

۷ کارنیل، بزرگترین شبکه موفقیت ایرانیان می باشد، که افرادی زیادی توانسته اند با آن به موفقیت برسند، فاطمه رتبه ۱۱ کنکور کارشناسی، محمد حسین رتبه ۶۸ کنکور کارشناسی، سپیده رتبه ۳ کنکور ارشد، مریم و همسرش راه اندازی تولیدی مانتو، امیر راه اندازی فروشگاه اینترنتی، کیوان پیوستن به تیم تراکتور سازی تبریز، میلاد پیوستن به تیم صبا، مهسا تحصیل در ایتالیا، و.... این موارد گوشه از افرادی بودند که با کارنیل به موفقیت رسیده اند، شما هم می توانید موفقیت خود را با کارنیل شروع کنید.

برای پیوستن به تیم کارنیلی های موفق روی لینک زیر کلیک کنید.

www.karnil.com

همچنین برای ورود به کانال تلگرام کارنیل روی لینک زیر کلیک کنید.

<https://telegram.me/karnil>



مؤسسه فرهنگی هنری
دیباگران تهران



مرجع آموزشی jQuery

write less, do more.



ترجمه و تألیف: احمد بادپی

به نام خدا



مؤسسه فرهنگی هنری
دیباگران تهران

مرجع آموزشی

jQuery

ترجمه و تألیف
احمد بادپی

مؤلفان
Karl Swedberg
Jonathan Chaffer

شناسنامه

فهرست مطالب

۷.....	مقدمه ناشر
۸.....	مقدمه
۱۰.....	پیشگفتار

فصل اول: تشریح پیکره اصلی اسکریپت‌های jQuery

۱۴.....	بررسی اجمالی کتابخانه‌های جاوااسکریپت
۱۶.....	مزیت‌های jQuery
۲۰.....	شروع به کار با jQuery
۲۶.....	مثالی عملی: فهرستی پویا از محتوای صفحه

فصل دوم: گزینشگرها

۳۹.....	تمام صفحات HTML یک درخت هستند!!!
۴۱.....	گزینشگرهای CSS
۵۱.....	گزینشگرهای صفت
۵۶.....	گزینشگرهای form
۵۸.....	گزینشگرهای سفارشی
۶۵.....	کارگاه

فصل سوم: متدهای پیمایش DOM

۶۷.....	تابع jQuery()
۷۳.....	نگاشت چیست؟
۷۳.....	متدهای پالایش (فیلتر کردن)
۸۳.....	متدهای پیمایش درخت عناصر
۱۰۳.....	متدهای گوناگون پیمایش DOM
۱۱۳.....	کارگاه

فصل چهارم: متدهای تغییر و دستکاری در DOM

۱۱۷.....	صفات عمومی.....
۱۲۱.....	دستکاری خواص عناصر.....
۱۲۶.....	دستکاری قواعد سبک.....
۱۳۹.....	صفت‌های کلاس.....
۱۴۵.....	جایگزینی عناصر صفحه.....
۱۵۶.....	متدهای درج درونی.....
۱۶۳.....	درج خارجی.....
۱۶۹.....	درج در اطراف عناصر.....
۱۷۷.....	حذف از DOM.....
۱۸۰.....	کارگاه.....

فصل پنجم: متدهای کار با رویدادها

۱۹۱.....	ثبت کنترلگرهای رویداد.....
۲۰۸.....	رویدادهای مربوط به بارگذاری صفحه.....
۲۱۳.....	رویدادهای ماوس.....
۲۳۳.....	رویدادهای form.....
۲۴۱.....	رویدادهای صفحه‌کلید.....
۲۴۸.....	رویدادهای مرورگر.....
۲۵۱.....	کارگاه.....

فصل ششم: کار با افکت (انیمیشن)ها

۲۶۳.....	افکت‌های پیش ساخته.....
۲۸۲.....	افکت (انیمیشن)های سفارشی.....
۲۹۵.....	کارگاه.....

فصل هفتم: متدهای کار با AJAX

۳۱۲.....	متدهای سطح پایین.....
۳۲۵.....	متدهای میانه برای کار با AJAX.....
۳۳۱.....	JSON چیست؟.....
۳۳۵.....	کنترلگرهای رویداد عمومی AJAX.....
۳۴۴.....	توابع کمکی در AJAX.....
۳۴۸.....	کارگاه.....

فصل هشتم: متدهای گوناگون در jQuery

۳۶۵.....	متدهای راه اندازی.....
۳۶۸.....	متدهای مربوط به عناصر DOM.....
۳۷۳.....	متدهای دستکاری مجموعه‌ها.....
۳۸۷.....	تشخیص انواع.....
۳۸۹.....	ذخیره‌سازی داده‌ها.....
۳۹۲.....	کارگاه.....

فصل نهم: خاصیت‌های jQuery

۳۹۷.....	خواص سراسری.....
۴۰۱.....	خواص شیء jQuery.....

فصل دهم: پلاگین‌ها

۴۰۳.....	استفاده از پلاگین‌ها.....
۴۰۵.....	ایجاد یک پلاگین.....

فصل یازدهم: آشنایی با پلاگین‌های مشهور jQuery

۴۱۵.....	آماده‌سازی پلاگین‌ها.....
۴۱۶.....	استفاده از پلاگین snowfall برای بارش برف در صفحه.....
۴۱۹.....	افزودن کادرهای راهنمای دلخواه با پلاگین Hovertip.....
۴۲۱.....	اعتبار سنجی فرم‌ها با استفاده از پلاگین validate.....
۴۲۵.....	ایجاد فیلدهای فرمت‌بندی شده با پلاگین Masked Input.....
۴۲۷.....	نمایش سرخط آخرین اخبار با پلاگین News Ticker.....
۴۲۹.....	ارسال فرم به روش jQuery Form Plugin با AJAX.....
۴۳۵.....	نمایش اسلایدی تصاویر و محتوا با EasySlider.....
۴۳۹.....	پلاگین Cycle.....

ضمیمه الف: منابع آنلاین

۴۴۳.....	مستندات jQuery.....
۴۴۴.....	مراجع جاوااسکریپت.....
۴۴۶.....	فشرده‌سازهای فایل‌های جاوااسکریپت.....
۴۴۷.....	برگردان‌های جاوااسکریپت.....
۴۴۸.....	مراجع HTML(X).....
۴۴۸.....	صفحه خانگی برگه‌های سبک آبخاری در سایت W3C.....
۴۴۹.....	وبلاگ‌های مفید.....
۴۵۲.....	چارچوب‌های توسعه وب مبتنی بر jQuery.....

ضمیمه ب: ابزارهای طراحی و توسعه

۴۵۳.....	ابزار Firefox.....
۴۵۵.....	ابزارهای Internet Explorer.....
۴۵۶.....	ابزار safari.....
۴۵۷.....	ابزارهای Opera.....
۴۵۷.....	سایر ابزارها.....
۴۶۰.....	منابع و مأخذ.....

مقدمه ناشر

خط مشی کیفیت انتشارات مؤسسه فرهنگی هنری دیباگران تهران در عرضه کتاب های است که تواند

خواسته های به روز جامعه فرهنگی و علمی کشور را تا حد امکان پوشش دهد

حمد و سپاس ایزد منان را که با الطاف بیکران خود این توفیق را به ما ارزانی داشت تا بتوانیم در راه ارتقای دانش عمومی و فرهنگ این مرز و بوم در زمینه چاپ و نشر کتب علمی دانشگاهی، علوم پایه و به ویژه علوم کامپیوتر و انفورماتیک گام هایی هر چند کوچک برداشته و در انجام رسالتی که بر عهده داریم، مؤثر واقع شویم. گستردگی علوم و توسعه روزافزون آن، شرایطی را به وجود آورده که هر روز شاهد تحولات اساسی چشمگیری در سطح جهان هستیم. این گسترش و توسعه نیاز به منابع مختلف از جمله کتاب را به عنوان قدیمی ترین و راحت ترین راه دستیابی به اطلاعات و اطلاع رسانی، بیش از پیش روشن می نماید. در این راستا، واحد انتشارات مؤسسه فرهنگی هنری دیباگران تهران با همکاری جمعی از اساتید، مؤلفان، مترجمان، متخصصان، پژوهشگران، محققان و نیز پرسنل ورزیده و ماهر در زمینه امور نشر در صدد هستند تا با تلاش های مستمر خود برای رفع کمبودها و نیازهای موجود، منابعی پربار، معتبر و با کیفیت مناسب در اختیار علاقه مندان قرار دهند.

کتابی که در دست دارید با همت "جناب آقای احمد بادپی" و تلاش جمعی از همکاران انتشارات میسر گشته که شایسته است از یکایک این گرامیان تشکر و قدردانی کنیم.

ویراستاری: شیوا غمگسار

ویرایش و صفحه آرایی کامپیوتری: آرزو مهدوی

طراح جلد: مینا دیده بان

ناظر چاپ:

در خاتمه ضمن سپاسگزاری از شما دانش پژوه گرامی درخواست می نماید با مراجعه به آدرس dibagaran.mft.info (ارتباط با مشتری) فرم نظرسنجی را برای کتابی که در دست دارید تکمیل و ارسال نموده، انتشارات دیباگران تهران را که جلب رضایت و وفاداری مشتریان را هدف خود می داند، یاری فرمایید.

امیدواریم همواره بهتر از گذشته خدمات و محصولات خود را تقدیم حضورتان نماییم.

مدیر انتشارات

مؤسسه فرهنگی هنری دیباگران تهران

publishing@mftmail.com

ای بیج خطی نکته‌ز اول بی حجت نام تو سبیل

jQuery به عنوان یک پدیده نوپا، با فراهم آوردن ابزار و ویژگی‌های منحصر به فرد و در عین حال ساده، عرصه تکنولوژی‌های طراحی وب را متحول کرده است. این تکنولوژی که به عنوان یکی از چارچوب‌های جاوااسکریپت در عمر پنج ساله خود و با شعار اختصاصی "نوشتن کمتر، انجام کارهای بیشتر" قدم در این میدان نهاده، توانسته است با دو ویژگی اساسی خود یعنی امکان دستکاری آسان عناصر موجود در صفحه و سهولت انجام عملیات از طریق Ajax جای خود را در بسیاری از پروژه‌های توسعه‌دهندگان وب به منظور افزایش جذابیت و پویایی هر چه بیشتر صفحات باز کند.

اهمیت و کاربرد روز افزون jQuery از یک سو و فقدان منابع و مراجع فارسی زبان در این زمینه از سوی دیگر، این حقیر را بر آن داشت تا با تکیه بر دانش ناچیز شخصی اقدام به ترجمه یکی از بهترین مراجع حال حاضر jQuery برای تمامی علاقه‌مندان کنم.

منبع اصلی کتاب پیش رو یکی از بهترین و محبوب‌ترین کتاب‌های منتشر شده در زمینه jQuery یعنی jQuery 1.4 Reference Guide بوده است که البته به منظور تفهیم هر چه بهتر مطالب و روشن شدن مقصود، از منابع دیگری که جزئیات آن‌ها در فهرست منابع و مأخذ آمده و نیز تجربیات شخصی کمک گرفته شده است.

این کتاب نخستین کتابی است که به صورت تخصصی در زمینه jQuery در ایران منتشر شده است و به بررسی دقیق تمامی جزئیات این کتابخانه می‌پردازد. در زمان گردآوری این اثر آخرین نسخه ارائه شده از jQuery نسخه 1.6.2 بوده که تمامی ویژگی‌های این نسخه نیز در کتاب گنجانده شده است. برای اینکه کتاب جنبه کاربردی‌تر پیدا کند در انتهای هر فصل بخش‌هایی با عنوان کارگاه که به بررسی مثال‌های کاربردی مرتبط با موضوع همان فصل می‌پردازد در نظر گرفته شده است.

برای مطالعه این کتاب باید با زبان‌هایی همچون HTML، CSS و جاوااسکریپت که جزء مبانی تکنولوژی‌های طراحی و توسعه وب محسوب می‌شوند، آشنایی داشته باشید. در این کتاب که به صورت تخصصی به بررسی jQuery می‌پردازد، تنها به یادآوری آن دسته از موضوعاتی از این زبان‌ها که در jQuery بسیار کاربرد و اهمیت داشته است پرداخته‌ایم. آشنایی با مبانی ابتدایی زبان جاوااسکریپت (همچون انواع داده‌ها و متغیرها، رشته‌ها، آرایه‌ها و DOM) که jQuery زاده آن به شمار می‌رود به شما در یادگیری سریع‌تر آن کمک خواهد کرد.

البته لازم به ذکر است که ترجمه، تخصص اینجانب نبوده و هدف از تهیه کتابی که هم اکنون در دستانتان شماس است هم چیزی جز جامه عمل پوشاندن به حدیث شریف «زکاة العلم نشره» نبوده است. از این رو این کتاب نیز خالی از اشکال و اشتباه نیست. لذا از تمامی خوانندگان تقاضا می‌شود به منظور گزارش

اشکال و بیان نظرات، انتقادات و پیشنهادات خود با ایمیل اینجانب به آدرس **ahmadbadpey@gmail.com** مکاتبه فرمایند.

CD همراه این کتاب حاوی تمامی کدها و تصاویر تمامی فصول به صورت مجزا، مجموعه‌ای دسته‌بندی شده از پلاگین‌های مشهور در موضوعات مختلف، بهترین کتاب‌های الکترونیکی ارائه شده در زمینه زبان‌هایی همچون جاوااسکریپت، CSS و jQuery، نسخه آفلاین سایت w3schools.com (که به صورت تخصصی به آموزش تکنولوژی‌های طراحی و توسعه وب می‌پردازد) و برخی برنامه‌های معرفی شده در کتاب می‌باشد.

لازم به ذکر است که همزمان با تهیه این کتاب، تصمیم به طراحی نخستین مرجع آموزشی jQuery به آدرس **www.learningjquery.ir** گرفتیم که می‌توانید برای بهره‌گیری از مطالب آموزشی و سایر مطالب مرتبط با jQuery، بیان سؤالات و مشکلات خود در کار با jQuery و تماس مستقیم با بنده به آن مراجعه نمایید.

کتاب پیش رو حاصل برگزاری دوره‌های آموزشی طراحی وب می‌باشد که به همت انجمن علمی مهندسی فناوری اطلاعات دانشگاه پیام نور مرکز آران و بیدگل تدارک دیده شد و به ثمر رسیدن آن را بدون یاری جمع زیادی از دوستان که جا دارد همین جا از آن‌ها تشکر کنم بعید می‌دانم. در درجه اول از پدر و مادر عزیزم که همواره در تمامی مراحل زندگی یاریگر و پشتیبان بنده بوده‌اند تشکر می‌کنم. از استاد عزیزم جناب آقای مهندس مصطفی قبایی نیز به خاطر راهنمایی‌ها و دلگرمی‌هایشان، دوستان خوبم صادق رحمانی به خاطر همراهی و همیاری بنده در تمامی مراحل مذاکره با ناشر و پیگیری مراحل چاپ و نشر کتاب، حسین بیدی به خاطر پشتیبانی‌های بی‌منتش از بنده به عنوان مدرس دوره‌های آموزشی طراحی وب، مجید بصیرتی به خاطر ویرایش و صفحه‌آرایی ابتدایی کتاب، خانم فریبا معظمی به خاطر اصلاح و خطایابی ابتدایی کتاب، تمامی اعضای محترم انجمن علمی مهندسی فناوری اطلاعات دانشگاه پیام نور مرکز آران و بیدگل و تمامی دوستانی که با اعتماد به بنده و حضور گرم خود در دوره‌های آموزشی برگزار شده موجبات دلگرمی و انگیزه برای تهیه این کتاب را فراهم آوردند صادقانه و صمیمانه تشکر می‌کنم.

احمد بادپی

Ahmadbadpey@gmail.com

مولای متقیان، علی (علیه السلام) می‌فرمایند:

إِذَا أَرَدَ اللَّهُ عَبْدًا حَظَرَ عَلَيْهِ الْعِلْمَ

هرگاه خداوند نخواهد بنده‌ای را به رذالت برساند، او را از علم و دانش و یادگرفتن محروم می‌سازد.

پیشگفتار

jQuery چیست؟

jQuery یکی از کتابخانه^۱های رایج زبان برنامه‌نویسی جاوااسکریپت است که ساخت برنامه‌ها و صفحات وب مبتنی بر این زبان را بسیار ساده و سریع‌تر می‌کند. اغلب می‌توان یک برنامه 10 تا 20 خطی جاوااسکریپت را با یک خط کد jQuery پیاده‌سازی کرد!
در ادامه نگاهی اجمالی به مهم‌ترین ویژگی‌های jQuery خواهیم داشت:

jQuery و مرورگرها

jQuery با ارائه یکسری توابع ساده به تمام دغدغه‌های یک برنامه‌نویس در مورد پشتیبانی از کدها در مرورگرهای مختلف پایان می‌دهد. تمامی قابلیت‌ها اعم از توابع و متدهای پیشنهادی jQuery با تمامی مرورگرها (حتی Internet Explorer 6) سازگاری دارند.

پشتیبانی از Ajax

jQuery از Ajax به خوبی پشتیبانی کرده و اجازه می‌دهد که برنامه‌های کاربردی اینترنت به برنامه‌های تحت ویندوز نزدیک شوند.
jQuery همچنین امکان ارسال و دریافت هر چه ساده‌تر داده‌ها به/از سرورها از طریق JSON را نیز فراهم می‌آورد.

پشتیبانی از انیمیشن‌ها

jQuery از انیمیشن‌ها و جلوه‌های بصری به خوبی پشتیبانی می‌کند.
این کتابخانه مجموعه‌ای عظیم از انواع انیمیشن‌ها و جلوه‌های بصری است که امکان تحت تأثیر قرار دادن کاربرانتان را می‌دهد.
jQuery همچنین از امکان کشیدن و رهاکردن^۲ عناصر صفحه به خوبی پشتیبانی می‌کند.

انتخاب و دستکاری عناصر صفحه

jQuery به صورت ویژه امکان انتخاب عناصر صفحه به منظور کار روی آن‌ها را می‌دهد.

1- Library
2- Drag and Drop

دسترسی به عناصر یک صفحه وب مانند <p> و <h1> در جاوااسکریپت کمی دشوار و نیاز به کدنویسی زیادی دارد. اما jQuery اجازه می‌دهد که از زبان CSS برای دسترسی به آن‌ها به آسانی استفاده کنید.

برای دسترسی به عناصر در jQuery از توابع ویژه‌ای استفاده می‌شود که یکی از آن‌ها تابع jQuery() است که البته حالت کوتاه شده آن، (\$) می‌باشد. این دو تابع برای انتخاب عناصر صفحه از گزینشگرهای CSS استفاده می‌کنند.

همچنین jQuery امکان حذف و اضافه عناصر به صفحه و تغییر محتوای صفحه را با استفاده از مجموعه‌ای از متدها فراهم می‌آورد.

پلاگین‌ها

یکی از ویژگی‌های بارز jQuery امکان ایجاد تکه برنامه‌هایی موسوم به پلاگین است که با کمک آن‌ها می‌توان امکانات جدیدی را به jQuery اضافه کرده و برای سایر برنامه و پروژه‌هایمان مجدداً از آن‌ها استفاده کنیم. تعداد پلاگین‌ها بسیار زیاد است و هر روز نیز به آن‌ها اضافه می‌گردد. پلاگین‌ها توسط افراد مختلف نوشته شده و در اختیار سایرین قرار می‌گیرند.

فصل

اول

تشریح پیکره اصلی اسکرپت‌های jQuery

یک نمونه برنامه به زبان jQuery معمولاً از مجموعه‌ای وسیع و طبقه‌بندی شده از متدها که این کتابخانه فراهم آورده است استفاده می‌کند. گزینشگرها، دستکاری عناصر موجود در DOM، کنترل رویدادها و... از جمله امکاناتی هستند که jQuery به عنوان ابزار اصلی برای انجام کارها در اختیار شما قرار می‌دهد. به منظور استفاده از این کتابخانه به بهترین شکل، بهتر است وسعت قابلیت‌هایی را که فراهم آورده است، به خاطر بسپارید.

این کتاب تک تک متدها و توابعی را که در jQuery وجود دارد تشریح خواهد کرد. از آنجایی که مجموعه وسیعی از آن‌ها وجود دارد آگاهی از دسته‌بندی اساسی این متدها و نقشی که در برنامه‌های jQuery ایفا می‌کنند مفید خواهد بود. در این کتاب اسکرپت‌های کاربردی کاملی را بررسی خواهیم کرد و تفاوت مفاهیمی از jQuery را که در هر بخش از آن‌ها استفاده شده است مورد ارزیابی قرار خواهیم داد.

jQuery یکی از کتابخانه‌های جاوااسکریپت است که برای کمک به طراحان و توسعه دهندگان وب به منظور نوشتن و گسترش سریع و هر چه مختصرتر برنامه‌های جاوااسکریپت، با مجموعه‌ای از متدهای تعریف شده برپایه توابع خود جاوااسکریپت ایجاد شده است. jQuery قابلیت جدیدی ارائه نمی‌کند اما با آسان‌سازی و استفاده از syntax بسیار ساده خود، آن دسته از مفاهیم جاوااسکریپت را که درک و نوشتن آن‌ها برای عموم دشوار است، برای طیف وسیعی از علاقه‌مندان قابل درک و فهم کرده است.

بررسی اجمالی کتابخانه‌های جاوااسکریپت

کتابخانه‌های جاوااسکریپت با فراهم آوردن چارچوبی از اعمال متداولی که با جاوااسکریپت و البته براساس توابع اولیه خود این زبان پیاده سازی شده‌اند به طراحان و توسعه‌دهندگان وب امکان ارتقاء ویژگی‌های **کاربردپذیری**^۱ و **تعامل‌پذیری**^۲ صفحات وب را می‌دهند.

کتابخانه‌های جاوااسکریپت، نوشتن اسکریپت‌هایی به این زبان را برای طراحان و توسعه‌دهندگان وب بسیار ساده و دلپذیرتر کرده است. در واقع آن‌ها را می‌توان نقطه شروعی بر جاوااسکریپت دانست. بسیاری از کتابخانه‌های مشهور جاوااسکریپت همچون YUI، Dojo، MooTools، Prototype و آنچه که در این کتاب توجه اصلی ما بر آن خواهد بود یعنی jQuery، امروزه به شکل گسترده‌ای در وب مورد استفاده قرار می‌گیرند. هر یک از این کتابخانه‌ها ویژگی‌های خاص خود را دارند.

DOM^۳ یا همان مدل شی‌گرای سند، نمایش درختی عناصر موجود در یک صفحه HTML است که در آن هر یک از شاخه‌ها با استفاده از گره‌ها به صورت سلسله‌مراتبی به همدیگر متصل شده‌اند. عموماً می‌توان به این گره‌ها که هر کدام نماینده یکی از عناصر موجود در صفحه هستند از طریق گزینشگرهای CSS و جاوااسکریپت دسترسی داشت. در واقع DOM به عنوان یکی از **API**^۴های زبان جاوااسکریپت چگونگی دستکاری صفحات وب از طریق روش‌هایی منطبق با استانداردهای تعریف شده HTML را فراهم می‌کند.

چارچوب‌های جاوااسکریپت به طراحان و توسعه‌دهندگان وب امکان تغییر و دستکاری در DOM را با اضافه کردن یک فایل جاوااسکریپت (library.js) به صفحه و سپس به‌کارگیری توابع تعریفی موجود در آن می‌دهند.

کتابخانه‌های جاوااسکریپت روش‌های متعددی برای کار با افکت‌ها، انیمیشن‌ها، رویدادها، Ajax و همچنین ایجاد انواع ابزارها برای برنامه‌نویسی سریع‌تر را فراهم می‌آورند. شما نیز به عنوان یک طراح و توسعه‌دهنده وب هیچ محدودیتی در استفاده از توابع فراهم شده توسط یک کتابخانه نداشته و حتی می‌توانید خودتان نیز توابعی را نوشته و به آن‌ها اضافه کنید.

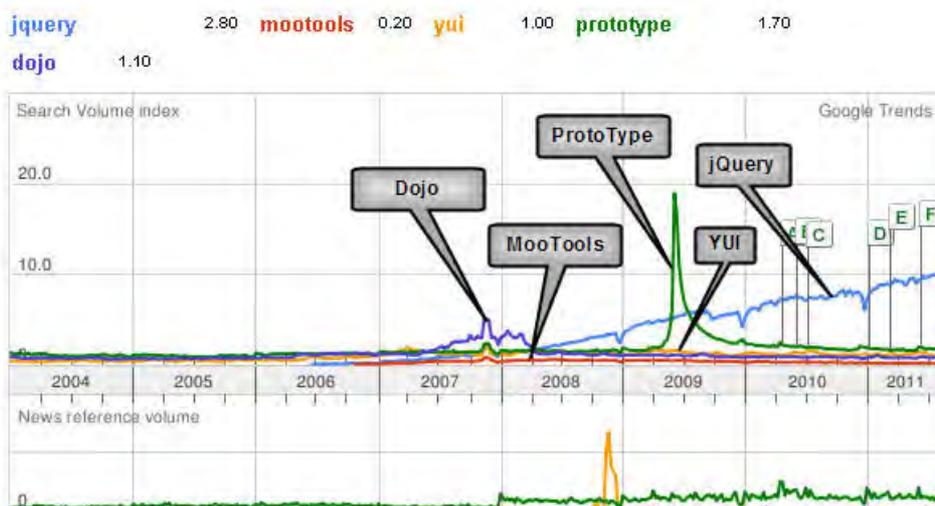
-
- 1- Usability
 - 2- Interactivity
 - 3- Document Object Model
 - 4- Application Programming Interface

زیبایی استفاده از کتابخانه‌های جاوااسکریپت برای توسعه‌دهندگان وبی که مفهوم DOM را به خوبی درک کرده‌اند این است که بدون شک تغییر و دستکاری در DOM از طریق کتابخانه‌ها نسبت به روش‌های محدود خود جاوااسکریپت اساساً بسیار ساده‌تر است. این در حالی است که چنانچه بخواهید همین کارها را از طریق خود جاوااسکریپت انجام دهید احتمالاً ناچار خواهید بود وقت زیادی را صرف برنامه‌نویسی، آزمایش و اصلاح خطاها کنید که حاصل آن حجم انبوهی از کدها خواهد بود. حال آنکه کتابخانه‌های جاوااسکریپت می‌توانند با کاهش حجم کدها، همان نتایجی را به شما بدهند که نوشتن آن‌ها توسط خود جاوااسکریپت ممکن است نیاز به کدنویسی چهار برابری داشته باشد!

جلوگیری از تکرار نیز یکی دیگر از مزیت‌های استفاده از کتابخانه‌هاست. زمانی که شروع به نوشتن توابعی برای انجام کارهای یکسان می‌کنید، در پایان با حجم زیادی از کدهای مشابه مواجه می‌شوید. با استفاده از یک کتابخانه از تکرار بی‌رویه کدها جلوگیری کنید.

در حال حاضر حدود 20 کتابخانه جاوااسکریپت وجود دارد که می‌توانید از میان آن‌ها چارچوب مورد نظر خود را انتخاب کنید. از میان همه این‌ها، 5 کتابخانه به نام‌های YUI، Prototype، MooTools، Dojo و jQuery به عنوان کتابخانه‌های اصلی به شمار رفته و در بین کاربران محبوبیت بیشتری دارند. تفاوت اصلی بین اکثر این کتابخانه‌ها در حجم و پشتیبانی مرورگرها از آن‌هاست.

تصویر زیر نمودار مقایسه‌ای کتابخانه‌های فوق از نظر میزان جستجوی نام آن‌ها توسط کاربران موتور جستجوی گوگل در سال‌های اخیر، در سرویس Google Trends را نمایش می‌دهد:



برای اطلاعات بیشتر می‌توانید به آدرس زیر مراجعه نمایید:

www.google.com/trends?q=jQuery+mootools+yui+prototype+dojo

تمامی این 5 کتابخانه منبع باز¹ هستند. بدین معنی که هر کسی می‌تواند در پیاده‌سازی کدهای منبع سازنده آن‌ها مشارکت کند. در حالی که به عنوان مثال برنامه‌های Microsoft منبع باز نبوده و به صورت اختصاصی متعلق به شرکت Microsoft هستند. این شرکت از برنامه‌نویسان خود برای توسعه برنامه‌هایش استفاده می‌کند.

برنامه‌های منبع باز کمی متفاوت بوده به طوری که هر کس می‌تواند آن‌ها را دانلود کرده و حتی در تغییر و دستکاری آن‌ها مشارکت داشته باشد و این باعث بهتر شدن کدها می‌شود زیرا بسیاری از افراد به صورت داوطلبانه و تنها با هدف نوشتن برنامه‌های بهتر و نه کسب درآمد بیشتر، در آن‌ها مشارکت دارند.

مزیت‌های jQuery

jQuery اولین بار در سال 2006 توسط John Resig به عنوان جایگزینی برای کتابخانه‌های پیچیده جاوااسکریپت ایجاد شد و به طراحان و توسعه دهندگان امکان بهره‌گیری از قابلیت‌های پیشرفته جاوااسکریپت البته با کدنویسی ساده‌تر، در وبسایت‌هایشان را داد.

jQuery به دلیل عدم نیاز به دانش برنامه‌نویسی پیچیده به منظور انجام انواع تغییرات و دستکاری در DOM، بسیار جذاب است.

این کتابخانه مزیت‌های بسیار زیادی دارد و در مقایسه با سایر کتابخانه‌ها از ساده‌ترین و مختصرترین syntax برخوردار است. در ادامه به بررسی ویژگی‌های اصلی jQuery می‌پردازیم:

منبع باز بودن

کتابخانه‌های جاوااسکریپت به وسیله جامعه منبع باز، پشتیبانی و به خوبی بررسی و به‌روزرسانی شده‌اند. جامعه منبع باز شبکه پشتیبانی گسترده‌ای دارد. طراحان و توسعه‌دهندگان وب به طور مداوم در حال ایجاد آموزش‌ها، کتاب‌ها و پلاگین‌هایی برای کمک به گسترش و توسعه jQuery هستند.

1- Open Source

مستندات گسترده

یکی از بزرگ‌ترین مزیت‌های jQuery که آن را به کتابخانه‌ای گسترده تبدیل کرده است مستندات ارائه شده برای آن است. تیم پشتیبان jQuery وقت زیادی را صرف مستندسازی نحوه کارکرد این کتابخانه کرده‌اند.

سایت اختصاصی مستندات jQuery شامل آموزش‌های کاملی همراه با نمونه کدها و به علاوه جامعه عظیمی از افراد برای پشتیبانی آن در سطح وب می‌باشد. می‌توانید برای اطلاعات بیشتر به آدرس docs.jquery.com مراجعه نمایید. جامعه برنامه‌نویسان و توسعه‌دهندگان jQuery به طور مداوم در حال ارتقا و انتشار نسخه‌های جدید آن هستند. به‌روزرسانی‌های مداوم نیز یکی دیگر از دلایل محبوبیت این کتابخانه به شمار می‌رود. کتابخانه‌هایی که به‌روزرسانی نمی‌شوند از محبوبیت چندانی برخوردار نیستند. زمانی که نسخه جدیدی از این کتابخانه ارائه می‌شود مستندات jQuery نیز برای آن دسته از متدهایی که در نسخه جدید منسوخ و یا اضافه شده‌اند و همچنین برای اطمینان از سازگاری آن با نسخه‌های قبلی به‌روزرسانی می‌شود.

جاوااسکریپت با کدهای کمتر

jQuery همان جاوااسکریپت است: هر کاری که می‌توان با جاوااسکریپت انجام داد با jQuery نیز انجام‌پذیر است. در هنگام کدنویسی jQuery سه اختیار خواهید داشت:

❖ استفاده از API گسترده jQuery

❖ استفاده یا ایجاد پلاگین‌های jQuery

❖ نوشتن دستورات جاوااسکریپت

یکی دیگر از مزایای جذاب این کتابخانه اختصار کدهاست. به عنوان مثال، چنانچه بخواهیم از طریق خود جاوااسکریپت رنگ پس زمینه یک عنصر <div>، با ID برابر mydiv را تغییر دهیم باید از کدی شبیه زیر استفاده کنیم:

```
document.getElementById('mydiv').style.backgroundColor = 'red';
```

در حالی که با استفاده از موتور قدرتمند گزینش jQuery، تنها نیاز به دستور کوتاه‌تری به صورت زیر داریم:

```
$('#mydiv').css('background-color', 'red');
```

یادگیری و به خاطر سپاری این syntax برای توسعه‌دهندگان وب بسیار آسان‌تر است. موتور گزینشگر jQuery یکی از برجسته‌ترین و دوست‌داشتنی‌ترین ویژگی‌های این کتابخانه است. این موتور با فراهم آوردن امکان استفاده از گزینشگرهای CSS2 و CSS3 برای انتخاب عناصر صفحه، کار را برای توسعه‌دهندگانی که با این زبان آشنایی دارند فوق‌العاده آسان کرده است.

زنجیره‌سازی^۱

یکی از بهترین ویژگی‌های jQuery زنجیره‌سازی است که امکان استفاده از چندین متد به صورت پشت سرهم و زنجیروار را می‌دهد. این ویژگی موجب کاهش حجم کدنویسی و در نتیجه افزایش سرعت دریافت و بارگذاری کدها از سرور و اجرای آن‌ها خواهد شد.

در کد زیر مثالی از کدهای jQuery را که از ویژگی‌های زنجیره‌سازی استفاده کرده است مشاهده می‌کنید:

```
$('#foo').addClass('active').prev().removeClass('active');
```

کد زیر مثال فوق را بدون استفاده از این ویژگی نشان می‌دهد:

```
$('#foo').addClass('active');
$('#foo').prev().removeClass('active');
```

مثالی که از ویژگی زنجیره‌سازی استفاده کرده است واضح‌تر بوده و روشی کوتاه‌تر برای نوشتن دستورات jQuery به شمار می‌رود.

سازگاری با مرورگرها

با توجه به روزرسانی‌های اخیر مرورگرهایی همچون Google, Internet Explorer, Firefox, Safari و Chrome و Opera ایجاد صفحاتی که به طور یکسان در تمامی آن‌ها کار کند در اولویت قرار می‌گیرد. جنگ مرورگرها در حال تبدیل شدن به بخشی از جدال روزانه توسعه‌دهندگان وب است.

1- Chaining

هنگام استفاده از jQuery می‌توانید از اینکه کدهایتان در تمامی نسخه‌های مرورگرهای مدرن همچون Google Chrome و Opera 9.0+، Safari 3.0+، Mozilla Firefox2+، Internet Explorer 6.0+ درستی کار کند اطمینان داشته باشید.

همیشه یکی از دغدغه‌های کار با جاوااسکریپت این بوده است که نیاز به کدهای متفاوتی برای پشتیبانی از برخی ویژگی‌ها در مرورگرهای مختلف داشته‌ایم. بسیاری از توسعه‌دهندگان وب از قواعد CSS خاص هر مرورگر برای پشتیبانی درست آن‌ها در مرورگرهای مختلف مخصوصاً Internet Explorer استفاده می‌کنند. همین موضوع برای کدهای جاوااسکریپت نیز صدق می‌کند. به این کدها که نشان‌دهنده پیاده‌سازی یک درخواست Ajax در جاوااسکریپت است و در چندین مرورگر به درستی کار می‌کند، دقت کنید:

```

If(window.XMLHttpRequest)
{
    xhr = new XMLHttpRequest(); //Code for Firefox/Safari
}
elseif(window.ActiveXObject) //Active X Version
{
    xhr = new ActiveXObject("Microsoft.XMLHTTP"); // For IE
}
  
```

همان‌طور که مشاهده می‌کنید استفاده از خود جاوااسکریپت برای این کار، نیاز به دو روش کاملاً مجزا، تست کردن و اصلاح خطاهای احتمالی موجود در آن‌ها دارد.

در مقابل، تکه کد زیر نشان‌دهنده سادگی انجام همان درخواست Ajax در jQuery است:

```
$.ajax();
```

مطابقت با CSS3

تمامی مرورگرهای مدرن از CSS1 و CSS2 به خوبی پشتیبانی کرده و بسیاری از توسعه‌دهندگان وب نیز این روزها از CSS2 در طراحی‌هایشان استفاده می‌کنند. CSS3 نیز ارائه شده است و از ویژگی‌های برجسته‌ای همچون استفاده از فونت‌های **توکار**^۱، گوشه‌های گرد برای انواع عناصر، تصاویر پس‌زمینه با قابلیت‌های پیشرفته، افکت‌های متنی و انیمیشن‌ها پشتیبانی می‌کند.

1- Embedded fonts

jQuery از گزینشگرهای CSS3 مانند گزینشگرهای صفت نیز به خوبی پشتیبانی می‌کند. این گزینشگرها امکان انتخاب و تغییر سبک عناصر براساس مقدار صفاتشان را می‌دهند و حتی می‌توان انتخاب آن‌ها را براساس مقادیر خاص موجود در صفت‌های آن‌ها فیلتر کرد. نمونه کد زیر را مشاهده کنید:

```
p[title=*foo] {background:black;color:white}
<p class="text" title="food is good foo you">This is my
sample text</p>
```

در فصل دوم «گزینشگرها» به بررسی تمامی گزینشگرهایی که jQuery از آن‌ها پشتیبانی می‌کند خواهیم پرداخت.

شروع به کار با jQuery

از آنجایی که مراحل لازم برای شروع به کدنویسی jQuery، نسبت به سایر کتابخانه‌ها حداقلی است، این کتابخانه به محبوب‌ترین گزینه انتخابی در میان طراحان و توسعه‌دهندگان وب تبدیل شده است. تنها کافی است کپی ساده‌ای از فایل این کتابخانه را دانلود کرده و با استفاده از یک تگ <script> ساده آن را به صفحه الحاق کنید. همانند تمامی کتابخانه‌های جاوااسکریپت، برای استفاده از ویژگی‌ها و قابلیت‌های jQuery باید آن را به روشی که گفته شد به صفحات وب و برنامه‌های کاربردی مورد نظر الحاق کنید.

دریافت jQuery

وبسایت رسمی jQuery به آدرس <http://www.jquery.com/> همیشه بهترین و به روزترین منبع برای کدها و اخبار مرتبط با این کتابخانه است. برای شروع کار با jQuery نیاز به یک کپی ساده از فایل این کتابخانه داریم که می‌توانیم آن را از صفحه ابتدایی سایت فوق به راحتی دانلود کنیم. همیشه چندین نسخه از این کتابخانه موجود است اما مناسب‌ترین نسخه آن، آخرین نسخه غیر فشرده^۱ است. در زمان نوشتن این کتاب آخرین نسخه موجود، 1.6.2 بوده است.

برای دریافت فایل این کتابخانه، پس از مراجعه به آدرس فوق، در قسمت "GRAB THE LATEST VERSION" دو نسخه متفاوت در اختیار شما قرار می‌گیرد. یکی نسخه Development که نسخه غیر فشرده و پر حجم (حدود 229 KB) این کتابخانه محسوب شده و برای آن

1- Uncompressed

دسته از افرادی که علاقه‌مند به مشاهده کدها و نحوه عملکرد این کتابخانه هستند، ارائه می‌شود و دیگری نسخه Production که نسخه کم حجم (حدود 31 KB) می‌باشد و به منظور افزایش سرعت بارگذاری در صفحه، بسیاری از کاراکترهای غیرضروری موجود در نسخه اصلی آن، همچون **توضیحات**^۱، **فضاهای خالی**^۲، **شکست خطوط**^۳ و زبانه‌ها، حذف شده‌اند:



اکنون کافی است پس از انتخاب نسخه مورد نظر، روی دکمه Download کلیک کنید. هیچ گونه نصبی برای استفاده از jQuery لازم نیست. برای استفاده از این کتابخانه، تنها کافی است آن را در قسمتی از سایتمان روی وب قرار دهیم. همانطور که جاوااسکریپت یک زبان فقط تفسیر شدنی است هیچ دغدغه‌ای برای کامپایل کدهای jQuery نیز وجود ندارد. در هر صفحه‌ای که نیاز به استفاده از قابلیت‌های jQuery داشته باشید کافی است با استفاده از یک تگ `<script>` ساده در صفحه HTML مورد نظر موقعیت فایل jQuery را تعیین نمایید. به صورت زیر:

```
<script src="jquery.1.6.2.js" language='javascript'
type="text/javascript"></script>
```

روشی جایگزین برای الحاق کردن فایل مربوط به کتابخانه jQuery، استفاده از شبکه **تحویل محتوا**^۴ (CDN) گوگل است. CDN شبکه‌ای از کامپیوترهاست که اختصاصاً برای ارائه انواع محتوا به کاربران به روشی سریع طراحی شده است. این سرویس‌دهنده‌ها اغلب به صورت جغرافیایی توزیع شده‌اند به طوری که هر درخواست ارسالی با نزدیک‌ترین سرور موجود در شبکه پاسخ داده می‌شود.

-
- 1- Comments
 - 2- White space
 - 3- Line breaks
 - 4- Content Delivery Network

گوگل از بسیاری از کتابخانه‌های مشهور و متن باز از جمله jQuery روی CDNهای خود میزبانی می‌کند. استفاده از CDN گوگل برای الحاق کتابخانه‌های جاوااسکریپت به صفحه، مزیت‌هایی دارد که عبارتند از:

- ❖ همیشه می‌توان به راحتی به آخرین نسخه‌های ارائه شده کتابخانه‌ها دسترسی داشت.
- ❖ به دلیل عدم قراردادن فایل کتابخانه‌ها روی سرورهای خود، در پهنای باند مصرفی صرفه‌جویی می‌شود.
- ❖ می‌توان از خصوصیات سرورهای گوگل، همچون سرعت بالا، قابل اطمینان بودن و همیشه در دسترس بودن بهره برد.
- ❖ از آنجایی که ممکن است سایر سایت‌ها نیز از این روش برای قرار دادن فایل‌های کتابخانه‌ها در صفحاتشان استفاده کرده باشند، با ذخیره آن‌ها در حافظه نهان مرورگرها، امکان بارگذاری آن‌ها از cache در هنگام بازدید از سایت شما به جای دانلود مجددشان، فراهم شده و به این شکل موجب افزایش سرعت بارگذاری صفحات می‌شود.

روش‌های گوناگونی برای الحاق فایل jQuery از طریق CDN گوگل وجود دارد که ساده‌ترین آن‌ها به صورت زیر است:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.2/jquery.min.js"></script>
```

کد فوق شبیه همان کد قبلی است جز اینکه در تگ <script>، به جای فایلی در داخل سایت خود، به یکی از سرورهای گوگل اشاره شده است. اگر به آدرسی که به سرور گوگل اشاره دارد نگاهی دقیق بیندازید می‌بینید که نسخه jQuery از طریق یکی از مسیرها مشخص می‌شود. (در مثال ما: 1.6.2). با این حال اگر بخواهید از آخرین نسخه سری خاصی از jQuery استفاده کنید می‌توانید یک عدد را از انتهای رشته حاوی نسخه حذف کنید (مثلاً 1.6). حتی می‌توان به جای آن‌ها از عدد ساده 1 استفاده کرد که در این صورت سرورهای گوگل آخرین نسخه ارائه شده از jQuery تا آن زمان را برمی‌گردانند. چنانچه مایل به بررسی روش‌های پیچیده استفاده از فایل‌های موجود روی CDNهای گوگل، هستید می‌توانید به آدرس زیر مراجعه نمایید:

<http://code.google.com/apis/ajaxlibs/documentation/>

ساختار اسکریپت‌های jQuery

اکنون که با نحوه قراردادن فایل jQuery در صفحه آشنا شدیم اجازه بدهید کمی در مورد کارهایی که این کتابخانه می‌تواند انجام دهد صحبت کنیم. شاید در نگاه اول syntax آن کمی عجیب و غریب به نظر برسد اما در اصل بسیار ساده می‌باشد. پس از نوشتن اولین برنامه، سبک و syntax دستورات jQuery ملکه ذهنتان شده و شما را به نوشتن کدهای بیشتر ترغیب خواهد کرد.

\$() جایگزینی برای jQuery()

الحاق کردن فایل jQuery به صفحه امکان استفاده از تابع ساده‌ای به نام jQuery() را فراهم می‌کند. از طریق این تابع است که jQuery هزاران قابلیت را به منظور افزودن بُعد جدیدی به صفحات، در اختیار شما قرار می‌دهد.

البته این کتابخانه یک تابع جایگزین کوتاه‌تر به نام \$() برای استفاده به جای تابع jQuery() و با همان قابلیت‌ها را فراهم آورده است. علامت \$، به عنوان نامی کوتاه، معتبر و خوش فرم برای یک متغیر در جاوااسکریپت به شمار می‌رود. استفاده از تابع \$() به جای jQuery() افزایش خوانایی و کوتاه‌تر شدن کدها را به همراه دارد.

آشنایی با مفهوم مجموعه عناصر انتخابی^۱

در سراسر این کتاب از مفهومی با عنوان **مجموعه عناصر انتخابی** به وفور استفاده کرده‌ایم که باید قبل از شروع به کار جدی و برنامه‌نویسی با کدهای jQuery با این مفهوم آشنا شویم. همان طور که دیدیم این کتابخانه از تابعی به نام jQuery() یا \$() پشتیبانی می‌کند. از این تابع برای انتخاب عناصر یک صفحه استفاده می‌شود. این تابع از گزینشگرهای CSS برای انتخاب عناصری از DOM و در اصل صفحه جاری استفاده می‌کند. این توابع همیشه شیء ویژه‌ای حاوی آرایه‌ای از عناصری که با گزینشگر مطابقت داده می‌شوند (و البته به همان ترتیبی که در صفحه قرار دارند) را برمی‌گردانند. از این شیء برای فراخوانی متدهای تعریف شده jQuery استفاده می‌شود. به این شیء به دلیل اینکه می‌تواند حاوی یک یا چندین عنصر از DOM باشد **مجموعه عناصر انتخابی** یا در برخی از منابع **مجموعه پوشاننده**^۲ (به دلیل اینکه چندین عنصر را پوشش می‌دهد) گفته می‌شود.

1- Matched Elements set

2- Wrapped set

تشریح دستورات jQuery

اکنون که متوجه شدیم دستورات jQuery تنها با فراخوانی ساده تابع jQuery() یا حالت کوتاه‌تر آن یعنی (\$) آغاز می‌شوند به سراغ سایر قسمت‌های باقیمانده از آن‌ها می‌رویم. این تصویر هر دو روش یکسان دستورات عمومی این کتابخانه (که در یکی، از نام کامل تابع و در دیگری از جایگزین آن یعنی \$ استفاده شده است) را نشان می‌دهد:

selector	action	parameters
jQuery('p')	.css	('color', 'blue');
\$('.p')	.css	('color', 'blue');

هر دستور از چهار بخش اساسی تشکیل شده است: تابع jQuery() (یا جایگزین آن)، گزینه‌گرها، عملیات و پارامترها. در ابتدا، از تابع jQuery() یا (\$) به همراه یک گزینه‌گر برای انتخاب یک یا چند عنصر موجود در صفحه استفاده کردیم. در کد فوق p یک گزینه‌گر نوع دار است که تمامی تگ‌های <p> موجود در صفحه را انتخاب می‌کند. (برای اطلاع از جزئیات تابع jQuery() به فصل سوم و انواع گزینه‌گرها به فصل دوم مراجعه کنید.) پس از آن عملی را که قرار است روی هر یک از این عناصر انجام شود انتخاب می‌کنیم. منظور از این عملیات همان متدها هستند که در سراسر این کتاب با انواع آن‌ها آشنا خواهیم شد. در آخر نیز پارامترهایی را برای نحوه دقیق انجام اعمال روی مجموعه عناصر انتخابی مشخص کرده‌ایم. سعی کنید هر موقع که کدهای jQuery را می‌بینید آن‌ها را به این بخش‌ها تقسیم‌بندی کنید. این کار فهم و درک آن‌ها را برایتان بسیار آسان‌تر می‌کند.

در مثال فوق، از گزینه‌گری برای انتخاب تمامی پاراگراف‌های صفحه (تگ‌های <p>) استفاده کرده‌ایم. و در ادامه متد (عمل) css() موجود در jQuery را انتخاب کرده‌ایم که از آن برای دستکاری قواعد CSS اعمال شده به عناصر استفاده می‌شود. در انتها نیز از دو پارامتر برای اینکه مقدار خاصیت color همه آن‌ها را برابر blue قرار دهیم استفاده کرده‌ایم. نتیجه پایانی چه خواهد بود؟ اینکه تمامی پاراگراف‌ها، آبی رنگ می‌شوند!

در مثال فوق، متد css() دو پارامتر (color و blue) را پذیرفته است اما تعداد پارامترهای ارسالی به متدها می‌تواند متفاوت باشد. برخی متدها به هیچ پارامتری نیاز نداشته، برخی دیگر چندین مجموعه از پارامترها را (برای انجام چندین تغییر به صورت همزمان) پذیرفته و برخی دیگر نیاز به تعیین تابعی به منظور اجرا در هنگام رخداد یک رویداد دارند (مانند عنصری که روی آن کلیک می‌شود). با این حال تمامی دستورات از این ساختار اساسی پیروی می‌کنند.

jQuery بارگذاری صفحه را در دست می‌گیرد

هنگام کار روی عناصر موجود در صفحه، مثلاً به منظور اعمال مجموعه‌ای از قواعد سبک، نیاز به دسترسی سریع و فوری به آن‌ها از همان ابتدای نمایش صفحه داریم. اما کدهای جاوااسکریپت معمولاً در قسمت <head> صفحه نوشته می‌شوند که قبل از سایر عناصر صفحه بارگذاری می‌شود در حالی که عناصر مورد نظر ما بعد از آن در قسمت <body> صفحه قرار دارند. برای این کار نمی‌توان از رویداد onload شیء window مرورگر استفاده کرد زیرا این رویداد نه تنها دسترسی به عناصر را تا زمان ایجاد درخت DOM بلکه تا زمان بارگذاری کامل تمام عناصر صفحه (مانند عکس‌ها، انیمیشن‌های Flash، فیلم‌های QuickTime و...) به تأخیر می‌اندازد:

```

window.onload = function() {
    // do stuff here
};

```

این دستور نحوه استفاده از رویداد onload شیء window را نشان می‌دهد. در این دستور از یک تابع بدون نام که به رویداد onload این شیء اختصاص داده شده، استفاده شده است. در این روش، دستورات باید داخل بدنه این تابع نوشته شوند. در نتیجه خود جاوااسکریپت راه حل مناسبی برای این کار ارائه نمی‌کند.

این در حالی است که jQuery امکان دسترسی به تمامی عناصر موجود در صفحه بدون منتظر ماندن بارگذاری عکس‌ها را فراهم می‌کند و می‌تواند کدهای موردنظر را فقط به محض بارگذاری و آماده شدن DOM اجرا نماید و دیگر برای بارگذاری سایر عناصر منتظر نماند.

jQuery متدهای بسیاری دارد که یکی از مهم‌ترین آن‌ها به نام document.ready() است که دقیقاً به محض بارگذاری DOM اجرا می‌شود. می‌توانید تمامی کدهای jQuery را داخل بدنه این متد به روشی که در ادامه خواهد آمد قرار داده و از اینکه آن‌ها به محض بارگذاری و آماده شدن DOM اجرا می‌شوند اطمینان داشته باشید.

روش استفاده از این متد به همراه شیء document به صورت زیر است:

```

<script type='text/javascript' language='javascript'>
    $(document).ready(function() {
        //your Code Comes Here
    });
</script>

```

1- Style

برای نمونه در مثال بعد با استفاده از تابع (\$) که برای انتخاب عناصر و اشیای خاصی از صفحه استفاده می‌شود- ابتدا شیء document را انتخاب کرده و سپس متد ready() را برای آن فراخوانی کرده و از یک تابع بدون نام برای نوشتن دستوراتی که می‌خواهیم به محض بارگذاری صفحه اجرا شوند استفاده کرده‌ایم.

با اجرای این دستور، به محض بارگذاری صفحه، تمامی عناصر <div> که صفت class آن‌ها برابر myDiv است مخفی خواهند شد.

```
<script type='text/javascript' language=' javascript' >
  $(document).ready(function() {
    $("div.myDiv").hide();
  });
</script>
```

در مورد تابع (\$) و متد ready() در فصل‌های بعد بیشتر خواهیم آموخت.

مثالی عملی: فهرستی پویا از محتوای صفحه

به عنوان مثالی عملی از jQuery، برنامه‌ای ساده که به صورت پویا، عناوین یک صفحه HTML را استخراج کرده و آن‌ها را در قالب فهرستی از محتوای صفحه گردآوری می‌کند خواهیم ساخت. فهرست نشان‌دهنده محتوا، همانطور که در تصویر ذیل مشاهده می‌کنید در گوشه بالا سمت راست صفحه جای خواهد گرفت.

[Page Contents](#)

Doctor Dolittle
by Hugh Lofting

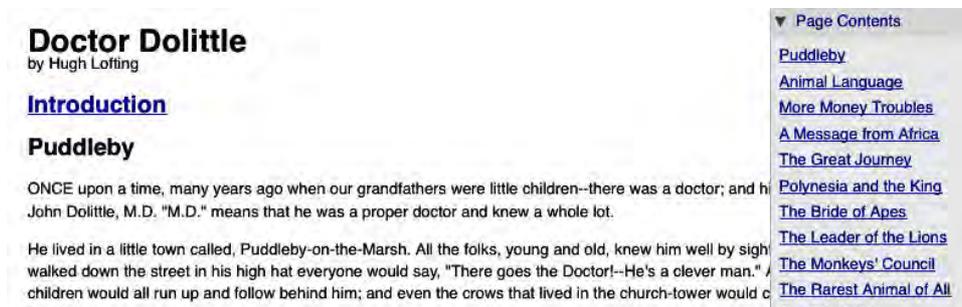
Introduction

Puddleby

ONCE upon a time, many years ago when our grandfathers were little children--there was a doctor; and his name was Dolittle--John Dolittle, M.D. "M.D." means that he was a proper doctor and knew a whole lot.

He lived in a little town called, Puddleby-on-the-Marsh. All the folks, young and old, knew him well by sight. And whenever he walked down the street in his high hat everyone would say, "There goes the Doctor!--He's a clever man." And the dogs and the children would all run up and follow behind him; and even the crows that lived in the church-tower would caw and nod their

همانطور که نشان داده شده است، از ابتدا تنها بخشی از فهرست را نمایش خواهیم داد که تنها با یک کلیک می‌توان آن را به صورت کامل مشاهده کرد.



Doctor Dolittle
by Hugh Lofting

Introduction

Puddleby

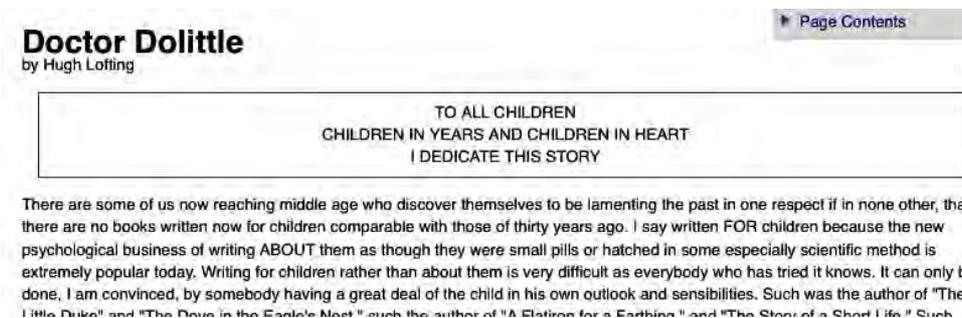
ONCE upon a time, many years ago when our grandfathers were little children--there was a doctor; and his name was John Dolittle, M.D. "M.D." means that he was a proper doctor and knew a whole lot.

He lived in a little town called, Puddleby-on-the-Marsh. All the folks, young and old, knew him well by sight. When he walked down the street in his high hat everyone would say, "There goes the Doctor!--He's a clever man." All the children would all run up and follow behind him; and even the crows that lived in the church-tower would c

Page Contents

- Puddleby
- Animal Language
- More Money Troubles
- A Message from Africa
- The Great Journey
- Polynesia and the King
- The Bride of Apes
- The Leader of the Lions
- The Monkeys' Council
- The Rarest Animal of All

همزمان ویژگی جدیدی را به متن اصلی صفحه اضافه خواهیم کرد. مقدمه متن از همان ابتدا در صفحه بارگذاری نخواهد شد اما با کلیک کاربر روی Introduction، متن مقدمه از فایلی دیگر فراخوانی شده و به جای آن قرار خواهد گرفت.



Doctor Dolittle
by Hugh Lofting

Page Contents

TO ALL CHILDREN
CHILDREN IN YEARS AND CHILDREN IN HEART
I DEDICATE THIS STORY

There are some of us now reaching middle age who discover themselves to be lamenting the past in one respect if in none other, that there are no books written now for children comparable with those of thirty years ago. I say written FOR children because the new psychological business of writing ABOUT them as though they were small pills or hatched in some especially scientific method is extremely popular today. Writing for children rather than about them is very difficult as everybody who has tried it knows. It can only be done, I am convinced, by somebody having a great deal of the child in his own outlook and sensibilities. Such was the author of "The Little Duke" and "The Dove in the Eagle's Nest" such the author of "A Flamingo for a Feather" and "The Story of a Short Life." Such

قبل از اینکه چگونگی انجام این کارها توسط اسکریپت را نشان بدهیم باید محیطی که اسکریپت بر پایه آن پیاده سازی شده است بررسی کنیم.

پیاده‌سازی صفحه HTML

در اکثر کاربردهای jQuery سه قسمت اصلی وجود دارد. خود صفحه HTML، فایل‌های CSS برای تعیین ظاهر نمایشی و فایل‌های جاوااسکریپت برای تعیین انجام عملیات مورد نظر روی آن. برای این مثال، از صفحه‌ای حاوی متن‌های یک کتاب استفاده خواهیم کرد:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
  <title>Doctor Dolittle</title>
  <link rel="stylesheet" href="dolittle.css"
type="text/css" media="screen" />
  <script src="jquery.js"
type="text/javascript"></script>
  <script src="dolittle.js"
type="text/javascript"></script>
</head>
<body>
  <div class="container">
    <h1>Doctor Dolittle</h1>
    <div class="author">by Hugh Lofting</div>
    <div id="introduction">
      <h2><a
href="introduction.html">Introduction</a></h2>
    </div>
    <div class="content">
      <h2>Puddleby</h2>
      <p>ONCE upon a time, many years ago when
our grandfathers were little children--there was a
doctor; and his name was Dolittle-- John Dolittle, M.D.
&quot;M.D.&quot; means that he was a proper
doctor and knew a whole lot.</p>
      <!-- More text follows... -->
    </div>
  </div>
</body>
</html>

```

نمونه پینش فایل‌ها روی سرور اهمیتی ندارد. اشاره به یک فایل از داخل فایل دیگر تنها نیاز به تنظیماتی براساس نوع سازماندهی انتقابی برای فایل‌ها، دارد. در بسیاری از مثال‌های این کتاب، برای اشاره به فایل‌ها از مسیرهای نسبی (./images/foo.png) به جای مسیرهای مبنی بر ریشه سایت (/images/foo.png) استفاده خواهیم کرد. این کار، امکان راه‌اندازی کدها به صورت مملی (local) و بدون نیاز به سرورها را فراهم می‌کند.



بلافاصله به دنبال عنصر <head>، فایل CSS بارگذاری شده است. بخشی از این فایل که روی عناصر پویای صفحه تأثیر می‌گذارد به صورت زیر است:

```
/** =page contents
*****/
#page-contents {
  position: absolute;
  text-align: left;
  top: 0;
  right: 0;
  width: 15em;
  border: 1px solid #ccc;
  border-top-width: 0;
  background-color: #e3e3e3;
}
#page-contents a {
  display: block;
  margin: .25em 0;
}
#page-contents a.toggler {
  padding-left: 20px;
  background: url(arrow-right.gif) no-repeat 0 0;
  text-decoration: none;
}
#page-contents a.arrow-down {
  background-image: url(arrow-down.gif);
}
#page-contents div {
  padding: .25em .5em .5em;
  display: none;
  background-color: #efefef;
}
/** =introduction
*****/

.dedication {
  margin: 1em;
  text-align: center;
  border: 1px solid #555;
  padding: .5em;
}
```

پس از فراخوانی فایل CSS، فایل‌های جاوااسکریپت به صفحه الحاق شده‌اند. قراردادن تگ `<script>` مربوط به کتابخانه jQuery **قبل** از تگ `<script>` مربوط به فایل‌های دلخواه دیگر، اهمیت دارد. اگر این ترتیب رعایت نشود ممکن است کتابخانه jQuery هنگامی که کدی به آن مراجعه می‌کند در دسترس نباشد.

به منظور نمایش سریع‌تر عناصر بصری موجود در صفحه، بسیاری از توسعه‌دهندگان ترمین می‌دهند فایل‌های جاوااسکریپت را در انتهای صفحه و درست قبل از تگ بسته `</body>` قرار دهند. به همین دلیل فایل‌های جاوااسکریپت تا زمان بارگذاری اکثر بخش‌های صفحه از سرور درخواست نخواهند شد. به منظور اطلاعات بیشتر در مورد بهبود کارایی‌های این روش به آدرس زیر مراجعه نمایید:

http://develpoper.yahoo.com/performance/rules.html#js_bottom

نوشتن کدهای jQuery

کدهای سفارشی و دلخواه ما در فایل دوم جاوااسکریپتی که هم اکنون خالی است و با استفاده از `<script src="dolittle.js" type="text/javascript"></script>` به صفحه HTML الحاق کرده‌ایم قرار دارد. با توجه به وسعت کاری که این کدها انجام می‌دهند، اما نسبتاً کوتاه به نظر می‌رسند:

```
jQuery.fn.toggleNext = function() {
    this.toggleClass('arrow-down')
    .next().slideToggle('fast');
    return this;
};
$(document).ready(function() {
    $('<div id="page-contents"></div>')
    .prepend('<a class="toggler" href="#">Page
Contents</a>')
    .append('<div></div>')
    .prependTo('body');
    $('<div> .content h2').each(function(index) {
        var $chapterTitle = $(this);
        var chapterId = 'chapter-' + (index + 1);
        $chapterTitle.attr('id', chapterId);
        $('<a></a>').text($chapterTitle.text())
        .attr({
            'title': 'Jump to ' +
$chapterTitle.text(),
            'href': '#' + chapterId
        })
        .appendTo('#page-contents div');
    });
    $('#page-contents > a.togger').click(function() {
        $(this).toggleNext();
        return false;
    });
    $('#introduction > h2 a').click(function() {
        $('#introduction').load(this.href);
        return false;
    });
});
```

اکنون فهرستی پویا از محتوا داریم که کاربر را به بخش‌های مربوطه از متن هدایت می‌کند و مقدمه‌ای که در صورت نیاز می‌تواند بارگذاری شود.

تشریح اسکرپت

این اسکرپت به دلیل اینکه قابلیت‌های گسترده‌ای از jQuery را نشان می‌دهد انتخاب شده است. اکنون که کد کامل این برنامه را دیده‌ایم می‌توانیم دسته‌بندی متدهای مورد استفاده در آن را مشخص کنیم.

ما جزئیات کاملی در مورد عملکرد این اسکرپت ارائه نخواهیم کرد اما اسکرپت مشابهی به همراه آموزش آن در وبلاگ آموزش jQuery در آدرس زیر موجود می‌باشد:

<http://www.learningjquery.com/2007/06/automatic-page-contents>

دستورات گزینشگر^۱

قبل از انجام عملیات روی صفحه HTML نیاز به انتخاب بخش‌های مورد نظر در آن خواهیم داشت. در اسکرپت‌های خود، اغلب از روشی ساده برای انتخاب یک عنصر به صورت زیر استفاده خواهیم کرد:

```
$('#introduction')
```

این دستور یک شیء jQuery جدید را که اشاره به عنصری با صفت ID برابر introduction خواهد داشت ایجاد می‌کند. از طرف دیگر گاهی اوقات نیاز به گزینشگر پیچیده‌تری داریم:

```
$('#introduction > h2 a')
```

در اینجا شیء jQuery را که اشاره به چندین عنصر خواهد داشت ایجاد کرده‌ایم. با این دستور، عناصر لینکی که جزء اولاد عناصر <h2> که خود این عناصر <h2>، فرزندان عنصری با صفت ID برابر introduction هستند انتخاب می‌شوند.

دستورات گزینشگرها می‌توانند ساده یا در صورت نیاز پیچیده باشند. فصل ۲ «گزینشگرها» تمامی گزینشگرهای موجود و نحوه ترکیب آن‌ها را توضیح خواهد داد.

متدهای پیمایش DOM

گاهی اوقات، شیء jQuery که اشاره به مجموعه‌ای از عناصر موجود در مدل شیء‌گرای سند (DOM) دارد در اختیار داشته اما می‌خواهیم عملیاتی را روی مجموعه‌ای دیگر از عناصر که در رابطه با آن‌ها هستند انجام دهیم. در این حالت، متدهای پیمایش DOM مفید خواهند بود. می‌توانیم این مورد را در

1- Selector Expression

بخشی از اسکریپت خود مشاهده کنیم:

```
this.toggleClass('arrow-down')
.next()
.slideToggle('fast');
```

در این بخش از کد، کلمه کلیدی `this` اشاره به یک شیء `jQuery` دارد. (در حالی که در اغلب اوقات از آن برای اشاره به عنصری از `DOM` استفاده می‌شود). در کد فوق این شیء `jQuery` اشاره به لینک `toggler` در فهرست محتوا دارد. فراخوانی `toggleClass()` موجب دستکاری و تغییر این عنصر خواهد شد. به دنبال آن، `next()` عنصری را که در حال کار روی آن هستیم تغییر داده در نتیجه فراخوانی `slideToggle()` عملیاتی را روی تگ `<div>` در برگیرنده محتوا به جای لینک کلیک شده انجام می‌دهد. متدهای مشابهی که امکان جابه‌جایی آسان در درخت `DOM` را فراهم می‌کنند در فصل ۳ «متدهای پیمایش `DOM`» فهرست شده‌اند.

متدهای دستکاری `DOM`

تنها انتخاب عناصر کافی نیست و اغلب می‌خواهیم قادر به تغییر آن‌ها نیز باشیم. بسیاری از این تغییرات می‌تواند به آسانی تغییر یک صفت ساده باشد:

```
$chapterTitle.attr('id', chapterId);
```

در اینجا صفت `ID` عنصری را که متغیر `$chapterTitle` به آن‌ها اشاره دارد تغییر داده‌ایم.

گاهی اوقات تغییرات گسترده‌تر هستند:

```
$('<div id="page-contents"></div>')
.prepend('<a class="toggler" href="#">Page Contents</a>')
.append('<div></div>')
.prependTo('body');
```

این بخش از اسکریپت، نشان می‌دهد که متدهای دستکاری `DOM`، نه تنها می‌توانند عناصر را در جایشان تغییر دهند بلکه از آن‌ها می‌توانیم برای حذف، ترکیب کردن و درج آن‌ها در سایر نقاط استفاده کنیم. این خط، لینکی جدید در ابتدای `<div id="page-contents">` اضافه کرده، سپس عنصر `<div>` دیگری را در انتهای آن درج کرده و سپس کل آن را در ابتدای بخش `<body>` صفحه قرار می‌دهد. فصل ۴ «متدهای دستکاری `DOM`» سایر روش‌های تغییر و دستکاری در `DOM` و جزئیات آن‌ها را تشریح خواهد کرد.

متدهای رویدادها

حتی زمانی که بتوانیم صفحه را با اختیار خود تغییر دهیم، باز هم صفحات ما نمی‌توانند در برابر اتفاقات رخ داده عکس‌العملی نشان دهند. به منظور واکنش نشان دادن به ورودی‌های کاربر و انجام تغییرات در زمان‌های مناسب، نیاز به متدهای رویدادها خواهیم داشت.

```
$('#introduction > h2 a').click(function() {
    $('#introduction').load(this.href);
    return false;
});
```

در این تکه برنامه، تابع کنترلگری را که در هر بار کلیک روی لینک انتخاب شده اجرا خواهد شد، ثبت کرده‌ایم. رویداد کلیک یکی از متداول‌ترین رویدادهایی است که همیشه دیده‌ایم در حالی که رویدادهای زیاد دیگری نیز وجود دارند. متدهای jQuery که با این رویدادها در تعامل هستند در فصل ۵ «متدهای کار با رویدادها» بررسی شده‌اند. فصل ۵ همچنین متد رویداد بسیار ویژه‌ای به نام ready() را توضیح می‌دهد.

```
$(document).ready(function() {
    // ...
});
```

این متد امکان ثبت رویدادهایی را می‌دهد که قرار است به محض در دسترس قرار گرفتن ساختار DOM، حتی قبل از بارگذاری تصاویر، رخ بدهند.

متدهای انیمیشن

متدهای رویدادها امکان واکنش نشان دادن به ورودی‌های کاربر را می‌دهند. متدهای انیمیشنی امکان انجام عملیاتی روی ظاهر عناصر را فراهم می‌کنند. به جای مخفی کردن و نمایش دادن یکباره عناصر می‌توانیم از یک انیمیشن برای این کار استفاده کنیم.

```
this.toggleClass('arrow-down')
.next()
.slideToggle('fast');
```

این متد موجب اجرای سریع افکت slide بر روی عنصر شده و با هر فراخوانی موجب مخفی شدن و مجدداً نمایش داده شدن متوالی آن خواهد شد. متدهای انیمیشنی پیش ساخته jQuery و روش‌های ایجاد نمونه‌های جدیدی از آن‌ها، در فصل ۶ «کار با افکت (انیمیشن)ها» توضیح داده شده‌اند.

متدهای کار با AJAX

بسیاری از سایت‌های پیشرفته و مدرن، از تکنیک‌هایی برای بارگذاری انواع محتوایی که درخواست می‌شوند و آن هم بدون بارگذاری مجدد صفحه، استفاده می‌کنند. متدهای کار با AJAX، این درخواست‌ها را پیاده‌سازی کرده و امکان کنترل روند پردازششان را می‌دهند.

```
$('#introduction > h2 a').click(function() {
    $('#introduction').load(this.href);
    return false;
});
```

در اینجا، متد load() تنها در یک خط، امکان دریافت صفحه HTML دیگری از سرور و قرار دادن آن در صفحه جاری را فراهم می‌کند. این متد و سایر روش‌های پیچیده‌تر دریافت اطلاعات از سرور در فصل ۷، متدهای کار با AJAX، فهرست شده‌اند.

متدهای گوناگون

دسته‌بندی برخی از متدها دشوارتر از بقیه است. کتابخانه jQuery، شامل چندین متد گوناگون، به منظور انجام کارهای متداول جاوااسکریپت است. حتی انجام دادن کارهایی اساسی مانند پیاده‌سازی حلقه‌ها در jQuery بسیار آسان است.

```
$('#content h2').each(function(index) {
    // ...
});
```

متد each() که در اینجا مشاهده می‌کنید در طول تمامی عناصر انتخاب شده پیمایش کرده و کدهای موجود در داخل بدنه تابع بدون نام را روی هر یک از آن‌ها اجرا می‌کند. در مورد این مثال، این متد به ما در جمع‌آوری تمامی عناوین موجود در صفحه که می‌توان از ترکیب آن‌ها به منظور ایجاد فهرستی کامل از محتوا استفاده کرد کمک می‌کند. سایر توابع کمکی مشابه در فصل ۸ «متدهای گوناگون در jQuery» توضیح داده شده است.

تعدادی از سایر اطلاعات اضافی مورد نیاز در jQuery، به صورت خواصی از اشیای مربوط به آن‌ها پیاده‌سازی شده‌اند. این خواص سراسری اشیای در فصل ۹ «خواص jQuery» بررسی شده‌اند.

پلاگین‌ها

نباید خود را به توابع و متدهای پیش ساخته محدود کنیم. پلاگین‌های jQuery امکان استفاده از قابلیت‌هایی را می‌دهند که سایر افراد متناسب با نیاز ما فراهم کرده‌اند. حتی در اسکرپت کوچکی که در زیر می‌بینید از یک پلاگین استفاده شده است:

```
jQuery.fn.toggleNext = function() {
  this.toggleClass('arrow-down')
    .next().slideToggle('fast');
  return this;
};
```

این کد متد جدیدی به نام toggleNext() که عنصر بعدی را باز و بسته می‌کند، به jQuery اضافه می‌کند. اکنون می‌توانیم در صورت نیاز، این متد جدید را فراخوانی کنیم:

```
$('#page-contents > a.toggler).click(function() {
  $(this).toggleNext();
  return false;
});
```

بهرتر است همیشه کدی را که باید در خارج از اسکرپت کنونی چندین بار مورد استفاده قرار گیرد به صورت پلاگین پیاده‌سازی کنید. فصل ۱۰ «پلاگین‌ها» به بررسی پلاگین‌ها و نحوه ساخت آن‌ها اشاره خواهد کرد.

خلاصه

در این فصل با مفهوم کتابخانه و انواع آن‌ها در زبان جاوااسکرپت آشنا شده و به این نتیجه رسیدیم که jQuery به دلیل syntax ساده و مختصرتر، محبوبیت و کاربرد بیشتری نسبت به سایر کتابخانه‌ها دارد. سپس به بررسی ویژگی‌های برجسته آن همچون منبع باز بودن، سازگاری با تمامی مرورگرها، دارا بودن مستندات گسترده، پشتیبانی از CSS3 و... پرداختیم و در ادامه با ساختار اساسی دستورات jQuery آشنا شده و به بررسی تابع jQuery() و جایگزین آن (\$) و سایر بخش‌های آن همچون گزینشگرها، متدها و پارامترها پرداختیم.

سپس یک اسکرپت کامل را که بر پایه ویژگی‌های jQuery پیاده‌سازی شده بود بررسی کردیم. این مثال، با وجود کوچک بودن، قابلیت‌های تعاملی و کاربردپذیری زیادی به صفحه اضافه کرد. این

اسکریپت همچنین، عمده‌ترین ابزار ارائه شده jQuery را نشان می‌دهد. مشاهده کردیم که چگونه می‌توانیم عناصر موجود در DOM را انتخاب کرده و در صورت نیاز تغییر دهیم. همچنین شاهد چگونگی واکنش نشان دادن به عملیات کاربر و استفاده از انیمیشن‌ها به منظور پاسخگویی به آن‌ها بودیم. علاوه بر این نحوه دریافت اطلاعات از سرورها بدون بارگذاری صفحه و چگونگی اضافه کردن متدها و ویژگی‌های جدید به jQuery در قالب پلاگین‌ها را مشاهده کردیم.

در فصل‌های بعدی، به بررسی هر یک از توابع، متدها و گزینشگرهای موجود در کتابخانه jQuery خواهیم پرداخت. هر متد با خلاصه‌ای از syntax، پارامترهای دریافتی و مقادیری که برمی‌گرداند معرفی خواهد شد. سپس توضیحاتی در مورد آن‌ها داده و مثال‌هایی مناسب ارائه خواهیم کرد. برای اطلاعات بیشتر در مورد هر یک از متدها، منابع آنلاینی در ضمیمه ۱، منابع آنلاین، ارائه خواهند شد. همچنین به بررسی ساختار پلاگین‌های jQuery پرداخته و نحوه استفاده و نوشتن آن‌ها را توضیح خواهیم داد.



در کانال تلگرام کارنیل هر روز انگیزه خود را شارژ کنید 😊

<https://telegram.me/karnil>

