

به نام ایزد یکتا

مرجع کامل

Pro jQuery 2

(جلد ۲)

Adam Freeman

ترجمه: مهندس خلیل غفوری پور

انتشارات پندار پارس

سرشناسه	: فریمن، آدام، ۱۹۷۲ - م.
عنوان و نام پدیدآور	: مرجع کامل Pro jQuery 2 / آدام فریمن؛ ترجمه خلیل غفوری پور.
مشخصات نشر	: تهران : پندار پارس، ۱۳۹۵
مشخصات ظاهری	: ج-۱: مصور، جدول.
شابک	: ج. ۲، 978-600-8201-06-9 : شابک دوره: 6-978-600-8201-07-6
وضعیت فهرست نویسی	: فیپا
یادداشت	: عنوان اصلی: Pro jQuery 2.0
موضوع	: جاوا اسکریپت (زبان برنامه نویسی کامپیوتر)
موضوع	: نرم افزار متن باز
موضوع	: وبگاه ها -- توسعه
موضوع	: وبگاه ها -- طراحی
شناسه افزوده	: غفوری پور، خلیل، ۱۳۵۶ - مترجم
رده بندی کنگره	: TK/۵۱۰۵/۸۸۸۵/۱۳۹۴ ۴۳ف۲/ج
رده بندی دیویی	: ۰۰۵/۲۷۶۲
شماره کتابشناسی ملی	: ۴۰۱۰۱۶۲

انتشارات پندار پارس



دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com
 تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۲۱۴۳۷۱۹۶۴
info@pendarepars.com



نام کتاب	: مرجع کامل Pro jQuery 2 (جلد ۲)
ناشر	: انتشارات پندار پارس
تالیف	: Adam Freeman
ترجمه	: خلیل غفوری پور
چاپ نخست	: اردیبهشت ماه ۹۵
شمارگان	: ۵۰۰ نسخه
طرح جلد	: مصطفی مصباحی
چاپ، صحافی	: روز : شابک دوره : ۶-۹۷۸-۶۰۰-۸۲۰۱-۰۷-۶
قیمت	: ۳۰۰۰۰ تومان : شابک : ۹-۹۷۸-۶۰۰-۸۲۰۱-۰۶-۹



هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد

خریدار گرامی: برای دریافت کد مثال های کتاب، به صفحه کتاب در سایت پندار پارس مراجعه فرمایید.

فهرست

فصل ۲۰؛ تنظیم JQuery UI	۴۷۷
بدست آوردن JQuery UI	۴۷۷
تصمیم گیری یک تم	۴۷۷
دانلود JQuery UI شخصی	۴۷۹
نصب JQuery UI برای توسعه	۴۷۹
افزودن JQuery UI به سند HTML	۴۸۰
نصب JQuery UI برای انتشار	۴۸۱
استفاده از JQuery UI از طریق شبکه توزیع محتوا	۴۸۱
فصل ۲۱؛ استفاده از ویجت های دکمه، نوار پیشرفت و لغزنده ها	۴۸۳
استفاده از دکمه JQuery UI	۴۸۴
پیکربندی دکمه	۴۸۶
استفاده از آیکن های JQuery UI در دکمه ها	۴۸۸
استفاده از تصویر شخصی	۴۸۹
استفاده از متدهای دکمه	۴۹۰
حذف ویجت	۴۹۰
فعال و غیرفعال سازی دکمه	۴۹۱
بازآوری وضعیت دکمه JQuery UI	۴۹۲
استفاده از رویداد دکمه	۴۹۲
ایجاد انواع متفاوت دکمه	۴۹۳
ایجاد یک دکمه وضعیتی	۴۹۴
ایجاد یک مجموعه دکمه	۴۹۵
ساخت مجموعه دکمه از دکمه های عادی و منظم	۴۹۶
استفاده از نوار پیشرفت JQuery UI	۴۹۷
ایجاد نوار پیشرفت کار	۴۹۷
ایجاد نوار پیشرفت نامعین	۴۹۸
استفاده از متدهای نوار پیشرفت	۴۹۹
استفاده از رویدادهای نوار پیشرفت	۵۰۱
استفاده از لغزنده JQuery UI	۵۰۳
پیکربندی لغزنده	۵۰۴
تغییر جهت لغزنده	۵۰۴
انیمیشن سازی لغزنده ها	۵۰۶
ایجاد یک لغزنده بازه ای	۵۰۷
استفاده از متدهای لغزنده	۵۰۸
استفاده از رویدادهای لغزنده	۵۱۰
چکیده	۵۱۲

فصل ۲۲: استفاده از ویجت‌های تکمیل خودکار و آکوردین ۵۱۳

استفاده از تکمیل خودکار jQuery UI	۵۱۴
ایجاد عنصر تکمیل خودکار	۵۱۴
استفاده از شیء آرایه به عنوان منبع داده	۵۱۵
پیکربندی تکمیل خودکار	۵۱۶
استفاده از منبع دور	۵۱۷
استفاده از فانکشن به عنوان منبع داده	۵۲۰
مکان پاپ‌آپ	۵۲۲
استفاده از متدهای تکمیل خودکار	۵۲۴
استفاده از رویدادهای تکمیل خودکار	۵۲۷
گرفتن جزئیات آیتم انتخاب شده	۵۲۷
اصلاح نتایج جست‌وجو	۵۲۹
صرف نظر از عمل پیش فرض select	۵۲۹
استفاده از آکوردین jQuery UI	۵۳۱
ایجاد یک آکوردین	۵۳۱
پیکربندی آکوردین	۵۳۳
تنظیم ارتفاع آکوردین	۵۳۴
استفاده از پدر جهت تعیین ارتفاع آکوردین	۵۳۷
تغییر نوع رویداد	۵۳۸
انتخاب سرلیست فعال	۵۳۹
تغییر آیکن‌های آکوردین	۵۴۰
استفاده از متدهای آکوردین	۵۴۱
استفاده از رویدادهای آکوردین	۵۴۲

فصل ۲۳: استفاده از ویجت برگه‌ها ۵۴۵

ایجاد برگه	۵۴۶
گرفتن محتوای برگه با آژاکس	۵۴۸
پیکربندی ویجت برگه	۵۵۰
انتخاب برگه فعال	۵۵۰
غیرفعال کردن برگه‌های خاص	۵۵۲
تغییر رویداد	۵۵۳
استفاده از برگه‌های کشویی	۵۵۴
استفاده از متدهای برگه jQuery UI	۵۵۵
اضافه و حذف برگه‌ها	۵۵۶
فراخوانی درخواست‌های آژاکس برگه	۵۵۷
استفاده از رویدادهای برگه	۵۵۸
متوقف کردن درخواست‌های آژاکس	۵۵۹
اصلاح داده برگه دور	۵۶۰
استفاده از برگه‌ها برای نمایش فرم	۵۶۱

۵۶۳		اعمال برگه‌ها
۵۶۵		مدیریت کلیک دکمه‌ها
۵۶۵		اعمال اعتبارسنجی
۵۶۹		فصل ۲۴؛ استفاده از ویجت اخذ تاریخ
۵۷۰		ایجاد اخذ تاریخ
۵۷۱		ایجاد اخذ تاریخ درون خطی
۵۷۲		پیکربندی اخذ تاریخ
۵۷۳		تعیین تاریخ پیش فرض
۵۷۴		تعیین عنصر جایگزین
۵۷۵		مدیریت فراخوانی پاپ آپ
۵۷۷		مدیریت انتخاب تاریخ
۵۷۸		محدودیت در کارکترهای ورودی و بازه تاریخ
۵۸۰		ایجاد نمایش چند ماهه
۵۸۲		دسترسی مستقیم به ماه‌ها و سال‌ها
۵۸۳		مدیریت ظاهر اخذ تاریخ
۵۸۴		نمایش هفته‌ها
۵۸۵		مجوز سرریز شدن ماه‌ها
۵۸۶		استفاده از نوار دکمه
۵۸۶		آماده سازی قالب راهنما برای کاربر
۵۸۸		استفاده از متدهای اخذ تاریخ
۵۸۸		گرفتن و تنظیم بصورت برنامه نویسی
۵۹۰		ظاهر و مخفی سازی اخذ تاریخ‌های پاپ آپ بصورت برنامه نویسی
۵۹۱		استفاده از رویدادهای اخذ تاریخ
۵۹۱		پاسخ به تغییر ماه و سال
۵۹۳		پاسخ به بسته شدن پاپ آپ
۵۹۳		بومپسازی انتخاب تاریخ
۵۹۷		فصل ۲۵؛ استفاده از ویجت دیالوگ و دکمه چرخاندنی
۵۹۸		ایجاد دیالوگ
۶۰۰		پیکربندی دیالوگ
۶۰۱		پیکربندی ظاهر آغازین دیالوگ
۶۰۳		تنظیم محل دیالوگ
۶۰۵		افزودن دکمه به دیالوگ
۶۰۶		کشیدن و پشته سازی دیالوگ‌ها
۶۰۷		ایجاد دیالوگ‌های مودال
۶۰۹		نمایش یک فرم در دیالوگ مودال
۶۱۱		استفاده از متدهای دیالوگ
۶۱۲		استفاده از رویدادهای دیالوگ
۶۱۳		نگهداری دیالوگ باز

۶۱۴	پاسخ به تغییر اندازه‌ها و تغییر مکان‌ها
۶۱۵	استفاده از ویجت واسط کاربری jQuery دکمه چرخاندنی
۶۱۶	استفاده از ویجت دکمه چرخاندنی با انواع عنصر ورودی HTML5
۶۱۸	پیکربندی ویجت دکمه چرخاندنی
۶۱۹	پیکربندی رفتار پایه ویجت دکمه چرخاندنی
۶۲۰	تغییر آیکن‌های دکمه چرخاندنی
۶۲۱	کنترل نرخ تغییرات
۶۲۲	تنظیم قالب عدد
۶۲۳	استفاده از متدهای دکمه چرخاندنی
۶۲۵	استفاده از رویدادهای دکمه چرخاندنی
۶۲۷	فصل ۲۶: استفاده از ویجت‌های منو و راهنما
۶۲۸	ایجاد منو
۶۲۹	قالب‌بندی آیتم‌های منو
۶۳۰	انجام مرورهای پایه
۶۳۱	پیکربندی منو
۶۳۱	استفاده از ساختار عنصر متفاوت
۶۳۳	استفاده از آیکن‌ها در منوها
۶۳۵	جانمایی زیرمنوها
۶۳۶	استفاده از متدهای منو
۶۳۶	استفاده از رویدادهای منو
۶۳۹	استفاده از ویجت راهنمای آئی jQuery UI
۶۳۹	ایجاد راهنمای آئی
۶۴۰	استفاده راهنما با عناصر ورودی
۶۴۱	پیکربندی راهنمای آئی
۶۴۱	تنظیم محتوای راهنما
۶۴۳	تولید محتوای ویجت راهنما با یک فانکشن
۶۴۴	به‌دست‌آوردن محتوا از راه دور
۶۴۵	افزودن کلاس‌های CSS بیشتر به راهنمای آئی
۶۴۷	ردیابی ماوس
۶۴۸	جانمایی راهنمای آئی
۶۴۹	استفاده از متدهای راهنمای آئی
۶۵۱	استفاده از رویدادهای راهنمای آئی
۶۵۳	فصل ۲۷: استفاده از تعاملات کشیدن و رهاکردن
۶۵۴	ایجاد یک تعامل کشیدنی
۶۵۵	پیکربندی تعاملات کشیدنی
۶۵۶	محدودیت محور کشیدن
۶۵۷	محدودیت منطقه کشیدن
۶۵۸	محدودیت کشیدن در شبکه

۶۶۰	وقفه در کشیدن
۶۶۱	استفاده از متدهای کشیدنی
۶۶۱	استفاده از رویدادهای کشیدنی
۶۶۲	استفاده از تعامل رهاشدنی
۶۶۴	پررنگ کردن مقصد رهاشدن
۶۶۵	برخورد با همپوشانی عناصر
۶۶۶	پیکربندی تعامل رهاشدنی
۶۶۶	محدودیت عناصر رهاشدنی قابل قبول
۶۶۸	پررنگ کردن رهاشدنی با استفاده از کلاس‌ها
۶۶۹	تغییر میزان همپوشانی
۶۷۲	استفاده از متدهای رهاشدنی
۶۷۳	تنظیم کشیدن و رهاکردن
۶۷۳	استفاده از میدان دید عنصر
۶۷۵	استفاده از عنصر راهنما
۶۷۷	دستکاری عنصر راهنما
۶۷۸	برخورد با لبه‌های عناصر
۶۸۱	فصل ۲۸: استفاده از دیگر تعاملات
۶۸۲	استفاده از تعامل مرتب‌سازی
۶۸۳	گرفتن اطلاعات مرتب‌سازی
۶۸۵	پیکربندی تعامل مرتب‌شدنی
۶۸۵	اتصال تعاملات مرتب‌شدنی
۶۸۷	اتصال عنصر کشیدنی به عنصر مرتب‌شدنی
۶۸۸	انتخاب آیتم‌های مرتب‌شده
۶۸۹	سبک بندی فضای خالی
۶۹۰	استفاده از متدهای مرتب‌شدنی
۶۹۱	لغو عمل مرتب‌سازی
۶۹۲	بازآوری عناصر مرتب‌شدنی
۶۹۳	استفاده از رویدادهای مرتب‌شدنی
۶۹۴	استفاده از تعامل انتخاب
۶۹۶	پیکربندی تعامل انتخاب
۶۹۷	استفاده از متدهای تعامل انتخاب
۶۹۸	استفاده از رویدادهای تعامل انتخاب
۶۹۸	استفاده از تعامل تغییر اندازه
۷۰۰	پیکربندی تعامل تغییر اندازه
۷۰۱	تغییر اندازه عناصر مرتبط
۷۰۲	محدودیت در تغییر اندازه عناصر
۷۰۳	محل دسته کشیدنی
۷۰۵	فصل ۲۹: عملیاتی کردن دوباره مثال بخش ۳

۷۰۵	بازنگری سند مثال
۷۰۷	نمایش محصولات
۷۰۸	افزودن سبد فروش
۷۱۰	بسته بندی آکوردین
۷۱۰	افزودن جدول
۷۱۱	مدیریت تغییرات مقدار ورودی
۷۱۲	حذف ردیف‌ها
۷۱۳	اصلاح ردیف‌های موجود
۷۱۴	اعمال سبک تم
۷۱۵	اعمال فرم ورک CSS بطور وسیع
۷۱۶	اعمال گوشه‌های گرد روی جدول
۷۱۸	ایجاد یک دکمه jQuery UI
۷۲۱	افزودن دیالوگ
۷۲۴	مدیریت کلیک دکمه Place Order
۷۲۷	تکمیل درخواست
۷۲۹	فصل ۳۰؛ آغاز کار با JQUERY MOBILE
۷۲۹	تنظیم jQuery موبایل
۷۳۰	به‌دست‌آوردن jQuery موبایل
۷۳۰	ایجاد یک تم
۷۳۰	گرفتن jQuery
۷۳۰	نصب jQuery موبایل
۷۳۱	فهم روش jQuery موبایل
۷۳۲	فهم بهبود خودکار
۷۳۳	فهم viewport
۷۳۵	فهم رویدادهای jQuery موبایل
۷۳۵	فهم رویداد pageinit
۷۳۶	فهم رویدادهای لمسی
۷۳۹	استفاده از متدهای رفتاری jQuery موبایل
۷۴۱	استفاده از رویدادهای مجازی ماوس jQuery موبایل
۷۴۴	پاسخ به تغییرات چرخش دستگاه
۷۴۵	استفاده از پرسش‌های محاورهای برای مدیریت جهت
۷۴۶	کار با وسایل موبایل
۷۴۷	پرهیز از ۲ اشتباه اساسی در توسعه گوشی همراه
۷۴۷	پرهیز از فرضیات بد
۷۴۷	جهان iPhone نیست
۷۴۷	جهان همه، گوشی نیست
۷۴۷	جهان لمسی نیست
۷۴۸	پهنای باند موبایل آزاد و نامحدود نیست

۷۴۸	پرهیز از شبیه سازی و تست غیرواقعی
۷۴۹	عدم لمس
۷۴۹	عدم گرفتگی و انسداد
۷۴۹	عدم دقت
۷۵۰	استفاده از شبیه ساز مرورگر موبایل
۷۵۰	استفاده از شبیه ساز اندروید
۷۵۰	استفاده از شبیه ساز آیفون
۷۵۰	استفاده از شبیه ساز موبایل اپرا
۷۵۱	استفاده از شبیه ساز فایرفاکس
۷۵۱	استفاده از نمایشگر چند مرورگر
۷۵۲	استفاده از شبیه ساز گوشی ویندوز ۷
۷۵۲	استفاده از مرورگرهای رومیزی
۷۵۵	فصل ۳۱: صفحات، تم‌ها و آرایش‌ها
۷۵۶	فهم صفحات jQuery موبایل
۷۵۷	افزودن header و footer به صفحه
۷۵۸	افزودن صفحات به یک سند
۷۶۰	پیکربندی انتقال صفحه
۷۶۲	اتصال به صفحه بیرونی
۷۶۴	برخورد با مساله شناسه از/اکس/صفحه
۷۶۷	واکشی آغازین صفحات
۷۶۹	استفاده از اسکرپت در کنترل صفحات jQuery موبایل
۷۶۹	تغییر صفحه جاری
۷۷۱	کنترل جهت افکت انتقال
۷۷۲	کنترل دیالوگ لود
۷۷۴	تعیین صفحه جاری
۷۷۶	لود صفحات در پس زمینه
۷۷۷	استفاده از رویدادهای صفحه
۷۷۸	مدیریت رویداد مقداردهی صفحه
۷۷۸	مدیریت رویدادهای لود صفحه
۷۷۹	پاسخ به انتقال صفحه
۷۸۰	اعمال تم‌های jQuery موبایل
۷۸۲	اعمال نمونه‌ها روی عناصر ویژه
۷۸۳	ایجاد آرایش شبکه‌ای
۷۸۵	فصل ۳۲: ویجت دیالوگ و پاپ‌آپ
۷۸۵	ایجاد ویجت دیالوگ
۷۸۷	ایجاد یک دیالوگ به صورت برنامه نویسی
۷۸۹	افزودن دکمه به دیالوگ‌ها
۷۹۱	پیکربندی ویجت دیالوگ

استفاده از متدهای دیالوگ.....	۷۹۴
استفاده از رویدادهای دیالوگ.....	۷۹۵
استفاده از ویجت پاپ آپ jQuery موبایل.....	۷۹۶
ایجاد ویجت پاپ آپ.....	۷۹۶
پیکربندی ویجت پاپ آپ.....	۷۹۷
پیکربندی لینکی که پاپ آپ را باز می کند.....	۷۹۷
پیکربندی پاپ آپ بطور مستقیم.....	۷۹۹
استفاده از تنظیم تاریخچه.....	۷۹۹
استفاده از پاپ آپ در نمایش متن پرمحتوا.....	۸۰۰
استفاده از متدهای پاپ آپ.....	۸۰۳
استفاده از رویدادهای پاپ آپ.....	۸۰۵
فصل ۳۳: دکمه ها و بلوک های کشویی	۸۰۷
استفاده از دکمه های jQuery موبایل.....	۸۰۸
فهم ایجاد دکمه ها به صورت اتوماتیک.....	۸۰۸
ایجاد دکمه ها از عناصر دیگر.....	۸۰۹
ایجاد دکمه های گروهی.....	۸۱۰
پیکربندی دکمه های jQuery موبایل.....	۸۱۲
افزودن آیکن ها به دکمه.....	۸۱۲
ایجاد دکمه های درون خطی و کوچک.....	۸۱۴
استفاده از متدهای دکمه.....	۸۱۵
استفاده از رویدادهای دکمه.....	۸۱۷
استفاده از نوار پیمایش jQuery موبایل.....	۸۱۷
پیکربندی نوار پیمایش.....	۸۱۹
رویدادها و متدهای نوار پیمایش.....	۸۲۰
استفاده از بلوک های محتوا با خاصیت کشویی.....	۸۲۱
ایجاد بلوک کشویی ساده.....	۸۲۱
پیکربندی بلوک های کشویی jQuery موبایل.....	۸۲۲
استفاده از متدهای بلوک کشویی.....	۸۲۴
استفاده از رویدادهای بلوک کشویی.....	۸۲۴
ایجاد آکوردین jQuery موبایل.....	۸۲۵
پیکربندی آکوردین.....	۸۲۷
استفاده از متدهای آکوردین.....	۸۲۷
استفاده از رویدادهای آکوردین.....	۸۲۷
فصل ۳۴: استفاده از فرم های JQUERY موبایل	۸۲۹
ایجاد ویجت های عنصر فرم.....	۸۳۱
استفاده از ویجت Textinput.....	۸۳۳
پیکربندی ویجت Textinput.....	۸۳۳
استفاده از متدهای ویجت textinput.....	۸۳۴

۸۳۵	استفاده از رویدادهای ویجت <code>textInput</code>
۸۳۵	استفاده از ویجت لغزنده و لغزنده محدود.....
۸۳۷	پیکربندی ویجت‌های لغزنده و لغزنده محدود.....
۸۳۸	استفاده از متدهای ویجت لغزنده و لغزنده محدود.....
۸۴۰	استفاده از رویدادهای لغزنده.....
۸۴۱	استفاده از رویدادهای لغزنده محدود.....
۸۴۲	استفاده از ویجت <code>Selectmenu</code>
۸۴۳	پیکربندی ویجت <code>Selectmenu</code>
۸۴۴	پیکربندی دکمه <code>selectmenu</code>
۸۴۵	پیکربندی پاپ‌آپ <code>selectmenu</code>
۸۴۶	تعیین عنصر جای خالی.....
۸۴۶	استفاده از متدهای <code>selectmenu</code>
۸۴۸	استفاده از رویدادهای <code>selectmenu</code>
۸۴۸	ایجاد سوئیچ‌های ضربه‌ای.....
۸۴۹	استفاده از ویجت چک باکس رادیو.....
۸۵۰	ایجاد چک باکس‌ها.....
۸۵۱	اعمال یک برچسب به چک باکس.....
۸۵۲	گروه بندی چک باکس‌ها.....
۸۵۴	ایجاد دکمه‌های رادیویی.....
۸۵۵	پیکربندی ویجت <code>Checkboxradio</code>
۸۵۵	استفاده از متدهای <code>Checkboxradio</code>
۸۵۶	استفاده از رویدادهای <code>Checkboxradio</code>
۸۵۷	فصل ۳۵: استفاده از لیست‌ها و پنل‌ها.....
۸۵۸	آغاز کار با لیست‌ها.....
۸۶۰	پیکربندی ویجت لیست.....
۸۶۱	ایجاد لیست‌های ریزشی.....
۸۶۲	ایجاد لیست‌های چندبخشی.....
۸۶۴	فیلترینگ لیست‌ها.....
۸۶۷	استفاده از فانکشن فیلترینگ شخصی.....
۸۶۸	افزودن جداکننده‌ها.....
۸۶۹	استفاده از پیکربندی مبتنی بر قرارداد.....
۸۶۹	افزودن حباب‌های شمارشی.....
۸۷۰	افزودن تأکیدات متنی.....
۸۷۱	افزودن فضای جدا.....
۸۷۲	استفاده از متدهای لیست.....
۸۷۲	استفاده از رویدادهای لیست.....
۸۷۲	استفاده از ویجت پنل.....
۸۷۳	پیکربندی ویجت پنل.....

۸۷۴		جانمایی و نمایش پتل
۸۷۶		کنارزدن پتل
۸۷۷		استفاده از متدهای پتل
۸۷۸		استفاده از رویدادهای پتل
۸۷۹		فصل ۳۶؛ عملیاتی کردن دوباره مثال بخش ۴
۸۷۹		آغاز با اصول
۸۸۱		درج محصولات به صورت برنامه نویسی
۸۸۵		استفاده دوباره از صفحات
۸۸۸		ایجاد سید فروش
۸۹۱		افزودن برای تغییر مقدار
۸۹۵		افزودن دکمه به صفحه اطلاعات
۸۹۷		اجرای فرایند پرداخت

فصل ۲۰

تنظیم JQuery UI

دانلود و نصب JQuery UI کمی پیچیده‌تر از دیگر کتابخانه‌های جاوااسکریپت است اما کار سنگینی نیست و نیاز به برخی جزئیات دارد که موضوع همین فصل است. برای این کتاب کافیهست تا تنظیمات را یاد بگیرید اما برای استقرار مناسب، جزئیات نصب فایلها را نیز بیان می‌کنم. همچنین به جزئیات استفاده از JQuery UI در داخل شبکه توزیع محتوای (CDN) می‌پردازیم.

بدست‌آوردن JQuery UI

فرایند دانلود JQuery UI کمی پیچیده‌تر از کتابخانه‌های جاوااسکریپت است اما ارزش آن را دارد. ۵ قابلیت اصلی در JQuery UI وجود دارد که وقتی به شکل اختصاصی دانلود می‌کنید شامل این قابلیتها و تنظیماتی است که روی هر کدام می‌توانید انجام دهید. در این بخش از کتاب این ویژگیها را نشان می‌دهم اما برای برنامه‌های واقعی می‌توانید بخشی از آن را استفاده نمایید و کتابخانه کوچکتری برای خود بسازید.

نکته: JQuery UI تنها ابزار واسط کاربری برای JQuery نیست هرچند خیلی محبوب است. البته منبع باز است و به رایگان قابل دانلود است: <http://flowplayer.org/tools>. گزینه‌های دیگری نیز وجود دارد مانند jQWidgets (www.jqwidgets.com) و Wijmo (<http://wijmo.com>). البته Mobile JQuery نیز وجود دارد که در بخش بعدی توضیح خواهیم داد.

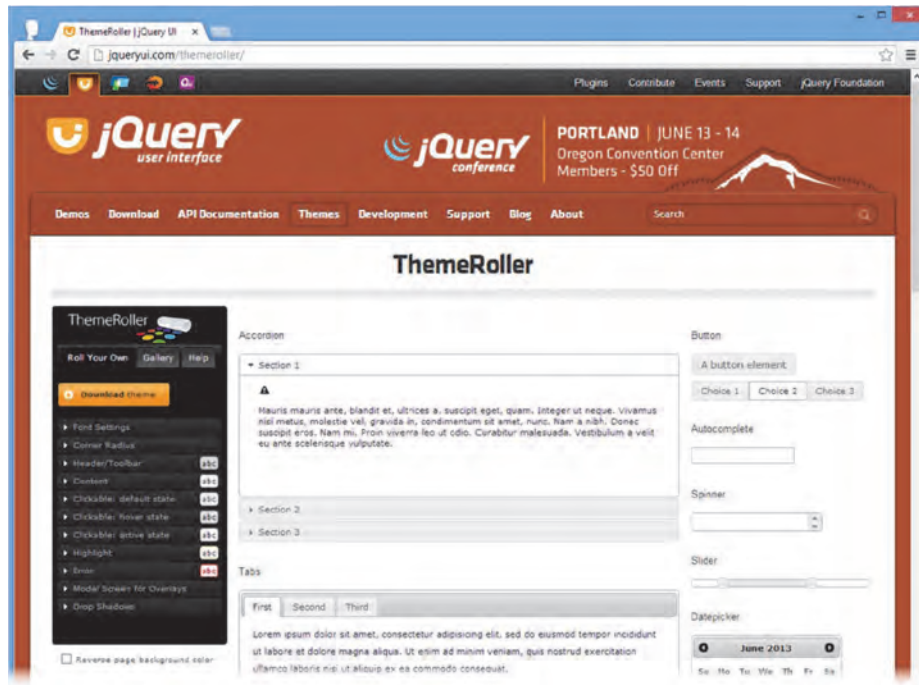
تصمیم‌گیری یک تم^۱

پیش از اینکه کتابخانه شخصی JQuery UI خود را بسازید نیاز است تا درباره‌ی انتخاب یک تم تصمیم بگیرید. در واقع می‌توانید ظاهر بسیاری از عناصر سندتان را تغییر دهید. بسیاری از اشیا وجود دارد که تغییر ظاهر دستی آنها کاری بسیار سخت و طاقت فرساست. یک گالری از تمهای تعریف شده وجود دارد که بسادگی می‌توانید از آنها استفاده کنید. برای آشنایی با هر یک از این تمها به سایت <http://jqueryui.com> مراجعه و روی دکمه Themes کلیک نمایید. صفحه‌ی ThemeRoller را بیاورید که شامل ویجت‌های JQuery UI است. یک پنل سمت چپ وجود دارد که می‌توانید تم را پیکربندی کنید.

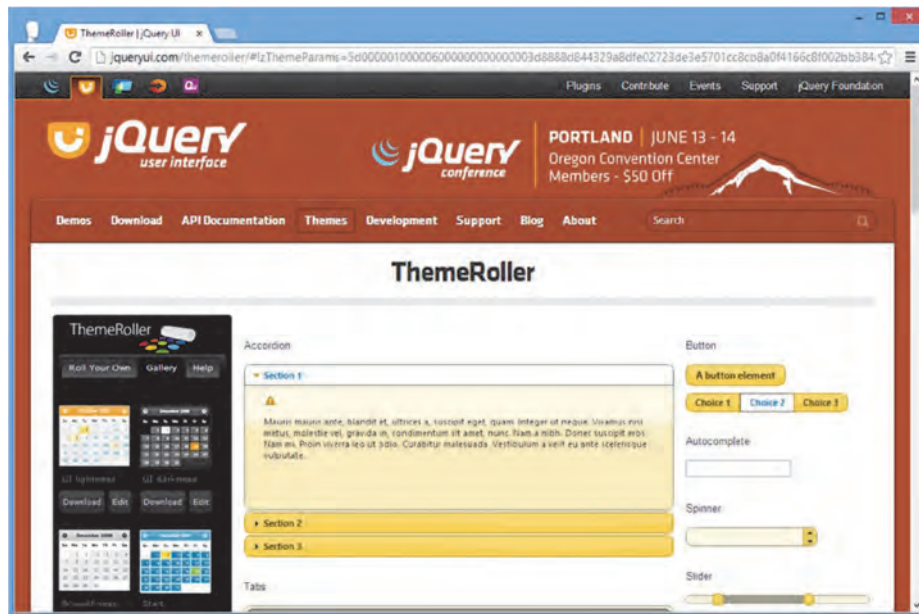
چنانچه سبک نمایشی منتخب خود را در برنامه کاربردی خود بخواهید تغییر دهید می‌توانید این کار را در برگه‌ی Roll Your Own انجام دهید. هر جنبه‌ای از CSS را که توسط JQuery UI استفاده شده می‌توان تغییر داد.

تمهای از پیش تعریف شده روی برگه‌ی Gallery در دسترس هستند. ۲۴ تم در گالری وجود دارد. اگر روی هر تم کلیک نمایید ویجت‌ها اصلاح می‌شوند تا نمایی از برنامه کاربردی شما را نشان دهد مانند شکل ۲-۲۰.

^۱ Theme



شکل ۲۰-۱: صفحه تم وب سایت jQuery UI



شکل ۲۰-۲: گالری تم sunny

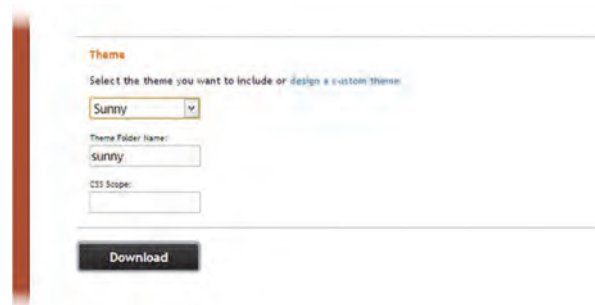
تم پیش فرض UI lightness نام دارد اما روشنایی کافی برای نمایش دادن و چاپ صفحات ندارد. بنابراین از تم sunny استفاده می‌کنم که در نمایش کمی بهتر است. در هر کاری نیاز به استفاده از تم ندارید و کافی است نام آنرا بدانید. تمها در هنگام چاپ خوب نیستند اما ظاهر بسیار زیبایی دارند و پیشنهاد می‌کنم جست‌وجو نمایید و تم خوبی را انتخاب نمایید.

دانلود jQuery UI شخصی

اینک که برای خود تم انتخاب کردید می‌توانید آنرا برای خود دانلود نمایید. دکمه‌ی Download را کلیک نمایید تا به سمت صفحه‌ی دانلود حرکت نمایید. این صفحه لیستی از ابزارهای jQuery UI را دارد که در گروه‌های کاربردی تقسیم شده‌اند؛ هسته UI، تعاملی‌ها، ویجت‌ها و افکت‌ها. تنها مواردی را نیاز دارید انتخاب نمایید تا مجموعه کوچتری از فایل‌ها برای مرورگرتان دانلود شود و این ایده خوبی است زیرا به نظر من باید به فکر کاهش پهنای باند نیز بود که در ادامه شرح خواهیم داد.

نکته: برخی از اجزا وابسته به دیگری است. نگران نباشید. در هنگام ساخت کتابخانه‌ی شخصی jQuery UI با انتخاب یک جزء، سایر اجزا مورد نیاز انتخاب و دانلود می‌شوند.

گام بعدی انتخاب تمی است که می‌خواهید. آنرا از انتخابگر انتخاب نمایید.



شکل ۳-۲۰: انتخاب تم

نکته: نگارش jQuery UI را نیز می‌توانید انتخاب کنید. در این فصل به نسخه‌ی stable نیاز دارید تا با همه‌ی نگارشهای jQuery و پس از 1.3.2 کار کند.

وقتی اجزا را انتخاب نمودید، تم مورد خواسته‌تان و نگارش آن انتخاب می‌شود. دکمه‌ی دانلود را کلیک کنید تا کتابخانه jQuery UI شخصی ساخته و دانلود شود.

نصب jQuery UI برای توسعه

jQuery UI دانلود شده شامل همه‌ی فایل‌های مورد نیاز برای توسعه و ساخت برنامه هست. برای این کتاب از فایل‌های توسعه که شامل کد منبع غیرفشرده است استفاده می‌کنید. این برای جست‌وجو داخل jQuery UI آسانتر است. کافی است موارد زیر را داخل شاخه‌ی مثال خود کپی نمایید:

- فایل js\jquery-ui-1.10.3.custom.js
- فایل themes\sunny\jquery-ui-1.10.3.custom.css

• شاخه‌ی themes\sunny\images

توجه داشته باشید که اینها فایل‌های CSS و جاوااسکریپت برای اجزا و ویژگی‌های خاص هستند که در شاخه‌های ui، themes قرار دارند. نیازی به استفاده از آنها ندارید اما هنگام کار با یکسری ویژگی‌های محدود jQuery UI می‌تواند مفید باشد.

نکته: نام فایل‌های جاوااسکریپت و CSSها شامل شماره نگارش نسخه‌ی دانلودی است. نگارش مثال فوق 1.10.3 می‌باشد؛ ممکن است شما نگارش دیگری را دانلود کرده باشید.

افزودن jQuery UI به سند HTML

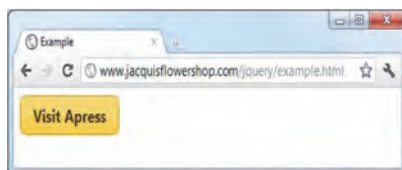
تنها چیزی که باقی مانده این است که jQuery UI را به سند HTML پیوند دهید. با استفاده از عناصر script و link می‌توانید فایل‌های فوق را مانند لیست ۱-۲۰ به سند بیفزایید. لازم نیست به شاخه‌ی Images ارجاع مستقیمی داشته باشید. شاخه‌ی Images و CSS در یک مکان قرار دارند و jQuery UI می‌تواند منابع خود را پیدا کند.

لیست ۱-۲۰: افزودن jQuery UI به سند

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
    <script src="jquery-2.0.2.js" type="text/javascript"></script>
    <script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="styles.css"/>
    <link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
    <script type="text/javascript">
      $(document).ready(function() {
        $('a').button();
      });
    </script>
  </head>
  <body>
    <a href="http://apress.com">Visit Apress</a>
  </body>
</html>
```

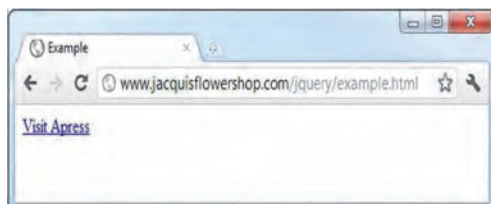
نکته: jQuery UI وابسته به jQuery است و بایستی jQuery را به سند افزوده باشید. jQuery UI یک کتابخانه مستقل نیست.

سند نشان داده شده در لیست شامل یک تست ساده است که تست می‌کند آیا jQuery UI بدرستی به سند افزوده شده است. با دیدن سند یک دکمه متمایز می‌بینید؛ همانطور که در شکل ۴-۲۰ مشخص است. نگران فراخوانی متد button نباشید اینکه چه اتفاقی می‌افتد موضوع بحث فصل ۲۱ است.



شکل ۴-۲۰: تست افزوده شدن jQuery UI به سند

اگر به درستی این کار انجام نشده باشد دکمه را به شکل زیر خواهید دید.



شکل ۵-۲۰: خطا در افزودن jQuery UI به سند

نصب jQuery UI برای انتشار

وقتی توسعه برنامه کاربردی وب را تمام کردید و برای انتشار آماده شد می‌توانید از فایل‌های کمینه که دانلود کرده‌اید استفاده کنید. این فایل‌ها کوچکتر هستند اما برای اشکال زدایی و خواندن سخت‌تر. برای استفاده از این فایل‌ها موارد زیر را روی سرور خود کپی نمایید.

- فایل js\jquery-ui-1.10.3.custom.min.js
- فایل themes\sunny\jquery-ui.1.10.3.custom.css
- شاخه‌ی images\themes\sunny

شاخه‌ی تصویر و فایل CSS در نگارش توسعه، در مکان یکسان کپی می‌شوند. این فایل‌ها را تنها کپی نمایید.

استفاده از jQuery UI از طریق شبکه توزیع محتوا^۱

در فصل ۵ استفاده از CDN را لمس کردید. اگر این رویکرد شما باشد خوشحال باشید زیرا در همین راستا با jQuery UI نیز می‌توانید برخورد کنید. هردوی گوگل و مایکروسافت برای فایل‌های jQuery UI میزبانی می‌کنند. برای مثال من از سرویس‌های مایکروسافت استفاده کردم زیرا میزبان تم‌های استاندارد است.

برای استفاده از CDN تنها نیاز به URL دارید که این فایل‌ها را می‌خواهید. برای سرویس مایکروسافت به آدرس www.asp.net/ajaxlibrary/cdn.ashx مراجعه کنید. اگر به پایین صفحه بروید یک لینک برای نسخه‌های jQuery UI می‌بینید که بر اساس نگارش تقسیم‌بندی شده‌اند. بر روی نگارشی که می‌خواهید کلیک نمایید. مسیر فایل کتابخانه jQuery UI را بصورت کمینه خواهید دید. URL برای فایل کمینه به شکل زیر است:

<http://ajax.aspnetcdn.com/ajax/jquery.ui/1.10.3/jquery-ui.min.js>.

^۱ Content Distribution Network

بقیه صفحه هر یک از تمهای پیش فرض را نشان می‌دهد. URL تم sunny به شکل زیر است:

<http://ajax.aspnetcdn.com/ajax/jquery.ui/1.10.3/themes/sunny/jquery-ui.css>

برای استفاده از این فایلها روی CDN کافی است به سادگی URL آنها را در عناصر script و link قرار دهید مانند لیست ۲۰-۲.

لیست ۲۰-۲: استفاده از jQuery UI از طریق CDN

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
    <script src="jquery-2.0.2.js" type="text/javascript"></script>
    <script src=http://ajax.aspnetcdn.com/ajax/jquery.ui/1.10.3/jquery-ui.min.js
      type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="styles.css"/>
    <link rel="stylesheet" type="text/css"
      href=http://ajax.aspnetcdn.com/ajax/jquery.ui/1.10.3/themes/sunny/jquery-ui.css/>
    <script type="text/javascript">
      $(document).ready(function() {
        $('a').button();
      });
    </script>
  </head>
  <body>
    <a href="http://apress.com">Visit Apress</a>
  </body>
</html>
```

یکبار دیگر می‌توانید تست کنید آیا بطور صحیح URLها را به سند وارد کرده‌اید و یا اینکه شکل ۴-۲۰ را می‌بینید.

چکیده

در این فصل مراحلی که نیاز بود تا jQuery UI را دانلود کنید نشان دادیم. در این ویژگیها انعطاف زیادی وجود دارد. شخصا قابلیت ThemeRoller را دوست دارم. روش خوبی است برای اینکه تمی را انتخاب کنید. در فصل بعدی در بین ویژگیهای متفاوت jQuery UI جست‌وجو خواهید کرد. کار را با محبوبترین آنها یعنی ویجتها آغاز می‌کنید.

فصل ۲۱

استفاده از ویجت‌های دکمه، نوار پیشرفت و لغزنده‌ها

اینک که jQuery UI را دانلود و پیکربندی کردید، می‌توانید درون ابزارها جست‌وجو کنید. بلوک‌های کاربردی اصلی jQuery UI وجود دارد که به ویجت‌های jQuery UI معروفند؛ اگرچه ویژگی‌های دیگری نیز وجود دارد (مانند افکتها).

در این فصل ۳ مورد از ساده‌ترین ویجت‌ها را شرح می‌دهیم. دکمه، نوار پیشرفت و لغزنده‌ها. همه‌ی ویجت‌ها خصوصیات مشترکی دارند؛ تنظیمات، متدها و رویدادها. یک سری اصولی وجود دارد که درباره‌ی همه‌ی ویجت‌ها مشترک است و در آغاز فصل به آن می‌پردازیم.

خیلی سخت است که همه‌ی ویجت‌ها را درون مثال گل‌فروشی بیاورم. بنابراین مثالهایی را خواهید دید که تنها ویجت‌ها را نشان می‌دهد. در فصل ۲۹ دوباره به مثال گل‌فروشی برمی‌گردید و دوباره از آغاز می‌سازید. جدول ۱-۲۱ چکیده‌ی فصل را شرح می‌دهد.

جدول ۱-۲۱: چکیده‌ی فصل

مساله	راه حل	لیست
ایجاد یک دکمه jQuery UI	انتخاب یک عنصر و استفاده از متد button	۱
پیکربندی یک عنصر دکمه	ارسال شیء map به متد button با استفاده از متد option	۲ و ۳
استفاده از آیکنها در دکمه‌های jQuery UI	استفاده از تنظیم icons	۴
استفاده از تصویر در دکمه jQuery UI	تنظیم محتوای دکمه به شکل img	۵
حذف ویجت دکمه jQuery UI	استفاده از متد destroy	۶
فعال و غیرفعال کردن دکمه jQuery UI	استفاده از متد enable یا disable	۷
بازآوری وضعیت دکمه jQuery UI با توجه به تغییرات برنامه‌نویسی روی اصل عنصر	استفاده از متد refresh	۸
جواب به دکمه jQuery UI در هنگام ایجاد	تعیین یک فانکشن برای رویداد create	۹

مساله	راه حل	لیست
ایجاد دکمه‌های هم شکل از انواع مختلف عناصر	ایجاد دکمه‌های jQuery UI از button، input و عنصر a	۱۰
ایجاد یک دکمه وضعیتی	ایجاد یک jQuery UI از چک باکس	۱۱
ایجاد یک مجموعه دکمه	استفاده از متد buttonSet	۱۲ و ۱۳
ایجاد نوارپیشرفت jQuery UI	استفاده از متد progressBar	۱۴ و ۱۵
گرفتن یا مقداردهی پیشرفت به کاربر	استفاده از متد value	۱۶
جواب به تغییرات در نوار پیشرفت	تعیین فانکشن برای رویداد create، change، complete	۱۷
ایجاد یک لغزنده jQuery UI	استفاده از متد slider	۱۸
تغییر جهت یک لغزنده jQuery UI	استفاده از تنظیم orientation	۱۹ و ۲۰
انیمیشن‌سازی حرکت دسته وقتی کاربر کلیک می‌کند	استفاده از تنظیم animate	۲۱
ایجاد یک لغزنده jQuery UI که به کاربر امکان تعیین بازه را بدهد	استفاده از تنظیم value، range	۲۲
کنترل یک لغزنده jQuery UI بطور برنامه‌نویسی	استفاده از متدهای value، values	۲۳
جواب به تغییرات در مکانهای دسته لغزنده	مدیریت رویداد start، stop، change، slide	۲۴

استفاده از دکمه jQuery UI

نخستین ویجت مورد بحث فضای خوبی از jQuery UI را معرفی می‌کند. ویجت دکمه ارتباط ساده‌ای دارد ولی یک تاثیر متحول کننده در اسناد HTML دارد. ویجت دکمه تم jQuery UI را روی عناصر button و a اعمال می‌کند. این بدین معنی است که وقتی یک تم jQuery UI شخصی را ایجاد می‌کنید، اندازه، شکل، فونت و رنگ عنصر با تم انتخاب شده منطبق می‌گردد. اعمال ویجت‌های jQuery UI بسیار ساده است.

لیست ۱-۲۱: یک سند HTML ساده

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="handlebars.js"></script>
<script src="handlebars-jquery.js"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
```

```

<link rel="stylesheet" type="text/css" href="styles.css"/>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script id="flowerTpl" type="text/x-handlebars-template">
    {{#flowers}}
        <div class="dcell">
            
            <label for="{{product}}">{{name}}</label>
            <input name="{{product}}" data-price="{{price}}" data-stock="{{stocklevel}}"
            value="0" required />
        </div>
    {{/flowers}}
</script>
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json",{
            success: function(data) {
                var tmplData = $("#flowerTpl")
                    .template({ flowers: data }).filter("*");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('#button').button();
    });
</script>
</head>
<body>
    <h1>Jacqui's Flower Shop</h1>
    <form method="post" action="http://node.jacquisflowershop.com/order">
        <div id="oblock">
            <div class="dtable">
                <div id="row1" class="drow">
                </div>
                <div id="row2" class="drow">
                </div>
            </div>
            <div id="buttonDiv"><button type="submit">Place Order</button></div>
        </form>
</body>
</html>

```

برای اعمال ویجت دکمه از jQuery برای انتخاب عنصری که می‌خواهید به این شکل درآید، استفاده می‌کنید. سپس متد button را فراخوانی می‌کنید. jQuery UI با دقت تمام کارها را انجام می‌دهد. اثر آنرا در شکل ۲۱-۱ می‌توانید ببینید. توجه کنید که متد button را روی شیء منتخب jQuery اعمال کردید.



شکل ۱-۲۱: اعمال ویجت دکمه

مانند همه‌ی ویجت‌های jQuery UI دکمه‌ی نشان داده‌شده در شکل، در واقع یک سری CSS اعمال شده روی عناصر HTML است. متد button عنصر را از حالت:

```
<button type="submit">Place Order</button>
```

به حالت زیر منتقل می‌کند:

```
<button type="submit" class="ui-button ui-widget ui-state-default ui-corner-all
    ui-button-text-only" role="button" aria-disabled="false">
    <span class="ui-button-text">Place Your Order</span>
</button>
```

این روش خوبی است زیرا به شما اجازه‌ی کار طبیعی با عناصر HTML را می‌دهد بدون اینکه در خصوص اعمال ویجت روی آنها نگران باشید.

پیکربندی دکمه

ویجت دکمه jQuery UI می‌تواند از طریق ویژگی‌های تنظیمات پیکربندی شود. این ویژگی‌ها در جدول ۲-۲۱ آمده- است.

جدول ۲-۲۱: ویژگی‌های تنظیم برای ویجت دکمه

ویژگی	شرح
disabled	گرفتن یا مقداردهی وضعیت غیرفعال دکمه. مقدار true نشانگر غیرفعال بودن دکمه است. jQuery UI به وضعیت اصل عنصر HTML کاری ندارد.
text	گرفتن یا مقداردهی اینکه آیا دکمه متن را نمایش دهد یا خیر. اگر مقدار icons false باشد از این تنظیم چشم‌پوشی می‌شود.
icons	گرفتن یا مقداردهی اینکه آیا دکمه icon را نمایش دهد یا خیر.
label	گرفتن یا مقداردهی متنی که توسط دکمه نمایش داده می‌شود.

این تنظیمات را می‌توانید به دو شکل انجام دهید. نخست در هنگام فراخوانی متدهای دکمه از شیء map استفاده کنید؛ مانند لیست ۲-۲۱.

لیست ۲-۲۱: پیکربندی ویجت دکمه با استفاده از شیء *map*

```
...
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json", {
            success: function(data, status) {
                var tmplData = $("#flowerTpl")
                    .template({ flowers: data }).filter("");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('button').button({
            label: "Place Your Order",
            disabled: true
        });
        $('button').button("option", "disabled", false);
    });
</script>
...
```

من با تنظیم label متن دکمه را مشخص کرده و با استفاده از تنظیم disabled دکمه را غیرفعال می‌کنم. از این روش در هنگام مقداردهی آغازین ویجت استفاده می‌شود و در واقع، ادامه بیشتر پیکربندیهای درخواست آژاکس است که پیش‌تر دیدید. لیست ۲-۲۱ همچنین تکنیکی را نشان می‌دهد که می‌توانید مقدار جدیدی را برای ویژگی در نظر بگیرید:

```
$('button').button("option", "disabled", false);
```

متد button را دوباره فراخوانی کردید اما این بار با ۳ آرگومان. نخستین آرگومان option است. دومین تنظیمی است که می‌خواهید تغییر دهید و سومین آرگومان مقدار جدید تنظیم است. این دستور مقدار false را برای تنظیم disabled در نظر می‌گیرد و مقدار پیشین آنرا هر چه هست تغییر می‌دهد.

می‌توانید این تکنیکها را ترکیب کنید بطوریکه متد button را با نخستین آرگومان option و دومین آرگومان شیء map فراخوانی کنید که به شما اجازه‌ی اعمال چندین تنظیم را می‌دهد.

لیست ۳-۲۱: استفاده از آرگومان *option* با یک شیء *map*

```
...
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json", {
            success: function(data, status) {
                var tmplData = $("#flowerTpl")
                    .template({ flowers: data }).filter("");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('button').button(
            $('button').button("option", {
```

```

        label: "Place Your Order",
        disabled: false
    });
    console.log("Enabled: " + $('button').button("option", "disabled"));
});
</script>
...

```

به همین شکل می‌توانید مقدار تنظیم را بخوانید. در این مورد، متد button را با دو آرگومان فراخوانی می‌کنید. نخست option و دوم تنظیماتی است که می‌خواهید مقدار آنرا بخوانید. به صورت زیر:

```
console.log("Enabled? " + $('button').button("option", "disabled"));
```

این دستور مقدار تنظیم disabled را در خروجی می‌نویسد:

Enabled: false

استفاده از آیکنهای jQuery UI دکمه‌ها

تمهای jQuery UI شامل یک مجموعه آیکن است که می‌توانید برای هر منظوری استفاده نمایید. لیست ۴-۲۱ استفاده از آیکن در دکمه jQuery UI را نشان می‌دهد.

لیست ۴-۲۱: نمایش یک آیکن در یک دکمه

```

...
$('button').button({
    icons: {
        primary: "ui-icon-star",
        secondary: "ui-icon-circle-arrow-e"
    }
});
...

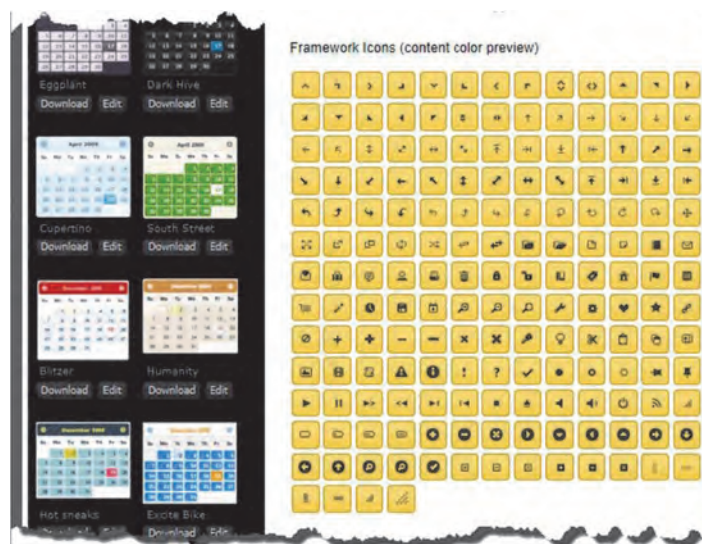
```

از تنظیم icons برای تعیین آیکن مورد نظر استفاده می‌شود. ویجت دکمه دو محل برای آیکن دارد. آیکن اصلی در سمت چپ متن و آیکن دوم در سمت راست متن. همانطور که لیست توضیح می‌دهد از یک شیء map برای ویژگیهای primary, secondary استفاده می‌کنید. می‌توانید یکی از ویژگیها را از قلم بیاندازید تا تنها یک آیکن را نمایش دهد. آیکنها کوچک هستند.



شکل ۲-۲۱: نمایش آیکنها در دکمه

آیکنها را با استفاده از فایل css مربوط به jQuery UI تعیین می‌کنید. ۱۷۳ آیکن متفاوت آماده هستند. آسانترین راه برای یافتن نام آیکنها مراجعه به سایت <http://jqueryui.com> است. صفحه تم را انتخاب و به پایین صفحه بروید. آیکنها لیست شده‌اند و با رفتن ماوس روی هر کدام نام آنها را می‌توانید ببینید.



شکل ۳-۲۱: لیست آیکن jQuery UI

نکته: نامی که روی صفحه وب نمایان می‌شود تنها پیشوندی است که در هنگام استفاده باید مورد توجه قرار گیرد. برای مثال اگر روی نخستین آیکن بروید ui-icon-caret-1-n را خواهید دید که برای استفاده از آن در دکمه باید ویژگی primary یا secondary را به ui-icon-caret-1-n مقدار دهید.

استفاده از تصویر شخصی

همه‌ی آیکنهای jQuery UI مفید نیستند. آنها معمولاً سائز کوچکی دارند. می‌توانید از تکنیکهایی که مقادیر متفاوتی را نمایش می‌دهند استفاده نمایید. نخست درج، یک عنصر img بر روی عنصر button است که قصد اعمال jQuery UI روی آن را دارید. دکمه jQuery UI راه خوبی برای حفظ محتوای اصلی عنصر button است و تصویر پس زمینه را به خوبی منتقل می‌کند. نگران نحوه ساخت آنها و تطبیق با تم خود نباشید. لیست ۵-۲۱ مثال خوبی را ارائه کرده است.

لیست ۵-۲۱: استفاده از تصویر شخصی با یک دکمه jQuery UI

```
...
$('button')
    .text("")
    .append("<img src=rightarrows.png width=100 height=30 />")
    .button();
...
```

از تنظیم text بدون مقدار استفاده کنید تا دکمه jQuery UI بدون متن باشد و البته تنظیم icons را با true مقداردهی نمایید. اگر متن را نخواسته باشید و از آیکن jQuery UI هم نخواهید استفاده کنید می‌توانید از متد text مربوط به jQuery استفاده کنید تا محتوا را بصورت رشته خالی تنظیم نمایید. از متد append برای درج عنصر img استفاده کنید و متد button را برای ایجاد یک دکمه jQuery UI فراخوانی کنید. نتیجه را در شکل ۴-۲۱ ببینید.



شکل ۴-۲۱: نمایش یک تصویر شخصی در دکمه

استفاده از متدهای دکمه

ویجت‌های jQuery UI متدها را طوری تعریف می‌کند که بتوان با آن کنترل ویجت را بدست گرفت. اینها متدهای متفاوتی نیستند و تنها آرگومانهای مشخصی را به متد button ارسال می‌کنند. جدول ۳-۲۱ متدها را شرح می‌دهد.

جدول ۳-۲۱: متدهای دکمه

ویژگی	شرح
button("destroy")	عنصر HTML را به وضعیت آغازین خود برمی‌گرداند
button("disable")	غیرفعال کردن دکمه
button("enable")	فعال کردن دکمه
button("option")	تنظیم یک یا چندین گزینه. بخش پیکربندی دکمه را ببینید
button("refresh")	بازآوری دکمه. بخش بازآوری وضعیت دکمه jQuery UI را ببینید

حذف ویجت

متد destroy ویجت jQuery UI را از عنصر HTML حذف می‌کند و آنرا به وضعیت اصلی برمی‌گرداند.

لیست ۶-۲۱: استفاده از متد destroy

```

...
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json", {
            success: function(data, status) {
                var tmplData = $("#flowerTmpl")
                    .template({ flowers: data }).filter("*");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('button').button().click(function(e) {
            $('button').button("destroy");
            e.preventDefault();
        });
    });
</script>
...

```

در این مثال من از متد click برای ثبت یک فانکشن مدیر رویداد برای دکمه استفاده کردم. توجه کنید که همان کاری که در فصل ۹ انجام می‌دادید، انجام می‌دهید و هیچ کار بیشتری برای jQuery UI لازم نیست. متد destroy را در فانکشن فراخوانی کردم. بدین معنی که کلیک دکمه باعث حذف ویجت خودش می‌شود. نتیجه‌ی کار را در شکل ۲۱-۵ ببینید.



شکل ۲۱-۵: از بین بردن ویجت دکمه jQuery UI

فعال و غیرفعال سازی دکمه

متدهای enable, disable اجازه‌ی تغییر وضعیت دکمه jQuery UI را می‌دهد.

لیست ۲۱-۷: فعال و غیرفعال سازی دکمه

```
...
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json", {
            success: function(data, status) {
                var tmplData = $("#flowerTmpl")
                    .template({ flowers: data }).filter("*");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('<span>Enabled:<span><input type="checkbox" checked="" />').prependTo('#buttonDiv');
        $(':checkbox').change(function(e) {
            $('button').button({
                'checked': $.checked.length == 1 ? "enable" : "disable"
            });
        });
        $('button').button();
    });
</script>
...
```

در این اسکرپیت یک چک باکس داخل سند ایجاد کرده و از متد change برای ثبت فانکشنی که در هنگام انتخاب شدن یا نشدن فراخوانی می‌شود استفاده کردم. نتیجه را در شکل ۲۱-۶ می‌بینید.



شکل ۶-۲۱: فعال و غیرفعال سازی دکمه jQuery UI

بازآوری وضعیت دکمه jQuery UI

متد refresh وضعیت دکمه jQuery UI را اصلاح می‌کند تا هر نوع تغییری در اصل عنصر HTML را انعکاس دهد. این برای انعکاس تغییرات برنامه‌نویسی مفید است.

لیست ۸-۲۱: بازآوری دکمه jQuery UI

```
...
<script type="text/javascript">
    $(document).ready(function() {
        $.ajax("mydata.json", {
            success: function(data, status) {
                var tmplData = $("#flowerTmpl")
                    .template({ flowers: data }).filter("");
                template.tmpl(data.slice(0, 3)).appendTo("#row1");
                template.tmpl(data.slice(3)).appendTo("#row2");
            }
        });
        $('<span>Enabled:<span><input type="checkbox" checked
        />').prependTo("#buttonDiv");
        $(':checkbox').change(function(e) {
            var buttons = $('button');
            if ($(':checked').length == 1) {
                buttons.removeAttr("disabled");
            } else {
                buttons.attr("disabled", "disabled");
            }
            buttons.button("refresh");
        });
        $('button').button();
    });
</script>
...
```

در این مثال از چک باکس استفاده کردم تا صفت disabled را از عنصر دکمه HTML حذف و اضافه نمایم. این تغییر را بصورت برنامه‌نویسی انجام می‌دهم و از متد refresh برای برگرداندن وضعیت نهایی استفاده کردم.

استفاده از رویداد دکمه

ویجت jQuery UI رویدادهایی که بیشتر از اصل عنصر هستند را تعریف می‌نماید. ویجت button یک رویداد ساده به نام create تعریف می‌کند که در هنگام ایجاد دکمه jQuery UI روی می‌دهد. مانند متدها از رویدادها نیز با استفاده از آرگومانهای از پیش تعریف شده استفاده می‌کنید.

لیست ۹-۲۱: استفاده از رویداد create دکمه jQuery UI

```
...
<script type="text/javascript">
    $(document).ready(function() {
        $('button').button({
            create: function(e) {
                $(e.target).click(function(e) {
                    e.preventDefault();
                    alert("Button was pressed");
                })
            }
        });
    });
</script>
...
```

در این مثال من از رویداد create استفاده کردم تا یک فانکشن در پاسخ به رویداد کلیک تنظیم نمایم. رویداد create خیلی مفید نیست و کاری که قرار است انجام دهد را به صورت عادی هم می‌توان انجام داد.

ایجاد انواع متفاوت دکمه

متد button به نوع عنصری که روی آن اعمال می‌شود حساس است. زمانی که متد button روی عناصر button، a یا عناصر input از نوع submit, reset یا button فراخوانی شود رفتارش عادی است. لیست ۱۰-۲۱ همه‌ی این عناصر را نشان می‌دهد.

لیست ۱۰-۲۱: ایجاد دکمه‌های استاندارد

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script type="text/javascript">
    $(document).ready(function() {
        $('.jqButton').click(function(e) {
            e.preventDefault();
            $(this).button();
        });
    });
</script>
</head>
<body>
    <form>
        <input class="jqButton" type="submit" id="inputSubmit" value="Submit">
        <input class="jqButton" type="reset" id="inputReset" value="Reset">
        <input class="jqButton" type="button" id="inputButton" value="Input Button">
        <button class="jqButton">Button Element</button>
    </form>
</body>
</html>
```

```
<a class="jqButton" href="http://apress.com">A Element</a>
</form>
</body>
</html>
```

در این مثال من هر یک از عناصر فوق را استفاده کردم و از متد click برای انتقال به دکمه jQuery UI بهره بردم. اثر آنرا در شکل ۲۱-۷ ببینید.



شکل ۲۱-۷: ایجاد دکمه‌های استاندارد jQuery UI

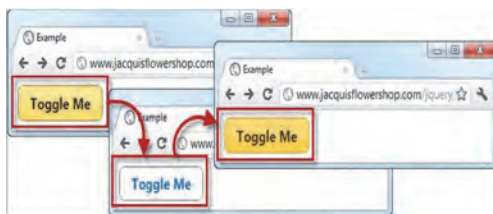
ایجاد یک دکمه وضعیتی

اگر متد button را روی عنصر input از نوع checkbox فراخوانی کنید آنگاه یک دکمه وضعیتی بدست می‌آورید. دکمه وضعیتی در هنگام کلیک شدن، روشن و خاموش می‌شود. (همان تیک خوردن و نخوردن در وضعیت اصلی).

لیست ۲۱-۱۱: اعمال jQuery UI روی چک باکس

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script type="text/javascript">
    $(document).ready(function() {
        $('#jqButton').button();
    });
</script>
</head>
<body>
    <form>
        <input class="jqButton" type="checkbox" id="toggle">
        <label for="toggle">Toggle Me</label>
    </form>
</body>
</html>
```

برای ایجاد یک دکمه jQuery UI از چک باکس باید یک عنصر input داشته باشید و با عنصر label منطبق سازید. jQuery UI یک دکمه ایجاد می‌کند که شبیه دکمه اصلی است. اما در هنگام کلیک روی آن وضعیت آن روشن/خاموش می‌شود. نتیجه کار را در شکل ۲۱-۸ ببینید.



شکل ۸-۲۱: ایجاد یک دکمه وضعیتی از چک باکس

بخاطر داشته باشید که jQuery UI اصل عنصر را تغییر نمی‌دهد. بنابراین چک باکس هنوز همان رفتاری دارد که مرورگر آن را درون فرم می‌بیند. تغییر وضعیت با استفاده از صفت checked منعکس شده است.

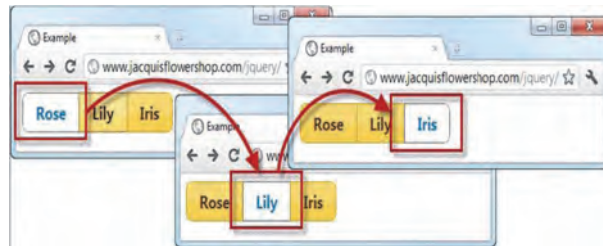
ایجاد یک مجموعه دکمه

با استفاده از متد buttonset دکمه‌های jQuery UI از عناصر دکمه‌های رادیویی را می‌توانید بسازید.

لیست ۱۲-۲۱: ایجاد یک مجموعه دکمه

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script type="text/javascript">
    $(document).ready(function() {
        $('#radioDiv').buttonset();
    });
</script>
</head>
<body>
    <form>
        <div id="radioDiv">
            <input type="radio" name="flower" id="rose" checked />
            <label for="rose">Rose</label>
            <input type="radio" name="flower" id="lily"/><label
            for="lily">Lily</label>
            <input type="radio" name="flower" id="iris"/><label
            for="iris">Iris</label>
        </div>
    </form>
</body>
</html>
```

توجه کنید که عنصر div (شامل چندین عنصر رادیویی) را بکار بردم و متد buttonset را روی آن فراخوانی کردم. متد button را روی عناصر input خاص نمی‌توان فراخوانی کرد. اثر متد buttonset را در شکل ۹-۲۱ می‌بینید.



شکل ۲۱-۹: ایجاد یک مجموعه دکمه

مانند دکمه‌های رادیویی که یک دکمه می‌تواند انتخاب می‌شود، در اینجا به شما اجازه‌ی کلیک یک دکمه را می‌دهد و این با اصل دکمه در تضاد است. توجه کنید که jQuery UI در فضاهایی که دکمه‌ها به هم می‌رسند با اعمال سبک‌های متفاوت، لبه‌ها را به هم می‌چسباند. در شکل ۲۱-۱۰ با دقت بیشتری این امر نشان داده شده است.



شکل ۲۱-۱۰: سبک jQuery UI روی مجموعه عناصر

ساخت مجموعه دکمه از دکمه‌های عادی و منظم

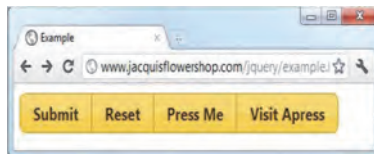
از متد buttonset روی هر عنصری که متد button روی آن قابل اعمال است، می‌توانید استفاده نمایید. این همان اثر ظاهری دکمه‌های رادیویی را دارد ولی اثر رفتاری آنرا ندارد. لیست ۲۱-۱۳ استفاده از متد buttonset را نشان می‌دهد.

لیست ۲۱-۱۳: ایجاد یک مجموعه دکمه از دکمه‌های عادی

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script type="text/javascript">
    $(document).ready(function() {
        $('#radioDiv').buttonset();
    });
</script>
</head>
<body>
    <form>
        <div id="radioDiv">
            <input type="submit" value="Submit"/>
            <input type="reset" value="Reset"/>
            <input type="button" value="Press Me"/>
            <a href="http://apress.com">Visit Apress</a>
        </div>
```

```
</form>
</body>
</html>
```

در شکل ۱۱-۲۱ عناصر ورودی که در عنصر div آمده و تشکیل دکمه داده‌اند و لبه‌هایشان به هم چسبیده است به خوبی نشان داده شده است.



شکل ۱۱-۲۱: ایجاد مجموعه دکمه از دکمه عادی

نکته: از این تکنیک با دقت استفاده کنید. می‌توان برای کاربر گیج‌کننده باشد؛ به ویژه اگر در جایی دیگر از سند از این دکمه‌های رادیویی استفاده کرده باشید.

استفاده از نوار پیشرفت jQuery UI

اینک که از دکمه‌ها برای فهم ساختار ویجت jQuery UI استفاده کرده‌اید، می‌توانید کنکاش در دیگر ویجتهای jQuery UI را آغاز کنید. کار را با نوار پیشرفت کار آغاز می‌کنیم.

نوار پیشرفت کار اجازه می‌دهد تا پیشرفت کار کاربر در تکمیل یک وظیفه را نمایش دهید. پیشرفت کار تنها روی وظایف معین و پایان پذیر طراحی شده است و جایی که بتوانید به کاربر این پیشرفت را بصورت درصد بیان کنید. زمانی که دیدی درباره‌ی اتفاقات در ادامه کار را ندارید و تنها به کاربر می‌توانید بگویید منتظر باشد، انتخاب دیگر وظایف نامعین است.

نمایش اطلاعات مفید پیشرفت کار

قوانین خاصی درباره‌ی چگونگی استفاده از ویجتها در برنامه‌های کاربردی وب وجود ندارد. اما انتظارات کاربر در مواردی مانند پیشرفت کار با توجه به استانداردهای سیستم عاملهایی مانند ویندوز شکل گرفته است. برای کمک به کاربر درباره احساس پیشرفت کار، ۲ قانون وجود دارد. نخست تنها افزایشها را نشان دهد و در زمانی که مثلاً بیش از حدانتظار پیشرفت نشان داده شده، آنرا کاهش ندهد. نوار پیشرفت کار، حدی از کار که کامل شده است را نشان می‌دهد. زمان باقی مانده تخمین زده نمی‌شود. اگر مسیرهای متفاوتی برای نشان دادن ادامه کار وجود دارد، بدبینانه‌ترین حالت انتخاب شود. حتی از جهش‌های بزرگ نیز می‌توانیم استفاده کنیم.

دوم نوار پیشرفت کار بیش از یک بار گردش نداشته باشد. اگر اطلاعات کافی درباره اتمام کار ندارید از نمونه نامعین استفاده کنید و وقتی پیشرفت نزدیک ۱۰۰ درصد شد کاربر انتظار دارد که کار تمام شده باشد و اگر دوباره نوار پیشرفت کار از ابتدا آغاز کند، کاربر گیج می‌شود و دیگر این پیشرفت کار بی‌معنی است.

ایجاد نوار پیشرفت کار

یک نوار پیشرفت کار را با انتخاب عنصر div و فراخوانی متد progressbar می‌توانید ایجاد کنید.

```
<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js" type="text/javascript"></script>
<script src="jquery-ui-1.10.3.custom.js" type="text/javascript"></script>
<link rel="stylesheet" type="text/css" href="jquery-ui-1.10.3.custom.css"/>
<script type="text/javascript">
    $(document).ready(function() {
        $('#progressDiv').progressbar({
            value: 21
        });
    });
</script>
</head>
<body>
    <div id="progressDiv"></div>
</body>
</html>
```

در این مثال، سند شامل یک عنصر div با شناسه progressDiv است. برای ایجاد نوار پیشرفت کار باید از یک div خالی استفاده کنید و اگر محتوایی داشته باشد تحت تاثیر ویجت قرار می‌گیرد. عنصر progressDiv را انتخاب و متد progressBar را فراخوانی می‌کنیم. با ارسال شیء map پیکربندی آغازین را آماده می‌کنیم. نوار پیشرفت کار ۲ تنظیم دارد که در جدول ۴-۲۱ آمده است.

تنظیم	شرح
disabled	اگر true باشد، نوار پیشرفت غیرفعال است مقدار پیش فرض آن false می‌باشد
value	درصد تکمیل نمایش را تنظیم می‌کند. پیش فرض آن صفر است
max	تنظیم بیشینه مقداری که نوار پیشرفت نمایش خواهد داد. پیش فرض ۱۰۰ است

در این مثال مقدار آغازین را ۲۱ درصد معین کرده‌ایم و اثر آنرا در شکل ۱۲-۲۱ می‌بینید.



شکل ۱۲-۲۱: ایجاد نوار پیشرفت

ایجاد نوار پیشرفت نامعین

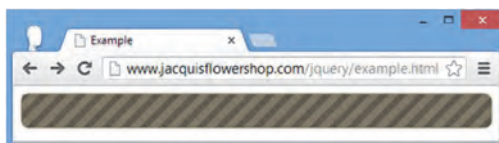
در jQuery UI 1.10 ویجت نوار پیشرفت، پیشرفت را برای کارهای نامعین نیز پشتیبانی کرده است. لازم است مقدار تنظیم value را بجای عدد false قرار داد.

```

<!DOCTYPE html>
<html>
<head>
<title>Example</title>
<script src="jquery-2.0.2.js"></script>
<script src="jquery-ui-1.10.3.custom.js"></script>
<link href="jquery-ui-1.10.3.custom.css" rel="stylesheet" />
<script type="text/javascript">
    $(document).ready(function () {
        $("#progressDiv").progressbar({
            value: false
        });
    });
</script>
</head>
<body>
    <div id="progressDiv"></div>
</body>
</html>

```

نتیجه را در شکل ۱۳-۲۱ می‌بینید.



شکل ۱۳-۲۱: ایجاد نوار پیشرفت نامعین

استفاده از متدهای نوار پیشرفت

ویجت نوار پیشرفت تعدادی متد تعریف می‌کند که از لحاظ سبکی شبیه دکمه است. به عبارت دیگر متد progressbar را فراخوانی می‌کنید و با نخستین آرگومان معین می‌کنید چه می‌خواهید. جدول ۵-۲۱ شرح این متدهاست.

جدول ۵-۲۱: متدهای نوار پیشرفت

ویژگی	شرح
progressbar("destroy")	عنصر HTML را به وضعیت آغازین خود برمی‌گرداند
progressbar ("disable")	غیرفعال کردن نوار پیشرفت
progressbar ("enable")	فعال کردن نوار پیشرفت
progressbar ("option")	تنظیم یک یا چندین گزینه. بخش پیکربندی دکمه را ببینید
progressbar ("value",value)	گرفتن یا مقداردهی مقدار نمایشی نوار پیشرفت