

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

متدهای (`post()`) و (`get()`)

مدرس : مهندس افشین رفوآ

دوره آموزشی JQuery

متدهای (`post()`) و (`get()`)

متدهای (`get()`) و (`post()`) به شما این امکان را می دهند تا به راحتی درخواست **HTTP** را به سرور بدهند (server) ارسال کرده و نتیجه ی آن را دریافت کنید. زمانی که یک فرم را ارسال می کنید، در واقع یکی از دو درخواست **GET** و **POST** مورد استفاده قرار می گیرد. با بهره گیری از دو تابع (`get()`) و (`post()`) کتابخانه ی **jQuery** می توان همین عملیات را شبیه و پیاده سازی کرد.

دو متد ذکر شده، از آنجایی هر یک نوع خاصی از درخواست را فراخوانی می کنند، از بسیاری از جهات شبیه به هم هستند. همچنین این دو متد **static** بوده و برای کار با آن ها نیازی به ساختن شی یا نمونه از کلاس **jQuery** نیست. کافی است یکی از دو متد نام برده را، با نوشتن دستور (`jQuery.get()`) یا میانبر آن "`$.get()`"، مستقیماً بر روی کلاس **jQuery** صدا بزنید. در ساده ترین حالت (`get()`) و (`post()`) یک پارامتر بیشتر نمی گیرند و آن هم **URL** فایلی است که می خواهید اطلاعات از آن خوانده شود (**URL** مورد درخواست می باشد). اما گاهی لازم است با اطلاعات برگشتی عملیات خاصی را انجام دهیم. در آن صورت یک تابع **callback** نیز به عنوان پارامتر به تابع پاس می دهیم که در صورت با موفقیت اجرا شدن درخواست، **jQuery** صدا می زند.

حال به چندین مثال کاربرد می پردازیم. در درس قبلی یک فایل **HTML** به نام "`content.html`" ایجاد کرده و به واسطه ی متد **load** کتابخانه ی جی کوئری در ظرف مورد نظر (**div**) بارگذاری کردیم. پیش از تست کردن نمونه ی زیر، اطمینان حاصل نمایید فایل "`content.html`" در پوشه ای که فایل حاوی مثال در آن قرار دارد، ایجاد شده است. محتویات آن چندان اهمیتی ندارد و می تواند هر چیزی باشد.

```

<script type="text/javascript">
$(function () {
$.get("content.html", function (data, textStatus) {
alert("Done, with the following status: " + textStatus + ". Here is the response: " + data);
});
});
</script>

```

اولین پارامتر یک **URL** است که مشخص کننده ی آدرس فایل درخواستی می باشد (**content.html**). دومین پارامتر یک تابع **callback** است که در صورت موفقیت درخواست توسط **jQuery** فراخوانی و اجرا می شود. اولین پارامتر تابع **callback** صرفاً محتوای صفحه ی درخواستی را مشخص می کند (اطلاعاتی است که می خواهیم همراه با درخواست به سرویس دهنده بفرستیم)، در حالی که پارامتر دوم وضعیت درخواست را به صورت رشته برمی گرداند (وضعیت درخواست را در قالب یک رشته مشخص می کند).

می توان مانند مثال بالا یک صفحه ی ساده ی **HTML** درخواست کرد. اما زمانی که از دو متد **get()** یا **post()** استفاده می کنیم، در واقع قصد داریم تعدادی پارامتر هم ارسال کنیم که توسط سرور (با کد **PHP** یا **ASP**) مورد پردازش قرار می گیرد و سپس نتیجه را برگرداند. **jQuery** قادر است مجموعه ای از پارامترهای **GET** یا **POST** را به عنوان آرگومان بپذیرد:

```

<script type="text/javascript">
$(function () {
$.post("test_post.php",
{
name: "John Doe",
age: "42"
},
function (data, textStatus) {
alert("Response from server: " + data);
});
});
</script>

```

مثال حاضر، مشابه نمونه ی قبلی است، اما این بار درخواست را با روش **POST** به صفحه ی دیگر که یک فایل **PHP** است، می دهیم. در واقع به عنوان پارامتر اول آدرس یک فایل **PHP** را ارسال می کنیم و به عنوان پارامتر دوم مجموعه (جفت کلید مقدار) پارامترهای **POST** را به متد **post()** پاس می دهیم. این مجموعه متشکل از دو آرگومان **name** و **age** می باشد. اگر داده ها را با روش **GET** به سرور ارسال می کردیم (**post**،

بر خلاف **get**، از **URL** پارامتر نمی گیرد) پارامترها به صورت **querystring** در **URL** (داخل نوار آدرس مرورگر) به نمایش در می آید، بدین صورت:

test_get.php?name=John Doe&age=42

اسکرپت **PHP** این پارامترها را خوانده، مورد پردازش قرار می دهد و سپس نتیجه را برمی گرداند. اسکرپت مربوطه در سمت سرور دو مقدار را دریافت کرده و یک رشته مانند "**<name> is <age> years old**" ایجاد می کند و سپس آن را بازمی گرداند.

