

مرجع برنامه‌نویسی

جاوا اسکریپت

دیوید فلانگان

ترجمه‌ی دکتر قاسم کیانی مقدم



مرجع برنامه‌نویسی جاوا اسکریپت

به نام خداوند جان و خرد

مرجع برنامه نویسی جاوا اسکریپت

ویراست سوم

دیوید فلاناگان

ترجمه‌ی دکتر قاسم کیانی مقدم

۱۳۹۲

سرشناسه:	فلناگن، دیوید	Flanagan, David
عنوان و نام پدیدآور:	مرجع برنامه‌نویسی جاوا اسکریپت / دیوید فلاناگان:	
ترجمه‌ی قاسم کیانی مقدم		
مشخصات نشر:	سبزوار: قاسم کیانی مقدم، ۱۳۹۲	
مشخصات ظاهری:	۳۰۰ ص: مصور، جدول	
شابک:	۹۷۸-۶۰۰-۰۴-۰۴۳۲-۱	
وضعیت فهرست‌نویسی:	فیپا	
یادداشت:	عنوان اصلی:	JavaScript Pocket Reference, 3rd ed., 2012
موضوع:	جاوا اسکریپت (زبان برنامه‌نویسی کامپیوتر)	
شناسه‌ی افزوده:	کیانی مقدم، قاسم، -۱۳۴۹، مترجم	
رده‌بندی کنگره:	۱۳۹۲ ف۸ ج۲/۷۳/۷۴ QAV۶	
رده‌بندی دیوپی:	۰۰۵,۲۷۶۲	
شماره‌ی کتاب‌شناسی ملی:	۳۳۶۹۱۰۶	



مرجع برنامه‌نویسی جاوا اسکریپت

نوشته‌ی دیوید فلاناگان

ترجمه‌ی دکتر قاسم کیانی مقدم

طراحی، صفحه‌آرایی، و انتشار توسط مترجم

نوبت چاپ: اول، ۱۳۹۲

شمارگان: ۱,۰۰۰ جلد

قیمت: ۱۸۰,۰۰۰ ریال

چاپ و صحافی: دقت

شابک: ۹۷۸-۶۰۰-۰۴-۰۴۳۲-۱ ISBN: 978-600-04-0432-1

© ۱۳۹۲. کلیه‌ی حقوق برای مترجم محفوظ است.

خراسان رضوی، سبزوار، مجتمع پزشکی امین، طبقه‌ی سوم

ghasemkiani.ir

تلفن: ۰۹۱۵۵۷۲۳۵۰۴

فهرست

۹	پیش‌گفتار
۱۳	فصل ۱: ساختار لغوی
۱۳	توضیحات
۱۴	شناسه‌ها و واژه‌های ذخیره
۱۵	ویرگول نقطه‌ی اختیاری
۱۷	فصل ۲: انواع، مقادیر، و متغیرها
۱۸	اعداد
۲۱	متن
۲۴	مقادیر بولی
۲۶	undefined و null
۲۷	شیء سراسری
۲۸	تبدیل انواع
۳۳	اعلام متغیر
۳۷	فصل ۳: عبارت‌ها و عملگرها
۳۸	عبارت‌ها
۴۲	عملگرها
۴۶	عملگرهای حسابی
۵۰	عملگرهای رابطه‌ای
۵۴	عبارت‌های منطقی

۵۶	عبارت‌های تخصیص
۵۷	عبارت‌های ارزیابی
۵۹	عملگرهای متفرقه
۶۳	فصل ۴: دستورالعمل‌ها
۶۵	دستورالعمل‌های عبارت
۶۶	دستورالعمل‌های مرکب و خالی
۶۷	دستورالعمل‌های اعلام
۷۰	دستورالعمل‌های شرطی
۷۴	حلقه‌ها
۷۹	پرش‌ها
۸۶	دستورالعمل‌های متفرقه
۹۱	فصل ۵: اشیا
۹۲	ایجاد کردن اشیا
۹۶	خصلت‌ها
۱۰۱	سریالیزه کردن خصلت‌ها و اشیا
۱۰۲	مقدارگیرها و مقدارگذارهای خصلت
۱۰۴	صفات خصلت‌ها
۱۰۸	صفات اشیا
۱۱۱	فصل ۶: آرایه‌ها
۱۱۲	ایجاد کردن آرایه‌ها
۱۱۳	عناصر و طول آرایه
۱۱۴	تکرار روی عناصر آرایه
۱۱۵	آرایه‌های چندبعدی
۱۱۶	متدهای آرایه
۱۲۱	متدهای آرایه در اکما اسکریپت ۵
۱۲۵	نوع آرایه
۱۲۶	اشیای شبه‌آرایه
۱۲۷	رشته‌ها به عنوان آرایه

۱۲۹	فصل ۷: توابع
۱۳۰	تعریف کردن تابع
۱۳۳	فرا خواندن توابع
۱۴۰	آوندها و پارامترهای تابع
۱۴۲	توابع به عنوان فضای نام
۱۴۳	بستارها
۱۴۸	خصلت‌ها، متدها، و سازنده‌ی تابع
۱۵۳	فصل ۸: کلاس‌ها
۱۵۴	کلاس‌ها و پیش‌نمونه‌ها
۱۵۷	کلاس‌ها و سازنده‌ها
۱۶۲	کلاس‌های سَبَکِ جاوا در جاوا اسکریپت
۱۶۴	کلاس‌های تغییرناپذیر
۱۶۶	زیرکلاس‌ها
۱۶۸	افزودن به کلاس‌ها
۱۷۱	فصل ۹: عبارت‌های قاعده‌مند
۱۷۱	توصیف الگوها با عبارت‌های قاعده‌مند
۱۸۰	انطباق الگوها با عبارت‌های قاعده‌مند
۱۸۵	فصل ۱۰: جاوا اسکریپت سمت مشتری
۱۸۵	جای‌گذاری جاوا اسکریپت در HTML
۱۸۷	برنامه‌نویسی مبتنی بر رویداد
۱۸۸	شیء پنجره
۲۰۱	فصل ۱۱: اسکریپت‌نویسی سند
۲۰۱	مروری بر DOM
۲۰۴	انتخاب عناصر سند
۲۱۱	ساختار و پیمایش سند
۲۱۳	صفت‌ها
۲۱۵	محتوای عنصر

۲۱۷	ایجاد، درج، و حذف گره‌ها
۲۲۰	شیوه‌ی عنصر
۲۲۴	هندسه و اسکرول کردن صفحه
۲۲۹	فصل ۱۲: کار با رویدادها
۲۳۱	انواع رویدادها
۲۳۹	ثبت کردن رویدادپردازها
۲۴۳	فراخوانی رویدادپرداز
۲۴۹	فصل ۱۳: کار با شبکه
۲۴۹	استفاده از XMLHttpRequest
۲۵۸	HTTP از طریق <script>: JSONP
۲۶۲	رویدادهای فرستاده شده از سرور
۲۶۳	وب‌سوکت‌ها
۲۶۵	فصل ۱۴: ذخیره‌سازی سمت مشتری
۲۶۶	localStorage و sessionStorage
۲۷۲	کوکی‌ها
۲۷۹	نمایه
۲۹۳	واژه‌نامه
۲۹۳	انگلیسی به فارسی
۲۹۷	فارسی به انگلیسی

پیش‌گفتار

جاوا اسکریپت زبان برنامه‌نویسی وب است. اکثریت قاطع وب‌سایت‌های امروزی از جاوا اسکریپت استفاده می‌کنند و همه‌ی مرورگرهای مدرن وب—روی رایانه‌های رومیزی، دستگاه‌های بازی، رایانه‌های لوحی، و تلفن‌های هوشمند—دارای تفسیرگر جاوا اسکریپت هستند. بر این اساس، زبان جاوا اسکریپت را می‌توان فراگیرترین زبان برنامه‌نویسی در طول تاریخ دانست. جاوا اسکریپت بخشی از فناوری‌های سه‌گانه‌ای است که هر برنامه‌نویس وب باید یاد بگیرد: HTML برای مشخص کردن محتوای صفحات وب، CSS برای مشخص کردن نحوه‌ی نمایش آن صفحات، و جاوا اسکریپت برای مشخص کردن رفتار آنها. اخیراً با ظهور نود (Node) (<http://nodejs.org>)، جاوا اسکریپت برای سرورهای وب نیز زبان برنامه‌نویسی مهمی شده است.

این کتاب گزیده‌ای از کتاب کامل‌تر راهنمای قاطع جاوا اسکریپت است. هیچیک از مطالب ویراست دوم که اکنون قدیمی شده است، در این کتاب باقی نمانده است. امیدوارم که برخی از خوانندگان که کتاب اصلی را بسیار قطور و وقت‌گیر می‌یابند، بتوانند از این کتاب کوچک‌تر و فشرده‌تر بهره‌ی لازم را برگیرند. این مرجع جیبی از نظر ساختار کلی همانند کتاب اصلی است: فصل‌های ۱ تا ۹ زبان هسته‌ی جاوا اسکریپت را در بر می‌گیرند، و از مباحث بنیادی نحو زبان—انواع، مقادیر، متغیرها، عملگرها، دستورالعمل‌ها—شروع کرده و در ادامه اشیا، آرایه‌ها، توابع، و کلاس‌های جاوا اسکریپت را مورد بررسی قرار می‌دهند. این فصول خود زبان را بررسی می‌کنند و هم برای برنامه‌نویسانی که از جاوا اسکریپت در مرورگرهای وب استفاده می‌کنند و هم برای برنامه‌نویسانی که از نود در سمت سرور استفاده می‌کنند، قابل استفاده هستند.

هر زبان برای اینکه مفید باشد، باید یک بستر یا کتابخانه‌ی استاندارد از توابع برای اجرای کارهایی مانند ورودی و خروجی اساسی داشته باشد. زبان جاوا اسکریپت هسته یک رابط برنامه‌نویسی حداقلی را برای کار با مت، آرایه‌ها، تاریخ، و عبارت‌های قاعده‌مند در اختیار ما می‌گذارد، ولی هیچگونه قابلیت ورودی یا خروجی ندارد. ورودی و خروجی (و نیز ویژگی‌های پیشرفته‌تر، از قبیل شبکه، ذخیره‌سازی، و گرافیک) بر عهده‌ی «محیط میزبان» هستند که جاوا اسکریپت درون آن جاگذاری شده است. شایع‌ترین محیط میزبان یک مرورگر وب است. فصل‌های ۱ تا ۹، رابط برنامه‌نویسی حداقلی توکار زبان را بررسی می‌کنند. فصل‌های ۱۰ تا ۱۴ محیط میزبان مرورگر وب را ارائه می‌کنند و چگونگی استفاده از «جاوا اسکریپت سمت مشتری [کارخواه]» را برای ایجاد صفحات وب پویا و برنامه‌های وب توضیح می‌دهند.

تعداد رابط‌های برنامه‌نویسی جاوا اسکریپت که در مرورگرهای وب پیاده‌سازی شده است، در سال‌های اخیر به طور انفجاری افزایش یافته است، و امکان ارائه‌ی همه‌ی آنها در کتابی با این حجم وجود ندارد. فصل‌های ۱۰ تا ۱۴ مهم‌ترین و بنیادی‌ترین بخش‌های جاوا اسکریپت سمت مشتری را بررسی می‌کنند: پنجره‌ها، سندها، عنصرها، شیوه‌ها، رویدادها، شبکه، و ذخیره‌سازی. وقتی که بر اینها تسلط پیدا کنید، یاد گرفتن رابط‌های برنامه‌نویسی دیگر سمت مشتری کار آسانی خواهد بود و برای این کار خواهید توانست از راهنمای قاطع جاوا اسکریپت استفاده کنید. (با هم از کتاب‌های مرجع جیبی *Canvas* و مرجع جیبی *jQuery* که آنها هم گزیده‌هایی از راهنمای قاطع جاوا اسکریپت هستند).

گرچه محیط برنامه‌نویسی نود روز به روز اهمیت بیشتری پیدا می‌کند، ولی واقعاً در این مرجع جیبی امکان گنجاندن اطلاعات در باره‌ی جاوا اسکریپت سمت سرور وجود ندارد. می‌توانید برای اطلاعات بیشتر به <http://nodejs.org> مراجعه کنید. به علاوه، در این کتاب جایی برای گنجاندن یک بخش مرجع رابط برنامه‌نویسی وجود ندارد. در این مورد هم شما را به راهنمای قاطع جاوا اسکریپت و یا به منابع آنلاین جاوا اسکریپت، مانند شبکه‌ی بسیار عالی برنامه‌نویسان موزیلا در نشانی <http://developer.mozilla.org/> ارجاع می‌دهم.

مثال‌های این کتاب را می‌توان از صفحه‌ی وب کتاب دانلود کرد (غلطنامه‌ی کتاب هم، در صورتی که بعد از انتشار غلطی در آن پیدا شود، در همین نشانی قابل دسترسی خواهد بود):

<http://shop.oreilly.com/product/0636920011460.do>

به طور کلی، شما می‌توانید از مثال‌های این کتاب در برنامه‌ها و مستندات خود استفاده کنید. نیازی نیست که جهت کسب اجازه با ما تماس بگیرید، مگر اینکه بخواهید بخش قابل توجهی از کد را به کار بگیرید. مایه‌ی امتنان خواهد بود، لیکن ضروری نیست، که مثلاً به صورت زیر استناد را ذکر نمایید: «بر گرفته از مرجع برنامه‌نویسی جاوا اسکریپت، ویراست سوم، نوشته‌ی دیوید فلاناگان (انتشارات اورایلی). کپی‌رایت ۲۰۱۲ دیوید فلاناگان، شابک ۳-۳۱۶۸۵-۴۴۹-۱-۹۷۸». اگر احساس می‌کنید که استفاده‌ی شما از کد، خارج از شمول استفاده‌ی منصفانه یا اجازه‌ی داده شده در اینجا است، می‌توانید با ما در نشانی permissions@oreilly.com تماس بگیرید.

نظرات یا پرسش‌های فنی خود را در باره‌ی این کتاب می‌توانید با ایمیل زیر در میان بگذارید:

bookquestions@oreilly.com

این کتاب از طریق سرویس Safari Books Online نیز عرضه می‌شود. برای دسترسی کامل دیجیتال به این کتاب و کتاب‌های دیگر در موضوعات مشابه از انتشارات اورایلی و ناشران دیگر، به نشانی <http://www.safaribooksonline.com/> مراجعه کنید.

مایلم از ویراستار کتاب، سایمون سنت لوران، که مرا بر آن داشت که کتاب راهنمای قاطع جاوا اسکریپت را به صورت حاضر تلخیص کنم، و نیز از کارکنان بخش تولید اورایلی، که همواره کتاب‌های مرا به شکلی مطلوب آماده می‌کنند، تشکر کنم.

ساختار لغوی

برنامه‌های جاوا اسکریپت با استفاده از مجموعه‌نویسه‌ی یونیکد نوشته می‌شوند. یونیکد ابرمجموعه‌ای از مجموعه‌نویسه‌های اسکی و لاتین-۱ است و تقریباً همه‌ی زبان‌های مکتوبی را که در حال حاضر در دنیا استفاده می‌شود، پشتیبانی می‌کند. جاوا اسکریپت به حروف کوچک و بزرگ حساس است. یعنی کلیدواژه‌ها، متغیرها، نام توابع، و سایر شناسه‌ها باید همیشه از نظر کوچک و بزرگ بودن حروف به یک صورت نوشته شوند. مثلاً کلیدواژه‌ی `while` باید به صورت `“while”` تایپ شود، نه به صورت `“While”` یا `“WHILE”`. به همین ترتیب، `online`، `OnLine`، `Online` و `ONLINE` چهار نام متغیر متمایز هستند.

توضیحات

جاوا اسکریپت دو نوع توضیح را پشتیبانی می‌کند. هر متنی که بین `//` و پایان سطر قرار داشته باشد، به عنوان توضیح تلقی می‌شود و جاوا اسکریپت از آن صرف نظر می‌کند. هر متن واقع شده بین نویسه‌های `/*` و `*/` نیز به عنوان توضیح تلقی می‌شود؛ این نوع توضیح می‌تواند چند سطر امتداد داشته باشد، ولی به صورت تودرتو نمی‌تواند باشد. سطرهای کد زیر همگی توضیحات قانونی در جاوا اسکریپت هستند:

```
// This is a single-line comment.  
/* This is also a comment */ // And here is another.  
/*  
 * This is yet another comment.
```

* It has multiple lines.

*/

شناسه‌ها و واژه‌های ذخیره

شناسه به یک نام گفته می‌شود. در جاوا اسکریپت از شناسه‌ها برای نامگذاری متغیرها و توابع و تعیین برچسب برای برخی حلقه‌ها در کد جاوا اسکریپت استفاده می‌شود. شناسه در جاوا اسکریپت باید با یک حرف، زیرخط (_)، یا علامت دلار (\$) شروع شود. نویسه‌های بعدی آن می‌توانند از نوع حروف، ارقام، زیرخط، یا علامت دلار باشند.

جاوا اسکریپت تعدادی از شناسه‌ها را به عنوان کلیدواژه‌های خود زبان ذخیره کرده است. از این واژه‌ها نمی‌توانید به عنوان شناسه در برنامه‌های خود استفاده کنید:

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

علاوه بر این، جاوا اسکریپت برخی دیگر از کلیدواژه‌ها را نیز که در حال حاضر در این زبان استفاده نمی‌شوند، ولی ممکن است در آینده مورد استفاده قرار گیرند، ذخیره کرده است. اکما اسکریپت ۵ واژه‌های زیر را ذخیره کرده است:

class	const	enum	export	extends	import	super
-------	-------	------	--------	---------	--------	-------

به علاوه، واژه‌های زیر که در کد معمولی جاوا اسکریپت، قانونی هستند، در حالت اکید واژه‌های ذخیره محسوب می‌شوند:

implements	let	private	public	yield
interface	package	protected	static	

حالت اکید برای کاربرد شناسه‌های زیر نیز محدودیت قابل می‌شود. اینها کاملاً واژه‌ی ذخیره نیستند، ولی نمی‌توان از آنها به عنوان نام متغیر، تابع، یا پارامتر استفاده کرد.

arguments	eval
-----------	------

اکما اسکریپت ۳ همه‌ی کلیدواژه‌های زبان جاوا را ذخیره کرده بود، و گرچه این وضعیت در اکما اسکریپت ۵ رها شده است، ولی اگر قصد داشته باشید کد خود را در یک پیاده‌سازی اکما اسکریپت ۳ از زبان جاوا اسکریپت اجرا کنید، باز هم باید از همه‌ی این شناسه‌ها اجتناب کنید:

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

ویرگول نقطه‌ی اختیاری

همانند بسیاری از زبان‌های برنامه‌نویسی، جاوا اسکریپت از ویرگول نقطه (;) برای جدا کردن دستورالعمل‌ها (رک. فصل ۴) از یکدیگر استفاده می‌کند. این امر برای روشن ساختن معنای برنامه‌های شما اهمیت دارد: بدون جدا کننده، ممکن است انتهای یک دستورالعمل شما با آغاز دستورالعمل بعدی یا بر عکس اشتباه شود. در جاوا اسکریپت، معمولاً می‌توانید ویرگول نقطه‌ی بین دو دستورالعمل را در صورتی که هر کدام در سطر جداگانه‌ای نوشته شده باشد، حذف کنید. (ویرگول نقطه‌ی آخر برنامه را هم در صورتی که نماد بعد از آن آکلاد بسته } باشد، می‌توانید حذف کنید.) بسیاری از برنامه‌نویسان جاوا اسکریپت (و تمام کد ارائه شده در این کتاب) از ویرگول نقطه برای تصریح پایان دستورالعمل استفاده می‌کنند، حتی در جایی که استفاده‌ی آن الزامی نیست. صرف نظر از اینکه شما از چه سبکی پیروی می‌کنید، جزئیات چندی را باید در باره‌ی ویرگول نقطه‌ی اختیاری در جاوا اسکریپت بدانید. کد زیر را در نظر بگیرید. از آنجا که دو دستورالعمل در سطرهای جداگانه هستند، ویرگول نقطه‌ی اول را می‌توان حذف کرد:

```
a = 3;
```

```
b = 4;
```

اما وقتی که به صورت زیر نوشته شود، ویرگول نقطه‌ی اول الزامی است:

```
a = 3; b = 4;
```


دقت کنید که جاوا اسکریپت هر پایان سطر را به عنوان یک ویرگول نقطه تلقی نمی‌کند: فقط در صورتی پایان سطر را به عنوان ویرگول نقطه محسوب می‌کند که امکان تجزیه‌ی کد بدون ویرگول نقطه وجود نداشته باشد. به طور رسمی‌تر، جاوا اسکریپت پایان سطر را در صورتی به عنوان ویرگول نقطه تلقی می‌کند که بعد از کلیدواژه‌های `break`، `return`، یا `continue`، یا قبل از عملگرهای `++` یا `--` قرار گرفته باشد، یا اینکه نویسه‌ی غیر فضای خالی بعد از آن را نتوان به عنوان ادامه‌ی دستورالعمل فعلی تفسیر کرد.

این قواعد پایان دستورالعمل منجر به برخی موارد غافلگیر کننده می‌شوند. کد زیر مانند دو دستورالعمل به نظر می‌رسد که با یک سطر جدید از هم جدا شده‌اند:

```
var y = x + f
(a+b).toString()
```

ولی پرانتزهای سطر دوم کد را می‌توان به عنوان فراخوانی تابع `f` از سطر اول تفسیر کرد، و جاوا اسکریپت کد را به صورت زیر تفسیر می‌کند:

```
var y = x + f(a+b).toString());
```

انواع، مقادیر، و متغیرها

برنامه‌های کامپیوتری بر روی مقادیر عمل می‌کنند، مانند عدد ۳/۱۴ یا رشته‌ی "Hello World". گونه‌های مختلف مقادیری که در یک زبان برنامه‌نویسی امکان نمایش و کار بر روی آنها وجود دارد، انواع نامیده می‌شوند. وقتی که یک برنامه نیاز دارد که یک مقدار را برای استفاده در آینده حفظ کند، آن مقدار را به یک متغیر تخصیص می‌دهد یا در آن «ذخیره» می‌کند. متغیر یک نام نمادین را برای یک مقدار تعریف می‌کند و امکان می‌دهد که از طریق آن نام به آن مقدار اشاره شود.

در جاوا اسکریپت انواع را می‌توان به دو دسته تقسیم کرد: انواع بدوی و انواع شیء. انواع بدوی جاوا اسکریپت شامل اعداد، رشته‌های متنی (به نام *string*) و انواع درستی بولی (به نام *boolean*) هستند. چند قسمت اول این فصل به تعریف انواع بدوی جاوا اسکریپت اختصاص دارد. (فصل‌های ۵، ۶، و ۷ سه گونه‌ی متفاوت نوع شیء در جاوا اسکریپت را توضیح می‌دهند.)

جاوا اسکریپت آزادانه مقادیر را از یک نوع به نوع دیگر تبدیل می‌کند. مثلاً اگر یک برنامه انتظار یک رشته را داشته باشد و شما یک عدد به آن بدهید، آن عدد را به طور خودکار برایتان تبدیل به یک رشته خواهد کرد. اگر در جایی که یک مقدار بولی مورد انتظار است، از یک مقدار غیربولی استفاده کنید، جاوا اسکریپت تبدیل لازم را انجام خواهد داد. در قسمت «تبدیل انواع» در صفحه‌ی ۲۸، تبدیل انواع در جاوا اسکریپت شرح داده شده است.

متغیرهای جاوا اسکریپت بدون نوع هستند. می‌توانید مقداری را از هر نوع به یک متغیر تخصیص دهید، و باز بعداً مقدار دیگری را از نوع دیگر به همان متغیر تخصیص دهید. متغیرها با کلیدواژه‌ی `var`، اعلام می‌شوند. جاوا اسکریپت از قلمرو

لغوی استفاده می‌کند. متغیرهایی که بیرون از یک تابع اعلام شده‌اند، متغیرهای سراسری هستند، و در همه جای یک برنامه‌ی جاوا اسکریپت قابل دستیابی هستند. متغیرهایی که داخل یک تابع اعلام شده‌اند، دارای قلمرو تابع هستند، و فقط برای کدی که در داخل آن تابع است، مشهود هستند. در قسمت «اعلام متغیر» در صفحه‌ی ۳۳، متغیرها را با تفصیل بیشتری شرح خواهیم داد.

اعداد

بر خلاف بسیاری از زبان‌ها، جاوا اسکریپت تمایزی بین مقادیر صحیح و مقادیر ممیز شناور قایل نمی‌شود. تمام اعداد در جاوا اسکریپت به صورت مقادیر ممیز شناور نمایش داده می‌شوند. جاوا اسکریپت اعداد را با استفاده از فرمت ۶۴ بیتی ممیز شناور که در استاندارد IEEE 754 تعریف شده است، نمایش می‌دهد، که معنایش آن است که می‌تواند اعدادی را به بزرگی 10^{308} و به کوچکی $10^{-324} \times \pm 5$ نمایش دهد.

فرمت اعداد در جاوا اسکریپت این امکان را به شما می‌دهد که تمام اعداد صحیح بین -9007199254740992 (-2^{53}) و 9007199254740992 (2^{53})، شامل خود این دو عدد، را به طور دقیق نمایش دهید. در صورتی که از اعداد صحیح بزرگ‌تر از این مقدار استفاده کنید، ممکن است دقت را در رقم‌های آخر از دست بدهید. با این حال، در نظر داشته باشید که برخی عملیات در جاوا اسکریپت (مانند اندیس‌گذاری آرایه‌ها و عملیات بیتی شرح داده شده در فصل ۳) با اعداد صحیح ۳۲ بیتی انجام می‌شوند.

وقتی که یک عدد به طور مستقیم در یک برنامه‌ی جاوا اسکریپت قرار داده می‌شود، به آن یک مقدار لفظی عددی می‌گویند. جاوا اسکریپت مقادیر لفظی عددی را با چند فرمت پشتیبانی می‌کند. دقت کنید که قبل از هر مقدار لفظی عددی می‌توان علامت منها (-) قرار داد تا عدد منفی به دست آید.

در یک برنامه‌ی جاوا اسکریپت، یک عدد صحیح مبنای 10 به صورت دنباله‌ای از ارقام نوشته می‌شود. به عنوان مثال:

0

1024

علاوه بر مقادیر لفظی عدد صحیح مبنای 10 ، جاوا اسکریپت مقادیر شانزدهگانی (مبنای ۱۶) را هم تشخیص می‌دهد. یک مقدار لفظی شانزدهگانی با

”0x” یا ”0X” شروع می‌شود و به دنبال آن رشته‌ای از ارقام شانزدهگانی قرار می‌گیرد. یک رقم شانزدهگانی یا یکی از رقم‌های 0 تا 9 است و یا یکی از حروف a (یا A) تا f (یا F)، که نشان دهنده‌ی مقادیر ۱۰ تا ۱۵ هستند. مثال‌هایی از مقادیر لفظی صحیح شانزدهگانی عبارت‌اند از:

```
0xff // 15*16 + 15 = 255 (base 10)
0xCAFE911
```

مقادیر لفظی عددی ممیز شناور می‌توانند ممیز اعشاری داشته باشند؛ دستور نوشتن آنها به روش معمول اعداد حقیقی صورت می‌گیرد. یک مقدار حقیقی به صورت بخش صحیح عدد، بعد ممیز اعشاری (نقطه)، و سپس بخش کسری عدد نوشته می‌شود.

مقادیر لفظی ممیز شناور را به صورت نمایی نیز می‌توان نمایش داد، یعنی به صورت یک عدد حقیقی، بعد حرف e (یا E)، بعد در صورت نیاز علامت به اضافه یا منهای، و سپس یک عدد صحیح به عنوان نما. این نماد نشان دهنده‌ی عدد حقیقی ضربدر ۱۰ به توان نما است.

به بیان کوتاه‌تر، نحو آن به صورت زیر است:

```
[digits][.digits][E|e][+|-]digits]
```

به عنوان مثال:

```
3.14
6.02e23 // 6.02 × 1023
1.4738223E-32 // 1.4738223 × 10-32
```

برنامه‌های جاوا اسکریپت با استفاده از عملگرهای حسابی که در این زبان وجود دارد، بر روی اعداد کار می‌کنند. این عملگرها شامل + برای جمع، - برای تفریق، * برای ضرب، / برای تقسیم، و % برای نهشته (باقیمانده‌ی بعد از تقسیم) هستند. جزئیات کامل در مورد اینها و سایر عملگرها در فصل ۳ ارائه شده است.

علاوه بر این عملگرهای پایه‌ی حسابی، جاوا اسکریپت عملیات پیچیده‌تر ریاضی را نیز از طریق مجموعه‌ای از توابع و مقادیر ثابت به عنوان خصلت‌های شیء Math پشتیبانی می‌کند:

```
Math.pow(2,53) // => 9007199254740992: 2 to the power 53
Math.round(.6) // => 1.0: round to the nearest integer
Math.ceil(.6) // => 1.0: round up to an integer
```

```

Math.floor(.6) // => 0.0: round down to an integer
Math.abs(-5) // => 5: absolute value
Math.max(x,y,z) // Return the largest argument
Math.min(x,y,z) // Return the smallest argument
Math.random() // Pseudo-random number 0 <= x < 1.0
Math.PI // π
Math.E // e: The base of the natural logarithm
Math.sqrt(3) // The square root of 3
Math.pow(3,1/3) // The cube root of 3
Math.sin(0) // Trig: also Math.cos, Math.atan, etc.
Math.log(10) // Natural logarithm of 10
Math.log(100)/Math.LN10 // Base 10 logarithm of 100
Math.log(512)/Math.LN2 // Base 2 logarithm of 512
Math.exp(3) // Math.E cubed

```

عملیات حسابی در جاوا اسکریپت در صورت بروز سرریز، زیرریز، یا تقسیم بر صفر، خطا بر نمی‌انگیزد. وقتی که نتیجه‌ی یک عملیات عددی از بزرگ‌ترین عدد قابل نمایش، بزرگ‌تر باشد (سرریز)، حاصل آن یک مقدار بی‌نهایت ویژه است که جاوا اسکریپت آن را به صورت *Infinity* نمایش می‌دهد. به همین ترتیب، وقتی که یک عدد منفی از بزرگ‌ترین عدد منفی قابل نمایش بزرگ‌تر شود، حاصل آن منهای بی‌نهایت است که به صورت *-Infinity* نشان داده می‌شود. مقادیر بی‌نهایت به همان صورتی که انتظار می‌رود، رفتار می‌کنند: جمع کردن، تفریق کردن، ضرب کردن، و تقسیم کردن آنها بر هر چیزی یک مقدار بی‌نهایت به دست می‌دهد (احیاناً با علامت مخالف).

تقسیم بر صفر در جاوا اسکریپت یک خطا محسوب نمی‌شود: این عمل صرفاً حاصل بی‌نهایت یا منهای بی‌نهایت را بر می‌گرداند. لیکن یک استثنا وجود دارد: صفر تقسیم بر صفر مقدار تعریف شده‌ای ندارد، و حاصل این عمل، مقدار ویژه‌ی ناعدد است که به صورت *NaN* نشان داده می‌شود. در مواردی هم که بی‌نهایت را بر بی‌نهایت تقسیم کنید، یا از یک عدد منفی جذر بگیرید، یا عملگرهای حسابی را با عملوندهای غیر عددی که قابل تبدیل به عدد نیستند، به کار ببرید، مقدار *NaN* حاصل می‌شود.

جاوا اسکریپت متغیرهای Infinity و NaN را از قبل برای مثبت بی‌نهایت و مقدار ناعدد تعریف کرده است.

مقدار ناعدد در جاوا اسکریپت ویژگی نامعمولی دارد: این مقدار با هیچ عددی دیگری، حتی با خودش، مساوی نمی‌شود. معنای این مطلب آن است که نمی‌توانید از عبارت $x == NaN$ برای فهمیدن اینکه مقدار متغیر x برابر با NaN است، استفاده کنید. بلکه باید بنویسید $x != NaN$. این عبارت اگر و تنها اگر x مقدار NaN داشته باشد، درست خواهد بود. تابع `isNaN()` نیز مشابه آن است. این تابع در صورتی که آوند آن NaN یا یک مقدار غیر عددی مانند یک رشته یا یک شیء باشد، مقدار `true` بر می‌گرداند. تابع مرتبط `isFinite()` در صورتی که آوند آن عددی غیر از `NaN`، `Infinity`، یا `-Infinity` باشد، `true` بر می‌گرداند. تعداد اعداد حقیقی نامتناهی است، ولی تعداد محدودی از آنها (دقیقاً 10^{62} یا 10^{63}) تعداد اعداد حقیقی (عدد) را می‌توان با فرمت ممیز شناور جاوا اسکریپت نمایش داد. معنای این مطلب آن است که وقتی در جاوا اسکریپت با اعداد حقیقی کار می‌کنید، نمایش عدد در جاوا اسکریپت غالباً تقریبی از عدد واقعی خواهد بود و خطای گرد کردن کوچکی بروز خواهد کرد.

متن

رشته یک دنباله‌ی مرتب تغییرناپذیر از مقادیر ۱۶-بیتی است که هر یک از آنها نشان دهنده‌ی یک نویسه‌ی یونیکد است—رشته‌ها در جاوا اسکریپت یک نوع برای نشان دادن متن هستند. طول یک رشته تعداد مقادیر ۱۶-بیتی موجود در آن است. رشته‌ها (و آرایه‌های) جاوا اسکریپت از صفر شماره‌گذاری می‌شوند: اولین مقدار ۱۶-بیتی در موقعیت ۰ است، دومی در موقعیت ۱، و الی آخر. رشته‌ی خالی رشته‌ای با طول صفر است. جاوا اسکریپت نوع خاصی برای نشان دادن هر یک از عناصر یک رشته ندارد. برای نشان دادن یک مقدار ۱۶-بیتی واحد، به سادگی از رشته‌ای به طول ۱ استفاده کنید.

مقادیر لفظی رشته‌ای

برای اینکه یک رشته را به طور تحت‌اللفظی در یک برنامه‌ی جاوا اسکریپت وارد کنید، کافی است نویسه‌های رشته را داخل یک جفت هم‌هانگ از علامت‌های نقل‌قول یکه یا دوتایی (' ' یا " ") قرار دهید. علامت نقل‌قول دوتایی را می‌توان داخل

رشته‌هایی که با علامت نقل‌قول یک‌ه محدود شده‌اند، قرار داد، و علامت نقل‌قول یک‌ه را می‌توان داخل رشته‌هایی که با علامت نقل‌قول دوتایی محدود شده‌اند، قرار داد. چند مثال از مقادیر لفظی رشته‌ای را ببینید:

```
" // The empty string: it has zero characters
'name="myform"
"Wouldn't you prefer O'Reilly's book?"
"This string\nhas two lines"
"π = 3.14"
```

نویسه‌ی کج‌خط وارون \ در رشته‌های جاوا اسکریپت کارکرد ویژه‌ای دارد. این نویسه به همراه نویسه‌ی بعد از آن، نویسه‌ای را نشان می‌دهد که به خودی خود در داخل رشته قابل نشان دادن نیست. مثلاً \n یک توالی گریز است که نویسه‌ی سطر جدید را نشان می‌دهد.

نمونه‌ی دیگر آن توالی گریز \' است که نشان دهنده‌ی نویسه‌ی نقل‌قول یک‌ه (یا آپوستروف) است. وقتی که می‌خواهید یک آپوستروف را داخل رشته‌ای که با علامت نقل‌قول یک‌ه محدود شده است، قرار دهید، می‌توانید از این توالی گریز استفاده کنید. بدین خاطر است که به اینها توالی گریز می‌گویند: کج‌خط وارون به شما امکان می‌دهد که از تفسیر معمول نویسه‌ی نقل‌قول یک‌ه بگریزید. به جای اینکه از آن به عنوان علامت پایان رشته استفاده کنید، می‌توانید از آن به جای آپوستروف بهره بگیرید:

```
'You're right, it can\'t be a quote'
```

جدول ۱-۲ توالی‌های گریز در جاوا اسکریپت و نویسه‌ی معادل آنها را نشان می‌دهد. دو توالی گریز حالت عمومی دارند و می‌توان از آنها برای نشان دادن هر گونه نویسه‌ای با معین کردن کد نویسه‌ی لاتین-۱ یا یونیکد آن به صورت یک عدد شانزدهگانی استفاده کرد. مثلاً توالی \xA9 نشان دهنده‌ی نماد حق‌التکثیر است، که کدگذاری لاتین-۱ آن به صورت عدد شانزدهگانی A9 داده شده است. به همین ترتیب، گریز \u نشان دهنده‌ی یک نویسه‌ی دلخواه یونیکد است که با چهار رقم شانزدهگانی مشخص شده است؛ مثلاً \u03C0 نشان دهنده‌ی نویسه‌ی π است.

جدول ۱-۲: توالی‌های گریز در جاوا اسکریپت

توالی	نویسه‌ی نشان داده شده
\0	نویسه‌ی NUL (\u0000)
\b	بک‌اسپیس [پس‌بر] (\u0008)
\t	تب [جهش] افقی (\u0009)
\n	سطر جدید (\u000A)
\v	تب عمودی (\u000B)
\f	صفحه‌ی بعد (\u000C)
\r	سر سطر (\u000D)
\"	نقل‌قول دوتایی (\u0022)
'	آپوستروف یا نقل‌قول یک‌ه (\u0027)
\\	کج‌خط وارون (\u005C)
\xxx	نویسه‌ی لاتین-۱ مشخص شده با دو رقم شانزدهگانی XX
\uXXXX	نویسه‌ی یونیکد مشخص شده با چهار رقم شانزدهگانی XXXX

اگر نویسه‌ی \ قبل از هر نویسه‌ی دیگری غیر از موارد مشخص شده در جدول ۲-۱ بیاید، کج‌خط وارون به سادگی مورد چشم‌پوشی قرار می‌گیرد (هر چند که البته ممکن است که ویراست‌های آینده‌ی زبان توالی‌های گریز جدیدی را تعریف کنند). مثلاً \# همان # محسوب می‌شود. اکما اسکریپت ۵ اجازه می‌دهد که قبل از پایان سطر یک کج‌خط وارون گذاشته شود تا مقدار لفظی رشته بتواند چند سطر را اشغال کند.

یکی از ویژگی‌های ذاتی جاوا اسکریپت، توانایی به هم پیوستن رشته‌ها است. اگر عملگر + را با اعداد به کار ببرید، آنها را با هم جمع می‌کند. ولی اگر این عملگر را با رشته‌ها استفاده کنید، با افزودن رشته‌ی دوم به آخر رشته‌ی اول، آنها را به هم می‌پیوندد. به عنوان مثال:

```
msg = "Hello, " + "world"; // => "Hello, world"
```

برای تعیین طول یک رشته، یعنی تعداد مقادیر ۱۶-بیتی موجود در آن، از خصلت length رشته استفاده کنید. طول رشته‌ی s به صورت زیر تعیین می‌شود:

s.length

علاوه بر خصلت length، متدهای چندی نیز وجود دارد که می‌توانید آنها را روی رشته‌ها فراخوانی کنید (باز برای جزئیات کامل به بخش مربوطه مراجعه کنید):

```
var s = "hello, world" // Start with some text.
s.charAt(0)           // => "h": the first character.
s.charAt(s.length-1) // => "d": the last character.
s.substring(1,4)     // => "ell": chars 2, 3, and 4
s.slice(1,4)         // => "ell": same thing
s.slice(-3)          // => "rld": last 3 characters
s.indexOf("l")       // => 2: position of first l.
s.lastIndexOf("l")  // => 10: position of last l.
s.indexOf("l", 3)    // => 3: position at or after 3
s.split(", ")       // => ["hello", "world"]
s.replace("h", "H") // => "Hello, world":
                    // replaces all instances
s.toUpperCase()     // => "HELLO, WORLD"
```

به یاد داشته باشید که رشته‌ها در جاوا اسکریپت تغییرناپذیر هستند. متدهایی مانند `replace()` و `toUpperCase()`، رشته‌های جدید بر می‌گردانند: این متدها رشته‌ای را که روی آن خوانده می‌شوند، تغییر نمی‌دهند.

در اکما اسکریپت ۵، با رشته‌ها می‌توانید به عنوان آرایه‌های فقط‌خواندنی برخورد کنید، یعنی می‌توانید به جای متد `charAt()`، با استفاده از کروش به نویسه‌های رشته (مقادیر ۰-۱۶-بیتی آن) دسترسی پیدا کنید:

```
s = "hello, world";
s[0]           // => "h"
s[s.length-1] // => "d"
```

مقادیر بولی

یک مقدار بولی نشان دهنده‌ی درست یا نادرست، روشن یا خاموش، آری یا نه است. برای این نوع، فقط دو مقدار ممکن وجود دارد. واژه‌های ذخیره شده‌ی `true` و `false` نشان دهنده‌ی این دو حالت هستند.

مقادیر بولی غالباً نتیجه‌ی مقایسه‌هایی هستند که در برنامه‌ی جاوا اسکریپت خود می‌کنید. به عنوان مثال:

```
a == 4
```

این کد امتحان می‌کند که آیا مقدار متغیر `a` برابر با عدد 4 است. اگر چنین باشد، نتیجه‌ی مقایسه، مقدار بولی `true` خواهد بود. اگر `a` برابر با 4 نباشد، آنگاه نتیجه‌ی مقایسه `false` است.

مقادیر بولی به طور شایعی در ساختارهای کنترلی جاوا اسکریپت استفاده می‌شوند. مثلاً دستورالعمل `if/else` در صورتی که یک مقدار بولی `true` باشد یک عمل را انجام می‌دهد و در صورتی که `false` باشد، عمل دیگری را انجام می‌دهد. معمولاً یک مقایسه‌ای را که مقدار بولی ایجاد می‌کند، مستقیماً با یک دستورالعمل که از آن استفاده می‌کند، ترکیب می‌کنیم. نتیجه به این صورت است:

```
if (a == 4)
  b = b + 1;
else
  a = a + 1;
```

این کد واری می‌کند که آیا `a` برابر با 4 است. اگر چنین باشد، به `b` عدد 1 اضافه می‌کند؛ در غیر این صورت، به `a` مقدار 1 اضافه می‌کند.

به طوری که در قسمت «تبدیل انواع» در صفحه‌ی ۲۸ خواهیم دید، هر مقدار جاوا اسکریپت را می‌توان به یک مقدار بولی تبدیل کرد. مقادیر زیر به `false` تبدیل می‌شوند و بنا بر این، مانند آن عمل می‌کنند:

```
undefined
null
0
-0
NaN
"" // the empty string
```

تمام مقادیر دیگر، از جمله تمام اشیا (و آرایه‌ها)، به `true` تبدیل می‌شوند و مانند آن رفتار می‌کنند. گاه به `false`، و شش مقداری که به آن تبدیل می‌شوند، مقادیر نادرست‌مانند، و به تمام مقادیر دیگر، مقادیر درست‌مانند می‌گویند. هر گاه

جاوا اسکریپت انتظار یک مقدار بولی دارد، یک مقدار نادرست مانند `false` رفتار می‌کند و یک مقدار درست مانند `true`.

به عنوان مثال، فرض کنید که متغیر `o` یا حاوی یک شیء است و یا حاوی مقدار `null` می‌توانید ناهیچ بودن `o` را با یک دستورالعمل `if` به صراحت به صورت زیر آزمایش کنید:

```
if (o !== null) ...
```

عملگر نامساوی `!=='`، `o` را با `null` مقایسه می‌کند و مقدار `true` یا `false` را بر می‌گرداند. ولی می‌توانید مقایسه را حذف کنید و به جای آن از این واقعیت استفاده کنید که `null` نادرست‌مانند است و اشیا درست‌مانند هستند:

```
if (o) ...
```

در حالت اول، بدنه‌ی دستورالعمل `if` تنها اگر `o`، `null` نباشد، اجرا خواهد شد. حالت دوم کمتر سخت‌گیری می‌کند: این حالت بدنه‌ی دستورالعمل `if` را فقط در صورتی که `o`، `false` یا هر مقدار نادرست‌مانند (مانند `null` یا `undefined`) نباشد، اجرا خواهد کرد. اینکه کدام دستورالعمل `if` برای برنامه‌ی شما مناسب است، در حقیقت، بستگی به مقادیری دارد که انتظار دارید به `o` تخصیص دهید. اگر نیاز داشته باشید که `null` را از `0` و `""` افتراق دهید، آنگاه باید از مقایسه‌ی صریح استفاده کنید.

undefined و null

`null` از کلیدواژه‌های زبان است که به یک مقدار ویژه ارزیابی می‌شود که معمولاً از آن برای نشان دادن فقدان مقدار استفاده می‌شود. اگر از عملگر `typeof` روی `null` استفاده کنید، رشته‌ی `"object"` بر می‌گرداند، بدان معنا که `null` را می‌توان شیء خاصی دانست که نشان دهنده‌ی «عدم وجود شیء» است. اما، در عمل، معمولاً `null` به عنوان تنها عضو نوع خاص خود به حساب می‌آید، و می‌توان از آن برای نشان دادن «عدم وجود مقدار» جهت اعداد و رشته‌ها و نیز اشیا استفاده کرد. اکثر زبان‌های برنامه‌نویسی، چیزی معادل `null` جاوا اسکریپت دارند: مثلاً ممکن است آن را به عنوان `null` یا `nil` بشناسید.

جاوا اسکریپت یک مقدار دیگر نیز دارد که نشان دهنده‌ی فقدان مقدار است. مقدار `undefined` نوع عمیق‌تری از فقدان را نشان می‌دهد. این مقدار، مقدار

متغیرهایی است که مقداردهی اولیه نشده‌اند و یا مقدار خصلتی که در یک شیء وجود ندارد یا عنصری که در یک آرایه موجود نیست. مقدار `undefined` به وسیله‌ی توابعی هم که هیچ مقدار برگشتی ندارند، بر گردانده می‌شود، و مقدار پارامترهای تابع که هیچ آوندی برای آنها داده نشده است، نیز همین است. `undefined` یک متغیر از پیش تعریف شده‌ی سراسری است (نه یک کلیدواژه‌ی زبان مانند `null`)، که به مقدار تعریف نشده مقداردهی اولیه شده است. اگر عملگر `typeof` را روی مقدار `undefined` اعمال کنید، مقدار `"undefined"` بر می‌گرداند، که نشان دهنده‌ی آن است که این مقدار، تنها عضو از یک نوع خاص است.

علیرغم این تفاوت، `null` و `undefined` هر دو نشان دهنده‌ی فقدان مقدار هستند، و غالباً می‌توان آنها را به جای هم به کار برد. عملگر تساوی `==` آنها را مساوی می‌داند. (برای افتراق آنها از عملگر تساوی اکید `===` استفاده کنید.) هر دو، مقادیر نادرست‌مانند هستند—یعنی وقتی مقدار بولی مورد نیاز است، همانند `false` رفتار می‌کنند. نه `null` و نه `undefined` هیچگونه خصلت یا روشی ندارند. در حقیقت، استفاده از `.` یا `[]` برای دسترسی به یک خصلت یا متد این مقادیر، موجب بروز یک `TypeError` می‌شود.

نشیء سراسری

در قسمت‌های فوق، انواع و مقادیر بدوی جاوا اسکریپت را مورد بحث قرار دادیم. انواع شیء—اشیا، آرایه‌ها، و توابع—هر کدام در فصل خود بعداً مورد بحث قرار خواهند گرفت. ولی یک مقدار شیئی بسیار مهم هست که لازم است در اینجا به آن بپردازیم. شیء سراسری یک شیء معمولی جاوا اسکریپت است که مقصودی بسیار مهم را بر آورده می‌سازد: خصلت‌های این شیء، نمادهای تعریف شده به صورت سراسری هستند که در دسترس برنامه‌ی جاوا اسکریپت قرار دارند. وقتی که تفسیرگر جاوا اسکریپت شروع به کار می‌کند (یا هر گاه یک مرورگر وب، صفحه‌ی جدیدی را بار می‌کند)، یک شیء سراسری جدید ایجاد می‌کند و مجموعه‌ی ابتدایی خصلت‌هایی را که تعریف می‌کند، به آن می‌دهد:

- خصلت‌های سراسری مانند `Infinity`، `undefined` و `NaN`

- توابع سراسری مانند `parseInt()`، `isNaN()` (رک). قسمت «تبدیل انواع» در صفحه ۲۸، و `eval()` (رک). قسمت «عبارت‌های ارزیابی» در صفحه ۵۷
- توابع سازنده مانند `Date()`، `RegExp()`، `String()`، `Object()`، و `Array()`
- اشیای سراسری مانند `Math`، و `JSON` (رک). قسمت «سریالیزه کردن خصلت‌ها و اشیا» در صفحه ۱۰۱

خصلت‌های ابتدایی شیء سراسری، واژه‌های ذخیره نیستند، ولی شایسته است که به عنوان واژه‌های ذخیره در نظر گرفته شوند. برخی از این خصلت‌های سراسری را قبلاً در این فصل بررسی کردیم. اکثر خصلت‌های دیگر را در ادامه‌ی دیگر این کتاب بررسی خواهیم کرد.

در کد سطح بالا—یعنی کد جاوا اسکریپتی که جزئی از یک تابع نیست—می‌توانید از کلیدواژه‌ی `this` برای اشاره به شیء سراسری استفاده کنید:

```
var global = this; // refer to the global object
```

در جاوا اسکریپت سمت مشتری، شیء `Window` به عنوان شیء سراسری عمل می‌کند. این شیء سراسری پنجره، یک خصلت `window` دارد که به خودش اشاره می‌کند و می‌توانید از آن برای اشاره به شیء سراسری استفاده کنید. شیء پنجره خصلت‌های هسته‌ی سراسری را تعریف می‌کند، ولی علاوه بر آنها، خصلت‌های سراسری زیاد دیگری را نیز تعریف می‌کند که مختص مرورگرهای وب و جاوا اسکریپت سمت مشتری هستند (رک. فصل ۱۰).

شیء سراسری وقتی که نخستین بار تعریف شد، تمام مقادیر سراسری از پیش تعریف شده‌ی جاوا اسکریپت را تعریف می‌کند. ولی این شیء ویژه، مقادیر سراسری تعریف شده توسط برنامه را نیز در خود دارد. اگر کد شما یک متغیر سراسری تعریف کند، آن متغیر یک خصلت شیء سراسری خواهد بود.

تبدیل انواع

جاوا اسکریپت در باره‌ی نوع مقادیری که لازم دارد، بسیار انعطاف‌پذیر است. این کار را در مورد مقادیر بولی دیدیم: وقتی که جاوا اسکریپت نیاز به یک مقدار بولی دارد، می‌توانید مقداری از هر نوع را به آن بدهید، و جاوا اسکریپت در صورت لزوم

آن را تبدیل خواهد کرد. بعضی مقادیر (مقادیر «درست‌مانند») به true تبدیل می‌شوند، و برخی دیگر (مقادیر «نادرست‌مانند») به false. همین مطلب برای انواع دیگر نیز درست است: اگر جاوا اسکریپت نیاز به یک رشته داشته باشد، هر مقداری به آن بدهید، آن را تبدیل به رشته خواهد کرد. اگر جاوا اسکریپت یک عدد بخواهد، تلاش خواهد کرد هر مقداری به آن بدهید، آن را به یک عدد (یا، اگر تبدیل معنی‌داری قابل انجام نباشد، به NaN) تبدیل کند. چند مثال:

```
10 + " objects" // => "10 objects". 10 -> string
"7" * "4"      // => 28: both strings -> numbers
var n = 1 - "x"; // => NaN: "x" can't convert to a number
n + " objects" // => "NaN objects": NaN -> "NaN"
```

جدول ۲-۲ چگونگی تبدیل مقادیر از نوعی به نوع دیگر در جاوا اسکریپت را نشان می‌دهد. مقادیری که با حروف پررنگ نوشته شده‌اند، مواردی هستند که ممکن است برایتان تعجب‌آور باشد. خانه‌های خالی نشان دهنده آن است که تبدیل لازم نیست و هیچ تبدیلی انجام نمی‌شود.

جدول ۲-۲: تبدیل انواع در جاوا اسکریپت

				مقدار
تبدیل به:				
رشته	عدد	بولی	شیء	
"undefined"	NaN	false	خطای <i>TypeError</i> بر می‌انگیزد	undefined
"null"	0	false	خطای <i>TypeError</i> بر می‌انگیزد	null
"true"	1		Boolean(true)	true
"false"	0		Boolean(false)	false
" (رشته‌ی خالی)	0	false	String("")	" (رشته‌ی خالی)
"1.2" (نا تهی، عددی)	1.2	true	String("1.2")	"1.2" (نا تهی، عددی)
"one" (نا تهی، غیر عددی)	NaN	true	String("one")	"one" (نا تهی، غیر عددی)
"0"		false	Number(0)	0
"0"		false	Number(-0)	-0

رشته را ذخیره نمایید و آن خصلت را مقدارگیری کنید تا آن را بازیابی کنید. ولی این اشیا یک رابط برنامه‌نویسی رسمی‌تر مبتنی بر متد را نیز تعریف می‌کنند. به منظور ذخیره‌سازی یک مقدار، نام و مقدار را به متد `setItem()` بدهید. برای بازیابی یک مقدار، نام و مقدار را به `getItem()` بدهید. برای حذف یک مقدار، نام آن را به `removeItem()` بدهید. (در اکثر مرورگرها، از عملگر `delete` نیز برای حذف کردن یک مقدار می‌توانید استفاده کنید، درست همانطور که روی یک شیء عادی عمل می‌کنید، ولی این روش در اینترنت اکسپلورر ۸ مؤثر نیست.) برای حذف همه‌ی مقادیر ذخیره شده، متد `clear()` را (بدون آوردن) فراخوانی کنید. و بالاخره، برای برشماری نام‌های تمام مقادیر ذخیره شده، از خصلت `length` استفاده کنید و اعداد ۰ تا `length-1` را به متد `key()` بدهید. در اینجا چند مثال را با استفاده از `localStorage` می‌بینید. همین‌کد می‌تواند با استفاده از `sessionStorage` نیز استفاده شود:

```
localStorage.setItem("x", 1); // Store an item "x"
localStorage.getItem("x"); // Retrieve its value

// Enumerate all stored name/value pairs
// Length gives the # of pairs
for(var i = 0; i < localStorage.length; i++) {
    // Get the name of pair i
    var name = localStorage.key(i);
    // Get the value of that pair
    var value = localStorage.getItem(name);
}

localStorage.removeItem("x"); // Delete the item "x"
localStorage.clear(); // Delete any other items, too
```

رویدادهای ذخیره‌سازی

هر گاه داده‌های ذخیره شده در `localStorage` یا `sessionStorage` تغییر می‌کند، مرورگر یک رویداد `storage` را روی تمام اشیای `Window` دیگری که داده‌ها برای آنها قابل رؤیت است (ولی نه روی پنجره‌ای که تغییر را انجام داده است)، بر می‌انگیزد. اگر در مرورگر دو برگه با صفحاتی از مبدأ یکسان در آنها باز باشد،

یکی از این دو صفحه مقداری را در `localStorage` ذخیره کند، برگه‌ی دیگر یک رویداد `storage` دریافت خواهد کرد. دقت کنید که قلمرو `sessionStorage` پنجره‌ی سطح بالا است، بنا بر این، رویدادهای `storage` برای تغییرات `sessionStorage` فقط زمانی برانگیخته می‌شود که از فریم استفاده شده باشد. همچنین، دقت کنید که رویدادهای `storage` تنها زمانی برانگیخته می‌شود که ذخیره‌سازی واقعاً تغییر کند. اگر به یک آیتم ذخیره شده همان مقدار فعلی آن را تخصیص دهید، رویدادی برانگیخته نمی‌شود، و حذف آیتمی که موجود نیست هم سبب برانگیخته شدن رویداد نخواهد شد.

برای ثبت رویدادپرداز برای رویدادهای `storage` از متد `addEventListener()` (یا در اینترنت اکسپلورر از `attachEvent()`) استفاده کنید. در اکثر مرورگرها، می‌توانید به خصلت `onstorage` شیء `Window` هم مقداره‌ی کنید، ولی در زمان نگارش این کتاب، فایرفاکس از این خصلت پشتیبانی نمی‌کند.

شیء رویداد مرتبط با یک رویداد `storage` پنج خصلت مهم دارد (متأسفانه اینها در اینترنت اکسپلورر ۸ پشتیبانی نمی‌شوند):

key

نام یا کلید آیتمی که مقداره‌ی یا حذف شده است. اگر متد `clear()` فراخوانی شود، این خصلت `null` خواهد بود.

newValue

حاوی مقدار جدید آیتم است، یا حاوی `null` در صورتی که `removeItem()` فراخوانی شده باشد.

oldValue

حاوی مقدار قدیم یک آیتم تغییر داده شده یا حذف شده است، و اگر آیتم جدیدی درج شده باشد، حاوی `null` است.

storageArea

این خصلت برابر با خصلت `localStorage` یا `sessionStorage` شیء `Window` هدف خواهد بود.

URL سندی که اسکریپت آن این تغییر ذخیره‌سازی را انجام داده است (به صورت یک رشته).

و بالاخره، توجه کنید که از `localStorage` و رویداد `storage` می‌توان به عنوان یک ساز و کار خبررسانی استفاده کرد که در آن مرورگر پیغامی را برای همه‌ی پنجره‌هایی که از وب‌سایت یکسانی دیدن می‌کنند، می‌فرستد. مثلاً اگر کاربر درخواست کند که وب‌سایت انجام پویانمایی را متوقف کند، سایت می‌تواند این ترجیح را در `localStorage` ذخیره‌سازی کند، به طوری که بتواند در دیدارهای آینده نیز به آن عمل کند. و بر اثر ذخیره کردن این ترجیح، رویدادی ایجاد می‌شود که به پنجره‌های دیگری نیز که همان سایت را نشان می‌دهند، امکان می‌دهد که آنها هم به آن درخواست عمل کنند. باز به عنوان یک مثال دیگر، یک برنامه‌ی ویرایش تصاویر مبتنی بر وب را در نظر بگیرید که به کاربر امکان می‌دهد که مجموعه‌ی ابزارها را در پنجره‌های جداگانه نمایش دهد. وقتی که کاربر ابزاری را انتخاب می‌کند، برنامه از `localStorage` برای ذخیره کردن حالت فعلی و نیز خبررسانی به پنجره‌های دیگری دایر بر اینکه ابزار جدید انتخاب شده است، استفاده می‌کند.

کوکی‌ها

کوکی مقدار کمی داده‌های دارای نام است که توسط مرورگر ذخیره می‌شود و با یک صفحه‌ی وب یا وب‌سایت مرتبط می‌شود. کوکی‌ها در ابتدا برای برنامه‌نویسی سمت سرور طراحی شدند و در پایین‌ترین سطح، به عنوان بسطی برای پروتکل HTTP پیاده‌سازی شده‌اند. داده‌های کوکی به طور خودکار بین مرورگر وب و سرور وب منتقل می‌شود، بنا بر این، اسکریپت‌های سمت سرور می‌توانند مقادیر کوکی را که در سمت مشتری ذخیره شده‌اند، بخوانند و بنویسند. در این قسمت نشان می‌دهیم که چگونه اسکریپت‌های سمت مشتری نیز با استفاده از خصلت `cookie` شیء `Document` می‌توانند کوکی‌ها را دستکاری کنند.

رابط برنامه‌نویسی دستکاری کوکی‌ها یک رابط برنامه‌نویسی قدیمی است و در نتیجه، به طور وسیعی از آن پشتیبانی می‌شود. متأسفانه این رابط برنامه‌نویسی تا حدودی مرموز هم هست. هیچ متدی در آن وجود ندارد: مقدارگیری، مقداردهی، و حذف کوکی‌ها از طریق خواندن و نوشتن خصلت `cookie` شیء `Document` با

استفاده از رشته‌های دارای قالب خاص انجام می‌شود. طول عمر و قلمرو هر کوکی را می‌توان جداگانه با استفاده از صفات کوکی تعیین کرد. این صفات‌ها نیز با فرمت خاص رشته روی همان خصلت `cookie` تعیین می‌شوند.

در قسمت‌های زیر، صفات کوکی را که مشخص کننده طول عمر و قلمرو آن هستند، توضیح می‌دهیم، و سپس چگونگی مقداردهی و مقدارگیری کوکی‌ها در جاوا اسکریپت را بررسی می‌کنیم.

صفات‌های کوکی: طول عمر و قلمرو

هر کوکی، علاوه بر نام و مقدار، می‌تواند دارای صفاتی اختیاری باشد که طول عمر و قلمرو آن را مشخص می‌کنند. کوکی‌ها به طور پیش‌فرض گذرا هستند؛ مقداری که آنها ذخیره می‌کنند، در مدت جلسه‌ی مرورگر وب دوام می‌آورد، ولی وقتی که کاربر از مرورگر خارج می‌شود، پاک می‌شود. دقت کنید که طول عمر کوکی با طول عمر `sessionStorage` به طور ظریفی متفاوت است: قلمرو کوکی‌ها محدود به یک پنجره نیست، و به طور پیش‌فرض، طول عمر آنها به اندازه‌ی تمام برنامه‌ی مرورگر است، نه طول عمر یک پنجره‌ی واحد آن. اگر بخواهید که یک کوکی فراتر از یک جلسه‌ی مرورگری دوام داشته باشد، باید با مشخص کردن صفت `max-age` به مرورگر بگویید که چه مدت (بر حسب ثانیه) می‌خواهید که کوکی را نگه دارید. اگر طول عمر را مشخص کنید، مرورگر کوکی‌ها را در یک فایل ذخیره خواهد کرد و آنها را تنها زمانی که تاریخ آنها منقضی شود، پاک خواهد کرد.

قابل رؤیت بودن کوکی، همانند `localStorage` و `sessionStorage`، بر حسب مبدأ سند و نیز بر حسب مسیر سند است. این قلمرو را می‌توان با استفاده از صفات `path` و `domain` کوکی تنظیم کرد. به طور پیش‌فرض، کوکی وابسته به صفحه‌ی وب ایجاد کننده‌ی آن و تمام صفحات وبی است که در همان پوشه یا در زیرپوشه‌های آن پوشه قرار دارند، و از طریق این صفحات قابل دسترس است. برای مثال، اگر صفحه‌ی `http://www.example.com/catalog/index.html` یک کوکی ایجاد کند، آن کوکی برای نشانی‌های `http://www.example.com/catalog/order.html` و `http://www.example.com/catalog/widgets/index.html` هم قابل رؤیت خواهد بود، ولی برای `http://www.example.com/about.html` قابل رؤیت نخواهد بود.

غالباً این رفتار رؤیت‌پذیری پیش‌فرض دقیقاً همان چیزی است که شما می‌خواهید. ولی بعضی از وقت‌ها ممکن است مایل باشید که در سرتاسر یک وب‌سایت از مقادیر کوکی‌ها استفاده کنید، صرف نظر از اینکه آن کوکی‌ها توسط کدام صفحه ایجاد شده‌اند. مثلاً اگر کاربر آدرس پستی خود را در فرمی در یک صفحه وارد کرده باشد، ممکن است تمایل داشته باشید که آن آدرس را ذخیره کنید تا دفعه‌ی بعد که به آن صفحه مراجعه می‌کند، از آن آدرس به عنوان پیش‌فرض استفاده شود، و در صفحات دیگر هم هر گاه در یک فرم کاملاً متفاوت از او خواسته می‌شود که آدرس پستی خود را وارد کند، از آن آدرس استفاده شود. برای اینکه امکان این کار فراهم شود، باید مقدار صفت *path* کوکی را مشخص کنید. آنگاه هر صفحه‌ی وبی که URL آن با پیشوندی که شما تعیین کرده‌اید، شروع شود، می‌تواند از آن کوکی استفاده کند. مثلاً اگر در یک کوکی که به وسیله‌ی صفحه‌ی <http://www.example.com/catalog/widgets/index.html> تعریف شده است، مقدار مسیر `"/catalog"` تعیین شود، آن کوکی در صفحه‌ی <http://www.example.com/catalog/order.html> نیز قابل رؤیت خواهد بود. یا اینکه اگر مسیر آن `"/"` تعیین شود، کوکی برای تمام صفحات موجود در سرور وب <http://www.example.com> قابل دسترس خواهد بود.

مقداردهی *path* کوکی به `"/"`، قلمروی همانند `localStorage` ایجاد می‌کند، و نیز مشخص می‌کند که مرورگر هر گاه یک صفحه‌ی وب از آن سایت را درخواست می‌کند، باید نام و مقدار کوکی را برای سرور بفرستد.

به طور پیش‌فرض، قلمرو کوکی‌ها محدود به مبدأ سند است. اما وب‌سایت‌های بزرگ ممکن است بخواهند که کوکی بین زیردامنه‌های آنها مشترک باشد. مثلاً سروری با نشانی `order.example.com` ممکن است بخواهد مقادیر کوکی تعیین شده توسط `catalog.example.com` را بخواند. در اینجا است که صفت `domain` وارد ماجرا می‌شود. اگر یک کوکی که به وسیله‌ی صفحه‌ای در `catalog.example.com` ایجاد شده است، مقدار صفت *path* خود را `"/"` و مقدار صفت `domain` خود را `"example.com"` تعیین کند، آن کوکی در دسترس تمام صفحات وب موجود در `catalog.example.com`، `orders.example.com` و هر سرور دیگری در دامنه‌ی `example.com` خواهد بود. اگر صفت `domain` برای یک کوکی تعیین نشود، مقدار پیش‌فرض آن نام میزبان سرور وبی است که صفحه را

می‌فرستد. دقت کنید که دامنه‌ی کوکی را نمی‌توانید به دامنه‌ای غیر از دامنه‌ی سرور خود مقداردهی دهید.

آخرین صفت کوکی یک صفت بولی به نام *secure* است که چگونگی انتقال مقادیر بر روی شبکه را مشخص می‌کند. به طور پیش‌فرض، کوکی‌ها ناامن هستند، یعنی روی اتصال عادی و ناامن HTTP فرستاده می‌شوند. اما اگر یک کوکی به عنوان کوکی امن تعیین شود، انتقال آن فقط زمانی صورت می‌گیرد که مرورگر و سرور از طریق HTTPS یا پروتکل امن دیگری به یکدیگر وصل شده باشند.

مقداردهی کوکی‌ها

برای اینکه یک مقدار کوکی گذرا را با سند فعلی مرتبط کنید، کافی است که رشته‌ای به صورت زیر به خصلت *cookie* تخصیص دهید:

```
name=value
```

برای مثال:

```
var v = encodeURIComponent(document.lastModified);  
document.cookie = "version=" + v;
```

دفعه‌ی بعدی که خصلت *cookie* را بخوانید، زوج نام/مقداری که ذخیره کردید، در لیست کوکی‌های سند مشاهده خواهد شد. مقدار کوکی‌ها نمی‌تواند حاوی ویرگول نقطه، ویرگول، یا فضای سفید باشد. بر این اساس، بهتر است که قبل از ذخیره‌سازی مقدار در کوکی، آن را با استفاده از تابع `encodeURIComponent()` در جاوا اسکریپت هسته کدگذاری کنید.

یک کوکی که صرفاً با ذکر نام و مقدار تعیین شده باشد، تا پایان جلسه‌ی کنونی مرورگری دوام خواهد یافت، ولی وقتی که کاربر از مرورگر خارج شود، پاک خواهد شد. برای ایجاد یک کوکی که بتواند در جلسات متعدد مرورگر دوام داشته باشد، باید طول عمر آن را (بر حسب ثانیه) با استفاده از صفت *max-age* تعیین کنید. برای این منظور، رشته‌ای به صورت زیر به خصلت *cookie* تخصیص دهید:

```
name=value; max-age=seconds
```

تابع زیر یک کوکی را با صفت اختیاری *max-age* تعیین می‌کند:

```
// Store the name/value pair as a cookie, encoding  
// the value with encodeURIComponent() in order to  
// escape semicolons, commas, and spaces.
```

```

// If daysToLive is a number, set the max-age attribute
// so that the cookie expires after the specified
// number of days. Pass 0 to delete a cookie.
function setCookie(name, value, daysToLive) {
    var cookie = name + "=" + encodeURIComponent(value);
    if (typeof daysToLive === "number")
        cookie += "; max-age=" + (daysToLive*60*60*24);
    document.cookie = cookie;
}

```

به همین ترتیب، می‌توانید مقادیر صفات `domain`، `path`، و `secure` کوکی را با اضافه کردن رشته‌هایی با قالب زیر به مقدار کوکی قبل از نوشتن آن در خصلت `cookie` مشخص کنید:

```

; path=path
; domain=domain
; secure

```

برای تغییر مقدار یک کوکی، آن را دوباره با همان نام، مسیر، و دامنه، به همراه مقدار جدید، مقداردهی کنید. طول عمر کوکی را می‌توانید با تغییر دادن مقدار آن و مشخص کردن صفت `max-age` تعیین کنید.

برای حذف یک کوکی، آن را دوباره با همان نام، مسیر، و دامنه، با یک مقدار دلخواه (یا خالی)، و با مقدار `max-age` برای صفت `max-age`، مقداردهی کنید.

خواندن کوکی‌ها

وقتی که از خصلت `cookie` در یک عبارت جاوا اسکریپت استفاده می‌کنید، مقداری که بر می‌گرداند، رشته‌ای است که حاوی همه‌ی کوکی‌هایی است که به سند فعلی مربوط می‌شود. این رشته لیستی از زوج‌های `name = value` است که با یک ویرگول نقطه و یک فضای خالی از هم جدا شده‌اند. `value` کوکی شامل صفت‌هایی که ممکن است برای کوکی تعیین شده باشند، نیست. برای اینکه از خصلت `document.cookie` استفاده کنید، معمولاً باید با استفاده از متد `split()` آن را به زوج‌های نام و مقدار بشکنید.

وقتی که مقدار یک کوکی را از خصلت `cookie` استخراج کردید، باید آن مقدار را بر اساس هر گونه قالب یا کدگذاری که توسط ایجاد کننده‌ی کوکی استفاده شده

است، تفسیر کنید. مثلاً شاید لازم باشد که مقدار کوکی را به تابع `decodeURIComponent()` و بعد به `JSON.parse()` بدهید.

مثال ۱-۱۴ یک تابع `getCookie()` تعریف می‌کند که خصلت `document.cookie` را تجزیه می‌کند و شیئی را بر می‌گرداند که خصلت‌های آن، نام و مقدار کوکی‌های سند را مشخص می‌کنند.

مثال ۱-۱۴: تجزیه‌ی خصلت `document.cookies`

```
// Return the document's cookies as an object of
// name/value pairs. Assume that cookie values
// are encoded with encodeURIComponent().
function getCookies() {
    var cookies = {}; // The object we return
    var all = document.cookie; // All cookies
    if (all === "") // If empty
        return cookies; // return an empty object
    // Split string into name=value pairs
    var list = all.split("; ");
    // Loop through the name=value pairs
    for(var i = 0; i < list.length; i++) {
        var cookie = list[i];
        // Split each pair at the = sign
        var p = cookie.indexOf("=");
        var name = cookie.substring(0,p);
        var value = cookie.substring(p+1);
        // Store the name and decoded value
        cookies[name] = decodeURIComponent(value);
    }
    return cookies;
}
```

محدودیت‌های کوکی‌ها

هدف از کوکی‌ها ذخیره‌سازی مقادیر اندک داده‌ها توسط اسکریپت‌های سمت سرور است، و هر بار که URL مرتبطی درخواست می‌شود، آن داده‌ها به سرور فرستاده

می‌شود. استاندارد‌ی که کوکی‌ها را تعریف کرده است، سازندگان مرورگر را تشویق می‌کند که تعداد نامحدود کوکی‌ها را با اندازه‌ی نامحدود اجازه دهند، ولی مرورگرها را ملزم نمی‌کند که جمعاً بیش از ۳۰۰ کوکی، برای هر سرور وب بیش از ۲۰ کوکی، و یا برای هر کوکی بیش از ۴ KB داده‌ها (که نام و مقدار هر دو جزء این ۴ KB هستند)، را اجازه دهند. در عمل، مرورگرها جمعاً خیلی بیشتر از ۳۰۰ کوکی را اجازه می‌دهند، ولی برخی از آنها ممکن است هنوز هم محدودیت ۴ KB را اعمال کنند.

نمایه

* نویسه‌ی تکرار، ۱۷۵
*= عملگر، ۵۷
/ عملگر تقسیم، ۱۹، ۴۶
/[]/ عبارت‌های قاعده‌مند، ۱۷۴
/* */ توضیح چندسطری، ۱۳
// توضیح یک‌سطری، ۱۳
\ انطباق با ارقام، ۱۷۶
\B غیر مرز کلمه، ۱۷۹
\b مرز کلمه، ۱۷۹
\d رقم اسکی، ۱۷۴
\D غیر رقم اسکی، ۱۷۴
\S غیر فضای سفید یونیکد، ۱۷۴
\s فضای سفید یونیکد، ۱۷۴
\W غیر نویسه‌ی کلمه‌ی اسکی، ۱۷۴
\w نویسه‌ی کلمه‌ی اسکی، ۱۷۴
& عملگر AND بیتی، ۴۳، ۴۸
&& AND منطقی، ۴۳، ۵۴
&= & عملگر، ۵۷
> عملگر بزرگ‌تر، ۴۳، ۵۲
>> جابجایی به راست با بسط علامت، ۴۳، ۵۰
>>> جابجایی به راست با بسط صفر، ۴۳، ۵۰
>= عملگر بزرگ‌تر یا مساوی، ۴۳، ۵۲
< عملگر کوچک‌تر، ۴۳، ۵۲
<= عملگر کوچک‌تر یا مساوی، ۴۳، ۵۲
#top شناسه، ۱۹۰
% عملگر نهشته، ۱۹، ۴۶
^ انطباق با ابتدا، ۱۷۹
^ دسته‌ی نویسه‌ای منفی، ۱۷۳
^ عملگر XOR بیتی، ۴۳، ۴۹

— نمادها —

_ شناسه، ۱۴
- عملگر تفریق، ۱۹، ۴۷
- عملگر منهای یگانی، ۴۷
- منفی کردن عدد، ۴۲
-- عملگر کاهش واحد، ۴۲، ۴۸، ۶۵
-- کاهش واحد پیشوندی/پسوندی، ۴۲
, عملگر ویرگول، ۴۴، ۶۱
, دستورالعمل خالی، ۶۷
, ویرگول نقطه، ۱۵
! عملگر یگانی منفی، ۳۱
! نقیض مقدار بولی، ۴۳، ۵۶
!= عملگر عدم تساوی غیراکید، ۲۶، ۴۳، ۵۲
== عملگر عدم تساوی اکید، ۴۳، ۵۱، ۹۹
? نویسه‌ی تکرار، ۱۷۵
?: عملگر شرطی، ۴۴، ۵۹
. عملگر نقطه، ۲۷
. مقدار خصلت، ۹۶
" (رشته)، ۲۱
" " رشته، ۲۱
(?): فقط گروه‌بندی، ۱۷۷
(?!): تأیید پیش‌نگری منفی، ۱۷۹
(?=): تأیید پیش‌نگری مثبت، ۱۷۹
[] عملگر، ۲۷، ۳۹، ۱۱۲، ۱۱۵
[] مقدار خصلت، ۹۶
{ } آکلا، ۳۹، ۶۶، ۸۴، ۹۲، ۱۳۰
* آوند همه‌کاره، ۲۰۷
* عملگر ضرب، ۱۹، ۴۳، ۴۶

درخواست‌های اسکریپتی، ۲۰۰
 درخواست‌های با مبدأ متفاوت، ۲۵۷
 دریافت پاسخ، ۲۵۴
 رویداد progress، ۲۵۶
 هدر USER-AGENT، ۱۹۳
 i (انطباق غیرحساس به حروف کوچک و بزرگ)،
 ۱۸۰
 location.hash، ۱۹۱
 lvalue، ۴۵
 m حالت چندسطری، ۱۸۰
 window.onhashchange، ۱۹۱
 window.onpopstate، ۱۹۲
 XML، ۲۰۳

— آ —

آرایه‌ها، ۲۸، ۳۸، ۹۳، ۱۱۱، ۱۱۲، ۱۱۲، ۱۱۳، ۱۱۴،
 ۱۱۴، ۱۱۵، ۱۱۶، ۱۲۰، ۱۱۶، ۱۱۷، ۱۱۷، ۱۱۸، ۱۱۸،
 ۱۱۹، ۱۲۵، ۱۲۵، ۱۲۶، ۱۴۱
 اشیای شبه‌آرایه، ۱۲۶، ۱۴۱
 انواع، ۱۲۵
 ایجاد، ۱۱۲
 تابع Array.isArray()، ۱۲۵
 تابع Array()، ۲۸، ۹۳، ۱۱۲
 تکرار، ۱۱۴

چندبعدی، ۱۱۵

خصلت length، ۱۱۴

متد Array.concat()، ۱۱۸

متد Array.join()، ۱۱۶

متد Array.reverse()، ۱۱۷

متد Array.slice()، ۱۱۸

متد Array.sort()، ۱۱۷

متد Array.splice()، ۱۱۹

متدها، ۱۱۶، ۱۲۰

مقادیر لفظی، ۱۱۳

مقدار دهنده‌ی اولیه‌ی «آرایه‌ی لفظی»، ۳۸

آرایه‌های با پایه‌ی صفر، ۱۱۱

آرایه‌های بی‌نوع، ۱۱۱

آرایه‌های پویا، ۱۱۱

آکلاذ {}, ۳۹، ۴۰، ۴۲، ۴۶، ۶۹، ۸۴، ۹۲، ۱۳۰

اشیای لفظی و، ۹۲

بلوک دستورالعمل، ۶۶، ۸۴

۷ سور عمومی، ۱۲۳

۳ سور وجودی، ۱۲۳

+ به هم پیوستن رشته‌ها، ۴۳

+ تبدیل به عدد، ۴۳

+ عملگر به علاوه‌ی یگانی، ۳۱، ۴۷

+ عملگر جمع، ۱۹، ۴۶

+ نویسه‌ی تکرار، ۱۷۵

++ افزایش واحدی پیشوندی/پسوندی، ۴۲

++ عملگر افزایش واحد، ۴۲، ۴۷، ۶۵

=+ عملگر، ۵۷، ۲۱۵

= تخصیص مقدار به متغیر/خصلت، ۴۴، ۵۶

== عملگر تساوی، ۲۷، ۳۰

=== عملگر تساوی غیراکید، ۴۳، ۵۱

=== عملکرد تساوی اکید، ۲۷، ۴۳، ۵۱

| جدا کننده، ۱۷۶

| عملگر OR بیتی، ۴۳، ۴۹

| گزینش، ۱۷۷

|| عملگر منطقی، ۴۳، ۵۵، ۱۴۱

~ عملگر NOT بیتی، ۴۹

~ معکوس کردن بیت‌ها، ۴۳

\$ انطباق با انتها، ۱۷۹

\$ شناسه، ۱۴

— A-Z —

arguments، ۱۲۹

CORS (تشریک منابع با مبدأ متفاوت)، ۲۵۸

CSS (شیوه‌نامه‌های آبخاری)، ۲۲۰، ۲۲۴

DOM (مدل شیئی سند)، ۲۰۱، ۲۰۴، ۲۰۴، ۲۴۶

g (انطباق سراسری)، ۱۸۰

HTML، ۱۹۴، ۲۰۱، ۲۰۴، ۲۰۳، ۲۰۳، ۲۰۶، ۲۱۳،

۲۱۳، ۲۳۵، ۲۳۸

HTMLElement، ۱۹۴، ۲۰۳، ۲۱۳

خصلت htmlFor، ۲۱۳

رویدادپردازهای HTML5، ۲۳۵، ۲۳۸

غیرحساس به حروف کوچک و بزرگ، ۲۰۶

محتوای عنصر به صورت، ۲۱۵

مرور DOM، ۲۰۱، ۲۰۴

نوع HTMLDocument، ۲۰۳

HTTP، ۱۹۳، ۱۹۹، ۲۰۰، ۲۵۴، ۲۵۶، ۲۵۷، ۲۵۸، ۲۶۱

JSONP، ۲۵۸، ۲۶۱

پروتکل http، ۱۹۹

- توابع و، ۴۰، ۶۹، ۱۳۰
مقدار دهنده‌ی اولیه‌ی شیء، ۳۹، ۴۲
آیفون/آیپد اپل، ۲۳۸
- ۱ —
- ابركلاس‌ها، ۱۶۶
اتصالات دوطرفه‌ی نوع سوکتی، ۲۶۴
اثر جانبی، ۴۵
اخلاف یک گره، ۲۰۲
ارائه‌های ورودی/خروجی ناهمگام، ۳۳۷
استاندارد IEEE 754، ۱۸
استثناها، ۸۲
اسکرول کردن، ۲۲۴، ۲۲۷
اسکرول کردن، ۲۲۴، ۲۲۷
اسلاف یک گره، ۲۰۲
اشاره‌ی دور کردن دو انگشت، ۳۳۹
اشاره‌ی نزدیک کردن دو انگشت، ۳۳۹
اشیای لفظی، ۹۲
اطلاعات صفحه‌ی نمایش، ۱۹۲
اطلاعات مرورگر و صفحه‌ی نمایش، ۱۹۲
اعداد، ۱۸، ۲۱
اعداد لفظی، ۱۸
اعضای کلاس، ۱۶۲
اعلام متغیرها، ۱۷، ۳۳
اکما اسکریپت ۳، ۱۵، ۱۳۴، ۱۳۸، ۱۴۹
اکما اسکریپت ۵، ۱۵، ۲۳، ۳۴، ۷۸، ۸۷، ۱۰۱، ۱۰۱
۱۰۲، ۱۰۴، ۱۱۲، ۱۲۱، ۱۲۵، ۱۲۷، ۱۳۴، ۱۳۸، ۱۴۹
Object.getOwnPropertyNames()، ۱۰۱
Object.keys()، ۱۰۱
تخصیص مقدار به متغیر اعلام نشده، ۳۴
حلقه‌ی for/in، ۷۸
رشته‌ها به عنوان آرایه، ۱۱۲، ۱۲۷
فراخوانی تابع، ۱۳۴
فرمان "use strict"، ۸۷
کچ خط وارون در، ۲۳
متد (bind)، ۱۴۹
متدهای آرایه، ۱۲۱، ۱۲۵
متدهای مقدارگیر و مقدارگذار، ۱۰۲
مقادیر null و defined، ۱۳۸
مقدارگیری و مقدارگذاری صفت‌های خصلت، ۱۰۴
- امنیت، ۲۶۶
امنیت و اسکریپت‌ها، ۲۶۲
انتخاب عناصر بر اساس ID، ۲۰۴
انتخاب عناصر بر اساس انتخابگرهای CSS، ۲۰۹
انتخاب عناصر بر اساس کلاس CSS، ۲۰۸
انتخاب عناصر بر اساس نام، ۲۰۵
انتخاب عناصر بر اساس نوع، ۲۰۶
انتخابگرهای CSS، ۲۰۹
انحراف اسکرول، ۲۲۵
اندیس، ۱۱۱
انشعاب، ۷۰
انطباق الگو، ۱۸۰، ۱۸۳
خصلت‌ها و متدهای RegExp، ۱۸۳
متدهای رشته برای، ۱۸۰
انطباق سراسری (g)، ۱۸۰
انطباق غیرحساس به حروف کوچک و بزرگ و (i)، ۱۸۰
انواع، ۱۷، ۱۷، ۲۸، ۳۲
تبدیل خودکار، ۱۷
تبدیل‌ها، ۲۸، ۳۲
انواع بدوی، ۱۷
انواع شیء، ۱۷، ۵۳
اینترنت اکسپلورر ۸، مایکروسافت، ۲۷۱
- ب —
- بازنویسی متدها، ۱۶۶
بدافزار، ۲۶۶
بدنه‌ی پاسخ، ۲۵۴
بر کشیدن، ۳۵
بر گرداندن مقدار، ۳۷، ۱۲۹، ۲۴۵
برادرِ گره، ۲۰۲
برنامه‌نویسی مبتنی بر رویداد، ۱۸۷
برنامه‌های وب آفلاین، ۲۳۷
یستارها، ۱۳۰، ۱۴۳، ۱۴۸
بلوک finally، ۸۴
بلوک‌های دستورالعمل، ۶۶
بند catch، ۸۳
پولی، ۱۷
به هم پیوستن، ۲۳
- پ —
- پارامترها، ۱۲۹، ۱۴۰

تابع (isFinite) .isFinite	۲۱	پارامترهای اختیاری، ۱۴۰
تابع (isFrozen) .isFrozen	۱۱۰	پاسخ "200 OK"، ۲۵۴، ۲۵۶
تابع (isNaN) .isNaN	۲۸، ۲۱	پاسخ "404 Not Found"، ۲۵۴، ۲۵۶
تابع (isSealed) .isSealed	۱۱۰	پایان سطر، ۱۶
تابع (JSON.parse) .JSON.parse	۲۷۷، ۲۵۹، ۲۵۵، ۱۰۱	پدگذاری، ۲۲۶
تابع (JSON.stringify) .JSON.stringify	۱۰۱	پرانتز، ۱۳۰، ۱۷۶
تابع (keys) .keys	۱۰۱	پرچم‌ها، ۱۷۹
تابع (Math.max) .Math.max	۱۴۱، ۱۳۹	پنجره‌ها و فریم‌های متعدد، ۱۹۵، ۲۰۰
تابع (Object) .Object	۲۸، ۳۰، ۹۳، ۹۲، ۹۳، ۹۴، ۹۷، ۱۰۰، ۱۰۱، ۱۰۱، ۱۰۶، ۱۰۷، ۱۰۷، ۱۰۷، ۱۵۴، ۱۶۶، ۱۹۰	پیش‌نمونه‌ها، ۹۳، ۹۴، ۱۴۸، ۱۵۴، ۱۶۳
تابع (Object.defineProperty) .Object.defineProperty	۱۰۶	زنجیره، ۹۴
تابع (Object.getOwnPropertyNames) .Object.getOwnPropertyNames	۱۰۱	شیء، ۱۴۸، ۱۶۳
تابع (Object.keys) .Object.keys	۱۰۱	و کلاس‌ها، ۱۵۴
تابع (Object.prototype) .Object.prototype	۱۰۰، ۹۷، ۹۳	پیش‌نویس استاندارد XHR2، ۲۵۶
متد (Object.create) .Object.create	۱۵۴، ۱۰۷، ۹۴، ۹۲	— ت —
متد (Object.defineProperties) .Object.defineProperties	۱۰۷	تابع «کارخانه‌ای»، ۱۵۵
متد (Object.defineProperty) .Object.defineProperty	۱۰۷	تابع (Boolean) .Boolean
متد (Object.reload) .Object.reload	۱۹۰	تابع (checkscope) .checkscope
تابع (parseFloat) .parseFloat	۳۲	تابع (classof) .classof
تابع (parseInt) .parseInt	۳۲، ۲۸	تابع (constfuncs) .constfuncs
تابع (preventExtensions) .preventExtensions	۱۰۹	تابع (count) .count
تابع (range) .range	۱۵۶	تابع (counter) .counter
تابع (RegExp) .RegExp	۲۸، ۹۳، ۱۷۱، ۱۸۱	تابع (Date) .Date
تابع (reset) .reset	۱۴۶	تابع (decodeURIComponent) .decodeURIComponent
تابع (square) .square	۱۳۳	تابع (defineClass) .defineClass
تابع (String) .String	۲۸، ۳۰	تابع (defineSubclass) .defineSubclass
تابع (textContent) .textContent	۲۱۶	تابع (encodeURIComponent) .encodeURIComponent
تابع (trace) .trace	۱۴۰	تابع (eval) .eval
تابع (uniqueInteger) .uniqueInteger	۱۴۵	تابع (freeze) .freeze
تبدیل نوع صریح، ۳۰		تابع (getCookie) .getCookie
تعامل بین پنجره‌ها، ۱۹۸		تابع (getElements) .getElements
تعیین مکان انطباق، ۱۷۸		تابع (getJSONP) .getJSONP
تقسیم بر صفر، ۲۰		تابع (getOwnPropertyDescriptor) .getOwnPropertyDescriptor
تکرار، ۱۷۴		تابع (getOwnPropertyNames) .getOwnPropertyNames
تکرار آزمندانه، ۱۷۵		تابع (getText) .getText
تکرار نآزمندانه، ۱۷۵		تابع (grabAttention) .grabAttention
توابع، ۱۸، ۳۴، ۶۸، ۱۲۹، ۱۳۰، ۱۳۳، ۱۳۴، ۱۴۰، ۱۴۰		تابع (handleResponse) .handleResponse
۱۴۲		تابع (hypotenuse) .hypotenuse
آوندها و پارامترها، ۱۴۰		تابع (inherit) .inherit
اعلام، ۶۸		تابع (isArray) .isArray
به عنوان فضای نام، ۱۴۲		تابع (isExtensible) .isExtensible

- تعریف، ۱۳۰
 تو در تو، ۱۳۳
 فراخوانی، ۱۳۴، ۱۴۰
 قلمرو، ۱۸، ۳۴
 توابع، ۱۴۲
 توابع با چندگانگی متغیر، ۱۴۲
 توابع چندگانی، ۱۴۲
 توالی‌های گریز، ۲۲
 توضیحات، ۱۳
- خ —
 خصلت abs، شیء Math، ۲۰
 خصلت altKey، رویدادهای ماوس، ۲۳۳، ۲۳۵
 خصلت appName، شیء Navigator، ۱۹۲
 خصلت appVersion، شیء Navigator، ۱۹۲
 خصلت availHeight، شیء Window، ۱۹۳
 خصلت availWidth، شیء Window، ۱۹۳
 خصلت borderLeftWidth، ۲۲۱
 خصلت ceil، شیء Math، ۱۹
 خصلت changedTouches، شیء رویداد، ۲۳۹
 خصلت charCode، شیء رویداد keypress، ۲۳۵
 خصلت childElementCount، ۲۱۳
 خصلت childNodes، شیء Node، ۲۱۱
 خصلت children، اشیای Element، ۲۱۲
 خصلت classList، ۲۲۳
 خصلت className، ۲۰۸، ۲۱۴، ۲۲۳
 خصلت clientX، رویدادهای ماوس، ۲۳۳
 خصلت clientY، رویدادهای ماوس، ۲۳۳
 خصلت color، ۲۲۰
 خصلت contentWindow، شیء Window، ۱۹۷
 خصلت cookie، شیء Document، ۲۷۲
 خصلت ctrlKey، رویدادهای ماوس، ۲۳۳، ۲۳۵
 خصلت data، اشیای رویداد، ۱۰۲، ۲۳۷، ۲۶۳
 خصلت dataTransfer، ۲۳۶
 خصلت detail، رویدادهای ماوس، ۲۳۳
 خصلت document، شیء Window، ۱۸۹
 خصلت documentElement، کلاس Document، ۲۰۷
 خصلت E، شیء Math، ۲۰
 خصلت exp، شیء Math، ۲۰
 خصلت firstChild، شیء Node، ۲۱۱
 خصلت firstElementChild، ۲۱۲
 خصلت float، ۲۲۱
 خصلت floor، شیء Math، ۲۰
 خصلت font-size، ۲۲۰
 خصلت font-weight، ۲۲۰
 خصلت forms، شیء HTMLCollection، ۲۰۷
 خصلت frameElement، شیء Window، ۱۹۷
 خصلت frames، شیء Window، ۱۹۵، ۲۰۰
 خصلت geolocation، شیء Navigator، ۱۹۳
- ث —
 ثبت رویدادپرداز، ۲۳۹، ۲۴۳
- ج —
 جابجایی چپ (<<)، ۴۳، ۴۹
 جاسوس‌افزار، ۲۶۶
 جاوا اسکریپت سمت مشتری، ۱۸۵، ۱۸۷، ۱۸۸، ۱۸۹
 ۱۸۹، ۱۹۱، ۱۹۲، ۱۹۳، ۱۹۴، ۱۹۵، ۲۰۰
 اطلاعات مرورگر و صفحه‌ی نمایش، ۱۹۲
 برنامه‌ریزی مبتنی بر رویداد، ۱۸۷
 پنجره‌ها و فریم‌های متعدد، ۱۹۵، ۲۰۰
 جای‌گذاری در HTML، ۱۸۵
 زمان‌سنج‌ها، ۱۸۹
 سابقه‌ی مرورگری، ۱۹۱
 شیء Window، ۱۸۸
 عناصر سند، ۱۹۴
 کادرهای گفتگو، ۱۹۳
 نشانی و ناوبری مرورگر، ۱۸۹
 جای‌گذاری جاوا اسکریپت در HTML، ۱۸۵
- چ —
 چندگانگی، ۴۵
- ح —
 حالت اکید، ۱۴، ۳۴، ۸۸، ۱۳۸
 حالت چندسطری (m)، ۱۸۰
 حریم خصوصی، ۲۶۶
 حساس بودن به حروف کوچک و بزرگ، ۱۳
 حلقه‌ی do/while، ۶۴، ۷۵
 حلقه‌ی for، ۳۳، ۶۴، ۱۱۴
 حلقه‌ی for/in، ۳۳، ۶۴، ۷۷، ۱۰۰
 حلقه‌ی while، ۶۶، ۶۹

۲۱۳، onclick، شیء event	۱۸۳، RegExp، شیء global
۲۴۲، ۲۴۰، onclick، شیء event	۱۹۰، Location، شیء hash
۱۹۳، Navigator، شیء onLine	۱۹۳، Window، شیء height
۲۵۷، ۲۴۰، onload، شیء event	۱۹۱، Window، شیء history
۲۴۰، onmouseover، شیء event	۱۹۰، Location، شیء host
۲۵۷، onprogress، شیء	۱۹۰، Location، شیء hostname
۲۵۵، XMLHttpRequest، شیء	۲۱۳، ۲۰۷، ۱۹۰، Location، شیء href
۲۷۱، Window، شیء onstorage	۱۸۳، RegExp، شیء ignoreCase
۲۳۹، Window، شیء orientation	۲۰۷، HTMLCollection، شیء images
۲۲۳، overflow، شیء	۱۸۳، RegExp، شیء index
۲۲۵، Window، شیء pageXOffset	۲۲۵، Window، شیء innerHeight
۲۲۵، Window، شیء pageYOffset	۲۱۵، Window، شیء innerHTML
۱۹۶، Window، شیء parent	۲۲۵، innerWidth، شیء
۲۱۱، Node، شیء parentNode	۱۸۳، RegExp، شیء input
۱۹۰، Location، شیء pathname	۲۷۱، event، شیء key
۲۰، Math، شیء PI	۲۳۵، keyCode، شیء
۱۹۳، Navigator، شیء platform	۲۱۱، Node، شیء lastChild
۱۹۰، Location، شیء port	۲۱۲، lastElementChild، شیء
۱۹، Math، شیء pow	۱۸۳، RegExp، شیء lastIndex
۲۱۲، previousElementSibling، شیء	۱۴۸، ۱۲۶، ۱۱۴، ۱۱۱، ۲۳، length، شیء
۲۱۱، Node، شیء previousSibling	۲۵۷، Event، شیء lengthComputable
۱۹۰، Location، شیء protocol	۲۰۷، HTMLCollection، شیء links
۱۵۹، ۱۵۷، ۱۴۸، ۱۳۷، ۹۳، ۹۳، prototype، شیء	۲۰، Math، LN10، شیء
۲۰، Math، شیء random	۲۰، Math، LN2، شیء
XMLHttpRequest، شیء readyState	۲۵۶، Event، شیء loaded
۲۵۶، ۲۵۴	۲۷۲، ۲۶۶، Window، شیء localStorage
۲۳۴، event، شیء relatedTarget	۱۸۹، ۱۸۸، Window، شیء location
XMLHttpRequest، شیء responseText	۲۰، Math، dog، شیء
۲۵۸، ۲۵۴	۲۰، Math، max، شیء
۲۳۸، event، شیء rotation	۸۳، Error، message، شیء
۱۹، Math، شیء round	۲۳۵، ۲۳۳، metaKey، شیء
۲۳۸، event، شیء scale	۲۰، Math، amin، شیء
۱۹۳، ۱۹۲، Window، شیء screen	۱۸۳، RegExp، شیء multiline
۱۹۰، Location، شیء search	۸۳، Error، شیء name
۲۶۶، Window، شیء sessionStorage	۱۹۷، Window، شیء name
۲۷۲	۱۹۲، Window، شیء navigator
۲۳۵، ۲۳۳، shiftKey، شیء	۲۷۱، event، شیء newValue
۲۰، Math، شیء sin	۲۱۲، nextElementSibling، شیء
۱۸۳، RegExp، شیء source	۲۱۱، Node، شیء nextSibling
۲۰، Math، شیء sqrt	۲۷۱، event، شیء oldValue
	۲۴۰، event، شیء onchange

خصلت‌های میانبر، ۲۰۷

خواندن کوکی‌ها، ۲۷۶

— د —

دستورالعمل break، ۶۴، ۶۴، ۸۰

دستورالعمل case، ۶۴

دستورالعمل continue، ۶۴، ۸۱

دستورالعمل debugger، ۶۴، ۸۷

دستورالعمل default، ۶۴

دستورالعمل else if، ۷۱

دستورالعمل for، ۶۴، ۷۶

دستورالعمل function، ۶۴

دستورالعمل if، ۶۳، ۶۹، ۷۰

دستورالعمل if/else، ۶۴

دستورالعمل label، ۶۴

دستورالعمل return، ۴۱، ۶۴، ۶۵، ۸۲

دستورالعمل switch، ۶۳، ۶۵، ۷۲

دستورالعمل throw، ۶۴، ۶۵، ۸۲

دستورالعمل try، ۶۵

دستورالعمل try/catch/finally، ۸۳

دستورالعمل use strict، ۶۵، ۸۷

دستورالعمل var، ۶۵، ۶۷

دستورالعمل while، ۶۴، ۶۵، ۷۵

دستورالعمل with، ۶۵، ۸۶، ۲۴۵

دستورالعمل خالی، ۶۴، ۶۷

دستورالعمل‌ها، ۱۵، ۱۶، ۶۳، ۶۴، ۶۵، ۶۶، ۶۷، ۷۰

۷۴، ۷۹، ۷۹، ۸۵، ۸۶، ۸۹

اعلام، ۶۷

پرش، ۷۹، ۸۵

جدا کننده (ویرگول نقطه) (:؛)، ۱۵

حلقه، ۷۴، ۷۹

شرطی، ۷۰، ۷۴

عبارت، ۶۵

قواعد ختم، ۱۶

مرکب و خالی، ۶۴

نحو، ۶۴

دستورالعمل‌های اعلام، ۶۷

دستورالعمل‌های برجسب‌دار، ۷۹

دستورالعمل‌های پرش، ۷۹، ۸۵

دستورالعمل‌های حلقه، ۷۴، ۷۹

دستورالعمل‌های شرطی، ۶۳، ۷۰، ۷۴

خصلت HTMLElement، src، ۲۰۰، ۲۱۳

خصلت status، شیء XMLHttpRequest، ۲۵۴

خصلت statusText، شیء XMLHttpRequest، ۲۵۴

۲۵۴

خصلت storageArea، شیء event، ۲۷۱

خصلت style، شیء Element، ۲۲۰

خصلت target، شیء event، ۲۳۰، ۲۳۶

خصلت textContent، شیء Node، ۲۱۶

خصلت timestamp، شیء Event، ۲۵۶

خصلت top، شیء Window، ۱۹۶

خصلت total، شیء Event، ۲۵۶

خصلت type، شیء event، ۲۳۰، ۲۵۶، ۲۶۳

خصلت upload، شیء XMLHttpRequest، ۲۵۷

خصلت url، شیء event، ۲۷۲

خصلت userAgent، شیء Navigator، ۱۹۳

خصلت width، شیء Window، ۱۹۳

خصلت window، شیء Window، ۲۸، ۱۸۸

خصلت سراسری Infinity، ۲۷

خصلت‌ها، ۳۹، ۴۳، ۵۳، ۶۴، ۹۱، ۹۶، ۹۶، ۹۸، ۹۸، ۹۸، ۱۰۰، ۱۰۲، ۱۰۲، ۱۰۴، ۱۰۵، ۱۰۹، ۱۴۸، ۱۵۱، ۱۴۸

۱۵۹، ۱۶۰، ۱۹۴، ۲۰۷

آزمایش، ۹۸

برشماری، ۶۴، ۱۰۰

پیش‌فون، ۱۴۸، ۱۵۹

تابع، ۱۴۸، ۱۵۱

توارث، ۹۶

توصیف کننده، ۱۰۵

حذف، ۴۳، ۹۸

داده‌ها، ۱۰۲

دسترسی، ۳۹، ۱۰۲

سازنده، ۱۶۰

صفت‌ها، ۱۰۴

عناصر سند به عنوان، ۱۹۴

گسترش‌پذیری و، ۱۰۹

متد ()propertyIsEnumerable، ۹۸

مقدارگیری و مقدارگذاری، ۹۶

میانبر، ۲۰۷

وجود (عملگر in)، ۵۳

خصلت‌های Math، ۱۹

خصلت‌های تجزیه‌ی URL، ۱۹۰

خصلت‌های دستیاب، ۱۰۲

— ش —

شانزدهگانی، ۱۸
شبکه، ۲۶۹، ۲۵۸، ۲۶۱، ۲۶۲، ۲۶۳
HTTP از طریق <script>: JSONP، ۲۵۸، ۲۶۱
استفاده از XMLHttpRequest، ۲۴۹، ۲۵۸
رویدادهای فرستاده شده از سرور، ۲۶۲
وب سوکت ها، ۲۶۳

شناسه ها، ۱۳، ۱۴
شناسه ی window، ۱۸۸
شناسه ی زیرخط (_)، ۱۴
شناسه ی علامت دلار (\$)، ۱۴
شیء Arguments، ۱۴۱
شیء CSSStyleDeclaration، ۲۲۰
شیء Link، ۱۹۰
شیء Window، ۱۸۸، ۱۹۷
شیء XMLHttpRequest، ۲۰۰، ۲۴۶، ۲۵۱، ۲۵۴
شیء سراسری، ۲۷
شیء نمونه، ۱۶۳
شیوه نامه های آبخاری، ۲۲۰، ۲۲۴
شیوه ی درون خطی HTML، ۲۲۱
شیوه ی عنصر، ۲۲۰، ۲۲۴

— ص —

صفت class، ۲۰۸، ۲۲۲
صفت configurable، ۱۰۵
صفت dir، ۲۱۳
صفت domain، ۲۷۳
صفت enumerable، ۹۹، ۱۰۵
صفت get، ۱۰۵
صفت id، ۱۹۴، ۱۹۷، ۲۰۴، ۲۱۳
صفت lang، ۲۱۳
صفت max-age، ۲۷۳
صفت name، ۱۹۷، ۲۰۵
صفت path، ۲۷۳
صفت prototype، ۱۰۸
صفت secure، ۲۷۵
صفت set، ۱۰۵
صفت title، ۲۱۳
صفت value، ۱۰۵
صفت writable، ۱۰۵
صفت ها، ۲۱۳

رویداد پردازهای فرم، ۲۳۱
رویداد پردازهای کلید، ۲۳۵
رویداد پردازهای ماوس، ۲۳۳
رویدادهای فرستاده شده از سرور، ۲۳۷
رویدادهای کشنداز، ۲۳۶
رویه ها، ۱۲۹

— ز —

زمان سنج ها، ۱۸۹
زنجیره ی قلمرو تغییر یافته، ۲۴۴
زیرروال ها، ۱۲۹
زیرریز، ۲۰
زیرکلاس ها، ۱۶۶

— س —

سابقه ی مرورگری، ۱۹۱
ساختار درختی، ۲۰۲
ساختار و پیمایش سند، ۲۱۱
ساختارهای کنترلی، ۶۳
ساز و کار اعتبارسنجی فرم، ۲۳۷
ساز و کار مدیریت سابقه، ۲۳۶
سازنده ها، ۹۳، ۱۴۸، ۱۵۰، ۱۵۷، ۱۵۹، ۱۶۰
Function()، ۱۴۸، ۱۵۰
توکار، ۹۳
خصلت، ۱۶۰
کلاس ها و، ۱۵۷، ۱۵۹
سازنده ی EventSource()، ۲۶۳
سازنده ی Function()، ۱۴۸، ۱۵۰
سازنده ی Range()، ۱۵۷، ۱۶۱
سازنده ی WebSocket()، ۲۶۴
ساعت دیجیتال، ۱۸۵
سافاری، ۲۳۸
سرریز، ۲۰
سریالیزه کردن، ۱۰۱
سلسله مراتب کلاس، ۲۰۳
سلسله مراتب نوع، ۲۰۳
سور عمومی، ۷، ۱۲۳
سور وجودی E، ۱۲۳
سیاست مبدأ، ۲۳۷
سیاست مبدأ، ۲۳۷
سیاست مبدأ یکسان، ۱۹۹

صفت‌های شیء، ۱۰۸، ۱۰۸، ۱۰۹
صفت class، ۱۰۸
صفت extensible، ۱۰۹
صفت prototype، ۱۰۸
صفر، تقسیم بر، ۲۰

— ط —

طول رشته، ۲۱
طول عمر، ۲۶۷، ۲۷۳

— ع —

عبارت آزمایش، ۷۶
عبارت افزایش واحد، ۷۶
عبارت مقداردهی اولیه، ۷۶
عبارت‌ها، ۳۷، ۳۸، ۳۸، ۳۹، ۴۰، ۴۱، ۴۱، ۴۲، ۵۰، ۵۳

۵۴، ۵۶، ۵۷

ارزیابی، ۵۷

اولیه، ۳۸

ایجاد شیء، ۴۲

تخصیص، ۵۶

تعریف، ۳۷

تعریف تابع، ۴۰

دسترسى خصلت، ۳۹

رابطه‌ای، ۵۰، ۵۳

فراخوانی، ۴۱

متد، ۴۱

مقدار دهنده‌های اولیه، ۳۸

منطقی، ۵۴، ۵۶

عبارت‌های ارزیابی، ۳۷، ۵۷

عبارت‌های اولیه، ۳۸

عبارت‌های ایجاد شیء، ۴۲

عبارت‌های تخصیص، ۵۶

عبارت‌های تعریف تابع، ۴۰

عبارت‌های قاعده‌مند، ۱۷۱، ۱۷۱، ۱۷۴، ۱۷۴، ۱۷۶

۱۷۸، ۱۷۹، ۱۸۰، ۱۸۴

انطباق الگوها به، ۱۸۰، ۱۸۴

پرچم‌ها، ۱۷۹

تعیین مکان انطباق، ۱۷۸

تکرار در، ۱۷۴

توصیف الگوها به، ۱۷۱

دسته‌های نویسه‌ها، ۱۷۴

گزينش، گروه‌بندی، و اشاره‌ها، ۱۷۶

عبارت‌های متد، ۴۱

عبارت‌های منطقی، ۵۴، ۵۶

عملگر delete، ۴۳، ۴۵، ۶۱، ۶۶

عملگر in، ۴۳، ۵۳، ۹۹

عملگر instanceof، ۴۳

عملگر typeof، ۴۳، ۴۴، ۶۰

عملگر void، ۴۳، ۶۱

عملگر سه‌گانی (?:)، ۴۵

عملگر کروشه []، ۹۶

عملگر نقطه (.)، ۲۷، ۳۹، ۹۶، ۱۷۴

عملگر ویرگول (،)، ۴۴، ۶۱، ۱۳۰، ۲۷۵

عملگرها، ۳۷، ۴۲، ۴۶، ۴۶، ۵۰، ۵۰، ۵۳، ۵۹

حسابی، ۴۶، ۵۰

رابطه‌ای، ۵۰، ۵۳

عملگرهای حسابی، ۴۶، ۵۰

عملگرهای دوگانی، ۴۵

عملگرهای یگانی، ۴۵، ۴۷

عملوندها، ۳۷

عمل‌های پیش‌فرض، ۲۳۱

عناصر، ۱۱۱

عناصر سند، ۱۹۴، ۲۰۴، ۲۰۵، ۲۰۶، ۲۰۸، ۲۰۹

انتخاب بر اساس ID، ۲۰۴

انتخاب بر اساس انتخابگرهای CSS، ۲۰۹

انتخاب بر اساس کلاس CSS، ۲۰۸

انتخاب بر اساس نام، ۲۰۵

انتخاب بر اساس نوع، ۲۰۶

عنصر <a>، ۲۱۳، ۲۳۰

عنصر <audio>، ۲۳۶

عنصر <body>، ۲۰۷، ۲۴۱

عنصر <button>، ۱۹۴، ۲۴۲

عنصر <div>، ۲۰۸

عنصر <form>، ۲۰۶، ۲۱۳، ۲۴۵، ۲۴۶، ۲۵۷

عنصر <frame>، ۱۹۵

عنصر <frameset>، ۱۹۵

عنصر <head>، ۲۰۷

عنصر <iframe>، ۱۹۱، ۱۹۶، ۲۰۵، ۲۲۴، ۲۵۷، ۲۶۹

عنصر ، ۲۰۵، ۲۱۳

عنصر <input>، ۲۱۴

عنصر <label>، ۲۱۳

عنصر <p>، ۲۰۷، ۲۱۵

عنصر <script>، ۱۸۵، ۲۰۰، ۲۴۹، ۲۵۸

عنصر ، ۲۰۶

عنصر <video>، ۲۳۶

— ف —

فایل‌های محلی و XMLHttpRequest، ۲۵۱

فرا خواندن تابع، ۱۶

فراخوانی، ۴۱، ۱۲۹، ۱۳۴، ۱۳۶، ۱۳۷، ۱۳۸، ۲۴۳، ۲۴۸

توابع، ۱۳۴

رویدادپرداز، ۲۴۳، ۲۴۸

سازنده، ۱۳۷

عبارت‌های، ۴۱

غیرمستقیم، ۱۳۸

متدها، ۱۳۴

محیط، ۱۲۹

فراخوانی تابع، ۶۶

فرزند گره، ۲۰۲

فرمان‌ها، ۸۷

فریم‌ها، ۱۹۵، ۲۰۰

فضای سفید، ۲۷۵

فضای نام، توابع به عنوان، ۱۴۲

فیلدهای کلاس، ۱۶۲

فیلدهای نمونه، ۱۶۲

— ق —

قلمرو، ۲۶۷، ۲۷۳

قلمرو بلوک، ۳۴

قلمرو رویدادپردازها، ۲۴۴

قلمرو سراسری، ۳۴، ۱۴۵

قلمرو لغوی، ۱۷، ۱۴۳

قلمرو متغیر، ۳۴

قلمرو محلی، ۱۴۴

— ک —

کارهای گفتگو، ۱۹۳

کد اکید، ۸۸

کدگذاری لاتین-۱، ۲۲، ۱۷۲

کلاس range، ۱۵۷

کلاس‌ها، ۱۵۳، ۱۵۴، ۱۵۷، ۱۶۴، ۱۶۶، ۱۶۸، ۱۷۴

افزودن، ۱۶۸

غیرقابل تغییر، ۱۶۴

گسترش دادن، ۱۶۶

و پیش‌نمونه‌ها، ۱۵۴

و سازنده‌ها، ۱۵۷

کلاس‌های CSS، ۲۰۴، ۲۰۸

کلاس‌های تغییرناپذیر، ۱۶۴

کلاس‌های سَبَکِ جاوا، ۱۶۲

کلیدواژه‌ی break، ۱۶، ۷۳

کلیدواژه‌ی catch، ۸۵

کلیدواژه‌ی continue، ۱۶

کلیدواژه‌ی for، ۷۷

کلیدواژه‌ی function، ۶۸، ۱۰۳، ۱۴۳، ۱۵۰، ۱۹۸

۲۴۰

کلیدواژه‌ی new، ۴۲، ۹۳، ۱۳۷، ۱۵۷

کلیدواژه‌ی null، ۲۶

کلیدواژه‌ی return، ۱۶، ۱۳۸

کلیدواژه‌ی this، ۲۸، ۴۲، ۱۲۱، ۱۳۴، ۱۴۷، ۲۴۴

کلیدواژه‌ی undefined، ۲۶

کلیدواژه‌ی var، ۱۷، ۳۳، ۱۹۸

کوکی‌ها، ۲۶۶، ۲۷۲، ۲۷۸

— گ —

گرد کردن، ۲۱

گره‌ها، ۲۰۱، ۲۰۵، ۲۱۱، ۲۱۱، ۲۱۱، ۲۱۶، ۲۱۷

۲۱۹

ایجاد، درج، و حذف، ۲۱۷، ۲۱۹

خصلت nodeName، شیء Node، ۲۱۱

خصلت nodeType، شیء Node، ۲۱۱

خصلت nodeValue، شیء Node، ۲۱۱، ۲۱۶

شیء NodeList، ۲۰۵، ۲۱۰

گره‌های سند، ۲۱۷، ۲۱۹

گره‌های متن، محتوای عنصر به صورت، ۲۱۶

گسترش دادن کلاس‌ها، ۱۶۸

— ل —

لنگر، ۱۷۸

لیست آوردها با طول متغیر، ۱۴۱

— م —

مبدأ سند، ۱۹۹

مبنای ۱۰، ۱۸

متد add()، ۲۲۳

متد addEventListener()، ۲۴۲، ۲۴۴، ۲۴۶، ۲۵۵

۲۷۱، ۲۵۷

متد alert()، ۸۵، ۱۸۸، ۱۹۳

متد (appendChild) ۲۱۸	متد (join) ۱۱۶
متد (apply) ۱۳۴، ۱۳۸، ۱۴۸	متد (key) ۲۷۰
متد (attachEvent) ۲۷۱	متد (lastIndexOf) ۱۲۵
متد (back) ۱۹۱	متد (map) ۱۲۲
متد (bind) ۱۴۹	متد (match) ۱۸۱، ۱۸۳
متد (call) ۱۰۹، ۱۳۴، ۱۳۸، ۱۴۸، ۱۶۷	متد (open) ۲۵۱
متد (charAt) ۲۴، ۱۲۷	متد (pop) ۱۲۰
متد (clear) ۲۷۰	متد (preventDefault) ۲۴۶
متد (clearTimeout) ۱۸۹	متد (prompt) ۸۵، ۱۹۴
متد (cloneNode) ۲۱۸	متد (push) ۱۲۰
متد (close) ۲۶۴	متد (pushState) ۱۹۲
متد (concat) ۱۱۸	متد (querySelectorAll) ۲۱۰
متد (confirm) ۱۹۴	متد (reduce) ۱۲۳
متد (contains) ۲۲۳	متد (reduceRight) ۱۲۳
متد (createElement) ۲۱۸	متد (reload) ۱۹۰
متد (Object.defineProperty) ۱۰۷، ۱۶۴	متد (remove) ۲۲۳
متد (Object.defineProperty) ۱۰۷	متد (removeAttribute) ۲۱۴
متد (every) ۱۲۳	متد (removeChild) ۲۱۹
متد (exec) ۱۸۳	متد (removeEventListener) ۲۴۳
متد (execScript) ۵۹	متد (removeItem) ۲۷۰
متد (factorial) ۸۵	متد (replace) ۲۴، ۱۸۰، ۱۸۲
متد (filter) ۱۲۲	متد (replaceChild) ۲۱۹
متد (forEach) ۱۱۵، ۱۲۲، ۱۶۶	متد (reverse) ۱۱۷
متد (forward) ۱۹۱	متد (scrollBy) ۲۲۶
متد (getAllResponseHeaders) ۲۵۴	متد (scrollIntoView) ۲۲۶
متد (getAttribute) ۲۱۴، ۲۲۲	متد (scrollTo) ۲۲۶
متد (getBoundingClientRect) ۲۲۵	متد (seal) ۱۰۹
متد (getElementById) ۱۹۷، ۲۰۴	متد (search) ۱۸۰
متد (getElementsByClassName) ۲۰۹	متد (send) ۲۵۳، ۲۵۶، ۲۶۴
متد (getElementsByName) ۲۰۵، ۲۰۸	متد (setAttribute) ۲۱۴، ۲۲۲
متد (getElementsByTagName) ۲۰۶، ۲۰۸	متد (setInterval) ۱۸۹
متد (getItem) ۲۷۰	متد (setItem) ۲۷۰
متد (getPrototypeOf) ۱۰۸	متد (setRequestHeader) ۲۵۲
متد (getResponseHeader) ۲۵۴	متد (setTimeout) ۱۸۸
متد (go) ۱۹۱	متد (shift) ۱۲۰
متد (hasAttribute) ۲۱۴	متد (slice) ۱۱۸
متد (hasOwnProperty) ۹۸	متد (some) ۱۲۳
متد (includes) ۱۶۶	متد (sort) ۱۱۷
متد (insertAdjacentHTML) ۲۱۵	متد (splice) ۱۱۹
متد (insertBefore) ۲۱۸	متد (split) ۱۱۶، ۱۸۲، ۲۷۶
متد (isPrototypeOf) ۱۰۸	متد (stopImmediatePropagation) ۲۴۸

مقدار دهنده‌های اولیه، ۳۸	متد (stopPropagation)، ۲۴۸
مقدار دهنده‌ی اولیه‌ی «شیء لفظی»، ۳۸	متد (test)، ۱۸۳
مقدار قبل از افزایش واحد، ۴۸	متد (toggle)، ۲۲۳
مقدار قبل از کاهش واحد، ۴۸	متد (toString)، ۱۰۹، ۳۱، ۱۲۱، ۱۵۰، ۱۶۶، ۱۹۰
مقدار منفی بی‌نهایت، ۲۰	متد (toUpperCase)، ۲۴، ۱۳۶
مقدار ناعدد (NaN)، ۲۰	متد (String.trim)، ۱۶۸
مقدارگذار، ۱۰۲	متد (unshift)، ۱۲۰
مقدارگیر، ۱۰۲	متدها، ۱۳۵، ۱۶۶
مکان انطباق، ۱۷۸	بازنویسی، ۱۶۶
میانبرهای صفحه‌کلید، ۱۸۷، ۲۳۵	فراخوانی، ۱۳۵
	متدهای کلاس، ۱۶۲
	متدهای نمونه، ۱۶۲
	متدهای نوعی، ۱۱۲
	متغیر سراسری (document)، ۲۰۴
	متغیرها، ۱۷، ۱۷، ۱۸، ۳۳، ۳۴
	اعلام، ۱۷، ۳۳
	بی‌نوع، ۱۷
	سراسری، ۱۸
	محلی، ۳۴
	متغیرهای سراسری، ۱۸، ۲۰۴
	متغیرهای محلی، ۳۴
	متن ساده، محتوای عنصر به صورت، ۲۱۶
	محتوای عنصر، ۲۱۵، ۲۱۷
	محیط رویدادپردازها، ۲۴۴
	محیط مرورگری، ۱۹۵
	محیط‌های مرورگری تو در تو، ۱۹۵
	مدت (indexOf)، ۱۲۵
	مدل شیئی سند (DOM)، ۲۰۱، ۲۰۴، ۲۴۶
	مقادیر، ۱۷
	مقادیر بولی، ۲۴
	مقادیر درست‌مانند، ۲۵
	مقادیر عدد حقیقی، ۱۹
	مقادیر عدد صحیح، ۱۸
	مقادیر لفظی، ۲۱، ۳۸، ۱۷۲
	رشته، ۲۱
	مقادیر، ۳۸
	نویسه‌های عبارت قاعده‌مند، ۱۷۲
	مقادیر ممیز شناور، ۱۸
	مقادیر نادرست‌مانند، ۲۵
	مقدار NaN (ناعدد)، ۲۰، ۲۷
	مقدار بی‌نهایت، ۲۰
— ن —	
ناوبری مرورگر، ۱۸۹	
نشانی و ناوبری مرورگر، ۱۸۹	
نشانی‌های اینترنتی، ۲۵۱، ۲۵۷، ۲۶۸	
نشانی‌های با مبدأ متفاوت، ۲۵۷	
نمایش نمایی، ۱۹	
نمونه‌ها، ۱۵۳، ۱۶۱	
نود، ۹، ۱۰	
نوع داده‌های شیء، ۹۱	
نوع رویداد (keydown)، ۲۲۹	
نویسه‌های یونیکد، ۲۲، ۱۷۳، ۱۷۴	
نویسه‌ی NUL، ۲۳، ۱۷۲	
نویسه‌ی آپوستروف، ۲۳	
نویسه‌ی بک‌اسپیس، ۲۳	
نویسه‌ی تب، ۱۷۲	
نویسه‌ی تب افقی، ۲۳	
نویسه‌ی تب عمودی، ۲۳، ۱۷۲	
نویسه‌ی سر سطر، ۲۳، ۱۷۲	
نویسه‌ی سطر جدید، ۲۳، ۱۷۲	
نویسه‌ی صفحه‌ی بعد، ۲۳، ۱۷۲	
نویسه‌ی کنترلی، ۱۷۳	
نویسه‌ی گریز (۱)، ۲۲، ۱۷۴	
نویسه‌ی گریز کج خط وارون (۱)، ۲۲، ۱۷۴	
نویسه‌ی نقل‌قول دوگانی، ۲۳، ۱۷۷	
نویسه‌ی نقل‌قول یکه، ۲۳، ۱۷۷	
— و —	
واژه‌های ذخیره، ۱۴، ۲۱۳	
والد گره، ۲۰۲	
وب‌سوکت‌ها، ۲۳۷، ۲۶۳	
وصله‌گذاری میمونی، ۱۴۰	

هدرهای پاسخ، ۲۵۴
هشتک (۸) دسته‌ی نویسه‌ای منفی، ۱۷۳
هشتک (۸) عملگر XOR بیتی، ۴۳، ۴۹

ویرگول نقطه (:)، ۱۵، ۲۷۵
ویرگول نقطه‌ی اختیاری، ۱۵
ویوپورت، ۲۲۴

— ه —

هدر USER-AGENT در HTTP، ۱۹۳

واژه‌نامه

انگلیسی به فارسی

accessor	دستیاب	binding	انتساب
action	عمل	block	بلوک
alternation	گزینش	body	بدنه
ancestors	اسلاف	bold	پررنگ
anchor	لنگر	border	کناره
animation	پویانمایی	breakpoint	نقطه‌ی انفصال
API	رابط برنامه‌نویسی	browser	مرورگر
Apple	اپل	bubbling	شناور شدن
argument	آوند	bug	اشکال
arithmetic operator	عملگر حسابی	built-in	توکار
arity	چندگانگی	cache	حافظه‌ی نهانی
array	آرایه	callback	بازخوان
array-like	شبه‌آرایه	callout	گفته‌ها
ASCII	اسکی	capturing	گرفت کننده
assertion	تأیید	carriage return	سر سطر
assignment	تخصیص	character	نویسه
associative array	آرایه‌ی ارتباطی	charset	نویسگان
asynchronous	ناهمگام	child	فرزند
attribute	صفت	Chrome	کروم
authentication	احراز هویت	class	کلاس
backslash	کج خط وارون	client	کارخواه
backspace	پس‌بر	closure	بستار
binary	دوگانی	collection	گردایه

comment	توضیح	event listener	رویدادشنو
concatenation	به هم پیوستن	event-driven	مبتنی بر رویداد
configurable	قابل پیکربندی	exception	استثنا
constructor	سازنده	exponent	نمای
container	گنجایه	exponential notation	نمایش نمایی
context	محیط	expression	عبارت
context menu	منوی موقعیتی	extend	گسترش دادن
control structure	ساختار کنترلی	extensible	قابل گسترش
currying	کاری سازی	factory function	تابع کارخانه‌ای
database	پایگاه داده‌ای	falsy	نادرست‌مانند
debugger	اشکال‌زدا	field	فیلد
declaration	اعلام	Firefox	فایرفاکس
decrement	کاهش واحد	flag	پرچم
default	پیش‌فرض	focus	کانون
deprecated	منسوخ	format	قالب
descendants	اخلاف	fragment	قطعه
desktop	میز کار	freeze	منجمد کردن
dialog box	کادر گفتگو	function	تابع
directive	فرمان	functional programming	برنامه‌نویسی تابعی
document	سند	generic	نوعی
Document Object Model	مدل شیئی سند	gesture	اشاره
domain	دامنه	getter	مقدارگیر
doubly linked list	لیست دوپیوندی	getting	مقدارگیری
drag	کشاندن	global	سراسری
drag and drop	کشنداز	greedy	آزمندانه
dynamic	پویا	hash	درهم‌سازی
ECMAScript	اکما اسکریپت	hashtable	جدول درهم‌سازی
element	عنصر	header	سرآیند
embedding	جای‌گذاری	hoisting	بر کشیدن
empty statement	دستورالعمل خالی	host	میزبان
encoding	کدگذاری	identifier	شناسه
enumerable	قابل برشماری	identity operator	عملگر همانی
error	خطا	immutable	تغییرناپذیر
escape sequence	توالی گریز	increment	افزایش واحد
event	رویداد	indentation	تورفتگی
event capturing	گرفتن رویداد	index	اندیس
event handler	رویدادپرداز	infinite loop	حلقه‌ی نامتناهی

initialization	مقداردهی اولیه	Netscape	نت اسکپ
initializer	مقدار دهندهی اولیه	newline	سطر جدید
inline	درون خطی	non-null	ناهیچ
insert	درج کردن	not-a-number value	مقدار ناعدد
instance	نمونه	null	هیچ
Internet Explorer	اینترنت اکسپلورر	object	شء
interpreter	تفسیرگر	octal	هشتگانی
invocation	فراخوانی	operand	عملوند
iPad	آپد	operator	عملگر
iPhone	آیفون	orientation	راستا
keyboard	صفحه کلید	origin	مبدأ
keyword	کلیدواژه	overflow	سرریز
label	برچسب	override	بازنویسی
Latin-1	لاتین-۱	own property	خصلت خودی
leading	ابتدایی	padding	حاشیهی درونی
lexical scope	قلمرو لغوی	parameter	پارامتر
life cycle	چرخه ی زندگی	parent	والد
link	پیوند	parse	تجزیه کردن
linked list	لیست پیوندی	parser	تجزیه گر
literal	لفظی	pinch close	نزدیک کردن دو انگشت
local	محلی	pinch open	دور کردن دو انگشت
lookahead	پیش نگر	port	درگاه
loop	حلقه	position	مکان
map	نگاشت	precedence	تقدم
margin	حاشیه	primitive	بدوی
member	عضو	procedure	رویه
metacharacter	فرانویسه	propagation	انتشار
method	متد	property	خصلت
Microsoft	مایکروسافت	property descriptor	توصیف کنندهی خصلت
modifier keys	کلیدهای ترکیب شونده	prototype	پیش نمونه
modulo	نهشته	prototype chain	زنجیره ی پیش نمونه
monkey-patching	وصله گذاری میمونی	proxy server	پیشکار
mouse	موشواره	query	پرس و جو
multiline	چندسطری	radio button	دکمه ی رادیویی
mutable	قابل تغییر	range	دامنه
namespace	فضای نام	read-only	فقط خواندنی
native	بومی		

recursive	تراجعی	string	رشته
redirect	تغییر مسیر	style	شیوه
reduce	کاهش	stylesheet	شیوه‌نامه
reference	اشاره	subclass	زیرکلاس
relational operator	عملگر رابطه‌ای	subdomain	زیردامنه
relative	نسبی	subroutine	زیرروال
rendering	آمایش	superclass	ابزرکلاس
request	درخواست	syntax	نحو
reset	بازنشانی	tab	جهش
response	پاسخ	tab	برگه
Safari	سافاری	tag	تگ
scaling	مقیاس‌دهی	target	هدف
scope	قلمرو	task bar	نوار وظیفه
scope chain	زنجیره‌ی قلمرو	ternary	سه‌گانی
screen	صفحه‌ی نمایش	timer	زمان‌سنج
script	اسکریپت	to raise an error	بر انگیزختن خطا
scrollbar	نوار نورد	toolbar	نوار ابزار
scrolling	نوردش	tooltip	نکته‌ها
seal	مهر و موم	touch	لمس
selection	انتخاب	trailing	انتهایی
selector	انتخابگر	traversal	پیمایش
semicolon	ویرگول نقطه	truthy	درست‌مانند
server	کارساز	unary	یگانی
setter	مقدارگذار	undefined	تعریف نشده
setting	مقدارگذاری	underflow	زیرریز
shortcut	میانبر	underscore	زیرخط
sibling	برادر	Unicode	یونیکد
side effect	اثر جانبی	unordered	بدون ترتیب
slash	کج‌خط	untyped	بی‌نوع
snapshot	عکس فوری	update	به‌نگام‌سازی
space	فضای خالی	value	مقدار
sparse	تُنک	varargs functions	توابع با آوندهای متغیر
stack	پشته	variable	متغیر
state	حالت	variable arity	توابع با چندگانگی
statement	دستورالعمل	functions	متغیر
static	ایستا	variadic functions	توابع چندگانگی
status code	کد وضعیت	version	ویرایش
strict mode	حالت اکید		

viewport	هماگه	writable	قابل نوشتن
Web Storage	ذخیره‌سازی وب	zero-based	با پایه‌ی صفر
whitespace	فضای سفید		

فارسی به انگلیسی

Internet Explorer	اینترنت اکسپلورر	array	آرایه
zero-based	با پایه‌ی صفر	associative array	آرایه‌ی ارتباطی
callback	بازخوان	greedy	آزمندانه
reset	بازنشانی	rendering	آمایش
override	بازنویسی	argument	آوند
body	بدنه	iPad	آپید
unordered	بدون ترتیب	iPhone	آیفون
primitive	بدوی	leading	ابتدایی
to raise an error	بر انگیختن خطا	superclass	ابركلاس
hoisting	بر کشیدن	Apple	اپل
sibling	برادر	side effect	اثر جانبی
label	برچسب	authentication	احراز هویت
tab	برگه	descendants	اخلاف
functional programming	برنامه‌نویسی تابعی	exception	استثنا
closure	بستار	script	اسکریپت
block	بلوک	ASCII	اسکی
native	یومی	ancestors	اسلاف
concatenation	به هم پیوستن	reference	اشاره
update	بهنگام‌سازی	gesture	اشاره
untyped	بی‌نوع	bug	اشکال
parameter	پارامتر	debugger	اشکال‌زدا
response	پاسخ	declaration	اعلام
database	پایگاه داده‌ای	increment	افزایش واحد
flag	پرچم	ECMAScript	اکما اسکریپت
bold	پررنگ	selection	انتخاب
query	پرس و جو	selector	انتخابگر
backspace	پس‌بر	binding	انتساب
stack	پشته	propagation	انتشار
dynamic	پویا	trailing	انتهای
animation	پویانمایی	index	اندیس
		static	ایستا

padding	حاشیهی درونی	default	پیش فرض
cache	حافظه‌ی نهانی	proxy server	پیشکار
state	حالت	lookahead	پیش‌نگری
strict mode	حالت اکید	prototype	پیش‌نمونه
loop	حلقه	traversal	پیمایش
infinite loop	حلقه‌ی نامتناهی	link	پیوند
property	خصلت	function	تابع
own property	خصلت خودی	factory function	تابع کارخانه‌ای
error	خطا	assertion	تأیید
range	دامنه	parse	تجزیه کردن
domain	دامنه	parser	تجزیه‌گر
insert	درج کردن	assignment	تخصیص
request	درخواست	recursive	تراجعی
truthy	درست‌مانند	undefined	تعریف نشده
port	درگاه	redirect	تغییر مسیر
inline	درون‌خطی	immutable	تغییرناپذیر
hash	درهم‌سازی	interpreter	تفسیرگر
statement	دستورالعمل	precedence	تقدم
empty statement	دستورالعمل خالی	tag	تگ
accessor	دستیاب	sparse	تُنک
radio button	دکمه‌ی رادیویی	varargs functions	توابع با آوندهای متغیر
pinch open	دور کردن دو انگشت	variable arity functions	توابع با چندگانگی متغیر
binary	دوگانی	variadic functions	توابع چندگانگی
Web Storage	ذخیره‌سازی وب	escape sequence	توالی گریز
API	رابط برنامه‌نویسی	indentation	تورفتگی
orientation	راستا	property descriptor	توصیف‌کننده‌ی خصلت
string	رشته	comment	توضیح
event	رویداد	built-in	توکار
event handler	رویدادپرداز	embedding	جای‌گذاری
event listener	رویدادشنو	hashtable	جدول درهم‌سازی
procedure	رویه	tab	جهش
timer	زمان‌سنج	life cycle	چرخه‌ی زندگی
prototype chain	زنجیره‌ی پیش‌نمونه	multiline	چندسطری
scope chain	زنجیره‌ی قلمرو	arity	چندگانگی
underscore	زیرخط	margin	حاشیه
subdomain	زیردامنه		
subroutine	زیرروال		
underflow	زیرریز		

read-only	فقط خواندنی	subclass	زیرکلاس
field	فیلد	control structure	ساختار کنترلی
enumerable	قابل برشماری	constructor	سازنده
configurable	قابل پیکربندی	Safari	سافاری
mutable	قابل تغییر	carriage return	سر سطر
extensible	قابل گسترش	header	سرآیند
writable	قابل نوشتن	global	سراسری
format	قالب	overflow	سرریز
fragment	قطعه	newline	سطر جدید
scope	قلمرو	document	سند
lexical scope	قلمرو لغوی	ternary	سه گانی
dialog box	کادر گفتگو	array-like	شبه آرایه
client	کارخواه	identifier	شناسه
server	کارساز	bubbling	شناور شدن
currying	کاری سازی	object	شیء
focus	کانون	style	شیوه
reduce	کاهش	stylesheet	شیوه نامه
decrement	کاهش واحد	attribute	صفت
slash	کج خط	keyboard	صفحه کلید
backslash	کج خط وارون	screen	صفحه‌ی نمایش
status code	کد وضعیت	expression	عبارت
encoding	کدگذاری	member	عضو
Chrome	کروم	snapshot	عکس فوری
drag	کشاندن	action	عمل
drag and drop	کشنداز	operator	عملگر
class	کلاس	arithmetic operator	عملگر حسابی
keyword	کلیدواژه	relational operator	عملگر رابطه‌ای
modifier keys	کلیدهای ترکیب	identity operator	عملگر همانی
	شونده	operand	عملوند
border	کناره	element	عنصر
collection	گردابه	Firefox	فایرفاکس
capturing	گرفت کننده	invocation	فراخوانی
event capturing	گرفتن رویداد	metacharacter	فرانویسه
alternation	گزینش	child	فرزند
extend	گسترش دادن	directive	فرمان
callout	گفته‌ها	space	فضای خالی
container	گنجایه	whitespace	فضای سفید
Latin-1	لاتین-۱	namespace	فضای نام

falsy	نادرست مانند	literal	لفظی
asynchronous	ناهمگام	touch	لمس
non-null	ناهیچ	anchor	لنگر
Netscape	نت اسکپ	linked list	لیست پیوندی
syntax	نحو	doubly linked list	لیست دوپیوندی
pinch close	نزدیک کردن دو انگشت	Microsoft	مایکروسافت
relative	نسبی	event-driven	مبتنی بر رویداد
breakpoint	نقطه‌ی انفصال	origin	مبدأ
tooltip	نکته‌ها	method	متد
map	نگاشت	variable	متغیر
exponent	نمای	local	محلی
viewport	نمایگاه	context	محیط
exponential notation	نمایش نمایی	Document Object Model	مدل شیئی سند
instance	نمونه	browser	مرورگر
toolbar	نوار ابزار	value	مقدار
scrollbar	نوار نورد	initializer	مقدار دهنده‌ی اولیه
task bar	نوار وظیفه	not-a-number value	مقدار ناعدد
scrolling	نوردش	initialization	مقداردهی اولیه
generic	نوعی	setter	مقدارگذار
charset	نویسگان	setting	مقدارگذاری
character	نویسه	getter	مقدارگیر
modulo	نہشته	getting	مقدارگیری
parent	والد	scaling	مقیاس‌دهی
monkey-patching	وصله‌گذاری میمونی	position	مکان
version	ویرایش	freeze	منجمد کردن
semicolon	ویرگول نقطه	deprecated	منسوخ
target	هدف	context menu	منوی موقعیتی
octal	هشتگانی	mouse	موشواره
null	هیچ	seal	مهر و موم
unary	یگانی	shortcut	میانبر
Unicode	یونیکد	desktop	میز کار
		host	میزبان

جاوا اسکریپت زبان برنامه‌نویسی وب است و با رشد انفجاری فناوری‌های استاندارد وب در سال‌های اخیر، بر کاربرد و اهمیت آن افزوده شده است. این کتاب راهنمای سریعی برای فراگیری برنامه‌نویسی جاوا اسکریپت است.

تُه فصل اول کتاب به آموزش جدیدترین استاندارد زبان هسته‌ی جاوا اسکریپت (**اکما اسکریپت 5**) اختصاص دارد و مباحث زیر را پوشش می‌دهد:

- انواع، مقادیر، و متغیرها

- عملگرها، عبارت‌ها، و دستورالعمل‌ها

- اشیا و آرایه‌ها

- توابع و کلاس‌ها

پنج فصل بعدی به بررسی رابط‌های برنامه‌نویسی بنیادی برای استفاده از جاوا اسکریپت به همراه **HTML5** می‌پردازد و شامل مباحث زیر است:

- کار با پنجره‌های مرورگر

- اسکریپت‌نویسی سندهای HTML و عناصر سند

- تغییر و اعمال شیوه‌ها و کلاس‌های CSS

- پاسخ به رویدادهای ورودی کاربر

- ارتباط با سرورهای وب

- ذخیره‌ی محلی داده‌ها روی کامپیوتر کاربر

این کتاب در زبان اصلی با عنوان «**مرجع جیبی جاوا اسکریپت**» منتشر شده است.

شابک ۹۷۸-۶۰۰-۰۴-۰۴۳۲-۱

