

UMC Server

9주차

Web Practice

01

과제 점검

02

학습 목표

- 학습목표

자신이 개발한 API를 웹에 적용할 수 있다.

클라이언트와 협업을 원활히 할 수 있는

서버 개발자가 된다!

- 학습목표

- HTML DOM

- Ajax

- 부트스트랩

- OPEN API

03

HTML DOM

■ HTML DOM

HTML 문서 자체로는 **정적** 웹페이지이다



동적으로 변경하고 싶을 때는 Javascript를 활용하게 된다



이 때 **DOM**에 접근해서 해당하는 영역을 동적으로 조작하게 된다

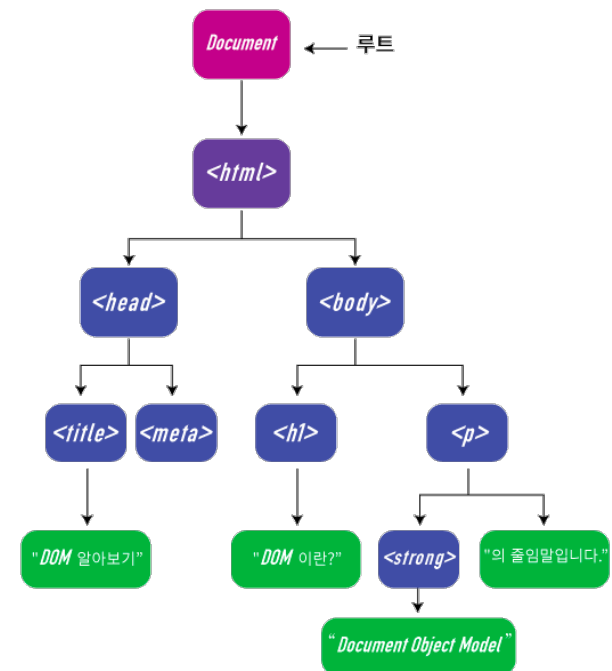
■ HTML DOM

: Document Object Model

HTML, XML 문서에 대한 인터페이스 역할

→ Javascript는 DOM을 통해

각 객체의 속성을 추가/수정/제거할 수 있다.



■ DOM API



https://developer.mozilla.org/ko/docs/Web/API/Document_Object_Model

■ DOM API

HTML 문서 내 특정 요소를 찾는 API

- Document.getElementById**Id**
 - Document.getElementsByClassName**ClassName**
-

■ DOM API

이벤트핸들러를 등록하는 API

- EventTarget.addEventListener()

마우스 이벤트:

- `click` - 요소 위에서 마우스 왼쪽 버튼을 눌렀을 때(터치스크린이 있는 장치에선 탭 했을 때) 발생합니다.
- `contextmenu` - 요소 위에서 마우스 오른쪽 버튼을 눌렀을 때 발생합니다.
- `mouseover` 와 `mouseout` - 마우스 커서를 요소 위로 움직였을 때, 커서가 요소 밖으로 움직였을 때 발생합니다.
- `mousedown` 과 `mouseup` - 요소 위에서 마우스 왼쪽 버튼을 누르고 있을 때, 마우스 버튼을 떼 때 발생합니다.
- `mousemove` - 마우스를 움직일 때 발생합니다.

폼 요소 이벤트:

- `submit` - 사용자가 `<form>` 을 제출할 때 발생합니다.
- `focus` - 사용자가 `<input>` 과 같은 요소에 포커스 할 때 발생합니다.

키보드 이벤트:

- `keydown` 과 `keyup` - 사용자가 키보드 버튼을 누르거나 떼 때 발생합니다.

문서 이벤트:

- `DOMContentLoaded` - HTML이 전부 로드 및 처리되어 DOM 생성이 완료되었을 때 발생합니다.

CSS 이벤트:

- `transitionend` - CSS 애니메이션이 종료되었을 때 발생합니다.

04

AJAX

■ XMLHttpRequest

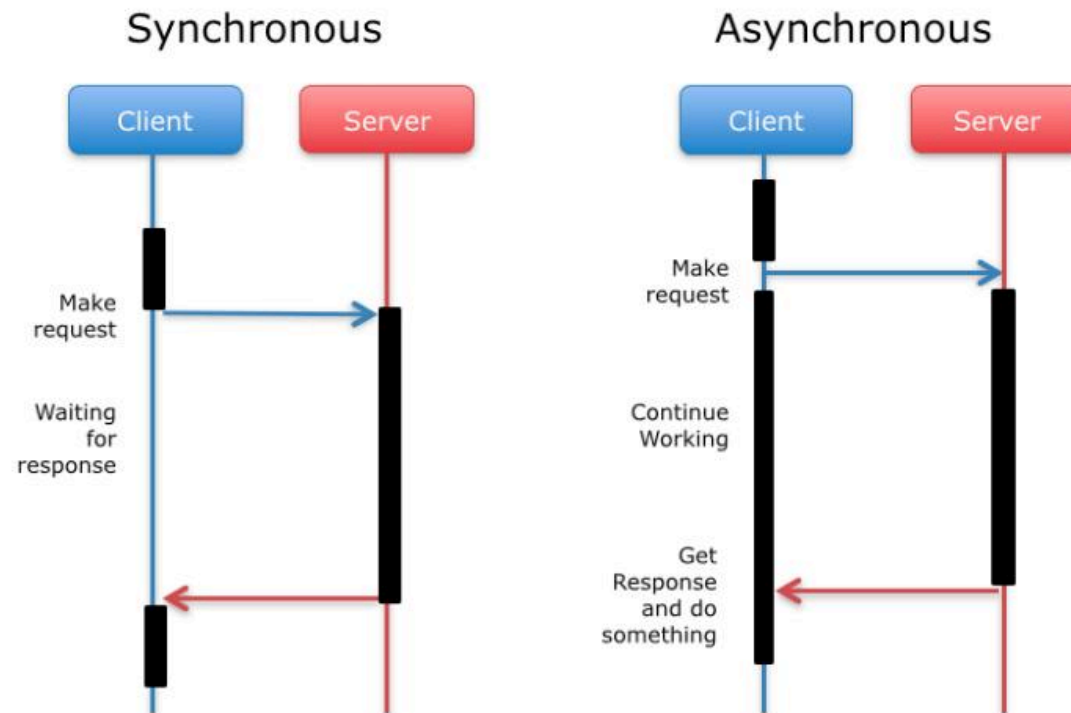
전체 웹페이지를 새로고침하는 것이 아니라
일부분만 **비동기적**으로 갱신



Q. 일부분만 업데이트 하는 방식을 사용하는 이유?

페이지를 아예 새로고침하는 경우,
모든 리소스를 불러와야하기 때문에
시간과 자원이 낭비된다!

- XMLHttpRequest



- XMLHttpRequest

HTML + CSS + Javascript + XMLHttpRequest

= **AJAX**

(Asynchronous Javascript And XML)

■ XMLHttpRequest

```
const xhr = new XMLHttpRequest(); //XMLHttpRequest 객체 생성
xhr.open("GET", "/posts"); //HTTP Method, URL 정의
xhr.setRequestHeader("content-type", "application/json"); //헤더 값 중 content-type 정의
xhr.send(); // 요청 전송
xhr.onload = () => {
    if(xhr.status === 200) { // 정상적으로 응답이 되면 ( status 값 200 )
        const res = JSON.parse(xhr.response); // 응답 데이터를 JSON.parse 함수로 JSON 객체로 변환
    } else { // 에러 발생
        console.error(xhr.status, xhr.statusText);
    }
}
```

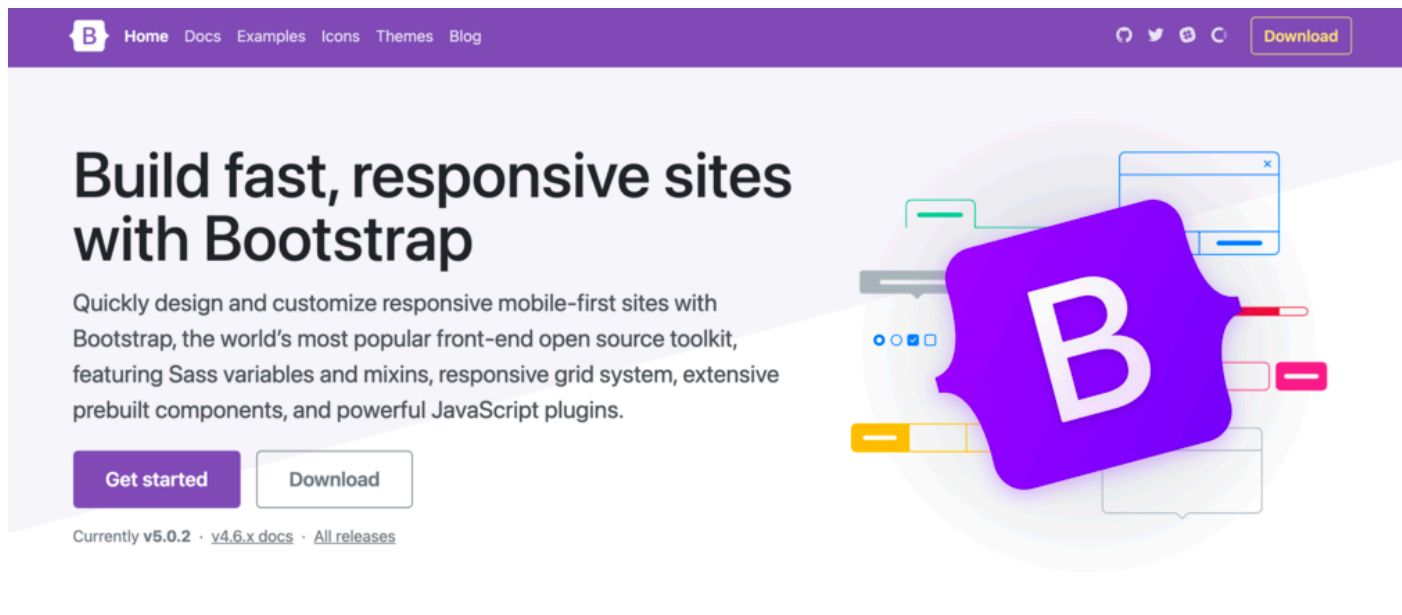

■ jQuery - AJAX

```
$.ajax({  
  url: 'https://jsonplaceholder.typicode.com/posts/1',  
  type: 'GET',  
  success: function(data) {  
    console.log(data);  
  },  
  error: function (error) {  
    console.error(error);  
  }  
});
```

05

Bootstrap

■ Bootstrap



<https://getbootstrap.com/>

06

OPEN API

07

실습

08

과제

- 과제

자신이 직접 만든 API or OPEN API를 활용해서
웹 페이지 만들어보기
