

به نام ایزد یکتا

مرجع کامل

Pro jQuery 2

(جلد ۱)

Adam Freeman

ترجمه: مهندس خلیل غفوری

انتشارات پندار پارس

فهرست

فصل ۱؛ درج JQUERY در محتوا.....	۱
فهم واسط کاربری jQuery UI و jQuery موبایل.....	۱
فهم پلاگین‌های jQuery.....	۲
چه چیزی لازم است بدانید.....	۲
ساختار کتاب چیست.....	۲
بخش ۱: آماده سازی.....	۲
بخش ۲: کار با jQuery.....	۲
بخش ۳: کار با دیتا و آژاکس.....	۲
بخش ۴: ویژگی‌های پیشرفته.....	۲
بخش ۵: استفاده از واسط کاربری jQuery.....	۳
بخش ۶: استفاده از jQuery موبایل.....	۳
آیا مثال‌های زیادی وجود دارد؟.....	۳
از کجا می‌توان کد مثال را به دست آورد.....	۵
برای این کتاب چه نرم‌افزاری مورد نیاز است.....	۵
گرفتن jQuery.....	۵
گرفتن محیط ویرایشگر HTML.....	۶
گرفتن یک مرورگر وب.....	۶
گرفتن وب سرور.....	۷
گرفتن Node.js.....	۷
تنظیم و تست Node.js.....	۷
اسناد تصویر.....	۸
فصل ۲؛ مبانی HTML.....	۹
معرفی یک سند HTML پایه.....	۹
فهم چهارچوب یک عنصر HTML.....	۱۱
فهم صفات.....	۱۱
صفات id و class.....	۱۲
استفاده از صفت id.....	۱۲
استفاده از صفت class.....	۱۲
فهم محتوای عنصر.....	۱۳
فهم عناصر خالی (void).....	۱۳
فهم ساختار سند.....	۱۴
فهم عناصر متادیتا.....	۱۵
فهم عنصر script.....	۱۵
فهم عنصر style.....	۱۶
فهم عناصر محتوی.....	۱۷
فهم جداسازی معنا / نمایش.....	۱۷

۱۹.....	فهم فرم‌ها و ورودی‌ها.....
۲۰.....	فهم عناصر ساختاری.....
۲۱.....	فهم عناصر با منابع بیرونی.....
۲۲.....	فهم سلسله مراتبی عنصر.....
۲۲.....	فهم ارتباط پدر - فرزندی.....
۲۳.....	فهم روابط جد-نسل.....
۲۳.....	فهم روابط برادری.....
۲۳.....	فهم مدل شیء گرایى سند (DOM).....
۲۳.....	استفاده از DOM.....
۲۶.....	اصلاح DOM.....
۲۷.....	اصلاح style.....
۲۸.....	اداره رویدادها.....
۲۹.....	فهم جریان رویداد.....
۳۰.....	فهم عملیات پیش فرض.....
۳۱.....	فصل ۳؛ مبانی CSS.....
۳۱.....	آغاز کار با CSS.....
۳۳.....	تنظیم به‌صورت مقدار درون خطی.....
۳۴.....	تعریف سبک همراه.....
۳۶.....	تعریف CSS بیرونی.....
۳۸.....	مفهوم انتخابگرهای CSS.....
۳۹.....	انتخاب با صفت.....
۴۰.....	انتخاب به‌وسیله‌ی روابط.....
۴۲.....	انتخاب با استفاده از انتخابگرهای شبه عنصر و شبه کلاس.....
۴۴.....	اجتماع و نفی انتخابگرها.....
۴۵.....	مفهوم آشناری سبک.....
۴۵.....	مفهوم آشناری بودن سبک‌ها.....
۴۶.....	جابه‌جایی اولویت در ترتیب سبک‌ها.....
۴۷.....	شکستن برابری اولویت.....
۴۹.....	مفهوم واحدهای CSS.....
۴۹.....	کار با رنگ‌های CSS.....
۵۰.....	تعیین رنگ‌های پیچیده.....
۵۱.....	مفهوم طول در CSS.....
۵۱.....	کار با طول‌های مستقل.....
۵۲.....	کار با واحدهای وابسته.....
۵۳.....	کار با پیکسل‌ها.....
۵۴.....	کار با درصد.....
۵۵.....	استفاده از ویژگی‌های مختصر و مقادیر عرف.....
۵۹.....	فصل ۴؛ مبانی جاوااسکریپت.....

۵۹.....	آغاز جاوااسکریپت.....
۶۰.....	استفاده از دستورات
۶۱.....	تعریف و استفاده از فانکشن
۶۲.....	تعریف توابع با پارامتر
۶۲.....	تعریف فانکشن‌ها با خروجی.....
۶۳.....	استفاده از متغیرها و انواع داده.....
۶۴.....	استفاده از انواع داده اصلی.....
۶۴.....	کار با رشته‌ها.....
۶۴.....	کار با مقادیر منطقی.....
۶۵.....	کار با اعداد.....
۶۵.....	ساخت اشیاء.....
۶۶.....	استفاده از فانکشن‌ها به عنوان متد
۶۷.....	کار با اشیاء.....
۶۷.....	خواندن و اصلاح مقدار خصوصیات.....
۶۸.....	دسترسی و شمردن یک‌یک خصوصیات شیء.....
۶۹.....	حذف و اضافه خصوصیات و متدها
۷۰.....	تست موجود بودن خصوصیت در شیء.....
۷۱.....	استفاده از عملگرهای جاوااسکریپت.....
۷۱.....	استفاده از دستورات شرطی
۷۲.....	عملگر تساوی و تعیین هویت
۷۵.....	تبدیل نوع داده
۷۵.....	تبدیل اعداد به رشته.....
۷۶.....	تبدیل رشته به عدد
۷۷.....	کار با آرایه‌ها.....
۷۸.....	آرایه‌های دقیق
۷۸.....	خواندن و اصلاح محتوای آرایه.....
۷۹.....	دسترسی و شمردن محتوای آرایه.....
۷۹.....	استفاده از متدهای آرایه
۸۰.....	مدیریت خطا.....
۸۱.....	مقایسه مقادیر تهی و تعریف نشده‌ها.....
۸۳.....	تست null یا undefined بودن متغیر یا خصوصیت.....
۸۴.....	تفاوت بین null و undefined.....
۸۷.....	فصل ۵؛ اصول JQUERY.....
۸۸.....	تنظیمات jQuery.....
۹۰.....	استفاده از توضیحات مشروط
۹۰.....	نخستین اسکریپت jQuery.....
۹۲.....	فهم فانکشن \$.....
۹۴.....	انتظار برای مدل شیء سند (DOM).....

۹۵.....	فراموش کردن فانکشن.....
۹۵.....	استفاده از عبارات جایگزین.....
۹۶.....	به تعویق انداختن رویداد ready.....
۹۷.....	انتخاب عنصر.....
۹۹.....	محدودیت انتخاب با متن.....
۱۰۰.....	فهم نتیجه انتخاب.....
۱۰۰.....	تعیین محتوا.....
۱۰۱.....	تعامل با اشیای DOM.....
۱۰۱.....	ساخت اشیای jQuery از اشیای DOM.....
۱۰۲.....	رفتار با شیء jQuery به شکل آرایه.....
۱۰۳.....	تکرار فانکشن روی اشیای DOM.....
۱۰۳.....	جست‌وجوی ایندکس‌ها و عناصر خاص.....
۱۰۵.....	اصلاح چندین عنصر و فراخوانی زنجیره‌های متدها.....
۱۰۷.....	مدیریت رویدادها.....
۱۰۹.....	فصل ۶؛ مدیریت انتخاب عنصر.....
۱۱۰.....	توسعه‌ی انتخاب.....
۱۱۲.....	کاهش انتخاب.....
۱۱۳.....	کاهش انتخاب به یک عنصر معین.....
۱۱۳.....	کاهش انتخاب به یک بازه.....
۱۱۴.....	فیلترینگ عناصر.....
۱۱۷.....	کاهش انتخاب بر اساس فرزندان.....
۱۱۸.....	انطباق انتخاب.....
۱۱۹.....	تست انتخاب.....
۱۱۹.....	تغییر انتخاب و لغو آن.....
۱۲۱.....	مرور DOM.....
۱۲۱.....	مرور سلسله مراتبی رو به پایین.....
۱۲۳.....	استفاده از متد find برای ایجاد حالت ضربدری.....
۱۲۴.....	مرور سلسله مراتبی رو به بالا.....
۱۲۵.....	انتخاب عناصر پدر.....
۱۲۶.....	انتخاب اجداد.....
۱۲۸.....	انتخاب نخستین جد منطبق شده.....
۱۳۰.....	مرور در عرض سلسله مراتبی.....
۱۳۲.....	انتخاب همه‌ی برادران.....
۱۳۳.....	انتخاب برادران بعدی و پیشین.....
۱۳۵.....	فصل ۷؛ اصلاح DOM.....
۱۳۶.....	ایجاد عنصر.....
۱۳۶.....	ایجاد عناصر با استفاده از فانکشن \$.....
۱۳۸.....	ایجاد عناصر جدید با کپی عناصر موجود.....

۱۳۹	ایجاد عناصر با استفاده از DOM API
۱۴۰	درج عناصر فرزند و نوادگان
۱۴۱	چسباندن عناصر
۱۴۳	درج عناصر مشابه در مکان‌های متفاوت
۱۴۵	درج از شیء jQuery
۱۴۶	درج عناصر با استفاده از فانکشن
۱۴۷	درج عناصر پدر و جد
۱۴۹	بسته بندی عناصر با هم
۱۵۱	بسته بندی محتوایی عناصر
۱۵۲	دسته بندی عناصر با استفاده از فانکشن
۱۵۳	درج عناصر برادر
۱۵۴	درج برادران از یک شیء jQuery
۱۵۵	درج برادران با استفاده از فانکشن
۱۵۶	جایگزینی عناصر
۱۵۷	جایگزینی عناصر با استفاده از فانکشن
۱۵۸	حذف عناصر
۱۵۹	جداسازی عناصر
۱۶۰	تهی سازی عناصر
۱۶۱	بازکردن عناصر از پوشش
۱۶۳	فصل ۸؛ اصلاح عناصر
۱۶۴	کار با صفات و ویژگی‌ها
۱۶۶	تنظیم مقدار صفت
۱۶۷	تنظیم چند صفت
۱۶۸	تنظیم مقادیر صفت به گونه‌ای دینامیک
۱۶۹	حذف صفت
۱۷۰	کار با ویژگی‌ها
۱۷۰	کار با کلاس‌ها
۱۷۲	اضافه و حذف کلاس‌ها با استفاده از فانکشن
۱۷۴	تغییر وضعیت کلاس‌ها
۱۷۶	تغییر وضعیت چندین کلاس
۱۷۷	تغییر وضعیت همه‌ی کلاس‌ها
۱۷۸	تغییر وضعیت در یک جهت
۱۷۹	تغییر وضعیت کلاس به صورت دینامیک
۱۸۰	کار با CSS
۱۸۱	تنظیم چندین ویژگی CSS
۱۸۲	تنظیم چند ویژگی CSS
۱۸۳	تنظیم مقادیر نسبی
۱۸۴	تنظیم ویژگی با استفاده از فانکشن

استفاده از متدهای متناظر صفات خاص CSS	۱۸۴
تنظیم عرض و ارتفاع با استفاده از فانکشن	۱۸۶
کار با محتوای عنصر	۱۸۷
تنظیم محتوای عنصر	۱۸۷
تنظیم محتوای عنصر با استفاده از فانکشن	۱۸۸
کار با عناصر form	۱۸۹
تنظیم مقادیر عناصر form	۱۹۰
تنظیم مقادیر عناصر form با استفاده از فانکشن	۱۹۱
ارتباط داده با عناصر	۱۹۱
فصل ۹؛ کار با رویدادها	۱۹۵
مدیریت رویداد	۱۹۶
ثبت یک فانکشن برای مدیریت چندین نوع رویداد	۱۹۹
ارائه‌ی دیتا به فانکشن مدیر رویداد	۲۰۰
منع عمل پیش فرض	۲۰۱
حذف فانکشن‌های مدیر رویداد	۲۰۲
حذف مدیر رویداد از داخل فانکشن مدیر رویداد	۲۰۳
اجرای مدیر رویداد تنها یکبار	۲۰۳
انجام ثبت رویداد به صورت زنده	۲۰۴
محدودیت پیمایش DOM برای مدیران رویداد زنده	۲۰۷
فراخوانی دستی مدیران رویداد	۲۰۸
استفاده از شیء Event	۲۰۹
استفاده از متد triggerHandler	۲۱۰
استفاده از متدهای چکیده رویداد	۲۱۱
استفاده از متدهای چکیده رویداد سند	۲۱۲
استفاده از متدهای چکیده رویداد مرورگر	۲۱۳
استفاده از متدهای چکیده ماوس	۲۱۳
استفاده از متدهای چکیده رویداد فرم	۲۱۴
استفاده از متدهای چکیده رویداد صفحه کلید	۲۱۵
فصل ۱۰؛ استفاده از افکت‌های JQUERY	۲۱۷
استفاده از افکت‌های پایه	۲۱۸
تغییر وضعیت عناصر	۲۲۱
تغییر وضعیت در یک جهت	۲۲۲
انیمیشن‌سازی عناصر نمایشی	۲۲۲
استفاده از پاسخ به تماس‌های افکت	۲۲۴
ایجاد افکت‌های حلقه‌ای	۲۲۶
استفاده از افکت‌های اسلاید	۲۲۸
استفاده از افکت‌های fade	۲۲۹
محو کردن تا سطح شفافیت مشخص	۲۳۰

۲۳۲	ایجاد افکت‌های شخصی
۲۳۳	استفاده از مقادیر مطلق ویژگی مقصد
۲۳۴	استفاده از مقادیر نسبی ویژگی مقصد
۲۳۵	ایجاد و مدیریت صفت افکت
۲۳۶	نمایش آیتم‌های صف افکت
۲۳۷	توقف افکت‌ها و خالی کردن صف
۲۴۱	ایجاد وقفه در صف
۲۴۱	درج فانکشن‌ها درون صف
۲۴۲	فعال و غیرفعال‌سازی انیمیشن‌های افکت
۲۴۵	فصل ۱۱؛ عملیاتی کردن دوباره مثال بخش ۱
۲۴۵	بازنگری سند مثال
۲۴۷	افزودن گل‌های بیشتر
۲۴۸	افزودن دکمه‌های چرخ فلکی
۲۵۱	رفتار با دکمه submit
۲۵۲	پیاده‌سازی فانکشن‌های مدیر رویداد چرخ فلکی
۲۵۴	جمع محصولات انتخابی
۲۵۶	جاوااسکریپت غیر فعال
۲۵۹	فصل ۱۲؛ استفاده از الگوهای داده
۲۶۰	فهم مسأله‌ی قابل حل با الگوها
۲۶۱	تنظیم کتابخانه‌ی الگوی داده
۲۶۲	نخستین مثال الگوی داده
۲۶۴	تعریف دیتا
۲۶۴	تعریف الگو
۲۶۵	اعمال الگو
۲۶۶	تقسیم نتیجه
۲۶۸	تقسیم ورودی
۲۶۹	استفاده از الگوی منطقی
۲۶۹	ایجاد محتوای شرطی
۲۷۱	بررسی تک تک آراییه‌ها و ویژگی‌ها
۲۷۳	تغییر محتوای داده
۲۷۵	استفاده از کمک الگوی #with
۲۷۵	دسترسی به محتوای داده والدین
۲۷۶	ایجاد کمک الگوهای شخصی
۲۷۸	ایجاد یک کمک الگوی شخصی
۲۸۰	برگرداندن محتوا با پیچیدگی بیشتر
۲۸۱	دریافت آرگومان‌های اضافی در فانکشن کمک الگو
۲۸۳	آماده‌سازی ویژگی‌های الگوی شخصی
۲۸۵	فصل ۱۳؛ کار با فرم‌ها

۲۸۶	آماده سازی سرور Node.js
۲۸۹	پوشش متدهای رویداد فرم
۲۹۰	برخورد با فوکوس فرم
۲۹۲	برخورد با تغییر مقدار
۲۹۳	برخورد با ارسال فرم
۲۹۴	اعتبارسنجی مقادیر فرم
۲۹۶	دریافت فایل جاوااسکریپت
۲۹۷	پیکربندی اعتبارسنجی
۲۹۷	تعریف قوانین اعتبارسنجی
۲۹۸	اعمال قوانین اعتبارسنجی
۲۹۸	استفاده از تست‌های اعتبارسنجی
۲۹۹	اعمال قوانین اعتبارسنجی در طریق کلاس‌ها
۳۰۲	اعمال قوانین اعتبارسنجی به صورت مستقیم بر روی عناصر
۳۰۵	اعمال قوانین اعتبارسنجی بر اساس صفت name عنصر
۳۰۶	اعمال قوانین اعتبارسنجی بر اساس صفات عناصر
۳۰۷	تعیین پیغام‌های اعتبارسنجی
۳۰۷	تعریف پیغام برای اعتبارسنجی نام و صفت
۳۱۰	تعریف پیغام‌های اعتبارسنجی به ازای هر عنصر
۳۱۱	ایجاد یک تست شخصی
۳۱۲	تعریف فانکشن اعتبارسنجی
۳۱۳	تعریف پیغام اعتبارسنجی
۳۱۴	قالب‌بندی نمایشی خطای اعتبارسنجی
۳۱۵	تنظیم کلاس برای عناصر نامعتبر
۳۱۵	تنظیم عنصر پیغام خطا
۳۱۶	تنظیم تاکید برای عناصر نامعتبر
۳۱۷	استفاده از اعتبارسنجی چکیده
۳۱۸	آماده سازی پیغام‌های اعتبارسنجی
۳۱۹	ایجاد اعتبارسنجی چکیده
۳۲۰	مرتب کردن پیغام خطای ترکیب
۳۲۳	فصل ۱۴؛ استفاده از آژاکس – بخش ۱
۳۲۴	استفاده از متدهای مختصر آژاکس
۳۲۵	انجام یک درخواست GET آژاکس
۳۲۸	پردازش دیتای جواب
۳۲۸	ساخت افکت ساده‌تر برای دیدن
۳۲۹	گرفتن انواع دیگر داده
۳۳۱	آماده‌سازی دیتا برای درخواست‌های GET
۳۳۲	انجام یک درخواست POST آژاکس
۳۳۲	آماده سازی Node.js برای دریافت دیتا

۳۳۴	فهم درخواست‌های آژاکس چند منطقه‌ای.....
۳۳۵	استفاده از متد post برای ارسال دیتای فرم.....
۳۳۹	ارسال دیتای دیگر با استفاده از متد post.....
۳۳۹	تعیین نوع داده‌ی مورد انتظار.....
۳۴۱	پرهیز از شایع‌ترین اشتباه آژاکس.....
۳۴۲	استفاده از متدهای مناسب نوع داده خاص.....
۳۴۲	گرفتن یک قطعه ی HTML.....
۳۴۳	اصلاح عناصر افزوده شده با متد load.....
۳۴۴	گرفتن و اجرای اسکریپت.....
۳۴۶	گرفتن دیتای JSON.....
۳۴۷	کار با JSONP.....
۳۴۸	استفاده از پلاگین فرم‌های آژاکس.....
۳۵۱	فصل ۱۵؛ استفاده از آژاکس - بخش ۲.....
۳۵۲	ساخت یک درخواست آژاکس ساده با API سطح پایین.....
۳۵۴	فهم شیء jqXHR.....
۳۵۵	تنظیم URL درخواست.....
۳۵۶	ساخت یک درخواست post.....
۳۵۷	مدیریت رویدادهای آژاکس.....
۳۵۸	برخورد با درخواست‌های موفقیت آمیز.....
۳۵۹	برخورد با خطاها.....
۳۶۱	برخورد با درخواست‌های کامل شده.....
۳۶۲	پیکربندی درخواست‌ها پیش از ارسال.....
۳۶۳	تعیین چندین فانکشن مدیر رویداد.....
۳۶۴	تنظیم متن برای رویداد.....
۳۶۵	استفاده از رویدادهای عمومی آژاکس.....
۳۶۷	کنترل رویدادهای عمومی.....
۳۶۸	پیکربندی تنظیمات پایه برای درخواست آژاکس.....
۳۶۹	تنظیم timeout و هدر.....
۳۷۰	ارسال دیتای JSON به سرور.....
۳۷۱	استفاده از تنظیمات پیکربندی پیشرفته.....
۳۷۱	ساخت درخواست همزمان.....
۳۷۲	نادیده گرفتن دیتای اصلاح نشده.....
۳۷۳	برخورد با کد وضعیت پاسخ.....
۳۷۴	پردازش دیتای پاسخ.....
۳۷۶	مدیریت تبدیل داده.....
۳۷۷	تنظیم و فیلتر کردن درخواست‌های آژاکس.....
۳۷۷	تعریف تنظیمات به صورت پیش فرض.....
۳۷۸	فیلتر کردن درخواست‌ها.....

فصل ۱۶؛ عملیاتی کردن دوباره مثال بخش ۲.....	۳۸۱
بازنگری سند مثال.....	۳۸۱
اصلاح اسکریپت Node.js.....	۳۸۴
آماده‌سازی برای آژاکس.....	۳۸۶
منبع اطلاعات محصول.....	۳۸۸
افزودن اعتبارسنجی فرم.....	۳۹۱
افزودن اعتبارسنجی دور.....	۳۹۵
ارسال دیتای فرم با استفاده از آژاکس.....	۳۹۷
پردازش پاسخ سرور.....	۴۰۳
افزودن فرم جدید.....	۴۰۹
تکمیل درخواست آژاکس.....	۴۱۰
پردازش دیتا.....	۴۱۱
فصل ۱۷؛ استفاده از متدهای سودمند JQUERY.....	۴۱۵
ملاقات دوباره صف‌ها: استفاده از صف‌های چندمنظوره.....	۴۱۶
پردازش دستی آیتم‌های صف.....	۴۱۸
متدهای سودمند آرایه‌ها.....	۴۱۹
استفاده از متد grep.....	۴۲۰
استفاده از متد inArray.....	۴۲۱
استفاده از متد map.....	۴۲۲
استفاده از متد merge.....	۴۲۳
استفاده از متد unique.....	۴۲۴
متدهای سودمند برای نوع.....	۴۲۴
استفاده از متد type.....	۴۲۵
متدهای سودمند برای دیتا.....	۴۲۶
نظم به داده‌های فرم.....	۴۲۷
تجزیه و تحلیل دیتا.....	۴۲۷
پیرایش رشته‌ها.....	۴۲۸
دیگر متدهای سودمند.....	۴۲۹
کنترل و بررسی محتوای عنصر.....	۴۲۹
فصل ۱۸؛ افکت‌های JQUERY UI و فریم‌ورک CSS.....	۴۳۱
استفاده از افکت‌های jQuery UI.....	۴۳۱
انیمیشن‌سازی رنگ‌ها.....	۴۳۲
انیمیشن‌سازی کلاس‌ها.....	۴۳۴
سوئیچ کردن کلاس‌ها.....	۴۳۵
استفاده از انیمیشن‌های jQuery UI.....	۴۳۶
استفاده از افکت‌ها برای نمایش و مخفی سازی عناصر.....	۴۳۷
اعمال افکت‌های رایج.....	۴۳۸
استفاده از فریم ورک CSS متعلق به jQuery UI.....	۴۳۹

۴۳۹ استفاده از کلاس‌های دربرگیرنده ویجت
۴۴۰ اعمال گوشه‌های گردشده
۴۴۲ استفاده از کلاس‌های وضعیت تعاملی
۴۴۴ استفاده از کلاس‌های اشاره
۴۴۹ فصل ۱۹؛ استفاده از اشیای معوقه
۴۵۰ نخستین مثال شیء معوقه
۴۵۲ فهم چرایی مفید بودن شیء معوقه
۴۵۵ سروسامان دادن به مثال
۴۵۸ استفاده از پاسخ به تماس‌های دیگر
۴۵۸ رد یک شیء معوقه
۴۶۱ فراخوانی زنجیروار متد شیء معوقه
۴۶۱ پوشش هر دو خروجی
۴۶۲ استفاده از پاسخ به تماس‌های خروجی بی اثر
۴۶۴ استفاده از چند پاسخ به تماس
۴۶۶ استفاده از خروجی اشیای معوقه چندگانه
۴۶۸ آماده‌سازی اطلاعات پیشرفت
۴۷۰ گرفتن اطلاعات درباره‌ی شیء معوقه
۴۷۳ استفاده از اشیای معوقه آژاکس

فصل 1

درج jQuery در محتوا

شاید jQuery در آغاز کسل کننده به نظر برسد. اجازه می‌دهد تا محتوای اسناد HTML را با دستکاری در مدل (که مرورگر در هنگام پردازش HTML ایجاد می‌کند و به آن DOM گفته می‌شود) اصلاح کنید. احتمالاً تاکنون DOM را دستکاری کرده‌اید، تفاوتی ندارد، اکنون با استفاده از کتابخانه جاوااسکریپت یا با استفاده از API که بیشتر مرورگرهای وب جدید از آن حمایت می‌کنند این کار را می‌کنید. اینک که این کتاب را برداشته‌اید و قصد مطالعه آن را دارید قطعاً می‌خواهید آن را از راه بهتری انجام دهید.

هر چه جلوتر می‌رویم وضع jQuery بهتر می‌شود. jQuery با دستکاری در DOM، یک لذت، فرصت و شادابی واقعی را ایجاد می‌کند. نکته ظریف و جالب توجه آن این است که jQuery کارهای سخت را به مسائل ساده و آسان تبدیل می‌کند. در این ظرافتی وجود دارد که دیگر راه برگشتی وجود ندارد. البته من به چند دلیل در پروژه‌هایم از jQuery استفاده می‌کنم:

- در هنگام استفاده از مرورگرهای DOM API کارهایم را با کمینه کد می‌توانم انجام دهم.
 - متدهای jQuery بر روی چندین عنصر انجام می‌شود. روش انتخاب، بررسی و اصلاح دیگر جایگاهی ندارد بدین معنی که با کمینه کردن حلقه‌های for برای بررسی عناصر، اشتباه کمتری هم به وجود می‌آید.
 - jQuery بین مرورگرها با اجرای متفاوت عمل می‌کند. من دیگر نگران نیستم آیا IE از ویژگیهای جدید پشتیبانی می‌کند یا خیر؟ برای نمونه، تنها به jQuery می‌گویم که چه می‌خواهم و او با توجه به تفاوت‌ها پیاده‌سازی مناسب انجام می‌دهد.
 - jQuery متن باز است. هنگامیکه از چگونگی کار چیزی نمی‌فهمم یا از نتیجه کار رضایت ندارم می‌توانم کد جاوااسکریپت متناظر آن را بخوانم و چنانچه لازم بود آن را تغییر دهم.
- البته درست است هیچ چیزی کامل نیست، موارد و مشکلاتی هم وجود دارد که بعداً شرح آن را خواهم داد. حتی با وجود این اشکالات jQuery را دوست دارم و امیدوارم شما هم سریع آن را بیابید و لذت ببرید. به نظر من jQuery مانند نابغه‌ای است که در بحث توسعه وب، مواردی که خیلی وقت‌گیر هستند را بسیار ساده و روان می‌سازد.

فهم واسط کاربری jQuery UI و jQuery موبایل

افزون بر هسته کتابخانه jQuery، jQuery UI و jQuery موبایل را نیز شرح خواهم داد. گفتنی است کتابخانه‌های واسط کاربری از روی jQuery ساخته شده‌اند. jQuery UI برای اهداف عمومی در نظر گرفته شده تا با هر دستگاهی کار کند و jQuery موبایل برای استفاده در دستگاه‌های لمسی همچون گوشی موبایل و تبلت‌ها به کار گرفته می‌شود.

فهم پلاگین‌های jQuery

پلاگین‌ها توانایی‌های کتابخانه اصلی را توسعه داده‌اند. شماری از آنها را که به‌گونه‌ای وسیع مورد استفاده قرار می‌گیرد، در این کتاب پوشش داده‌ام. شمار زیادی از اینها در دسترس هستند (اگرچه از لحاظ کیفیت متفاوت هستند). چنانچه پلاگین‌های مورد بحث خوشایند شما نبود مطمئن باشید راه‌های جایگزین دیگری وجود دارد:

-1 jQuery UI

-2 jQuery Mobile

چه چیزی لازم است بدانید

پیش از مطالعه کتاب باید با اصول وب آشنا باشید. کار با CSS و HTML را بدانید. از جاوااسکریپت اطلاعاتی داشته باشید. اگر چنانچه در هر یک از این موارد ابهاماتی دارید، در فصول 2 و 3 و 4 مطالبی را جهت یادآوری آماده کرده‌ام. ممکن است مرجع خاصی برای عناصر HTML و ویژگی‌های CSS پیدا نکنید. کتاب jQuery هم فضای خوبی درباره‌ی پرداختن به HTML، CSS و پوشش مطالب مرتبط با آن نیست. چنانچه لازم است یک مرجع کامل درباره‌ی HTML و CSS داشته باشید پیشنهاد می‌کنم کتاب دیگر من با عنوان "The Defenitive Guide to HTML5" از همین ناشر را ببینید.

ساختار کتاب چیست

این کتاب در 2 جلد و در مجموع به 6 بخش تقسیم شده است که هر بخش مجموعه‌ای از موضوعات مرتبط را پوشش می‌دهد.

بخش 1: آماده سازی

بخش 1 این کتاب اطلاعاتی که در ادامه کتاب لازم است بدانید، آماده می‌کند؛ شامل مطالب اساسی و یادآوری‌های HTML، CSS و جاوااسکریپت. در ادامه فصل نرم‌افزارهایی که در ادامه نیاز دارید معرفی می‌شود.

بخش 2: کار با jQuery

بخش 2 این کتاب معرفی کتابخانه jQuery است. با مثال پایه آغاز می‌شود و بر روی آن هر یک از ویژگی‌های هسته مثل انتخاب عنصر، دستکاری DOM، رویدادها و افکت‌ها به مرور ساخته می‌شود.

بخش 3: کار با دیتا و آژاکس

بخش 3 این کتاب نشان می‌دهد چگونه jQuery راه را برای کار با دیتای درون خط و دیتای دور فراهم می‌سازد. نشان می‌دهد چگونه از دیتا محتوای HTML را بسازد. چگونه اعتبار داده‌های ورودی را بسنجد و چگونه با استفاده از jQuery، عملیات غیرهمزمان با دخالت آژاکس انجام شود.

بخش 4: ویژگی‌های پیشرفته

آخرین بخش از جلد 1 کتاب درباره‌ی برخی از ویژگی‌های jQuery و jQuery UI است که معمولاً مورد استفاده عموم نیست. اما در پروژه‌های پیچیده مفید هستند. این ویژگی‌های پیشرفته نیازمند درک بهتر از HTML، CSS و خود jQuery است. درخصوص فصل 18 یک دانش بنیانی از برنامه نویسی غیرهمزمان می‌تواند بسیار مفید باشد.

بخش 5: استفاده از واسط کاربری jQuery

jQuery UI یکی از دو کتابخانه واسط کاربری است که شرح آن را خواهیم داد. کتابخانه‌ای که از روی هسته jQuery ساخته شده و اجازه می‌دهد تا برای برنامه‌های وب، واسطی پاسخگو و با کیفیت بسازید.

بخش 6: استفاده از jQuery موبایل

jQuery موبایل واسط کاربری دیگری است که توضیح خواهیم داد. از روی jQuery ساخته شده و شامل ویژگی‌های پایه jQuery UI است که برای واسط‌های کاربری گوشی‌های هوشمند و تبلت‌ها بهینه‌سازی شده است. ویجت‌های واسط کاربری در jQuery موبایل کمتر در دسترس هستند اما از آنهایی که برای استفاده در وسایل لمسی در نظر گرفته شده‌اند استفاده کرده، آنها را بهینه نموده تا در وسایلی با نمایشگرهای کوچک استفاده شوند.

آیا مثال‌های زیادی وجود دارد؟

یکی از جنبه‌های خوب jQuery این است که هر کاری می‌تواند به روش‌های مختلفی انجام شود (بدلیل اجازه داشتن در طراحی یک سبک jQuery شخصی). برای نشان دادن روش‌های متفاوت که در اختیار شما است، مثال‌های زیادی انتخاب کرده‌ایم. در واقع در هر مورد کاری در هر فصل یکبار، یک سند کامل HTML آورده‌ایم. نخستین مثال در هر فصل با یک سند کامل آغاز می‌شود. همان‌گونه که در لیست 1-1 نشان داده شده است.

لیست 1-1: یک سند نمونه کامل

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <script src="jquery-2.0.2.js" type="text/javascript"></script>
  <link rel="stylesheet" type="text/css" href="styles.css"/>
  <script type="text/javascript">
    $(document).ready(function() {
      var labelElems = document.getElementsByTagName("label");
      var jq = $('img[src*=daffodil]');
      $('img:even').add('img[src*=primula]').add(jq)
        .add(labelElems).css("border", "thick double red");
    });
  </script>
</head>
<body>
  <h1>Jacqui's Flower Shop</h1>
  <form method="post">
    <div id="oblock">
      <div class="dtable">
        <div id="row1" class="drow">
          <div class="dcell">
            <label
              for="astor">Astor:</label>
            <input name="astor" value="0" required>
          </div>
          <div class="dcell">
```



```

<label
for="daffodil">Daffodil:</label>
<input name="daffodil" value="0" required >
</div>
<div class="dcell">
  <label
for="rose">Rose:</label>
  <input name="rose" value="0" required>
</div>
</div>
<div id="row2" class="drow">
<div class="dcell">
  <label
for="peony">Peony:</label>
  <input name="peony" value="0" required>
</div>
<div class="dcell">
  <label
for="primula">Primula:</label>
  <input name="primula" value="0" required>
</div>
<div class="dcell">
  <label
for="snowdrop">Snowdrop:</label>
  <input name="snowdrop" value="0" required>
</div>
</div>
</div>
</div>
<div id="buttonDiv"><button type="submit">Place Order</button></div>
</form>
</body>
</html>

```

این مثال از فصل 5 گرفته شده است. درباره‌ی کارکردش نگران نباشید. تنها خواستم نشان دهم که در آغاز هر فصل یک سند کامل آورده می‌شود.

تقریباً همه مثال‌ها حول یک سند پایه HTML هستند؛ یک گل‌فروشی ساده شاید مثال خیلی مهمی نباشد ولی این مثال همه آن چیزی که در هنگام کار با jQuery نیاز داریم و علاقمند به آن هستیم، در خود دارد.

برای مثال‌های بعدی در هر فصل تنها تغییرات را می‌آورم و معمولاً این تغییرات درباره‌ی عنصر script است که در قسمت کد jQuery قرار دارد. این تغییرات را با چند نقطه (...) که در آغاز و پایان کد است می‌توانید تشخیص دهید. مانند لیست 1-2

لیست 1-2: بخشی از لیست

```

...
<script type="text/javascript">
  jQuery(document).ready(function () {
    jQuery("img:odd").mouseenter(function(e) {
      jQuery(this).css("opacity", 0.5);
    });
  });

```

```
    }).mouseout(function(e) {  
        jQuery(this).css("opacity", 1.0);  
    });  
});  
</script>  
...
```

این لیست نیز از فصل 5 است. بخشی از کد را به صورت پررنگ می بینید. بدینوسیله تنها خواستم توجه شما را به این بخش از کد جلب کنم. در این قسمت جدا شده تنها عناصری که تغییر کرده اند نشان داده شده است.

در این کتاب با توجه به ویژگی های خاص، روی مثال ها تمرکز کردم. این بهترین راه برای پوشش دادن به چگونگی عملکرد jQuery است. اما در این روش ممکن است در اتصال این تکه کدها آشفتگی به وجود آید و به اصطلاح رشته کلام از دست برود. بنابراین در پایان هر بخش از کتاب فصل کوتاهی با نام عملیات دوباره مثال وجود دارد تا همه موضوع های فصول پیشین را یک جا و با هم در معرض دید قرار دهد.

از کجا می توان کد مثال را به دست آورد

می توانید تمامی مثال های کتاب را از سایت Aprees.com به دست آورید. دسترسی به مثال ها رایگان و شامل همه منابع پشتیبانی است (مانند عکس ها، کتابخانه های جاوا اسکریپت و CSS ها). کدها را می توانید دانلود کنید ولی یک راه ساده همیشگی یعنی کپی و paste کردن نیز وجود دارد.

نکته: حتی اگر در هر بخش و فصل، بخشی از کد آورده شود، می توانید از روی سند HTML که توسط مرورگر دانلود شده، نسخه کامل آن را به دست آورید.

برای این کتاب چه نرم افزاری مورد نیاز است

در ادامه به نرم افزارهایی که در زیر آورده شده نیاز دارید.

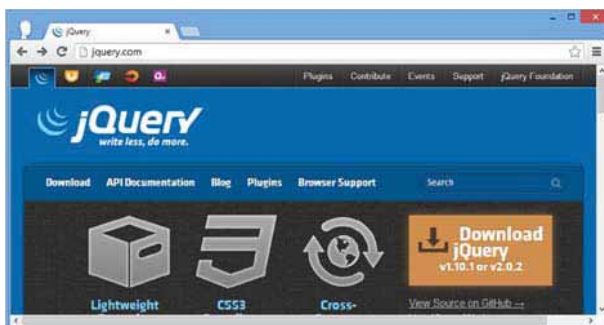
گرفتن jQuery

نخستین چیزی که نیاز دارید توابع jQuery است که در سایت <http://jquery.com> در دسترس است.

یک دکمه دانلود در صفحه آغازین وجود دارد که با انتخاب نگارش تولید یا توسعه مانند شکل 1-1 می توانید آن را به دست آورید.

در این کتاب از نسخه توسعه استفاده می شود. از نسخه 2.x در فصل های 1 تا 4 و 6 و از 1.x برای فصل 5 استفاده می شود. در ادامه، تفاوت این نسخه ها، و نحوه استفاده از کتابخانه jQuery در فصل 5 شرح داده می شود.

نکته: درباره ی نحوه به دست آوردن و نصب jQuery UI و jQuery موبایل در فصل های 20 و 30 توضیح خواهیم داد.



شکل 1-1: دانلود کتابخانه jQuery

گرفتن محیط ویرایشگر HTML

یکی از مهم‌ترین ابزارهایی که در بحث توسعه وب از آن یاد می‌شود، محیط ویرایشگری است که در آن می‌توان سند HTML را ایجاد کرد. HTML تنها یک متن است؛ بنابراین در یک محیط بسیار ساده نیز امکان ویرایش وجود دارد اما بسته‌های اختصاصی در دسترس هستند که امر توسعه و ایجاد را بسیار ساده کرده و بسیاری از اینها هم رایگان هستند.

Komodo Edit را پیشنهاد می‌کنم. رایگان و بسیار ساده است و پشتیبانی خوبی برای HTML، جاوااسکریپت و jQuery دارد. وابستگی خاصی به آن ندارم و خیلی‌ها از آن استفاده می‌کنند. می‌توانید آن را از آدرس <http://activestate.com> بردارید. البته نسخه‌های مخصوص لینوکس و Mac نیز وجود دارد.

به تازگی به ویژوال استادیو میکروسافت سویچ کردم و بسیاری از کدهایم را در آن نوشتم. پشتیبانی خوبی از ویرایشگر HTML دارد. در این کتاب از ویژوال استادیو اکسپرس نسخه 12 استفاده کردم که رایگان است. <http://www.microsoft.com/visualstudio> را ببینید.

گزینه دیگر JsFiddle است که یک ویرایشگر آنلاین است و از jQuery نیز پشتیبانی کرده است. من با آن کار نمی‌کنم (ساختار آن با روش توسعه من در تضاد است) ولی بسیار منعطف و قدرتمند به نظر می‌رسد. رایگان است و در آدرس <http://jsfiddle.net> در دسترس است.

نکته: چند پیشنهاد در این کتاب دادم که تنها به این دلیل نیست که خودم استفاده می‌کنم. بدانید که هیچ ارتباطی با میکروسافت، Active State یا دیگر کمپانی‌ها ندارم.

گرفتن یک مرورگر وب

برای کار، به یک مرورگر نیاز دارید تا مثال‌ها و مستندات HTML را ببینید و کد jQuery و جاوااسکریپت را تست کنید. گوگل کروم دوست‌داشتنی و سریع است. واسط کاربری ساده و ابزار توسعه زیبایی دارد. در هر جای کتاب که نگاه کنید گوگل کروم را خواهید دید. این بدین معنی نیست که از مرورگرهای دیگر استفاده نشود اما پیشنهاد من استفاده از یک ابزار خوب است. Mozilla Firefox برخی ابزارهای جاوااسکریپت خیلی خوب داخل Firebug در اختیار دارد که می‌توانید از سایت <http://getfirebug.com> بگیرید.

اگر از گوگل کروم یا فایرفاکس استفاده نمی‌کنید گزینه بعدی IE است. بسیاری از برنامه نویسان وب با IE مشکل دارند ولی نگارش 9 آن با توجه به تجربه من خوب است. نسخه 10 آن بسیار امیدوارکننده است. ابزارهای توسعه خیلی کامل نیستند ولی گوگل کروم و فایرفاکس برای اهداف این کتاب بسیار مناسب هستند.

گرفتن وب سرور

چنانچه بخواهید مثال‌های کتاب را بسازید نیاز به وب سرور دارید تا مرورگر بتواند مثال‌های سند HTML و منابع وابسته آن‌را (مقادیر و فایل‌های جاوااسکریپت) لود کند. وب سرورهای زیادی در دسترس هستند و بسیاری نیز منبع باز می‌باشند و البته رایگان. اینکه از کدام وب سرور استفاده می‌کنید اهمیتی ندارد. در این کتاب از IIS استفاده کرده‌ام. این تنها گزینه‌ی من است زیرا از سرور ویندوز استفاده می‌کنم.

Node.js گرفتن

در آغاز بخش 3 استفاده از Node.js را در کنار وب سرور آغاز خواهید کرد. Node.js بسیار محبوب است اما من بدلیل اینکه بر پایه‌ی جاوااسکریپت است از آن استفاده می‌کنم. لازم نیست وارد جزئیات Node.js شوم، بلکه با آن مانند یک جعبه سیاه برخورد می‌کنم. (اگرچه اسکریپت‌های سرور را به شما نشان می‌دهم تا شاهد اتفاق‌های روی سرور باشید، البته اگر علاقمند باشید). می‌توانید Node.js را از آدرس <http://nodejs.org> بگیرید. یک کد باینری پیش کامپایل شده برای ویندوز و کد منبعی که بتوان برای دیگر پلاتفرم‌ها آن را آماده کرد قابل دریافت است. در این کتاب من از نسخه 0,10,13 استفاده می‌کنم که احتمالاً در زمان خواندن این کتاب جایگزین دیگری آمده است. اما اسکریپت‌های سرور باید بتوانند بدون هیچ مسأله‌ای کار کنند.

تنظیم و تست Node.js

آسانترین راه برای تست Node.js استفاده از یک اسکریپت ساده است. محتوای لیست 3-1 را درون یک فایل به نام NodeTest.js ذخیره نمایید. من این را در همان شاخه Node.js قرار دادم.

لیست 3-1: اسکریپت تست 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");
```

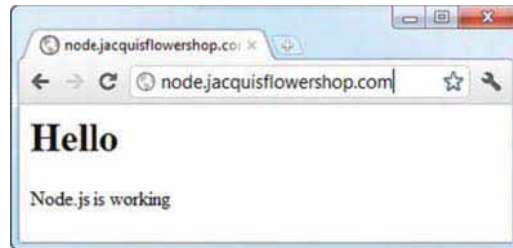
این کد یک اسکریپت تست ساده است. وقتی درخواستی از نوع Get دریافت کند، یک قطعه از HTML را برمی‌گرداند.

نکته: اگر از جمله آخر احساس خوبی ندارید نگران نباشید. هم اینک نیازی نیست بدانید چگونه HTTP و وب سرورها با استفاده از jQuery کار می‌کنند. در فصل 2 توضیح خواهیم داد.

برای تست Node.js فایل باینری خاص را - که به شکل یک آرگومان می‌باشد - اجرا کنید. به شکل ویندوزی، دستور زیر را در کنسول تایپ می‌کنم:

```
node NodeTest.js
```

برای اطمینان از عملکرد درست، پورت 80 را روی ماشینی که Node.js در حال اجراست نوبری کنید. باید چیزی همانند شکل 2-1 ببینید.



شکل 1-2: تست Node.js

Node.js را روی ماشینی متفاوت با وب سرور اجرا کردیم و این بدین معنی است که در استفاده از پورت 80 مشکلی وجود ندارد. چنانچه تنها یک ماشین در اختیار دارید که وب سرور روی پورت 80 اجرا می‌شود، اسکریپت Node.js را روی پورت دیگری تغییر دهید. این قسمت از کد را در شکل لیست 1-3 پررنگ‌تر نشان داده‌ام.

اسناد تصویر

در سراسر کتاب از تصاویری استفاده کرده‌ام که در همین جا از همه کسانی که اجازه‌ی استفاده از آن‌ها را داده‌اند، تشکر می‌کنم.

چکیده

در این فصل محتوا، ساختار این کتاب و تنظیمات نرم‌افزاری که برای توسعه jQuery لازم است، به همراه هر آن چیزی که به طور رایگان می‌توانید به دست آورید توضیح داده شده است. سه فصل آینده، مهارت‌های اصولی CSS، HTML و جاوااسکریپت را یادآوری می‌کنیم. اگر نسبت به این موضوعات آشنایی دارید از فصل 5 آغاز کنید (جایی که jQuery معرفی می‌شود).

فصل 2

مبانی HTML

در این کتاب زمان زیادی را صرف کار با اسناد HTML می‌کنیم. در این فصل مجموعه‌ای از اطلاعاتی که درباره‌ی کارهای آینده لازم است بدانیم گردآوری شده است. این آموزش HTML نیست اما چکیده‌ای از نکات کلیدی HTML است که در فصول بعدی بر آن تکیه می‌کنیم.

آخرین نگارش HTML، HTML5 است که در جای خود موضوع کاملی است. HTML5 بیشتر از 100 عنصر دارد و هر یک از آنها اهداف و قابلیت‌های خاص خود را دارند. بنابر آنچه گفته شد شما نیاز به دانش اولیه درباره‌ی HTML دارید تا بفهمید jQuery چگونه کار می‌کند. چنانچه بخواهید جزئیات بیشتری از HTML یاد بگیرید، پیشنهاد می‌کنم کتاب دیگر مرا با عنوان *The Defenitive Guide to HTML5* مطالعه نمایید (انتشارات Apress).

معرفی یک سند HTML پایه

بهترین مکان برای آغاز، نگاه کردن به یک سند HTML است. می‌توانید ساختار پایه و سلسله مراتبی که همه اسناد HTML دارند را ببینید. لیست 1-2 یک سند HTML ساده را نشان می‌دهد. در این کتاب از این سند برای معرفی مفاهیم اصلی HTML استفاده خواهیم کرد.

لیست 1-2: یک سند HTML ساده

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
  <script src="jquery-2.0.2.js" type="text/javascript"></script>
  <style>
    h1 {
      width: 700px; border: thick double black; margin-left: auto;
      margin-right: auto; text-align: center; font-size: x-large; padding: .5em;
      color: darkgreen; background-image: url("border.png");
      background-size: contain; margin-top: 0;
    }
    .dtable {display: table;}
    .drow {display: table-row;}
    .dcell {display: table-cell; padding: 10px;}
    .dcell > * {vertical-align: middle}
    input {width: 2em; text-align: right; border: thin solid black; padding: 2px;}
    label {width: 5em; padding-left: .5em; display: inline-block;}
    #buttonDiv {text-align: center;}
    #oblock {display: block; margin-left: auto; margin-right: auto; width: 700px;}
```

```

</style>
</head>
<body>
  <h1>Jacqui's Flower Shop</h1>
  <form method="post">
    <div id="oblock">
      <div class="dtable">
        <div class="drow">
          <div class="dcell">
            <label for="astor">Astor:</label>
            <input name="astor" value="0" required>
          </div>
          <div class="dcell">
            <label
for="daffodil">Daffodil:</label>
            <input name="daffodil" value="0" required >
          </div>
          <div class="dcell">
            <label for="rose">Rose:</label>
            <input name="rose" value="0" required>
          </div>
        </div>
        <div class="drow">
          <div class="dcell">
            <label for="peony">Peony:</label>
            <input name="peony" value="0" required>
          </div>
          <div class="dcell">
            <label
for="primula">Primula:</label>
            <input name="primula" value="0" required>
          </div>
          <div class="dcell">
            <label
for="snowdrop">Snowdrop:</label>
            <input name="snowdrop" value="0" required>
          </div>
        </div>
      </div>
      <div id="buttonDiv"><button type="submit">Place Order</button></div>
    </form>
  </body>
</html>

```

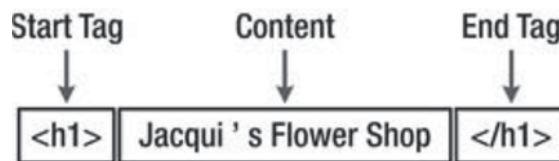
این یک سند HTML پایه و البته کوتاهی است؛ اما شامل موارد بسیار مهمی در سند HTML است. نتیجه‌ی کد بالا سندی است که می‌توانید ظاهر آن را در یک مرورگر مانند شکل 1-2 ببینید.



شکل 1-2: نمایش یک سند HTML ساده در مرورگر

فهم چهارچوب یک عنصر HTML

قلب یک HTML مفهومی است به نام عنصر. عنصر به مرورگر می‌گوید که در هر بخش از محتوای سند HTML چه چیزی قرار دارد. در اینجا یک مثال همه چیز را توضیح می‌دهد. این عنصر شامل 3 بخش است. تگ آغاز، تگ پایان و محتوی مانند شکل 2-2.



شکل 2-2: قالب یک عنصر HTML ساده

نام این عنصر (که به نام تگ هم معروف است) h1 است و به مرورگر می‌گوید که محتوای مابین تگ‌ها باید به صورت سرلیست نشان داده شود. تگ آغاز همیشه در یک جفت <> است و تگ پایانی به شکل </> می‌باشد.

فهم صفات

می‌توانید با افزودن صفات، اطلاعات بیشتری درباره‌ی عناصر ارائه دهید. لیست 2-2 یک عنصر دارای یک صفت را نشان می‌دهد.

لیست 2-2: تعریف یک صفت

```
<label for="astor">Astor:</label>
```

عنصر بالا یک عنصر برچسب است و یک صفت به نام for دارد. عمدا صفت را پررنگ کردم که دیدنش آسان‌تر باشد. همیشه صفات در قسمت تگ هستند. صفت، یک نام و یک مقدار دارد. نام صفت بالا for و مقدارش astor است. لازم نیست همه صفات مقدار داشته باشند. تعریف عناصر در واقع یک سیگنالی است به مرورگر به مفهوم اینکه یک رفتار ویژه از این عنصر مورد انتظار است. لیست 2-3 یک مثال دیگر از یک عنصر با صفت را نشان می‌دهد.

لیست 2-3: تعریف یک صفت که مقدار ندارد

```
<input name="snowdrop" value="0" required>
```

این عنصر 3 صفت دارد دو صفت ابتدایی نام و مقدار دارند (این مثال شاید کمی گیج کننده باشد زیرا نام‌های این صفات name و value هستند. مقدار صفت name، snowdrop و مقدار صفت value صفر می‌باشد) صفت سوم واژه required است که مقدار ندارد. اگرچه می‌توانید همین صفت را با مقدار یا با جای خالی مقداردهی کنید (required="required") و (required="").

صفات class و id

دو صفت بسیار مهم در این کتاب به نام‌های id و class وجود دارند. یکی از رایج‌ترین کارهایی که با jQuery انجام می‌دهید جست‌وجوی یک یا چندین عنصر در سند است؛ بطوری‌که پس از جست‌وجو می‌توانید انواع عملیات متنوع روی آنها انجام دهید. صفات id و class برای انجام جست‌وجوی عناصر بسیار مفید هستند.

استفاده از صفت id

از صفت id برای شناسایی هر یک از عناصر درون سند استفاده می‌کنید. هیچ دو عنصری نباید مقدار id یکسانی داشته باشد. لیست 2-4، سند بسیار ساده‌ای که از صفت id استفاده می‌کند را نشان می‌دهد.

لیست 2-4: استفاده از صفت id

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
</head>
<body>
<h1 id="mainheader">Welcome to Jacqui's Flower Shop</h1>
<h2 id="openinghours">We are open 10am-6pm, 7 days a week</h2>
<h3 id="holidays">(closed on national holidays)</h3>
</body>
</html>
```

صفت id را برای 3 عنصر درون سند تعریف کردیم. عنصر h1 یک id با مقدار mainheader، عنصر h2 با مقدار openinghours و عنصر h3 با مقدار holidays از هم متمایز شده‌اند. استفاده از مقدار id به شما اجازه‌ی جست‌وجوی یک عنصر مشخص در سند را می‌دهد.

استفاده از صفت class

مهم‌ترین استفاده از صفت class شبیه ساختن عناصر دلخواه است. بسیاری از عناصر می‌توانند به کلاس مشابه منسوب شوند و یک عنصر هم می‌تواند به بیش از یک کلاس تعلق داشته باشد همان‌گونه که در لیست 2-5 نشان داده شده است.

لیست 2-5: استفاده از صفت class

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Example</title>
</head>
<body>
<h1 id="mainheader" class="header">Welcome to Jacqui's Flower Shop</h1>
<h2 class="header info">We are open 10am-6pm, 7 days a week</h2>
<h3 class="info">(closed on national holidays)</h3>
</body>
</html>
```

در این مثال عنصر h1 به کلاس header، عنصر h2 به کلاس‌های header و info و عنصر h3 به کلاس info تعلق دارد. همان‌گونه که می‌بینید می‌توانید تنها با یک فضای خالی بین نام کلاس‌ها، یک عنصر را به چندین کلاس بیفزایید.

فهم محتوای عنصر

عناصر می‌توانند متن داشته باشند و نکته کلیدی اینکه می‌توانند شامل عناصر دیگر نیز باشند. در اینجا یک مثال از عنصری است که شامل چندین عنصر دیگر است.

```
<div class="dcell">

<label for="rose">Rose:</label>
<input name="rose" value="0" required>
</div>
```

عنصر div شامل 3 عنصر دیگر است (img,label,input) و شما حق دارید به صورت تودرتو عناصر را تعریف کنید. تودرتویی عناصر یک مفهوم کلیدی در HTML است و این برای عناصر بیرونی یک پیام دارد، اینکه بدانند دارای عناصر داخلی هستند. (موضوعی که بعداً به آن برمی‌گردیم) همچنین می‌توانید محتوای متنی را با عناصر دیگر ترکیب کنید. به این شکل:

```
<div class="dcell">
Here is some text content

Here is some more text!
<input name="rose" value="0" required>
</div>
```

فهم عناصر خالی (void)

لازم نیست همه عناصر محتوی داشته باشند؛ به این عناصر عناصر تهی گفته می‌شود و بدون تگ پایانی نوشته می‌شوند. عنصر تهی در یک تگ ساده نوشته می‌شود و در پایان با / بسته می‌شود.

```

```

البته یک فضای خالی مابین آخرین کارکتر و کارکتر / وجود دارد. مرورگرها نسبت به تفسیر HTML بسیار سخت‌گیر نیستند و می‌توانید این فضای خالی را حتی حذف کنید. عناصر خالی معمولاً برای مراجعه به منابع بیرونی استفاده می‌شوند. در این مثال عنصر img با فایل تصویری به نام rose.jpg ارتباط برقرار می‌کند.

فهم ساختار سند

در هر سند HTML عناصر کلیدی وجود دارند که ساختار سند را تعریف می‌کنند. مانند DOCTYPE، html، head و body. لیست 2-6 ارتباط بین عناصر - با محتوی حذف شده - را نشان می‌دهد.

لیست 2-6: ساختار پای‌های یک سند HTML

```
<!DOCTYPE html>
<html>
<head>
    ...head content...
</head>
<body>
    ...body content...
</body>
</html>
```

هر یک از این عناصر نقش خاصی درون سند HTML بازی می‌کنند. نخستین آنها عنصر DOCTYPE است که به مرورگر می‌فهماند سند پیش رو یک سند HTML است و به‌طور خاص مثلاً یک سند HTML5 است. نگارش‌های جدیدتر HTML، به اطلاعات بیشتری نیاز دارد. برای مثال تعریف زیر یک عنصر DOCTYPE برای سند HTML4 را نشان می‌دهد.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

عنصر html نماینده‌ی بخشی از سند است که محتوی HTML را دربر می‌گیرد. همیشه این عنصر شامل 2 عنصر دیگر به نام‌های head، body است. همان‌گونه که در آغاز این فصل گفته شد قرار نیست به توضیح همه جانبه عناصر HTML پرداخته شود. برخی از این موضوع‌ها و شرح حال HTML5 را به‌طور کامل در یک کتاب بیش از 1000 صفحه‌ای آورده‌ام و تنها می‌خواهم توضیح چکیده‌ای درباره‌ی عناصر داده باشم تا در این باره دید خوبی داشته باشید. جدول 2-1 چکیده‌ای از عناصر مورد استفاده در اسناد HTML را آورده که برخی از آنها را به‌صورت مختصر در همین فصل توضیح می‌دهیم.

جدول 2-1: عناصر استفاده شده در سند مثال

عنصر	شرح
DOCTYPE	نوع محتوای سند را نشان می‌دهد
body	نشان‌دهنده بخشی از سند است که محتوای عناصر را دربر می‌گیرد
button	معرف یک دکمه است که بیشتر برای ارسال اطلاعات به سمت سرور استفاده می‌شود
div	یک عنصر عمومی بیشتر برای سازماندهی سند در نمایش هدفمند استفاده می‌شود
form	معرف یک فرم از HTML است و اجازه‌ی دریافت اطلاعات از کاربر و ارسال به سمت سرور - برای پردازش - را می‌دهد

عنصر	شرح
h1	معرف یک سرلیست است
head	معرف بخشی از سند شامل متادیتاها است
html	معرف بخشی از سند شامل html است
img	معرف یک تصویر می‌باشد
input	معرف یک فیلد ورودی برای گرفتن یک دیتای ساده از طرف کاربر می‌باشد. معمولاً در بخش form قرار می‌گیرد.
script	معرف یک اسکریپت (معمولاً جاوااسکریپت) است و بایستی اجرا گردد.
style	معرف بخشی برای تنظیمات CSS (Cascading Style Sheet) است. فصل 3 را ببینید.
title	معرف عنوان سند است که مرورگر از آن برای نمایش عنوان استفاده می‌کند.

فهم عناصر متادیتا

عنصر head دربرگیرنده متادیتاهای سند است. به دیگر سخن، عناصری که روی محتوای سند شرح می‌دهند یا عملیاتی انجام می‌دهند ولی خود دارای ظاهری نمایشی در مرورگر نیستند. سند این مثال شامل 3 عنصر متادیتا در بخش head است: title, script و style. عنصر title یکی از مهم‌ترین آنها است. محتوای این عنصر توسط مرورگر جهت تنظیم عنوان پنجره یا تب درون پنجره استفاده می‌شود. همه اسناد HTML به عنصر title نیاز دارند. دو بخش دیگر برای این کتاب بسیار مهم هستند و در ادامه شرح خواهیم داد.

فهم عنصر script

عنصر script معرف کد جاوااسکریپت است. به بیان ساده، همان عنصری که وقت زیادی از شما می‌گیرد! و در طول کتاب به صورت عمقی پوشش داده می‌شود. سند مثال شامل یک عنصر اسکریپت است. لیست 2-7 را ببینید.

لیست 2-7: عنصر script از سند مثال

```
<script src="jquery-2.0.2.js" type="text/javascript"></script>
```

با تعریف صفت src برای عنصر script، در واقع به مرورگر می‌فهمانید که قصد لودکردن فایل دیگری - که شامل کدهای جاوااسکریپت است - دارید. در این مثال مقدار صفت src مهم‌ترین کتابخانه jQuery است که مرورگر فایل jquery-2.0.2.js را جست‌وجو می‌کند. یک سند HTML ساده می‌تواند چندین عنصر اسکریپت را دربرگیرد. می‌توان کدهای جاوااسکریپت - که داخل تگ script هستند - را در هر جای سند قرار داد و در این زمینه محدودیتی وجود ندارد. لیست 2-8 نمونه‌ای از استفاده از تگ script است.

لیست 2-8: استفاده از عنصر script در تعریف کد جاوااسکریپت درون خطی

```
<!DOCTYPE html>
<html>
```

```

<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $('#mainheader').css("color", "red");
    });
  </script>
</head>
<body>
  <h1 id="mainheader" class="header">Welcome to Jacqui's Flower Shop</h1>
  <h2 class="header info">We are open 10am-6pm, 7 days a week</h2>
  <h3 class="info">(closed on national holidays)</h3>
</body>
</html>

```

این مثال شامل دو اسکریپت است. اسکریپت نخست کتابخانه jQuery را به درون سند می‌آورد و دومی یک اسکریپت است که از توابع اصلی jQuery استفاده می‌کند. درباره‌ی اسکریپت دوم نگران نباشید. در فصل 5 وارد دنیای jQuery خواهیم شد. عنصر Script می‌تواند در head یا body سند قرار بگیرد. در این کتاب script را در قسمت head قرار داده‌ام و البته این تنها نظر شخصی است.

فهم عنصر style

یکی از روش‌هایی که می‌توانید خصوصیات داخل سند را تعیین نمایید عنصر style است. مختصر و مفید اینکه شما با CSS، سند نمایشی خود را در مرورگر مدیریت می‌کنید. لیست 2-9 عنصر style را نشان می‌دهد.

لیست 2-9: عنصر style

```

<style>
  h1 {
    width: 700px; border: thick double black; margin-left: auto;
    margin-right: auto; text-align: center; font-size: x-large; padding: .5em;
    color: darkgreen; background-image: url("border.png");
    background-size: contain; margin-top: 0;
  }
  .dtable {display: table;}
  .drow {display: table-row;}
  .dcell {display: table-cell; padding: 10px;}
  .dcell > * {vertical-align: middle}
  input {width: 2em; text-align: right; border: thin solid black; padding: 2px;}
  label {width: 5em; padding-left: .5em; display: inline-block;}
  #buttonDiv {text-align: center;}
  #oblock {display: block; margin-left: auto; margin-right: auto; width: 700px;}
</style>

```

مرورگر مجموعه‌ای از خصوصیات را نگهداری می‌کند که مقادیر آن در نحوه نمایش هر عنصر استفاده می‌شود. عنصر style به شما این اختیار را می‌دهد تا عنصر مورد نظرتان را در هر کجای سند یافته و خصوصیات آن را تغییر دهید. توضیحات بیشتر در فصل 3 آمده است.

عنصر style مانند script در بخش head یا body سند قرار می‌گیرد. اما در این کتاب تنها در بخش head قرار گرفته است. بازهم این نظر شخصی است و CSS می‌تواند به صورت یک فایل وارد سند شود.

فهم عناصر محتوی

عنصر body دربرگیرنده محتوای سند HTML است. عناصری وجود دارد که مرورگر آنها را نمایش می‌دهد. عناصری مثل متادیتا یا اسکریپت، نمایشی نیستند و برای توضیح یا انجام عملیاتی استفاده می‌شوند.

فهم جداسازی معنا / نمایش

یکی از موضوعات مهم در HTML5 (کمی فیلسوفانه) جداسازی بین معنی (سماتیک) عنصر و اثر عنصر بر روی نمایش محتوا است. این ایده معقولی است. از عناصر HTML برای ساختار دادن و معنی دادن به محتوی استفاده می‌شود و سپس با اعمال CSS روی عناصر، نحوه نمایش آنها کنترل می‌شود. نیازی به نمایش همه آنچه درون سند HTML تزریق می‌شود، نیست. با جداسازی نمایش، HTML برای پردازش ساده‌تر شده و به همین شکل درک مفهوم آن آسان‌تر می‌شود.

هر عنصر HTML یک مفهوم خاصی دارد. برای مثال عنصر article نماینده‌ی یک بخش بزرگی از محتوی است که برای مقررات و مقالات مناسب است. عنصر h1 هم برای مشخص کردن سرلیست بخش محتوا استفاده می‌شود.

این مفهوم در قلب HTML نهفته است و آن را تنها بر روی عناصر اعمال می‌کنید. با این اعمال، نوع محتوایی که انتظار دارید رقم می‌خورد. خوانندگان به وسیله متن، یک استخراج منطقی خوب و بهتری خواهند داشت. مثلاً فوراً می‌فهمند که مفهوم یک بخش از صفحه در سرلیست آن چکیده شده و البته به صورت مختصر آورده شده است. (زیرا این الگو در بیشتر کتاب‌ها رعایت شده است). این عناصر را روی هر بخش محتوا اعمال می‌کنید و ارتباط با عنصر دیگر را مشخص می‌سازید. لیست 2-10 نمونه سندی را نمایش می‌دهد که از عناصر برای شناخت بیشتر مفهوم نمایش و معنی استفاده می‌کند.

لیست 2-10: استفاده از عناصر برای افزودن ساختار و مفهوم محتوی

```
<!DOCTYPE html>
<html>
<head>
  <title>Example</title>
</head>
<body>
  <article>
    <header>
      <hgroup>
        <h1>New Delivery Service</h1>
        <h2>Color and Beauty to Your Door</h2>
      </hgroup>
    </header>
    <section>
      We are pleased to announce that we are starting a home delivery
      service for
      your flower needs. We will deliver within a 20 mile radius of the store
```