

# Web Application Development

jQuery

# Outline

- Concepts of jQuery
- jQuery syntax
- jQuery events
- jQuery effects
- jQuery DOM manipulation
- jQuery AJAX

# Introduction

- jQuery is a lightweight JavaScript library
- jQuery wraps many complicated JavaScript codes into simple ones
- jQuery simplifies the use of JavaScript to develop the websites

# Introduction

The main uses of jQuery on the websites include:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX methods

# Installing jQuery

- jQuery can be added to the webpages using off-line method or on-line method
- In off-line method, jQuery source file is downloaded from [jQuery.com](http://jquery.com) and added into the HTML `<script>` tag

```
<head>  
  <script src="jquery-1.12.4.min.js">  
  </script>  
</head>
```

# Installing jQuery

- In on-line method, jQuery is included from a Content Delivery Network (CDN). Both Google and Microsoft host jQuery
- **Include jQuery from Google CDN:**

```
<head>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/  
1.12.4/jquery.min.js"></script>  
</head>
```
- **Include jQuery from Microsoft CDN:**

```
<head>  
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/  
jquery-1.12.4.min.js"></script>  
</head>
```

# jQuery Syntax

- jQuery syntax allows to select HTML elements and perform actions on these elements
- Basic syntax is:
  - `$(selector).action()`
  - `$` sign allows to define/access jQuery. It is a shortcut for jQuery
  - *selector* allows to find/query HTML elements
  - *action()* allows to perform some actions on these HTML elements

# Examples of jQuery Syntax

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



# jQuery Selectors

- jQuery selectors allow to select and manipulate HTML elements
- Selectors can be id, classes, types, attributes, values of attributes, etc.
- Most jQuery selectors are based on existing CSS selectors
- Selector syntax: the dollar sign with parentheses: `$()`

# Element Selector

In jQuery, element selector selects elements based on **element name**

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("p").hide();  
    });  
});
```

```
<html>  
...  
    <p>Hello</p>  
    <button>Click me</button>  
...  
</html>
```

# ID Selector

In jQuery, id selector selects elements based on **id attribute**

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("#p1").hide();  
    });  
});
```

```
<html>  
...  
    <p>jQuery Introduction</p>  
    <p id="p1">Hello</p>  
    <button>Click me</button>  
...  
</html>
```

# Class Selector

In jQuery, class selector selects elements based on **class name**

```
$(document).ready(function() {  
    $("button").click(function() {  
        $(".intro").hide();  
    });  
});
```

```
<html>  
...  
    <p class="intro">jQuery Introduction</p>  
    <p class="intro">Hello</p>  
    <button>Click me</button>  
...  
</html>
```

# Popular jQuery Selectors

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code>&lt;p&gt;</code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code>&lt;p&gt;</code> element
<code>\$("ul li:first")</code>	Selects the first <code>&lt;li&gt;</code> element of the first <code>&lt;ul&gt;</code>
<code>\$("ul li:first-child")</code>	Selects the first <code>&lt;li&gt;</code> element of every <code>&lt;ul&gt;</code>
<code>\$("[href]")</code>	Selects all elements with an <code>href</code> attribute

# Popular jQuery Selectors

Syntax	Description
<code>\$("a[target='_blank']")</code>	Selects all <a> elements with a target attribute value equal to "_blank"
<code>\$("a[target!='_blank']")</code>	Selects all <a> elements with a target attribute value NOT equal to "_blank"
<code>\$(":button")</code>	Selects all <button> elements and <input> elements of type="button"
<code>\$("tr:even")</code>	Selects all even <tr> elements
<code>\$("tr:odd")</code>	Selects all odd <tr> elements

# jQuery Events

- jQuery events allow to perform HTML events
- Most DOM events have an equivalent jQuery method
- Event syntax:  
`$(“selector”).jQueryEventMethod()`

# Example of jQuery Events

```
$(document).ready(function() {  
    $("button").click(function() {  
        alert("Hello ICT5 Bachelor Class!");  
    });  
});
```

```
<html>  
...  
    <p class="intro">jQuery Introduction</p>  
    <p class="intro">Hello</p>  
    <button>Click me</button>  
...  
</html>
```



# Popular Events

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 Effects

- jQuery provides a lot of effects for website function and design
- Examples of jQuery effects are hide/show HTML elements, fade elements in/out of visibility, slide elements up/down, or create custom animations, etc.

# Example of jQuery Effects

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("#div").animate({  
            left: '100px'  
        });  
    });  
});
```

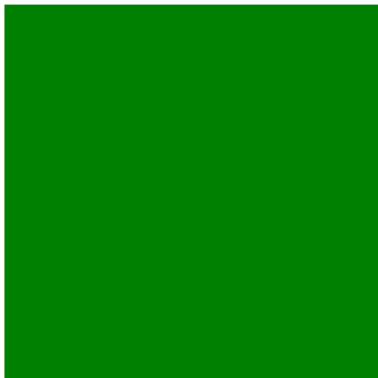
```
<html>  
...  
  <button>Start Animation</button>  
  <div style="background:green;height:100px;width:  
100px;position:absolute;"></div>  
...  
</html>
```

# Example of jQuery Effects

```
$(document).ready(function() {  
  $("button").click(function() {  
    $("#div").animate({  
      left: '100px'  
    });  
  });  
});
```

Before calling animate()

Start Animation



After calling animate()

Start Animation



# jQuery DOM Manipulation

- Manipulating DOM is an important capability of jQuery
- jQuery supports a bunch of DOM related methods for accessing and manipulating HTML elements and attributes

# Basic DOM Manipulation Functions

- Functions to **get web content**:
  - `text()`: gets the text content of selected elements
  - `html()`: gets the content with HTML tag of selected elements
  - `val()`: gets the value of form fields
  - `attr("attribute_name")`: gets attribute values of selected elements

# Basic DOM Manipulation Functions

- Functions to get web content:

```
$(document).ready(function(){
    $("button").click(function(){
        alert("Value: " + $("#p1").text());
    });
});
```

```
<html>
...
  <p>jQuery Introduction</p>
  <p id="p1">Hello</p>
  <button>Click me</button>
...
</html>
```

# Basic DOM Manipulation Functions

- Functions to **set web content**:
  - `text("set_value")`: sets the text content of selected elements
  - `html("set_value")`: sets the content with HTML tag of selected elements
  - `val("set_value")`: sets the value of form fields
  - `attr("attribute_name", "set_value")`: sets attribute values of selected elements



# Basic DOM manipulation functions

- Functions to set web content:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#p1").text("Hello Web Class");  
    });  
});
```

```
<html>  
...  
    <p>jQuery Introduction</p>  
    <p id="p1"></p>  
    <button>Click me</button>  
...  
</html>
```

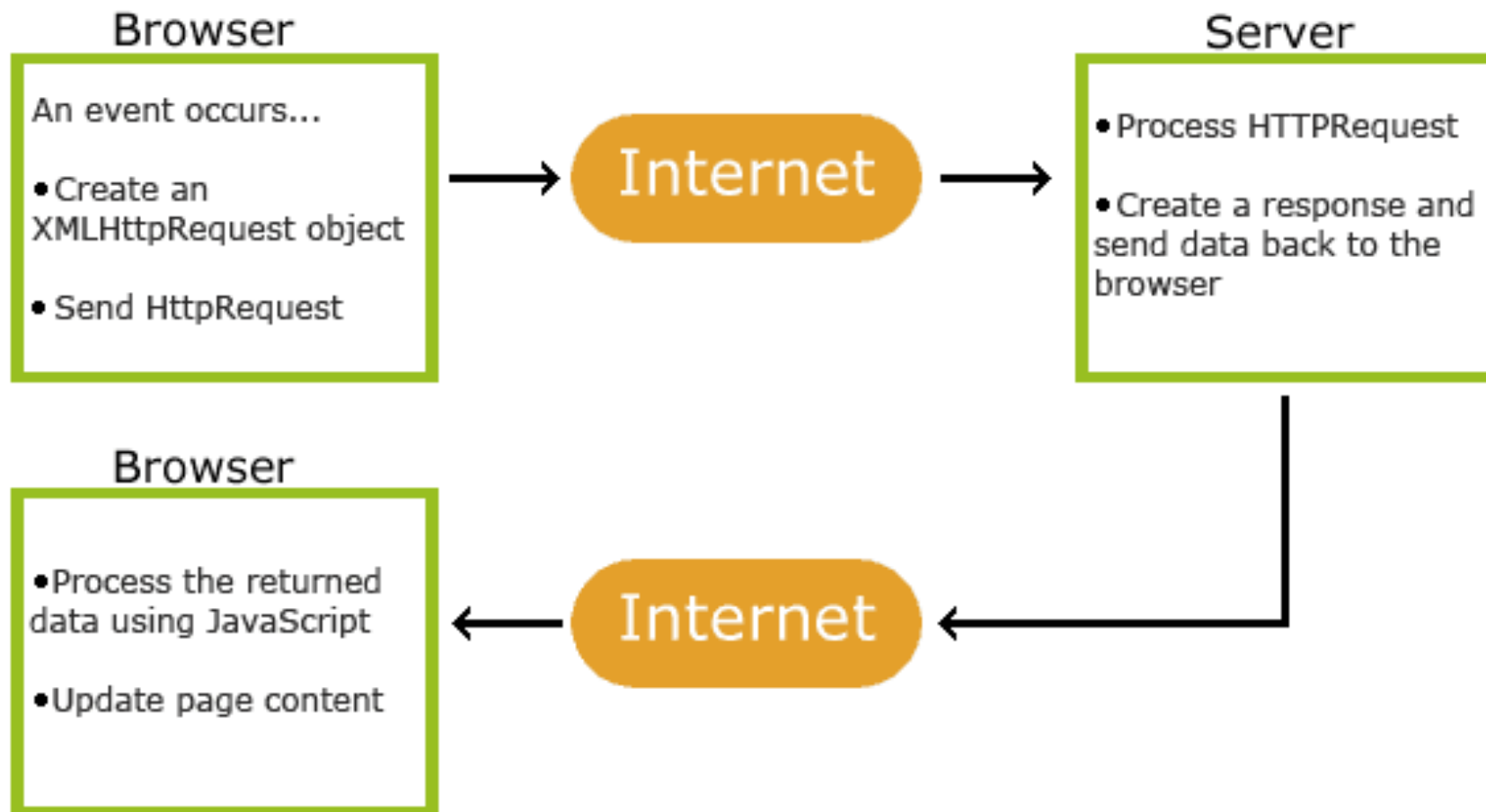
# jQuery AJAX

- jQuery AJAX is an important capability of jQuery which allows to simplify the communication between web client and web server

# What is AJAX?

- AJAX = Asynchronous JavaScript and XML
- AJAX is a technique for creating fast and dynamic web pages
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes This means that it is possible to update parts of a web page, without reloading the whole page
- Examples of applications using AJAX: Google Maps, Gmail, Youtube, and Facebook

# What is AJAX?



# jQuery AJAX

- jQuery AJAX methods allow to request text, HTML, XML, or JSON data from a remote server using both HTTP Get and HTTP Post requests
- External data can also be loaded directly into the selected HTML elements of the web page

# jQuery load() Method

- The load() method loads data from a server and puts the returned data into the selected element
- Syntax:
  - `$(selector).load(URL,data,callback);`
  - “URL” parameter specifies the source to load data
  - “data” parameter specifies query value to send along with the request
  - “callback” parameter is a function to be executed after the load() method is completed

# jQuery load() method

Having a “data.txt” file with the following contents:

```
<h2>Welcome to the class about jQuery!  
</h2>  
<p id="p1">jQuery is a very powerful  
JavaScript library.  
</p>
```

# jQuery load() method

Use load() method to load data from “data.txt” file:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#div1").load("data.txt");  
    });  
});
```

```
<html>  
...  
    <div id="div1"></div>  
    <button>Load data file</button>  
...  
</html>
```



# jQuery load() method

- Use load() method to load data from “data.txt” file:

```
$(document).ready(function(){  
    $("button").click(function(){  
        $("#div1").load("data.txt");  
    });  
});
```

- Loaded Results:

Welcome to the class about jQuery!  
jQuery is a very powerful JavaScript library

# jQuery GET and POST methods

- Two commonly used methods for a request-response between a client and server are: GET and POST.
  - GET - Requests data from a specified resource
  - POST - Submits data to be processed to a specified resource
- Syntax:
  - `$.get(URL, callback);`
  - `$.post(URL, data, callback);`

# jQuery \$.get() method

Having an Active Server Page (get\_test.asp) file as follows:

```
<%  
response.write("Welcome to the class  
about get and post methods using jQuery")  
%>
```

# jQuery \$.get() method

- Use \$.get() method to retrieve data from “get\_test.asp” file on the server:

```
$(document).ready(function(){
    $("button").click(function(){
        $.get("get_test.asp", function(data, status){
            alert("Data: " + data + "\nStatus: " + status);
        });
    });
});
```

```
<html>
...
    <button>Get content of asp file</button>
...
</html>
```

# jQuery \$.get() method

- Use \$.get() method to retrieve data from “get\_test.asp” file on the server:

```
$(document).ready(function(){
    $("button").click(function(){
        $.get("get_test.asp", function(data, status){
            alert("Data: " + data + "\nStatus: " + status);
        });
    });
});
```

- Response Results:

```
Data: Welcome to the class about get and post methods
using jQuery
Status: success
```

# jQuery \$.post() method

Having an Active Server Page (post\_test.asp) file as follows:

```
<%  
dim name  
name=Request.Form("name")  
Response.Write("Hello " & name)  
%>
```

# jQuery \$.post() method

Use \$.post() method to send some data along with the request to server:

```
$(document).ready(function(){
    $("button").click(function(){
        $.post("post_test.asp",
            {
                name: "John Smitt"
            },
            function(data,status){
                alert("Data: " + data + "\nStatus: " + status);
            });
    });
});
```

```
<body>
    <button>Post content of asp file</button>
</body>
```

# jQuery \$.post() method

- Use \$.post() method to send some data along with the request to server:

```
$(document).ready(function(){
    $("button").click(function(){
        $.post("post_test.asp",
            {
                name: "John Smitt"
            },
            function(data,status){
                alert("Data: " + data + "\nStatus: " + status);
            });
    });
});
```

- **Response Results:**

```
Data: Hello John Smitt
Status: success
```



