

فهرست مطالب

فصل ۱: شروع به کار ۱۳

۱۴	jQuery چه کاری انجام می دهد
۱۶	چرا jQuery خوب کار می کند
۱۷	آشنایی با پلاگین های jQuery
۱۷	برای استفاده از این کتاب باید چه چیزهایی بدانم
۱۷	مثال های کتاب
۱۷	چه نرم افزارهایی برای این کتاب نیاز دارم؟
۱۷	دریافت jQuery
۱۸	یک ویرایشگر HTML
۱۸	داشتن یک مرورگر وب
۱۹	یک سرور وب
۱۹	Node.js

فصل ۲: آشنایی با جاوااسکریپت ۲۱

۲۲	آماده شدن برای استفاده از جاوااسکریپت
۲۳	استفاده از دستورات
۲۴	تعریف و استفاده از توابع
۲۵	تعریف توابع با پارامتر
۲۶	تعریف توابعی که مقداری را بر می گردانند
۲۷	استفاده از متغیرها و انواع
۲۸	استفاده از انواع اصلی
۳۰	ایجاد اشیا
۳۱	استفاده از لیترال های شی (مقادیر مستقیم)
۳۲	استفاده از توابع به عنوان متد
۳۳	کار کردن با اشیا
۳۵	شمارش مشخصات یک شی
۳۶	اضافه و حذف مشخصات و متدها
۳۷	تعیین این که آیا یک شی دارای یک مشخصه است یا خیر
۳۸	استفاده از عملگرهای جاوااسکریپت
۳۸	استفاده از دستورات شرطی
۳۹	عملگر تساوی و عملگر یکسان بودن
۴۲	تبدیل صریح انواع
۴۴	کار کردن با آرایه ها
۴۵	استفاده از لیترال های آرایه ای
۴۵	خواندن و تغییر محتویات یک آرایه



۴۶	شمارش محتویات یک آرایه
۴۶	استفاده از متدهای آرایه داخلی
۴۷	اداره خطاها
۴۹	مقایسه مقادیر null و undefined
۵۱	بررسی برای null یا undefined
۵۲	ایجاد تمایز بین null و undefined

فصل ۳: اصول jQuery ۵۳

۵۴	آماده کردن jQuery
۵۷	استفاده از توضیحات شرطی
۵۷	اولین اسکریپت jQuery
۶۰	آشنایی با تابع \$ در jQuery
۶۲	آشنایی با مدل شی سند (DOM)
۶۲	استفاده از DOM
۶۶	تغییر DOM
۶۶	تغییر شیوه‌ها
۶۷	اداره رویدادها
۶۹	آشنایی با جریان رویداد
۷۰	عمل‌های پیش‌فرض
۷۰	انتظار برای DOM
۷۲	استفاده از نمادگذاری جایگزین
۷۳	تعویق رویداد ready
۷۴	انتخاب عناصر
۷۷	محدود کردن انتخاب با یک محتوا
۷۸	درک نتیجه انتخاب
۷۹	تعیین محتوا
۸۰	کار کردن با اشیای DOM
۸۱	در نظر گرفتن یک شی jQuery به عنوان یک آرایه
۸۲	تکرار یک تابع روی اشیای DOM
۸۲	پیدا کردن شاخص‌ها و عناصر مشخص شده
۸۵	تغییر چند عنصر و فراخوان‌های متد زنجیره‌ای
۸۹	اداره رویدادها

فصل ۴: مدیریت انتخاب عناصر ۹۱

۹۲	بسط دادن انتخاب
۹۵	محدود کردن انتخاب
۹۵	کاهش انتخاب به یک عنصر خاص
۹۶	کاهش انتخاب بر حسب بازه



۹۷	فیلتر کردن عناصر
۱۰۰	محدود کردن انتخاب بر مبنای فرزندان
۱۰۱	نگاشت انتخاب
۱۰۳	آزمایش انتخاب
۱۰۴	تغییر و سپس باز کردن انتخاب
۱۰۶	حرکت در DOM
۱۰۶	حرکت درون سلسله مراتب
۱۰۹	استفاده از متد find برای ایجاد یک قسمت مشترک
۱۰۹	حرکت به بالا در سلسله مراتب
۱۱۱	انتخاب عناصر پدر
۱۱۲	انتخاب اجداد
۱۱۴	انتخاب اولین جد منطبق
۱۱۷	حرکت روی سلسله مراتب
۱۱۸	انتخاب همه خویشاوندان
۱۲۰	انتخاب خویشاوندان قبلی و بعدی

فصل ۵: مدیریت DOM..... ۱۲۱

۱۲۲	ایجاد عناصر جدید
۱۲۲	ایجاد عناصر با تابع \$
۱۲۴	ایجاد عناصر جدید با کپی عناصر موجود
۱۲۵	ایجاد عناصر با استفاده از DOM API
۱۲۷	قرار دادن عناصر فرزند
۱۲۹	اضافه کردن عناصر به ابتدا
۱۳۱	قرار دادن عناصر یکسان در مکان‌های متفاوت
۱۳۴	قرار دادن از یک شی jQuery
۱۳۴	قرار دادن عناصر با استفاده از یک تابع
۱۳۶	قرار دادن عناصر پدر
۱۳۸	بسته‌بندی عناصر با یکدیگر در یک پدر منفرد
۱۴۱	قرار دادن محتویات عناصر
۱۴۲	بسته‌بندی عناصر با استفاده از یک تابع
۱۴۳	قرار دادن عناصر خویشاوند
۱۴۵	قرار دادن خویشاوندان از یک شی jQuery
۱۴۶	قرار دادن خویشاوندان با یک تابع
۱۴۷	جایگزین کردن عناصر
۱۴۹	عوض کردن عناصر با استفاده از یک تابع
۱۵۰	حذف عناصر
۱۵۲	جدا کردن عناصر
۱۵۳	خالی کردن عناصر



از بسته بندی خارج کردن عناصر ۱۵۳

فصل ۶: مدیریت عناصر ۱۵۵

کار کردن با خاصیت‌ها و مشخصات ۱۵۶
تعیین مقدار یک خاصیت ۱۵۹
تعیین چند خاصیت ۱۶۰
تعیین دینامیک مقادیر ۱۶۱
حذف یک خاصیت ۱۶۲
کار کردن با مشخصات ۱۶۲
کار کردن با کلاس‌ها ۱۶۳
اضافه یا حذف کلاس‌ها با استفاده از یک تابع ۱۶۶
فعال یا غیرفعال کردن کلاس‌ها ۱۶۸
فعال یا غیرفعال کردن چند کلاس ۱۷۱
فعال و غیرفعال کردن همه کلاس‌ها ۱۷۲
فعال یا غیرفعال کردن کلاس‌ها در یک جهت ۱۷۳
فعال و غیرفعال کردن کلاس‌ها به صورت دینامیک ۱۷۴
کار کردن با CSS ۱۷۶
دریافت و تعیین یک مقدار CSS ۱۷۷
دریافت چند مشخصه CSS ۱۷۸
تعیین چند مشخصه CSS ۱۷۹
تعیین مقادیر نسبی ۱۷۹
تعیین مشخصات با استفاده از یک تابع ۱۸۱
استفاده از متدهای CSS مخصوص مشخصه ۱۸۲
تعیین پهنا و ارتفاع با استفاده از یک تابع ۱۸۳
کار کردن با محتویات عناصر ۱۸۴
تعیین محتویات عناصر ۱۸۵
تعیین محتویات عناصر با استفاده از یک تابع ۱۸۵
کار کردن با عناصر فرم ۱۸۷
تعیین مقادیر عناصر فرم ۱۸۷
تعیین مقادیر عنصر فرم با استفاده از یک تابع ۱۸۸
ارتباط دادن داده‌ها با عناصر ۱۸۹

فصل ۷: کار کردن با رویدادها ۱۹۳

اداره رویدادها ۱۹۴
ثبت یک تابع برای اداره چند نوع رویداد ۱۹۸
فراهم آوردن داده برای تابع اداره کننده رویداد ۱۹۹
غیرفعال کردن عمل پیش‌فرض ۲۰۰
حذف توابع اداره کننده رویداد ۲۰۲



۲۰۴	جدا کردن از درون تابع اداره کننده رویداد
۲۰۵	اجرای یک اداره کننده برای یک بار
۲۰۵	انجام اتصال رویداد زنده
۲۱۰	محدود کردن حرکت در DOM برای اداره کننده‌های رویداد زنده
۲۱۱	فراخواندن مستقیم اداره کننده‌های رویداد
۲۱۳	استفاده از یک شی Event
۲۱۴	استفاده از متد triggerHandler
۲۱۵	استفاده از متدهای خلاصه شده رویداد
۲۱۸	استفاده از متدهای خلاصه رویداد document
۲۱۸	استفاده از متدهای خلاصه‌شده رویداد مرورگر
۲۱۸	استفاده از متدهای خلاصه شده رویداد ماوس
۲۲۰	استفاده از متدهای خلاصه شده رویداد فرم
۲۲۱	استفاده از متدهای خلاصه شده رویداد صفحه کلید

فصل ۸: استفاده از افکتهای jQuery ۲۲۳

۲۲۴	استفاده از افکتهای ساده
۲۲۸	فعال/غیرفعال کردن عناصر
۲۲۹	فعال/غیرفعال کردن در یک جهت
۲۳۰	ایجاد انیمیشن روی قابلیت دیده شدن عناصر
۲۳۲	استفاده از کال‌بک‌های افکت‌ها
۲۳۵	ایجاد افکتهای تکرار شونده
۲۳۷	استفاده از افکتهای اسلاید
۲۳۹	استفاده از افکتهای محو شدن
۲۴۰	محو کردن به یک کد ری خاص
۲۴۲	ایجاد افکتهای دلخواه
۲۴۴	استفاده از مقادیر مشخصه مطلق
۲۴۵	استفاده از مقادیر مشخصه نسبی
۲۴۶	ایجاد و مدیریت صف افکت
۲۴۸	نمایش موارد درون صف افکت
۲۵۰	متوقف کردن افکت‌ها و پاک کردن صف
۲۵۵	قرار دادن تأخیر در صف
۲۵۶	قرار دادن توابع به درون صف
۲۵۸	فعال و غیرفعال کردن انیمیشن‌های افکت

فصل ۹: تمرین ۲۵۹

۲۶۰	نگاهی به سند مثال
۲۶۲	اضافه کردن گل‌های دیگر
۲۶۴	اضافه کردن دکمه‌هایی برای حرکت بین صفحات



۲۶۷	 دكمه ارسال
۲۷۰	 پياده‌سازي توابع اداره كننده رويداد براي دكمه‌ها
۲۷۲	 جمع زدن انتخاب محصول
۲۷۵	 غيرفعال كردن جاواسكريپت

فصل ۱۰: استفاده از الكوهای داده‌ای..... ۲۷۷

۲۷۸	 مشكلاتي كه الكوها آن‌ها را حل مي‌كنند
۲۷۹	 آماده كردن كتابخانه الكو
۲۸۱	 اولين مثال الكوي داده‌اي
۲۸۳	 تعريف داده‌ها
۲۸۴	 تعريف الكو
۲۸۵	 اعمال الكو
۲۸۶	 تغيير نتيجه
۲۸۸	 تنظيم ورودی
۲۸۹	 استفاده از منطق الكو
۲۹۰	 ايجاد محتويات شرطی
۲۹۳	 شمارش آرايه‌ها و مشخصات
۲۹۶	 تغيير محتوای داده‌ای
۲۹۷	 استفاده از كمك كننده #with
۲۹۸	 دسترسي به محتويات داده پدر
۳۰۰	 ايجاد كمك كننده‌های الكوي دلخواه
۳۰۲	 ايجاد يك كمك كننده الكوي شرطی
۳۰۵	 برگرداندن محتويات پيچيده‌تر
۳۰۶	 دريافت آرگومان‌های دلخواه از تابع كمك كننده
۳۰۹	 فراهم آوردن مشخصات الكوي دلخواه

فصل ۱۱: کار کردن با فرم‌ها..... ۳۱۱

۳۱۲	 آماده كردن سرور Node.js
۳۱۶	 متدهای مربوط به رویدادهای فرم
۳۱۷	 کار کردن با فوکوس فرم
۳۱۹	 کارکردن با تغييرات مقدار
۳۲۰	 کار کردن با ارسال فرم
۳۲۳	 تائيد اعتبار مقادير فرم
۳۲۶	 وارد كردن فايل جاواسكريپت
۳۲۷	 پيكربندي تائيد اعتبار
۳۲۷	 تعريف قوانين تائيد اعتبار
۳۲۸	 اعمال قوانين تائيد اعتبار
۳۲۹	 استفاده از بررسي‌های تائيد اعتبار



۳۳۰.....	اعمال قوانین تأیید اعتبار با استفاده از کلاس‌ها.
۳۳۳.....	اعمال مستقیم قوانین تأیید اعتبار روی عناصر.....
۳۳۷.....	اعمال قوانین تأیید اعتبار با استفاده از خاصیت name عنصر.....
۳۳۹.....	اعمال قوانین تأیید اعتبار با استفاده از خاصیت‌های عنصر.....
۳۴۱.....	مشخص کردن پیغام‌های تأیید اعتبار.....
۳۴۱.....	مشخص کردن پیغام برای تأیید اعتبار خاصیت و name.....
۳۴۴.....	مشخص کردن پیغام‌ها برای تأیید اعتبار روی عنصر.....
۳۴۶.....	ایجاد یک تأیید اعتبار دلخواه.....
۳۴۸.....	تعریف تابع تأیید اعتبار.....
۳۴۹.....	تعریف پیغام تأیید اعتبار.....
۳۵۱.....	اعمال فرمت روی نمایش خطای تأیید اعتبار.....
۳۵۲.....	تعیین کلاس برای عناصر نامعتبر.....
۳۵۲.....	تعیین عنصر پیغام خطا.....
۳۵۳.....	های‌لایت کردن عناصر نامعتبر.....
۳۵۴.....	استفاده از یک جمع‌بندی تأیید اعتبار.....
۳۵۶.....	آماده کردن پیغام‌های تأیید اعتبار.....
۳۵۸.....	ایجاد خلاصه تأیید اعتبار.....
۳۵۹.....	مرتب کردن ترکیب پیغام خطا.....

فصل ۱۲: استفاده از Ajax ۳۶۱

۳۶۲.....	استفاده از متدهای خلاصه شده Ajax.....
۳۶۲.....	انجام یک درخواست HTTP GET.....
۳۶۶.....	پردازش داده‌های پاسخ.....
۳۶۷.....	آسان‌تر کردن دیدن نتیجه.....
۳۶۸.....	دریافت انواع دیگر داده‌ها.....
۳۷۰.....	فراهم آوردن داده برای درخواست‌های GET.....
۳۷۱.....	انجام یک درخواست POST.....
۳۷۱.....	آماده کردن Node.js برای دریافت داده‌های فرم.....
۳۷۴.....	آشنایی با درخواست‌های مستقل از مبدأ در Ajax.....
۳۷۵.....	استفاده از متد post برای ارسال داده‌های فرم.....
۳۸۰.....	ارسال داده‌های دیگر با استفاده از متد post.....
۳۸۱.....	مشخص کردن نوع داده مورد انتظار.....
۳۸۳.....	جلوگیری از معمول‌ترین مشکل Ajax.....
۳۸۴.....	استفاده از متدهای مخصوص نوع.....
۳۸۴.....	دریافت یک قطعه کد HTML.....
۳۸۵.....	مدیریت عناصر اضافه شده با متد load.....
۳۸۷.....	دریافت و اجرای اسکریپت‌ها.....
۳۸۹.....	دریافت داده‌های JSON.....
۳۹۰.....	کار کردن با JSONP.....



استفاده از پلاگین jQuery Form ۳۹۲

فصل ۱۳: استفاده از Ajax: بخش ۲ ۳۹۵

اعمال یک درخواست ساده Ajax با API سطح پایین ۳۹۶

شی jqXHR ۳۹۸

تعیین URL درخواست ۴۰۰

ایجاد یک درخواست POST ۴۰۰

اداره کردن کال بک‌های Ajax ۴۰۲

کار کردن با درخواست‌های موفق ۴۰۲

کار کردن با خطاها ۴۰۴

کار کردن با درخواست‌های کامل شده ۴۰۶

پیکربندی درخواست‌ها قبل از ارسال آن‌ها ۴۰۸

مشخص کردن چند تابع اداره کننده رویداد ۴۰۹

تعیین محتوا برای رویدادها ۴۱۰

استفاده از رویدادهای جهانی Ajax ۴۱۱

کنترل رویدادهای جهانی ۴۱۵

پیکربندی تنظیمات اصلی برای یک درخواست Ajax ۴۱۵

تعیین هدرها و زمان ۴۱۶

ارسال داده‌های JSON به سرور ۴۱۷

استفاده از تنظیم پیکربندی پیشرفته‌تر ۴۱۹

همزمان کردن درخواست ۴۱۹

چشم‌پوشی از داده‌های تغییر نکرده ۴۲۰

کار کردن با کد وضعیت پاسخ ۴۲۲

پاک‌سازی داده‌های پاسخ ۴۲۴

مدیریت تبدیل داده ۴۲۵

آماده و فیلتر کردن درخواست‌های Ajax ۴۲۷

تعریف تنظیمات پیش‌فرض ۴۲۷

فیلتر درخواست‌ها ۴۲۸

فصل ۱

شروع به کار

jQuery

design



وب جهانی (World Wide Web) امروزی یک محیط دینامیک است و کاربران آن علاقه زیادی به ایجاد سایت‌های جذاب و اینتراکتیو دارند و توسعه‌دهندگان نیز از کتابخانه‌های جاوااسکریپت مانند jQuery استفاده کرده تا وظایف پرکاربرد را خودکار کنند و کارهای پیچیده را ساده‌تر کنند. یک دلیل برای محبوب بودن کتابخانه jQuery توانایی آن برای کمک در انواع مختلفی از وظایف است.

می‌توان دید که دانستن جایی که باید شروع کرد سخت است زیرا jQuery عملکردهای بسیار زیادی را ارائه کرده است. با این حال در طراحی این کتابخانه پیوستگی و تقارن خوبی وجود دارد؛ بسیاری از مفاهیم آن از ساختار HTML و CSS قرض گرفته شده‌اند. طراحی این کتابخانه اجازه یک شروع سریع برای طراحانی را می‌دهد که اندکی دارای تجربه برنامه‌نویسی هستند. اما برنامه‌نویس‌های باتجربه‌تر نیز می‌توانند از این یکپارچگی مفاهیم نیز بهره ببرند.

در این فصل مباحث زیر پوشش داده می‌شوند:

- ◀ امکانات اصلی jQuery
- ◀ آماده کردن محیط کدنویسی jQuery
- ◀ یک مثال ساده از اسکریپت jQuery
- ◀ دلایل انتخاب jQuery به کد عادی جاوااسکریپت
- ◀ ابزارهای توسعه معمول جاوااسکریپت

jQuery چه کاری انجام می‌دهد

کتابخانه jQuery یک لایه انتزاعی با کاربرد عمومی برای اسکریپت‌نویسی معمول وب فراهم آورده است و بنابراین در اغلب شرایط اسکریپت‌نویسی مفید است. طبیعت قابل انبساط آن به این معنی است که هیچ وقت نمی‌توانیم تمام کاربردها و توابع آن را در یک کتاب توضیح داد زیرا پلاگین‌های (Plugins) مرتباً در حال توسعه هستند تا امکانات جدیدی به آن اضافه کنند. با این حال امکانات اصلی در انجام کارهای زیر به ما کمک می‌کنند:

- ◀ دسترسی به عناصر درون یک سند: بدون استفاده از یک کتابخانه جاوااسکریپت، توسعه دهندگان وب اغلب باید کدهای زیادی بنویسند تا بتوانند در درخت Document Object Model یا DOM حرکت کرده و مکان‌های خاصی از ساختار یک سند HTML را انتخاب کنند. با استفاده از jQuery توسعه دهندگان یک مکانیسم انتخاب‌گر قدرتمند و مفید دارند که باعث شده دریافت قسمتی از سند که باید اداره شود بسیار آسان شود:

```
$('div.content').find('p');
```

- ◀ تغییر ظاهر یک صفحه وب: CSS روشی قدرتمند برای تأثیر گذاری روی روش رندر یک سند فراهم آورده است. اما به دلیل این که همه مرورگرها از یک استاندارد ثابت پشتیبانی نمی‌کنند، به مشکل بر می‌خورد. با استفاده از jQuery، توسعه دهندگان می‌توانند این شکاف را پر کرده و روی یک پشتیبانی



استاندارد یکسان روی همه مرورگرها تکیه کنند. به علاوه jQuery می تواند کلاس ها یا مشخصات شیوه منفرد اعمال شده روی قسمتی از سند را حتی بعد از اتمام رندر صفحه تغییر دهند.

```
$('#ul > li:first').addClass('active');
```

◀ تغییر محتویات یک سند: jQuery تنها به تغییرات ظاهری محدود نیست و می تواند محتویات خود سند را با تنها مقدار کمی کد تغییر دهد. متن می تواند تغییر داده شده، تصاویر را می توان درون سند قرار داده یا عوض کرد، می توان ترتیب لیست ها را تغییر داده یا کل ساختار HTML را بازنویسی کرد یا بسط داد که همگی این موارد به راحتی با استفاده از Application Programming Interface یا API انجام می شوند.

```
$('#container').append('<a href="more.html">more</a>');
```

◀ پاسخ به برهم کنش کاربر: حتی قدرتمندترین رفتارها نیز وقتی مفید هستند که کنترل روی مکان و زمان آن ها داشته باشید. کتابخانه jQuery یک روش عالی برای تفسیر و اداره انواع مختلفی از رویدادها (مانند کلیک کاربر روی یک لینک) فراهم آورده بدون این که لازم باشد خود کد HTML را با اداره کننده های رویداد به هم بریزید. در همین حال API اداره کننده رویداد آن باعث شده تا بین مرورگرها یکپارچگی به وجود بیاید.

```
$('#button.show-details').click(function() {
    $('#div.details').show();
});
```

◀ ایجاد انیمیشن روی تغییرات درون سند: برای پیاده کردن چنین رفتارهای اینتراکتیوی، یک طراح باید فیدبک های بصری برای کاربر فراهم بیاورد. کتابخانه jQuery انواع مختلفی از افکت ها مانند محو و پاک شدن را به همراه یک جعبه ابزار برای ایجاد افکت های جدید فراهم آورده است.

```
$('#div.details').slideDown();
```

◀ دریافت اطلاعات از یک سرور بدون نیاز به بارگذاری مجدد صفحه: این الگوی کد با نام Ajax شناخته می شود که در اصل مخفف XML and JavaScript Asynchronous است اما هم اکنون از تکنولوژی های بیشتری برای برقراری ارتباط بین کلاینت و سرور استفاده می کند. کتابخانه jQuery پیچیدگی مخصوص به مرورگر را از این فرآیند پرامکانات برداشته و به برنامه نویسان اجازه داده تا روی عملکرد مبتنی بر سرور متمرکز شوند.

```
$('#div.details').load('more.html #content');
```

◀ ساده کردن وظایف جاوااسکریپت: به غیر از همه امکانات مخصوص سند jQuery، این کتابخانه بهبودهایی برای ساختارهای اصلی جاوااسکریپت مانند اداره تکرارها و آرایه ها فراهم آورده است.

```
$.each(obj, function(key, value) {
    total += value;
});
```



چرا jQuery خوب کار می کند

jQuery چند استراتژی فراهم آورده تا بتواند امکانات ذکر شده قبلی را در عین فشردگی باقی ماندن فراهم بیاورد:

◀ **دانش CSS:** با قرار دادن مکانیسم تعیین مکان عناصر صفحه روی انتخاب گره های CSS، jQuery یک روش عالی برای بیان کردن ساختار یک سند فراهم آورده است. کتابخانه jQuery یک نقطه شروع برای طراحانی است که می خواهند رفتارهایی را به صفحات خود اضافه کنند زیرا تنها دانشی که برای انجام کارهای توسعه پیشرفته وب نیاز دارند، روش استفاده از CSS است.

◀ **پشتیبانی از افزونه ها:** jQuery به کاربران اجازه استفاده از پلاگین ها را می دهد. روش ایجاد پلاگین های جدید ساده بوده و به خوبی مستند شده است که اجازه انجام انواع کارها را با استفاده از ماژول های ساده اما مفید می دهد. حتی اغلب امکانات در داندلود ساده jQuery به صورت داخلی از طریق معماری پلاگین ها انجام پذیر بوده و در صورت نیاز می توان آن ها را حذف کرده تا یک کتابخانه کوچک تر باقی بماند.

◀ **لایه انتزاعی برای مرورگرها:** یکی از سختی های طراحی وب این است که هر مرورگر دارای نسخه های خود از استانداردهای منتشر شده است. بخش بزرگی از هر برنامه وب برای اداره امکاناتی است که روی هر پلتفرم متفاوت هستند. jQuery یک لایه انتزاعی فراهم آورده است که وظایف معمول را عادی کرده و باعث کاهش اندازه کد شده و در عین حال آن را ساده نگه می دارد.

◀ **اجازه کار کردن با ست ها:** وقتی به jQuery می گوئید همه عناصر را با کلاس collapse پیدا کرده و آن ها را پنهان کند، نیازی به ایجاد حلقه روی هر عنصر برگردانده شده نیست. به جای آن متدهایی مانند `hide()`، طراحی شده تا به صورت خودکار روی یک ست هایی از اشیاء کار کنند تا دیگر مجبور نباشید به صورت منفرد آن ها را اداره کنید. با این تکنیک که **تکرار غیر صریح (Implicit Iteration)** نام دارد، بسیاری از ساختارهای تکرار بدون استفاده شده که باعث کوتاه شدن کد می شود.

◀ **اجازه انجام چند عمل در یک خط:** برای جلوگیری از ایجاد بیش از حد متغیرهای موقت یا تکرارهای وقت تلف کن، jQuery یک الگوی برنامه نویسی با نام **زنجیرهای کردن (Chaining)** برای اغلب متدهای خود فراهم آورده است. این بدان معنی است که نتیجه اغلب عملیات روی یک شی خود آن شی است که آماده انجام عمل بعدی است که روی آن اعمال می شود.

این استراتژی ها باعث شده تا اندازه بسته jQuery بسیار کوچک شود و در عین حال تکنیک هایی برای نگه داشتن کدهای سفارشی فراهم می آورد که از این کتابخانه فشرده استفاده می کنند.

ظرافت این کتابخانه قسمتی به دلیل طراحی و قسمتی به دلیل فرآیند انقلابی است که توسط جامعه توسعه آن فراهم آمده است. کاربران jQuery نه تنها در حال بحث درباره توسعه پلاگین ها هستند، بلکه درباره بهبود خود کتابخانه اصلی نیز بحث و تبادل نظر می کنند. کاربران و توسعه دهندگان می توانند روی بهبود اسناد پروژه رسمی نیز کمک کنند که می توان در <http://api.jquery.com> به آن دسترسی داشت.



بدون در نظر گرفتن همه تلاش‌های لازم برای مهندسی چنین سیستم قدرتمند و منعطفی، محصول نهایی کاملاً رایگان است. این پروژه متن باز دارای مجوز استفاده تحت MIT License است که اجازه استفاده رایگان از jQuery روی هر وب سایت را داده و می‌توان از آن حتی در نرم‌افزارهای غیر رایگان نیز استفاده کرد. اگر یک پروژه به آن نیاز داشته باشد، توسعه دهندگان می‌توانند مجوزهای jQuery را دوباره و تحت GNU Public License تولید کرده تا بتوان از آن در دیگر پروژه‌های متن باز دارای مجوز GNU استفاده کرد.

آشنایی با پلاگین‌های jQuery

پلاگین‌های jQuery عملکرد کتابخانه اصلی را بیشتر می‌کنند. بعضی از پلاگین‌ها آن قدر مفید و پرکاربرد هستند که من از آن‌ها در این کتاب استفاده کردم. تعداد زیادی پلاگین وجود دارد (که البته کیفیت آن‌ها متغیر است) بنابراین اگر پلاگین‌های توصیف شده در این کتاب مناسب شما نبودند، می‌توانید با خیال راحت از یک پلاگین دیگر استفاده کنید.

برای استفاده از این کتاب باید چه چیزهایی بدانم

قبل از خواندن این کتاب باید با اصول توسعه وب آشنا بوده و بدانید که HTML و CSS چگونه کار می‌کنند و البته مقداری دانش درباره جاوااسکریپت نیز داشته باشید. من فرض می‌کنم که روی HTML و CSS مسلط هستید اما مقداری جاوااسکریپت را در فصل ۲ توضیح می‌دهم. در این جا یک مرجع کامل برای جاوااسکریپت نمی‌بینید زیرا این کتاب درباره jQuery است نه جاوااسکریپت.

مثال‌های کتاب

این کتاب شامل تعداد زیادی مثال است. یکی از موارد جالب درباره jQuery این است که می‌توان یک کار را از چند روش مختلف انجام داد که به شما اجازه می‌دهد یک شیوه توسعه jQuery مخصوص به خود داشته باشید. برای دیدن روش‌های مختلف انجام کار، من از مثال‌های زیادی استفاده کردم. من سند HTML که با آن کار می‌کنید را یکبار در بعضی از فصل‌ها قرار می‌دهم تا همه چیز بر مبنای آن انجام شود. اولین مثال در هر فصل یک سند HTML کامل است.

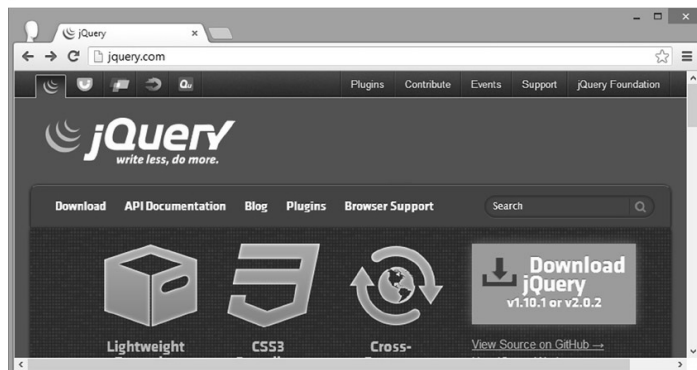
برای مثال‌های بعدی من تنها عناصری را نشان می‌دهم که تغییر می‌کنند. این معمولاً عنصر script است که کد jQuery شما در آن قرار می‌گیرد.

چه نرم‌افزارهایی برای این کتاب نیاز دارم؟

برای دنبال کردن مثال‌های این کتاب به تعدادی نرم‌افزار نیاز دارید که در ادامه توضیح داده می‌شوند.

دریافت jQuery

اولین چیزی که نیاز دارید کتابخانه jQuery است که در <http://jquery.com> قرار دارد. یک دکمه دانلود درون صفحه اول این وب سایت وجود دارد و یک گزینه نیز اجازه انتخاب توزیع Production یا Development را می‌دهد که در شکل ۱-۱ نشان داده شده است.



شکل ۱-۱

شما برای این کتاب به نسخه 2.x نیاز دارید و از نسخه‌های Development استفاده می‌کنید. تفاوت این دو توزیع را در فصل ۲ توضیح می‌دهم.

یک ویرایشگر HTML

یکی از مهم‌ترین ابزارها برای توسعه وب ویرایشگری است که بتواند اسناد HTML را ایجاد کند. HTML متن ساده است و بنابراین می‌توانید از ویرایشگرهای بسیار ساده مانند نت‌پد ویندوز نیز استفاده کنید اما برنامه‌های مخصوصی نیز هستند که امکان توسعه بهتر و روان‌تر را فراهم می‌آورند و البته بسیاری از آن‌ها رایگان هستند.

من قبلاً از Lomodo Edit از Active State استفاده می‌کردم که یک برنامه رایگان بوده که پشتیبانی خوبی از HTML، جاوااسکریپت و jQuery دارد. برای جزئیات بیشتر سری به <http://activestate.com> بزنید.

با این حال در این اواخر به ویژوال استودیوی مایکروسافت مهاجرت کردم. ویژوال استودیو پشتیبانی خوبی از HTML دارد. من از Visual Studio 2012 Express برای این کتاب استفاده کردم که به رایگان در آدرس زیر موجود است (البته یک نسخه غیر رایگان نیز وجود دارد اما مسلماً به آن امکانات بیشتر برای jQuery نیاز ندارید):

<http://www.microsoft.com/visualstudio>

یکی دیگر از برنامه‌های عالی که البته غیر رایگان است، ادابی دریم‌ویور (Adobe Dreamweaver) است که این برنامه نیز پشتیبانی عالی برای HTML، جاوااسکریپت و jQuery دارد و یک برنامه تخصصی برای توسعه وب است. برای جزئیات بیشتر به آدرس زیر مراجعه کنید:

www.adobe.com/products/dreamweaver.html

داشتن یک مرورگر وب

برای دیدن اسناد HTML خود و آزمایش jQuery و جاوااسکریپت به یک مرورگر وب نیاز دارید. من از گوگل کروم استفاده می‌کنم اما شما می‌توانید از هر مرورگری استفاده کنید. توصیه می‌کنم مرورگری را انتخاب کنید که دارای ابزارهای توسعه خوبی باشد. فایرفاکس دارای ابزارهای توسعه جاوااسکریپت عالی از طریق افزونه Firebug است (<http://getfirebug.com>).

اگر خواستید تا از اینترنت اکسپلورر استفاده کنید، از نسخه حداقل ۱۰ آن استفاده کنید که عملکرد بهتری دارد.



یک سرور وب

اگر بخواهید تا مثال‌های این کتاب را ایجاد کنید، به یک سرور وب نیز احتیاج خواهید داشت تا مرورگر شما بتواند اسناد HTML و منابع مرتبط را بارگذاری کند (مانند فایل‌های تصویر و جاوااسکریپت). تعداد زیادی سرور وب رایگان وجود دارد. مهم نیست چه سروری را انتخاب می‌کنید. من خودم از Internet Information Services (IIS) مایکروسافت استفاده می‌کنم. پیشنهاد دیگر XAMP است که یک پروژه منبع باز در آدرس زیر است: www.apachefriends.org

Node.js

در بعضی از فصل‌های کتاب من از Node.js به غیر از سرور وب عادی استفاده می‌کنم. Node.js در حال حاضر خیلی طرفدار دارد اما من بیشتر به خاطر سادگی و بر مبنای جاوااسکریپت بودن آن استفاده می‌کنم تا دیگر مجبور نباشید تا با فریم‌ورک برنامه وب دیگری کار کنید. البته وارد جزئیات آن نمی‌شویم (با این حال اسکریپت‌های سرور را به شما نشان می‌دهم تا بتوانید ببینید که چه اتفاقی در سرور می‌افتد).

می‌توانید Node.js را از آدرس <http://nodejs.org> دانلود کنید. یک فایل باینری از پیش کامپایل شده برای ویندوز وجود دارد و برای پلتفرم‌های دیگر می‌توانید از کد منبع آن استفاده کنید. در این کتاب من از نسخه 0.10.13 استفاده می‌کنم که مسلماً در زمان خواندن این کتاب نسخه‌های جدیدتری از آن ارائه شده است.

آماده کردن و آزمایش Node.js

ساده‌ترین روش برای آزمایش Node.js استفاده از یک اسکریپت ساده است. محتویات لیست ۱-۱ را درون یک فایل با نام NodeTest.js ذخیره کنید. من این را در پوشه یکسانی با فایل باینری Node.js قرار می‌دهم.

لیست ۱-۱: اسکریپت آزمایش Node.js.

```
var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
  console.log("Request: " + req.method + " to " + req.url);

  res.writeHead(200, "OK");
  res.write("<h1>Hello</h1>Node.js is working");
  res.end();
}).listen(80);
console.log("Ready on port 80");
```

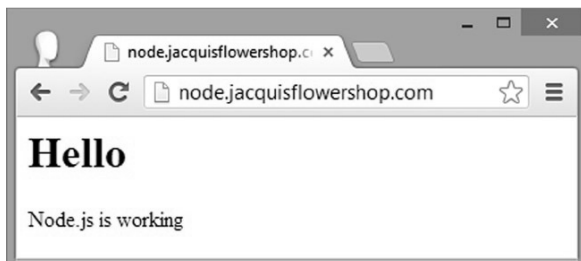
این یک اسکریپت آزمایشی بسیار ساده است و هنگام دریافت یک درخواست GET برای HTTP یک قطعه HTML بر می‌گرداند.

برای آزمایش Node.js باینری را اجرا کرده و فایل ایجاد شده را به‌عنوان آرگومان برای آن مشخص کنید. برای نصب ویندوز من، دستور زیر را در خط فرمان و در پوشه حاوی Node.js تایپ می‌کنم:

```
node NodeTest.js
```



برای این که همه چیز کار کند، به پورت 80 روی کامپیوتری بروید که Node.js روی آن اجرا می‌شود. باید چیزی مشابه با شکل ۲-۱ را ببینید که نشان می‌دهد همه چیز به درستی کار می‌کند.



شکل ۲-۱

من Node.js را روی کامپیوتری متفاوت از سرور وب عادی نصب کردم و بنابراین پورت 80 باعث مشکلی برای من نمی‌شود. اگر تنها یک کامپیوتر دارید، سرور وب را روی پورت 80 اجرا کرده و اسکریپت Node.js را برای یک پورت دیگر تغییر دهید. قسمتی که باید تغییر دهید را در لیست ۱-۱ به صورت توپر مشخص کردم که پورت استفاده شده را مشخص می‌کند.