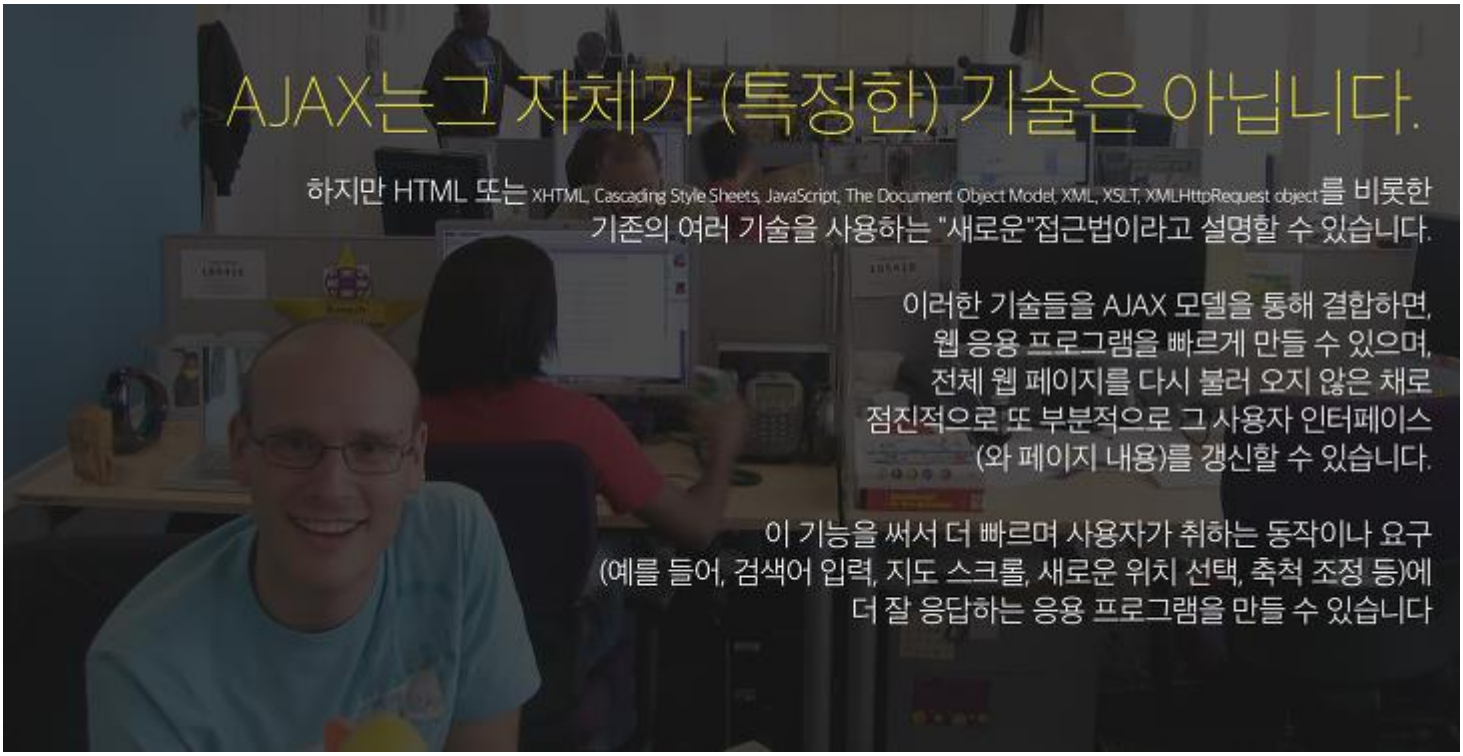
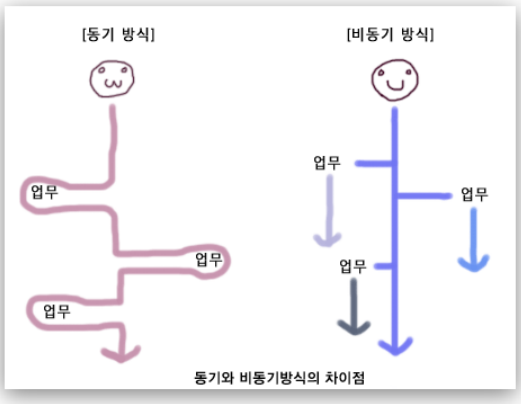
AJAX (Asynchronous Javascript And XML)

Ajax는 JavaScript의 라이브러리 중 하나이며 Asynchronous Javascript And Xml(비동기식 자바스크립트와 xml)의 약자입니다. 브라우저가 가지고있는 XMLHttpRequest 객체를 이용해서 전체 페이지를 새로 고치지 않고도 페이지의 일부만을 위한 데이터를 로드하는 기법 이며 Ajax를 한마디로 정의하자면 JavaScript를 사용한 비동기 통신, 클라이언트와 서버간에 XML 데이터를 주고받는 기술이라고 할 수 있겠습니다.



※ 비동기(async)방식이란?

비동기 방식은 웹페이지를 리로드하지 않고 데이터를 불러오는 방식입니다. 이 방식의 장점은 페이지 리로드의 경우 전체 리소스를 다시 불러와야하는데 이미지, 스크립트 , 기타 코드등을 모두 재요청할 경우 불필요한 리소스 낭비가 발생하게 되지만 비동기식 방식을 이용할 경우 필요한 부분만 불러와 사용할 수 있으므로 매우 큰 장점이 있습니다.



▶ 왜 사용하는가?

단순하게 WEB화면에서 무언가 부르거나 데이터를 조회하고 싶을 경우, 페이지 전체를 새로고침하지 않기 위해 사용한다고 볼 수 있다.
기본적으로 HTTP프로토콜은 클라이언트쪽에서 Request를 보내고 Server쪽에서 Response를 받으면 이어졌던 연결이 끊기게 되어있습니다. 그래서 화면의 내용을 갱신하기 위해서는 다시 request를 하고 response를 하면서 페이지 전체를 갱신하게 된다. 하지만 이렇게 할 경우 페이지의 일부분만 갱신할 경우에도 페이지 전체를 다시 로드해야 하는데 엄청난 자원낭비와 시간낭비이다. 하지만 AJAX는 html 페이지 전체가 아닌 일부분만 갱신할 수 있도록 XML HttpRequest객체를 통해 서버에 request를 하고 JSON이나 XML형태로 필요한 데이터만 받아 갱신하기 때문에 그만큼의 자원과 시간을 아낄 수 있다. 웹에서 가장 중요한 것은 속도이며 Ajax를 사용해야 하는 이유로써 충분하다.

▶AJAX 를 사용 가능하게 만드는 것들

AJAX 라는 기술은 여러가지 기술이 혼합적으로 사용되어 이루어진다. 대표적인 예로는 아래와 같은 것들이 있다.

- HTML
- DOM
- JavaScript
- XMLHttpRequest
- Etc

AJAX 의 장단점

1. AJAX 의 장점

- 웹 페이지의 속도향상
- 서버의 처리가 완료될 때까지 기다리지 않고 처리가 가능하다.
- 서버에서 Data 만 전송하면 되므로 전체적인 코딩의 양이 줄어든다.
- 기존 웹에서는 불가능했던 다양한 UI 를 가능하게 해준다. (Flickr 의 경우, 사진의 제목이나 태그를 페이지의 리로드 없이 수정할 수 있다.)

2. AJAX 의 단점

- 히스토리 관리가 되지 않는다.
- 페이지 이동없는 통신으로 인한 보안상의 문제가 있다.
- 연속으로 데이터를 요청하면 서버 부하가 증가할 수 있다.
- XMLHttpRequest 를 통해 통신하는 경우, 사용자에게 아무런 진행 정보가 주어지지 않는다. (요청이 완료되지 않았는데 사용자가 페이지를 떠나거나 오작동할 우려가 발생하게 된다.)
- AJAX 를 쓸 수 없는 브라우저에 대한 문제 이슈가 있다.
- HTTP 클라이언트의 기능이 한정되어 있다.
- 지원하는 Charset 이 한정되어 있다.
- Script 로 작성되므로 디버깅이 용이하지 않다.
- 동일-출처 정책으로 인하여 다른 도메인과는 통신이 불가능하다. (Cross-Domain 문제)

▶Jquery 와의 시너지

Ajax하면 Jquery에 대한 설명을 빼놓을 수 없습니다. 일반 Javascript만으로 Ajax를 하게되면 코딩량도 많아지고 브라우저별로 구현방법이 다른 단점이 있는데 jquery를 이용하면 더 적은 코딩량과 동일한 코딩방법으로 대부분의 브라우저에서 같은 동작을 할 수 있게 됩니다. jquery ajax를 사용하면, HTTP Get방식과 HTTP Post방식 모두를 사용하여 원격 서버로부터 데이터를 요청할 수 있습니다. Jquery는 Ajax처럼 JavaScript의 라이브러리 중 하나인데 자바스크립트를 좀 더 사용하기 쉽게 패키징화 시켜놓은 것입니다. Jquery에대한 내용은 추후에 포스팅하겠습니다.

AJAX 의 진행과정

1. XMLHttpRequest Object 를 만든다.
    - request 를 보낼 준비를 브라우저에게 시키는 과정
    - 이것을 위해서 필요한 method 를 갖춘 object 가 필요함
2. callback 함수를 만든다.
    - 서버에서 response 가 왔을 때 실행시키는 함수
    - HTML 페이지를 업데이트 함
3. Open a request
    - 서버에서 response 가 왔을 때 실행시키는 함수
    - HTML 페이지를 업데이트 함

4. send the request

AJAX 가 쓰이는 방법

XMLHttpRequest 객체를 얻은 뒤, url 을 통해 요청하고 응답을 받으면 응답 결과에 맞는 함수를 실행하는 구조로 되어 있다. Ajax 가 효율적이라고는 해도 이렇게 하게 될 경우, 코드가 길어지기 때문에 jQuery 에서 그 문제를 해결해주고 있다.

예시로 보는 AJAX - 1

```
// This function gets invoked when server sends the response
function reqListener (e) {
    console.log(e.currentTarget.response);
}

var oReq = new XMLHttpRequest();
var serverAddress = "https://hacker-news.firebaseio.com/v0/topstories.json";

oReq.addEventListener("load", reqListener);
oReq.open("GET", serverAddress);
oReq.send();
```

위의 예제는 **자바스크립트를 이용하여 특정 서버에 요청을 보내고 그에 대한 자료를 성공적으로 받아올 수 있음을 확인**해볼 수 있다. 위 예제에서는 XMLHttpRequest 를 이용하여 요청을 보냈지만 일반적으로는 아래와 같이 jQuery 나 기타 AJAX 기능이 내장되어 있는 라이브러리를 이용하여 AJAX 요청을 처리한다.

```
var serverAddress = 'https://hacker-news.firebaseio.com/v0/topstories.json';

// jQuery 의 .get 메소드 사용
$.ajax({
    url: ,
    type: 'GET',
    success: function onData (data) {
        console.log(data);
    },
    error: function onError (error) {
        console.error(error);
    }
});
```

예시로 보는 AJAX - 2

```
var xhr= new XMLHttpRequest();

xhr. onreadystatechange = function(){
```

```
        if(xhr.readyState===4){
        document.getElementById('ajax').innerHTML= xhr.responseText;
    }
}

xhr.open('GET',"sidebar.html");  // html 메소드와 URL 을 보낸다. (open 함수는
준비를 시키는것이지 보내는 것은 아니다.)
xhr.send();
```

위의 예제는 AJAX 가 XHR 객체를 형성하고 이 객체의 콜백을 만들고
HTML 메소드와 URL 을 결정한 뒤, XHR 객체의 메소드로 정보를 보내는 방식이다.

- **var xhr= new XMLHttpRequest();** : browser response 를 얻었을 때 작동하는 함수 (callback 함수)
- **xhr.onreadystatecjange** : AJAX Request 에 어떠한 변화라도 있으면 작동한다. (callback 함수를 포함하고 있다고 생각하면 된다.)
- **xhr.readyState** : response 가 돌아왔는지 아닌지를 추적하는 property