

# Ajax

# Ajax

- Up until now, our interaction with JavaScript has been mostly limited to: push a button, something happens.
- We still don't have to entirely reload our page, but there is still some degree of user interaction.
- Ajax (formerly *Asynchronous JavaScript and XML*) allows us to dynamically update a webpage even more dynamically.
  - Though, for now, we won't go too crazy!

# Ajax

- Central to our ability to asynchronously update our pages is to make use of a special JavaScript object called an XMLHttpRequest.

# Ajax

- Central to our ability to asynchronously update our pages is to make use of a special JavaScript object called an XMLHttpRequest.

```
var xhttp = new XMLHttpRequest();
```

# Ajax

- Central to our ability to asynchronously update our pages is to make use of a special JavaScript object called an XMLHttpRequest.
- After obtaining your new object, you need to define its `onreadystatechange` behavior.
  - This is a function (typically an anonymous function) that will be called when the asynchronous HTTP request has completed, and thus typically defines what is expected to change on your site.

# Ajax

- Central to our ability to asynchronously update our pages is to make use of a special JavaScript object called an XMLHttpRequest.
- After obtaining your new object, you need to define its `onreadystatechange` behavior.
  - This is a function (typically an `anonymous function`) that will be called when the asynchronous HTTP request has completed, and thus typically defines what is expected to change on your site.

# Ajax

- XMLHttpRequests have two additional properties that are used to detect when the page finishes loading.
  - The readyState property will change from 0 (request not yet initialized) to 1, 2, 3, and finally 4 (request finished, response ready).
  - The status property will (hopefully!) be 200 (OK).
- Then just make your asynchronous request using the open( ) method to define the request and the send( ) method to actually send it.

# Ajax

- XMLHttpRequests have two additional properties that are used to detect when the page finishes loading.
  - The `readyState` property will change from 0 (request not yet initialized) to 1, 2, 3, and finally 4 (request finished, response ready).
  - The `status` property will (hopefully!) be 200 (OK).
- Then just make your asynchronous request using the `open()` method to define the request and the `send()` method to actually send it.
  - There is a slightly different way to do this syntactically with jQuery!



# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

```
function ajax_request(argument)
{
    var aj = new XMLHttpRequest();
    aj.onreadystatechange = function() {
        if (aj.readyState == 4 && aj.status == 200)
            // do something to the page
    };

    aj.open("GET", /* url */, true);
    aj.send();
}
```

# Ajax

- More commonly, you'll see Ajax requests written using jQuery instead of “raw” JavaScript.



# Ajax

- More commonly, you'll see Ajax requests written using jQuery instead of “raw” JavaScript.

**<http://api.jquery.com/jquery.ajax/>**