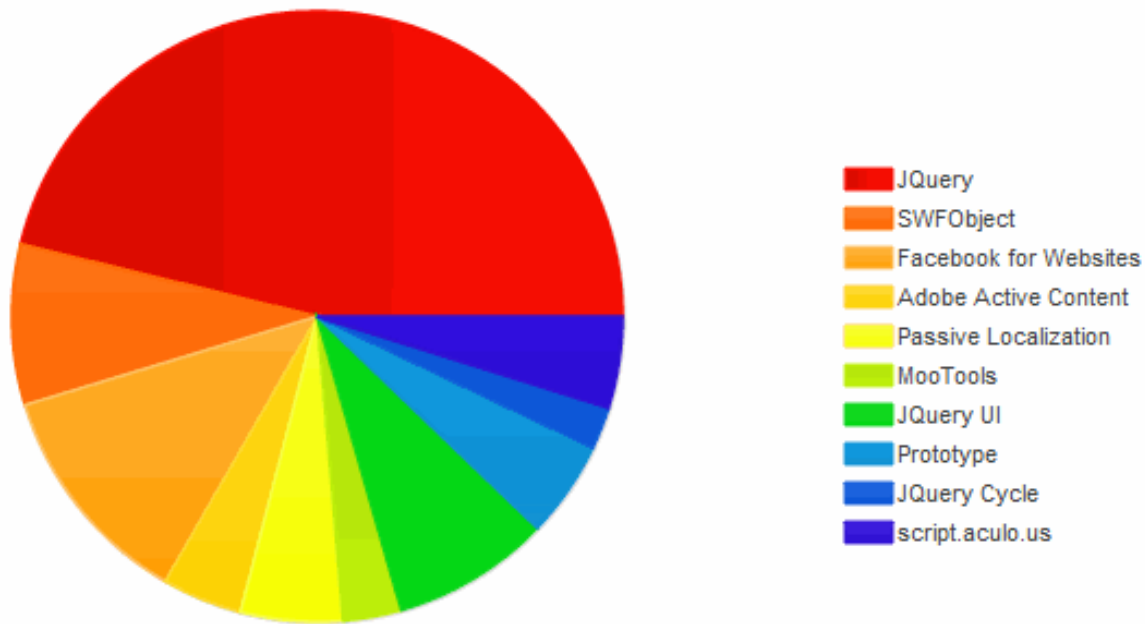


WEB PROGRAMMING LANGUAGES

jQuery and AJAX

JavaScript Frameworks

JavaScript Distribution in Top Million Sites



jQuery: Introduction

- jQuery: **write less, do more**
 - popular JavaScript library
 - used by: Google, IBM, Microsoft, Nokia, Netflix, etc.
 - designed to simplify DOM-oriented client-side scripting
 - lightweight, "write less, do more", JavaScript library
- jQuery makes it easier to write client-side scripting code
- jQuery creates higher-level methods needed for achieving common tasks by wrapping several JavaScript statements into a single call
 - Example: AJAX calls and DOM manipulation

Features

- The jQuery library contains the following features:
 - HTML/DOM manipulation
 - CSS manipulation
 - HTML event methods
 - Effects and animations
 - AJAX
 - Utilities
 - plugins to deal several real-world tasks

jQuery: Introduction

- How to use jQuery:
 - Download the jQuery library from **jQuery.com**
 - Two versions:
 - Production version - live website (production system): minified and compressed
 - Development version - testing and development environments: uncompressed and readable code
 - The jQuery library is a single JavaScript file
 - Place the jQuery library file in the same directory as the pages where you wish to use it
 - Reference jQuery library file with the HTML `<script>` tag

```
<head>
  <script src="jquery-3.3.1.min.js"></script>
</head>
```
 - Include jQuery library from CDNs (Content Delivery Network) like Google or Microsoft:

```
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
  </script>
</head>
```

jQuery: Syntax

- Use jQuery Selector to select/query HTML elements
- Actions can then be performed on the selected/queried elements
- jQuery functions can be stored in a separate .js file and referenced using the “script” element’s “src” attribute
- Basic syntax is: **`$(selector).action()`**
 - \$ sign to specify jQuery
 - *selector* to specify the HTML elements to select/query
 - *action* is the action to be performed on the element(s)

jQuery: Syntax

- Examples:
 - `$(this).hide()` - hides the current element
 - `$("p").hide()` - hides all `<p>` elements
 - `$(".test").hide()` - hides all elements with `class="test"`
 - `$("#test").hide()` - hides the element with `id="test"`

jQuery: Document Ready Event

- Document Ready Event is used to prevent any jQuery code from running before the document is finished loading
- Good practice to wait for the document to be fully loaded and ready before manipulating it
- Document Ready Event allows for JavaScript code to be present in the HTML document head section
- Problems if JavaScript Statements run before document is fully loaded:
 - hide an element that is not created yet
 - get the size of an image that is not loaded yet

jQuery: Document Ready Event

- Usage:

```
$(document).ready(function(){  
    // jQuery methods go here...  
});
```

OR

```
$(function(){  
    // jQuery methods go here...  
});
```

jQuery: Selectors

- jQuery uses CSS syntax to select elements
- jQuery selectors are used to find/select/query HTML elements based on:
 - id
 - classes
 - types
 - attributes
 - values of attributes
 - etc.
- jQuery selectors are based on the existing CSS selectors
- In addition, jQuery has custom selectors
- jQuery selectors start with the dollar sign and open parentheses and end with closed parenthesis
- Examples:
 - `$(this)` – select current element
 - `$("p")` – select all `<p>` elements
 - `$(".test")` – select all elements with `class="test"`
 - `$("#test")` - select the element with `id="test"`

jQuery: Selectors

| | |
|--|---|
| <code>\$("*")</code> | Selects all elements |
| <code>\$(this)</code> | Selects the current HTML element |
| <code>\$("p.intro")</code> | Selects all <code><p></code> elements with <code>class="intro"</code> |
| <code>\$("p:first")</code> | Selects the first <code><p></code> element |
| <code>\$("ul li:first")</code> | Selects the first <code></code> element of the first <code></code> |
| <code>\$("ul li:first-child")</code> | Selects the first <code></code> element of every <code></code> |
| <code>\$("[href]")</code> | Selects all elements with an <code>href</code> attribute |
| <code>\$("a[target='_blank']")</code> | Selects all <code><a></code> elements with a <code>target</code> attribute value equal to <code>"_blank"</code> |
| <code>\$("a[target!='_blank']")</code> | Selects all <code><a></code> elements with a <code>target</code> attribute value NOT equal to <code>"_blank"</code> |
| <code>\$(":button")</code> | Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code> |
| <code>\$("tr:even")</code> | Selects all even <code><tr></code> elements |
| <code>\$("tr:odd")</code> | Selects all odd <code><tr></code> elements |

jQuery: Selector Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>

<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

jQuery: Events

- jQuery can respond to events in an HTML page
- Most DOM events have an equivalent jQuery method
- Examples:

| Mouse Events | Keyboard Events | Form Events | Document/Window Events |
|--------------|-----------------|-------------|------------------------|
| click | keypress | submit | load |
| dblclick | keydown | change | resize |
| mouseenter | keyup | focus | scroll |
| mouseleave | | blur | unload |

jQuery: Event Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>

<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>

</body>
</html>
```

jQuery: Event Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("You entered p1!");
  },
  function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>

<p id="p1">This is a paragraph.</p>

</body>
</html>
```

jQuery: Page Effects – Hide & Show

- **Hide**

- hide HTML elements
- `$(selector).hide(speed,callback);`

- **Show**

- show HTML elements
- `$(selector).show(speed,callback);`

- **Toggle**

- toggle between showing and hiding HTML elements
- `$(selector).toggle(speed,callback);`

- Most jQuery page effect methods have the following optional parameters:

- *speed*: optional speed parameter specifies the duration of the effect with the following values:
 - Slow
 - normal
 - fast
 - numeric value (milliseconds)
- *callback*: optional callback parameter is an additional function to be executed after the original jQuery method completes

jQuery: Page Effects – Hide/Show

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"><script>
$(document).ready(function(){
  $("#hide").click(function(){
    $("p").hide();
  });
  $("#show").click(function(){
    $("p").show();
  });
});
</script>
</head>
<body>
<p>If you click on the "Hide" button, I will disappear.</p>
<button id="hide">Hide</button>
<button id="show">Show</button>
</body>
</html>
```

jQuery: Page Effects – Hide/Show

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").toggle();
  });
});
</script>
</head>
<body>

<button>Toggle</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
</body>
</html>
```

jQuery: Page Effects - Fading

- Fading
 - jQuery fading methods can fade an element in and out of visibility
 - jQuery has the following fade methods:
 - fadeOut()
 - fade out an visible element
 - `$(selector).fadeOut(speed,callback);`
 - fadeIn()
 - fade in a hidden element
 - `$(selector).fadeIn(speed,callback);`
 - fadeToggle()
 - toggles between fading in and fading out
 - `$(selector).fadeToggle(speed,callback);`
 - fadeTo()
 - fade an element to a given opacity
 - opacity value between 0 and 1
 - `$(selector).fadeTo(speed,opacity,callback);`

jQuery: Page Effects - Fading

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
  });
});
</script>
</head>

<body>
<p>Demonstrate fadeIn() with different parameters.</p>
<button>Click to fade in boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>

</body>
</html>
```

jQuery: Page Effects - Fading

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"> </script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeTo("slow",0.15);
    $("#div2").fadeTo("slow",0.4);
    $("#div3").fadeTo("slow",0.7);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeTo() with different parameters.</p>
<button>Click to fade boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

jQuery: Page Effects - Sliding

- jQuery slide methods slides elements up and down
- slideDown() method: slide down an element
 - `$(selector).slideDown(speed,callback);`
- slideUp() method: slide up an element
 - `$(selector).slideUp(speed,callback);`
- slideToggle() method: toggle between slide down/up for an element
 - `$(selector).slideToggle(speed,callback);`

jQuery: Page Effects - Sliding

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideToggle("slow");
  });
});
</script>

<style type="text/css">
#panel,#flip
{
padding:5px; text-align:center; background-color:#e5e5cc; border:solid 1px #c3c3c3;
}
#panel
{
padding:50px; display:none;
}
</style>
</head>
<body>

<div id="flip">Click to slide the panel down or up</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

jQuery: Page Effects - Animations

- jQuery `animate()` method provides support to create custom animations
 - `$(selector).animate(css_params,speed,callback);`
 - *css_params*: mandatory parameter that specifies the CSS properties to be animated
 - *speed*: optional speed parameter specifies the duration of the effect with the following values:
 - slow
 - normal
 - fast
 - numeric value (milliseconds)
 - *callback*: optional callback parameter is an additional function to be executed after the original jQuery method completes
 - multiple properties can be animated at the same time
 - CSS property values to be modified can be:
 - absolute
 - relative to element's current value
 - predefined options such as "show", "hide", or "toggle":
 - multiple `animate()` calls
 - jQuery creates queue of `animate()` calls, executing them in their order of appearance

jQuery: Page Effects - Animations

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({
      left:'250px',
      opacity:'0.5',
      height:'150px',
      width:'150px'
    });
  });
});
</script>
</head>

<body>
<button>Start Animation</button><br/><br/>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;">
</div>

</body>
</html>
```

jQuery: Page Effects – Callback Functions

- JavaScript statements are executed line by line
- However, with effects, the next line of code can be run even though the effect is not finished and hence cause errors
- Use a callback function to prevent errors
- Callback function is executed after the current effect is finished
- Example:

- Hide effect (without callback)

```
$("#button").click(function(){  
    $("#p").hide(1000);  
    alert("The paragraph is now hidden");  
});
```

alert box will be displayed before the hide effect is completed

- Hide effect (with callback)

```
$("#button").click(function(){  
    $("#p").hide("slow",function(){  
        alert("The paragraph is now hidden");  
    });  
});
```

jQuery: Page Effects - Chaining

- Nice if we can chain together several actions/methods for a same element into a single statement
- jQuery allows such chaining of actions/methods
- Avoids the overhead of the Selector running multiple times

jQuery: Page Effects – Chaining

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function()
{
  $("button").click(function(){
    $("#p1").css("color","red").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>

<p id="p1">jQuery is fun!!</p>
<button>Click me</button>

</body>
</html>
```

jQuery: HTML Methods - Get

- jQuery contains methods for DOM modification i.e. accessing and changing HTML elements and attributes
- jQuery get content methods:
 - `text()` - gets the text content of selected element(s)
 - `html()` - gets the content of selected element(s)
 - gets the HTML syntax/markup also
 - `val()` - gets the value of form field(s)
- jQuery get attribute method:
 - `attr()` – gets the attribute value(s)

jQuery: HTML Methods - Get

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    alert("Text: " + $("#test").text());
  });
  $("#btn2").click(function(){
    alert("HTML: " + $("#test").html());
  });
});
</script>
</head>

<body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
</body>
</html>
```

jQuery: HTML Methods - Get

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Value: " + $("#test").val());
    });
});
</script>
</head>

<body>
<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
<button>Show Value</button>
</body>
</html>
```

jQuery: HTML Methods - Get

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#utd").attr("href"));
    });
});
</script>
</head>

<body>
<p><a href="http://www.utdallas.edu" id="utd">UT Dallas</a></p>
<button>Show href Value</button>
</body>
</html>
```


jQuery: HTML Methods - Set

- jQuery set content methods:
 - text() - sets the text content of selected element(s)
 - html() - sets the content of selected element(s)
 - sets the HTML syntax/markup also
 - val() - sets the value of form field(s)
- jQuery set attribute method:
 - attr() – sets the attribute value(s)

jQuery: HTML Methods - Set

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("#test1").text("Hello world!");
  });
  $("#btn2").click(function(){
    $("#test2").html("<b>Hello world!</b>");
  });
  $("#btn3").click(function(){
    $("#test3").val("Dolly Duck");
  });
});
</script>
</head>

<body>
<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>
<p>Input field: <input type="text" id="test3" value="Mickey Mouse"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body>
</html>
```

jQuery: HTML Methods - Set

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#utd").attr("href","http://cs.utdallas.edu/");
    });
});
</script>
</head>

<body>
<p><a href="http://www.utdallas.edu" id="utd">UTDallas</a></p>
<button>Change href value to UTD CS department</button>
</body>
</html>
```

jQuery: HTML Methods - Add

- The following are the jQuery methods to add new content:
 - `append()` - Inserts content at the end of the selected element(s)
 - `$("#p").append("Some appended text.");`
 - `prepend()` - Inserts content at the beginning of the selected element(s)
 - `$("#p").prepend("Some prepended text.");`
 - `after()` - Inserts content after the selected element(s)
 - `$("#img").after("Some text after");`
 - `before()` - Inserts content before the selected element(s)
 - `$("#img").before("Some text before");`

jQuery: HTML Methods - Remove

- The following two jQuery methods are used to remove elements and content:
 - `remove()` - Removes the selected element and its child elements (if any)
 - Example: `$("#div1").remove();`
 - one optional parameter to select the elements to be removed
 - parameter value can be any of the jQuery selector
 - Example: removes all `<p>` elements with `class="italic"`:
`$("p").remove(".italic");`
 - `empty()` - Removes the child elements (if any) from the selected element
 - `$("#div1").empty();`

jQuery: HTML Methods - Remove

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").remove();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>

</div>
<br>
<button>Remove div element</button>

</body>
</html>
```

jQuery: HTML Methods - Remove

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".italic");
  });
});
</script>
</head>
<body>

<p>This is a paragraph in the div.</p>
<p class="italic"><i>This is another paragraph in the div.</i></p>
<p class="italic"><i>This is another paragraph in the div.</i></p>
<button>Remove all p elements with class="italic"</button>

</body>
</html>
```

jQuery: HTML Methods - Remove

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").empty();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>

</div>
<br>
<button>Empty the div element</button>

</body>
</html>
```


jQuery: HTML Methods - CSS

- The following jQuery methods can be used for CSS manipulation:
 - `addClass()` - Adds one or more classes to the selected element(s)
 - `$("div").addClass("important");`
 - `removeClass()` - Removes one or more classes from the selected element(s)
 - `$("h1,h2,p").removeClass("blue");`
 - `toggleClass()` - Toggles between adding/removing classes for the selected element(s)
 - `$("h1,h2,p").toggleClass("blue");`
 - `css()` - Sets or returns the style attribute for the selected element(s)
 - `$("p").css("background-color");`

jQuery: HTML Methods - CSS

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.
3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1,h2,p").addClass("blue");
    $("div").addClass("important blue");
  });
});
</script>
<style type="text/css">
.important
{
font-weight:bold; font-size:xx-large;
}
.blue
{
color:blue;
}
</style>
```

```
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<div>This is some important text!</div>
<br>
<button>Add classes to elements</button>

</body>
</html>
```

jQuery: HTML Methods - CSS

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3
.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1,h2,p").removeClass("blue");
  });
});
</script>
<style type="text/css">
.important
{
font-weight:bold; font-size:xx-large;
}
.blue
{
color:blue;
}
</style>
</head>
```

```
<body>

<h1 class="blue">Heading 1</h1>
<h2 class="blue">Heading 2</h2>
<p class="blue">This is a paragraph.</p>
<p>This is another paragraph.</p>
<br>
<button>Remove class from elements</button>
</body>
</html>
```

jQuery: HTML Methods - CSS

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1,h2,p").toggleClass("blue");
  });
});
</script>
<style type="text/css">
.blue
{
color:blue;
}
</style>
</head>
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Toggle class</button>
</body>
</html>
```

jQuery: DOM Traversing

- jQuery has methods to traverse through the DOM element hierarchy and find/select HTML elements
- jQuery traversing methods can easily move up (to ancestors), down (to descendants) and sideways (to siblings) in the DOM tree

jQuery: DOM Traversing - Ancestors

- The following jQuery methods can be used to traverse up the DOM tree:
 - `parent()`
 - returns the direct parent element of the selected element
 - `parents()`
 - returns all ancestor elements of the selected element all the way up to the document's root element (`<html>`)
 - an optional parameter to filter the search for ancestors
 - `parentsUntil()`
 - returns all ancestor elements between the two given arguments

jQuery: DOM Traversing - Ancestors

```
<!DOCTYPE html>
<html>
<head>
<style>
.ancestors *
{
display: block;border: 2px solid lightgrey;color: lightgrey;padding: 5px;margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("span").parentsUntil("div").css({"color":"red","border":"2px solid red"});
});
</script>
</head>
<body class="ancestors"> body (great-great-grandparent)
  <div style="width:500px;">div (great-grandparent)
    <ul>ul (grandparent)
      <li>li (direct parent)
        <span>span</span>
      </li>
    </ul>
  </div>
</body>
</html>
```

jQuery: DOM Traversing - Descendants

- The following jQuery methods can be used to traverse down the DOM tree:
 - `children()`
 - returns all direct children of the selected element
 - an optional parameter to filter the search for children
 - `find()`
 - returns descendant elements of the selected element, all the way down to the last descendant

jQuery: DOM Traversing - Descendants

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants *
{
display: block; border: 2px solid lightgrey; color: lightgrey; padding: 5px; margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("div").children("p.1").css({"color":"red","border":"2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p class="1">p (child)
    <span>span (grandchild)</span>
  </p>
  <p class="2">p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

jQuery: DOM Traversing - Descendants

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants *
{
display: block;border: 2px solid lightgrey;color: lightgrey;padding: 5px;margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("div").find("span").css({"color":"red","border":"2px solid red"});
});
</script>
</head>
<body>

<div class="descendants" style="width:500px;">div (current element)
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
  <p>p (child)
    <span>span (grandchild)</span>
  </p>
</div>

</body>
</html>
```

jQuery: DOM Traversing - Siblings

- The following jQuery methods can be used to traverse sideways on the DOM tree:
 - `siblings()`
 - returns all sibling elements of the selected element
 - an optional parameter to filter the search for siblings
 - `next()`
 - returns the next sibling element of the selected element
 - `nextAll()`
 - returns all next sibling elements of the selected element
 - `nextUntil()`
 - returns all next sibling elements between two given arguments
 - `prev()`
 - returns the previous sibling element of the selected element
 - `prevAll()`
 - returns all previous sibling elements of the selected element
 - `prevUntil()`
 - returns all previous sibling elements between two given arguments

jQuery: DOM Traversing - Siblings

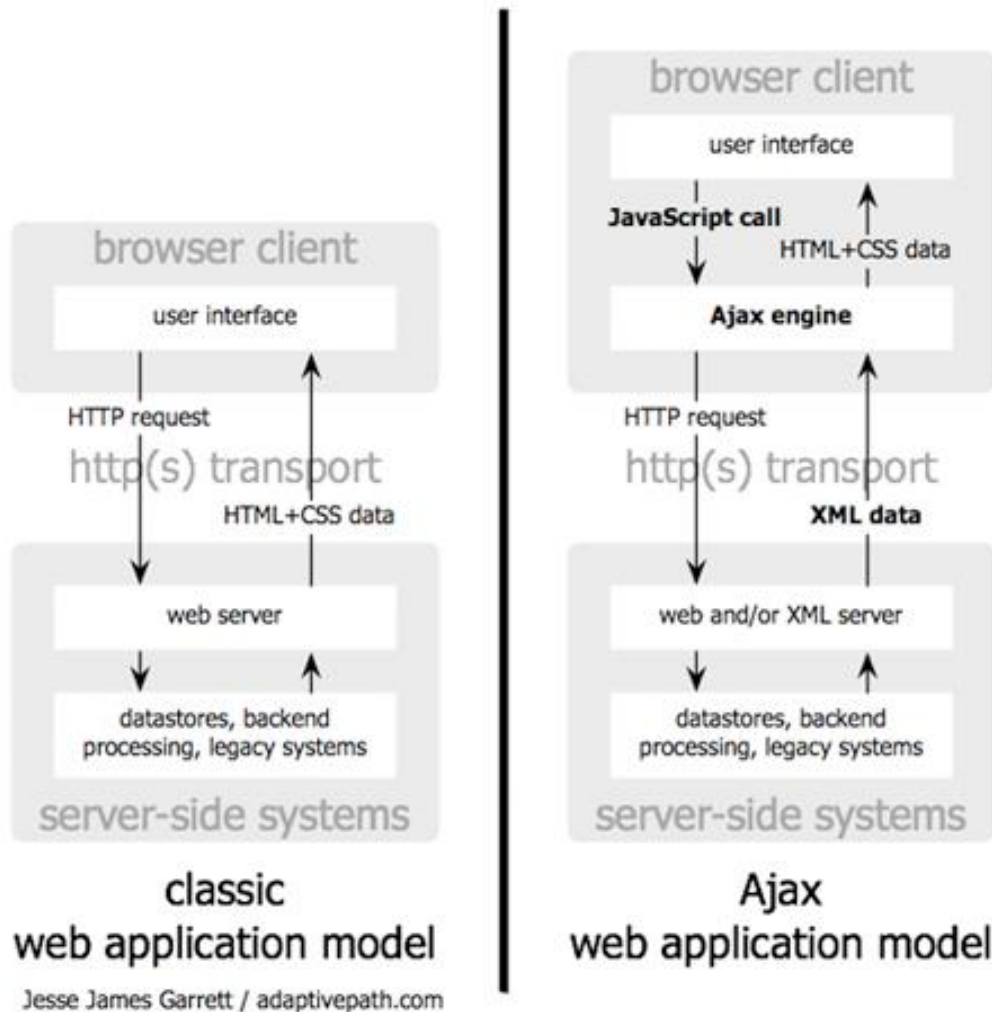
```
<!DOCTYPE html>
<html>
<head>
<style>
.siblings *
{
display: block;border: 2px solid lightgrey;color: lightgrey;padding: 5px;margin: 15px;
}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("h2").nextUntil("h6").css({"color":"red","border":"2px solid red"});
});
</script>
</head>
<body class="siblings">

<div>div (parent)
  <p>p</p>
  <span>span</span>
  <h2>h2</h2> <h3>h3</h3> <h4>h4</h4> <h5>h5</h5> <h6>h6</h6>
  <p>p</p>
</div>
</body>
</html>
```

AJAX: Brief Introduction

- AJAX: Asynchronous JavaScript + XML
- AJAX allows client-side JavaScript to request and receive data from a server without refreshing the web page.
- This technique enables the developer to create an application that is uninterrupted, making only portions of the web page reload with the new data.
- Data can be retrieved using the XMLHttpRequest object

AJAX: Brief Introduction



AJAX Examples

- Google Maps
- Google Suggest