدادپرداز برای *اعتبارسنجی* استفاده کنید: یعنی مثلاً بررسی اینکه تمام فیلدهای لازم در فرم پر شده‌اند. اگر رویدادپرداز onsubmit مقدار false بر گرداند، فرم تحویل داده

نمی‌شود. برای مثال:

```
<form name="address" onsubmit="checkAddress()">
<!-- form elements go here -->
</form>
<script>
// A simple form validation function
function checkAddress() {
  var f = document.address; // The form to check
  // Loop through all elements
  for(var i = 0; i < f.elements.length; i++) {
    // Ignore all but text input elements
    if (f.elements[i].type != "text") continue;
    // Get the user's entry
    var text = f.elements[i].value;
    // If it is not filled in, alert the user
    if (text == null || text.length == 0) {
      alert("Please fill in all form fields.");
      return false;
    }
  }
}
</script>
```

۲/۴/۴ روان‌دازی تصاویر

DOM قدیمی به شما امکان ایجاد یکی از جلوه‌های ویژه‌ی شایع را می‌دهد، و آن توانایی جایگزینی پویای یک تصویر با تصویر دیگر است. از این فن عموماً برای روان‌دازی تصاویر استفاده می‌شود، یعنی اینکه وقتی نشانگر موشواره روی تصویر قرار می‌گیرد، تصویر عوض می‌شود. آرایه‌ی images[] از شیء سند حاوی اشیایی از نوع تصویر است که معرف تصاویر موجود در سند هستند. هر تصویر یک خصلت src دارد که نشانی تصویری را که باید نمایش داده شود، مشخص می‌کند. برای تغییر تصویر نمایش داده شده، کافی است که به این خصلت یک نشانی جدید بدهید:

```
document.images[0].src = "newbanner.gif";
```

برای اینکه از این تکنیک جهت جلوه‌ی روان‌دازی تصاویر استفاده کنیم، باید از آن به همراه رویدادپردازهای onmouseover و onmouseout استفاده کنیم. این رویدادپردازها به ترتیب زمانی که موشواره وارد تصویر می‌شود و از آن خارج می‌شود، فعال می‌شوند . در مثال زیر، متن جاوا اسکریپت کوچکی را به همراه رویدادپردازها می‌بینید که جلوه‌ی روان‌دازی تصویر را ایجاد می‌کند:

```
<img name="button" src="b1.gif"
     onmouseover="document.button.src='b2.gif';"
     onmouseout="document.button.src='b1.gif';">
```

هنگامی که قرار است که تصویری به صورت پویا نمایش داده شود، بهتر است که قبلاً آن را به حافظه‌ی نهان مرورگر بار کنیم تا موقع ظاهر شدن تأخیری

پیش نیاید. برای این منظور، می‌توانید از یک تصویر پویای خارج از صفحه استفاده کنید:

```
var i = new Image(); // Create Image object
i.src="b2.gif";      // Load, but don't display image
```

۲/۴/۵ کار با کوکی‌ها

خصلت cookie شیء سند خصلت ویژه‌ای است که با آن می‌توانید کوکی‌های مربوط به سند را بخوانید و بنویسید. برای همراه کردن یک کوکی گذرا با سند، کافی است که به خصلت کوکی، رشته‌ای به صورت زیر اختصاص دهید:

name=value

با این کار، یک کوکی با نام و مقدار مشخص شده برای این سند ساخته می‌شود. اگر بخواهید کوکی‌ای ایجاد کنید که حتی بعد از خارج شدن کاربر از مرورگر نیز ذخیره شود، باید با استفاده از رشته‌ای به صورت زیر، تاریخ انقضای آن را مشخص نمایید:

name=value; expires=*date*

تاریخ انقضا باید به همان صورتی باشد که به وسیله‌ی روش Date.toGMTString() بر گردانده می‌شود. اگر بخواهید که کوکی برای سندهای دیگر پایگاه اینترنتی شما نیز قابل دستیابی باشد، باید مسیر آن را نیز مشخص نمایید:

name=value; expires=*date*; path=*prefix*

یک سند واحد ممکن است با بیش از یک کوکی همراه باشد. برای خواندن کوکی‌های سند، کافی است که مقدار خصلت cookie را بخوانید. این رشته حاوی رشته‌های به صورت name=value است که با یک ویرگول‌نقطه و یک فضا از هم جدا شده‌اند. هنگام خواندن کوکی‌ها، هیچگاه به قسمت‌های path= یا expires= برخورد نمی‌کنید؛ در این رشته صرفاً نام و مقدار کوکی‌ها ذکر شده است. در اینجا تابعی را می‌بینید که مقدار یک کوکی با نام مشخص را در خصلت cookie پیدا می‌کند. فرض بر این است که در مقادیر کوکی‌ها هرگز نویسه‌ی ویرگول‌نقطه یافت نمی‌شود.

```
function getCookie(name) {
  // Split cookies into an array
  var cookies = document.cookie.split('; ');
  for(var i = 0; i < cookies.length; i++) {
    var c = cookies[i];          // One cookie
    var pos = c.indexOf('=');    // Find = sign
    var n = c.substring(0, pos); // Get name
    if (n == name)               // If it matches
      return c.substring(pos+1);  // Return value
  }
  return null;  // Can't find the named cookie
```

```
}
```

W3C DOM ۲٫۵

این نوع DOM، ویژگی‌های DOM قدیمی را استاندارد می‌کند، لیکن ویژگی‌های مهم جدیدی را نیز اضافه می‌نماید. این نوع، علاوه بر آرایه‌های forms[]، images[]، و سایر خصلت‌های شیء سند، روش‌هایی را نیز تعریف می‌کند که به اسکریپت‌ها اجازه می‌دهد که هر کدام از عناصر سند، و نه فقط عناصر خاصی مانند فرم‌ها و تصاویر، را بخوانند و یا تغییر دهند.

۲٫۵٫۱ یافتن عناصر بر اساس شناسه

هنگامی که سندی ایجاد می‌کنید که می‌خواهید عناصر خاصی از آن را با استفاده از اسکریپت تغییر دهید، می‌توانید به هر عنصر خاص یک خصلت id (شناسه) با مقدار منحصر به فرد بدهید. بعد می‌توانید برای دست یافتن به عنصر بر اساس شناسه‌ی آن، از روش getElementById() متعلق به شیء سند استفاده کنید:

```
<h1 id="title">Title</h1>
<script>
var t = document.getElementById("title");
</script>
```

۲٫۵٫۲ یافتن عناصر بر اساس نام برگه

راه دیگر برای دستیابی به عناصر سند، استفاده از نام برگه است. روش getElementsByTagName() متعلق به شیء سند، آرایه‌ای از تمام عناصر سند با آن نام را بر می‌گرداند. هر کدام از عناصر سند نیز همان روش را دارا است، و بنا بر این، می‌توانید مثلاً عناصری با برگه‌ی مورد نظر را که همگی زیرمجموعه‌ی عنصر دلخواهی هستند، را پیدا کنید:

```
// Get an array of all <ul> tags
var lists = document.getElementsByTagName("ul");
// Find the 3rd <li> tag in the second <ul>
var item = lists[1].getElementsByTagName("li")[2];
```

۲٫۵٫۳ عبور از درخت سند

W3C DOM هر سند را به صورت درختی نمایش می‌دهد. گره‌های این درخت نشان دهنده‌ی برگه‌های HTML، رشته‌های متن، و توضیحات HTML موجود در سند هستند. هر گره به وسیله‌ی یک شیء جاوا اسکریپت نمایش داده می‌شود که

مانند متن نمونه‌ی زیر، امکان حرکت کردن در درخت را برای شما فراهم می‌آورد:

```
// Look up a node in the document
var n = document.getElementById("mynode");
var p = n.parentNode;  // The containing tag
var c0 = n.firstChild;    // First child of n
var c1 = c0.nextSibling;  // 2nd child of n
var c2 = n.childNodes[2]; // 3rd child of n
var last = n.lastChild;   // last child of n
```

برای جزئیات بیشتر به قسمت مرجع مراجعه فرمایید.

خود شیء سند نیز نوعی گره است، و همان خصلت‌ها را دارا است. خصلت documentElement از شیء سند، به عنصر منفرد برگه‌ی <html> در سطح بالای هر سند HTML اشاره دارد، و خصلت body به برگه‌ی <body> اشاره می‌کند.

۲/۵/۴ انواع گره‌ها

هر گره موجود در درخت سند، یک خصلت nodeType دارد که نوع آن گره را تعیین می‌کند. انواع مختلف گره‌ها، زیرکلاس‌های متفاوت شیء گره هستند. بعضی از انواع گره‌ها که برای برنامه‌نویسانی که با HTML کار می‌کنند، اهمیت دارند، ذیلاً ذکر شده‌اند (انواع گره دیگری هم برای سندهای XML وجود دارند):

نوع گره	معنا
۱	عنصر: یک برگه‌ی HTML
۲	متن: متن موجود در سند
۸	توضیح: یک توضیح HTML
۹	سند: سند HTML

در گرهی که از نوع عنصر است، خصلت nodeName نام برگه‌ی مربوطه را نشان می‌دهد. در گره‌هایی که از نوع متن یا توضیح هستند، خصلت nodeValue متن سند یا متن توضیح را نشان می‌دهد. برای جزئیات بیشتر به مباحث مربوط به عنصر، متن، توضیح، و سند در قسمت مرجع مراجعه فرمایید. همچنین، برای اطلاع بیشتر در باره‌ی خصلت‌ها و روش‌های مشترک آنها به مبحث گره در قسمت مرجع مراجعه فرمایید.

۲/۵/۵ صفت‌های HTML

به طوری که در بالا دیدیم، برگه‌های HTML در یک درخت سند با اشیایی از نوع عنصر نمایش داده می‌شوند. در سندهای HTML، هر شیء عنصر خصلت‌هایی دارد که دقیقا متناظر با صفت‌های برگه‌ی HTML هستند. برای مثال، برای خواندن یا

نوشتن مقدار صفت caption از یک برگه‌ی <table>، می‌توانید از خصلت caption شیء عنصر متناظر با آن برگه استفاده کنید. برای جزئیات بیشتر به مبحث مربوط به عنصر در قسمت مرجع مراجعه فرمایید.

۲٫۵٫۶ کار با عناصر سند

یک راه ساده برای تغییر دادن سندهای HTML با W3C DOM، تعیین مقدار خصلت‌های متناظر با صفات HTML است. به طوری که در DOM قدیمی دیدیم، به عنوان مثال می‌توان مقدار خصلت src عنصری را که متناظر با یک برگه‌ی img است، تغییر داد. یک راه بسیار مؤثر برای تغییر عناصر سند، از طریق خصلت style است، که شیوه‌های CSS را کنترل می‌کند. بعداً این موضوع مهم را با جزئیات بیشتری بررسی خواهیم کرد.

۲٫۵٫۷ تغییر متن سند

برای تغییر محتوای متنی سند، کافی است که مقدار خصلت nodeValue را در یک گره متنی تغییر دهید:

```
// Find the first <h1> tag in the document
var h1 = document.getElementsByTagName("h1")[0];
// Set new text of its first child
h1.firstChild.nodeValue = "New heading";
```

علاوه بر تغییر خصلت nodeValue، شیء متن اجازه‌ی تغییر دادن خصلت data را نیز می‌دهد، و همچنین، روش‌هایی برای درج کردن، حذف کردن، افزودن، و یا جایگزین کردن متن دارد.

دقت کنید که مشکل متن مذکور این است که فرض می‌کند که محتوای برگه‌ی <h1> متن ساده است. حال اگر مثلاً محتوای برگه به صورت زیر باشد، این متن موفق نخواهد بود، زیرا در اینجا متن عنوان نوه‌ی برگه‌ی <h1> است، نه فرزند مستقیم آن:

```
<h1><i>Original Heading</i></h1>
```

یک راه برای حل این مسئله، استفاده از خصلت innerHTML گره عنوان است. این خصلت بخشی از IE4 DOM است، نه W3C DOM، ولی از آنجا که بسیار مفید است، در بسیاری از مرورگرهای مدرن پشتیبانی می‌شود. وقتی دوباره به بررسی IE4 DOM بپردازیم، مجدداً با این خصلت رو به رو خواهیم شد. راه دیگر برای حل این مسئله، جایگزین کردن گره عنوان با یک برگه‌ی جدید <h1> و گره متن با متن دلخواه است، که این راه را در قسمت بعد می‌بینیم.

۲/۵/۸ تغییر ساختار سند

علاوه بر تغییر متن سند و صفات عناصر سند، W3C DOM امکان تغییر ساختار درخت خود سند را نیز می‌دهد. برای این کار از روش‌های گره که امکان درج، پیوست، حذف، و جایگزین کردن فرزندان یک گره را می‌دهند، و روش‌های سند که امکان ایجاد گره‌های عنصر و متن جدید را فراهم می‌آورند، انجام می‌شود. متن زیر نشان دهنده‌ی این مطلب است:

```
// Find a <ol> element by name:
var list = document.getElementById("mylist");
// Create a new <li> element
var item = document.createElement("li");
// Append it to the list
list.appendChild(item);
// Create a Text node
var text = document.createTextNode("new item");
// Append it to the new <li> node
item.appendChild(text);
// Remove the new item from the list
list.removeChild(item);
// Place the new item at the start of the list
list.insertBefore(item, list.firstChild);
```

باز به عنوان یک مثال دیگر، متن زیر یک تابع جاوا اسکریپت را نشان می‌دهد که از W3C DOM برای پررنگ کردن یک گره دلخواه سند استفاده می‌کند؛ این تابع برای گره مذکور والد جدیدی از نوع <b> ایجاد می‌کند:

```
function embolden(node) {  // Embolden node n
  var b = document.createElement("b");
  var p = n.parentNode; // Get parent of n
  p.replaceChild(b, n); // Replace n with <b>
  b.appendChild(n);     // Insert n into <b> tag
}
```

۲/۶ DOM اینترنت اکسپلورر ۴

IE4 DOM در روایت ۴ مرورگر اینترنت اکسپلورر میکروسافت عرضه شد. این نوع یک DOM قوی ولی استاندارد نشده است که قابلیت‌های آن مشابه W3C DOM است. روایت‌های ۵ و بالاتر اینترنت اکسپلورر اکثر ویژگی‌های پایه‌ی W3C DOM را پشتیبانی می‌کنند، لیکن به علت آنکه هنوز هم از IE4 DOM به طور شایع استفاده می‌شود، بحث زیر را در مورد آن ارائه می‌کنیم. در قسمت زیر، IE4 DOM از نظر تفاوت‌هایی که با W3C DOM دارد، مورد بحث قرار می‌گیرد، لذا لازم است که قبلاً با W3C DOM آشنایی داشته باشید.

۲/۶/۱ دستیابی به عناصر سند

IE4 DOM از روش getElementById() پشتیبانی نمی‌کند. در عوض، برای یافتن عناصر دلخواه سند باید از آرایه‌ی all[] متعلق به شیء سند استفاده کنید:

```
var list = document.all["mylist"];
list = document.all.mylist;  // this also works
```

IE4 DOM به جای پشتیبانی از روش getElementsByTagName()، کاری غیرعادی می‌کند و آن اینکه روی آرایه‌ی all[]، یک روش tags() تعریف می‌کند، که این هم برای عناصر سند و هم برای خود شیء سند وجود دارد. در اینجا نحوه‌ی یافتن برگه‌های <li> را در داخل اولین برگه‌ی <ul> می‌بینید:

```
var lists = document.all.tags("UL");
var items = lists[0].all.tags("LI");
```

دقت کنید که برای روش all.tags() باید نام برگه‌ی HTML را با حروف بزرگ وارد کنید.

۲/۶/۲ عبور از درخت سند

عبور از درخت سند در IE4 DOM به همان صورت W3C DOM امکان‌پذیر است. تفاوت فقط در نام خصلت‌های مربوطه است: اینترنت اکسپلورر ۴ به جای childNodes[]، خصلت children[] دارد، و به جای parentNode، از خصلت parentElement استفاده می‌کند. اینترنت اکسپلورر چیزی مشابه firstChild، nextSibling، و خصلت‌های مربوطه در W3C DOM ندارد. یک تفاوت عمده بین IE4 DOM و W3C DOM در این است که IE4 DOM فقط حاوی برگه‌های HTML است: توضیحات چشمپوشی می‌شوند، و متن سند نیز بخشی از خود درخت نیست. در عوض، متن موجود در داخل هر عنصری از طریق خصلت‌های innerHTML و innerText شیء عنصر قابل دستیابی است. (در قسمت بعد، اطلاعات بیشتری در باره‌ی innerHTML ارائه خواهد شد.)

۲/۶/۳ تغییر دادن محتوا و ساختار سند

گره‌های یک درخت سند اینترنت اکسپلورر ۴، اشیای عنصر مشابه گره عنصر در W3C DOM هستند. این اشیا هم، مانند گره‌های عنصر در W3C DOM، خصلت‌هایی متناظر با صفات برگه‌های HTML دارند، و می‌توانید مقدار این خصلت‌ها را به دلخواه بخوانید یا بنویسید. برای تغییر محتوای متنی یک عنصر سند، متن مورد نظر را به خصلت innerText آن اختصاص دهید. با این کار هر برگه یا

متنی در داخل آن عنصر پاک می‌شود، و متن معین شده جایگزین آن می‌گردد.

IE4 DOM هیچ روش صریحی برای ایجاد، درج، حذف، و یا جایگزینی گره‌های یک درخت سند ندارد. اما از خصلت بسیار مهم innerHTML پشتیبانی می‌کند، که به شما امکان می‌دهد که محتوای هر عنصر سند را با یک رشته‌ی HTML دلخواه جایگزین کنید. انجام این کار باعث فراخوانی تجزیه‌گر HTML می‌شود، و لذا این روش نسبت به تغییر مستقیم خود گره‌ها کارایی کمتری دارد. با این حال، این خصلت بسیار سودمند است، به حدی که موزیلا، نت‌اسکیپ ۶ و بالاتر، و سایر مرورگرهای مدرن این خصلت را — با آنکه استاندارد نیست — پیاده‌سازی کرده‌اند.

IE4 DOM دارای خصلت مشابه دیگری به نام outerHTML نیز هست که عنصر و محتوای آن را جایگزین می‌کند، و نیز روش‌های insertAdjacentHTML() و insertAdjacentText() دارد. از اینها زیاد استفاده نمی‌شود، و به علاوه، پشتیبانی از آنها در خارج از اینترنت اکسپلورر بر خلاف innerHTML زیاد نیست؛ می‌توانید در باره‌ی آنها در مبحث مربوط به عنصر در قسمت مرجع اطلاعات بیشتری کسب کنید.

۲/۶/۴ سازگاری DOM

اگر بخواهید اسکریپتی بنویسید که در صورت موجود بودن از W3C DOM استفاده کند، و در غیر این صورت از IE4 DOM در صورت فراهم بودن بهره بگیرد، می‌توانید از روش خاصی استفاده کنید که در آن با بررسی موجود بودن یک روش یا خصلت مشخص می‌شود که پشتیبانی از DOM مورد نظر وجود دارد یا نه. برای مثال:

```
if (document.getElementById) {
  // If the W3C method exists, use it
}
else if (document.all) {
  // If the all[] array exists, use it
}
else {
  // Otherwise use the legacy DOM
}
```

۲/۷ DHTML: اسکریپت‌نویسی شیوه‌های CSS

DHTML با HTML پویا حاصل ترکیب کردن HTML، CSS، و جاوا اسکریپت است: از اسکریپت برای تغییر پویای شیوه استفاده می‌شود، که ممکن است شامل مواردی مانند موقعیت و مرئی بودن عناصر سند باشد. در DOMهای

W3C و IE4، هر عنصر سند یک خصلت style دارد که متناظر با صفت style در HTML است و شیوه‌ی مستقیم عنصر را تعیین می‌کند. اما، خصلت style به جای اینکه به رشته‌ی ساده‌ای اشاره داشته باشد، معرف یک شیء شیوه است که خصلت‌هایی متناظر با صفات CSS شیوه دارد.

برای مثال، اگر یک عنصر e یک خصلت style داشته باشد که مقدار صفت CSS رنگ (color) را تعیین کرده باشد، برای خواندن مقدار آن می‌توانید از عبارت e.style.color استفاده کنید. هنگامی که نام یک صفت CSS دارای فاصله است، نام خصلت متناظر آن در جاوا اسکریپت فاقد فاصله است، و از حروف بزرگ مختلط با حروف کوچک استفاده می‌کند. بنا بر این، برای تعیین مقدار صفت CSS background-color در یک عنصر e، از خصلت e.style.backgroundColor استفاده می‌کنیم. یک مورد ویژه نیز وجود دارد: صفت CSS float در جاوا اسکریپت یک واژه‌ی ذخیره شده است، و لذا نام خصلت متناظر آن در جاوا اسکریپت cssFloat است.

استاندارد CSS خصلت‌های زیادی تعریف می‌کند که می‌توانید از آن برای تنظیم منظره‌ی بصری سندهای خود استفاده کنید. لیستی از این خصلت‌ها در جدولی در مبحث شیوه در قسمت مرجع آمده است. خصلت‌های موقعیت و ظهور بالاخص برای اسکریپت‌نویسی پویا کاربرد دارند. اگر به خصلت position مقدار absolute داده شود، می‌توانید از خصلت‌های top و left برای تعیین موقعیت مطلق یک عنصر سند (بر حسب پیکسل، درصد، یا واحدهای دیگر) استفاده کنید. به همین ترتیب، خصلت‌های width و height اندازه‌ی عنصر را تعیین می‌کنند. یک عنصر را می‌توان در آغاز با دادن مقدار hidden به خصلت visibility نامرئی کرد، و سپس در زمان مناسب با دادن مقدار visible آن را مرئی کرد.

دقت کنید که مقدار تمام خصلت‌های شیوه همواره از نوع رشته است، حتی برای خصلت‌هایی مانند left و width که معرف عدد هستند. هنگام تعیین این خصلت‌های طولی و بعدی، باید حتماً اعداد را به رشته تبدیل کنید، و واحد آن را هم به این رشته اضافه کنید (معمولاً از رشته‌ی px که معرف پیکسل است، استفاده می‌شود). در جدول زیر، این خصلت‌های مربوط به موقعیت و نمایش به طور خلاصه معرفی شده‌اند:

خصلت	توصیف/مقادیر
position	چگونگی قرارگیری موقعیت عنصر. absolute، relative، fixed، و یا static (پیش‌فرض).
left, top	مختصات X و Y لبه‌های بالا و چپ عنصر.
width	پهنای عنصر
height	بلندای عنصر.
zIndex	ترتیب پشته‌سازی. مقادیر این خصلت عدد صحیح‌اند. عناصری که مقدار این خصلت آنها بالاتر است، روی عناصری که مقدار پایین‌ترین دارند، ترسیم می‌شوند.
display	چگونگی نمایش عنصر. مقادیر شایع عبارت‌اند از block، inline، و none برای عناصری که اصلاً ترسیم نمی‌شوند.
visibility	مرئی (visible) یا نامرئی (hidden) بودن عنصر. برای عناصر مخفی هم فضا اختصاص داده می‌شود، مگر آنکه موقعیت خاصی به آنها داده شده باشد.
overflow	وقتی محتوای عنصر از اندازه‌ی آن تجاوز کند، چه اتفاقی بیفتد. مقادیر: visible (محتوا سرریز می‌شود)؛ hidden (محتوای مازاد مخفی است)؛ scroll (نوارپیمای همیشگی نشان داده شود)؛ و auto (نوارپیما فقط در صورت لزوم).
clip	چه بخشی از محتوای عنصر نمایش داده شود. دستور: rect(top right bottom left)

متن زیر یک پویانمایی ساده‌ی DHTML را نشان می‌دهد. هر بار فراخوانی می‌شود، تابع nextFrame() یک عنصر را ۱۰ پیکسل به طرف راست می‌برد و از setTimeout() استفاده می‌کند تا به جاوا اسکریپت بگوید که آن را ۵۰ میلی‌ثانیه بعد دوباره فراخوانی کند.پس از ۲۰ بار فراخوانی، تابع با استفاده از خصلت visibility، عنصر را مخفی می‌کند و دست از فراخوانی مجدد خود می‌کشد.

```
<h1 id='title'>DHTML Animation<h1>
<script>
// Look up the element to animate
var e = document.getElementById("title");
// Make it position-able.
e.style.position = "absolute";
var frame = 0;  // Initialize frame counter.
// This function moves the element one frame
// at a time, then hides it when done.
function nextFrame() {
  if (frame++ < 20) { // Only do 20 frames
    e.style.left = (10 * frame) + "px";
    // Call ourselves again in 50ms.
    setTimeout("nextFrame()", 50);
  }
  else e.style.visibility="hidden"; // Hide it.
}
nextFrame();  // Start animating now!
</script>
```

۲٫۸ رویدادها و رویدادپردازی

در آغاز این قسمت دیدیم که یک راه برای قرار دادن جاوا اسکریپت سمت مشتری در درون سندهای HTML، استفاده از صفت‌های رویدادپرداز برگه‌های HTML است. در جدول زیر، لیست صفت‌های رویدادپرداز استاندارد HTML و برگه‌هایی که این صفت‌ها برای آنها قابل استفاده هستند، ذکر شده است. در ستون اول جدول، نام صفت رویدادپرداز آمده است: این نام‌ها همیشه با on شروع می‌شوند. در ستون دوم جدول، برگه‌های HTMLای که این صفت‌ها را می‌توان برای آنها به کار برد، ذکر شده است. در ضمن، در صورت لزوم ذکر شده است که چه رویدادهایی متن رویدادپرداز را فراخوانی می‌کنند.

رویدادپرداز	برگه‌های پشتیبانی کننده/ زمان فراخوانی
onabort	<img>؛ قطع بار شدن تصویر
onblur	<body> و عنصرهای فرم؛ منحرف شدن کانون توجه صفحه‌کلید از پنجره یا عنصر
onchange	عنصرهای فرم؛ تغییر مقدار نمایش داده شده
onclick	تمام عنصرها؛ فشار داده شدن و رها شدن دکمه‌ی موشواره؛ برای لغو، باید مقدار false بر گرداند
ondblclick	تمام عنصرها؛ دوکلیک موشواره
onerror	<img>؛ ناموفق بودن بار شدن تصویر
onfocus	<body> و عنصرهای فرم؛ منتقل شدن کانون توجه صفحه‌کلید به پنجره یا عنصر
onkeydown	<body> و عنصرهای فرم؛ فشار داده شدن کلید؛ برای لغو، باید مقدار false بر گرداند
onkeypress	<body> و عنصرهای فرم؛ فشار داده شدن و رها شدن کلید؛ برای لغو، باید مقدار false بر گرداند
onkeyup	<body> و عنصرهای فرم؛ رها شدن کلید
onload	<body>، <frameset>، <img>، <iframe>، <object>؛ بار شدن کامل سند، تصویر، یا شیء
onmousedown	تمام عنصرها؛ فشار داده شدن دکمه‌ی موشواره
onmousemove	تمام عنصرها؛ حرکت نشانگر موشواره
onmouseout	تمام عنصرها؛ خارج شدن موشواره از روی عنصر
onmouseover	تمام عنصرها؛ وارد شدن موشواره به روی عنصر؛ برای جلوگیری از نمایش نشانی پیوند در سطر وضعیت پایین مرورگر، باید مقدار true بر گرداند
onmouseup	تمام عنصرها؛ رها شدن دکمه‌ی موشواره
onreset	<form>؛ تقاضای پاک کردن فرم؛ برای جلوگیری از پاک کردن، باید مقدار false بر گرداند
onresize	<body>، <frameset>؛ تغییر اندازه‌ی پنجره
onsubmit	<form>؛ تقاضای تحویل فرم؛ برای جلوگیری از تحویل، باید مقدار false بر گرداند
onunload	<body>، <frameset>؛ خالی شدن سند

دقت کنید که وقتی مرورگر بعضی از رویدادپردازها، از قبیل onclick، onmouseover، و onsubmit، را فراخوانی می‌کند، به مقدار برگشتی رویدادپرداز (در صورت وجود) نگاه می‌کند تا ببیند آیا باید عمل پیش‌فرض مربوط به رویداد را انجام دهد یا نه. به طور معمول، اگر یک رویدادپرداز مقدار false بر گرداند، عمل پیش‌فرض (مانند رفتن به یک پیوند یا تحویل دادن یک فرم) انجام نمی‌شود. استثنای این مطلب، رویدادپرداز onmouseover است: وقتی موشواره روی یک پیوند حرکت می‌کند، مرورگر نشانی پیوند را در سطر وضعیت خود نمایش می‌دهد مگر آن که رویدادپرداز مقدار true بر گرداند.

۲/۸/۱ رویدادپردازها به عنوان تابع‌های جاوا اسکریپت

دیدیم که در انواع مختلف مدل شیء سند (DOM)، برگه‌های HTML به عنوان اشیای جاوا اسکریپت نمایش داده می‌شوند، و صفات برگه به عنوان خصلت‌های شیء منظور می‌گردند. اگر سند شما فقط حاوی یک برگه‌ی <form> با یک صفت رویدادپرداز onsubmit باشد، رویدادپرداز مذکور به صورت زیر قابل دسترس است:

```
document.forms[0].onsubmit
```

گرچه صفات رویدادپرداز HTML به عنوان رشته‌های متن جاوا اسکریپت نوشته می‌شوند، ولی، در واقع، مقدار خصلت جاوا اسکریپت متناظر با آنها از نوع رشته‌ی متن نیست، بلکه تابع است. لذا برای ایجاد یک رویدادپرداز جدید کافی است که یک تابع را به عنوان مقدار به خصلت مربوطه اختصاص دهید:

```
function validate() { // Form validation function
  // check validity here
  return valid;       // return true or false
}
// Now check user input before submitting it
document.forms[0].onsubmit = validate;
```

۲/۸/۲ رویدادپردازی پیشرفته

در قسمت‌های قبل، مدل اساسی رویدادپردازی در جاوا اسکریپت سمت مشتری مورد بحث قرار گرفت. رویدادپردازی ویژگی‌های پیشرفته‌ای نیز دارد، ولی متأسفانه سه مدل رویداد وجود دارد که با هم ناسازگارند: مدل استاندارد W3C، مدل اینترنت اکسپلورر (میکروسافت استاندارد W3C را نپذیرفته است)، و مدل نت‌اسکیپ ۴. این مدل‌های رویداد خیلی پیچیده‌اند، لذا در قسمت زیر ما تنها خلاصه‌ای از ویژگی‌های پیشرفته‌ی این مدل‌ها را بیان می‌کنیم. برای جزئیات بیشتر به کتاب‌های مفصل‌تر مراجعه کنید.

- **جزئیات رویداد**

در مدل‌های پیشرفته‌ی رویدادپردازی، جزئیات رویداد، از قبیل نوع رویداد، وضعیت دکمه‌ها و مختصات موشواره، حالت کلیدهای تغییرگر، و غیره، به عنوان خصلت‌های شیء رویداد در اختیار ما قرار می‌گیرد. در مدل‌های رویداد W3C و نت‌اسکیپ، این شیء رویداد به عنوان آوند به رویدادپرداز داده می‌شود. در مدل اینترنت اکسپلورر، شیء رویداد به عنوان آوند داده نمی‌شود، بلکه در خصلت `event` پنجره‌ای که رویداد در آن روی می‌دهد، ذخیره می‌شود. متأسفانه در این سه مدل پیشرفته از نام‌های متفاوتی برای خصلت‌های رویداد استفاده می‌شود، و بنا بر این، سازگاری بین مرورگرهای مختلف مشکل است. برای اطلاعات بیشتر در باره‌ی شیء رویداد در هر یک از این سه مدل به مبحث رویداد در قسمت مرجع مراجعه فرمایید.

- **انتشار رویداد**

در مدل پایه‌ی رویدادها، رویدادپردازها فقط برای عنصری از سند که رویداد روی آن بروز کرده است، فرا خوانده می‌شوند. اما در مدل‌های پیشرفته، رویدادها می‌توانند در درخت عناصر سند به طرف بالا یا پایین عنصر مربوطه انتشار پیدا کنند، و به وسیله‌ی بیش از یک رویدادپرداز، مورد پردازش قرار گیرند. در مدل‌های نت‌اسکیپ و W3C، رویدادها از شیء سند شروع می‌شوند و در درخت عناصر سند پایین می‌آیند تا به عنصری برسند که رویداد برای آن روی داده است. اگر هر کدام از عناصر بالاتر رویدادپردازهای ثبت شده‌ای داشته باشند، این رویدادپردازها رویداد را می‌گیرند و اولین قدم را در پردازش آن بر می‌دارند. در مدل‌های اینترنت اکسپلورر و W3C، برخی از انواع رویدادها، مانند کلیک موشواره، بعد از اینکه در عنصر منبع مورد پردازش قرار گرفتند، حباب‌وار به قسمت‌های بالاتر سند سرایت می‌کنند. به این ترتیب، برای پردازش تمام کلیک‌های موشواره‌ای که روی عناصر درون یک عنصر `<div>` به وقوع می‌پیوندند، می‌توانید یک رویدادپرداز برای آن عنصر به ثبت برسانید. تمام انواع رویدادپرداز ارسالی، حبابی، و طبیعی این توانایی را دارند که مانع از انتشار بیشتر رویداد شوند، ولی روش انجام این کار در هر مدل متفاوت است.

- **ثبت رویدادپرداز**

در مدل رویداد W3C، رویدادپردازها صرفاً به عنوان مقدار به خصلت‌های اشیای سند اختصاص داده نمی‌شوند. بلکه در این مدل، هر شیء سند یک روش addEventListener() دارد که با فراخوانی آن می‌توانید یک تابع رویدادپرداز را برای رویداد مورد نظر خود به ثبت برسانید. به این طریق، در برنامه‌های پیشرفته می‌توان برای یک رویداد، چند رویدادپرداز به ثبت رسانید.

۲٫۹ محدودیت‌های امنیتی جاوا اسکریپت

به دلایل امنیتی، معمولاً در پیاده‌سازی‌های جاوا اسکریپت سمت مشتری محدودیت‌هایی بر اعمالی که اسکریپت‌ها می‌توانند انجام دهند، اعمال می‌شود. واضح‌ترین محدودیت‌ها حذف قابلیت‌های خطرناک است: مثلاً، اسکریپت‌های سمت مشتری به هیچ طریقی نمی‌توانند پرونده‌ای را از روی دیسک محلی یک کاربر پاک کنند. محدودیت‌های دیگری نیز برای جلوگیری از افشاسازی اطلاعات خصوصی و یا جلوگیری از آزار کاربران توسط اسکریپت‌ها وضع شده‌اند. البته لیست استانداردی از محدودیت‌های امنیتی وجود ندارد، ولی موارد زیر محدودیت‌هایی هستند که در پیاده‌سازی‌های معمول مرورگرها یافت می‌شوند. تلاش نکنید اسکریپت‌هایی بنویسید که این کارها را انجام بدهند: اگر هم این اسکریپت‌ها در مرورگر خودتان کار کنند، ممکن است در مرورگرهای دیگر با مشکل مواجه شوند.

- **سیاست مبدأ یکسان**

اسکریپت‌ها فقط خصلت‌های پنجره‌ها و سندهایی را که از سرور شبکه‌ی یکسانی بار شده باشند، می‌توانند بخوانند. این محدودیت غلاظ و شدادی برای اسکریپت‌نویسی بین‌پنجره‌ای است، و مانع از آن می‌شوند که اسکریپت‌ها به مطالب سندهای نامربوط دیگری که کاربر می‌خواند، دسترسی پیدا کنند. علاوه بر این، این محدودیت مانع از به ثبت رساندن رویدادپردازها و رویدادهای کلاهبردارانه روی سندهای نامربوط می‌شود.

- **فراگذاری پرونده‌ها**

اسکریپت‌ها قادر نیستند مقدار خصلت value را در عنصر فرم FileUpload تعیین کنند.

- **فرستادن نامه‌ی الکترونیک و خبر**
اسکریپت‌ها نمی‌توانند بدون تأیید کاربر فرمی را برای نشانی‌های mailto: یا news: بفرستند.

- **بستن پنجره‌ها**
یک اسکریپت فقط می‌تواند پنجره‌های مرورگری را که خود ایجاد کرده است، ببندد، مگر اینکه تأیید کاربر را بگیرد.

- **سرک کشیدن به حافظه‌ی نهانی**
اسکریپت نمی‌تواند نشانی‌های about:، از قبیل about:cache را بخواند.

- **پنجره‌های پنهان و تزیینات پنجره**
اسکریپت نمی‌تواند پنجره‌های کوچک یا خارج از صفحه‌ی نمایش و یا پنجره‌های بدون نوار عنوان ایجاد کند.

دقت کنید که لیست محدودیت‌های امنیتی ثابت نیست. با افزایش استفاده از جاوا اسکریپت، آگهی دهندگان و اشخاص موذی کم‌کم شروع به استفاده‌های موذیانه از آن کرده‌اند. در نتیجه، مرورگرهای جدید مانند موزیلای ۱٫۰، امکانات قابل تنظیم توسط کاربری برای محدودیت‌های اسکریپتی دارند که مانع از باز کردن پنجره‌های جدید (مانند پنجره‌های آگهی) توسط اسکریپت و یا حرکت دادن یا تغییر اندازه‌ی پنجره‌های موجود می‌شوند.

۳ مرجع جاوا اسکریپت

قسمت باقیمانده‌ی این کتاب حاوی مرجع سریعی برای رابط برنامه‌نویسی کاربردی جاوا اسکریپت هسته و سمت مشتری است. در اینجا، توابع جاوا اسکریپت هسته به طور کامل و DOM قدیمی (سطح ۰) مورد بررسی قرار می‌گیرد، و DOM سطح ۲ W3C نیز به طور ساده شده ارائه می‌شود. قسمت‌هایی از رابط برنامه‌نویسی مذکور که به برنامه‌نویسان جاوا اسکریپت در سندهای HTML مربوط نمی‌شود، حذف شده است. در آغاز هر مبحث، مشخص شده است که آیا جزئی از جاوا اسکریپت هسته است یا جاوا اسکریپت سمت مشتری، و اینکه ویژگی مورد نظر در چه روایتی از جاوا اسکریپت، چه مرورگرهایی، و چه نسخه‌ای از DOM ارائه شده است.

از آنجا که جاوا اسکریپت زبانی با نوع‌های ضعیف است، لذا کلاس‌ها و اشیای موجود در رابط برنامه‌نویسی جاوا اسکریپت لیست رسمی و مشخصی ندارد. لیست درایه‌هایی که بحث خواهند شد، ذیلاً درج شده است. می‌توانید مبحث مورد نظر خود را در این لیست به سهولت پیدا کنید.

Anchor	موقعیت خاصی در سند همراه با یک نام
Applet	یک برنامک جاوا
Arguments	آوندهای یک تابع
Array	ایجاد و عملیات آرایه
Attr	صفت یک عنصر سند
Boolean	یک شیء لفافه برای مقادیر بولی
Comment	یک توضیح HTML
DOMException	خطای DOM را نشان می‌دهد
DOMImplementation	سند ایجاد می‌کند، و ویژگی‌های DOM را بررسی می‌کند
Date	انجام عملیات تاریخ و زمان
Document	یک سند HTML
DocumentFragment	عملیات بر روی تعدادی از گره‌هایی با هم
Element	یک برگه‌ی HTML در یک سند
Error	انواع استثنای از پیش تعریف شده
Event	جزئیات رویداد
Form	یک فرم ورودی HTML
Function	یک تابع جاوا اسکریپت
Global	خصلت‌ها و توابع سراسری
History	سابقه‌ی مرور
Image	یک تصویر HTML
Input	یک عنصر ورودی در فرم
Layer	یک لایه‌ی مستقل در سند
Link	پیوند از نوع <a> یا <area>
Location	نشانی فعلی مرورگر
Math	توابع و ثابت‌های ریاضی
Navigator	اطلاعات در باره‌ی مرورگر
Node	گرهی در درخت سند
Number	پشتیبانی از اعداد
Object	کلاس پایه‌ی همه‌ی اشیا در جاوا اسکریپت
Option	یک گزینه‌ی قبل انتخاب
RegExp	عبارت مرتب برای انطباق الگو
Screen	اطلاعات در باره‌ی صفحه‌ی نمایش
Select	یک لیست نمایش گزینه‌ها
String	کار با رشته‌ها
Style	خواص مستقیم CSS یک عنصر HTML
Text	قطعه‌ای از متن در یک سند
Textarea	ورودی متن چندسطری
Window	پنجره یا کادر مرورگر

Anchor ۳/۱

(جاوا اسکریپت سمت مشتری ۱/۲)
یک محل نامگذاری شده در سند.
مشتق شده از: Element

۳/۱/۱ دستور

```
document.anchors[index]
document.anchors[name]
```

۳/۱/۲ شرح

شیء Anchor نشان دهنده‌ی یک برگه‌ی <a> با صفت name است، که برای ایجاد یک محل نامگذاری شده در سند به کار می‌رود.

۳/۱/۳ خصلت‌ها

- **name**

مقدار صفت name از برگه‌ی <a>.

۳/۱/۴ ارجاع

Document.anchors[]، Link، Location.hash

Applet ۳/۲

(جاوا اسکریپت سمت مشتری ۱/۱)
یک برنامک جاوا.

۳/۲/۱ دستور

```
document.applets[i]
document.applets[appletName]
document.appletName
```

۳/۲/۲ خصلت‌ها و روش‌ها

خصلت‌ها و روش‌های یک شیء برنامک، همان فیلدها و روش‌های عمومی برنامک جاوایی هستند که این شیء معرف آن است. متن جاوا اسکریپت می‌تواند

فیلدهای جاوا را بخواند یا بنویسد، و روش‌های جاوای برنامک را فراخوانی کند.

۳٫۳ Arguments

(جاوا اسکریپت هسته ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱)
آوندهای یک تابع.

۳٫۳٫۱ دستور

```
arguments[n]
arguments.length
```

۳٫۳٫۲ شرح

شیء آوندها فقط در درون بدنه‌ی یک تابع تعریف شده است، و در درون بدنه‌ی هر تابع، متغیر arguments به شیء Arguments مربوط به آن تابع اشاره دارد. شیء آوندها، آرایه‌ای است که عناصر آن مقادیری هستند که به عنوان آوند به تابع داده شده‌اند. عنصر ۰ آوند اول است، عنصر ۱ آوند دوم، و الخ. تمام مقادیری که به عنوان آوند به تابع داده شده‌اند، تبدیل به عنصرهای شیء آوندها می‌شوند، ولو آنکه در تعریف تابع به این آوندها نامی داده نشده باشد.

۳٫۳٫۳ خصلت‌ها

- **callee**

اشاره‌ای به تابعی که هم‌اکنون در حال اجرا است. برای تراجع در توابع بی‌نام مفید است. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۱؛ فقط در درون بدنه‌ی یک تابع تعریف شده است.

- **length**

تعداد آوندهای داده شده به تابع. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۲، ECMA ۱؛ فقط در درون بدنه‌ی یک تابع تعریف شده است.

۳٫۳٫۴ ارجاع

Function.

Array ۳٫۴

(جاوا اسکریپت هسته ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱)
ایجاد آرایه و کار کردن با آن.

۳٫۴٫۱ سازنده

```
new Array()             // empty
new Array(n)            // n undefined elements
new Array(e0, e1, ...)   // specified elements
```

۳٫۴٫۲ دستور مستقیم

در جاوا اسکریپت ۱٫۲، جی‌اسکریپت ۳٫۰، و ECMA ۳ می‌توانید با قرار دادن لیستی از عبارت‌ها در درون کروشه که با ویرگول از هم جدا شده‌اند، یک آرایه را ایجاد و آغازش کنید. مقادیر این عبارت‌ها، عناصر آرایه خواهند بود. برای مثال:

```
var a = [1, true, 'abc'];
var b = [a[0], a[0]*2, f(x)];
```

۳٫۴٫۳ خصلت‌ها

- **length**

یک عدد صحیح خواندنی/نوشتنی که تعداد عناصر آرایه را نشان می‌دهد، و یا، در مواردی که عناصر آرایه مجاور هم نیستند، عددی را نشان می‌دهد که یک واحد از آخرین عنصر موجود در آرایه بالاتر است. تغییر دادن مقدار این خصلت، آرایه را از وسط قطع کرده و یا اینکه آن را بسط می‌دهد.

۳٫۴٫۴ روش‌ها

- **concat(*value*, ...)**

آرایه‌ی جدیدی بر می‌گرداند که از ادغام هر یک از آوندهای داده شده با این آرایه تشکیل می‌شود. اگر هر کدام از آوندهای `concat()` خود یک آرایه باشد، عناصر آن به این آرایه ملحق می‌شوند، نه خود آن. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **join(*separator*)**

رشته‌ای را بر می‌گرداند که از تبدیل هر یک از عناصر آرایه به رشته و

سپس به هم پیوستن همه‌ی آنها با استفاده از `separator` به عنوان جدا کننده در بین آنها، حاصل می‌شود.

- **`pop()`**

آخرین عنصر آرایه را بر می‌دارد و بر می‌گرداند، و طول آرایه را یک واحد کم می‌کند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **`push(value, ...)`**

مقدار یا مقادیر داده شده را به انتهای آرایه ملحق می‌کند، و طول جدید آرایه را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **`reverse()`**

ترتیب عناصر آرایه را معکوس می‌کند. چیزی بر نمی‌گرداند.

- **`shift()`**

اولین عنصر آرایه را حذف می‌کند، و بر می‌گرداند، و هر کدام از عناصر باقیمانده را یک شماره جلوتر می‌آورد، و طول آرایه را یک واحد کاهش می‌دهد. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **`slice(start, end)`**

آرایه‌ی جدیدی بر می‌گرداند که حاوی عناصر آرایه از عنصر شماره‌ی *`start`* تا عنصر شماره‌ی *`end`* (به استثنای خود آن عنصر) است. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **`sort(orderfunc)`**

عناصر آرایه را مرتب می‌کند، و اشاره‌ای به آن آرایه را بر می‌گرداند. دقت کنید که خود آرایه مرتب می‌شود، نه اینکه نسخه‌ی جدیدی از آن ساخته شود. آوند اختیاری *`orderfunc`* می‌تواند تابعی را برای تعیین چگونگی مرتب‌سازی ارائه نماید. تابع مذکور باید دو آوند بگیرد، و در صورتی که آوند اول کمتر از دومی باشد، مقداری کمتر از ۰ بر گرداند، اگر هر دو برابر باشند، ۰ بر گرداند، و اگر آوند اول بزرگ‌تر از دومی باشد، مقداری بالاتر از ۰ بر گرداند.

- **`splice(start, deleteCount, value, ...)`**

تعداد مشخص شده از عناصر آرایه را با شروع از شماره‌ی تعیین شده حذف می‌کند، و بعد آوندهای باقیمانده را در آن محل در آرایه درج

می‌کند. آرایه‌ای بر می‌گرداند که حاوی عناصر حذف شده است. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toLocaleString()**

نمایش رشته‌ای محلی شده‌ی آرایه را بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۱.

- **toString()**

نمایش رشته‌ای آرایه را بر می‌گرداند.

- **unshift(*value*, ...)**

آوند یا آوندهای داده شده را در آغاز آرایه درج می‌کند، و عناصر قبلی را جابجا می‌کند، تا برای اینجا جا باز شود. طول جدید آرایه را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

۳٫۵ Attr

(DOM سطح ۱)
صفت یک عنصر HTML.
مشتق شده از: Node

۳٫۵٫۱ خصلت‌ها

- **name**

نام صفت. فقط-خواندنی.

- **ownerElement**

شیء عنصری که حاوی این صفت است. فقط-خواندنی. DOM سطح ۲.

- **specified**

در صورتی که مقدار صفت به صراحت در متن سند و یا توسط یک اسکریپت تعیین شده باشد، مقدار آن true است، و در غیر این صورت، مقدار آن false است. فقط-خواندنی.

- **value**

مقدار صفت به صورت یک رشته. خواندنی/نوشتنی.

۳٫۵٫۲ ارجاع

Document.createAttribute()،
Element.getAttributeNode()،
Element.setAttributeNode().

۳٫۶ Boolean

(جاوا اسکریپت هسته ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱)
شیء لفافه برای مقادیر بولی.

۳٫۶٫۱ سازنده

```
new Boolean(value)
Boolean(value)
```

وقتی Boolean() به عنوان یک تابع، بدون عملگر new، فرا خوانده شود، مقدار داده شده را به یک مقدار بولی (نه یک شیء Boolean) تبدیل می‌کند، و آن را بر می‌گرداند. تمام مقادیر تبدیل به مقدار true می‌شوند، به جز ۰، NaN، null، undefined، و رشته‌ی تهی، "". وقتی سازنده‌ی Boolean() با عملگر new فرا خوانده شود، همان کار را انجام می‌دهد، و حاصل را در یک شیء Boolean می‌پیچد.

۳٫۶٫۲ روش‌ها

- **toString()**

بسته به مقدار شیء بولی، مقدار "true" یا "false" بر می‌گرداند.

- **valueOf()**

مقدار بولی بدوی را که درون شیء بولی پیچیده شده است، بر می‌گرداند.

۳٫۷ Comment

(DOM سطح ۲)
یک توضیح HTML.
مشتق شده از: Node

۳٫۷٫۱ خصلت‌ها

گره‌های توضیح دقیقاً دارای همان خصلت‌های گره‌های متن هستند.

۳٫۷٫۲ روش‌ها

گره‌های توضیح تمام روش‌های گره‌های متن را، به جز `splitText()`، پشتیبانی می‌کنند.

۳٫۷٫۳ ارجاع

`Text`.

۳٫۸ DOMException

(DOM سطح ۱)
خطای DOM را نشان می‌دهد.

۳٫۸٫۱ خصلت‌ها

- **code**

رمز خطا که جزئیاتی را در باره‌ی آنچه سبب بر انگیخته شدن استثنا شده است، نشان می‌دهد. بعضی از مقادیر ممکنه‌ی این خصلت (و معنای آنها) توسط ثابت‌های ذکر شده در زیر تعریف شده‌اند.

۳٫۸٫۲ ثابت‌ها

ثابت‌های زیر مقادیر `code` را که ممکن است در کار با سندهای HTML با آنها مواجه شوید، نشان می‌دهند. دقت کنید که این ثابت‌ها خصلت‌های ایستای `DOMException` هستند، نه خصلت‌های اشیای نمونه‌ی حاصل از آن.

- **DOMException.INDEX_SIZE_ERR = 1**

خطای خروج از کران در مورد نمایه‌ی آرایه یا رشته.

- **DOMException.HIERARCHY_REQUEST_ERR = 3**

تلاش برای گذاشتن یک گره در یک محل غیرقانونی در درخت انشعابات سند.

- **DOMException.WRONG_DOCUMENT_ERR = 4**
تلاش برای استفاده از گره در سندی غیر از سندی که آن را ساخته است.
- **DOMException.INVALID_CHARACTER_ERR = 5**
نویسه‌ای غیرقانونی (مثلاً در نام یک عنصر) یافت شده است.
- **DOMException.NOT_FOUND_ERR = 8**
یک گره در جایی که باید می‌بود، یافت نشد.
- **DOMException.NOT_SUPPORTED_ERR = 9**
یک روش یا خصلت در پیاده‌سازی کنونی DOM پشتیبانی نمی‌شود.
- **DOMException.INUSE_ATTRIBUTE_ERR = 10**
تلاش برای ربط دادن یک `Attr` به یک `Element` در شرایطی که قبلاً به گره `Element` دیگری ربط داده شده است.
- **DOMException.SYNTAX_ERR = 12**
خطای دستوری، مثلاً در تعیین خصلت CSS.

۳٫۹ DOMImplementation

(DOM سطح ۱)
سند می‌سازد و خصوصیات DOM را وارسی می‌کند.

۳٫۹٫۱ دستور

document.implementation

۳٫۹٫۲ روش‌ها

- **createHTMLDocument(*title*)**
یک شیء جدید از نوع سند HTML ایجاد می‌کند، و در آن عناصر <html>، <head>، <title>، و <body> قرار می‌دهد، و آن را بر می‌گرداند. *title* متنی است که در عنصر <title> درج می‌شود. DOM سطح ۲.
- **hasFeature(*feature*, *version*)**
اگر پیاده‌سازی موجود ویراست مشخص شده از ویژگی داده شده را

پشتیبانی بکند، مقدار true بر می‌گرداند، و در غیر این صورت، مقدار false بر می‌گرداند. اگر هیچ ویراستی داده نشده باشد، در صورتی مقدار true بر می‌گرداند که این پیاده‌سازی ویژگی مشخص شده را به طور کامل با تمام ویراست‌ها پشتیبانی بکند. *feature* و *version* هر دو از نوع رشته هستند؛ مثلاً، "core", "1.0" یا "html", "2.0".

Date ۳٫۱۰

(جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱)
کار با تاریخ و زمان.

۳٫۱۰٫۱ سازنده

```
new Date();             // current time
new Date(milliseconds) // from timestamp
new Date(datestring);  // parse string
new Date(year, month, day, hours, minutes, seconds, ms)
```

سازنده‌ی Date() بدون آوند، یک شیء Date ایجاد می‌کند که روی تاریخ و زمان فعلی تنظیم شده است. اگر یک آوند عددی به آن داده شود، آن را به عنوان نمایش درونی تاریخ بر حسب میلی‌ثانیه، به گونه‌ای که توسط روش getTime() بر گردانده می‌شود، فرض می‌کند. اگر یک آوند رشته‌ای داده شود، آن را به عنوان نمایش رشته‌ای یک تاریخ تعبیر می‌کند. در غیر این صورت، به سازنده دو تا هفت آوند عددی داده می‌شود که فیلدهای تاریخ و زمان محلی را مشخص می‌کنند. تمام فیلدها، به استثنای دو تای اول (سال و ماه) اختیاری هستند. روش ایستای Date.UTC()، جایگزین مناسبی است که به جای زمان محلی از زمان جهانی استفاده می‌کند.

هنگامی که Date() بدون عملگر new فرا خوانده شود، به آوندهای داده شده توجهی نمی‌کند، و نمایش رشته‌ای تاریخ و زمان کنونی را بر می‌گرداند.

۳٫۱۰٫۲ روش‌ها

شیء Date هیچگونه خصلتی ندارد، بلکه هر گونه دستیابی به مقادیر تاریخ و زمان از طریق روش‌های آن صورت می‌گیرد. اکثر روش‌ها دو شکل دارند: یکی نوعی که با زمان محلی کار می‌کند، و دیگری نوعی که واجد حروف "UTC" در نام خود است، و با استفاده از زمان جهانی (UTC یا GMT) کار می‌کند. این

روش‌های جفتی را در زیر ذکر می‌کنیم. دقت کنید که مقادیر برگشتی و آوندهای اختیاری اکثر روش‌های set() که ذیلا ذکر می‌شود، قبل از استانداردسازی ECMA پشتیبانی نمی‌شوند. برای اطلاع از دامنه‌های قانونی هر کدام از فیلدهای تاریخ به روش گوناگون get() مراجعه کنید.

- **get[UTC]Date()**

روز ماه را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۱ و ۳۱ است.

- **get[UTC]Day()**

روز هفته را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۰ (یکشنبه) و ۶ (شنبه) است.

- **get[UTC]FullYear()**

سال را با تمام ۴ رقم آن به وقت محلی یا جهانی بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **get[UTC]Hours()**

فیلد ساعت را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۰ (نیمه‌شب) و ۲۳ (۱۱ شب) است.

- **get[UTC]Milliseconds()**

فیلد میلی‌ثانیه را به وقت محلی یا جهانی بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **get[UTC]Minutes()**

فیلد دقیقه را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۰ و ۵۹ است.

- **get[UTC]Month()**

فیلد ماه را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۰ (ژانویه) و ۱۱ (دسامبر) است.

- **get[UTC]Seconds()**

فیلد ثانیه را به وقت محلی یا جهانی بر می‌گرداند. مقدار برگشتی بین ۰ و ۵۹ است.

- **getTime()**

نمایش درونی تاریخ را بر حسب میلی‌ثانیه بر می‌گرداند؛ درواقع، تعداد میلی‌ثانیه‌های سپری شده از نیمه شب ۱ ژانویه‌ی ۱۹۷۰ (به وقت جهانی) و تاریخ و زمان موجود در شیء `Date` را بر می‌گرداند. دقت کنید که این مقدار ربطی به منطقه‌ی زمانی (وقت محلی) ندارد.

- **getTimezoneOffset()**

اختلاف بین نمایش محلی و جهانی این تاریخ و زمان را بر حسب دقیقه بر می‌گرداند. دقت کنید مقدار بر گردانده شده بستگی به فعال بودن سیستم تغییر ساعت در تابستان نیز دارد.

- **getYear()**

مقدار فیلد سال منهای ۱۹۰۰ را بر می‌گرداند. این روش به نفع روش `getFullYear()` منسوخ شده است.

- **set[UTC]Date(day_of_month)**

مقدار فیلد روز ماه را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند.

- **set[UTC]FullYear(year, month, day)**

مقدار سال (و به طور اختیاری ماه و روز) را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **set[UTC]Hours(hours, mins, secs, ms)**

مقدار ساعت (و به طور اختیاری فیلدهای دقیقه، ثانیه، و میلی‌ثانیه) را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند.

- **set[UTC]Milliseconds(millis)**

مقدار فیلد میلی‌ثانیه‌ی تاریخ را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **set[UTC]Minutes(minutes, seconds, millis)**

مقدار فیلد دقیقه (و به طور اختیاری فیلدهای ثانیه و میلی‌ثانیه) را به وقت

محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند.

- **set[UTC]Month(month, day)**

مقدار فیلد ماه (و به طور اختیاری فیلد روز ماه) را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند.

- **set[UTC]Seconds(seconds, millis)**

مقدار فیلد ثانیه (و به طور اختیاری فیلد میلی‌ثانیه) را به وقت محلی یا جهانی تعیین می‌کند. نمایش میلی‌ثانیه‌ی تاریخ جدید را بر می‌گرداند.

- **setTime(*milliseconds*)**

مقدار نمایش درونی میلی‌ثانیه‌ی تاریخ را تعیین می‌کند. مقدار آوند *milliseconds* را بر می‌گرداند.

- **setYear(year)**

مقدار فیلد سال ۲ رقمی را تعیین می‌کند. این روش به نفع روش set[UTC]FullYear() منسوخ شده است.

- **toDateString()**

قسمت تاریخ را با استفاده از وقت محلی به صورت رشته‌ای بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toGMTString()**

قسمت تاریخ را با استفاده از وقت گرینویچ به صورت رشته‌ای بر می‌گرداند. این روش به نفع روش toUTCString() منسوخ شده است.

- **toLocaleDateString()**

قسمت تاریخ را با استفاده از وقت محلی و با استفاده از قراردادهای محلی فرمت تاریخ به صورت رشته‌ای بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toLocaleString()**

تاریخ را با استفاده از وقت محلی و با استفاده از قراردادهای محلی فرمت تاریخ به صورت رشته‌ای بر می‌گرداند.

- **toLocaleTimeString()**

قسمت زمان را با استفاده از وقت محلی و با استفاده از قراردادهای محلی فرمت تاریخ به صورت رشته‌ای بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toString()**

نمایش رشته‌ای تاریخ را با استفاده از وقت محلی بر می‌گرداند.

- **toTimeString()**

قسمت زمان را با استفاده از وقت محلی به صورت رشته‌ای بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toUTCString()**

شیء تاریخ را با استفاده از زمان جهانی به یک رشته تبدیل می‌کند و آن را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **valueOf()**

نمایش میلی‌ثانیه‌ی تاریخ را، درست مانند روش `getTime()`، بر می‌گرداند. جی‌اسکریپت ۱٫۱؛ ECMA ۱.

۳٫۱۰٫۳ تابع‌های ایستا

علاوه بر روش‌های نمونه که قبلاً گفتیم، شیء `Date` دو روش ایستا نیز تعریف کرده است. این روش‌ها از طریق خود سازنده‌ی `Date()` فراخوانی می‌شوند، و نه از طریق اشیای `Date`.

- **Date.parse(date)**

یک نمایش رشته‌ای تاریخ و زمان را تجزیه می‌کند، و نمایش درونی میلی‌ثانیه را باز می‌گرداند.

- **Date.UTC(yr, mon, day, hr, min, sec, ms)**

نمایش درونی میلی‌ثانیه را برای تاریخ و زمان داده شده باز می‌گرداند.

۳٫۱۱ Document

(جاوا اسکریپت سمت مشتری ۱٫۰؛ DOM سطح ۱)
یک سند HTML.

مشتق شده از: Node (در DOM سطح ۱)

۳/۱۱/۱ دستور

```
window.document
document
```

۳/۱۱/۲ شرح

شیء Document نشان دهنده‌ی یک سند HTML است، و یکی از مهم‌ترین اشیا در جاوا اسکریپت سمت مشتری است. این شیء در جاوا اسکریپت ۱/۰ ارائه شد، و در جاوا اسکریپت ۱/۱، تعدادی روش و خصلت به آن اضافه شد. نت‌اسکیپ و اینترنت اکسپلورر هر کدام روش‌ها و خصلت‌های غیراستانداردی به شیء سند اضافه می‌کنند، W3C DOM نیز در استاندارد خود خصلت‌ها و روش‌های دیگری ارائه کرده است.

۳/۱۱/۳ خصلت‌های مشترک

تمام پیاده‌سازی‌های شیء سند خصلت‌های زیر را پشتیبانی می‌کنند. متعاقباً، لیست‌های جداگانه‌ای نیز برای خصلت‌های تعریف شده در W3C DOM، نت‌اسکیپ، و اینترنت اکسپلورر ارائه خواهیم کرد.

- **alinkColor**

رشته‌ای که رنگ پیوندهای فعال شده را مشخص می‌کند. منسوخ شده است.

- **anchors[]**

آرایه‌ی اشیای لنگر، بدین صورت که برای هر لنگری که در سند وجود دارد، یک شیء در این آرایه موجود است. جاوا اسکریپت ۱/۲.

- **applets[]**

آرایه‌ی اشیای برنامک، بدین صورت که برای هر برنامکی که در سند وجود دارد، یک شیء در این آرایه موجود است. جاوا اسکریپت ۱/۲.

- **bgColor**

رشته‌ای که رنگ زمینه‌ی سند را مشخص می‌کند. منسوخ شده است.

- **cookie**

خصلتی با مقدار رشته‌ای که رفتار خاصی دارد، و امکان خواندن/نوشتن

کوکی‌های مربوط به سند را فراهم می‌کند.

- **domain**

رشته‌ای که دامنه‌ی اینترنتی مبدأ سند را مشخص می‌کند. برای مقاصد امنیتی مورد استفاده قرار می‌گیرد. جاوا اسکریپت ۱/۱.

- **embeds[]**

آرایه‌ای از اشیا که معرف داده‌های خوابانده شده در سند با استفاده از برگه‌ی <embed> هستند. مترادف plugins[] است. بعضی از جازن‌ها و کنترل‌های ActiveX را می‌توان با استفاده از جاوا اسکریپت کنترل کرد. رابط برنامه‌نویسی این کار بستگی به نوع کنترل مورد نظر دارد. جاوا اسکریپت ۱/۲.

- **fgColor**

رشته‌ای که رنگ پیش‌فرض متن سند را مشخص می‌کند. منسوخ شده است.

- **forms[]**

آرایه‌ای از اشیای فرم، شامل یک شیء برای هر فرم موجود در سند.

- **images[]**

آرایه‌ای از اشیای تصویر، شامل یک شیء برای هر تصویری که با استفاده از برگه‌ی <img> در سند قرار داده شده است. جاوا اسکریپت ۱/۱.

- **lastModified**

رشته‌ای فقط-خواندنی، که تاریخ آخرین تغییر انجام شده روی سند را (بر اساس گزارش سرور شبکه) نشان می‌دهد. جاوا اسکریپت ۱/۰.

- **linkColor**

رشته‌ای که رنگ پیوندهای بازدید نشده را نشان می‌دهد. منسوخ شده است.

- **links[]**

آرایه‌ای از اشیای پیوند، شامل یک شیء برای هر پیوند ابرمتن موجود در سند.

- **location**
نشانی سند. به نفع خصلت URL منسوخ شده است.
- **plugins[]**
مترادف آرایه‌ی embeds[]. جاوا اسکریپت ۱/۱.
- **referrer**
رشته‌ای فقط-خواندنی، که نشانی سندی را که به سند جاری پیوند داده بود، در صورت وجود، نشان می‌دهد.
- **title**
محتوای متنی برگه‌ی <title>. تا قبل از DOM سطح ۱، فقط-خواندنی بود.
- **URL**
رشته‌ای فقط-خواندنی، که نشانی سند را نشان می‌دهد. جاوا اسکریپت ۱/۱.
- **vlinkColor**
رشته‌ای که رنگ پیوندهای بازدید شده را نشان می‌دهد. منسوخ شده است.

۳/۱۱/۴ خصلت‌های W3C DOM

در مرورگرهای سازگار با DOM، شیء سند خصلت‌های گره را به ارث می‌برد، و خصلت‌های اضافی زیر نیز برای آن تعریف شده است.

- **body**
اشاره‌ای به شیء عنصری که نشان دهنده‌ی برگه‌ی <body> این سند است.
- **defaultView**
پنجره‌ای که سند در آن نشان داده می‌شود. DOM سطح ۲.
- **documentElement**
اشاره‌ای فقط-خواندنی به برگه‌ی <html> سند.

- **implementation**

شیء DOMImplementation که نشان دهنده‌ی پیاده‌سازی DOM سازنده‌ی سند است. فقط-خواندنی.

۳/۱۱/۵ خصلت‌های اینترنت اکسپلورر ۴

خصلت‌های غیراستاندارد (و انتقال‌ناپذیر) زیر در اینترنت اکسپلورر ۴ و بعد از آن تعریف شده‌اند.

- **activeElement**

خصلتی فقط-خواندنی که به عنصر ورودی‌ای که هم‌اکنون فعال است (یعنی کانون ورودی را در دست دارد) اشاره می‌کند.

- **all[]**

آرایه‌ای از تمام اشیای عنصر موجود در سند. در این آرایه می‌توان بر اساس شماره به ترتیب به عناصر دستیابی پیدا کرد، و یا اینکه می‌توان به جای شماره از id یا name استفاده کرد.

- **charset**

نویسگان سند.

- **children[]**

آرایه‌ای متشکل از عنصرهای HTML که مستقیماً از سند منشعب شده‌اند. دقت کنید که این با all[] متفاوت است، زیرا all[] حاوی تمام عناصر موجود در سند صرف نظر از موقعیت آنها در درخت سند است.

- **defaultCharset**

نویسگان پیش‌فرض سند.

- **expando**

این خصلت، اگر مقدار false به آن داده شود، مانع از گسترش یافتن اشیای سمت مشتری می‌شود. به عبارت دیگر، اگر برنامه‌ای سعی داشته باشد که در شیء به خصلتی که وجود ندارد، مقدار بدهد، خطای زمان اجرا ایجاد می‌شود. این کار می‌تواند به یافتن اشکالات برنامه‌نویسی که ناشی از اشتباه تایپی در نوشتن نام خصلت است، کمک کند. این خصلت خصوصاً می‌تواند برای برنامه‌نویسانی که جدیداً به زبان جاوا اسکریپت

روی آورده‌اند، و از قبل عادت به زبان‌هایی دارند که حروف کوچک و بزرگ در آنها اهمیت ندارد، سودمند واقع شود. گرچه این خصلت فقط در اینترنت اکسپلورر کار می‌کند، ولی مقداردهی به آن در نت‌اسکیپ هم بی‌خطر است (گرچه اثری ندارد).

- **parentWindow**

پنجره‌ای که حاوی این سند است.

- **readyState**

وضعیت بار شدن سند را مشخص می‌کند. مقدار آن یکی از چهار رشته‌ی زیر است:

- **uninitialized**

سند شروع به بار شدن نکرده است.

- **loading**

سند در حال بار شدن است.

- **interactive**

سند به قدر کافی بار شده که کاربر بتواند با آن تعامل کند.

- **complete**

سند کاملاً بار شده است.

۳/۱۱/۶ خصلت‌های نت‌اسکیپ ۴

خصلت‌های غیراستاندارد (و انتقال‌ناپذیر) زیر در نت‌اسکیپ ۴ تعریف شده‌اند.

- **height**

بلندای سند، بر حسب پیکسل.

- **layers[]**

آرایه‌ای از اشیای لایه که معرف لایه‌های موجود در سند است. این خصلت فقط در نت‌اسکیپ ۴ موجود است؛ از نت‌اسکیپ ۶ به بعد حذف شده است.

- **width**

پهنای سند، بر حسب پیکسل.

۳/۱۱/۷ روش‌های مشترک

تمام پیاده‌سازی‌های شیء سند روش‌های زیر را پشتیبانی می‌کنند. متعاقباً، لیست‌های جداگانه‌ای نیز برای روش‌های تعریف شده در W3C DOM، نت‌اسکیپ، و اینترنت اکسپلورر ارائه خواهیم کرد.

- **clear()**

تمام محتویات سند را پاک می‌کند، و هیچ چیز بر نمی‌گرداند. جاوا اسکریپت ۱/۰. در جاوا اسکریپت ۱/۱ منسوخ شده است.

- **close()**

جریان سند را که با استفاده از روش open() باز شده است، می‌بندد، و چیزی بر نمی‌گرداند.

- **open()**

محتویات فعلی سند را پاک می‌کند، و جریانی باز می‌کند که محتویات سند را می‌توان در آن نوشت. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱/۰.

- **write(value, ...)**

رشته یا رشته‌های داده شده را در سندی که هم‌اکنون در حال تجزیه شدن است، درج می‌کند، و یا به آخر سندی که با open() باز شده است، اضافه می‌کند.

- **writeln(value, ...)**

همانند write() است، جز اینکه به آخر خروجی یک نویسه‌ی سطر جدید اضافه می‌کند. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱/۰.

۳/۱۱/۸ روش‌های W3C DOM

در مرورگرهای سازگار با DOM، شیء سند روش‌های گره را به ارث می‌برد، و روش‌های اضافی زیر نیز برای آن تعریف شده است.

- **createAttribute(name)**

یک گره جدید Attr با نام داده شده ایجاد می‌کند و بر می‌گرداند.

- **createComment(text)**

یک گره جدید Comment با متن داده شده ایجاد می‌کند و

بر می‌گرداند.

- **createDocumentFragment()**
یک گره جدید DocumentFragment خالی ایجاد می‌کند و بر می‌گرداند.

- **createElement(tagName)**
یک گره جدید Element با نام برگه‌ی داده شده ایجاد می‌کند و بر می‌گرداند.

- **createTextNode(text)**
یک گره جدید Text با متن داده شده ایجاد می‌کند و بر می‌گرداند.

- **getElementById(id)**
عنصری را در این سند که مقدار صفت id آن معادل با مقدار داده شده است، بر می‌گرداند، و اگر چنین عنصری موجود نباشد، null بر می‌گرداند.

- **getElementsByName(name)**
آرایه‌ای از گره‌های تمام عناصر موجود در سند را که مقدار صفت name آنها معادل مقدار داده شده است، بر می‌گرداند. اگر هیچ عنصری با این ویژگی موجود نباشد، آرایه‌ای با طول صفر بر می‌گرداند.

- **getElementsByTagName(tagname)**
آرایه‌ای از گره‌های تمام عناصر موجود در سند را که نام برگه‌ی آنها معادل نام داده شده است، بر می‌گرداند. ترتیب این عنصرها در این آرایه، همان ترتیب آنها در متن سند است.

- **importNode(importedNode, deep)**
یک نسخه از گرهی در سند دیگر که قابل درج در سند حاضر است، ایجاد می‌کند، و آن را بر می‌گرداند. اگر به آوند deep مقدار true داده شود، به صورت تراجعی گره‌های منشعب شده را هم کپی می‌کند. DOM سطح ۲.

۳/۱۱/۹ روش‌های نت‌اسکیپ

- **getSelection()**

متن انتخاب شده‌ی فعلی در سند را بدون برگه‌های HTML بر می‌گرداند.

۳/۱۱/۱۰ روش‌های اینترنت اکسپلورر ۴

- **elementFromPoint(x, y)**

عنصری را که در نقطه‌ی داده شده واقع است، بر می‌گرداند.

۳/۱۱/۱۱ رویدادپردازها

در مرورگرهای سازگار با DOM و اینترنت اکسپلورر ۴، شیء سند همان رویدادپردازهای عمومی شیء عنصر را پشتیبانی می‌کند. گرچه پردازنده‌های onload و onunload از نظر منطقی به شیء سند تعلق دارند، ولی به عنوان خصلت‌های شیء پنجره پیاده‌سازی شده‌اند.

۳/۱۱/۱۲ ارجاع

Anchor، Applet، Element، Form، Image، Layer، Link، Window.

۳/۱۲ DocumentFragment

(DOM سطح ۱)
گره‌هایی که با هم مورد دستکاری قرار می‌گیرند.
مشتق شده از: Node

۳/۱۲/۱ شرح

شیء DocumentFragment، روش‌ها و خصلت‌های Node را به ارث می‌برد، و هیچ خصلت یا روش جدیدی تعریف نمی‌کند. اما یک رفتار خیلی مهم دارد: وقتی که یک DocumentFragment در داخل سند درج می‌شود، خود شیء درج نمی‌شود، بلکه گره‌های منشعب از آن در سند درج می‌گردد. این باعث می‌شود که DocumentFragment به عنوان جایگاهی برای نگهداری گره‌هایی که می‌خواهید همه را با هم در داخل سند درج کنید، بسیار سودمند باشد.

۳/۱۲/۲ ارجاع

.Document.createDocumentFragment()

۳/۱۳ Element

(DOM سطح ۱، اینترنت اکسپلورر ۴)
یک برگه‌ی HTML در یک سند
مشتق شده از: Node (در DOM سطح ۱)

۳/۱۳/۱ شرح

شیء عنصر نشان دهنده‌ی یک عنصر یا برگه‌ی HTML است. اینترنت اکسپلورر ۴ و بعداً مرورگرهای سازگار با DOM، مانند اینترنت اکسپلورر ۵ به بالا و نت‌اسکیپ ۶ به بالا امکان دسترسی به تمام عناصر سند را فراهم می‌کنند. به علاوه، برای هر عنصر خصلت‌ها و روش‌های زیر را تعریف می‌کنند. متأسفانه، خصلت‌ها و روش‌های اینترنت اکسپلورر ۴ با خصلت‌ها و روش‌های W3C و استاندارد DOM متفاوت است. از این جهت، آنها را در گروه‌های مجزا در زیر بیان می‌کنیم.

۳/۱۳/۲ خصلت‌های W3C DOM

در مرورگرهای شبکه که از W3C DOM پشتیبانی می‌کنند، تمام عناصر در سند HTML خصلت‌هایی متناظر با صفت‌های HTML خود دارند، که صفت‌های عمومی، مانند dir، id، lang، و title نیز از این جمله‌اند. در مواردی که صفت HTML از چند کلمه تشکیل می‌شود، در خصلت متناظر با آن از حروف کوچک و بزرگ برای مشخص کردن کلمات استفاده می‌شود. در سایر موارد، خصلت‌ها در جاوا اسکریپت با حروف کوچک است (مانند id و href، در مقابل tagIndex و accessKey). در دو مورد، نام صفت HTML کلمه‌ای ذخیره شده در زبان جاوا اسکریپت است، و لذا برای این صفات از نام‌های ویژه‌ای استفاده شده است. جاوا اسکریپت از خصلت className برای اشاره به صفت class مربوط به تمام برگه‌های HTML استفاده می‌کند. همچنین، برای صفت for مربوط به برگه‌های <label> و <script>، از خصلت htmlFor استفاده می‌نماید. تمام عناصر، علاوه بر صفات HTML خود، دارای خصلت‌های زیر نیز هستند. در ضمن، به یاد داشته باشید که در مرورگرهای سازگار با DOM، تمام عنصرهای HTML خصلت‌های شیء Node را به ارث می‌برند.

- **className**

مقدار رشته‌ای صفت class عنصر، که یک یا چند کلاس CSS را ذکر می‌کند. دقت کنید که نام این خصلت className است، نه class، زیرا واژه‌ی اخیر از واژگان ذخیره شده‌ی جاوا اسکریپت است.

- **style**

یک شیء Style که معرف صفت style عنصر HTML است.

- **tagName**

نام برگه‌ی عنصر که یک خصلت فقط-خواندنی است؛ صرف نظر از اینکه نام برگه در سند با حروف بزرگ یا کوچک نوشته شده باشد، نام برگه با حروف بزرگ بر گردانده می‌شود. در سندهای XHTML، این مقدار با حروف کوچک است.

۳/۱۳/۳ خصلت‌های DOM اینترنت اکسپلورر

اینترنت اکسپلورر ۴ و ویرایش‌های بعدی یک DOM تجارتی تعریف کرده‌اند. در DOM اینترنت اکسپلورر ۴، همانند W3C DOM، هر عنصر HTML خواص جاوا اسکریپتی متناظر با صفت‌های HTML خود دارد. علاوه بر این، DOM اینترنت اکسپلورر ۴ خصلت‌های زیر را نیز برای هر عنصر تعریف کرده است:

- **all[]**

آرایه‌ای از تمام اشیای عنصر که از این عنصر منشعب شده‌اند. این آرایه را می‌توان شماره‌گذاری کرد، و به ترتیب حضور عناصر در متن سند به آنها دسترسی پیدا کرد. راه دیگر استفاده از id یا name برای یافتن عنصر در این آرایه است. همچنین، رک. Document.all[].

- **children[]**

آرایه‌ای از اشیای عنصر که مستقیماً از این عنصر منشعب شده‌اند. دقت کنید که DOM اینترنت اکسپلورر ۴ هیچ شیئی متناظر با گره‌های متن یا توضیح ندارد، بنا بر این، فرزندان یک عنصر خود عنصرهای دیگری هستند.

- **className**

رشته‌ای خواندنی/نوشتنی که مقدار صفت class یک عنصر را تعیین می‌کند.

- **document**

اشاره‌ای به شیء سندی که در بر دارنده‌ی این عنصر است.

- **innerHTML**

متن HTML موجود در درون عنصر، به استثنای برگه‌ی ورودی و خروجی خود عنصر. هر مقداری به این خصلت داده شود، جایگزین محتوای عنصر می‌شود. از آنجا که این خصلت غیر استاندارد بسیار قوی و پرکاربرد است، مرورگرهای دیگر هم مانند نت‌اسکیپ ۶ و موزیلا آن را پیاده‌سازی کرده‌اند.

- **innerText**

متن ساده‌ی موجود در درون عنصر، به استثنای برگه‌ی ورودی و خروجی خود عنصر. هر مقداری به این خصلت داده شود، آن را بدون تجزیه جایگزین محتوای عنصر می‌کند.

- **offsetHeight**

بلندای عنصر و کل محتوای آن بر حسب پیکسل.

- **offsetLeft**

مختصه‌ی X عنصر نسبت به عنصر گنجایه‌ی offsetParent.

- **offsetParent**

عنصر گنجایه‌ای را تعریف می‌کند که مختصات offsetLeft و offsetTop بر حسب آن اندازه‌گیری می‌شود. برای اکثر عنصرها، offsetParent شیء سندی است که آنها را در خود جای داده است. اما اگر عنصری والدی داشته باشد که به صورت پویا محل آن تعیین شده باشد، همان والد به عنوان offsetParent تلقی می‌شود. خانه‌های جدول نسبت به ردیفی که در آن واقع شده‌اند، جای داده می‌شوند.

- **offsetTop**

مختصه‌ی Y عنصر، نسبت به عنصر گنجایه‌ی offsetParent.

- **offsetWidth**

پهنای عنصر و کل محتویات آن بر حسب پیکسل.

- **outerHTML**

متن HTML عنصر، مشتمل بر برگه‌ی ورودی و خروجی خود عنصر. هر مقداری به این خصلت داده شود، به طور کامل جایگزین عنصر و محتویات آن می‌شود.

- **outerText**

متن ساده‌ی یک عنصر، مشتمل بر برگه‌ی ورودی و خروجی خود عنصر. هر مقداری به این خصلت داده شود، به صورت متن تجزیه نشده جایگزین عنصر و محتویات آن می‌شود.

- **parentElement**

عنصری که والد مستقیم این عنصر است. این خصلت فقط-خواندنی است.

- **sourceIndex**

نمایه‌ی عنصر در آرایه‌ی `Document.all[]` سندی که آن را در خود جای داده است.

- **style**

یک شیء شیوه که نشان دهنده‌ی صفت شیوه‌ی CSS مستقیم این عنصر است. تعیین خصلت‌های این شیء، شیوه‌ی نمایش این عنصر را تغییر می‌دهد.

- **tagName**

رشته‌ای فقط-خواندنی که نام برگه‌ی HTML این عنصر را نشان می‌دهد.

۳٫۱۳٫۴ روش‌های W3C DOM

در مرورگرهایی که از W3C DOM پشتیبانی می‌کنند، تمام عنصرها علاوه بر اینکه روش‌های `Node` را به ارث می‌برند، روش‌های زیر را نیز دارا هستند. بسیاری از این روش‌های برای خواندن یا نوشتن مقادیر صفت‌ها به کار می‌روند، و به ندرت مورد استفاده قرار می‌گیرند، زیرا اشیای عنصر دارای خصلت‌هایی متناظر با هر کدام از صفات HTML خود هستند.

- **getAttribute(name)**

مقدار صفت مشخص شده را به صورت یک رشته بر می‌گرداند.

- **getAttributeNode(name)**
مقدار صفت مشخص شده را به صورت یک گره Attr بر می‌گرداند.

- **getElementsByTagName(name)**
آرایه‌ای از تمام عناصر مشتق شده از این عنصر را که دارای نام برگه‌ی داده شده هستند، به ترتیبی ظهور در متن سند بر می‌گرداند.

- **hasAttribute(name)**
اگر عنصر صفتی با نام داده شده داشته باشد، مقدار true و در غیر این صورت، مقدار false بر می‌گرداند. DOM سطح ۲.

- **removeAttribute(name)**
صفت مشخص شده را از این عنصر حذف می‌کند، و چیزی بر نمی‌گرداند.

- **removeAttributeNode(oldAttr)**
گره Attr مشخص شده را از لیست صفت‌های این عنصر حذف می‌کند. گره Attr حذف شده را بر می‌گرداند.

- **setAttribute(name, value)**
به صفتی که نام آن داده شده است، مقدار داده شده را اختصاص می‌دهد، و چیزی بر نمی‌گرداند.

- **setAttributeNode(newAttr)**
گره Attr مشخص شده را به لیست صفت‌های این عنصر اضافه می‌کند. اگر صفتی با همان نام از قبل موجود باشد، مقدار جدید جایگزین آن می‌شود. گره Attr جایگزین شده با newAttr را بر می‌گرداند، و یا در صورتی هیچگونه جایگزینی انجام نشده باشد، مقدار null بر می‌گرداند.

۳/۱۳/۵ روش‌های DOM اینترنت اکسپلورر

اینترنت اکسپلورر ۴ و ویرایش‌های بعدی، روش‌های غیراستاندارد زیر را برای همه‌ی عنصرهای سند پشتیبانی می‌کنند.

- **contains(target)**
در صورتی که این عنصر حاوی عنصر target باشد، مقدار true و

در غیر این صورت، مقدار false بر می‌گرداند.

- **getAttribute(name)**

مقدار صفت مشخص شده از این عنصر را بر می‌گرداند، و اگر چنین صفتی وجود نداشته باشد، مقدار null بر می‌گرداند.

- **insertAdjacentHTML(where, text)**

متن HTML text را در نزدیکی این عنصر در موقعیتی که به وسیله‌ی where مشخص می‌شود، در داخل سند درج می‌کند. where باید یکی از رشته‌های "BeforeBegin"، "AfterBegin"، "BeforeEnd" و یا "AfterEnd" باشد. چیزی بر نمی‌گرداند.

- **insertAdjacentText(where, text)**

متن ساده‌ی text را در نزدیکی این عنصر در موقعیتی که به وسیله‌ی where مشخص می‌شود، در داخل سند درج می‌کند. چیزی بر نمی‌گرداند.

- **removeAttribute(name)**

صفت مشخص شده و مقدار آن را از عنصر حذف می‌کند. در صورت موفقیت مقدار true، و در غیر این صورت، مقدار false بر می‌گرداند.

- **scrollIntoView(top)**

سند را در صفحه جا به جا می‌کند تا اینکه این عنصر در بالا یا پایین پنجره قرار گیرد. اگر به top مقدار true داده شود و یا اینکه حذف شود، عنصر در بالای پنجره ظاهر می‌شود. اگر به top مقدار false داده شود، عنصر در پایین پنجره ظاهر می‌شود.

- **setAttribute(name, value)**

به صفت مشخص شده، مقدار رشته‌ای داده شده را اختصاص می‌دهد، و هیچ چیز بر نمی‌گرداند.

۳/۱۳/۶ رویدادپردازها

عنصرهای یک سند HTML رویدادپردازهای زیر را برای پاسخ به رویدادهای خام موشواره و صفحه‌کلید تعریف می‌کنند. انواع خاص عنصر (مانند اشیای Form و

Input) ممکن است رویدادپردازهای تخصصی‌تری (مثلاً onsubmit و onchange) تعریف کنند، که به رویدادهای ورودی خام تفسیرهای خاصی را نسبت می‌دهند.

- **onclick**
  هنگامی که کاربر روی عنصر کلیک می‌کند، فرا خوانده می‌کند.
- **ondblclick**
  هنگامی که کاربر روی عنصر دوکلیک می‌کند، فرا خوانده می‌کند.
- **onhelp**
  هنگامی که کاربر تقاضای کمک می‌کند، فرا خوانده می‌کند. فقط اینترنت اکسپلورر.
- **onkeydown**
  هنگامی که کاربر کلیدی را فشار می‌دهد، فرا خوانده می‌کند.
- **onkeypress**
  هنگامی که کاربر کلیدی را فشار داده و رها می‌کند، فرا خوانده می‌کند.
- **onkeyup**
  هنگامی که کاربر کلیدی را رها می‌کند، فرا خوانده می‌کند.
- **onmousedown**
  هنگامی که کاربر یک دکمه‌ی موشواره را فشار می‌دهد، فرا خوانده می‌کند.
- **onmousemove**
  هنگامی که موشواره حرکت می‌کند، فرا خوانده می‌شود.
- **onmouseout**
  هنگامی که کاربر موشواره را از روی عنصر خارج می‌کند، فرا خوانده می‌کند.
- **onmouseover**
  هنگامی که کاربر موشواره را روی عنصر حرکت می‌دهد، فرا خوانده می‌کند.

- **onmouseup**

هنگامی که کاربر یک دکمه‌ی موشواره را رها می‌کند، فرا خوانده می‌کند.

۳٫۱۳٫۷ ارجاع

Form، Input، Node، Select، Textarea.

۳٫۱۴ Error

(جاوا اسکریپت هسته ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳)
انواع استثنای از پیش تعریف شده.
مشتق شده از: Object

۳٫۱۴٫۱ سازنده

```
new Error(message)
new EvalError(message)
new RangeError(message)
new ReferenceError(message)
new SyntaxError(message)
new TypeError(message)
new URIError(message)
```

این سازنده‌ها نمونه‌ای از کلاس Error و یا یکی از زیرکلاس‌های آن می‌سازند. آوند message اختیاری است.

۳٫۱۴٫۲ خصلت‌ها

Error و تمام زیرکلاس‌های آن دو خصلت دارند:

- **message**

پیغام خطایی که جزئیات استثنا را نشان می‌دهد. این خصلت حاوی رشته‌ی داده شده به سازنده و یا یک رشته‌ی پیش‌فرض وابسته به پیاده‌سازی است.

- **name**

رشته‌ای که نوع استثنا را نشان می‌دهد. این خصلت همواره نام سازنده‌ای است که برای ساختن شیء استثنا مورد استفاده قرار گرفته است.

۳/۱۴/۳ روش‌ها

- **toString()**

نمایش رشته‌ای این شیء خطا را بر می‌گرداند.

۳/۱۵ Event

(DOM سطح ۲، اینترنت اکسپلورر ۴، نت‌اسکیپ ۴)
جزئیات رویداد

۳/۱۵/۱ شرح

شیء رویداد هم جزئیات رویداد را مشخص می‌کند، و هم چگونگی انتشار آن را تعیین می‌نماید. DOM سطح ۲ یک شیء رویداد استاندارد تعریف کرده است، ولی اینترنت اکسپلورر ۴، ۵، و ۶ به جای آن از شیء مخصوصی استفاده می‌کنند. نت‌اسکیپ ۴ نیز شیء مخصوصی دارد که با هر دو متفاوت است. DOM سطح ۲ رویدادهای صفحه‌کلید را استانداردسازی نکرده است، بنا بر این، شیء رویداد نت‌اسکیپ ۴ ممکن است هنوز هم برای برنامه‌نویسانی که به رویدادهای کلید در نت‌اسکیپ ۶ و بعد را آن علاقه‌مندند، سودمند باشد. خصلت‌های شیء رویداد DOM، اینترنت اکسپلورر، و نت‌اسکیپ ۴ ذیلا در قسمت‌های جداگانه ارائه می‌شود.

در مدل‌های رویداد DOM و نت‌اسکیپ، شیء رویداد به عنوان آوند به رویدادپرداز داده می‌شود. اما در مدل رویداد اینترنت اکسپلورر، شیء رویدادی که جدیدترین رویداد را توصیف می‌کند، در خصلت event شیء پنجره ذخیره می‌شود.

۳/۱۵/۲ ثابت‌های DOM

این ثابت‌ها مقادیر مجاز برای خصلت eventPhase هستند؛ اینها فاز کنونی انتشار رویداد را برای رویداد مربوطه نشان می‌دهند.

- **Event.CAPTURING_PHASE = 1**

رویداد در مرحله‌ی ارسال است.

- **Event.AT_TARGET = 2**

رویداد در حال پردازش شدن توسط گره هدف است.

- **Event.BUBBLING_PHASE = 3**
رویداد در مرحله‌ی انتشار حبابی است.

۳/۱۵/۳ خصلت‌های DOM

تمام خصلت‌های این شیء فقط-خواندنی‌اند.

- **altKey**
اگر در زمان بروز رویداد، کلید Alt فشار داده شده باشد، مقدار آن true است. برای رویدادهای موشواره تعریف شده است.

- **bubbles**
اگر رویداد از نوعی باشد که انتشار حبابی پیدا می‌کند، مقدار آن true است. برای تمام رویدادها تعریف شده است.

- **button**
مشخص می‌کند که در رویدادهای فشار داده شدن یا رها شدن و یا کلیک کردن، وضعیت کدام دکمه‌ی موشواره تغییر کرده است. ۰ معرف دکمه‌ی چپ، ۱ دکمه‌ی وسط، و ۲ دکمه‌ی راست است. دقت کنید که این خصلت فقط زمانی تعریف شده است که دکمه‌ای تغییر وضعیت بدهد: بنا بر این، مثلاً زمانی که در حین بروز یک رویداد حرکت موشواره دکمه‌ای پایین نگهداشته شده است، این خصلت استفاده نمی‌شود. به علاوه، این خصلت به صورت چندبیتی نیست، یعنی اگر بیش از یک دکمه‌ی موشواره پایین نگهداشته شده باشد، آنها را گزارش نمی‌دهد. نت‌اسکیپ ۶/۰ به جای ۰، ۱، و ۲ از ۱، ۲، و ۳ استفاده می‌کند. این مسئله در نت‌اسکیپ ۶/۱ اصلاح شده است.

- **cancelable**
در صورتی که بتوان با فراخوانی preventDefault()، عمل پیش‌فرض مربوط به رویداد را لغو کرد، مقدار این خصلت true و در غیر این صورت، false است.

- **clientX, clientY**
این خصلت‌ها مختصات X و Y نشانگر موشواره را نسبت به ناحیه‌ی مشتری پنجره‌ی مرورگر نشان می‌دهند. دقت کنید که این خصلت‌ها در نوردیدن سند را در نظر نمی‌گیرند.

- **ctrlKey**

اگر در زمان بروز رویداد، کلید Ctrl فشار داده شده باشد، مقدار آن true است. برای رویدادهای موشواره تعریف شده است.

- **currentTarget**

گره سند که هم اکنون در حال رسیدگی به این رویداد است. در حین گرفتن و انتشار حبابی رویداد، این گره متفاوت از گره target است. برای تمام رویدادها تعریف شده است.

- **detail**

تعداد کلیک: ۱ برای تک‌کلیک، ۲ برای دوکلیک، ۳ برای سه‌کلیک، و الخ. برای رویدادهای کلیک، فشار داده شدن، و رها شدن موشواره تعریف شده است.

- **eventPhase**

مرحله‌ی کنونی انتشار رویداد. ثابت‌های تعریف شده در بالا سه مقدار مجاز این خصلت را نشان می‌دهند. برای تمام رویدادها تعریف شده است.

- **metaKey**

اگر در زمان بروز رویداد، کلید Meta فشار داده شده باشد، مقدار آن true است. برای رویدادهای موشواره تعریف شده است.

- **relatedTarget**

برای رویدادهای mouseover، این گرهی است که موشواره قبل از وارد شدن به گره فعلی از آن خارج شده است. برای رویدادهای mouseout، این گرهی است که موشواره بعد از خارج شدن از گره فعلی به آن وارد شده است. برای سایر انواع رویداد، تعریف نشده است.

- **screenX, screenY**

این خصلت‌ها مختصات X و Y نشانگر موشواره را نسبت به گوشه‌ی بالا و چپ صفحه‌ی کاربر تعیین می‌کنند. برای رویدادهای موشواره تعریف شده است.

- **shiftKey**

اگر در زمان بروز رویداد، کلید Shift فشار داده شده باشد، مقدار آن true است. برای رویدادهای موشواره تعریف شده است.

- **target**

گره هدف این رویداد؛ گرهی که رویداد را ایجاد کرده است. دقت کنید که این می‌تواند هر گرهی، حتی گره متن باشد، و منحصر به گره‌های عنصر نیست. برای تمام رویدادها تعریف شده است.

- **timeStamp**

یک شیء تاریخ که تاریخ و ساعت بروز رویداد را نشان می‌دهد. برای تمام رویدادها تعریف شده است، ولی پیاده‌سازی‌ها مجبور نیستند تاریخ معتبری را گزارش کنند.

- **type**

نوع رویداد. این همان نام رویدادپرداز با برداشتن حروف **on** از آغاز آن است. مثلاً، رویداد click، load، و یا mousedown. برای تمام رویدادها تعریف شده است.

- **view**

شیء پنجره‌ای که رویداد در آن پدید آمده است.

۳/۱۵/۴ روش‌های DOM

- **preventDefault()**

به مرورگر شبکه می‌گوید که عمل پیش‌فرض مربوط به این رویداد را (اگر چنین عملی وجود داشته باشد)، انجام ندهد. اگر رویداد از نوع قابل لغو نباشد، این روش اثری ندارد. چیزی بر نمی‌گرداند.

- **stopPropagation()**

از انتشار بیشتر رویداد در مراحل ارسال، هدف، و انتشار حبابی جلوگیری می‌کند. چیزی بر نمی‌گرداند.

۳/۱۵/۵ خصلت‌های اینترنت اکسپلورر ۴

- **altKey**

مقداری بولی که مشخص می‌کند در زمان بروز رویداد، کلید `Alt` فشار داده شده است یا نه.

- **button**

در رویدادهای موشواره، خصلت button مشخص می‌کند چه کلید یا کلیدهای موشواره فشا رداده شده‌اند. این عدد صحیح فقط-خواندنی یک بیت‌ماسک است: بیت ۱ در صورتی روشن است که دکمه‌ی چپ موشواره فشار داده شده باشد؛ بیت ۲ در صورتی روشن است که دکمه‌ی راست موشواره فشار داده شده باشد؛ بیت ۴ در صورتی روشن است که دکمه‌ی میانی موشواره (در موشواره‌های سه‌دکمه‌ای) فشار داده شده باشد.

- **cancelBubble**

اگر رویدادپرداز بخواهد از انتشار رویداد به اشیای بالاتر جلوگیری کند، باید به این خصلت مقدار true بدهد.

- **clientX, clientY**

مختصات X و Y محل بروز رویداد، نسبت به صفحه‌ی مرورگر شبکه.

- **ctrlKey**

مقداری بولی که مشخص می‌کند در زمان بروز رویداد، کلید Ctrl فشار داده شده است یا نه.

- **fromElement**

برای رویدادهای mouseover و mouseout، fromElement به شیئی اشاره دارد که نشانگر موشواره از آن خارج می‌شود.

- **keyCode**

برای رویدادهای صفحه‌کلید، keyCode رمز نویسه‌ی یونیکد ایجاد شده به وسیله‌ی کلید مورد نظر را مشخص می‌کند.

- **offsetX, offsetY**

مختصات X و Y محل بروز رویداد، نسبت به سیستم مختصات عنصر مبدأ رویداد (رک. srcElement).

- **returnValue**

اگر به این خصلت مقداری داده شود، مقدار آن بر مقداری که واقعاً به وسیله‌ی رویدادپرداز بر گردانده می‌شود، تقدم دارد. برای لغو عمل پیش‌فرض عنصر مبدائی که رویداد روی آن رخ داده است، به این خصلت مقدار false بدهید.

- **screenX, screenY**

مختصات X و Y محل بروز رویداد، نسبت به صفحه‌ی نمایش.

- **shiftKey**

مقداری بولی که مشخص می‌کند در زمان بروز رویداد، کلید Shift فشار داده شده است یا نه.

- **srcElement**

شیء پنجره، سند، و یا عنصری که رویداد را ایجاد کرده است.

- **toElement**

برای رویدادهای mouseover و mouseout، toElement به شیئی اشاره دارد که نشانگر موشواره به آن داخل می‌شود.

- **type**

یک خصلت رشته‌ای که نوع رویداد را مشخص می‌کند. مقدار آن نام رویدادپرداز منهای پیشوند **on** است. بنا بر این، وقتی رویدادپرداز onclick() فرا خوانده می‌شود، مقدار خصلت type شیء رویداد، click است.

- **x, y**

مختصات X و Y محل بروز رویداد. این خصلت، مختصات را نسبت به درونی‌ترین عنصر در بر گیرنده که با استفاده از CSS به صورت پویا تعیین محل شده است، بیان می‌کند.

۳/۱۵/۶ خصلت‌های نت‌اسکیپ ۴

- **height**

فقط در رویدادهای resize تعیین می‌شود. بلندای جدید پنجره یا کادر را پس از تغییر اندازه مشخص می‌کند.

- **layerX, layerY**

مختصات X و Y محل بروز رویداد را نسبت به لایه‌ی در بر گیرنده مشخص می‌کند.

- **modifiers**

مشخص می‌کند که کدام کلیدهای تغییرگر در هنگام بروز رویداد، فشار داده شده‌اند. این مقدار عددی، یک بیت‌ماسک است که می‌تواند متشکل از هر کدام از ثابت‌های Event.ALT_MASK، Event.CONTROL_MASK، Event.META_MASK، و یا Event.SHIFT_MASK باشد. به علت یک خطا، این خصلت در نت‌اسکیپ ۶ یا ۶/۱ تعریف نشده است.

- **pageX, pageY**

مختصات X و Y محل بروز رویداد، نسبت به صفحه‌ی مرورگر. دقت کنید که این مختصات نسبت به صفحه‌ی سطح بالا هستند، نه نسبت به لایه‌ی در بر گیرنده.

- **screenX, screenY**

مختصات X و Y محل بروز رویداد، نسبت به صفحه‌ی نمایش.

- **target**

شیء پنجره، سند، لایه، و یا عنصری که رویداد در آن بروز کرده است.

- **type**

یک خصلت رشته‌ای که نوع رویداد را مشخص می‌کند. مقدار آن نام رویدادپرداز منهای پیشوند **on** است. بنا بر این، وقتی رویدادپرداز onclick() فرا خوانده می‌شود، مقدار خصلت type شیء رویداد، click است.

- **which**

برای رویدادهای صفحه‌کلید یا موشواره، which مشخص می‌کند که کدام کلید یا دکمه‌ی موشواره فشار داده یا رها شده است. برای رویدادهای صفحه‌کلید، این خصلت حاوی رمز نویسه‌ای کلید فشار داده شده است. برای رویدادهای موشواره، مقدار آن ۱، ۲، و یا ۳ است، که به ترتیب، دکمه‌ی چپ، میانی، و راست را نشان می‌دهد.

- **width**

فقط در رویدادهای resize تعیین می‌شود. پهنای جدید پنجره یا کادر را پس از تغییر اندازه مشخص می‌کند.

- **x, y**

مختصات X و Y محل بروز رویداد. این خصلت‌ها مترادف با layerX و layerY هستند، و موقعیت را نسبت به لایه‌ی در بر گیرنده (در صورت وجود) نشان می‌دهند.

Form ۳٫۱۶

(جاوا اسکریپت سمت مشتری ۱٫۰)
یک فرم ورودی HTML
مشتق شده از: Element

۳٫۱۶٫۱ دستور

```
document.forms[form_number]
document.forms[form_name]
document.form_name
```

۳٫۱۶٫۲ خصلت‌ها

شیء فرم، خصلت‌هایی را برای هر کدام از صفت‌های عنصر <form> در HTML، از قبیل action، encoding، method، name، و target تعریف می‌کند. به علاوه، خصلت‌های زیر را هم تعریف می‌کند.

- **elements[]**

آرایه‌ای فقط-خواندنی از شیءهای ورودی که نشان دهنده‌ی عنصرهای موجود در فرم هستند. آرایه را می‌توان با شماره نمایه‌گذاری کرد، و یا برای عنصرهایی که مقدار صفت name آنها تعیین شده است، از نام استفاده کرد.

- **length**

تعداد عنصرهای موجود در فرم. معادل elements.length است.

۳٫۱۶٫۳ روش‌ها

- **reset()**

مقدار هر کدام از عناصر ورودی فرم را به مقدار پیش‌فرض بر می‌گرداند.

چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱.

- **submit()**

فرم را تحویل می‌دهد، ولی رویدادپرداز onsubmit را فعال نمی‌کند. چیزی بر نمی‌گرداند.

۳٫۱۶٫۴ رویدادپردازها

- **onreset**

درست پیش از بر گرداندن مقدار عناصر فرم فراخوانی می‌شود. برای جلوگیری از بر گردانده شدن مقدار عنصرها، باید مقدار false بر گرداند.

- **onsubmit**

درست پیش از تحویل داده شدن فرم فراخوانی می‌شود. این رویدادپرداز، امکان بررسی صحت موارد وارد شده توسط کاربر را فراهم می‌کند. برای جلوگیری از تحویل داده شدن فرم، باید مقدار false بر گرداند.

۳٫۱۶٫۵ ارجاع

Element، Input، Select، Textarea.

۳٫۱۷ Function

(جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱)
یک تابع جاوا اسکریپت.

۳٫۱۷٫۱ سازنده

```
new Function(argument_names..., body)
```

این سازنده در جاوا اسکریپت ۱٫۱ وارد شد. ولی در جاوا اسکریپت ۱٫۲ منسوخ شده، و به جای آن از دستور مستقیم تعریف تابع استفاده می‌شود.

۳٫۱۷٫۲ خصلت‌ها

- **length**

تعداد آوندهای نام برده شده در هنگام تعریف تابع را نشان می‌دهد. برای

مشخص کردن تعداد واقعی آوندهای داده شده، می‌توانید از Arguments.length استفاده کنید. جاوا اسکریپت ۱٫۱، جی‌اسکریپت ۲٫۰، ECMA ۱.

- **prototype**

شیئی که برای یک تابع سازنده، تمام خصلت‌ها و روش‌هایی را تعریف می‌کند که بین اشیای ساخته شده توسط سازنده مشترک است. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱.

۳٫۱۷٫۳ روش‌ها

- **apply(thisobj, args)**

تابع را به عنوان روشی از thisobj فراخوانی می‌کند، و عنصرهای آرایه‌ی args را به عنوان آوندهای تابع به آن می‌دهد. مقدار بر گردانده شده توسط تابع را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **call(thisobj, args...)**

تابع را به عنوان روشی از thisobj فراخوانی می‌کند، و آوندهای باقیمانده را به عنوان آوندهای تابع به آن می‌دهد. مقدار بر گردانده شده توسط تابع را بر می‌گرداند. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toString()**

نمایش رشته‌ای تابع را بر می‌گرداند. در برخی از پیاده‌سازی‌ها، نمایش رشته‌ای به صورت متن برنامه‌ی تابع است. جاوا اسکریپت ۱٫۰؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱.

۳٫۱۷٫۴ ارجاع

Arguments

۳٫۱۸ Global

(جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱)

خصلت‌ها و تابع‌های سراسری.

۳/۱۸/۱ دستور

```
this
```

۳/۱۸/۲ شرح

شیء Global، خصلت‌ها و روش‌های سراسری را در خود نگه می‌دارد. این خصلت‌ها و روش‌ها را لازم نیست از طریق شیء خاصی فراخوانی کنیم. هر گونه متغیر یا تابعی در متن سطح بالا تعریف کنید، تبدیل به خصلت‌های شیء سراسری می‌شود. شیء سراسری نامی ندارد، ولی می‌توانید در متن سطح بالا (یعنی در خارج از روش‌ها) با استفاده از کلیدواژه‌ی this به آن اشاره کنید. در جاوا اسکریپت سمت مشتری، شیء پنجره به عنوان شیء سراسری عمل می‌کند. این شیء، خصلت‌ها و روش‌های متعدد دیگری نیز دارد، و می‌توان به صورت window به آن اشاره کرد.

۳/۱۸/۳ خصلت‌های سراسری

- **Infinity**

مقداری عددی که مثبت بی‌نهایت را نشان می‌دهد. جاوا اسکریپت هسته ۱/۳؛ جی‌اسکریپت ۳/۰؛ ECMA ۱.

- **NaN**

مقدار غیرعددی. جاوا اسکریپت هسته ۱/۳؛ جی‌اسکریپت ۳/۰؛ ECMA ۱.

- **undefined**

مقدار تعریف نشده. جاوا اسکریپت هسته ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

۳/۱۸/۴ تابع‌های سراسری

- **decodeURI(uri)**

یک نسخه‌ی رمزگشایی شده از uri را بر می‌گرداند، که در آن به جای سلسه‌های گریز شانزدهگانی، نویسه‌ی مربوطه قرار داده شده است. جاوا اسکریپت هسته ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

- **decodeURIComponent(s)**

یک نسخه‌ی رمزگشایی شده از s را بر می‌گرداند، که در آن به جای سلسله‌های گریز شانزدهگانی، نویسه‌ی مربوطه قرار داده شده است. جاوا اسکریپت هسته ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **encodeURI(uri)**

یک نسخه‌ی رمزگذاری شده از uri را بر می‌گرداند، که در آن به جای برخی نویسه‌ها، سلسله‌های گریز شانزدهگانی قرار داده شده است. نویسه‌هایی مانند #، ?، و @ که برای جدا کردن اجزای URI به کار می‌روند، تبدیل به سلسله‌ی گریز نمی‌شوند. جاوا اسکریپت هسته ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **encodeURIComponent(s)**

یک نسخه‌ی رمزگذاری شده از s را بر می‌گرداند، که در آن به جای برخی نویسه‌ها، سلسله‌های گریز شانزدهگانی قرار داده شده است. نویسه‌های نقطه‌گذاری را هم که برای جدا کردن اجزای URI به کار می‌روند، تبدیل به سلسله‌ی گریز می‌کند. جاوا اسکریپت هسته ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **escape(s)**

یک نسخه‌ی رمزگذاری شده از s را بر می‌گرداند، که در آن به جای برخی نویسه‌ها، سلسله‌های گریز شانزدهگانی قرار داده شده است. جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱؛ در ECMA ۳ منسوخ شده و به جای آن باید از encodeURI و encodeURIComponent استفاده کرد.

- **eval(code)**

رشته‌ای از متن جاوا اسکریپت را ارزیابی می‌کند، و حاصل را بر می‌گرداند.

- **isFinite(n)**

در صورتی که n یک عدد متناهی بوده و یا قابل تبدیل به چنین عددی باشد، true بر می‌گرداند. اگر مقدار n (یا حاصل تبدیل آن) NaN (غیرعدد) و یا مثبت یا منفی بی‌نهایت باشد، false بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **isNaN(x)**

اگر مقدار x (یا حاصل تبدیل آن) NaN (غیرعدد) باشد، true بر می‌گرداند. در صورتی که x یک مقدار عددی بوده و یا قابل تبدیل به آن باشد، false بر می‌گرداند. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱.

- **parseFloat(s)**

رشته‌ی s (و یا قطعه‌ی پیشوند s را) به یک عدد تبدیل می‌کند، و آن عدد را بر می‌گرداند. اگر s با عدد معتبری شروع نشده باشد، مقدار NaN (و یا ۰ در جاوا اسکریپت ۱٫۰) بر می‌گرداند. جاوا اسکریپت ۱٫۰؛ جی‌اسکریپت ۱٫۱؛ ECMA ۱.

- **parseInt(s, radix)**

رشته‌ی s (و یا قطعه‌ی پیشوند s را) به یک عدد صحیح تیدیل می‌کند، و آن عدد را بر می‌گرداند. اگر s با عدد معتبری شروع نشده باشد، مقدار NaN (و یا ۰ در جاوا اسکریپت ۱٫۰) بر می‌گرداند. آوند اختیاری radix، مبنای عدد (بین ۲ و ۳۶) را مشخص می‌کند. اگر این آوند داده نشده باشد، مبنای پیش‌فرض ۱۰، و یا در صورتی که رشته با پیشوند شانزدهگانی یعنی "0x" یا "0X" شروع شده باشد، ۱۶ است. جاوا اسکریپت ۱٫۰؛ جی‌اسکریپت ۱٫۱؛ ECMA ۱.

- **unescape(s)**

رشته‌ای را که با استفاده از ()escape رمزگذاری شده است، رمزگشایی می‌کند. نسخه‌ی رمزگشایی شده‌ی s را بر می‌گرداند. جاوا اسکریپت ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱. در ECMA ۳ منسوخ شده، و باید به جای آن از ()decodeURI و ()decodeURIComponent استفاده کنید.

۳٫۱۸٫۵ ارجاع

.Window

History ۳٫۱۹

(جاوا اسکریپت سمت مشتری ۱٫۰)

عقب یا جلو رفتن در سابقه‌ی مرور.

۳٫۱۹٫۱ دستور

```
window.history
history
```

۳٫۱۹٫۲ روش‌ها

- **back()**

در سابقه‌ی مرور، به نشانی مرور شده‌ی قبلی بر می‌گردد. چیزی بر نمی‌گرداند.

- **forward()**

در سابقه‌ی مرور، جلو می‌رود. چیزی بر نمی‌گرداند.

- **go(n)**

به نشانی nام نسبت به نشانی فعلی می‌رود. اگر این روش با مقدار 1- فرا خوانده شود، همانند فرا خواندن back() است. چیزی بر نمی‌گرداند.

۳٫۲۰ Image

(جاوا اسکریپت سمت مشتری ۱٫۱)
یک تصویر HTML.
مشتق شده از: Element

۳٫۲۰٫۱ دستور

```
document.images[i]
document.images[image-name]
document.image-name
```

۳٫۲۰٫۲ سازنده

```
new Image(width, height);
```

این سازنده تصویری خارج از صفحه ایجاد می‌کند که نمی‌توان آن را نمایش داد. آوندهای width و height اختیاری هستند. اگر در شیء حاصله، مقدار صفت src تعیین شود، سبب می‌شود که مرورگر تصویر را به حافظه‌ی نهانی خود پیش‌خوانی کند.

۳/۲۰/۳ خصلت‌ها

شیء تصویر برای هر یک از صفت‌های عنصر HTML <img>، از قبیل src، border، width، height، vspace، و hspace، خصلتی تعریف می‌کند. علاوه بر این، خصلت‌های خاص زیر را هم دارا است:

- **complete**

اگر تصویر هنوز در حال بار شدن باشد، مقدار آن false است. اگر خواندن تصویر تمام شده و یا در اثنای آن خطایی بروز کرده است، مقدار آن true است. این خصلت فقط-خواندنی است.

- **src**

رشته‌ای خواندنی/نوشتنی که نشانی تصویر را نشان می‌دهد. این خصلت صرفاً منعکس کننده‌ی صفت src برگه‌ی <img> است، ولی علت توجه خاص به آن در اینجا این است که بسیاری از جلوه‌های مهم HTML پویا (DHTML) با تعیین مقدار این خصلت به صورت پویا انجام می‌شود، که سبب می‌شود تصویری با تصویر دیگر جایگزین شود.

۳/۲۰/۴ رویدادپردازها

شیء تصویر رویدادپرداز عنصر را به ارث می‌برد، و در ضمن، موارد زیر را هم تعریف می‌کند:

- **onabort**

اگر کاربر فروگذاری تصویر را قطع کند، فرا خوانده می‌شود.

- **onerror**

اگر هنگام فروگذاری تصویر خطایی بروز کند، فرا خوانده می‌شود.

- **onload**

زمانی که بار شدن تصویر با موفقیت به اتمام رسید، فرا خوانده می‌شود.

Input ۳/۲۱

(جاوا اسکریپت سمت مشتری ۱/۰)
یک عنصر ورودی فرم
مشتق شده از: Element

۳٫۲۱٫۱ دستور

```
form.elements[i]
form.elements[name]
form.name
```

۳٫۲۱٫۲ خصلت‌ها

شیء ورودی خصلت‌هایی متناظر با هر یک از صفت‌های برگه‌ی <input>، از قبیل maxLength، readOnly، size، و tabIndex، دارد. علاوه بر این، خصلت‌های زیر را نیز تعریف می‌کند:

- **checked**

یک مقدار بولی خواندنی/نوشتنی که مشخص می‌کند یک عنصر ورودی از نوع چک‌باکس یا دکمه‌ی رادیویی، چک شده (true) است یا نه (false).

- **defaultChecked**

یک مقدار بولی خواندنی/نوشتنی که مشخص می‌کند یک عنصر ورودی از نوع چک‌باکس یا دکمه‌ی رادیویی، وقتی در آغاز تشکیل می‌شود و یا به حالت اولیه بر گردانده می‌شود، چک شده است یا نه.

- **defaultValue**

مقداری رشته‌ای که متن نشان داده شده در ورودی «متن» یا «گذرواژه» در آغاز تشکیل و یا پس از بر گردانده شدن به مقدار اولیه را مشخص می‌کند. به دلایل امنیتی، این خصلت بر عناصر ورودی از نوع «پرونده» اثری ندارد.

- **form**

اشاره‌ای فقط-خواندنی به شیء فرمی که عنصر در آن قرار گرفته است. این خصلت برای تمام انواع عنصرهای ورودی تعریف شده است.

- **name**

نام این عنصر ورودی، که به وسیله‌ی صفت name در HTML تعیین می‌شود. این خصلت برای تمام انواع عنصرهای ورودی تعریف شده است.

- **type**

رشته‌ای که نوع عنصر فرم را مشخص می‌کند. این خصلت منعکس

کننده‌ی مقدار صفت type در HTML است. مقادیر قانونی آن در جدول زیر لیست شده است؛ مقدار پیش‌فرض "text" است. اشیای Submit و Textarea هم یک خصلت دارند که مقادیر ممکن آن "select-one"، "select-multiple"، و "textarea" است. جاوا اسکریپت ۱/۱.

نوع	شرح
"button"	دکمه
"checkbox"	چک‌باکس
"file"	عنصر فراگذاری پرونده
"hidden"	عنصر پنهان
"image"	دکمه‌ی گرافیکی تحویل فرم
"password"	فیلد ورود متن مخفی شده
"radio"	دکمه‌های رادیویی مانعةالجمع
"reset"	دکمه‌ی برگردان
"text"	ورودی متن یک‌سطری
"submit"	دکمه‌ی تحویل فرم

- **value**

مقدار رشته‌ای که هنگام تحویل فرم فرستاده می‌شود. برای عناصر ورودی از نوع متن، گذرواژه، و پرونده، این متن قابل ویرایشی نشان داده شده در درون عنصر است. می‌توانید برای تغییر مقدار نمایش داده شده، مقدار این خصلت را تغییر دهید. برای عنصرهای ورودی از نوع دکمه، تحویل، و برگردان، مقدار این خصلت رشته‌ای است که به عنوان برچسب روی دکمه نمایش داده می‌شود. در انواع دیگر، رشته‌ی value نمایش داده نمی‌شود. دقت کنید که به دلایل امنیتی، خصلت value در عنصرهای نوع پرونده معمولاً فقط-خواندنی است.

۳/۲۱/۳ روش‌ها

- **blur()**

کانون صفحه‌کلید را تحویل می‌دهد و چیزی بر نمی‌گرداند. برای تمام عنصرها غیر از نوع پنهان تعیین شده است.

- **click()**

مانند کلیک شدن موشواره روی عنصر عمل می‌کند، و چیزی

بر نمی‌گرداند. برای انواع عنصرهای دکمه‌ای — دکمه، چک‌باکس، رادیو، برگردان، و تحویل — تعریف شده است.

- **focus()**

کانون صفحه‌کلید را می‌گیرد و چیزی بر نمی‌گرداند. برای تمام عنصرها غیر از نوع پنهان تعیین شده است.

- **select()**

متن موجود در عنصر را انتخاب می‌کند و چیزی بر نمی‌گرداند. برای عنصرهای نوع متن، گذرواژه، و پرونده کار می‌کند. برای شیء Textarea هم تعریف شده است.

۳٫۲۱٫۴ رویدادپردازها

- **onblur**

وقتی که عنصر کانون صفحه‌کلید را تحویل می‌دهد، فراخوانی می‌شود. برای تمام عنصرها غیر از نوع پنهان تعیین شده است.

- **onchange**

این رویدادپرداز، برای عنصرهای ورود متن، شامل متن، گذرواژه، و پرونده، زمانی فرا خوانده می‌شود که کاربر متن درون عنصر را تغییر می‌دهد و کانون صفحه‌کلید را از عنصر بیرون می‌برد، بدان معنا که کار ویرایش تمام شده است. به عبارت دیگر، برای هر کلیدی که زده می‌شود، این فراخوانی صورت نمی‌پذیرد.

- **onclick**

این رویدادپرداز، برای عنصرهای دکمه‌ای — دکمه، چک‌باکس، رادیو، برگردان، و تحویل — هنگام کلیک شدن موشواره روی عنصر فرا خوانده می‌شود. برای انواع تحویل و برگردان، اگر این تابع false بر گرداند، کار تحویل یا برگردان انجام نمی‌شود.

- **onfocus**

هنگامی که عنصر کانون صفحه‌کلید را دریافت می‌کند، فراخوانی می‌شود. برای تمام عنصرها غیر از نوع پنهان تعیین شده است.

۳٫۲۱٫۵ ارجاع

Form، Option، Select، Textarea.

۳٫۲۲ Layer

(فقط نت‌اسکیپ ۴ سمت مشتری)
یک لایه‌ی مستقل سند.

۳٫۲۲٫۱ دستور

```
document.layers[i]
document.layers[layer-name]
document.layer-name
```

۳٫۲۲٫۲ سازنده

```
new Layer(width, parent_layer)
```

۳٫۲۲٫۳ شرح

شیء لایه فقط در نت‌اسکیپ ۴ پشتیبانی می‌شود، و در نت‌اسکیپ ۶ حذف شده است. این شیء کاملاً غیراستاندارد است، ولی علت ذکرش در اینجا این است که تنها راه تعیین پویای جای اشیا در نت‌اسکیپ ۴ است. هر عنصر HTML که به صفت CSS position آن مقدار absolute داده شده باشد، در جاوا اسکریپت با یک لایه نمایش داده می‌شود. در ضمن، می‌توانید با برگه‌ی غیراستاندارد <layer> و یا با سازنده‌ی Layer() لایه ایجاد کنید.

۳٫۲۲٫۴ خصلت‌ها

- **above**
  لایه‌ی روی این لایه، اگر وجود داشته باشد. فقط-خواندنی.
- **background**
  تصویر زمینه‌ی لایه.
- **below**
  لایه‌ی زیر این لایه، اگر وجود داشته باشد. فقط-خواندنی.

- **bgColor**
  رنگ زمینه‌ی لایه.
- **clip.bottom**
  مختصه‌ی Y لبه‌ی پایینی ناحیه‌ی برش لایه، نسبت به top.
- **clip.height**
  بلندای ناحیه‌ی برش لایه. با دادن مقدار جدید به آن، مقدار clip.bottom هم تغییر می‌کند.
- **clip.left**
  مختصه‌ی X لبه‌ی چپ ناحیه‌ی برش لایه، نسبت به left.
- **clip.right**
  مختصه‌ی X لبه‌ی راست ناحیه‌ی برش لایه، نسبت به left.
- **clip.top**
  مختصه‌ی Y لبه‌ی بالایی ناحیه‌ی برش لایه، نسبت به top.
- **clip.width**
  پهنای ناحیه‌ی برش لایه. با دادن مقدار جدید به آن، مقدار clip.right هم تغییر می‌کند.
- **document**
  اشاره‌ای فقط-خواندنی به شیء سند که در درون لایه واقع است.
- **hidden**
  پنهان یا مرئی بودن لایه را مشخص می‌کند. برای پنهان کردن لایه به آن مقدار true بدهید، و برای مرئی کردن آن، مقدار false به آن بدهید.
- **layers[]**
  آرایه‌ای حاوی تمام شیءهای لایه که از این لایه منشعب شده‌اند. این همانند آرایه‌ی لایه‌های document.layers[] است.
- **left**
  مختصه‌ی X این لایه، نسبت به لایه یا سند حاوی آن. تغییر مقدار این خصلت سبب انتقال لایه به چپ یا راست می‌شود. left مترادف با x است.

- **name**

صفت name برگه‌ی HTML مربوط به این لایه.

- **pageX, pageY**

مختصه‌ی X و Y این لایه نسبت به سند سطح بالا. دقت کنید که این مختصات نسبت به صفحه‌ی سطح بالا است، نه نسبت به هیچکدام از لایه‌های حاوی این لایه.

- **parentLayer**

اشاره‌ای فقط-خواندنی به لایه یا پنجره‌ی حاوی (والد) این لایه.

- **siblingAbove, siblingBelow**

این خصلت‌ها لایه‌ی رو یا زیر این لایه را که با آن برادر است (یعنی والد یکسانی دارد) نشان می‌دهد. اگر چنین لایه‌هایی وجود نداشته باشد، این خصلت null بر می‌گرداند.

- **src**

رشته‌ای خواندنی/نوشتنی که نشانی محتویات لایه را مشخص می‌کند. دادن نشانی جدید به این خصلت سبب می‌شود که مرورگر محتویات آن نشانی را بخواند و آن را در لایه نمایش دهد.

- **top**

مختصه‌ی Y این لایه، نسبت به لایه یا سند در بر گیرنده‌ی آن. تعیین این خصلت سبب بالا یا پایین رفتن لایه می‌شود. top مترادف با y است.

- **visibility**

رشته‌ای خواندنی/نوشتنی که مرئی بودن لایه را مشخص می‌کند. سه مقدار قانونی دارد: "show"، "hide"، و "inherit".

- **window**

شیء پنجره‌ای که این لایه، صرف نظر از عمق، در آن قرار گرفته است.

- **x, y**

مختصات X و Y لایه. x مترادف با left، و y مترادف با خصلت top است.

- **zIndex**

موقعیت لایه در ترتیب z یا ترتیب پشته‌ای لایه‌ها. وقتی دو لایه روی هم می‌افتند، لایه‌ای که zIndex بالاتری داشته باشد، در بالا قرار می‌گیرد و لایه‌ی دیگر را محو می‌کند. اگر دو لایه‌ی هم‌سطح دارای zIndex مساوی باشند، لایه‌ای که در آرایه‌ی layers[] بَعدتر قرار گرفته است، بعداً ظاهر می‌شود و لذا روی لایه‌ای که قبلاً ظاهر شده است، می‌افتد.

۳٫۲۲٫۵ روش‌ها

- **load(src, width)**

یک نشانی را به لایه بار می‌کند، مقدار پهنای لایه را مطابق مقدار داده شده تنظیم می‌کند، و چیزی بر نمی‌گرداند.

- **moveAbove(other_layer)**

این لایه را به بالای لایه‌ی دیگر می‌برد، و چیزی بر نمی‌گرداند.

- **moveBelow(other_layer)**

این لایه را به زیر لایه‌ی دیگر می‌برد، و چیزی بر نمی‌گرداند.

- **moveBy(dx, dy)**

لایه را نسبت به موقعیت فعلی حرکت می‌دهد، و چیزی بر نمی‌گرداند.

- **moveTo(x, y)**

لایه را به موقعیت (x, y) نسبت به لایه‌ی والد یا پنجره می‌برد، و چیزی بر نمی‌گرداند.

- **moveToAbsolute(x, y)**

لایه را به موقعیت (x, y) نسبت به صفحه می‌برد، و چیزی بر نمی‌گرداند.

- **resizeBy(dw, dh)**

اندازه‌ی لایه را به میزان داده شده تغییر می‌دهد، و چیزی بر نمی‌گرداند.

- **resizeTo(width, height)**

اندازه‌ی لایه را به اندازه‌ی داده شده تغییر می‌دهد، و چیزی بر نمی‌گرداند.

Link ۳٫۲۳

(جاوا اسکریپت سمت مشتری ۱٫۰)
یک پیوند <a> یا <area>.
مشتق شده از: Element

۳٫۲۳٫۱ دستور

```
document.links[i]
```

۳٫۲۳٫۲ خصلت‌ها

بسیاری از خصلت‌های پیوند، در واقع، معرف قسمتی از نشانی آن هستند. برای هر کدام از خصلت‌های زیر، مقداری که داده می‌شود، بر اساس نشانی (فرضی) زیر است:

```
http://www.oreilly.com:1234/catalog/search.html?
   q=JavaScript&m=10#results
```

- **hash**
یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت لنگر نشانی را، به همراه نویسه‌ی # ابتدای آن، مشخص می‌کند. مثال: "#result".

- **host**
یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت نام میزبان و شماره‌ی درگاه نشانی را مشخص می‌کند. مثال: "www.oreilly.com:1234".

- **hostname**
یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت نام میزبان نشانی را مشخص می‌کند. مثال: "www.oreilly.com".

- **href**
یک خصلت رشته‌ای خواندنی/نوشتنی که متن کامل نشانی را نشان می‌دهد.

- **pathname**
یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت نام مسیر نشانی را مشخص می‌کند. مثال: "/catalog/search.html".

- **port**

یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت شماره‌ی درگاه نشانی را مشخص می‌کند. مثال: "1234".

- **protocol**

یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت پروتکل نشانی را مشخص می‌کند. مثال: "http:".

- **search**

یک خصلت رشته‌ای خواندنی/نوشتنی که قسمت پرسجوی نشانی را، به همراه نویسه‌ی ? ابتدای آن، مشخص می‌کند. مثال: "?q=JavaScript&m=10".

- **target**

یک خصلت رشته‌ای خواندنی/نوشتنی که نام یک شیء پنجره (یعنی کادر یا پنجره‌ی سطح بالای مرورگر) را که سند پیوندی باید در آن نمایش داده شود، مشخص می‌کند. این خصلت معادل صفت استاندارد target در HTML است. در ضمن، می‌توان از نام‌های خاص "_blank"، "_top"، "_parent"، و "_self" استفاده کرد.

۳٫۲۳٫۳ رویدادپردازها

- **onclick**

وقتی کاربر روی پیوند کلیک می‌کند، فراخوانی می‌شود. در جاوا اسکریپت ۱٫۱، این رویدادپرداز می‌تواند با بر گرداندن مقدار false، مانع از تعقیب پیوند شود.

- **onmouseout**

وقتی کاربر موشواره را از روی عنصر بیرون می‌برد، فراخوانی می‌شود. جاوا اسکریپت ۱٫۱.

- **onmouseover**

وقتی کاربر موشواره را روی پیوند می‌برد، فراخوانی می‌شود. در اینجا می‌توان مقداری برای خصلت status پنجره‌ی فعلی تعیین کرد. اگر مقدار true بر گردانید، مرورگر می‌فهمد که نباید نشانی را در سطر

وضعیت نمایش دهد.

۳٫۲۳٫۴ ارجاع

Anchor، Location.

۳٫۲۴ Location

(جاوا اسکریپت سمت مشتری)
مکان کنونی مرورگر.

۳٫۲۴٫۱ دستور

```
location
window.location
```

۳٫۲۴٫۲ خصلت‌ها

شیء مکان دارای همان خصلت‌های مربوط به نشانی موجود در شیء پیوند است، به استثنای target. برای شرح hash، host، hostname، href، pathname، port، protocol، و search به شیء پیوند مراجعه کنید. تعیین هر کدام از این صفات سبب می‌شود که مرورگر سند را از نشانی جدید بار کند. به عنوان میانبر، می‌توانید با اختصاص دادن یک رشته‌ی نشانی به خصلت location پنجره، سند جدیدی بار کنید.

۳٫۲۴٫۳ روش‌ها

- **reload(force)**

سند جاری را از حافظه‌ی نهان و یا سرور دوباره می‌خواند. آوند force اختیاری است. اگر مقدار آن true باشد، باعث بازخوانی کامل می‌شود، ولو آنکه سند تغییر نکرده باشد. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱.

- **replace(url)**

سند جاری را با سند دیگری جایگزین می‌کند، بدون اینکه درایه‌ی جدیدی در سابقه‌ی مرورگر ایجاد کند. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱.

۳٫۲۴٫۴ ارجاع

Link، Window.location.

۳٫۲۵ Math

(جاوا اسکریپت هسته ۱٫۰، جی‌اسکریپت ۱٫۰، ECMA ۱)
تابع‌ها و ثابت‌های ریاضی.

۳٫۲۵٫۱ دستور

```
Math.constant
Math.function()
```

۳٫۲۵٫۲ شرح

شیء Math محلی برای تابع‌ها و ثابت‌های ریاضی است. این کلاس، بر خلاف Date و String، طبقه‌ای از اشیا را تعریف نمی‌کند. هیچگونه سازنده‌ی Math() وجود ندارد، و توابعی مانند Math.sin() فقط تابع‌اند، نه روش‌هایی که روی شیء عمل کنند.

۳٫۲۵٫۳ ثابت‌ها

- **Math.E**
  ثابت *e*، مبنای لگاریتم طبیعی.
- **Math.LN10**
  لگاریتم طبیعی ۱۰.
- **Math.LN2**
  لگاریتم طبیعی ۲.
- **Math.LOG10E**
  لگاریتم مبنای ۱۰ *e*.
- **Math.LOG2E**
  لگاریتم مبنای ۲ *e*.

- **Math.PI**
  ثابت π.
- **Math.SQRT1_2**
  ۱ تقسیم بر ریشه‌ی دوم ۲.
- **Math.SQRT2**
  ریشه‌ی دوم ۲.

۳٫۲۵٫۴ تابع‌ها

- **Math.abs(x)**
  قدر مطلق x را بر می‌گرداند.
- **Math.acos(x)**
  آرک‌کسینوس x را بر می‌گرداند؛ مقدار برگشتی بین ۰ و π رادیان است.
- **Math.asin(x)**
  آرک‌سینوس x را بر می‌گرداند؛ مقدار برگشتی بین $-\pi/2$ و $\pi/2$ رادیان است.
- **Math.atan(x)**
  آرک‌تانژانت x را بر می‌گرداند؛ مقدار برگشتی بین $-\pi/2$ و $\pi/2$ رادیان است.
- **Math.atan2(y, x)**
  مقداری بین $-\pi$ و π رادیان بر می‌گرداند که معرف زاویه‌ی بین قسمت مثبت محور X و نقطه‌ی (x, y) است. به ترتیب آوندهای تابع توجه کنید.
- **Math.ceil(x)**
  نزدیک‌ترین عدد صحیحی را که بزرگ‌تر یا برابر با x است، بر می‌گرداند.
- **Math.cos(x)**
  کسینوس مقدار داده شده‌ی x را بر می‌گرداند.

- **Math.exp(x)**
مقدار ثابت e به توان x را بر می‌گرداند.

- **Math.floor(x)**
نزدیک‌ترین عدد صحیحی را که کوچک‌تر یا برابر با x است، بر می‌گرداند.

- **Math.log(x)**
لگاریتم طبیعی x را بر می‌گرداند.

- **Math.max(args...)**
بزرگ‌ترین آوند را از میان آوندهای داده شده بر می‌گرداند. اگر آوندی داده نشده باشد، Infinity بر می‌گرداند. اگر یکی از آوندها NaN باشد، و یا مقداری غیرعددی باشد که نتوان آن را به عدد تبدیل کرد، NaN بر می‌گرداند. تا قبل از ECMA ۳، این تابع دقیقاً به ۲ آوند نیاز داشت.

- **Math.min(args...)**
کوچک‌ترین آوند را از میان آوندهای داده شده بر می‌گرداند. اگر آوندی داده نشده باشد، Infinity بر می‌گرداند. اگر یکی از آوندها NaN باشد، و یا مقداری غیرعددی باشد که نتوان آن را به عدد تبدیل کرد، NaN بر می‌گرداند. تا قبل از ECMA ۳، این تابع دقیقاً به ۲ آوند نیاز داشت.

- **Math.pow(x, y)**
مقدار x به توان y را بر می‌گرداند.

- **Math.random()**
یک عدد شبه‌تصادفی را بین ۰٫۰ و ۱٫۰ بر می‌گرداند. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱.

- **Math.round(x)**
نزدیک‌ترین عدد صحیح به x را بر می‌گرداند.

- **Math.sin(x)**
سینوس x را بر می‌گرداند.

- **Math.sqrt(x)**
جذر x را بر می‌گرداند. اگر x کمتر از صفر باشد، NaN بر می‌گرداند.

- **Math.tan(x)**
تانژانت x را بر می‌گرداند.

۳٫۲۵٫۵ ارجاع

Number.

۳٫۲۶ Navigator

(جاوا اسکریپت سمت مشتری ۱٫۰)
اطلاعات در باره‌ی مرورگر.

۳٫۲۶٫۱ دستور

```
navigator
```

۳٫۲۶٫۲ خصلت‌ها

- **appCodeName**
رشته‌ای فقط-خواندنی که لقب مرورگر را بیان می‌کند. این لقب در تمام مرورگرهای نت‌اسکیپ "Mozilla" است. برای سازگاری، در مرورگرهای اینترنت اکسپلورر هم این لقب "Mozilla" است.

- **appName**
رشته‌ای فقط-خواندنی که نام مرورگر را بیان می‌کند. برای نت‌اسکیپ، مقدار این خصلت "Netscape" است. برای اینترنت اکسپلورر، مقدار این خصلت "Microsoft Internet Explorer" است.

- **appVersion**
رشته‌ای فقط-خواندنی که اطلاعات مربوط به روایت و سکوی مرورگر را بیان می‌کند. قسمت اول این رشته، شماره‌ی روایت است. اگر این رشته را به تابع parseInt() بدهید، فقط مقدار روایت اصلی را بر می‌گرداند، و اگر آن را به تابع parseFloat() بدهید، مقدار روایت اصلی و فرعی را به صورت یک عدد ممیز شناور بر می‌گرداند. بقیه‌ی مقدار

رشته‌ای این خصلت، جزئیات دیگری را در باره‌ی روایت مرورگر، از جمله سیستم عاملی که روی آن اجرا می‌شود، را نشان می‌دهد. اما متأسفانه قالب این اطلاعات بین مرورگرهای مختلف بسیار متغیر است.

- **cookieEnabled**

یک مقدار بولی فقط-خواندنی که نشان می‌دهد کوکی‌ها در مرورگر فعال هستند (true) یا نه (false). اینترنت اکسپلورر ۴، نت‌اسکیپ ۶.

- **language**

رشته‌ای فقط-خواندنی که زبان پیش‌فرض روایت مرورگر را نشان می‌دهد. مقدار این خصلت، رمز استاندارد دوحرفی زبان مورد نظر است، مثلاً "en" برای انگلیسی و "fr" برای فرانسوی [و یا "fa" برای فارسی]. در ضمن، ممکن است رشته‌ای پنج حرفی باشد که علاوه بر زبان، ناحیه‌ی جغرافیایی را هم نشان می‌دهد، مثلاً "fr_CA" که معرف فرانسوی کانادا است [و یا "fa_IR" برای فارسی ایران]. نت‌اسکیپ ۴؛ دقت کنید که اینترنت اکسپلورر ۴ دو خصلت متفاوت در رابطه با زبان ارائه می‌کند.

- **platform**

رشته‌ای فقط-خواندنی که سیستم عامل و/یا سکوی سخت‌افزاری مورد نیاز مرورگر را نشان می‌دهد. گرچه مجموعه‌ی مقادیر استانداردی برای این خصلت وجود ندارد، ولی برخی از مقادیر معمول آن "Win32"، "MacPPC"، و "Linux i586" هستند. جی‌اسکریپت ۱٫۲.

- **systemLanguage**

رشته‌ای فقط-خواندنی که زبان پیش‌فرض سیستم عامل را با همان رمزهایی که در مورد خصلت ویژه‌ی نت‌اسکیپ language گفته شد، بیان می‌کند. اینترنت اکسپلورر ۴.

- **userAgent**

رشته‌ای فقط-خواندنی که مقدار مورد استفاده برای سرفصل عامل کاربر در تقاضاهای HTTP توسط مرورگر را تعیین می‌کند. معمولاً مقدار آن متشکل از مقدار navigator.appCodeName به علاوه‌ی یک کج‌خط و بعد مقدار navigator.appVersion است.

- **userLanguage**

رشته‌ای فقط-خواندنی که زبان مرجح کاربر را با همان رمزهایی که در مورد خصلت ویژه‌ی نت‌اسکیپ language گفته شد، بیان می‌کند. اینترنت اکسپلورر ۴.

۳٫۲۶٫۳ روش‌ها

- **javaEnabled()**

اگر جاوا در مرورگر فعلی پشتیبانی شده و فعال باشد، مقدار true، و در غیر این صورت، مقدار false بر می‌گرداند. جاوا اسکریپت ۱٫۱.

۳٫۲۶٫۴ ارجاع

.Screen

Node ۳٫۲۷

(DOM سطح ۱)
گرهی در درخت سند.

۳٫۲۷٫۱ زیرکلاس‌ها

Attr، Comment، Document، DocumentFragment،
Element، Text.

۳٫۲۷٫۲ ثابت‌ها

تمام گره‌ها در یک سند HTML نمونه‌هایی از زیرکلاس‌های گره هستند که در بالا بر شمرده شده‌اند. هر شیء گره یک خصلت nodeType دارد که مشخص می‌کند نمونه‌ای از کدامیک از زیرکلاس‌های فوق است. ثابت‌های زیر، مقادیر قانونی nodeType هستند. دقت کنید که اینها خصلت‌های ایستای Node هستند، نه خصلت‌های اشیایی که از نوع گره ساخته می‌شوند. اینها در اینترنت اکسپلورر ۴، ۵، و یا ۶ تعریف نشده‌اند؛ در مرورگرهای مذکور باید از مقادیر عددی مربوطه استفاده کنید:

```
Node.ELEMENT_NODE           = 1;  // Element
Node.ATTRIBUTE_NODE         = 2;  // Attr
Node.TEXT_NODE              = 3;  // Text
```

```
Node.COMMENT_NODE            = 8;  // Comment
Node.DOCUMENT_NODE           = 9;  // Document
Node.DOCUMENT_FRAGMENT_NODE = 11; // DocumentFragment
```

۳٫۲۷٫۳ خصلت‌ها

- **attributes[]**

اگر گرهی یک عنصر باشد، خصلت attributes آرایه‌ای فقط-خواندنی از اشیای Attr است که معرف صفت‌های عنصر هستند. این آرایه را می‌توان به شماره یا نام صفت نمایه‌دهی کرد. چون برای هر صفت HTML یک خصلت متناظر در عنصر وجود دارد، لذا معمولاً از آرایه‌ی attributes[] استفاده نمی‌شود.

- **childNodes[]**

آرایه‌ای فقط-خواندنی از گره‌ها است که حاوی گره‌های فرزند این گره است. اگر گره فرزندی نداشته باشد، این خصلت آرایه‌ای با طول صفر است.

- **firstChild**

این خصلت فقط-خواندنی به اولین گره فرزند این گره اشاره می‌کند، و اگر گره فرزندی نداشته باشد، مقدار آن null است.

- **lastChild**

این خصلت فقط-خواندنی به آخرین گره فرزند این گره اشاره می‌کند، و اگر گره فرزندی نداشته باشد، مقدار آن null است.

- **nextSibling**

گره خواهری که در آرایه‌ی childNodes[] مربوط به parentNode بلافاصله بعد از این گره قرار می‌گیرد، و یا null در زمانی که چنین گرهی وجود نداشته باشد. فقط-خواندنی.

- **nodeName**

نام گره. برای گره‌های عنصر، این خصلت نام عنصر را مشخص می‌کند، که می‌توان با خصلت tagName نیز آن را به دست آورد. برای گره‌های Attr، این خصلت نام صفت را مشخص می‌کند. برای انواع دیگر گره، این مقدار یک رشته‌ی ثابت است که نوع گره را مشخص می‌کند. فقط-خواندنی.

- **nodeType**

نوع گره. مقادیر قانونی این خصلت ثابت‌هایی هستند که در بالا گفته شد.

- **nodeValue**

مقدار رشته‌ای یک گره. برای گره‌های متن و توضیح، این خصلت حاوی محتوای متنی گره است. برای گره‌های Attr، حاوی مقدار صفت است. این خصلت خواندنی/نوشتنی است.

- **ownerDocument**

شیء سندی که این گره بخشی از آن است. برای گره‌های سند، این خصلت null است. فقط-خواندنی.

- **parentNode**

گره والد یا گنجایه‌ی این گره، و یا null در صورتی که والدی نداشته باشد. دقت کنید که گره‌های سند و Attr هرگز گره والد ندارند. در گره‌هایی که از سند بر داشته شده و یا جدیداً ایجاد شده‌اند و هنوز به درخت سند افزوده نشده‌اند، مقدار این خصلت null است. فقط-خواندنی.

- **previousSibling**

گره خواهری که در آرایه‌ی childNodes[] مربوط به parentNode بلافاصله قبل از این گره قرار می‌گیرد، و یا null در زمانی که چنین گرهی وجود نداشته باشد.

۳٫۲۷٫۴ روش‌ها

- **addEventListener(type, listener, useCapture)**

یک رویدادشنو را برای این گره به ثبت می‌رساند. type رشته‌ای است که نوع رویداد را با برداشتن حروف "on" از آغاز آن بیان می‌کند (مثلاً، click یا submit). listener تابع رویدادپرداز است. وقتی این تابع فرا خوانده می‌شود، شیئی از نوع Event به عنوان آوند به آن داده می‌شود. اگر مقدار useCapture صحیح (true) باشد، این رویدادپرداز از نوع گیرنده است. اگر false باشد و یا وجود نداشته باشد، یک رویدادپرداز عادی است. چیزی بر نمی‌گرداند. DOM سطح ۲؛

در اینترنت اکسپلورر ۴، ۵، و یا ۶ پشتیبانی نمی‌شود.

- **appendChild(newChild)**

گره newChild را با افزودن به انتهای آرایه‌ی childNodes[] این گره، به درخت سند اضافه می‌کند. اگر گره از قبل در درخت سند باشد، اول بر داشته می‌شود، و بعد در موقعیت جدید اضافه می‌شود. آوند newChild را بر می‌گرداند.

- **cloneNode(deep)**

نسخه‌ی دیگری از این گره بر می‌گرداند. اگر مقدار deep صحیح (true) باشد، فرزندان گره نیز به صورت تراجعی نسخه‌برداری می‌شوند.

- **hasAttributes()**

اگر این گره یک عنصر دارای برخی صفات باشد، صحیح بر می‌گرداند. DOM سطح ۲.

- **hasChildNodes()**

اگر گره فرزندی داشته باشد، صحیح بر می‌گرداند.

- **insertBefore(newChild, refChild)**

گره newChild را درست قبل از گره refChild به درخت سند اضافه می‌کند. refChild باید فرزندی از این گره باشد. اگر گره newChild از قبل در درخت سند باشد، اول بر داشته می‌شود. newChild را بر می‌گرداند.

- **isSupported(feature, version)**

اگر ویژگی مورد نظر با روایت مشخص شده پشتیبانی شده باشد، مقدار صحیح بر می‌گرداند. همچنین، رک. DOMImplementation.hasFeature(). DOM سطح ۲.

- **normalize()**

تمام فرزندان این گره را که از نوع گره متنی هستند، بهنجار می‌کند، یعنی گره‌های متنی خالی را حذف می‌کند، و گره‌های متنی مجاور را با هم ادغام می‌کند. چیزی بر نمی‌گرداند.

- **removeChild(oldChild)**

گره oldChild را از درخت سند بر می‌دارد. oldChild باید فرزندی از این گره باشد. oldChild را بر می‌گرداند.

- **removeEventListener(type, listener, useCapture)**

رویدادشنوی مشخص شده را بر می‌گرداند. چیزی بر نمی‌گرداند. DOM سطح ۲؛ در اینترنت اکسپلورر ۴، ۵، و یا ۶ پشتیبانی نمی‌شود.

- **replaceChild(newChild, oldChild)**

به جای گره oldChild (که باید از فرزندان این گره باشد)، گره newChild را می‌گذارد. اگر newChild از قبل در درخت سند باشد، اول از جای قبلی بر داشته می‌شود. oldChild را بر می‌گرداند.

۳٫۲۷٫۵ ارجاع

Attr، Comment، Document، DocumentFragment، Element، Text.

۳٫۲۸ Number

(جاوا اسکریپت هسته ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱)
پشتیبانی برای اعداد.

۳٫۲۸٫۱ سازنده

```
new Number(value)
Number(value)
```

سازنده‌ی Number() به همراه عملگر new، آوند خود را به مقداری عددی تبدیل می‌کند، و شیء عدد جدیدی را که به دور آن مقدار پیچیده شده است، بر می‌گرداند. بدون new، Number() یک تابع تبدیلی است که آوند خود را به عدد تبدیل می‌کند، و آن مقدار را بر می‌گرداند.

۳٫۲۸٫۲ ثابت‌ها

این ثابت‌ها خصلت‌های خود Number هستند، نه شیءهای عدد ساخته شده از آن.

- **Number.MAX_VALUE**
بزرگ‌ترین عدد قابل نمایش. تقریباً 1.79E+308.

- **Number.MIN_VALUE**
کوچک‌ترین عدد قابل نمایش. تقریباً 5E-324.

- **Number.NaN**
مقدار غیرعددی. همان NaN سراسری.

- **Number.NEGATIVE_INFINITY**
مقدار منهای بی‌نهایت.

- **Number.POSITIVE_INFINITY**
مقدار بی‌نهایت. همان Infinity سراسری.

۳٫۲۸٫۳ روش‌ها

- **toExponential(digits)**

نمایش رشته‌ای عدد را به صورت نمایی با یک رقم صحیح و digits رقم اعشاری بر می‌گرداند. برای رسیدن به طول مورد نظر، قسمت کسری گرد شده و یا به آن صفر اضافه می‌شود. مقدار digits باید بین ۰ و ۲۰ باشد، و اگر حذف شود، هر تعداد رقم لازم باشد، استفاده می‌شود. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toFixed(digits)**

نمایش رشته‌ای عدد را به صورت غیرنمایی دقیقاً با تعداد digits رقم اعشاری بر می‌گرداند. مقدار digits باید بین ۰ و ۲۰ باشد. برای رسیدن به طول مورد نظر، قسمت کسری گرد شده و یا به آن صفر اضافه می‌شود. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toLocaleString()**

نمایش رشته‌ای خاصی از عدد را که بر اساس قراردادهای محلی آراسته شده است، بر می‌گرداند. مثلاً این آرایش شامل نوع نویسه‌ی مورد استفاده برای ممیز اعشاری و جدا کننده‌ی هزارگان است. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toPrecision(precision)**

نمایش رشته‌ای عدد را با تعداد رقم معنی‌دار معادل precision بر می‌گرداند. precision باید بین ۱ و ۲۱ باشد. رشته‌ی بر گردانده شده در صورت امکان از نماد ممیز ثابت و یا از نماد نمایی استفاده می‌کند. برای رسیدن به طول مورد نظر، قسمت کسری گرد شده و یا به آن صفر اضافه می‌شود. جاوا اسکریپت ۱٫۵؛ جی‌اسکریپت ۵٫۵؛ ECMA ۳.

- **toString(radix)**

عدد را در مبنای داده شده به رشته تبدیل کرده و این رشته را بر می‌گرداند. radix باید بین ۲ و ۳۶ باشد. اگر حذف شود، از مبنای ۱۰ استفاده می‌شود.

۳٫۲۸٫۴ ارجاع

Math

Object ۳٫۲۹

(جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱)
کلاس والد همه‌ی اشیای جاوا اسکریپت.

۳٫۲۹٫۱ سازنده

```
new Object();
```

این سازنده شیئی خالی ایجاد می‌کند که می‌توانید خصلت‌های دلخواه به آن اضافه کنید.

۳٫۲۹٫۲ خصلت‌ها

تمام اشیای جاوا اسکریپت، صرف نظر از اینکه چگونه ایجاد شده باشند، دارای خصلت‌های زیر هستند.

- **constructor**

اشاره‌ای به تابع جاوا اسکریپتی که سازنده‌ی شیء بوده است. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۲٫۰؛ ECMA ۱.

۳/۲۹/۳ روش‌ها

تمام اشیای جاوا اسکریپت، صرف نظر از اینکه چگونه ایجاد شده باشند، دارای روش‌های زیر هستند.

- **hasOwnProperty(propname)**

اگر شیء خصلتی غیرموروثی با نام مشخص شده داشته باشد، true بر می‌گرداند. اما اگر شیء فاقد آن خصلت باشد، و یا آن را از شیء سرمشق خود به ارث برده باشد، false بر می‌گرداند. جاوا اسکریپت ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

- **isPrototypeOf(o)**

اگر این شیء سرمشق o باشد، true بر می‌گرداند. اما اگر o یک شیء نباشد، و یا این شیء سرمشق آن نباشد، false بر می‌گرداند. جاوا اسکریپت ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

- **propertyIsEnumerable(propname)**

اگر شیء یک خصلت برشمردنی با نام مشخص شده داشته باشد، درست بر می‌گرداند، و الا غلط بر می‌گرداند. خصلت‌های برشمردنی خصلت‌هایی هستند که در حلقه‌های for/in بر شمرده می‌شوند. جاوا اسکریپت ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

- **toLocaleString()**

نمایش رشته‌ای محلی شده‌ی این شیء را بر می‌گرداند. در پیاده‌سازی پیش‌فرض، این رشته صرفاً ()toString را فرا می‌خواند، اما زیرکلاس‌ها می‌توانند برای محلی‌سازی آن را تغییر دهند. جاوا اسکریپت ۱/۵؛ جی‌اسکریپت ۵/۵؛ ECMA ۳.

- **toString()**

نمایش رشته‌ای شیء را بر می‌گرداند. پیاده‌سازی پیش‌فرضی که در کلاس Object ارائه شده، خیلی عمومی است، و چندان اطلاعات مفیدی به ما نمی‌دهد. لیکن زیرکلاس‌ها معمولاً این روش را جایگزین می‌کنند، و آن را به گونه‌ای تعریف می‌کنند که اطلاعات سودمندتری ارائه دهد. جاوا اسکریپت ۱/۰؛ جی‌اسکریپت ۲/۰؛ ECMA ۱.

- **valueOf()**

مقدار بدوی شیء را در صورت وجود بر می‌گرداند. در مورد اشیایی که از نوع Object هستند، این روش صرفاً خود شیء را بر می‌گرداند. اما زیرکلاس‌هایی همچون عدد و بولی، مقدار بدوی مربوط به خود را بر می‌گردانند. جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۲٫۰، ECMA ۱.

۳٫۲۹٫۴ ارجاع

Array، Boolean، Function، Number، String.

۳٫۳۰ Option

(جاوا اسکریپت سمت مشتری ۱٫۰)
یک گزینه‌ی قابل انتخاب.
مشتق شده از: Element

۳٫۳۰٫۱ دستور

```
select.options[i]
```

۳٫۳۰٫۲ سازنده

در جاوا اسکریپت ۱٫۱ و بعد از آن، می‌توان اشیای گزینه را به صورت پویا با استفاده از سازنده‌ی Option() ایجاد کرد:

```
new Option(text, value, defaultSelected, selected)
```

۳٫۳۰٫۳ خصلت‌ها

- **defaultSelected**

یک مقدار بولی خواندنی/نوشتنی که مشخص می‌کند در آغاز این گزینه از Select انتخاب شده است یا نه.

- **index**

عدد صحیح فقط-خواندنی که شماره‌ی این گزینه را در آرایه‌ی options[] شیء انتخاب حاوی آن نشان می‌دهد.

- **selected**

یک مقدار بولی خواندنی/نوشتنی که مشخص می‌کند در حال حاضر این

گزینه انتخاب شده است یا نه. برای بررسی انتخاب شدن یک گزینه می‌توانید از این خصلت استفاده کنید. به علاوه، با دادن مقدار جدید به آن می‌توانید گزینه را انتخاب کنید یا بر عکس. دقت کنید که وقتی به این صورت انتخاب گزینه را تغییر می‌دهید، رویدادپرداز `Select.onchange()` فرا خوانده نمی‌شود.

- **text**

رشته‌ای خواندنی/نوشتنی که به عنوان گزینه برای کاربر نشان داده می‌شود.

- **value**

رشته‌ای خواندنی/نوشتنی که در صورت انتخاب شدن این گزینه در هنگام تحویل داده شدن فرم به سرور شبکه داده می‌شود.

۳٫۳۰٫۴ ارجاع

`Select`.

RegExp ۳٫۳۱

(جاوا اسکریپت هسته ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳)
عبارت‌های مرتب برای انطباق الگو.

۳٫۳۱٫۱ دستور مستقیم

/pattern/attributes

۳٫۳۱٫۲ سازنده

`new RegExp(`*pattern*`, `*attributes*`)`

الگوهای عبارت‌های مرتب با استفاده از دستور پیچیده‌ای بیان می‌شوند که قبلاً در این کتاب در باره‌ی آنها صحبت کردیم.

۳٫۳۱٫۳ خصلت‌های اشیای نمونه

- **global**

یک مقدار بولی فقط-خواندنی که مشخص می‌کند که این شیء دارای صفت `g` است، و لذا انطباق را به صورت سراسری انجام می‌دهد.

- **ignoreCase**

یک مقدار بولی فقط-خواندنی که مشخص می‌کند که این شیء دارای صفت i است، و لذا انطباق را بدون توجه به حروف کوچک و بزرگ انجام می‌دهد.

- **lastIndex**

برای اشیای RegExp سراسری، این خصلت خواندنی/نوشتنی موقعیت نویسه‌ی بلافاصله بعد از آخرین انطباق را نشان می‌دهد؛ این اولین نویسه‌ای است که برای انطباق بعدی مورد بررسی قرار می‌گیرد.

- **multiline**

یک مقدار بولی فقط-خواندنی که مشخص می‌کند که این شیء دارای صفت m است، و لذا انطباق را به صورت چندسطری انجام می‌دهد.

- **source**

رشته‌ای فقط-خواندنی که متن مبدأ عبارت مرتب pattern را به استثنای کج‌خط‌ها و صفت‌ها در خود نگه می‌دارد.

۳/۳۱/۴ روش‌ها

- **exec(string)**

رشته را با این عبارت مرتب انطباق می‌دهد، و آرایه‌ای از نتایج انطباق را بر می‌گرداند، و یا اگر هیچ انطباقی یافت نشد، null بر می‌گرداند. عنصر ۰ آرایه عبارت انطباق یافته است. عناصر بعدی آرایه حاوی زیررشته‌هایی هستند که با زیرعبارت‌های موجود در عبارت مرتب منطبق شده‌اند. آرایه‌ی برگردانده شده یک خصلت index نیز دارد که موقعیت شروع انطباق را بر می‌گرداند.

- **test(string)**

اگر string حاوی متن قابل انطباق با این عبارت مرتب باشد، صحیح بر می‌گرداند، وگرنه غلط.

۳/۳۱/۵ ارجاع

String.match()، String.replace()،

.String.search()

Screen ۳٫۳۲

(جاوا اسکریپت سمت مشتری ۱٫۲)
اطلاعات در باره‌ی صفحه‌ی نمایش.

۳٫۳۲٫۱ دستور

screen

۳٫۳۲٫۲ خصلت‌ها

- **availHeight**
بلندای موجود صفحه بر حسب پیکسل.
- **availWidth**
پهنای موجود صفحه بر حسب پیکسل.
- **colorDepth**
ژرفای جعبه‌رنگ مرورگر، و یا تعداد بیت‌برپیکسل صفحه‌ی نمایش.
- **height**
بلندای کلی صفحه بر حسب پیکسل.
- **width**
پهنای کلی صفحه بر حسب پیکسل.

۳٫۳۲٫۳ ارجاع

.Navigator

Select ۳٫۳۳

(جی‌اسکریپت سمت مشتری ۱٫۰)
یک لیست انتخاب گرافیکی
مشتق شده از: Element

۳٫۳۳٫۱ دستور

```
form.elements[i]
form.elements[element_name]
form.element_name
```

۳٫۳۳٫۲ خصلت‌ها

شیء انتخاب یک خصلت برای هر کدام از صفت‌های برگه‌ی <select> در HTML، از قبیل disabled، multiple، name، و size، دارد. علاوه بر این، خصلت‌های زیر را هم تعریف کرده است:

- **form**

شیء فرمی که حاوی این شیء انتخاب است. فقط-خواندنی.

- **length**

یک عدد صحیح فقط-خواندنی که تعداد عنصرهای موجود آرایه‌ی options[] را مشخص می‌کند. مقدار این خصلت برابر با options.length است.

- **options[]**

آرایه‌ای از اشیای گزینه که هر کدام معرف یکی از گزینه‌های موجود در شیء انتخاب است. می‌توانید با دادن مقدار کمتری به options.length، این آرایه را کوتاه‌تر کنید (حتی با دادن مقدار صفر می‌توانید تمام گزینه‌های آن را بر دارید). با دادن مقدار null به یکی از عناصر آرایه، می‌توانید گزینه‌ی مربوط به آن را بر دارید — این کار بقیه‌ی گزینه‌های بعد از آن را جا به جا می‌کند، و آرایه را یک عنصر کوتاه‌تر می‌کند. می‌توانید با استفاده از سازنده‌ی Option() یک گزینه‌ی جدید ایجاد کنید و آن را به options[options.length] اختصاص دهید، و به این ترتیب یک گزینه به انتهای گزینه‌ها اضافه کنید.

- **selectedIndex**

یک عدد صحیح خواندنی/نوشتنی که شماره‌ی گزینه‌ی انتخاب شده را در شیء انتخاب نشان می‌دهد. اگر هیچ گزینه‌ای انتخاب نشده باشد، مقدار این خصلت 1- است. اگر بیش از یک گزینه انتخاب شده باشند، selectedIndex فقط شماره‌ی اولین گزینه‌ی انتخاب شده را نشان

می‌دهد. اگر به این خصلت مقداری داده شود، تمام گزینه‌های دیگر از حالت انتخاب خارج می‌شوند. اگر مقدار 1- شود، تمام گزینه‌ها از حالت انتخاب خارج می‌شوند.

- **type**

یک خصلت رشته‌ای فقط-خواندنی که نوع عنصر را مشخص می‌کند. اگر شیء انتخاب فقط اجازه‌ی انتخاب یک گزینه را بدهد (یعنی در تعریف HTML آن صفت multiple ذکر نشده باشد)، مقدار این خصلت "select-one" است. در غیر این صورت، مقدار آن "select-multiple" است. همچنین، رک. Input.type. جاوا اسکریپت ۱/۱.

۳/۳۳/۳ روش‌ها

- **add(new, old)**

شیء گزینه‌ی new را درست قبل از گزینه‌ی old در آرایه‌ی options[] درج می‌کند. اگر old مقدار null داشته باشد، گزینه‌ی new به انتهای آرایه اضافه می‌شود. چیزی بر نمی‌گرداند. DOM سطح ۱.

- **blur()**

کانون صفحه‌کلید را تحویل می‌دهد و چیزی بر نمی‌گرداند.

- **focus()**

کانون صفحه‌کلید را در اختیار می‌گیرد و چیزی بر نمی‌گرداند.

- **remove(n)**

عنصر n-اُم آرایه‌ی options[] را بر می‌دارد. چیزی بر نمی‌گرداند. DOM سطح ۱.

۳/۳۳/۴ رویدادپردازها

- **onblur**

وقتی کانون ورودی از دست می‌رود، فرا خوانده می‌شود.

- **onchange**

وقتی کاربر یکی از اقلام را انتخاب می‌کند یا از حالت انتخاب شده خارج

می‌کند، فرا خوانده می‌شود.

- **onfocus**

وقتی کانون ورودی به دست می‌آید، فرا خوانده می‌شود.

۳٫۳۳٫۵ ارجاع

Form، Input، Option.

۳٫۳۴ String

(جاوا اسکریپت هسته ۱٫۰؛ جی‌اسکریپت ۱٫۰؛ ECMA ۱)
کار با رشته‌ها.
مشتق شده از: Element

۳٫۳۴٫۱ سازنده

```
String(s)
new String(s)
```

بدون عملگر new، تابع String() آوند خود را به یک رشته تبدیل می‌کند. با عملگر new، String() سازنده‌ای است که مقدار داده شده را در یک شیء رشته قرار می‌دهد.

۳٫۳۴٫۲ خصلت‌ها

- **length**

تعداد نویسه‌ها در رشته. فقط-خواندنی.

۳٫۳۴٫۳ روش‌ها

- **charAt(n)**

نویسه‌ی موجود در موقعیت n در رشته را بر می‌گرداند.

- **charCodeAt(n)**

رمز یونیکد نویسه‌ی موجود در موقعیت n در رشته را بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۵٫۵؛ ECMA ۱.

- **concat(value, ...)**

رشته‌ی جدیدی بر می‌گرداند که از تبدیل کردن هر یک از آوندها به رشته و ادغام رشته‌های حاصله به دست می‌آید. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **indexOf(substring, start)**

موقعیت اولین ظهور `substring` در این رشته را از نمایه‌ی `start` به بعد بر می‌گرداند، و اگر یافت نشود، مقدار `-1` بر می‌گرداند. اگر `start` حذف شود، مقدار آن صفر منظور می‌شود.

- **lastIndexOf(substring, start)**

موقعیت آخرین ظهور `substring` در این رشته را تا نمایه‌ی `start` بر می‌گرداند، و اگر یافت نشود، مقدار `-1` بر می‌گرداند. اگر `start` حذف شود، مقدار آن برابر طول رشته منظور می‌شود.

- **match(regexp)**

این رشته را با عبارت مرتب داده شده مقایسه می‌کند، و آرایه‌ای متشکل از نتایج انطباق بر می‌گرداند، و اگر هیچگونه انطباقی یافت نشده باشد، `null` بر می‌گرداند. اگر `regexp` یک عبارت مرتب سراسری نباشد، در آن صورت، آرایه‌ی بر گردانده شده همان حاصل روش `RegExp.exec()` است. اگر `regexp` سراسری باشد (یعنی صفت `"g"` داشته باشد)، عناصر آرایه‌ی بر گردانده شده حاوی متن انطباق‌های یافت شده است. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **replace(regexp, replacement)**

رشته‌ی جدیدی بر می‌گرداند که در آن متن منطبق شده با `regexp` با `replacement` جایگزین شده است. `regexp` ممکن است یک عبارت مرتب و یا یک رشته‌ی ساده باشد. `replacement` ممکن است یک رشته باشد، که حاوی سلسله‌های گریز اختیاری عبارت مرتب (از قبیل `$1`) باشد که به وسیله‌ی قسمت‌هایی از متن منطبق شده جایگزین می‌شوند. در ضمن، می‌تواند تابعی باشد که رشته‌ی جایگزین را بر اساس جزئیات انطباق داده شده در آوندها محاسبه می‌کند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **search(regexp)**

موقعیت شروع اولین زیررشته‌ی این رشته را که با regexp منطبق می‌شود، بر می‌گرداند، و یا اگر انطباقی یافت نشود، 1- بر می‌گرداند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **slice(start, end)**

رشته‌ی جدیدی را بر می‌گرداند که حاوی تمام نویسه‌های این رشته از شماره‌ی start (شامل خود آن) تا شماره‌ی end (اما بدون آن) است. اگر end حذف شود، قطعه تا پایان رشته ادامه می‌یابد. آوندهای منفی به معنای شمارش از آخر رشته هستند. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۳.

- **split(delimiter, limit)**

آرایه‌ای از رشته‌ها بر می‌گرداند که از در هم شکستن رشته تشکیل می‌شود؛ محل‌های قطع رشته به وسیله‌ی delimiter مشخص می‌شود. delimiter ممکن است یک رشته یا یک عبارت مرتب باشد. اگر delimiter یک عبارت مرتب با زیرعبارتی در درون پرانتز باشد، متن جدا کننده‌ی منطبق شده با زیرعبارت در آرایه‌ی بر گردانده شده گنجانده می‌شود. همچنین، رک. Array.join() جاوا اسکریپت ۱٫۱؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

- **substring(from, to)**

رشته‌ی جدیدی متشکل از نویسه‌های شماره‌ی from تا to - 1 این رشته بر می‌گرداند. اگر to حذف شود، زیررشته تا آخر رشته ادامه می‌یابد. آوندهای منفی مجاز نیستند.

- **substr(start, length)**

رشته‌ی جدیدی متشکل از تعداد length نویسه از شماره‌ی from این رشته بر می‌گرداند؛ اگر length حذف شود، زیررشته تا آخر رشته ادامه می‌یابد. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ این تابع غیراستاندارد است، و بهتر است به جای آن از slice() یا substring() استفاده کنید.

- **toLowerCase()**

نسخه‌ی دیگری از این رشته را بر می‌گرداند، که در آن، تمام حروف

بزرگ در صورت موجود بودن، به حرف کوچک معادل خود تبدیل شده‌اند.

- **`toUpperCase()`**

نسخه‌ی دیگری از این رشته را بر می‌گرداند، که در آن، تمام حروف کوچک در صورت موجود بودن، به حرف بزرگ معادل خود تبدیل شده‌اند.

۳٫۳۴٫۴ توابع ایستا

- **`String.fromCharCode(c1, c2, ...)`**

رشته‌ی جدیدی بر می‌گرداند که از نویسه‌هایی با رمزهای داده شده تشکیل شده است. جاوا اسکریپت ۱٫۲؛ جی‌اسکریپت ۳٫۰؛ ECMA ۱.

۳٫۳۵ Style

(DOM سطح ۲؛ اینترنت اکسپلورر ۴)
خصلت‌های CSS مستقیم یک عنصر.

۳٫۳۵٫۱ دستور

```
element.style
```

۳٫۳۵٫۲ خصلت‌ها

شیء شیوه خصلت‌های زیادی دارد: یعنی به ازای هر صفت CSS تعریف شده در تعیین‌نامه‌ی CSS2، یک خصلت معادل دارد. نام خصلت معادل نام صفت است، با مختصر تغییراتی که برای اجتناب از خطاهای دستوری در جاوا اسکریپت لازم است. صفت‌های چندکلمه‌ای که حاوی خط فاصله هستند، مانند `font-family`، در جاوا اسکریپت بدون خط فاصله نوشته می‌شوند، و بعد از کلمه‌ی اول، هر کدام از کلمات بعدی با حروف بزرگ شروع می‌شوند: `fontFamily`. به علاوه، از آنجا که نام صفت `float` در جاوا اسکریپت یک کلمه‌ی ذخیره شده است، لذا به جاوا اسکریپت آن از `cssFloat` استفاده می‌شود.

خصلت‌های بصری CSS در جدول زیر نشان داده شده‌اند. از آنجا که خصلت‌ها مستقیماً متناظر با صفت‌های CSS هستند، لذا برای هر کدام در اینجا شرح جداگانه‌ای ارائه نمی‌شود. برای اطلاع از معنا و مقادیر قانونی هر صفت، به یک مرجع

CSS مراجعه کنید. دقت کنید که مرورگرهای فعلی همه‌ی این خصلت‌ها را پیاده‌سازی نمی‌کنند.

تمام خصلت‌ها رشته‌ای‌اند، و در کار کردن با خصلت‌هایی که مقادیر عددی دارند، باید دقت کرد. برای گرفتن مقدار این خصلت‌ها باید از `parseFloat()` برای تبدیل کردن رشته به عدد استفاده کنید. برای مقداردهی به اینگونه خصلت‌ها، باید عدد را به رشته تبدیل کنید، که معمولاً این کار را با افزودن واحد آن انجام می‌دهید، مانند `"px"`.

```
background
backgroundAttachment
backgroundColor
backgroundImage
backgroundPosition
backgroundRepeat
border
borderBottom
borderBottomColor
borderBottomStyle
borderBottomWidth
borderCollapse
borderColor
borderLeft
borderLeftColor
borderLeftStyle
borderLeftWidth
borderRight
borderRightColor
borderRightStyle
borderRightWidth
borderSpacing
borderStyle
borderTop
borderTopColor
borderTopStyle
borderTopWidth
borderWidth
bottom
captionSide
clear
clip
color
content
counterIncrement
counterReset
cssFloat
cursor
direction
display
emptyCells
font
fontFamily
fontSize
fontSizeAdjust
fontStretch
fontStyle
fontVariant
fontWeight
height
left
letterSpacing
lineHeight
listStyle
listStyleImage
listStylePosition
listStyleType
margin
marginBottom
marginLeft
marginRight
marginTop
markerOffset
marks
maxHeight
maxWidth
minHeight
minWidth
orphans
outline
outlineColor
outlineStyle
outlineWidth
overflow
padding
paddingBottom
paddingLeft
paddingRight
paddingTop
page
pageBreakAfter
pageBreakBefore
pageBreakInside
position
quotes
right
size
tableLayout
textAlign
textDecoration
textIndent
textShadow
textTransform
top
unicodeBidi
verticalAlign
visibility
whiteSpace
widows
width
wordSpacing
zIndex
```

Text ۳٫۳۶

(DOM سطح ۱)

قطعه‌ای از متن در یک سند.
مشتق شده از: Node

۳٫۳۶٫۱ شرح

شیء متن معرف قطعه‌ای از متن ساده بدون نشانه‌گذاری در یک درخت سند DOM است. البته، این را نباید با عنصر ورودی متن یک‌سطری در HTML، که با شیء Input نشان داده می‌شود، اشتباه کنید.

۳٫۳۶٫۲ خصلت‌ها

- **data**
رشته‌ی متنی موجود در این گره.
- **length**
تعداد نویسه‌های موجود در این گره. فقط-خواندنی.

۳٫۳۶٫۳ روش‌ها

- **appendData(text)**
متن داده شده را به آخر متن این گره اضافه می‌کند، و چیزی بر نمی‌گرداند.
- **deleteData(offset, count)**
متن این گره را از نویسه‌ی شماره‌ی offset به تعداد count نویسه حذف می‌کند. چیزی بر نمی‌گرداند.
- **insertData(offset, text)**
متن داده شده را در موقعیت نویسه‌ی offset به متن این گره اضافه می‌کند. چیزی بر نمی‌گرداند.
- **replaceData(offset, count, text)**
متن این گره را از نویسه‌ی شماره‌ی offset به تعداد count نویسه با متن داده شده جایگزین می‌کند. چیزی بر نمی‌گرداند.
- **splitText(offset)**
این گره متنی را در نویسه‌ی موقعیت offset به دو گره تبدیل می‌کند،

و گره جدید را بعد از گره اصلی به سند اضافه می‌کند، و آن را بر می‌گرداند.

- **substringData(offset, count)**

رشته‌ای متشکل از count نویسه، از نویسه‌ی شماره‌ی offset به بعد، بر می‌گرداند.

۳٫۳۶٫۴ ارجاع

Node.normalize().

۳٫۳۷ Textarea

(جاوا اسکریپت سمت مشتری ۱٫۰)
ورودی متن چندسطری.
مشتق شده از: Element

۳٫۳۷٫۱ دستور

```
form.elements[i]
form.elements[name]
form.name
```

۳٫۳۷٫۲ شرح

شیء Textarea (ناحیه‌ی متنی) خیلی شبیه شیء Input (ورودی) است.

۳٫۳۷٫۳ خصلت‌ها

شیء Textarea خصلت‌هایی را برای هر یک از صفت‌های برگه‌ی <textarea> در HTML، از قبیل cols، defaultValue، disabled، name، readOnly، و rows تعریف می‌کند. علاوه بر این، خصلت‌های زیر را هم تعریف می‌کند:

- **form**

شیء فرمی که حاوی این شیء Textarea است. فقط-خواندنی.

- **type**

یک خصلت فقط-خواندنی که نوع عنصر را مشخص می‌کند؛ در مورد

شیء Textarea، مقدار آن همواره "textarea" است.

- **value**

رشته‌ای خواندنی/نوشتنی که متن موجود داخل این ناحیه‌ی متنی را مشخص می‌کند. مقدار اولیه‌ی آن، همان مقدار خصلت defaultValue است.

۳/۳۷/۴ روش‌ها

- **blur()**

کانون صفحه‌کلید را تحویل داده و چیزی بر نمی‌گرداند.

- **focus()**

کانون صفحه‌کلید را تحویل گرفته و چیزی بر نمی‌گرداند.

- **select()**

کل محتوای ناحیه‌ی متنی را انتخاب می‌کند. چیزی بر نمی‌گرداند.

۳/۳۷/۵ رویدادپردازها

- **onblur**

وقتی کانون صفحه‌کلید از دست می‌رود، فرا خوانده می‌شود.

- **onchange**

وقتی کاربر محتوای ناحیه‌ی متنی را تغییر داده، و کانون صفحه‌کلید را به جای دیگری منتقل می‌کند، فر اخوانده می‌شود. این رویدادپرداز فقط وقتی فرا خوانده می‌شود که کاربر کار ویرایش متن داخل ناحیه‌ی متنی را به پایان ببرد.

- **onfocus**

وقتی کانون صفحه‌کلید حاصل می‌شود، فرا خوانده می‌شود.

۳/۳۷/۶ ارجاع

Element، Form، Input.

Window ۳٫۳۸

(جاوا اسکریپت سمت مشتری ۱٫۰)
پنجره یا کادر مرورگر.

۳٫۳۸٫۱ دستور

```
self
window
window.frames[i]
```

۳٫۳۸٫۲ خصلت‌ها

شیء پنجره خصلت‌های زیر را تعریف می‌کند. خصلت‌های انتقال‌ناپذیر مختص مرورگر بعد از این لیست به صورت جداگانه فهرست شده‌اند. دقت کنید که شیء پنجره، در جاوا اسکریپت سمت مشتری، شیء سراسری محسوب می‌شود؛ بنا بر این، شیء پنجره واجد خصلت‌هایی که برای شیء سراسری ذکر شده‌اند، نیز هست.

- **closed**

یک مقدار بولی فقط-خواندنی که مشخص می‌کند پنجره بسته شده است یا نه.

- **defaultStatus**

رشته‌ای خواندنی/نوشتنی که پیغامی را که در صورت نبودن پیغام دیگر، در سطر وضعیت مرورگر نمایش داده می‌شود، تعیین می‌کند.

- **document**

اشاره‌ای فقط-خواندنی به شیء سند موجود در این پنجره یا کادر. رک. Document.

- **frames[]**

آرایه‌ای از اشیای پنجره، که معرف کادرهای موجود در داخل پنجره است. دقت کنید که این کادرها هر کدام ممکن است مشتمل بر کادرهایی در درون خود باشند، که از طریق آرایه‌ی frames[] هر کدام از آنها قابل دستیابی است.

- **history**

اشاره‌ای فقط-خواندنی به شیء سابقه‌ی مربوط به این پنجره یا کادر. رک.

History.

- **length**

تعداد کادرهای موجود در داخل این پنجره یا کادر را نشان می‌دهد. معادل frames.length است.

- **location**

شیء مکان مربوط به این پنجره یا کادر. رک. Location. این خصلت رفتاری ویژه دارد: اگر رشته‌ی یک نشانی را به آن بدهید، مرورگر آن نشانی را می‌خواند و نمایش می‌دهد.

- **name**

رشته‌ای که حاوی نام پنجره یا کادر است. نام از طریق روش Window.open() و یا با صفت name در برگه‌ی <frame> مشخص می‌شود. در جاوا اسکریپت ۱٫۰ فقط-خواندنی؛ در جاوا اسکریپت ۱٫۱ خواندنی/نوشتنی.

- **navigator**

اشاره‌ای فقط-خواندنی به شیء مرورگر، که اطلاعات مربوط به روایت و پیکربندی مرورگر شبکه را نشان می‌دهد. رک. Navigator.

- **opener**

اشاره‌ای خواندنی/نوشتنی به شیء پنجره‌ای که این پنجره را باز کرده است. جاوا اسکریپت ۱٫۱.

- **parent**

اشاره‌ای فقط-خواندنی به شیء پنجره‌ای که حاوی این پنجره یا کادر است. اگر این پنجره یک پنجره‌ی سطح بالا باشد، parent به خود آن اشاره می‌کند.

- **screen**

اشاره‌ای فقط-خواندنی به شیء صفحه‌ی نمایش که اطلاعاتی را در باره‌ی صفحه‌ی نمایشی که مرورگر در آن نمایش داده می‌شود، نشان می‌دهد. رک. Screen. جاوا اسکریپت ۱٫۲.

- **self**

اشاره‌ای فقط-خواندنی به خود این پنجره. این مترادف با خصلت window است.

- **status**

رشته‌ای خواندنی/نوشتنی که می‌توان برای نمایش پیغامی گذرا در سطر وضعیت مرورگر از آن استفاده کرد.

- **top**

اشاره‌ای فقط-خواندنی به پنجره‌ی سطح بالایی که حاوی این پنجره است. اگر این پنجره یک پنجره‌ی سطح بالا باشد، top به خود آن اشاره می‌کند.

- **window**

خصلت window مترادف با خصلت self است؛ حاوی اشاره‌ای به این پنجره است.

۳٫۳۸٫۳ خصلت‌های نت‌اسکیپ ۴

- **innerHeight, innerWidth**

خصلت‌هایی خواندنی/نوشتنی که بلندا و پهنای ناحیه‌ی نمایش سند در این پنجره را بر حسب پیکسل مشخص می‌کنند. این ابعاد، مشتمل بر بلندای نوار منو و ابزار و امثال آن نیست.

- **outerHeight, outerWidth**

خصلت‌هایی خواندنی/نوشتنی که کل بلندا و پهنای این پنجره را مشخص می‌کنند. این ابعاد، مشتمل بر بلندای نوار منو و ابزار و امثال آن است.

- **pageXOffset, pageYOffset**

اعداد صحیح فقط-خواندنی که تعداد پیکسل‌های نَوَردیده شده‌ی سند به راست (pageXOffset) و پایین (pageYOffset) را نشان می‌دهند.

- **screenX, screenY**

اعداد صحیح فقط-خواندنی که مختصات X و Y گوشه‌ی بالا و چپ پنجره را نسبت به صفحه‌ی نمایش نشان می‌دهند. اگر این پنجره یک کادر باشد، این خصلت‌ها مختصات پنجره‌ی سطح بالایی را که این کادر در آن

واقع شده است، نشان می‌دهند.

۳٫۳۸٫۴ خصلت‌های اینترنت اکسپلورر ۴

- **clientInformation**

این خصلت مختص اینترنت اکسپلورر، مترادف خصلت navigator است، و به شیء مرورگر اشاره دارد.

- **event**

این خصلت یک شیء رویداد است که جزئیات جدیدترین رویدادی را که در این پنجره رخ داده است، نشان می‌دهد. در مدل رویداد اینترنت اکسپلورر، شیء رویداد به عنوان آوند به رویدادپرداز داده نمی‌شود، بلکه به این خصلت اختصاص داده می‌شود.

۳٫۳۸٫۵ روش‌ها

شیء پنجره روش‌های انتقال‌پذیر زیر را دارا است. در ضمن، از آنجا که شیء پنجره در جاوا اسکریپت سمت مشتری، شیء سراسری به شمار می‌رود، لذا روش‌های ذکر شده برای شیء سراسری را هم دارا است.

- **alert(message)**

رشته‌ی message را در یک پنجره‌ی گفتگو نشان می‌دهد. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۰.

- **blur()**

کانون صفحه‌کلید را تحویل می‌دهد. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱.

- **clearInterval(intervalId)**

اجرای تکراری مشخص شده با شناسه‌ی intervalId را لغو می‌کند. رک. setInterval(). چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **clearTimeout(timeoutId)**

زمان انتظار مشخص شده با شناسه‌ی timeoutId را لغو می‌کند. رک. setTimeout(). چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۰.

- **close()**

پنجره را می‌بندد و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۰.

- **confirm(question)**

متن question را در یک پنجره‌ی گفتگو نشان می‌دهد، و منتظر یک جواب آری یا نه می‌شود. اگر کاربر دکمه‌ی OK را بزند، true بر می‌گرداند، و اگر دکمه‌ی Cancel را بزند، false بر می‌گرداند جاوا اسکریپت ۱٫۰.

- **focus()**

کانون صفحه‌کلید را تقاضا می‌کند؛ در ضمن، در اکثر سکوها این روش باعث می‌شود که پنجره‌ی مورد نظر به جلوی پنجره‌های دیگر بیاید. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱.

- **getComputedStyle(elt)**

یک شیء شیوه‌ی فقط-خواندنی بر می‌گرداند که حاوی تمام شیوه‌های CSS (و نه فقط شیوه‌های مستقیم) مربوط به عنصر elt است. صفت‌های موقعیت، مانند left، top، و width در اینجا همیشه بر حسب پیکسل بر گردانده می‌شود. DOM سطح ۲.

- **moveBy(dx, dy)**

پنجره را از موقعیت فعلی به میزان مشخص شده جا به جا می‌کند. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **moveTo(x, y)**

پنجره را به موقعیت مشخص شده منتقل می‌کند. چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **open(url, name, features)**

نشانی url را در پنجره‌ی با نام مشخص شده نشان می‌دهد. اگر آوند نام حذف شود و یا پنجره‌ای با آن نام وجود نداشته باشد، پنجره‌ی جدیدی ایجاد می‌شود. آوند اختیاری features رشته‌ای است که اندازه و تزیینات پنجره‌ی جدید را به صورت لیستی جدا شده با ویرگول معین می‌کند. نام ویژگی‌هایی که در عموماً همه‌ی سکوها پشتیبانی می‌شوند، عبارت‌اند از: width=pixels، height=pixels، location،

menubar، resizable، status، و toolbar. در اینترنت اکسپلورر، موقعیت پنجره به صورت left=x و top=y مشخص می‌شود. در نت‌اسکیپ، به جای آن از screenX=x و screenY=y استفاده کنید. شیء پنجره‌ی موجود یا جدید را بر می‌گرداند. جاوا اسکریپت ۱٫۰.

- **print()**

همانند کلیک کردن دکمه‌ی «چاپ» پنجره‌ی مرورگر عمل می‌کند، و چیزی بر نمی‌گرداند. نت‌اسکیپ ۴؛ اینترنت اکسپلورر ۵.

- **prompt(message, default)**

پیغام message را در یک پنجره‌ی گفتگو نشان می‌دهد، و منتظر می‌شود که کاربر متن پاسخ را وارد کند. مقدار اختیاری default را به عنوان پاسخ پیش‌فرض نشان می‌دهد. رشته‌ی وارد شده توسط کاربر را بر می‌گرداند، و یا اگر کاربر رشته‌ای وارد نکند، یک رشته‌ی خالی بر می‌گرداند، و در صورتی که کاربر دکمه‌ی «انصراف» را بزند، null بر می‌گرداند. جاوا اسکریپت ۱٫۰.

- **resizeBy(dw, dh)**

پنجره را به میزان مشخص شده تغییر اندازه می‌دهد، و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **resizeTo(width, height)**

پنجره را به اندازه‌ی مشخص شده می‌رساند، و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **scroll(x, y)**

پنجره را به مختصات مشخص شده اسکرول می‌کند، و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۱. در جاوا اسکریپت ۱٫۲ به نفع scrollTo() منسوخ شده است.

- **scrollBy(dx, dy)**

پنجره را به میزان مشخص شده اسکرول می‌کند، و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱٫۲.

- **scrollTo(x, y)**

پنجره را به موقعیت مشخص شده اسکرول می‌کند، و چیزی بر نمی‌گرداند. جاوا اسکریپت ۱/۲.

- **setInterval(code, interval, args...)**

رشته‌ی متن جاوا اسکریپت code را هر interval میلی‌ثانیه ارزیابی می‌کند. در نت‌اسکیپ ۴ و اینترنت اکسپلورر ۵، code می‌تواند به جای یک رشته، اشاره‌ای به یک تابع باشد. در آن حالت، تابع مشخص شده هر interval میلی‌ثانیه یک بار اجرا می‌شود. در نت‌اسکیپ، هر گونه آوندی بعد از interval وجود داشته باشد، در موقع فراخوانی به عنوان آوند به تابع مذکور داده می‌شود، ولی این ویژگی به وسیله‌ی اینترنت اکسپلورر پشتیبانی نمی‌شود. یک شناسه بر می‌گرداند که می‌توان برای لغو اجرای تکراری، آن را به عنوان آوند به روش clearInterval() داد. جاوا اسکریپت ۱/۲.

- **setTimeout(code, delay)**

رشته‌ی متن جاوا اسکریپت code را پس از سپری شدن delay میلی‌ثانیه ارزیابی می‌کند. در نت‌اسکیپ ۴ و اینترنت اکسپلورر ۵، code می‌تواند به جای یک رشته، اشاره‌ای به یک تابع باشد؛ به بحث مربوط به setInterval() مراجعه کنید. یک شناسه بر می‌گرداند که می‌توان برای لغو اجرای تعلیقی، آن را به عنوان آوند به روش clearTimeout() داد. دقت کنید که این روش بلافاصله بر می‌گردد، و برای بر گشتن منتظر سپری شدن زمان delay نمی‌شود. جاوا اسکریپت ۱/۰.

۳/۳۸/۶ رویدادپردازها

رویدادپردازهای شیء پنجره به وسیله‌ی صفت‌های برگه‌ی <body> سند تعریف می‌شوند.

- **onblur**

زمانی که پنجره کانون صفحه‌کلید را از دست می‌دهد، فرا خوانده می‌شود.

- **onerror**

وقتی یک خطای جاوا اسکریپت بروز می‌کند، فرا خوانده می‌شود. این رویدادپرداز ویژه‌ای است که با سه آوند فرا خوانده می‌شود: پیغام خطا، نشانی سند حاوی خطا، و شماره‌ی سطر خطا، در صورت موجود بودن.

- **onfocus**

زمانی که پنجره کانون صفحه‌کلید را به دست می‌آورد، فرا خوانده می‌شود.

- **onload**

وقتی سند (یا مجموعه‌ی کادر) به طور کامل خوانده شد، فرا خوانده می‌شود.

- **onresize**

وقتی اندازه‌ی پنجره تغییر می‌کند، فرا خوانده می‌شود.

- **onunload**

وقتی مرورگر سند فعلی را ترک می‌کند، فرا خوانده می‌شود.

۳/۳۸/۷ ارجاع

.Document

۴ واژه‌نامه

۴/۱ فارسی به انگلیسی

ابرمتن	hypertext
اجرا	execute
اختصاص	assignment
آرایش	format
آرایه	array
آرایه‌ی ارتباطی	associated array
استثنا	exception
اسکریپت	script
اسکریپت‌نویسی	scripting
اشاره	reference
اشاره‌گر	pointer
اشکال	bug
اعتبارسنجی	validation
اعلام	declaration
آغازش	initialization
افزایش	increment
امنیت	security
انتخاب	select
انتشار	propagation
انتقال‌پذیر	portable
انتقال‌ناپذیر	non-portable
انطباق	match
آوند	argument
بار شدن	load
بدنه	body
بدوی	primitive
برادر/خواهر	sibling
برچسب	label
برشماری	enumeration
برشمردنی	enumerable
برگردان	reset
برگه	tag

برنامک	applet
بند	clause
بولی	Boolean
بی‌نهایت	infinity
پاسخ	response
پررنگ	bold
پرسجو	query
پرونده	file
پسوند	extension
پشته	stack
پنهان	hidden
پویا	dynamic
پیاده‌سازی	implementation
پیش‌خوانی	preload
پیش‌فرض	default
پیکربندی	configuration
پیکسل	pixel
پیوند	link
تابع	function
تجزیه	parse
تجزیه‌گر	parser
تحویل	submit
تدوین	compile
تدوین‌گر	compiler
تراجع	recursion
تراجعی	recursive
ترتیب z	z-order
تزیینات	decorations
تصویر	image
تعلیقی	suspended
تعیین‌نامه	specification
تغییرگر	modifier

تغییرناپذیر	immutable
تفسیر	interpret
تفسیرگر	interpreter
تقاضا	request
توضیح	comment
ثابت	constant
ثبت کردن	register
جازن	plug-in
جاوا اسکریپت سمت مشتری	client-side JavaScript
جاوا اسکریپت هسته	core JavaScript
جدا کننده‌ی هزارگان	thousands separator
جریان	stream
جعبه‌رنگ	palette
جهش	tab
جهشی	popup
چک‌باکس	checkbox
حافظه‌ی نهان	cache
حلقه	loop
خصلت	property
خصوصی	private
خطا	error
خواباندن	embed
خواهر/برادر	sibling
در نوردیدن	scrolling
درایه	entry
درج کردن	insert
درخت	tree
دستور	command
دستور	syntax
دستورالعمل	instruction
دکمه	button
دکمه‌ی رادیویی	radio button
دوکلیک	double-click
رابط	interface
رابطه‌ای	relational
رشته	string
رشته	string
رقم	digit
رمزگذاری	encoding
رواندازی	rollover
روایت	version
روش	method

رویداد	event
رویدادپرداز	event handler
رویدادشنو	event listener
زمان اجرا	runtime
زمان‌سنج	timer
زیررشته	substring
زیرریز	underflow
زیرکلاس	subclass
سابقه	history
سازنده	constructor
سراسری	global
سرریز	overflow
سرسطر	carriage return
سرفصل	header
سرمشق	prototype
سطح بالا	top-level
سطر جدید	line feed
سطر وضعیت	status line
سکو	platform
سلسله‌ی گریز	escape sequence
سند	document
سیستم عامل	operating system
شانزدهگانی	hexadecimal
شناسه	ID
شناسه	identifier
شیء	object
شیءگرا	object-oriented
شیوه	style
صفت	attribute
صفحه‌ی جدید	form feed
صفحه‌ی نمایش	screen
ظهور	visibility
عامل کاربر	user agent
عبارت	expression
عبارت مرتب	regular expression
عدد صحیح	integer
عدم یکسانی	non-identity
عملگر	operator
عملوند	operand
عمومی	public
عنصر	element
عنوان	caption
فاصله	hyphen

فراخوانی	calling
فراگذاری	upload
فرزند	child
فروگذاری	download
فضا	space
فضای سفید	white space
فضای وارون	backspace
فقط-خواندنی	read-only
قرینه	context
قلاب	curly brace
قلمرو	domain
کادر	frame
کانون	focus
کاهش	decrement
کجخط	slash
کجخط وارون	backslash
کران	bound
کلیدواژه	keyword
کلیک	click
گرفتن (رویداد)	capture
گره	node
گزینه	option
گنجایه	container
الگو	pattern
لایه	layer
لفافه	wrapping
لفظی	literal
لقب	nickname
لنگر	anchor
مبدأ	source
مبنا	radix
متغیر	variable
متغیر	variable
متن برنامه	source code
متن ساده	plain text
متناهی	finite
مجموعه‌ی کادر	frame set
محلی	local
محلی شده	localized
مختصه	coordinate
مدل شیء سند	document object model

مرئی	visible
مرورگر	browser
مرورگر	navigator
مسیر	path
مشتری	client
مکان	location
ممیز اعشاری	decimal point
ممیز شناور	floating-point
منسوخ	deprecated
منطقه‌ی زمانی	time zone
موروثی	inherited
موشواره	mouse
موقعیت	position
میانبر	shortcut
ناحیه‌ی برش	clipping region
ناحیه‌ی متنی	textarea
نامرئی	hidden
ناوبری	navigation
نسخه	copy
نسخه‌برداری	copy
نشانگر	cursor
نشانه‌گذاری	markup
نشانی	URL
نقل‌قول	quote
نما	exponent
نماد	symbol
نمایه	index
نمایه‌دهی	indexing
نمایی	exponential
نمونه	instance
نوار نشانی	address bar
نوارپیما	scrollbar
نویسگان	character set
نویسگان	charset
نویسه	character
والد	parent
ورودی	input
وضوح	resolution
ویراست	version
ویرگول‌نقطه	semicolon
یکسانی	identity

۴٫۲ انگلیسی به فارسی

address bar	نوار نشانی
anchor	لنگر
applet	برنامک
argument	آوند
array	آرایه
assignment	اختصاص
associated array	آرایه‌ی ارتباطی
attribute	صفت
backslash	کجخط وارون
backspace	فضای وارون
body	بدنه
bold	پررنگ
Boolean	بولی
bound	کران
browser	مرورگر
bug	اشکال
button	دکمه
cache	حافظه‌ی نهان
calling	فراخوانی
caption	عنوان
capture	گرفتن (رویداد)
carriage return	سرسطر
character	نویسه
character set	نویسگان
charset	نویسگان
checkbox	چک‌باکس
child	فرزند
clause	بند
click	کلیک
client	مشتری
client-side JavaScript	جاوا اسکریپت سمت مشتری
clipping region	ناحیه‌ی برش
command	دستور
comment	توضیح
compile	تدوین
compiler	تدوین‌گر
configuration	پیکربندی
constant	ثابت
constructor	سازنده
container	گنجایه
context	قرینه
coordinate	مختصه
copy	نسخه
copy	نسخه‌برداری
core JavaScript	جاوا اسکریپت هسته
curly brace	قلاب
cursor	نشانگر
decimal point	ممیز اعشاری
declaration	اعلام
decorations	تزیینات
decrement	کاهش
default	پیش‌فرض
deprecated	منسوخ
digit	رقم
document	سند
document object model	مدل شیء سند
domain	قلمرو
double-click	دوکلیک
download	فروگذاری
dynamic	پویا
element	عنصر
embed	خواباندن
encoding	رمزگذاری
entry	درایه
enumerable	برشمردنی
enumeration	برشماری
error	خطا
escape sequence	سلسله‌ی گریز
event	رویداد
event handler	رویدادپرداز
event listener	رویدادشنو
exception	استثنا
execute	اجرا
exponent	نما
exponential	نمایی
expression	عبارت
extension	پسوند
file	پرونده

finite	متناهی
floating-point	ممیز شناور
focus	کانون
form feed	صفحه‌ی جدید
format	آرایش
frame	کادر
frame set	مجموعه‌ی کادر
function	تابع
global	سراسری
header	سرفصل
hexadecimal	شانزدهگانی
hidden	پنهان
hidden	نامرئی
history	سابقه
hypertext	ابرمتن
hyphen	فاصله
ID	شناسه
identifier	شناسه
identity	یکسانی
image	تصویر
immutable	تغییرناپذیر
implementation	پیاده‌سازی
increment	افزایش
index	نمایه
indexing	نمایه‌دهی
infinity	بی‌نهایت
inherited	موروثی
initialization	آغازش
input	ورودی
insert	درج کردن
instance	نمونه
instruction	دستورالعمل
integer	عدد صحیح
interface	رابط
interpret	تفسیر
interpreter	تفسیرگر
keyword	کلیدواژه
label	برچسب
layer	لایه
line feed	سطر جدید
link	پیوند
literal	لفظی
load	بار شدن
local	محلی
localized	محلی شده
location	مکان
loop	حلقه
markup	نشانه‌گذاری
match	انطباق
method	روش
modifier	تغییرگر
mouse	موشواره
navigation	ناوبری
navigator	مرورگر
nickname	لقب
node	گره
non-identity	عدم یکسانی
non-portable	انتقال‌ناپذیر
object	شیء
object-oriented	شیءگرا
operand	عملوند
operating system	سیستم عامل
operator	عملگر
option	گزینه
overflow	سرریز
palette	جعبه‌رنگ
parent	والد
parse	تجزیه
parser	تجزیه‌گر
path	مسیر
pattern	الگو
pixel	پیکسل
plain text	متن ساده
platform	سکو
plug-in	جازن
pointer	اشاره‌گر
popup	جهشی
portable	انتقال‌پذیر
position	موقعیت
preload	پیش‌خوانی
primitive	بدوی
private	خصوصی
propagation	انتشار
property	خصلت
prototype	سرمشق
public	عمومی

پرسجو	query
نقل‌قول	quote
دکمه‌ی رادیویی	radio button
مبنا	radix
فقط-خواندنی	read-only
تراجع	recursion
تراجعی	recursive
اشاره	reference
ثبت کردن	register
عبارت مرتب	regular expression
رابطه‌ای	relational
تقاضا	request
برگردان	reset
وضوح	resolution
پاسخ	response
رواندازی	rollover
زمان اجرا	runtime
صفحه‌ی نمایش	screen
اسکریپت	script
اسکریپت‌نویسی	scripting
نوارپیما	scrollbar
در نوردیدن	scrolling
امنیت	security
انتخاب	select
ویرگول‌نقطه	semicolon
میانبر	shortcut
برادر/خواهر	sibling
خواهر/برادر	sibling
کجخط	slash
مبدأ	source
متن برنامه	source code
فضا	space
تعیین‌نامه	specification
پشته	stack
سطر وضعیت	status line
جریان	stream
رشته	string
رشته	string
شیوه	style
زیرکلاس	subclass
تحویل	submit
زیررشته	substring
تعلیقی	suspended
نماد	symbol
دستور	syntax
جهش	tab
برگه	tag
ناحیه‌ی متنی	textarea
جدا کننده‌ی هزارگان	thousands separator
منطقه‌ی زمانی	time zone
زمان‌سنج	timer
سطح بالا	top-level
درخت	tree
زیرریز	underflow
فراگذاری	upload
نشانی	URL
عامل کاربر	user agent
اعتبارسنجی	validation
متغیر	variable
متغیر	variable
روایت	version
ویراست	version
ظهور	visibility
مرئی	visible
فضای سفید	white space
لفافه	wrapping
ترتیب z	z-order