

# Lab 11. JavaScript 04

인터넷과웹기초



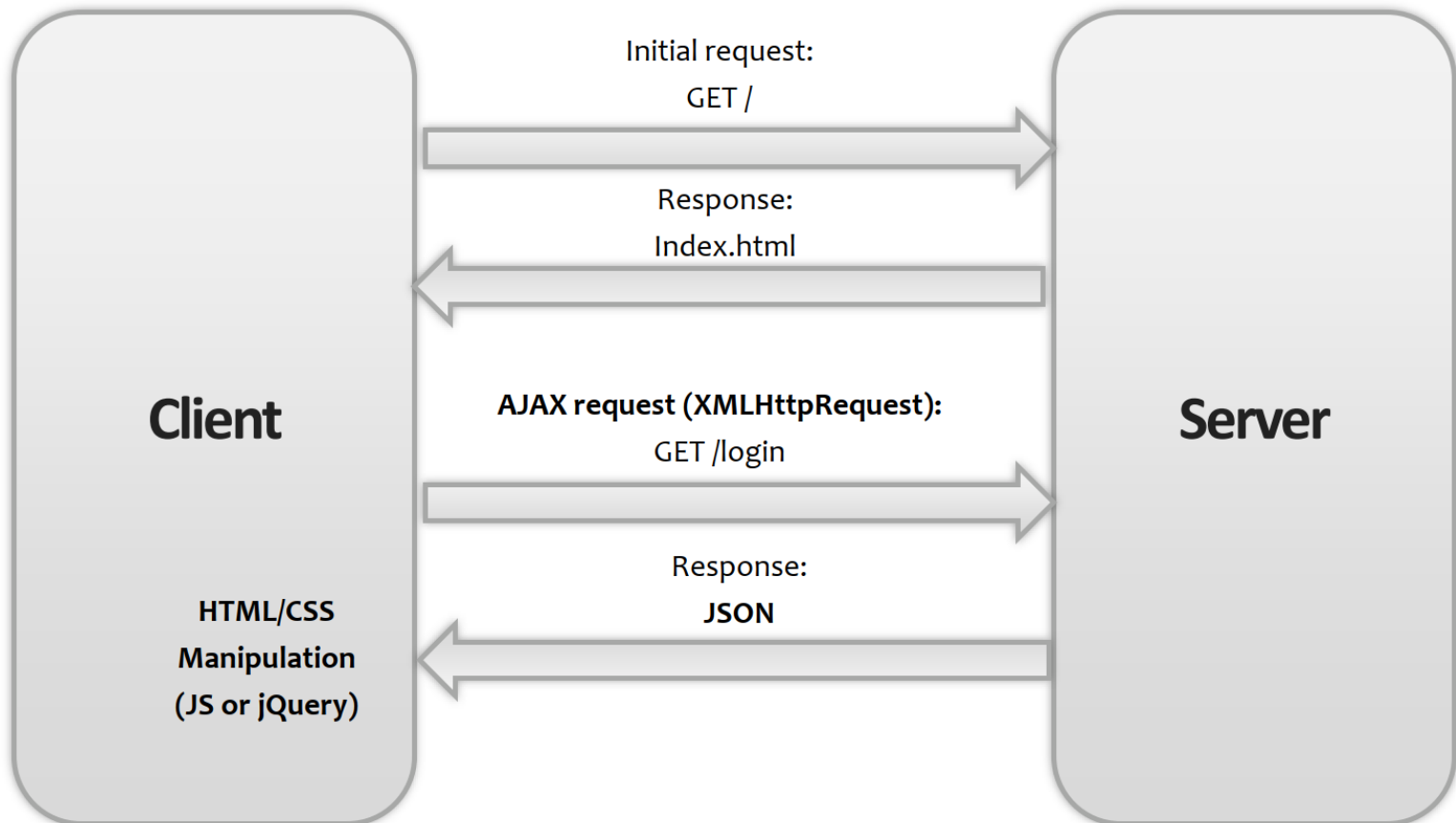
# AJAX

## XML / JSON



# AJAX

- Asynchronous Javascript And XML
  - to request to the server without reloading the web page.
  - to receive and work with data from the server



# AJAX

- request with XMLHttpRequest object

ajax.html

```
<!DOCTYPE html>
<html>
  <body>
    <h2>The XMLHttpRequest Object</h2>

    <p id="demo">Let AJAX change this text.</p>

    <button type="button" onclick="loadDoc()">Click</button>

    <script>
      function loadDoc() {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function () {
          if (this.readyState == 4 && this.status == 200) {
            document.getElementById("demo").innerHTML = this.responseText;
          }
        };
        xhttp.open("GET", "ajax_info.txt", true);
        xhttp.send();
      }
    </script>
  </body>
</html>
```

# fetch API (참고)

- Request with fetch API (ES6)
  - [https://developer.mozilla.org/ko/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/ko/docs/Web/API/Fetch_API/Using_Fetch)
  - check the log in console. → what is promise?

ajax.html

```
<!DOCTYPE html>
<html>
  <body>
    <h2>The XMLHttpRequest Object</h2>

    <p id="demo">Let AJAX change this text.</p>

    <button type="button" onclick="loadDoc()">Click</button>

    <script>
      function loadDoc() {
        fetch('ajax_info.txt')
          .then((response) => {
            res = response.text();
            console.log(res);
            document.getElementById("demo").innerHTML = this.res;
          })
      }
    </script>
  </body>
</html>
```

# XML

- eXtensible Markup Language

menu.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
  <food>
    <name>Belgian Waffles</name>
    <price>$5.95</price>
    <description>
      Two of our famous Belgian Waffles with plenty of real maple syrup
    </description>
    <calories>650</calories>
  </food>
  <food>
    <name>Strawberry Belgian Waffles</name>
    <price>$7.95</price>
    <description>
      Light Belgian waffles covered with strawberries and whipped cream
    </description>
    <calories>900</calories>
  </food>
</breakfast_menu>
```

# parse XML

- XML parsing using responseXML

ajax\_xml.html

```
<!DOCTYPE html>
<html>
  <body>
    <h2>The XMLHttpRequest Object</h2>
    <p id="demo"></p>

    <script>
      const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
        const xmlDoc = xhttp.responseXML;
        console.log(xmlDoc)
        const x = xmlDoc.getElementsByTagName("name");
        let txt = "";
        for (let i = 0; i < x.length; i++) {
          txt = txt + x[i].childNodes[0].nodeValue + "<br>";
        }
        document.getElementById("demo").innerHTML = txt;
      }
      xhttp.open("GET", "menu.xml");
      xhttp.send();
    </script>
  </body>
</html>
```

# XML vs. JSON

- JavaScript Object Notation
  - It is more structured than xml.

## Input

```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
<food>
  <name>Belgian Waffles</name>
  <price>$5.95</price>
  <description>
Two of our famous Belgian Waffles with plenty of real maple syrup
</description>
  <calories>650</calories>
</food>
<food>
  <name>Strawberry Belgian Waffles</name>
  <price>$7.95</price>
  <description>
Light Belgian waffles covered with strawberries and whipped cream
</description>
  <calories>900</calories>
</food>
<food>
  <name>Berry-Berry Belgian Waffles</name>
  <price>$8.95</price>
  <description>
Belgian waffles covered with assorted fresh berries and whipped cream
```

## Output

```
{
  "breakfast_menu": {
    "food": [
      {
        "name": "Belgian Waffles",
        "price": "$5.95",
        "description": "\n  Two of our famous Belgian Waffles with plenty of real maple syrup\n  ",
        "calories": "650"
      },
      {
        "name": "Strawberry Belgian Waffles",
        "price": "$7.95",
        "description": "\n  Light Belgian waffles covered with strawberries and whipped cream\n  ",
        "calories": "900"
      },
      {
        "name": "Berry-Berry Belgian Waffles",
        "price": "$8.95",
        "description": "\n  Belgian waffles covered with assorted fresh berries and whipped cream\n  ",
        "calories": "900"
      },
      {
        "name": "French Toast"
```

<https://www.utilities-online.info/xmltojson>



# parse JSON

- JSON parsing using JSON.parse()

ajax\_json.html

```
<!DOCTYPE html>
<html>
  <body>
    <h2>The XMLHttpRequest Object</h2>
    <p id="demo"></p>

    <script>
      const xhttp = new XMLHttpRequest();
      xhttp.onload = function() {
        const jsonDoc = xhttp.responseText;
        console.log(jsonDoc);
        var obj = JSON.parse(jsonDoc);
        console.log(obj);
        let txt = "";
        for (let i = 0; i < obj.breakfast_menu.food.length; i++) {
          txt = txt + obj.breakfast_menu.food[i].name + "<br>";
        }
        document.getElementById("demo").innerHTML = txt;
      }
      xhttp.open("GET", "menu.json");
      xhttp.send();
    </script>
  </body>
</html>
```

# jQuery



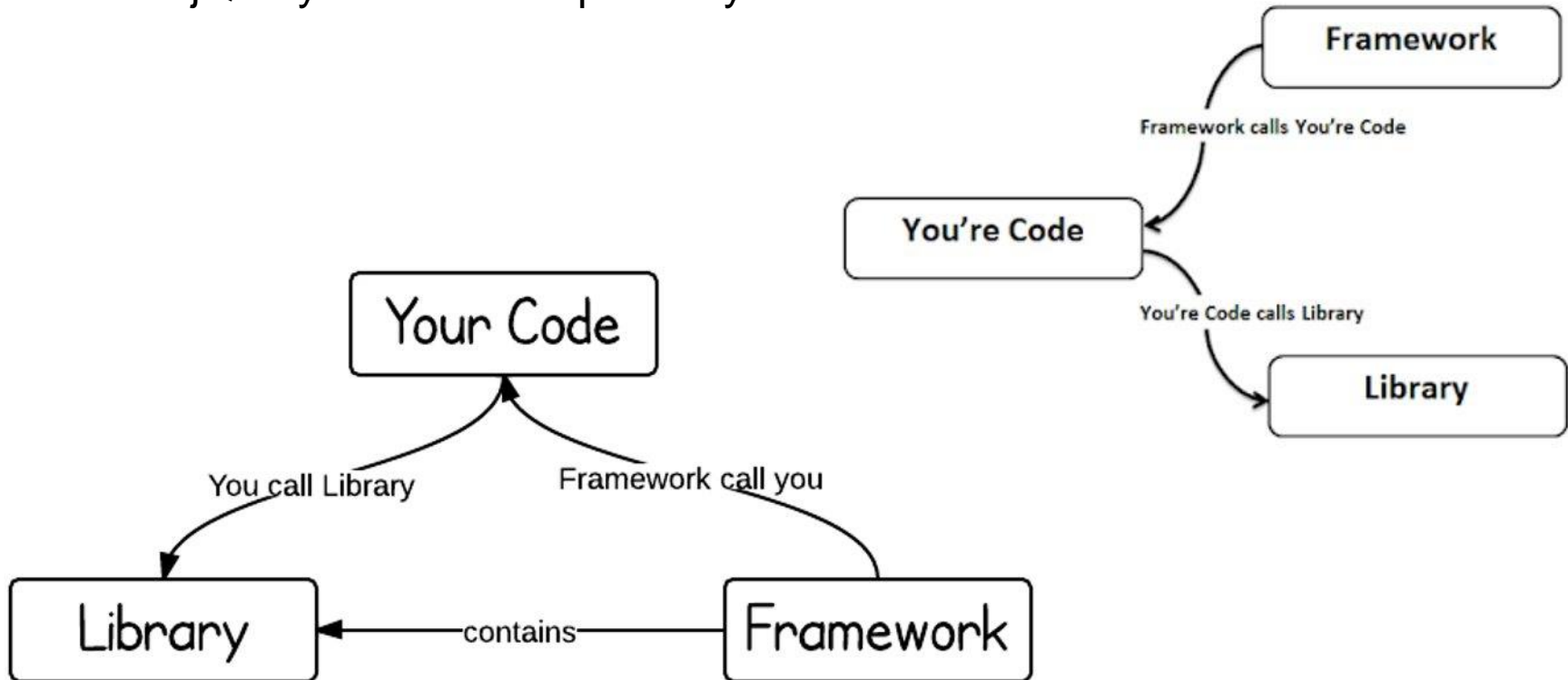
# What is jQuery

- History
  - In August 2005 jQuery founder John Resig was looking into how to better combine CSS selectors with succinct JavaScript notation.
  - jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

```
<body>
  <h3 id="one">something</h3>
  <script>
    //Vanilla JavaScript
    document.getElementById('one').innerHTML = 'test';
    //jQuery library
    $('#one').html('test');
  </script>
</body>
```

# jQuery is a library

- Library vs. Framework
  - Both helps developer to code easily.
  - Framework is standard for coding in a way.
  - jQuery is a JavaScript library.



# How to use jQuery (1)

- Download jQuery and import it ([link](#))

## Downloading jQuery

Compressed and uncompressed copies of jQuery files are available. The uncompressed file is best used during development or debugging; the compressed file saves bandwidth and improves performance in production. You can also download a [sourcemap file](#) for use when debugging with a compressed file. The map file is *not* required for users to run jQuery, it just improves the developer's debugger experience. As of jQuery 1.11.0/2.1.0 the `//# sourceMappingURL` comment is [not included](#) in the compressed file.

To locally download these files, right-click the link and select "Save as..." from the menu.

## jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We also recommend using the [jQuery Migrate plugin](#).

[Download the compressed, production jQuery 3.6.0](#)

[Download the uncompressed, development jQuery 3.6.0](#)

[Download the map file for jQuery 3.6.0](#)

```
<script src="jquery-3.6.0.min.js"></script>
```

# How to use jQuery (2)

- Download jQuery with CDN ([link](#))
  - Recommend this way

## jQuery CDN – Latest Stable Versions

Powered by **STACKPATH**

### jQuery Core

Showing the latest stable release in each major branch. [See all versions of jQuery Core.](#)

#### jQuery 3.x

- jQuery Core 3.6.0 - [uncompressed](#), [minified](#), [slim](#), [slim minified](#)

#### jQuery 2.x

- jQuery Core 2.2.4 - [uncompressed](#), [minified](#)

#### jQuery 1.x

- jQuery Core 1.12.4 - [uncompressed](#), [minified](#)

### jQuery Migrate

- jQuery Migrate 3.3.2 - [uncompressed](#), [minified](#)

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"
integrity="sha256-/xUj+30JU5yExlq6GSYGS7k7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"> </script>
```

# jQuery APIs

- jQuery API Documentation
  - <https://api.jquery.com/>

## jQuery API

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. If you're new to jQuery, we recommend that you check out the [jQuery Learning Center](#).

If you're updating to a newer version of jQuery, be sure to read the release notes published on [our blog](#). If you're coming from a version prior 1.9, you should check out the [1.9 Upgrade Guide](#) as well.

Note that this is the API documentation for jQuery core. Other projects have API docs in other locations:

- [jQuery UI API docs](#)
- [jQuery Mobile API docs](#)
- QUnit API docs

### **.add()**

Traversing > Miscellaneous Traversing

Create a new jQuery object with elements added to the set of matched elements.

### **.addBack()**

Traversing > Miscellaneous Traversing

Add the previous set of elements on the stack to the current set, optionally filtered by a selector.

### **.addClass()**

Attributes | Manipulation > Class Attribute | CSS

Adds the specified class(es) to each element in the set of matched elements.

### **.after()**

Manipulation > DOM Insertion, Outside

Insert content, specified by the parameter, after each element in the set of matched elements.

# Basic Methods

- How to use jQuery's methods

```
$('CSS Selector').jQuery_Method('');
```

- 1. `$()` is function of jQuery
- 2. '**CSS Selector**' is passed as an argument to `$()` function
- 3. `$()` function returns jQuery object
- 4. This object has methods to control the element

- For example,

```
$('#one').html('test');
```

- Replacing the html code with 'test' of '#one' element
- `.html` is a method, jQuery object has various APIs

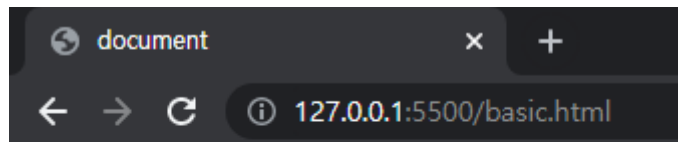


# Use simple methods

```
<body>

  <h3 id="one">first</h3>
  <h3 id="two">second</h3>

  <script>
    $('#one').html('it is not first');
    $('#two').css('color', 'blue');
  </script>
</body>
```



it is not first

second

# Node Control

- Methods to control the node in DOM
  - <https://api.jquery.com/category/manipulation/>

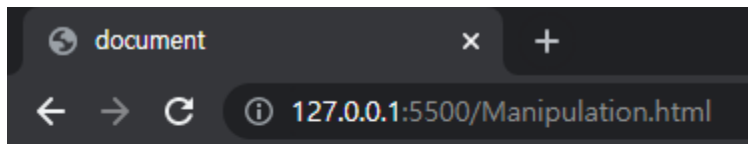
```
<body>

  <div class="box">content</div>

  <script>

    $('<div> before </div>');
    $('<div> prepend </div>');
    $('<div> append </div>');
    $('<div> after </div>');

  </script>
</body>
```



before  
prepend  
content  
append  
after

```
.. <body> == $0
  <div> before </div>
  <div class="box">
    <div> prepend </div>
    "content"
    <div> append </div>
  </div>
  <div> after </div>
```

# Event

- Method to Attach an event handler function
  - <https://api.jquery.com/on/>

```
<body>
  <input type="button" id="vanilla" value="Vanilla" /> <br><br>
  <input type="button" id="jquery" value="jQuery" />
  <script src="//code.jquery.com/jquery-3.5.1.min.js"></script>

  <script>
    // Vanilla JavaScript
    var target = document.getElementById('vanilla');
    if(target.addEventListener){
      target.addEventListener('click', function(event){
        alert('vanilla');
      });
    }
    else {
      target.attachEvent('onclick', function(event){
        alert('vanilla');
      });
    }

    // jQuery
    $('#jquery').on('click', function(event){
      alert('jQuery');
    })
  </script>
</body>
```

127.0.0.1:5500 내용:

vanilla

확인

127.0.0.1:5500 내용:

jQuery

확인

# jQuery and AJAX

ajax\_jquery.html

```
<!DOCTYPE html>
<html>
  <head>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    </head>
    <body>
      <h2>The XMLHttpRequest Object</h2>

      <p id="demo">Let AJAX change this text.</p>

      <button type="button">Click</button>

      <script>
        $(document).ready(function() {
          $("button").click(function() {
            $.get("ajax_info.txt", function(data) {
              $("#demo").html(data);
            })
          })
        })
      </script>
    </body>
  </html>
```