

Module 4: Learning Outcomes

- Handling Events, Http Basics, URI VS URL, Http Methods, Typical HTTP Response, Response Status Codes, AJAX, Synchronous vs Asynchronous Execution, JSON, JQuery Basics



Handling Events

- What are event handlers?
- Event handlers are basically functions that you bind using specific methods to certain events that happen to a browser.
- Those events maybe trigger by user interaction like user click something.
- Create Example1.html,Example2.html,Example3.html, and Example3A,Example4.html,Example5.html, and Example6.html

Http Basics

- What is HTTP?
- HyperText Transfer Protocol
- Based on request/response stateless protocol
- Stateless means when a server response to a request from a browser that response does not depend on any previous request that the browser make.
- The server has no idea if the browser has request anything from it before.

Http Basic

- Based on request/response stateless protocol
 - Client opens connection to server.
 - Client sends HTTP request for a resource
 - Server sends HTTP response to the client(usually with the resource the client request)
 - Client closes connection to server.

URI

- URI: Uniform Resource Identifier
- Uniquely identifies resource or location of resource.
- Uniform Resource Identifier(URI) is a string of characters used to identify a name or a resource on the Internet.
- A URI has two specializations known as URL and URN.

URI

- For Example:

- Name: Albert Lam
- Address: 32 Fitzwilliam Avenue
Richmond Hill, Ontario

URI is using Albert's name or address to identify the person we want to get to.

URL

- A Uniform Resource Locator (URL) is a subset of the Uniform Resource Identifier (URI) that specifies where an identified resource is available and the mechanism for retrieving it.
- A URL can be “http://” or [ftp://](#)
- We can identify Albert by just using his location:
 - Address: 32 Fitzwilliam Avenue

URL

- <http://www.UT.ca/courses/course.php>
- http → protocol
- www.UT.ca → Domain. Domain is case insensitive.
- 80 → port 80 by default if don't specify in the url. A Port is a type of software connection point used by the TCP/IP protocol and the connecting computer.
 - Every protocol has default port number.
 - http port default is 80
 - https port default is 443
- /Courses/course.php → is the path.
 - Query String
 - &id=jadeite100&password=Jadeite1

URN

- URN: Uniform Resource Name
- Uniquely identifies resource or name of resource
- Does not tell us how to get the resource
- Example
 - Name: Albert Lam
- URN may be used to talk about a resource without implying its location or how to access it. For example, in the [International Standard Book Number \(ISBN\)](#) system, ISBN 0-486-27557-4 identifies a specific edition of Shakespeare's play [Romeo and Juliet](#). The URN for that edition would be urn:isbn:0-486-27557-4. To gain access to the book, its location is needed, for which a URL would have to be specified. (https://en.wikipedia.org/wiki/Uniform_Resource_Identifier)

HTTP Methods

- GET
 - Retrieves the resource
 - Data is passed to server as part of the URI
 - I.E., query string
- POST
 - Send data to server in order to be processed
 - Data is sent in the message body.

HTTP Request Structure(POST)

- POST / index.html HTTP/1.1
- HOST: www.UT.ca
- Accept-Charset: utf-8



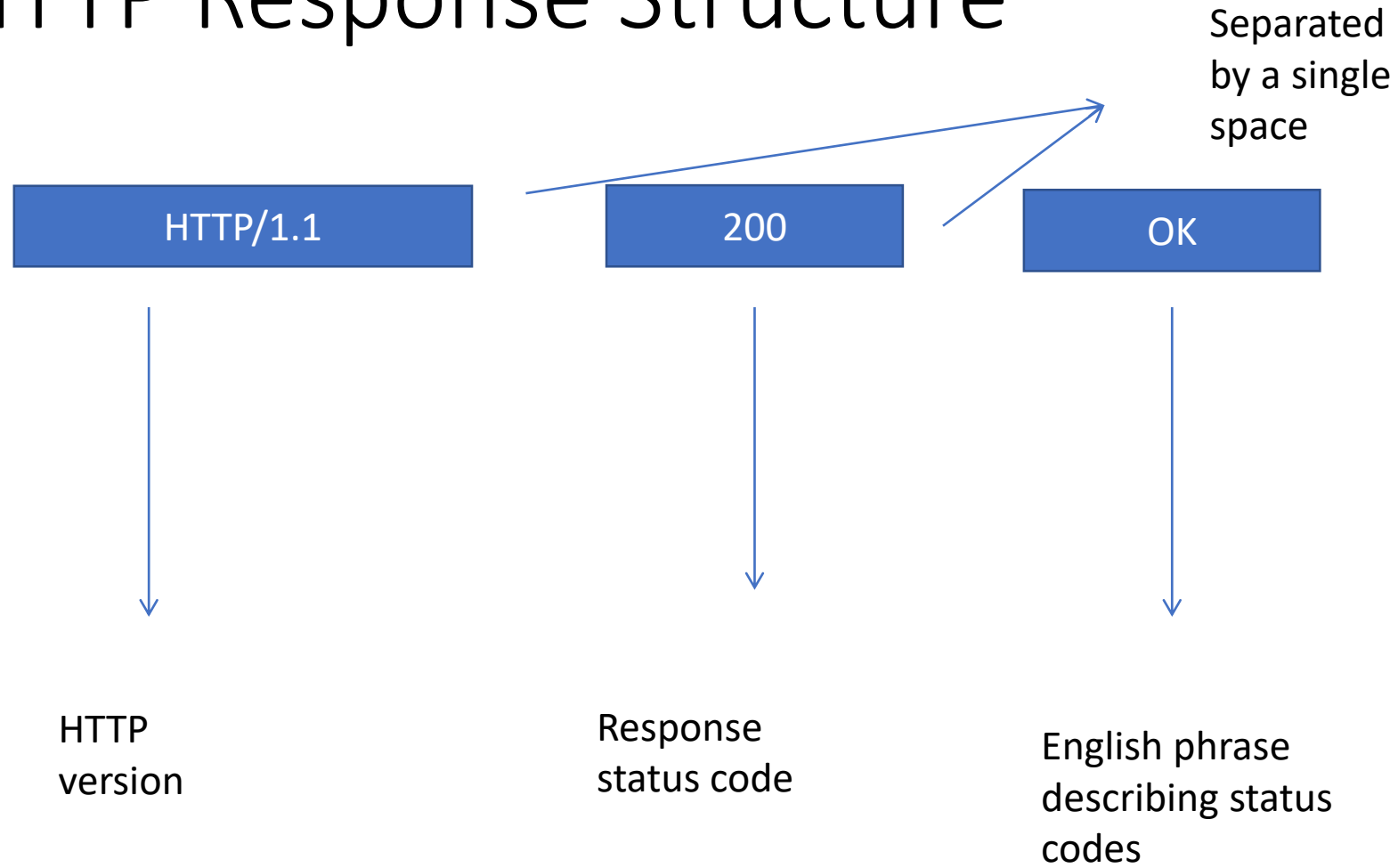
Host: www.ut.ca
Accept-Charset;ut-f8

Request
headers

firstName=Albert

Message
body

HTTP Response Structure



Typical HTTP Response

- HTTP/1.1 200 OK
- Date: Tue, 11, Feb 2017 19:00:01 GMT
- Content-Type: text/html
- <html>
- <body>
- <h1>UT Course</h1>
- </body>
- </html>

Some Response Status Codes

- 200 OK
 - OK, here is the content you requested
- 404 Not Found
 - Server can't find the resource requested
- 403 Forbidden
 - Unauthenticated client tried to access a secure resource
- 500 Internal Server Error
 - Some unhandled error was raised on the server.

JavaScript Closures

- Closures is an important concept to understand without this concept implement in Javascript something like Ajax wouldn't be possible.
- Create Example7.html

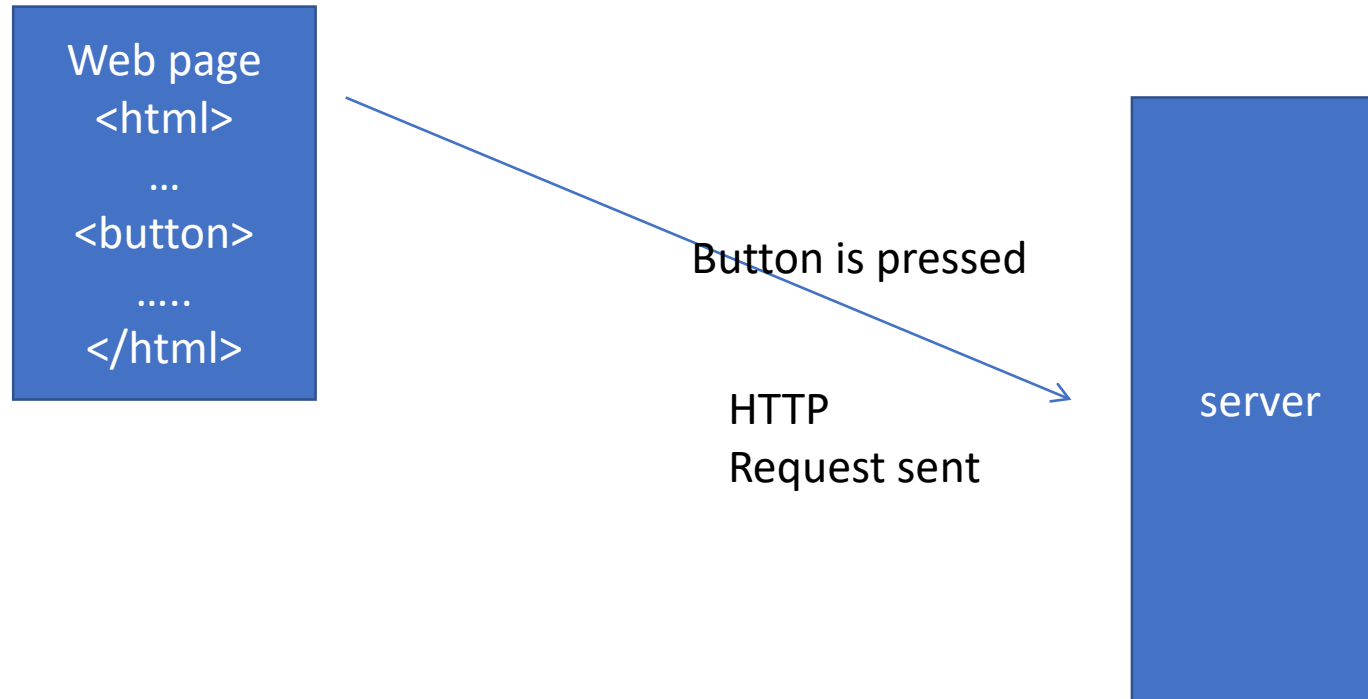
Faked Namespace

- Created Example8.html to indicate problem.
- Created Example9.html how to use faked namespace to solve problem in Example8.html
- Created Example10.html add another variable called “greeting”.
- Created Example11.html to demonstrate Immediately Invoked Function Expression
- Created Example12.html to demonstrate solution to Example10.html

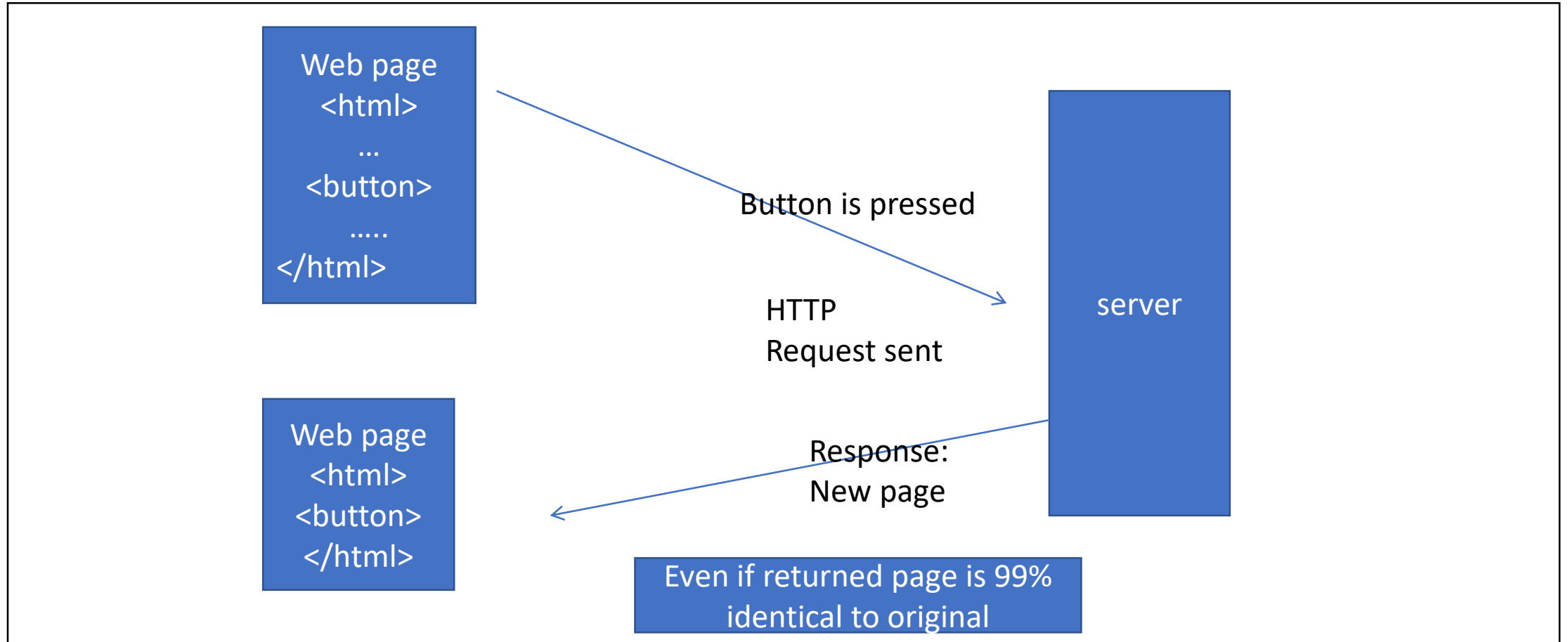
AJAX

- What is Ajax?
- Stands for
 - Asynchronous Javascript and xml
 - While Ajax started with XML, very few apps use it nowadays.
 - Json is used instead.

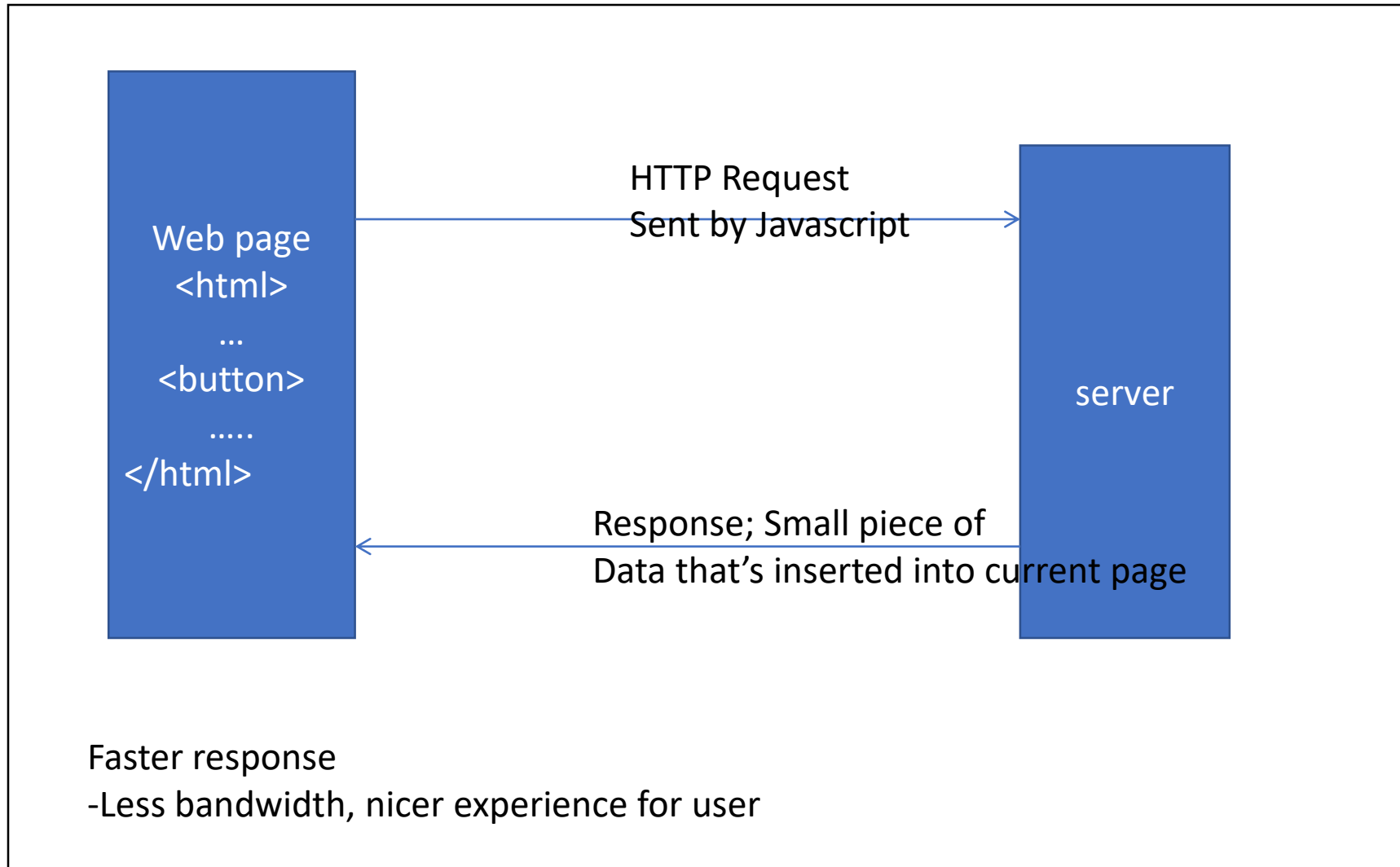
Traditional Web App Flow



Traditional Web App Flow



Ajax Web App Flow



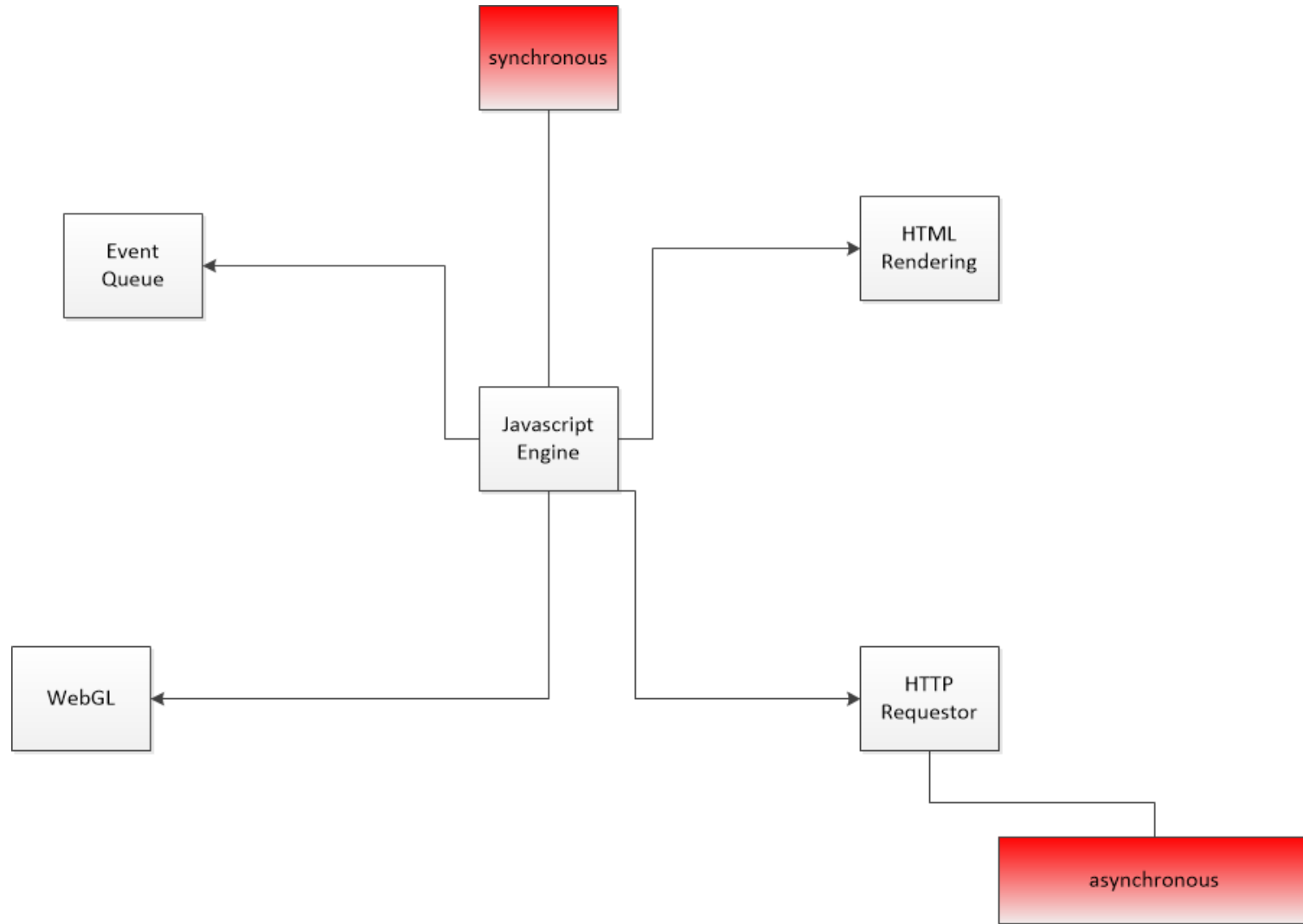
Synchronous Execution

- Execution of one instruction at a time.
- Can't start execution of another instruction until the first instruction finished its execution.

Asynchronous Execution

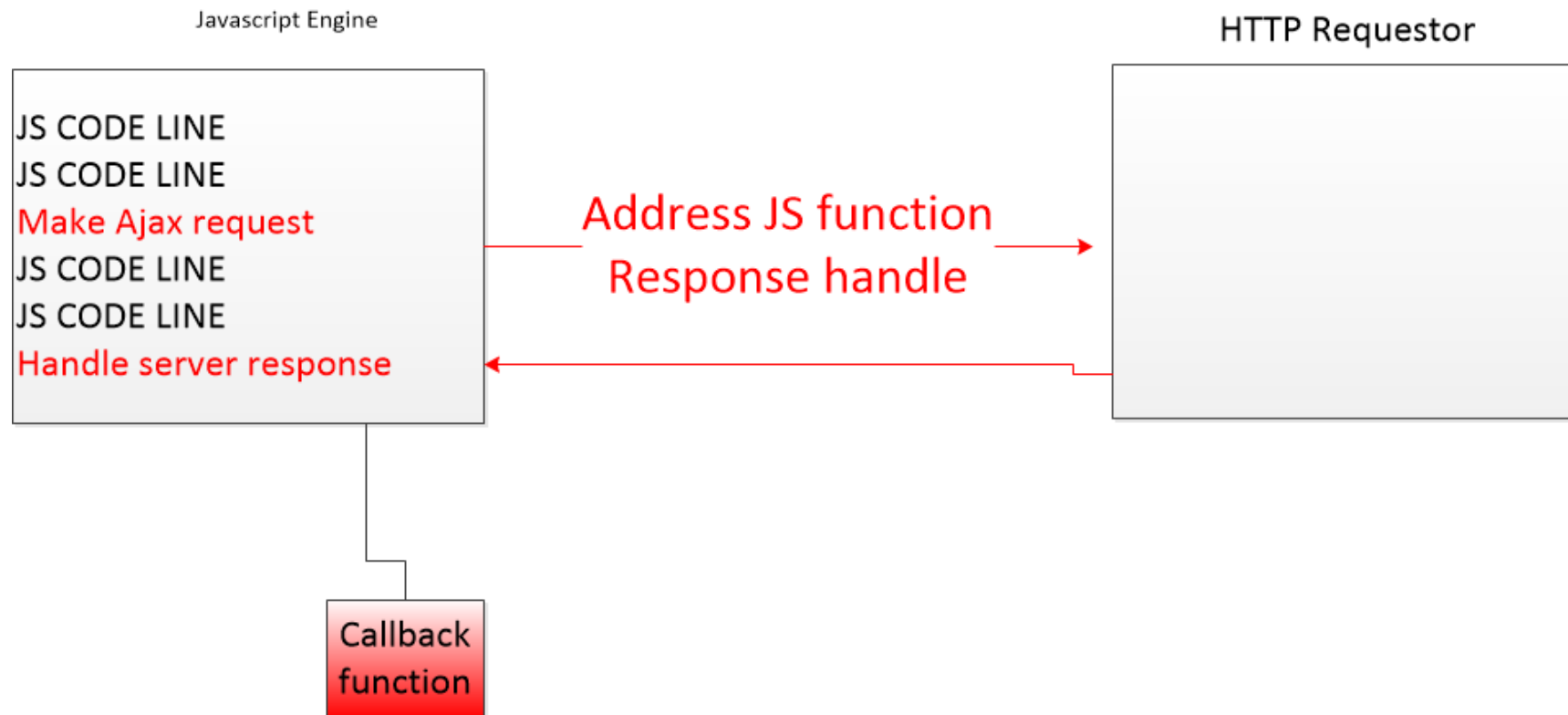
- Execution of more than one instruction at a time.
- Asynchronous instruction returns right away.
- The actual execution is done in a separate thread or process.

How Does Ajax Work Then



AJAX PROCESS

AJAX PROCESS



Ajax

- Create Example13.html –This example shows how to call server side code.

What is JSON?

- Stands for
 - JavaScript Object Notation
 - Lightweight data-interchange format
 - Simple textual representation of data
 - Easy for humans to read and write
 - Easy for machines to parse and generate
 - Completely independent of any language

JSON Syntax Rules

- Subset of Javascript object literal syntax..but
 - Property names must be in double quotes
 - String values must be in double quotes
 - Separate each value by a comma.
- Syntax for everything else is exactly like for object literal

JSON Example

- See name.json

```
{  
  "firstName": "Albert",  
  "lastName": "Lam",  
  "likesChineseFood": true,  
  "numberOfDisplays": 2  
}
```

Converting JSON To String & Back to JSON

- Converts from json string to object

```
var obj = JSON.parse(jsonString);
```

- Converts from object to json string

```
var str = JSON.stringify(obj);
```

Created Example14.html

JQuery Basic

- JQuery is a client side-scripting library of JavaScript.
- You can access any element and validate input by using the library.
- No extra code can achieve the result by writing one or few lines of code instead of writing dozen lines of code.

Advantages of JQuery

- It helps to run with all kind of browsers and is compatible various browsers.
- It helps to implement critical functionality without writing hundreds of line of codes.
- It is fast to implement customized action.

Install JQuery

- We can use JQuery in two ways:
 - If you want to use JQuery file locally , then download it. Download jQuery from <http://jquery.com/download/>, put the downloaded file in the same folder with your jQuery files, and reference it in <head> section of html document as following:

```
<head>
```

```
    <script src="jquery-3.0.0.js"></script>
```

```
</head>
```


Install JQuery

- 2.) If you do not want to download jQuery, then you can include it in html document as following:

```
<head>
```

```
    <script src="http://code.jquery.com/jquery-latest.js"></script>
```

```
</head>
```

First JQuery Script

- Created Example15.html

Explanation:

`<script src="jquery-3.0.0.js"></script>`

Means to use jQuery code in current document.

`$(document).ready(function () {})` means to run the function automatically when the web page is completely loaded.

`$("#divID")` accesses the tag whose id is "divID".

`html()` displays the contents.

`$("#divID").html("Hello World !")` displays "Hello World!" in a tag whose id is "divID".

`<div id="divID">` defines a tag named "divID" where some contents will be shown.

Run When Page Loaded

- “\$(document).ready(function (){})” means to execute the function when the page is loaded completely.
- Create Example16.html

What is \$("") meaning?

- "\$" is a symbol of jQuery.
- "\$()" accesses an element in current html document.
- For example:
- \$("span") accesses a tag, its tag name is "span".
- \$("#3") accesses a tag, its id is "3".
- \$(".clss5") accesses a tag, its class name is "clss5".

Show Content

- `$(“tag”)` accesses the “tag”.
- `.html()` shows the contents without html symbol.
- Create Example17.html
- **Explanation:**
- `“<script src=“jquery-3.0.0.js”></script>”` uses jQuery code in current document.
- `“$(document).ready(function(){ }”`

Executes the function when the web page is loaded completely.

`“$(“div”).html(“<h1> Hello World!” </h1>”)”` displays “Hello world!” in tag “div”.

Note: You cannot see `<h1>..</h1>` because `.html()` shows contents without html symbol.

Show Content

“\$(“tag”)” accesses the “tag”

.text() shows contents with html symbol.

Create Example18.html

Explanation:

“\$(“div”).text(“<h1>Hello world!</h1>”)” displays
“Hello world!” in tag “div”.

Note: You can see <h1>..</h1>, because .text() shows
contents with html symbol.

Set CSS Style

- `$(“selector”).css(“style”)`
- What is selector?
- All HTML elements based on their id, classes, types(text, radio etc), attributes (id, title,src etc), tag name(div,p,form,table,tr,th,td etc.) etc are jQuery selectors.
- `$(“selector”)` means to access one specified element.
- What is `css()`?
- The `css()` method sets or returns one or more style properties for the selected elements.
- `Css(“style”)` means to set a style for a specified element.
- `$(“selector”).css(“style”)` accesses a specified element, and sets a css style for it.
- `$(“#id”).css()` accesses a tag by its id, and sets a css style.
- If we want to access any element by its “id”, then we have to use “#” to select that element. `css()` method can set or change any property.
- Create Example19.html

Set CSS Style

- **Explanation:**
- `$("#divTest").css("background-color","red")` access a tag whose id is "divTest", and sets its background color as "red".
- `<div id="divTest onclick="changeColor()"` executes the function "changeColor()" when clicking a tag whose id is "divTest".

Set CSS by Tag

- `$("#tag").css()` accesses a tag by its name, and sets a css style.
- If we want to access any element by its “tag” name, then we have to use “tag” name to select that element, `css()` method can set or change any property.
- Create Example20.html
- **Explanation:** `$("#div").css("background-color","red")` accesses a tag whose tag name is “div” and set its background color as “red”.
- `<div onclick="changeColor()"` executes the function “changeColor()” when clicking a tag whose tag name is “div”.
- In Example20.html you can see I am changing “div” background color by using tag “div”.

Set CSS by Class

- “`$(".class").css()`” accesses an element by its class name, and sets a css style.
- If we want to access any element by “class” name then we have to use “class name” to select that element, `css()` method can set or change any property.
- Created Example21.html
- **Explanation:** “`$(".divClick").css("background-color","red")`” accesses a tag whose class name is “divClick”, and sets its background color as “red”.
- “`<div class="divClick" onclick="changeColor()"`” executes the function “`changeColor()`” when clicking a tag whose class name is “divClick”.

Set CSS by Specified Tag

- “\$(“p:nth-child(n)”).css()” accesses the “n” tag, and sets a css style.
- “\$(“p:nth-child(1)”).css()” accesses the first tag, and sets a css style.
- “\$(“p:nth-child(4)”).css()” accesses the fourth tag, and sets a css style.
- Created Example22.html
- **Explanation:** “\$(“p:nth-child(1)”).css()” accesses the first tag, and sets a css style.
- “\$(“p:nth-child(4)”).css()” accesses the fourth tag, and sets a css style.
- “onclick=“setCSS()”” runs the setCSS() when clicking the button.

Set CSS by First Tag

- `$("#tag:first").css()` accesses the first element, and sets a css style.
- `":first"` selector is used to select the first element from matched elements.
- Create Example23.html
- **Explanation:** `$("#div:first").css("background-color","black")` accesses the first tag "div", and sets its background color as black.
- `"<input type="button" value="Click Me" onclick="changeColor()"/>"` runs `"changeColor()"` when clicking the button.
- The color of 1st location div has been change from aqua to black.

Set CSS by Last Tag

- `$(“tag:last”).css()` access the last element, and sets a css style.
- “:last” selector is used to select the last tag from matched elements.
- Created Example24.html
- **Explanation:** “`$(“div:last”).css(“background-color”,“black”)`” accesses the last tag “div”, and sets its background color as black.
- “`<input type=“button” value=“Click Me” onclick=“changeColor()”/>>`” runs “changeColor()” when clicking the button.

Not Filter

- “:not” is used to get any elements except the specified element.
- Create Example25.html
- **Explanation:** “:not” is used to get any elements except the specified element.
- `$(“li:not(:first)”).css(“color”,“red”);` get any other elements except first element “li”.