

[WEB UI BASIC]

JavaScript 3탄

NHN NEXT 우재우

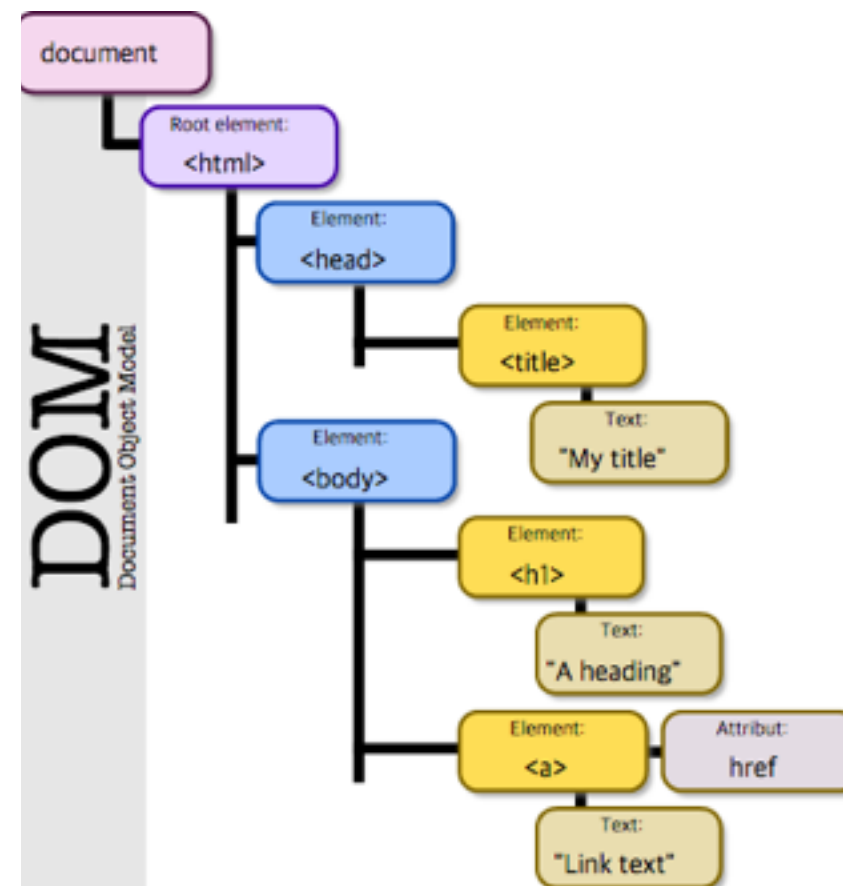
본 자료는 NHN NEXT 윤지수 교수님의 자료를 바탕으로 제작했습니다.

오늘은 JS를 이용해서 DOM을 조작합시다!

그리고 조작한 DOM에 이벤트도 걸어보구요!

DOM???

DOM(Document Object Model, 문서 객체 모델)은 HTML 및 XML 문서를 처리하는 API입니다. 문서의 구조적 형태를 제공하므로 자바스크립트(Javascript)와 같은 스크립트 언어를 사용하여 문서 내용과 시각적 표현을 수정할 수 있습니다.



DOM을 탐색해봅시다!!

원하는 영역을 가져오려면 어떻게 해야할까요?

이미 답은 우리가 작성했다!

```
<div id= “ hello ” class= “ world ” >
```

<div id= " hello " class= " world " >

tag / id / class

DOM Element를 찾는 세가지 방법

- Tag Name으로 찾기

```
document.getElementsByTagName( ' div ' );
```

- ID로 찾기

```
document.getElementById( ' hello ' );
```

- Class로 찾기

```
document.getElementsByClassName( ' world ' );
```

S를 주의합시다!!!

- **Tag Name**으로 찾기

`document.getElementByIdSByTagName(' div ');`

- **ID**로 찾기

`document.getElementByIdById(' hello ');`

- **Class**로 찾기

`document.getElementByIdSByClassName(' wo1rd ');`

크롬 개발자 도구 콘솔에서 확인해보요~

```
> var hello = document.getElementById('hello');  
< undefined  
-----  
> hello;  
< <div id="hello" class="world"></div>
```

나머지 두 개도 해봅시다!!!

크롬 개발자 도구 콘솔에서 확인해보요~

```
var div = document.getElementsByTagName("div");  
undefined  
div;  
[ <div id="hello1" class="world"></div>, <div id="hello2" class="world"></div>, <div id="hello3" class="world"></div>]
```

```
var world = document.getElementsByClassName("world");  
undefined  
world  
[ <div id="hello1" class="world"></div>, <div id="hello2" class="world"></div>, <div id="hello3" class="world"></div>]
```

The `getElementsByClassName()` method returns a collection of all elements in the document with the specified class name, as a **NodeList object**.

출처: w3schools.com

그치만 **id**나 **class**를 입히지 않은 엘리먼트는

어떻게 탐색해야 하죠??

tagName으로 잡아올 수도 없는 노릇이고...

우리의 필살기

CSS Selector

CSS Selector로 가져오기

```
document.querySelector( ' #hello ' );
```

일치하는 첫 번째 Element 반환

```
document.querySelectorAll( ' .world ' );
```

일치하는 모든 Element 반환

크롬 개발자 도구에서 확인해봅시다!

**그래도 ID 검색이 빠르기 때문에
섞어 쓰는 것이 이상적입니다!**

```
document.getElementById( ' wrap ' ).querySelector( ' ul li:last-child ' );
```



잡아왔으니 이제 수정해봅시다!!!

바로 HTML코드 패러박기

```
var hello = document.getElementById( ' hello ' );
```

```
hello.innerHTML = “ <div>Bye, world!</div> ” ;
```

근데 이렇게 하면... 원래 있던 내용은 다 날아가는데?

**방법은 많다! 사용법은 생략한다.
구글링해라!**

```
element.innerHTML += "html";
```

```
element.insertAdjacentHTML(position, html);
```

```
element.insertBefore(new node, existing node);
```

Element 만들어서 넣기

1단계: Element를 만들자!

```
var newDiv = document.createElement( ' div ' );
```

2단계: newDiv에 ID를 줄까?

```
newDiv.id = ' newId ' ;
```

3단계: class도 줘볼까?

```
newDiv.className = ' newClass ' ;
```

4단계: hello element(var hello) 밑에 추가해주자!

```
hello.appendChild(newDiv);
```

5단계: 아냐, 잘못 넣었어. 지우자!

```
hello.removeChild(newDiv);
```

Text 조작하기

```
element.textContent = “ blah blah ” ;
```

```
document.createTextNode = “ blah blah ” ;
```

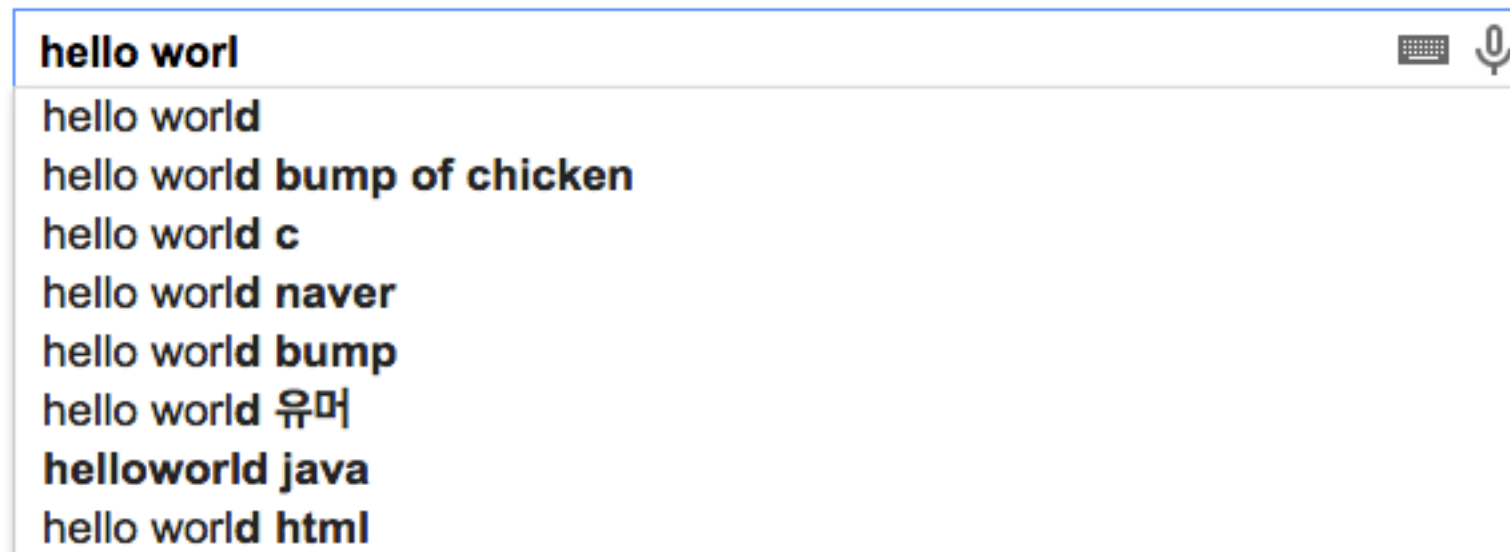
이쯤에서 DOM 조작을 마치고 이벤트로 가봅시다!

걱정 마세요~

JavaScript 이벤트가 어렵다고 해도

여자친구 이벤트보다 어렵지는 않아요

이벤트가 어디있다는 거냐?



이거 이벤트 사용한 겁니다!

키보드 입력이 있으면 자동완성 문구들을 가져오죠??

이벤트(Event)?

이벤트는 해당 엘리먼트에서 어떤 동작이나 사건이 발생했음을 알려주는 신호입니다. 예를 들어서...

마우스 버튼을 누른다 / 마우스가 움직이고 있다 / 키보드가 눌렀다
입력양식이 선택되었다 / 웹브라우저 문서를 읽었다 등등등 많아요.

그리고 이 이벤트를 관리하고, 함수 처리를 도와주는 녀석을

Event Listener 또는 **Event Handler**라고 합니다.

이벤트가 발생하면 다양한 이벤트 정보를 얻을 수 있어요.

이벤트가 발생한 element가 어디인지, 마우스 좌표가 어디인지,
무슨 버튼이 눌렸는지, 밥은 잘 먹는지 윈?

이벤트를 설정해봅시다
크게 HTML에 넣는 방법과
JS로 접근하는 방법이 있어요

이벤트를 HTML에 직접 넣기

click event를 등록해볼까요?

```
<div onclick= ' alert( " CLICK!! " ); ' >CLICK ME!</div>
```

JavaScript에 있는 함수를 불러올 수도 있어요~

```
<script>  
    function greeting() {  
        alert( " hello, world " );  
    }  
</script>
```

```
<div onclick= ' greeting(); ' >CLICK ME!</div>
```


HTML EVENT

onclick 외에도 HTML EVENT에 관해서 더 알고 싶다면

http://www.w3schools.com/tags/ref_eventattributes.asp

여기서 공부해보세요~ 저도 요즘 여기 찬찬히 보고 있는데 재미있
더라구요. 조만간 정리해서 공유할게요 :)

한 가지 아쉽다면...

HTML은 구조를 잡는 용도인데 JavaScript를 넣다니!!!

HTML과 CSS를 분리했듯이, HTML에 직접 이벤트를 거는 것은
그리 권장되는 바는 아닙니다.

JS로 이벤트 걸기

일단 무따기(무조건 따라하기)!

```
function greeting() {  
    alert( " hello! " );  
}
```

```
document.addEventListener( " click " , greeting, false);
```

addEventListener();

함수는 이름만 쓰세요! 괄호() 없어요~
아니면 여기서 함수를 바로 쓸 수도 있긴하구요
나중에 같이 볼게요

document.**addEventListener**(이벤트, 함수, false);

캡처링(true) / 버블링(false)에 관한
인자인데 지금은 그냥 넘어갑시다~
디폴트가 false니까 안 써도 괜찮아요!

이벤트의 종류는 <http://help.dottoro.com/larrqqck.php>
여기를 참고하세요~

특정 Element에 걸 수도 있어요!

```
var hello = document.getElementById( ' hello ' );  
  
function greeting() {  
    alert( ' hi! ' );  
}  
  
hello.addEventListener( " click " , greeting, false);
```

불인 EventListener를 뗄 때는

removeEventListener();

Event Loop??

JavaScript는 Single thread입니다. 그래서 이벤트를 등록하고 나서 JavaScript가 스스로 이벤트를 계속 기다리게 되면 다른 작업을 할 수 없겠죠? 그래서 Event Queue가 존재합니다.

아래 링크는 JSConf EU 2014에서 발표한 자료인데 이 영상 보시면 JavaScript가 내부적으로 어떻게 동작하는지 정말 잘 설명되어 있으니 꼭!꼭!꼭! 참고하시길!!

<http://youtu.be/8aGhZQkoFbQ>

오늘 너무 많이 배운 것 같으니까

여기까지 하죠 ㅋㅋㅋ

다음 시간에도 이벤트 이어서 합니다!

그렇지만 아직 끝난 거 아닙니다!

실습이 남았어요~

[실습]

버튼을 클릭하면 1부터 45 사이의 랜덤한 숫자가 적힌 상자가
중복 없이 7개까지 추가되는 프로그램 짜기

예)

추 가

32

23

1

17

14

41

5

이때까지 배운거 다 써야 짤 수 있을걸요? 후훗