

# AJAX QUICK REFERENCE

---

# D

Chapters 19 and 20 present why and how Ajax can be used to add data interactivity to your Web pages without having to reload the entire page. This online resource gives you quick access to the properties, methods, and state codes you will need to use on a regular basis, as well as the `ajaxBasics()` function.

- ◆ **Table D.1:** Ajax Properties (Chapter 20)
- ◆ **Table D.2:** Ajax Methods (Chapter 20)
- ◆ **Table D.3:** `readyState` Variables (Chapter 20)

Table D.1

Ajax Properties	
PROPERTY	DESCRIPTION
onreadystatechange	Event handler triggered when the state of the request object changes.
readyState	Status code of the current state of the server request object (see Table 20.3).
responseText	String value of the data from the server.
responseXML	DOM-compatible document object of the data from the server.
status	Status code of the response from the server.
statusText	String value of the status message.

Table D.2

Ajax Methods	
METHOD	DESCRIPTION
abort()	Cancel the current HTTP request.
getAllResponseHeaders()	Value of all HTTP headers
getResponseHeader(<label>)	Value of specified HTTP header.
open(<method>, <URL>, <asynchFlag>)	Initializes the XML HTTP request.
send(<content>)	Sends the request to the server and receives the response.
setRequestHeader(<label>, <value>)	Sets the name of the HTTP header sent.

Table D.3

readyState Values	
VALUE	DESCRIPTION
0	Uninitialized. The object has no data.
1	Loading. The object has data loading.
2	Loaded. The object has finished loading data.
3	Interactive. Data is still loading, but the object can be interacted with.
4	Complete. Data is fully loaded and ready for use.

## AjaxBasics Functions

This external JavaScript file (`ajaxBasics.js`) contains the `fetchData()` function so that you do not have to add it directly to every page in the site.

```
// ajaxBasics.js: Include this file
→ when creating Ajax enabled
functions.

function fetchData(url,dataToSend,
→ objectID){
    var pageRequest = false
    if (window.XMLHttpRequest) {
        pageRequest = new
        → XMLHttpRequest()
    }
    else if (window.ActiveXObject){
        try {
            pageRequest = new
            → ActiveXObject("Msxml2.
            → XMLHTTP")
        }
        catch (e) {
            try{
                pageRequest = new
                → ActiveXObject
                → ("Microsoft.XMLHTTP")
            }
            catch (e){}
        }
    }
    else return false
    pageRequest.onreadystatechange=
    → function() {
        filterData(pageRequest,objectID)
```

*continues on next page*

```
}
  if (dataToSend) {
    var sendData = 'sendData='
    → + dataToSend;
    pageRequest.open
    → ('POST',url,true);
    pageRequest.setRequestHeader
    → ('Content-Type',
    → 'application/x-www-form-
    → urlencoded');
    pageRequest.send(sendData);
  }
  else {
    pageRequest.open('GET',url,true)
    pageRequest.send(null)
  }
}
```