

# 삼성 청년 SW 아카데미

Web Programming 기초

## <알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로  
보안서약서에 의거하여  
강의 내용을 어떠한 사유로도 임의로 복사,  
촬영, 녹음, 복제, 보관, 전송하거나  
허가 받지 않은 저장매체를  
이용한 보관, 제3자에게 누설, 공개,  
또는 사용하는 등의 행위를 금합니다.

# JavaScript 기초

## • • • 챕터의 포인트

- JavaScript 개요
- JavaScript 기본 문법

# JavaScript 개요

HTML 문서에 글, 표, 이미지만 넣을 수 있는 것이 아니다.

HTML 문서에 소스코드를 삽입할 수 있다.

- HTML 문서에 if / for / 변수 / 배열 / 함수 등을 사용한 소스코드 넣기

소스코드 Run은 누가?

- 웹브라우저가 이 HTML을 여는 순간, 웹브라우저가 직접 수행한다.

다양한 기능을 HTML 문서에서 수행이 가능하다.

- Ex. 특정 영역 클릭했을 때 반응
- Ex. 사용자 ID / PASS에 따라 로그인 기능

HTML 문서로 게임 / App 도 제작 가능하다.

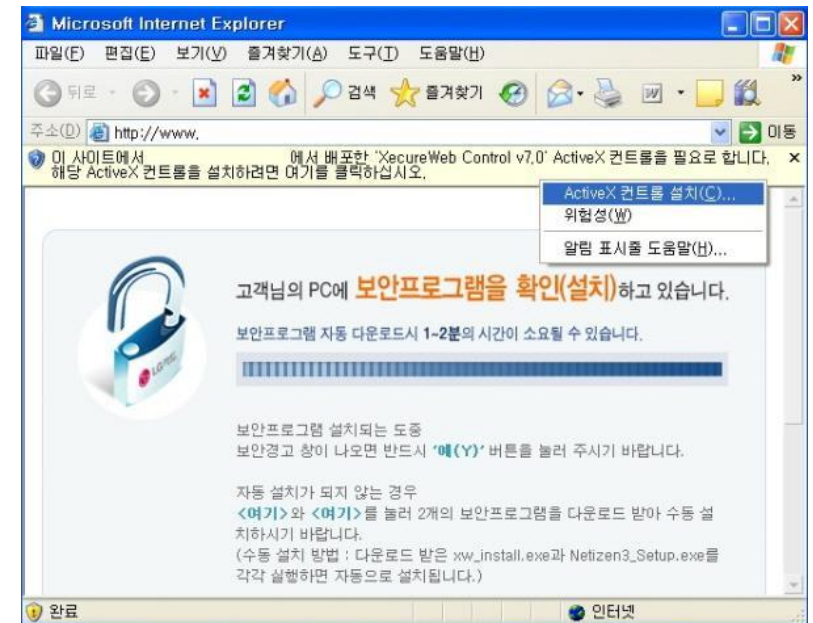
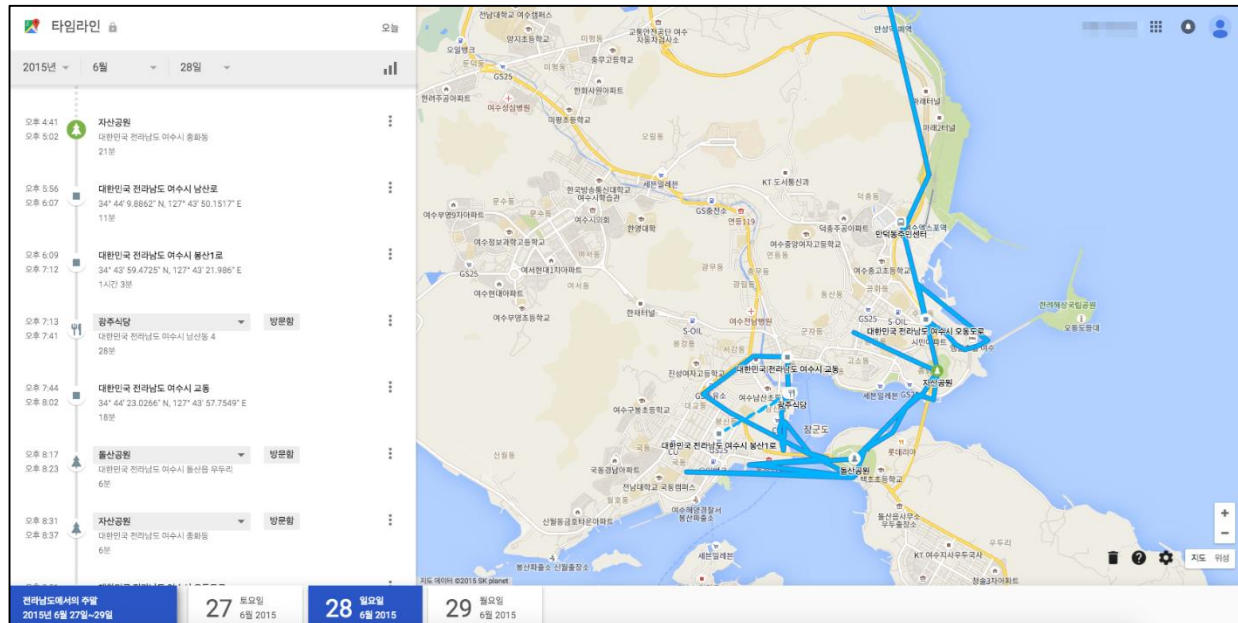
## JavaScript는 예전부터 있어왔다.

- 1990년대부터 다양한 웹브라우저에 JavaScript 인터프리터를 내장하였다.
- 사용처
  - HTML에서 클릭시 Form 내용 조건 검사
  - ActiveX 실행
  - if / for / 변수 / 배열 / for문 수행 가능
  - 특정 영역 값 변경 가능



## Asynchronous JavaScript and XML

- JavaScript 와 XML 을 이용하여, 비동기 통신을 구현하는 기법
- 이 기법을 구글이 세상에 알림
- 현재 XML 대신 JSON을 사용함



## 웹 페이지 새로고침 없이, 일부만 변경 가능

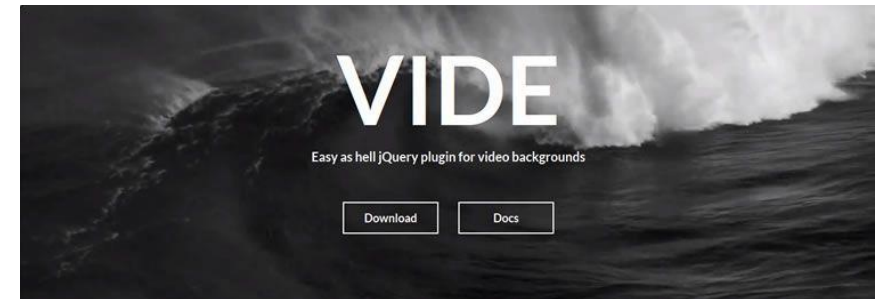
- 서버에 있는 새로운 정보를 받아오기 위해 웹 페이지 전체를 서버에게 전달 받지 않고 일부만, 서버에게 요청 받아서 변경 가능한 방식
- 클릭하더라도, 새로고침이 발생하지 않고, 특정 영역만 바뀌게 하는 기술
- 서버의 부담을 줄일 수 있다.



실시간 급상승   뉴스토픽	
1~10위	11~20위
1 프리허그	
2 황사	
3 정현	
4 미세먼지 농도	
5 공기청정기	
6 미세먼지	
7 비밀 예능 연수원	
8 시카고 타자기	
9 맨투맨	
10 기상청	

## JavaScript의 Library, jQuery

- 아주 쉽게 자주쓰는 javascript 소스코드들을 모아 둔 Library
- jQuery의 다양한 플러그인으로 홈페이지에 멋진 효과를 줄 수 있다.
- jQuery만의 방식으로 다음과 같은 항목을 쉽게 구현 가능
  - 애니메이션
  - 이벤트 제어
  - Ajax 기술



## JavaScript의 업그레이드

### Client Side 언어 -> Server Side 언어로

- 구글이 JavaScript 인터프리터를 OpenSource로 공개 (V8)
- V8을 사용하여 Node.js 라는 JavaScript 런타임이 탄생함

## Hello world 출력하기

1. `alert` 사용
2. `console.log` 사용

# JavaScript 기본 문법

## 강의 진행 방식

- C언어를 할 줄 아는 사람을 대상으로 강의를 진행
- JavaScript의 if / for / 객체 / 배열 / 함수 를 익힌다.

객체에 포함되어 있는 세부적인 메서드들에 대한 사용법은 차후 다룰 예정

## var 대신 사용하는 문법

- var
  - 재 할당이 가능하다
  - 함수 레벨 스코프
- let : C언어 변수와 비슷
  - 재 할당이 가능하다
  - 블록 레벨 스코프
- const : C언어 const와 비슷
  - 재 할당 불가
  - 블록 레벨 스코프

```
<> hello.html > script
1  <script>
2
3      var a = 10;
4      var a = 20;
5
6  </script>
```

가능

```
<> hello.html > script
1  <script>
2
3      let a = 10;
4      let a = 20;
5
6  </script>
```

불가능

```
<> hello.html > script
1  <script>
2
3      let a = 10;
4      a = 20;
5
6  </script>
```

가능

var가 무엇인지 알고있되, 더 이상 var를 쓰지 말자.



## 함수 레벨 스코프

- 함수 내에서 선언된 변수는 함수 내에서만 유효하다
- var는 함수 레벨 스코프

```
function hello(){  
  var a = 5;  
}  
hello();  
console.log(a);
```

► Uncaught ReferenceError: a is not defined  
at <anonymous>:5:13

```
{  
  var a = 5;  
}  
console.log(a);  
5
```

## 블록 레벨 스코프

- 블록 { } 내에서 선언된 변수는 코드 블록 내부에서만 유효하다.
- let / const 는 블록 레벨 스코프다

```
function hello(){  
  const a = 5;  
}  
hello();  
console.log(a);
```

► Uncaught ReferenceError: a is not defined  
at <anonymous>:5:13

```
{  
  const a = 10;  
}  
  
console.log(a);
```

► Uncaught ReferenceError: a is not defined  
at <anonymous>:5:13

자바스크립트는 크게 2가지 데이터 타입으로 나뉘어져 있다.

- 원시 자료형 타입
  - number
  - string
  - Boolean
  - null
  - undefined
- 참조 자료형(객체) 타입
  - Array/Object/Function은 모두 객체로 취급된다.
  - Array
  - Object
  - Function

## Number

- 원시 자료형
- 정수, 실수 전부 다 Number 타입으로 통일된다.

```
typeof(100)
```

```
'number'
```

```
typeof(-300.1)
```

```
'number'
```

## Boolean

- true/false 로 나누는 진리값을 뜻한다.

```
typeof(true)  
'boolean'
```

```
typeof(false)  
'boolean'
```

- null

- 값이 비어있다고 명시하는것

- undefined

- 아무 값도 할당받지 않은 상태
- 개발자가 할당한것이 아니라 자바스크립트 엔진이 변수를 초기화 할 때 사용

## Array

- [ ] 로 표기한다
- `const a = [ 1, 2, 3, 4 ]`
  - 배열에 접근하는 경우 `a[0]` 등 index를 활용해 접근한다.

```
const a = [1, 2, 3, 4]
undefined
console.log ( a[1] );
2
```

기본적으로 C언어와 비슷하게 쓸 수 있다.

- function 으로 시작한다.

```
<> test.html > ...
1  <script>
2
3  function kfc()
4  {
5      alert("#");
6  }
7
8  kfc();
9
10 </script>
11
```

# 출력

```
<> test.html X
<> test.html > ...
1  <script>
2
3  function kfc(a, b)
4  {
5      return a + b;
6  }
7
8  a = kfc(5, 3);
9  alert(a);
10
11 </script>
12
```

8 출력

```
<> test.html X
<> test.html > ...
1  <script>
2
3  function kfc(a, b)
4  {
5      bbq();
6      return String(a + b);
7  }
8
9  function bbq()
10 {
11     alert("BBQ");
12 }
13
14 a = kfc(5, 3);
15 alert(typeof a);
16
17 </script>
18
```

BBQ 출력  
string 출력



## 함수 표현식

- 함수를 변수에 저장
- 함수가 저장된 변수를 호출하면, 해당 함수가 호출 됨

```
<> test.html > ...
1  <script>
2
3  let go = function() { alert("#"); };
4
5  go();
6
7  </script>
8
```

```
<> test.html > ...
1  <script>
2
3  function ABC() {
4      console.log("##");
5  }
6
7  bbc = ABC;
8  bbc();
9
10 </script>
11
```

```
<script>
let run = function() {
    for (let i = 0; i < 3; i++) {
        console.log("#");
    }
};

run();

</script>
```

```
const result = { a: 5, b: 10 }
```

- 객체 값에 접근하기 위한 방법
  - result['a']
  - result.a 로 접근 가능하다.

```
const result = {a: 5, b: 10}
```

```
undefined
```

```
result['a']
```

```
5
```

```
result.a
```

```
5
```

var ssafy = 19;

- var ssafy; (변수 선언)
- ssafy = 19; (변수 할당) 의 형태로 할당된다.



런타임 이전에 변수의 “선언문”들 만, 먼저 끌어져 올라오는 행위

- `var ssafy; let ssafy; const ssafy;`
- `let`과 `const`에서는 호이스팅이 동작하는걸 막는다.

```
console.log(ssafy);  
var ssafy = 19;
```

undefined

undefined

var 사용시 undefined

```
> console.log(ssafy)  
let ssafy = 19;
```

```
✖ ▶ Uncaught ReferenceError: ssafy is  
not defined VM1747:1  
at <anonymous>:1:13
```

let 사용시 error

```
> console.log(ssafy)  
const ssafy = 19
```

```
✖ ▶ Uncaught ReferenceError: ssafy is  
not defined VM2548:1  
at <anonymous>:1:13
```

const 사용시 error

함수의 호이스팅은 두가지 방법에 따라 다르다.

- 함수 선언식
  - 함수 호이스팅이 발생
- 함수 표현식
  - 변수 호이스팅이 발생

```
hello();  
  
function hello () {  
    console.log("hello world")  
}  
  
hello world
```

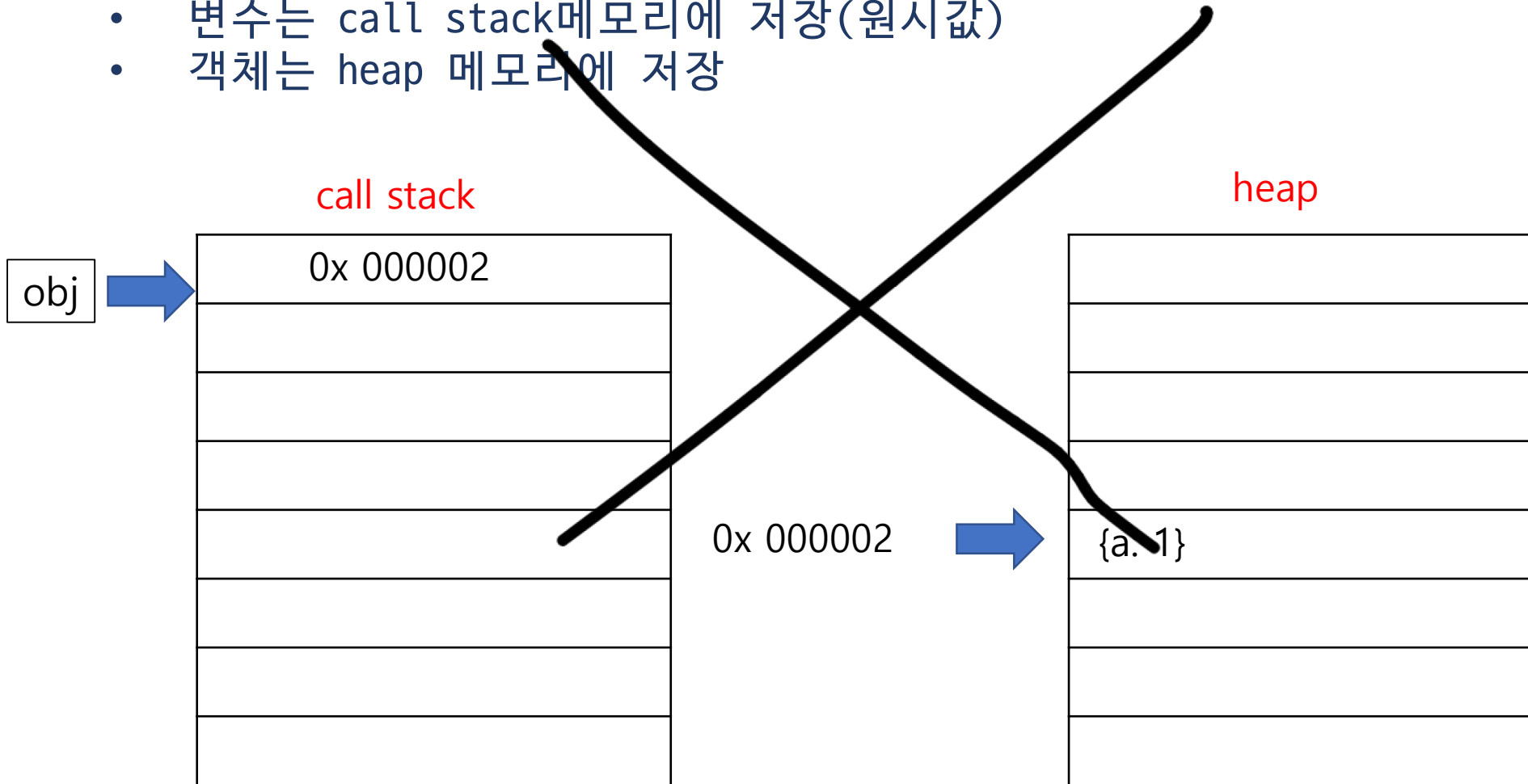
함수 선언식: 함수의 호이스팅

```
hello();  
  
const hello = function() {  
    console.log("hello world");  
}  
  
▶ Uncaught ReferenceError: hello is not defined  
   at <anonymous>:1:1
```

함수 표현식: 변수의 호이스팅

```
const obj = { a: 1};
```

- 변수는 call stack 메모리에 저장(원시값)
- 객체는 heap 메모리에 저장



## const 로 선언된 참조 자료형에서 값 바꾸기

- call stack 메모리의 값을 변경시키려 했기때문에 에러가 발생.

```
const obj = { a : 1};
```

```
undefined
```

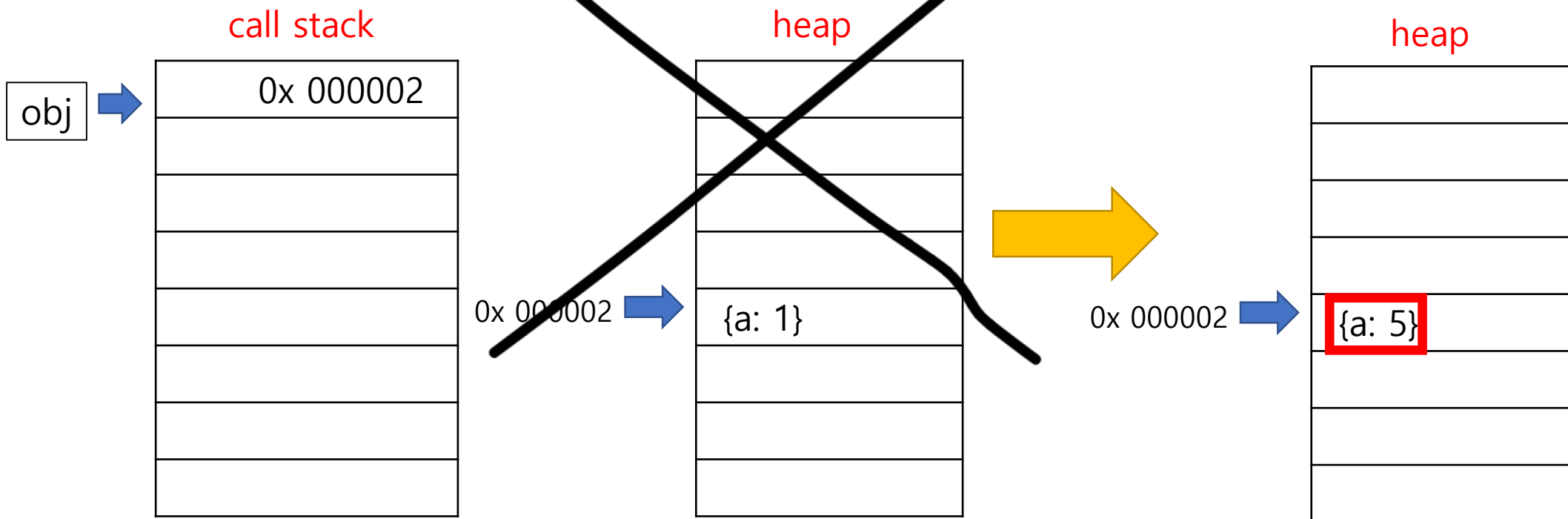
```
obj = { b: 1}
```

```
▶ Uncaught TypeError: Assignment to constant variable.  
  at <anonymous>:1:5
```

## const 로 선언된 참조 자료형에서 값 바꾸기

- const는 기본적으로 call stack에 대한 변경을 막아두었다.
- 객체의 값은 heap 메모리에 저장되기 때문에 변경 가능

```
> const obj = { a : 1};  
< undefined  
> obj.a = 5;  
< 5  
> console.log(obj);  
▶ {a: 5}
```





## prompt로 입력받기

- 문자열 입력 받고, 콘솔 출력하기

```
test.html > ...  
1 <script>  
2   var txt = prompt("입력하십시오");  
3   console.log(txt);  
4 </script>
```

127.0.0.1:5500 내용:

입력하십시오

확인 취소

Elements Console Sou

top

안녕

>

## alert로 출력하기

- 입력 후 type 출력하기
- typeof(txt) 또는 typeof txt 모두 가능

```
<script>  
  var txt = prompt("입력하십시오");  
  alert(typeof txt);  
</script>
```



127.0.0.1:5500 내용:

입력하십시오

확인 취소

127.0.0.1:5500 내용:

string

확인

## ECMAScript

- ECMA 국제기구에서 만든 Script 언어 기준, ES 라고 부른다.  
(JavaScript가 아님, Script 언어에 대한 표준)
- 2020년 7월 기준, ES11이 최신

## JavaScript 에서 ES

- JavaScript 언어는 ES 표준을 따르며 제작된다.
- ES6 가 등장함으로, 기존에 없던 JavaScript 문법과 권장사항이 추가되었다.

현재 ES11이 최신이지만,  
ES6 당시 혁신적 변화로 인해  
ES6 이후를 "모던 JavaScript" 라고 표현한다.

## Number와 String 형 변환이 자유롭다.

- Number(변수)
- String(변수)

### [참고] 세미콜론

- 파이썬 : 세미콜론 없는 것이 기본 (있어도 에러 안남)
- JavaScript : 세미콜론 있는 것이 기본 (없어도 에러 안남)

```
hello.html > script
1  <script>
2
3      let a = 10;
4      let b = Number(a);
5      let c = String(a);
6
7      console.log(a, typeof a);
8      console.log(b, typeof b);
9      console.log(c, typeof c);
10
11  </script>
```

10	"number"
10	"number"
10	string
>	

## 파이썬 처럼 '+' 연산자로 문자열 붙이기 가능

- 문자열.length : 문자열 길이가 저장된 속성

```
hello.html > script
1 <script>
2
3   let a = "ABC" + "BBQ";
4   let b = "BBQ" + 10
5
6   console.log(a, b)
7
8 </script>
```

ABCBQ BBQ10

```
hello.html > script
1 <script>
2
3   let a = "KFC" + 119 + "BBQ";
4   console.log(a.length)
5
6 </script>
```

9

## C언어와 동일하다.

- >, >=
- <, <=

## 같다와 다르다 주의

- == 와 != : Type 달라도 True
- === 와 !== : Type 도 같아야만 True

JavaScript에서는  
같다와 다르다를 === / !== 를 사용하자

```
<> hello.html ●
<> hello.html > ...
1  <script>
2
3      let a = 5;
4      let b = 10;
5
6      if (a < b) {
7          alert("B BIG");
8      }
9      else if (a === b) {
10         alert("A == B");
11     }
12     else {
13         alert("A BIG");
14     }
15
16 </script>
17
```

C언어와 for / while문이 동일하다.

```
hello.html > ...
1 <script>
2
3 for (let i = 10; i>=1; i--) {
4   console.log(i);
5 }
6
7 </script>
8
```

▶	⊘	top
10		
9		
8		
7		
6		
5		
4		
3		
2		
1		
>		

```
hello.html > ...
1 <script>
2
3 let a = 1;
4
5 while (a <= 10) {
6   console.log(a);
7   a++;
8 }
9
10 </script>
11
```

▶	⊘	top
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
>		

1. 숫자 2개(a, b) 입력받고, a ~ b 까지 출력하기

1. 문자열 입력받고,  
“BBQ” 이라면 1 ~ 30 까지 합 출력  
“KFC” 이라면 1 ~ 30 까지 홀수 출력  
그 외라면 “MOMS” 출력

1. 세 숫자 입력 받고, if문으로 가장 큰 값을 출력하기



## C++스러운 예시

- `let arr = new arr();`
- `let arr = new arr("MINCO", 1, 2, 3, "MINCO");`

## Python스러운 예시

- `let arr = [];`
- `let arr = ["MINCO", 1, 2, 3, "MINCO"];`

```
hello.html X
hello.html > ...
1 <script>
2
3 let a = [];
4
5 a.push(3);
6 a.push(4);
7 a.push(5);
8
9 console.log(a);
10
11 </script>
```

```
▼ Array(3) ⓘ
  0: 3
  1: 4
  2: 5
  length: 3
  ▶ __proto__: Array(0)
```

## C언어와 비슷하다.

- index 접근이 가능하다.
- 배열 끼리 합칠때는 concat( ) 메서드 이용

```
test.html > script
1  <script>
2    let arr = [1, 2, 3, 4, 5];
3
4    for (let i = 0; i < arr.length; i++) {
5      console.log(arr[i]);
6    }
7  </script>
```

```
test.html > ...
1  <script>
2    let arr = [1, 2, 3];
3    let bbb = ['KFC', "BBQ"];
4
5    let ccc = arr.concat(bbb);
6
7    console.log(ccc);
8
9  </script>
```

1. [1, 2, 3, 1, 2, 3, 1, 2, 3] 을 하드코딩하고, 1 이 몇 개인지 Counting
1. ['A', 'E', 'W', 'Q', 'A']을 하드코딩하고, 'A'가 존재하는지  
0 또는 X 출력하기

기본적으로 C언어와 비슷하게 쓸 수 있다.

- function 으로 시작한다.

```
<> test.html > ...
1  <script>
2
3  function kfc()
4  {
5      alert("#");
6  }
7
8  kfc();
9
10 </script>
11
```

# 출력

```
<> test.html X
<> test.html > ...
1  <script>
2
3  function kfc(a, b)
4  {
5      return a + b;
6  }
7
8  a = kfc(5, 3);
9  alert(a);
10
11 </script>
12
```

8 출력

```
<> test.html X
<> test.html > ...
1  <script>
2
3  function kfc(a, b)
4  {
5      bbq();
6      return String(a + b);
7  }
8
9  function bbq()
10 {
11     alert("BBQ");
12 }
13
14 a = kfc(5, 3);
15 alert(typeof a);
16
17 </script>
18
```

BBQ 출력  
string 출력

## 함수 표현식

- 함수를 변수에 저장
- 함수가 저장된 변수를 호출하면, 해당 함수가 호출 됨

```
<> test.html > ...
1  <script>
2
3  let go = function() { alert("#"); };
4
5  go();
6
7  </script>
8
```

```
<> test.html > ...
1  <script>
2
3  function ABC() {
4      console.log("##");
5  }
6
7  bbc = ABC;
8  bbc();
9
10 </script>
11
```

```
<script>
let run = function() {
    for (let i = 0; i < 3; i++) {
        console.log("#");
    }
};

run();

</script>
```

first / second 변수에 무명함수를 저장한다.

- first와 second를 각각 한번씩 호출하기

first =

1 ~ 10 까지 출  
력하는 함수

second =

5 ~ -5 까지  
출력하는 함수

# 1장. DOM 제어

## 챕터의 **포인트**

- DOM Tree의 이해
- QuerySelector 다루기

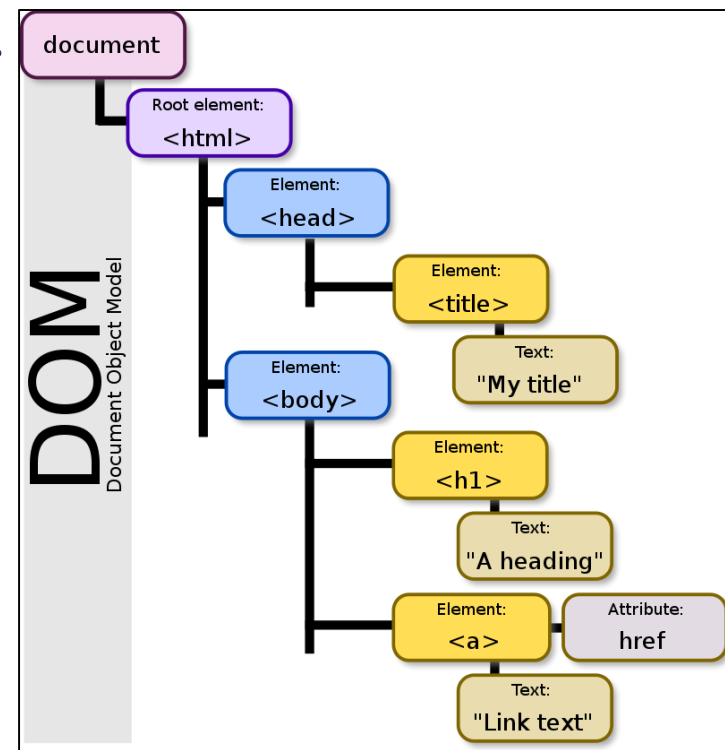


# DOM Tree의 이해

# Document Object Model

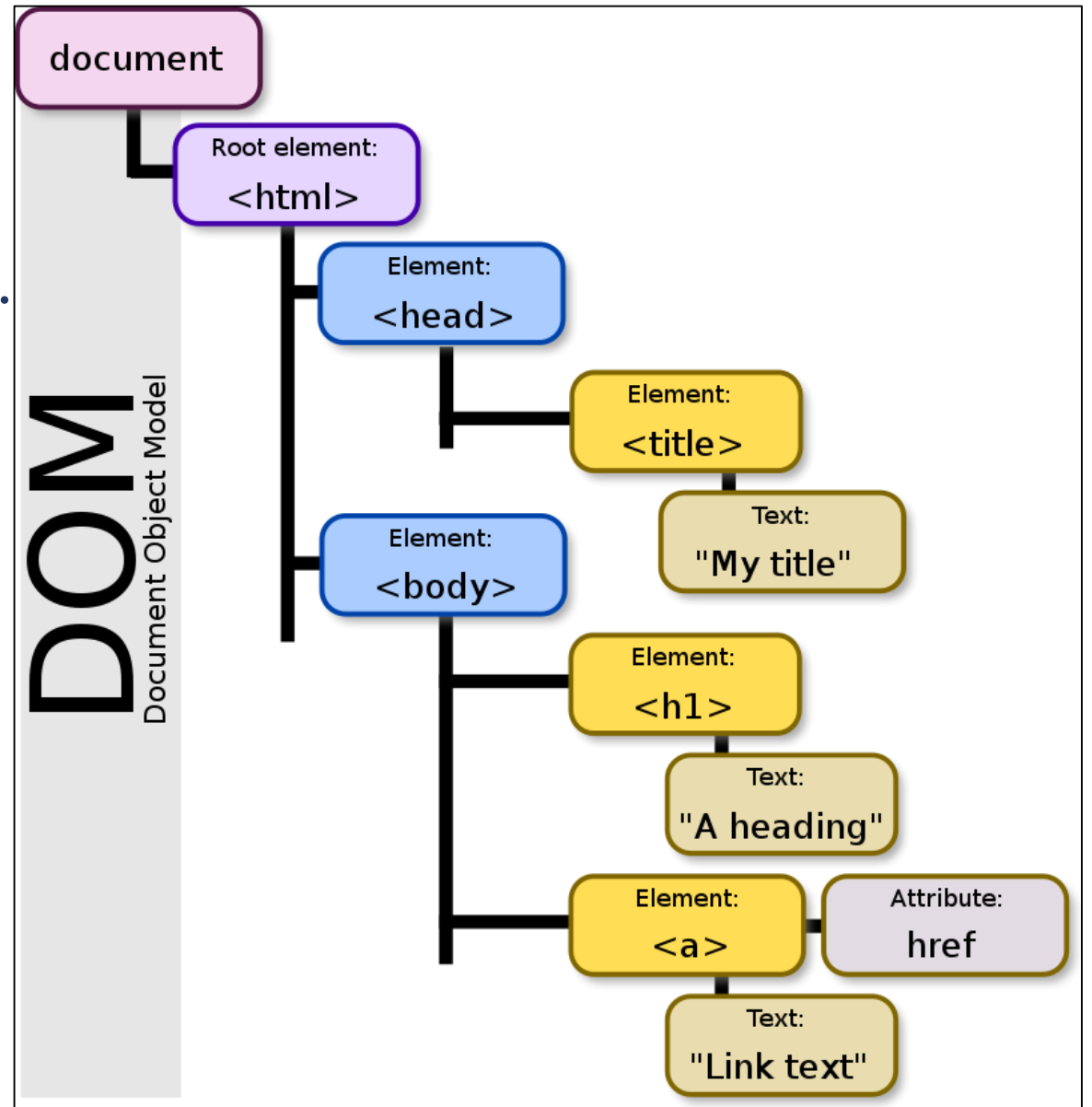
- HTML 문서의 구조적 표현 방법
- HTML 문서의 요소들을 Tree 구조로 표현한다.
- 웹브라우저가 HTML 문서를 읽고, DOM Tree를 생성한다.

```
<> test.html > ...  
1  <html>  
2  <head>  
3    <title>My title</title>  
4  </head>  
5  <body>  
6    <h1>A heading</h1>  
7    <a href="mcdonald.co.kr">Link text</a>  
8  </body>  
9  </html>
```



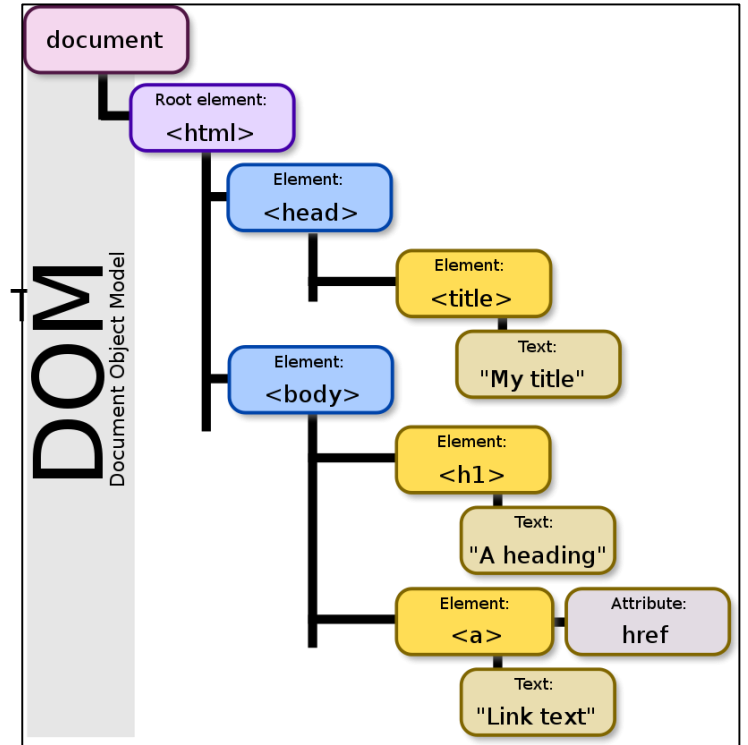
DOM 구조를 보고,  
직접 HTML 코드를 작성해보자.

- a tag에서 href는 **mcdonald.co.kr** 으로 한다.



## 개발자 모드에서 DOM Tree를 읽어보자.

- children 속성으로 접근 가능
  - document.children[0] : HTML Element
  - document.children[0].children[1] : Body Element
  - document.children[0].children[1].children[0] : <h1>
  - document.children[0].children[1].children[0].textContent :



## JavaScript에서 DOM 표현으로 특정 요소를 지정 가능

- 내가 원하는 HTML 요소의 값 읽기 가능

```
<> test.html X
<> test.html > html > script
1  <html>
2    <head>
3      <title>My title</title>
4    </head>
5    <body>
6      <h1>A heading</h1>
7      <a href="mcdonald.co.kr">Link text</a>
8    </body>
9
10   <script>
11     let a = document.children[0].children[1].children[1].textContent
12     alert(a);
13   </script>
14 </html>
15
16
```

이 페이지 내용:

Link text

확인

## JavaScript에서 DOM 표현으로 특정 요소를 지정 가능

- 내가 원하는 HTML 요소의 값 수정 가능

```
<> test.html > html > script
1  <html>
2    <head>
3      <title>My title</title>
4    </head>
5    <body>
6      <h1>A heading</h1>
7      <a href="mcdonald.co.kr">Link text</a>
8    </body>
9
10   <script>
11     let a = document.children[0].children[1].children[1];
12     a.textContent = "BBQ";
13   </script>
14 </html>
15
16
17
```

A heading

[Link text](#)



A heading

[BBQ](#)

## Head에 script가 삽입되는 경우

- Script를 모두 로딩 후 브라우저 렌더링을 시작한다.
- 페이지가 뜨는 속도가 느려 진다.
- 문서를 초기화하거나, 설정하는 가벼운 스크립트들을 넣는다.

## Body 맨 밑에 script가 삽입되는 경우

- DOM Tree가 완성된 상태에서 Script가 실행된다.
- 브라우저가 화면에 노출 된 상태에서 Script가 실행된다.

### defer

- body태그의 마지막에 스크립트를 넣는 경우 HTML 코드 파싱 -> 스크립트 로드 후 실행
- HTML 파싱과 동시에 스크립트를 로드하고 HTML 파싱이 끝난 후 바로 로드 된 스크립트 실행
- 대부분의 경우 가장 성능면에서 유리하다
- head 태그안에 js 코드를 넣고 defer을 사용한다.



JavaScript으로 HTML 문서에 있는 값을  
마음대로 편집하거나, 값을 읽을 수 있다.

- 사용자 입력에 따라 화면을 바꿀 수 있다.
- 특정 조건에 따라 문서를 변경할 수 있다.
- HTML의 사용자 입력을 JavaScript로 읽을 수 있다.

# QuerySelector 다루기

## DOM으로 특정 노드를 선택하는 다양한 방법 중 Best

- `querySelector` : CSS Selector를 이용하여, 특정 DOM 영역 1개 선택하기
- `querySelectorAll` : CSS Selector를 이용하여, 해당하는 모든 DOM 영역 Array로 가져오기

## QuerySelector

- “A heading” 값을 alert으로 출력하기
- “Link text” 값을 BBQ WORLD 로 변경하기

```
<> test.html > ...
1  <html>
2  <head>
3    <title>My title</title>
4  </head>
5  <body>
6    <h1>A heading</h1>
7    <a href="mcdonald.co.kr">Link text</a>
8  </body>
9  </html>
```

## CGV의 무비차트 페이지

- 페이지 소스보기
- Local로 HTML 소스코드 복사하기

뒤로(B) Alt+왼쪽 화살표

앞으로(F) Alt+오른쪽 화살표

새로고침(R) Ctrl+R

다른 이름으로 저장(A)... Ctrl+S

인쇄(P) Ctrl+P

전송(C)...

한국어(으)로 번역(T)

페이지 소스 보기(V) Ctrl+U

프레임 소스 보기(V)


프레임 새로고침(F)

검사(N) Ctrl+Shift+I

무비차트


영작만 보기

NO.1



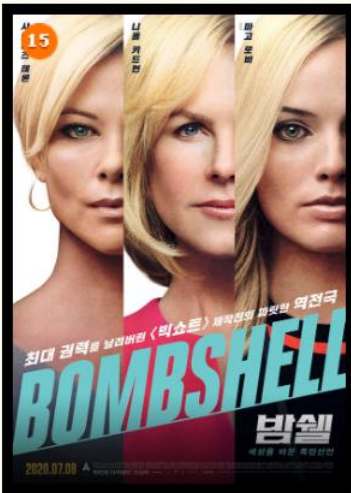
반도

NO.2



소년시절의 너

NO.3



밤쉘-세상을 바꾼 폭탄선언

## split 메서드

- 구분문자를 기준으로 array 객체를 생성해준다.

## for of 문법

- 파이썬에서 for in과 동일
- C++에서 ranged for문과 동일

```
<> test.html > script
1  <script>
2
3  let a = "BTS|KFC|MC|BBQ";
4  let arr = a.split("|");
5
6  for (let str of arr) {
7    console.log(str);
8  }
9
10 </script>
```

BTS
KFC
MC
BBQ
>

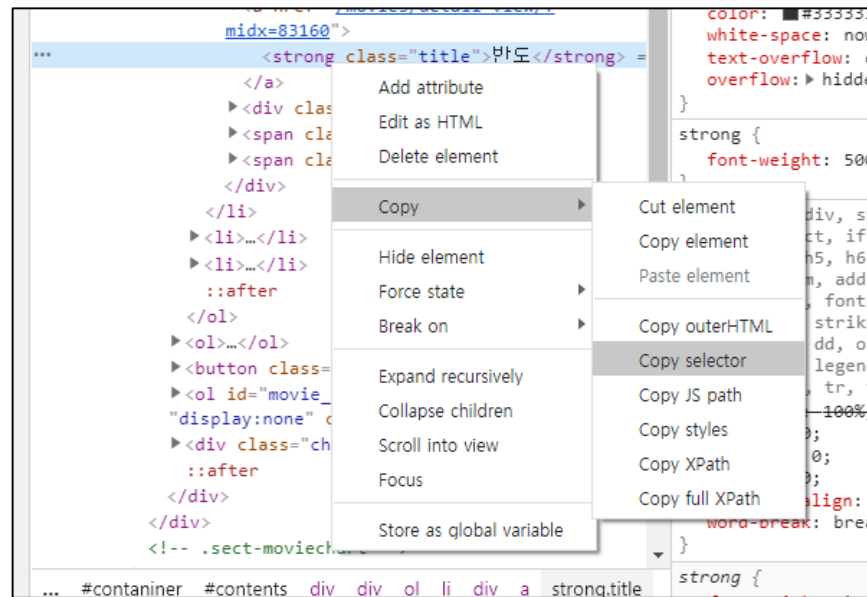
## 개발자 도구를 이용하여 쉽게 CSS Select 알아내기

- 인기순위 1등 영화 제목만 가져와 alert으로 출력하기
- 인기순위 1등 영화 제목을 변경해보기

## Selector 사용처

- CSS 개발시
- 크롤링
- DOM 선택에 사용

selector를 해석해보자



## QuerySelector 사용하기

1. 인기순위 1등 영화 제목 / 예매율 / 개봉일자 가져와 console로 출력하기
2. 인기순위 2등 영화 정보 변경하기
  - 영화제목 : 삼겹살과 그대라면
  - 예매율 : 1등 예매율 - 5%
  - 개봉일 : 1985.07.07 로 변경
3. 인기순위 3등과 1등 정보 변경하기
  - 이미지 SWAP
  - URL SWAP
  - 글자 정보 모두 SWAP



## 하나의 정보가 아닌, 모든 정보를 List로 가져오는 방법

- 모든 이미지를 가져와보자.

```
<script>
  let a = document.querySelectorAll(".thumb-image");

  for (let i = 0; i < a.length; i++) {
    console.log(a[i].innerHTML);
  }
</script>
```

<u>83160_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-15">15세 이상</span>
<u>83353_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-15">15세 이상</span>
<u>83099_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-15">15세 이상</span>
<u>83344_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-15">15세 이상</span>
<u>77201_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-all">전체</span>
<u>83285_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-15">15세 이상</span>
<u>83567_320.jpg</u>	<small>test.html:1746</small>  <span class="ico-grade grade-12">12세 이상</span>

## Input type checked 된 속성들을 가져오기

- querySelectorAll을 활용해서 가져오기
- CSS 선택자에 대해 추가로 알아보기

## QuerySelectorAll 사용하기

1. 차트에 표시된 모든 영화 제목을 array로 저장한다.
2. 모든 영화 제목들을 \* 로 모자이크 처리한다.
3. 모든 영화의 예매율을 30% 올려준다.

## 2장. Event 와 DOM

## 챕터의 • • • 포인트

- Event Driven 구현
- Create Element
- Change와 Key Event

# Event Driven 구현

## 브라우저에서 Event (사건) 발생시 지정된 함수가 호출되도록 함

1. GOGO 함수 만든다.
1. 클릭시 GOGO 함수가 자동 호출되도록 “등록”해둔다.
1. 사용자가 버튼을 클릭하면, GOGO 가 호출된다.

Event Driven : Event 발생시 만들어둔 Script가 동작되게 하는 방식

## 콜백 함수

1. 다른 코드의 Argument로 함수 이름을 넘겨주는 함수
2. 소스코드의 직접 호출이 아닌, 이벤트로 인해 호출되는 함수

## 이벤트 핸들러

- 이벤트 발생시 호출되는 함수이다.
- “콜백 함수” 중 용도를 더 명확하게 나타낸 용어이다.

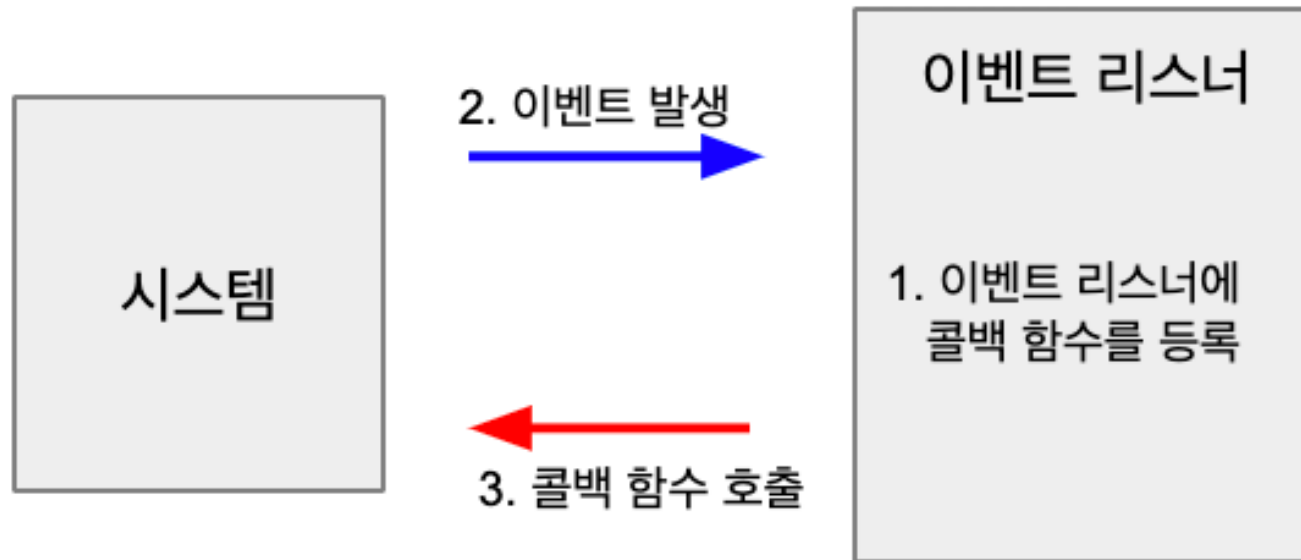
## 이벤트 리스너

- 이벤트가 발생하는지 감지하여, 예약된 이벤트 핸들러를 호출해주는 객체



## 이벤트 리스너에 함수를 등록한다.

- addEventListener 메서드를 사용하여 **이벤트 핸들러 (콜백함수)**를 등록한다.
- addEventListener('이벤트이름', '핸들러이름');



gogo 함수를 이벤트 리스너에 등록한다.

```
<> test.html > ...  
1  <button id="btn">고고</button>  
2  
3  <script>  
4  
5      const btn = document.querySelector("#btn");  
6      btn.addEventListener('click', gogo);  
7  
8      function gogo() {  
9          alert("GOGO");  
10     }  
11  
12 </script>
```

이 페이지 내용:

GOGO

확인

## HTML 에서 즉시 JavaScript 함수를 호출한다.

- onclick 속성을 이용한다 : inline 방식
  - 장점 : 간단
  - 단점 : js와 HTML의 완전 분리가 안됨 (유지보수 측면 좋지 않음)

```
<> test.html > ...
1  <button onclick=gogo()>고고</button>
2
3  <script>
4      function gogo() {
5          alert("GOGO");
6      }
7
8  </script>
```

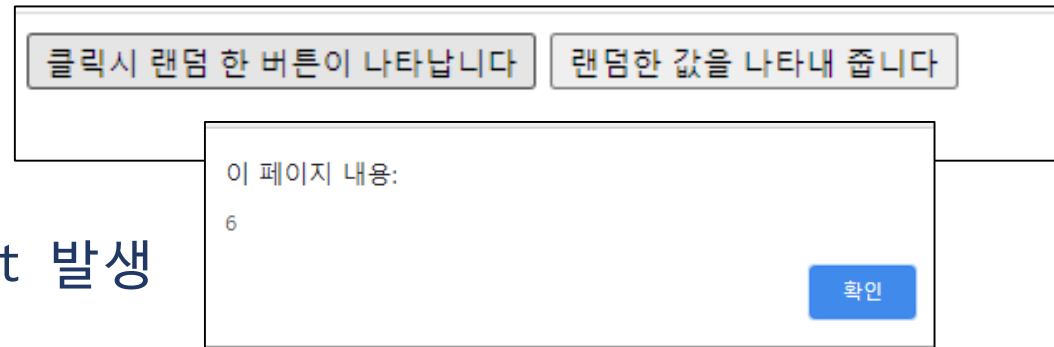
inline 방식, 이벤트 처리

```
<> test.html > ...
1  <button id="btn">고고</button>
2
3  <script>
4      const btn = document.querySelector("#btn");
5      btn.addEventListener('click', gogo);
6
7      function gogo() {
8          alert("GOGO");
9      }
10
11 </script>
```

addEventListener 방식,  
이벤트 처리 **(권장 방식)**

## 버튼 2개를 생성

- inline으로 이벤트핸들러 BBQ 호출
- addEventListener로 BBQ 호출
- BBQ 함수 : 1 ~ 10 까지 랜덤 정수 값 alert 발생



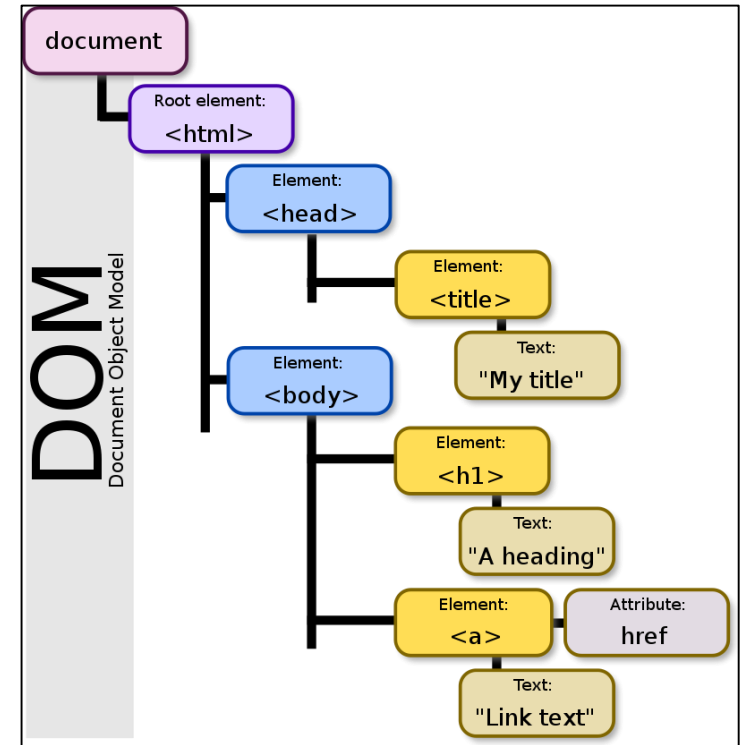
## Math.random 메서드

- Math.random( ) : 0 ~ 1 사이 소수점 랜덤 값 생성
- Math.random( ) \* 10 : 0 ~ 10 사이 소수점 랜덤 값 생성
- Math.ceil(Math.random( ) \* 10) : 1 ~ 10 사이 정수 값 생성

# Create Element

## JavaScript로 HTML 생성하기

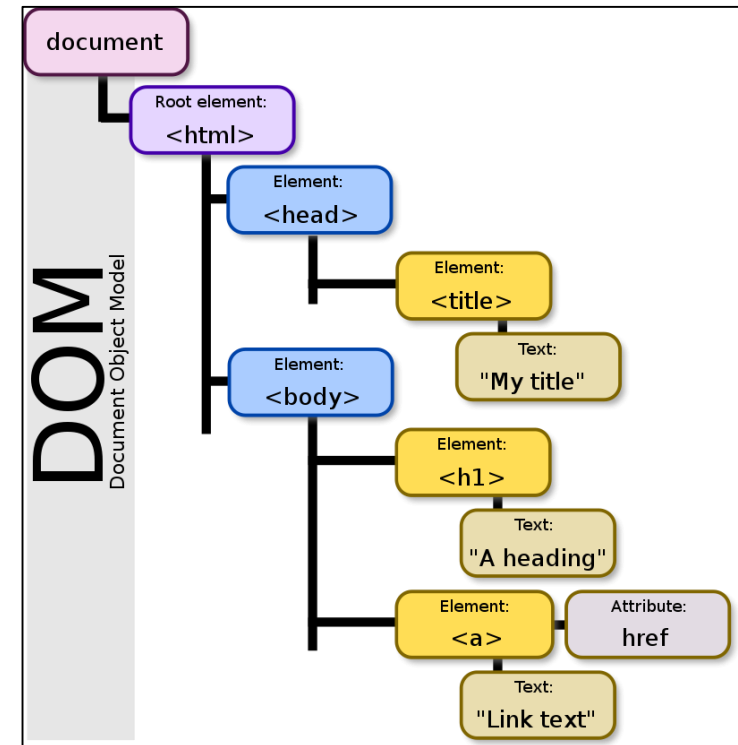
- DOM Tree에 노드를 추가하면, Element가 생성된다.
- 과정
  1. Element와 Text 객체를 생성한다.
  2. 특정 노드의 자식 요소로 추가한다.



## document.createElement( 'Tag 이름' )

- Element Node를 생성한다.
- ex) let btn = document.createElement('button');

element  
<button>

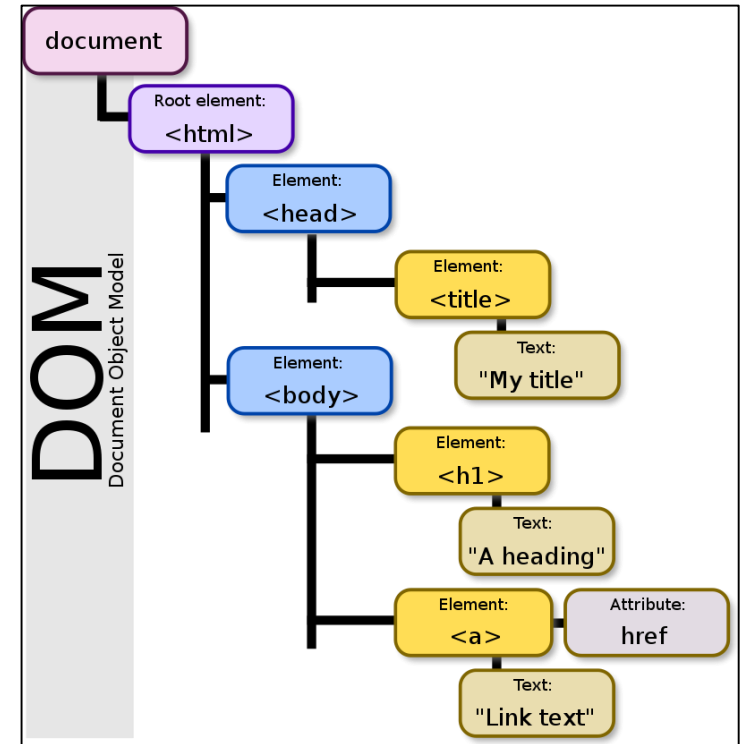


## document.createTextNode( '들어갈 Text' )

- Text Node를 생성한다.
- ex) let txt = document.createTextNode( "민코월드" );

Text:  
"민코월드"

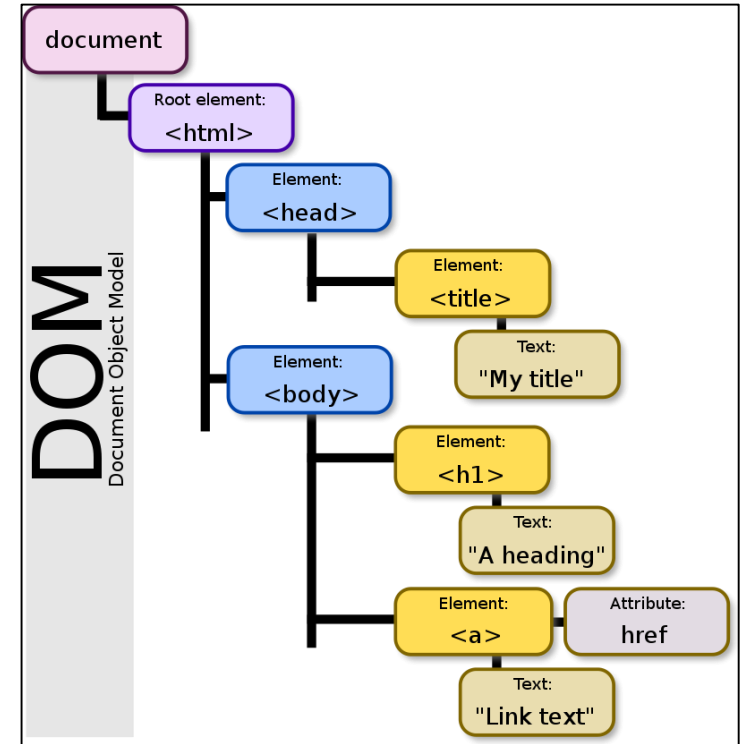
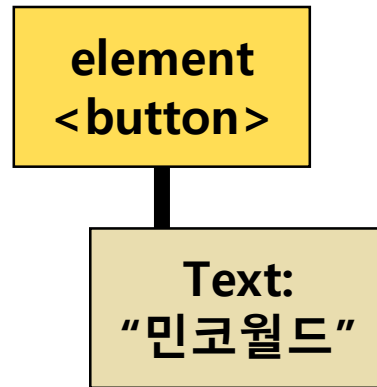
element  
<button>





## 객체.append( Target노드 )

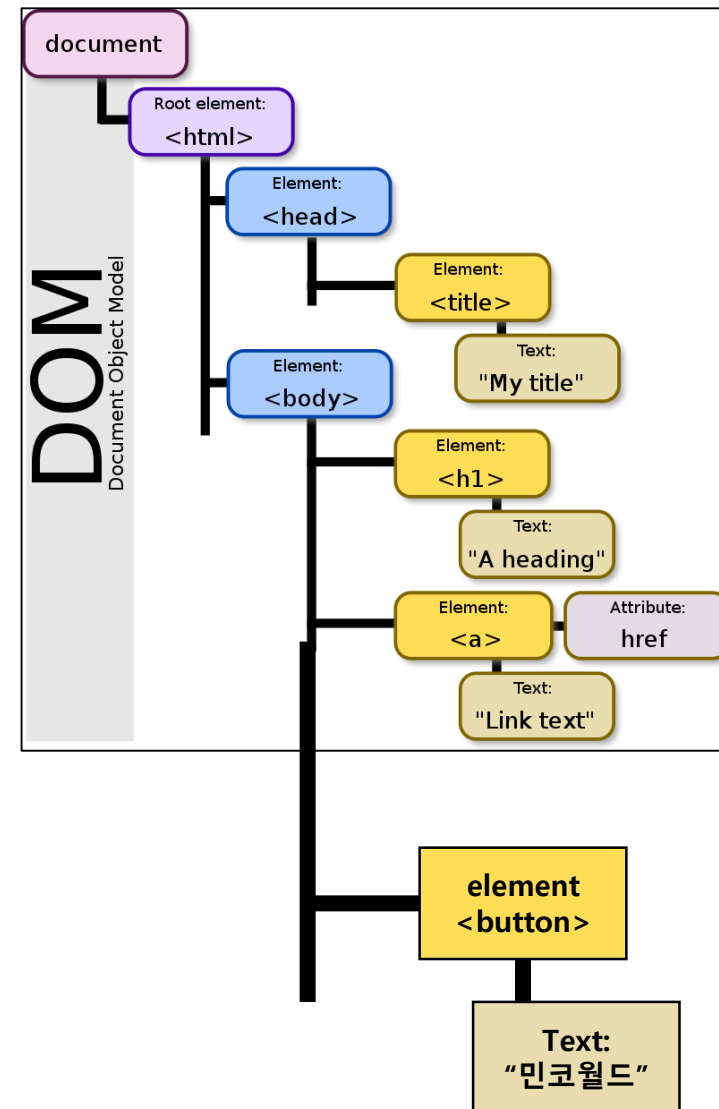
- 객체 밑으로 자식 노드를 추가한다.
- ex) btn.append( txt );



## Body 가장 아래 만든 element 추가하기

- 원하는 곳 밑에 Element를 추가한다.
- append는 가장 마지막 자식에 추가한다.

```
<> test.html > ...
1  <html>
2    <head>
3      <title>My title</title>
4    </head>
5    <body>
6      <h1>A heading</h1>
7      <a href="mcdonald.co.kr">Link text</a>
8
9      <script>
10         let btn = document.createElement('button');
11         let txt = document.createTextNode('민코월드');
12         btn.append(txt);
13         document.querySelector("body").append(btn);
14      </script>
15    </body>
16  </html>
```



## 객체.setAttribute(속성이름, 속성 값)

- 간단하게 속성을 부여할 수 있다.

```
<> test.html > ...
1  <html>
2    <head>
3      <title>My title</title>
4    </head>
5    <body>
6      <h1>A heading</h1>
7      <a href="mcdonald.co.kr">Link text</a>
8
9      <script>
10         let btn = document.createElement('button');
11         let txt = document.createTextNode('민코월드');
12         btn.setAttribute("id", "BTS");
13         btn.append(txt);
14         document.querySelector("body").append(btn);
15       </script>
16
17     </body>
18   </html>
```

```
▶ <a href="mcdonald.co.kr">...</a href="mcdonald.co.k
... <button id="BTS">민코월드</button> == $0
    </body>
  </html>
```

## 1. for를 이용하여 5개의 button을 추가한다.

- class 이름을 지정한다.
- Text 이름은 “버튼1 ~ 5” 으로 한다.

## 2. “안녕” alert 출력

- 아무 버튼이나 눌렀을 때 alert가 출력된다.

### 1. “안녕 버튼 n” alert 출력








- 1번 버튼을 눌렀다면 “안녕 버튼 1” 을 출력한다.
- n번 버튼을 눌렀다면 “안녕 버튼 n” 을 출력한다.

## 네비게이션 메뉴 가져오기

- querySelector 및 querySelectorAll을 활용해서 해당 내용들의 text들을 가져오기
- 배열 선언하기
- 해당 배열에 네비게이션 메뉴의 text들을 배열안에 넣기



네이버를

 메일  카페  블로그  지식iN  쇼핑  Pay  TV 사전 뉴스 증권 부동산 지도 영화 VIBE 책 웹툰 더보기 ▼

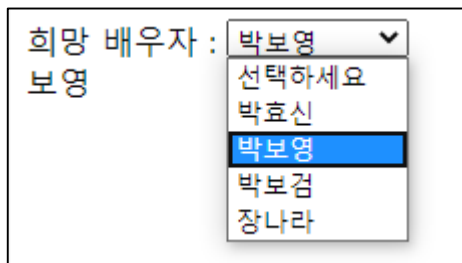
## Change와 Key Event

input 에서 값이 바뀔 때 발생하는 이벤트

```
test.html > ...
1  <input>
2  </div></div>
3
4  <script>
5      const inp = document.querySelector("input");
6      const div = document.querySelector("div");
7
8      inp.addEventListener('change', gogo);
9
10     function gogo(e) {
11         |     div.textContent = e.target.value;
12     }
13 </script>
14
```

ABC

## select 에서도 사용 가능



```
<> test.html > script
1  <label>희망 배우자 :
2      <select>
3          <option value="">선택하세요</option>
4          <option value="효신">박효신</option>
5          <option value="보영">박보영</option>
6          <option value="보검">박보검</option>
7          <option value="나라">장나라</option>
8      </select>
9  </label>
10
11 <div></div>
12
13 <script>
14     const sel = document.querySelector("select");
15     const txt = document.querySelector("div");
16
17     sel.addEventListener('change', gogo);
18
19     function gogo(e) {
20         document.querySelector("div").textContent = e.target.value;
21     }
22 </script>
```



## 키를 눌렀을 때 반응하는 Event

- keydown
- keyup

```
<input type="text" class="input1">
<div class="result1"></div>
<script>
  const inputElement1 = document.querySelector(".input1");
  const divElement1 = document.querySelector(".result1");
  inputElement1.addEventListener("keydown", function(e){
    divElement1.textContent = e.code;
  });
```

s|

Enter

Id및 password가 공백일때 회원가입 방지하기

아이디

비밀번호

회원가입하기

# 내일 방송에서 만나요!

삼성 청년 SW 아카데미