

# Dynamic Web Apps 2

제주대학교 2010. 4. 3.

# Review

## ◎ Function itself vs. function call

```
elem.onclick = menuClick; // function  
elem.onclick = menuClick(1); // null
```

- introducing anonymous function

```
elem.onclick = function menuClick1() {  
    menuClick(1);  
}
```

## ◎ Javascript beautifier

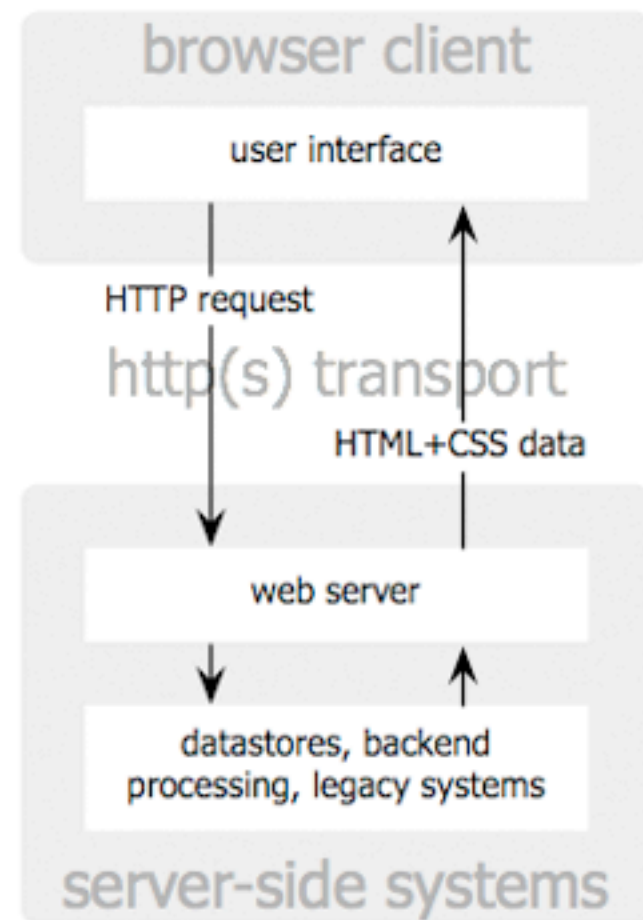
- <http://jsbeautifier.org/>

# Introduction of AJAX

## AJAX: Asynchronous Javascript + XML

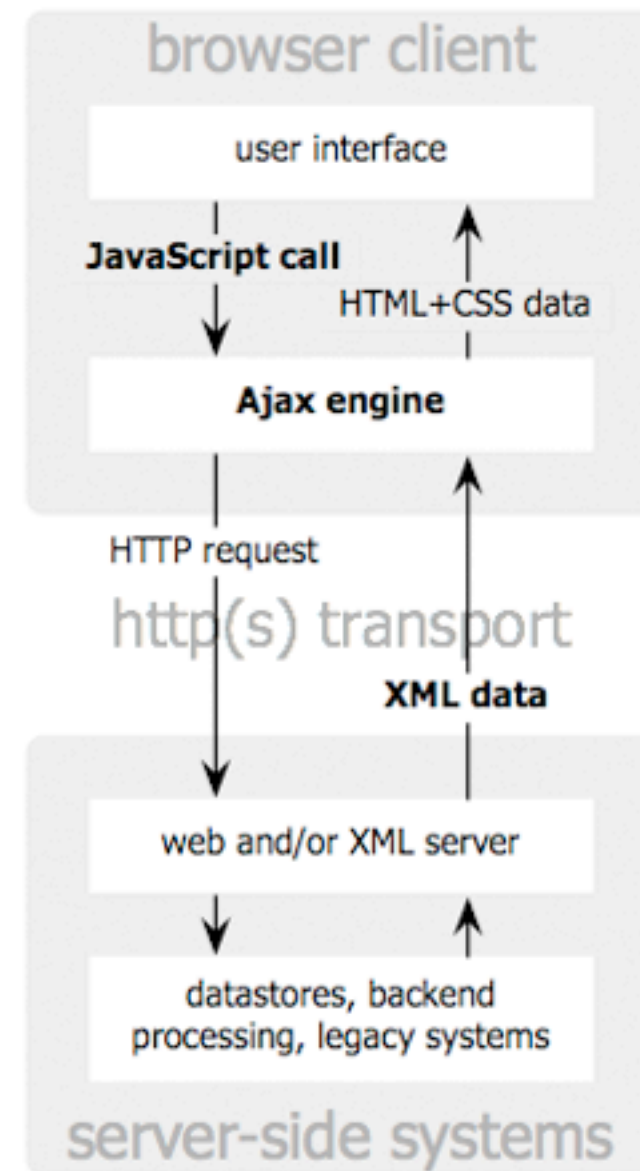
- ◎ 인터랙티브 웹 어플리케이션을 제작하기 위한 기술셋
- ◎ 화면 갱신 없이 서버와 통신 가능
  - 사용자 친화적인 어플리케이션을 만들 수 있음
  - 전체 페이지를 갱신하지 않기 때문에 서버 트래픽을 줄일 수 있음
- ◎ 넓게는 XHTML, CSS, DOM, Javascript, XHR, XML, Jason 등을 사용하여 웹 어플리케이션을 만드는 기법을 총칭함
- ◎ Gmail, Google docs, Spring note, Search keyword suggestion, etc.

# Introduction of AJAX



**classic**  
**web application model**

Jesse James Garrett / [adaptivepath.com](http://adaptivepath.com)



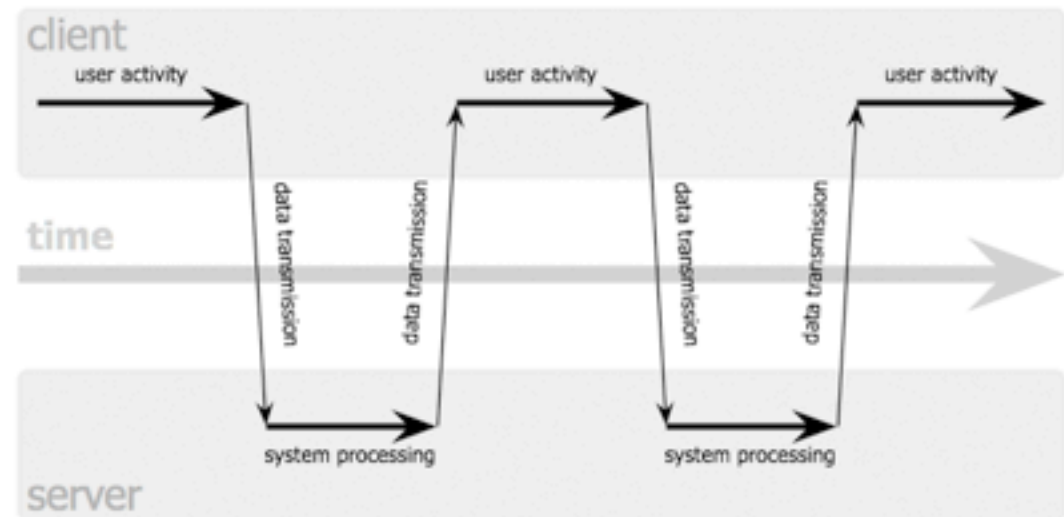
**Ajax**  
**web application model**

# Introduction of AJAX

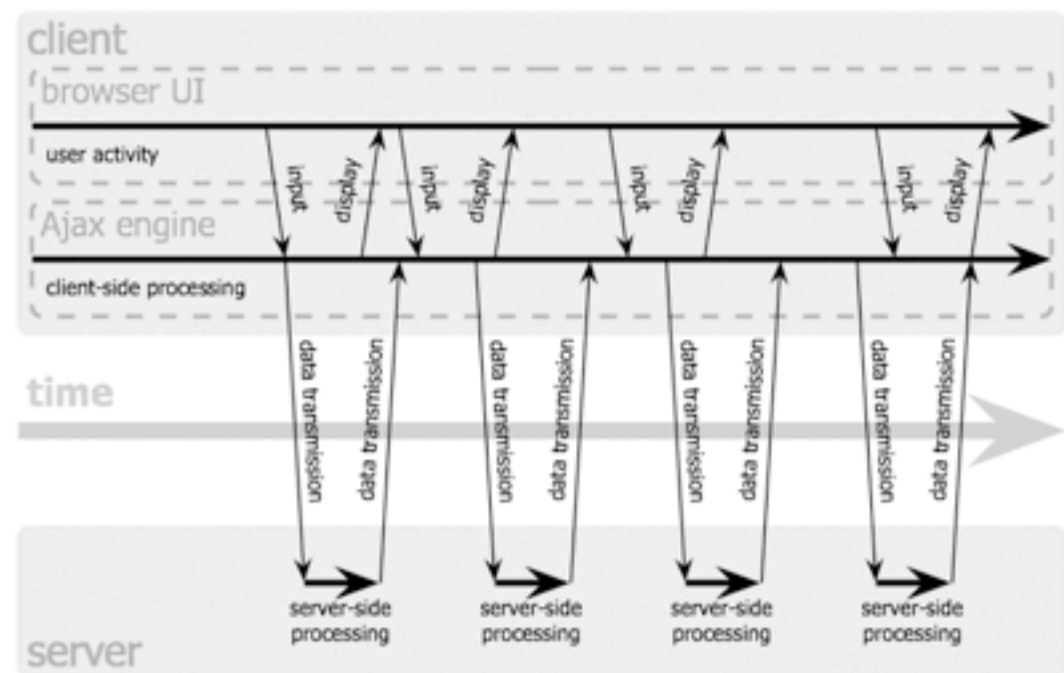
## Synchronous & Asynchronous

- 동기화: 상대방이 처리를 끝낼 때 까지 기다림
- 비동기화: 상대방의 처리 유무와 관계없이 인터랙션을 계속할 수 있음
  - 처리가 복잡
  - 데이터 무결성 체크가 중요함

classic web application model (synchronous)

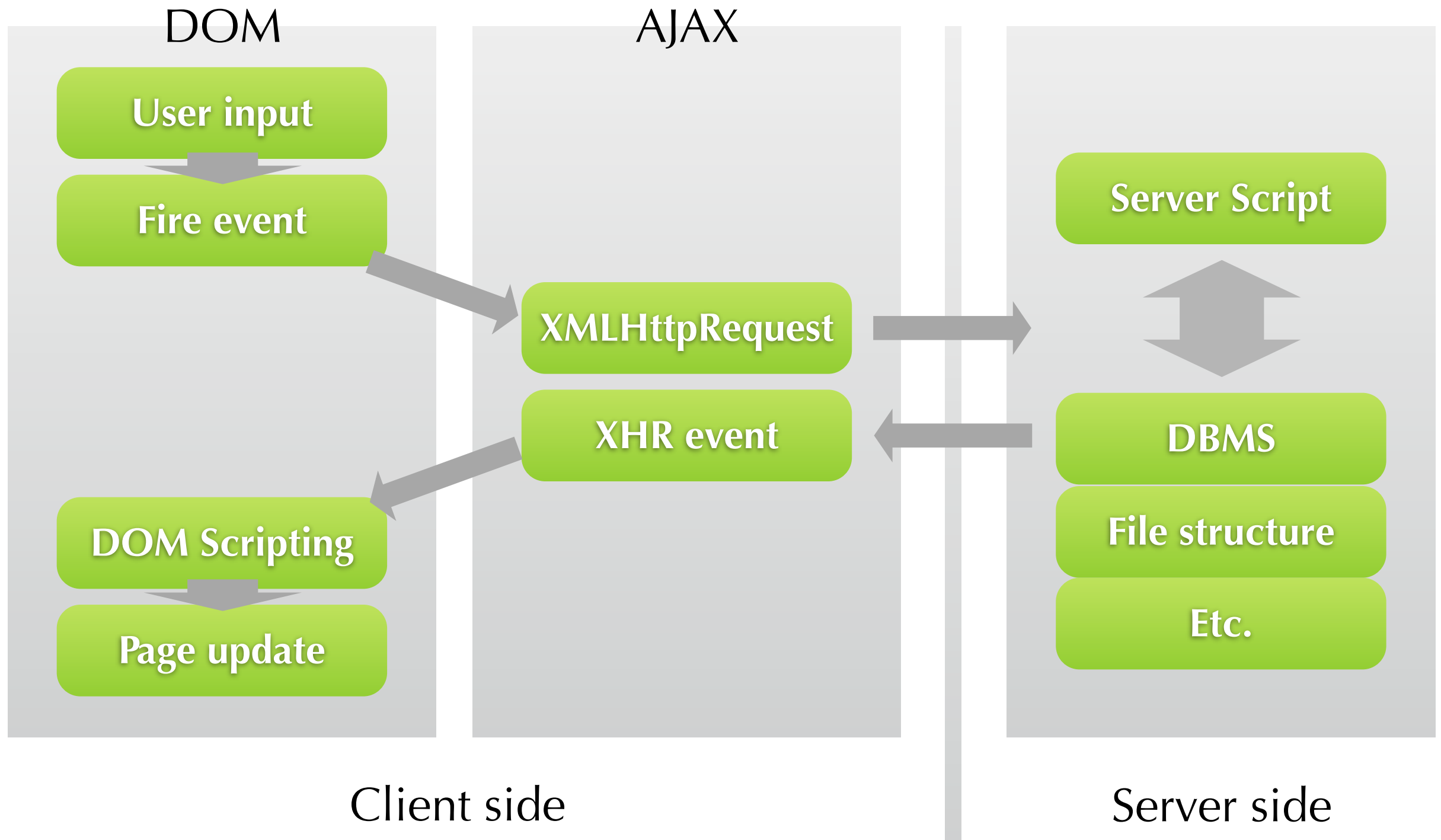


Ajax web application model (asynchronous)



Jesse James Garrett / adaptivepath.com

# Introduction of AJAX



# XHR object

## XMLHttpRequest

- ◎ open: XHR를 설정
- ◎ setRequestHeader: 요청에 대한 헤더 정보 설정
- ◎ send: 요청을 서버에 전송
- ◎ readyState, onreadystatechange: 요청에 대한 상태와 그 상태가 변경되었을 때 발생하는 이벤트
- ◎ responseText, responseXML: 텍스트, XML 형태의 서버 응답

# XHR object

## Cross domain policy

- ◎ file 프로토콜에서는 실행되지 않음
  - 로컬 파일 접근 방지
  - 웹서버에서만 실행 가능
- ◎ 서로 다른 도메인의 데이터를 불러올 수 없음
  - AJAX를 사용하는 파일과 데이터를 보내는 파일이 같은 서버(같은 도메인)에 있어야 함



# XHR object

## Make XMLHttpRequest

```
var req = new XMLHttpRequest();  
req.onreadystatechange = processReqChange;  
req.open("GET", url, true);  
req.send(null);
```

- ◎ XHR 객체 생성
- ◎ readyState가 변경될 때 실행될 함수 설정
  - 서버 응답시 실행될 기능을 지정
- ◎ 연결 설정
- ◎ 요청 전송

# XHR object

## POST and GET request

- ◎ POST로 요청을 보낼 경우 C-T을 전송함
- ◎ POST로 요청할 값을 value라고 할 때, 아래와 같이 GET과 POST를 구분하여 전송 요청

```
if (value) {  
    req.open("POST", url, true);  
    req.setRequestHeader('Content-Type',  
        'application/x-www-form-urlencoded');  
    req.send(value);  
} else {  
    req.open("GET", url, true);  
    req.send(null);  
}
```

# XHR object

## readyState property

- ◎ XMLHttpRequest의 상태를 나타냄
  - 0: not initialized
  - 1: set up
  - 2: sent
  - 3: in progress
  - 4: complete
- ◎ onreadystatechange를 사용하여 상태가 바뀔 때 처리할 함수를 지정

# XHR object

## onreadystatechange event

- ◎ readyState의 값이 변경될 때 기능을 수행
  - AJAX 요청 후 응답이 왔을 때 수행할 기능을 지정

```
function processReqChange() {  
    // only if req shows "loaded"  
    if (req.readyState == 4) {  
        // only if "OK"  
        if (req.status == 200) {  
            do_something();  
        } else {  
            alert("There was a problem retrieving the XML data:\n" +  
req.statusText);  
        }  
    }  
}
```

# XHR object

## Internet Explorer

### ◎ XHR 객체의 선언 방법이 다름

```
function loadXMLDoc(url, value) {  
    // branch for native XMLHttpRequest object  
    if (window.XMLHttpRequest) {  
        req = new XMLHttpRequest();  
        // ...  
    // branch for IE/Windows ActiveX version  
    } else if (window.ActiveXObject) {  
        req = new ActiveXObject("Microsoft.XMLHTTP");  
        // ...  
    }  
}
```

# XHR object

## **responseText, responseXML**

- ◎ 응답 데이터를 담고 있는 속성
  - XML 형식 외에 텍스트 형식도 사용 가능함
- ◎ XML + DOM: 다른 프로그램에서 인식이 가능한 형식
- ◎ JSON + eval(): JavaScript Object Notation, 자바스크립트에  
서 직접 인식 가능한 형식
- ◎ HTML + innerHTML: 웹페이지에 바로 사용할 수 있는 형식

# H/W

## 1부터 n까지의 합

### ◎ sigma.html, sigma.php

- 웹서버에 파일을 올린 후 sigma.html의 setXMLDoc, sigma 함수에 기능을 추가하여 1부터 n까지의 합을 구하는 기능을 완성한다.
  - php 서버가 아닌 경우 자신에게 익숙한 언어로 수정 해서 사용
  - HTML, CSS, 자바스크립트 중 setXMLDoc, sigma 외는 수정하지 않고 구현
- 프로그램 URL을 hyeonseok@gmail.com 으로 아래의 제목 형식으로 보내주세요.
  - 제주대학교 학과 이름 학번

# jQuery

## JavaScript library

- ◎ 기능과 UI가 많아질 수록 복잡도가 증가한다.
  - 재사용될 수 있는 기능들을 모아 라이브러리화 한다.
- ◎ 웹에서 많이 사용되는 기능을 라이브러리화 하는 프로젝트들이 많다.
  - Prototype, mooTools, YUI, Dojo, Ext JS, etc.
- ◎ 개발해야 하는 제품의 특성에 따라서 사용한다.
  - 직접 만들 수도 있고 공개 라이브러리를 사용할 수도 있다.
  - 효율성(크기, 기능, 시간 등)을 생각해 본다.



# jQuery

**jQuery** write less, do more

- ◎ 문법이 간결하다.
- ◎ 사용하기 쉽다.
- ◎ 빠르게 배울 수 있다.
- ◎ 다른 라이브러리들과 충돌을 일으키지 않는다.
- ◎ 다양한 플러그인이 존재한다.
- ◎ 브라우저 호환성 문제를 해결해준다.
- ◎ MIT and GPL licenses.

# jQuery

## Installation

- ◎ jquery.com에서 스크립트 파일을 다운로드하여 링크한다.

```
<script src="jquery.js"></script>  
<script>  
// your script goes here.  
</script>
```

- ◎ CDN(Content delivery network)
  - Google Ajax API CDN: <http://ajax.googleapis.com/ajax/libs/jquery/1.5.1/jquery.min.js>
  - Microsoft CDN: <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.5.2.min.js>
  - jQuery CDN: <http://code.jquery.com/jquery-1.5.2.min.js>

# jQuery

## jQuery(), \$()

- ◎ 요소를 참조하는 기본함수이다.
  - \$()로 축약해서 사용한다.
- ◎ CSS 선택자를 인자로 사용한다.

```
jQuery('#header') // id 선택자  
jQuery('article h1') // 하위 선택자  
$('aside section.notice li') // $()로 축약한 경우
```

- 다른 CSS2나 CSS3 선택자들도 풍부하게 지원한다.

# jQuery

## Chaining

- ◎ 대부분의 jQuery 메서드는 jQuery 객체를 반환한다.
  - 메서드를 연이어서 지정할 수 있다.

```
$('div.section').hide().addClass('gone');
```

- ◎ 다른 객체를 반환하는 경우 .end()로 이전 객체를 받을 수 있다.

```
$('ul.first').find('.foo')  
    .css('background-color', 'red')  
.end().find('.bar')  
    .css('background-color', 'green')  
.end();
```

# jQuery

## Extended selectors

### ◎ Basic Filter

- :first, :last - 첫번째, 마지막 요소를 선택(first-child와는 다름)
- :even, :odd, :eq(), :lt(), :gt() - 순서를 기준으로 선택
- :header - 제목 요소를 선택
- :not() - 선택자와 맞지 않는 요소를 선택
- :animated - 움직이고 있는 요소 선택

# jQuery

## Extended selectors

### ◎ Content Filter

- :contains() - 특정 텍스트를 포함하는 요소 선택
- :empty, :parent - 빈요소, 자식이 있는 요소를 선택
- :has(selector) - 셀렉터 요소를 포함한 요소를 선택

### ◎ Form

- :disable, :enable, :password, :selected, :text, ...

### ◎ Visibility Filter

- :hidden, :visible - 감춰졌거나 보이는 요소를 선택

# jQuery

## DOM traversing

- ◎ 표준 DOM 메서드 이외에도 자주 사용되거나 편하게 사용할 수 있는 메서드들을 지원한다.

```
$('ul.level-2').children() // 자식 노드

$('li').first(), $('li').last() // 처음, 마지막 노드

$('li.third-item').next(), $('li.third-item').nextAll() // 다음 (전부)

$('li.third-item').prev(), $('li.third-item').prevAll() // 이전 (전부)

$('li.item-a').parent(), $('li.item-a').parents() // 부모 (전부)
```

# jQuery

## DOM manipulation

- ◎ 다양한 DOM을 다루는 메서드를 제공한다.
- ◎ 인자가 없으면 값을 읽고 인자가 있으면 치환을 한다.

```
$('div.demo-container').text('<p>This is a test.</p>'); // 텍스트 치환

$('div.demo-container').html('<p>All new content. <em>You bet!</em></p>'); // HTML 치환

$('#greatphoto').attr('alt', 'Beijing Brush Seller'); // 속성 치환

$('#greatphoto').attr({ // 여러개의 속성 치환
    alt: 'Beijing Brush Seller',
    title: 'photo by Kelly Clark'
});

$('input:text.items').val(function(index, value) { // value 치환
    return value + ' ' + this.className;
});
```



# jQuery

## CSS methods

- ◎ CSS 속성과 클래스를 다루는 다양한 메서드를 제공한다.

```
$('#mydiv').css('color', 'green') // 스타일 치환

$('div.example').css('width', function(index) { // 순서에 따른 스타일치환
    return index * 50;
});

$("p").removeClass("myClass noClass").addClass("yourClass"); // 클래스
치환

$('div.tumble').toggleClass('bounce'); // 토글 클래스
```

# H/W

## Graph

### ◎ changestyle.html

- 지난 시간에 표준 DOM 메서드로 제작했던 상자의 색을 바꾸는 기능을 jQuery로 제작
- jQuery를 script 요소로 외부참조한다.
- jQuery 메서드를 사용하여 상자의 색을 #0f0으로 변경한다.
- 파일을 hyeonseok@gmail.com 으로 제목을 '제주대학교 학과 이름 학번'으로 하여 보내주세요.

# jQuery

## Event handling

```
$('a:first').click(function(ev) {  
    $(this).css({backgroundColor: 'orange'});  
    return false; // Or ev.preventDefault();  
});  
$('a:first').click();
```

- ◎ .keydown(), .keyup(), .keypress()
- ◎ .mouseover(), .mouseout(), mouseup(), mousedown()
- ◎ .mouseenter(), .mouseleave(), **.hover()**
- ◎ .click(), .dblclick(), **.toggle()**
- ◎ .bind(), .live()

# jQuery

## Event handling

### ◎ .toggle()

```
$("td").toggle(  
    function () {  
        $(this).addClass("selected");  
    },  
    function () {  
        $(this).removeClass("selected");  
    }  
);
```

### ◎ .hover()

```
$("td").hover(  
    function () {  
        $(this).addClass("hover");  
    },  
    function () {  
        $(this).removeClass("hover");  
    }  
);
```

# H/W

## Graph

### ◎ graph.html

- 입력값을 그래프로 그리는 기능
- 인풋박스의 값 세개를 받아서 이를 바의 그래프 너비로 적용한다.
- 파일을 hyeonseok@gmail.com 으로 제목을 '제주대학교 학과 이름 학번'으로 하여 보내주세요.

# jQuery

## AJAX

- `.load( url, [data], [complete(responseText, textStatus, XMLHttpRequest)] )`

```
$('#result').load('ajax/test.html');  
$('#result').load('ajax/test.html', function() {  
    alert('Load was performed.');
```

- `$.get(url, params, callback)`
- `$.post(url, params, callback)`
- `$.getJSON(url, params, callback)`
- `$.getScript(url, callback)`

# jQuery

## Effects

- ◎ `.show()`, `.hide()`, `.toggle()`
- ◎ `.fadeIn()`, `.fadeOut()`, `.fadeTo()`, `fadeToggle()`
- ◎ `.slideUp()`, `.slideDown()`, `.slideToggle()`

```
$('.target').hide('slow'); // 요소를 높이 너비를 0으로 줄여서 감춘다.
```

```
$('.target').fadeOut(2000); // 투명도를 높여 서서히 사라진다.
```

```
$('.target').slideUp('fast'); // 높이를 줄여서 감춘다.
```

# jQuery

## Effects

- ◎ .animate() - CSS 속성(숫자)으로 애니메이션을 만든다.
  - width, height, left, scrollTop, scrollLeft 등을 사용이 가능하다.
  - shorthand는 지원되지 않는다.

```
$('#clickme').click(function() {  
    $('#book').animate({  
        opacity: 0.25,  
        left: '+=50',  
        height: 'toggle'  
    }, 5000, function() {  
        // Animation complete.  
    });  
});
```

- <http://api.jquery.com/animate/> 참고



# H/W

## Graph

### ◎ slide.html

- 이미지를 움직이면서 보여주는 기능
- html 파일에는 이미지와 디자인이 이미 적용이 되어 있다.
- 마우스를 이미지 영역에 올리면 좌, 우 버튼이 나온다.
- 좌우 버튼을 누르면 이미지가 좌, 우로 움직인다.
- 더이상 보여줄 이미지가 없으면 처음으로 돌아간다.
- 파일을 hyeonseok@gmail.com 으로 제목을 '제주대학교 학과 이름 학번'으로 하여 보내주세요.

# References

## Ajax: A New Approach to Web Applications

- ◎ <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

## Ajax (programming)

- ◎ [http://en.wikipedia.org/wiki/Ajax\\_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))

## jQuery

- ◎ <http://jquery.com>, <http://api.jquery.com>