SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

삼성 청년 SW 아카데미

Web Programming 기초

<알림>

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

5장. CSS Training

CSS 마스터란?

Confidential

원하는 대로 CSS로 표현하기

1. 이미지를 원하는 대로 표현하기
2. 원하는 Layout대로 자신이 생각한 대로 나타내는 것
3. Tag를 원하는 형태로 그리는 것

미션을 수행하는 방식으로 훈련을 시작

훈련 규칙

Confidential

한 단계가 끝나면, 손을 들어 검사를 받는다.

- 한 단계씩 미션을 Clear 하면 된다.
- CSS 디버깅에 힘을 최대한 쏟는다.
- 인터넷 참고 얼마든지 가능하다.

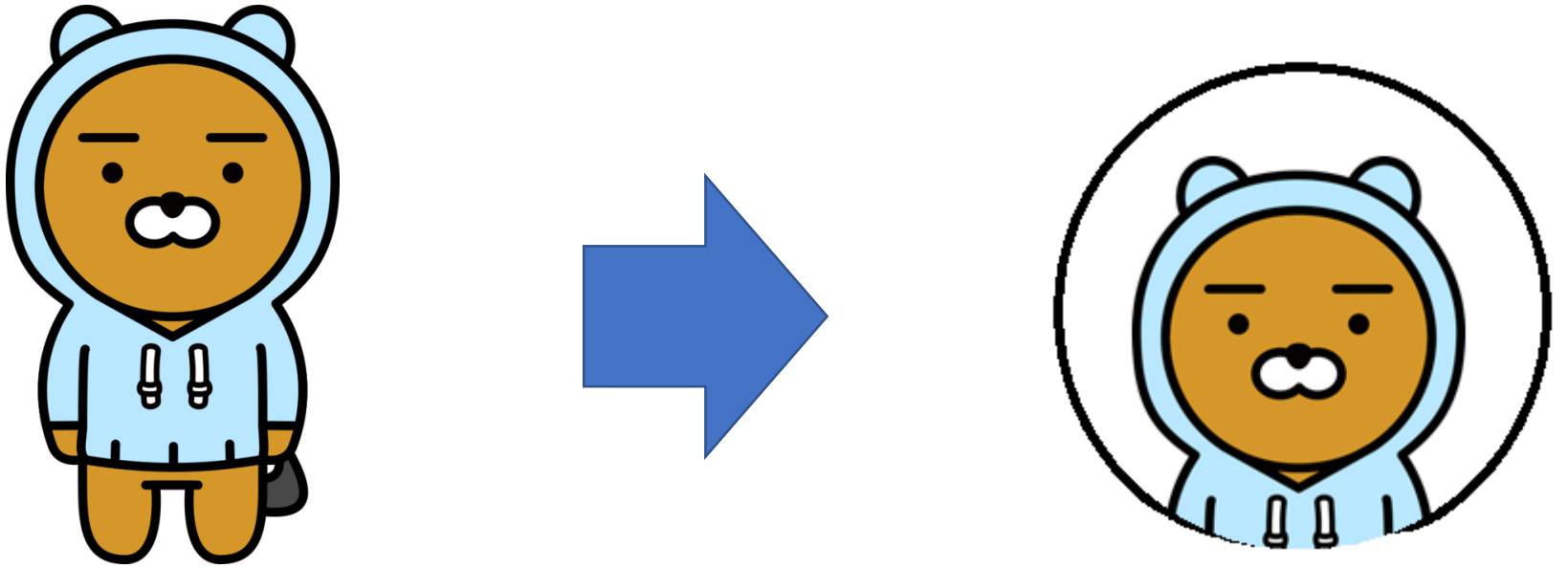
CSS 디버깅시

- CSS 디버깅은 웹 개발 중 가장 어렵다. (가장 노가다스러움)
- 최대한 노력 후, 정 안된다면 다음 미션으로 넘어가도록 하자.
  (그래도 잘 안되면 Matermost로 압축해서 남길 것)

미션 1. 카카오톡 프로필 사진

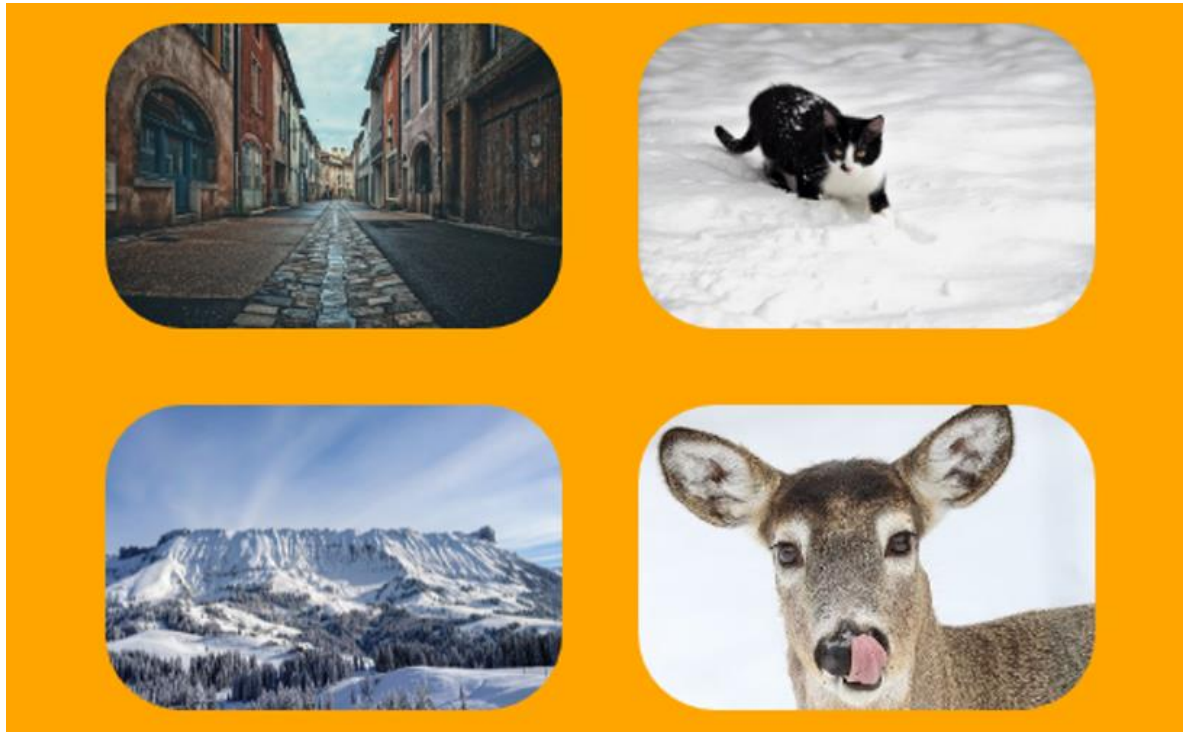
카카오톡 프로필 사진 형태 만들기

- 이미지 하나를 선정하여, 원형 태두리로 만든다.



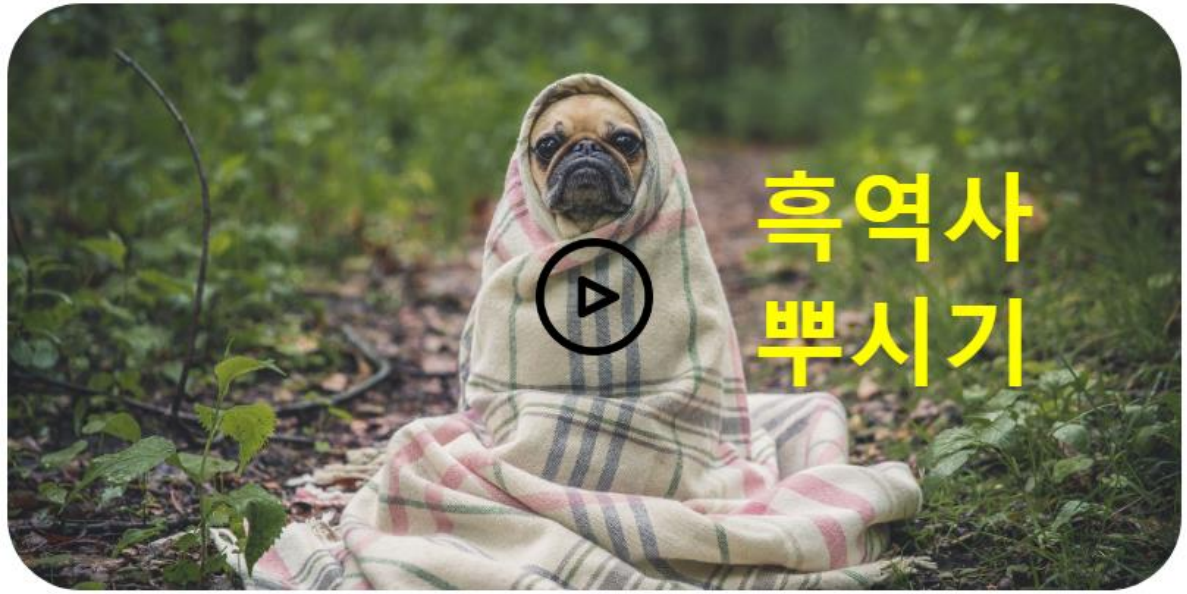
미션2. 윈도우 이미지

- 네 개의 사진을 둥군 사각 형태로 표현하기
  - 이미지 4개 파일을 준비한다.
  - Window 형태의 이미지를 구성한다.
  - 간격은 일정한 간격을 유지한다.



미션3. 동영상 이미지 만들기

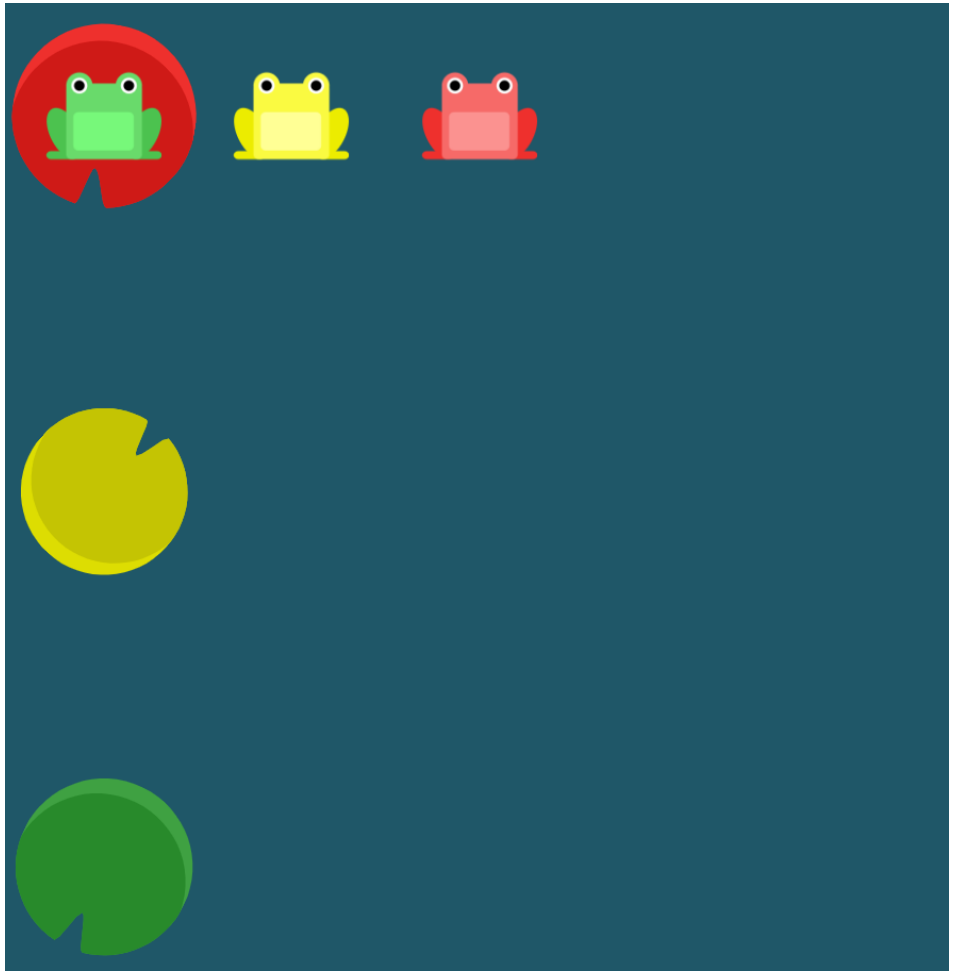
- 배경 이미지 선택 (이미지 아무거나)
- 테두리 둥글게
- 플레이 버튼 아이콘 가운데 배치
- 글씨 이미지 위 띄우기



[과제] flexbox 개구리 정복하기

https://flexboxfroggy.com/#ko

- flex를 활용한 개구리 배치 게임 사이트
- 모든 문제를 마무리 한 후 마지막 페이지를 캡쳐 해서



SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

6장. CSS 심화

챕터의 포인트

- Transform
- Transition
- Animation

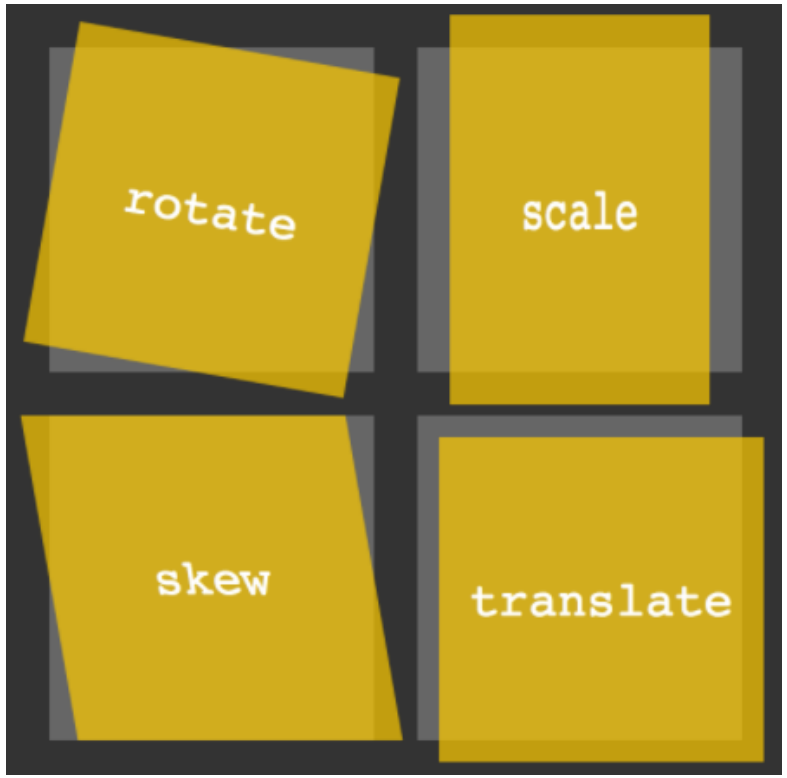
SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Transform

Transform

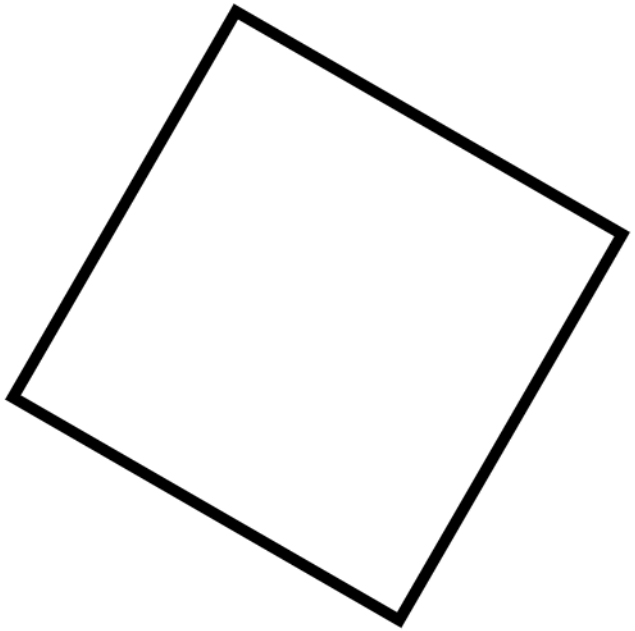
- element 요소에 회전, 크기 조절, 기울이기, 이동 및 3D효과를 부여 할 수 있다.
- GPU를 사용해서 매끄럽게 보인다.



https://www.htmldog.com/references/css/properties/transform/

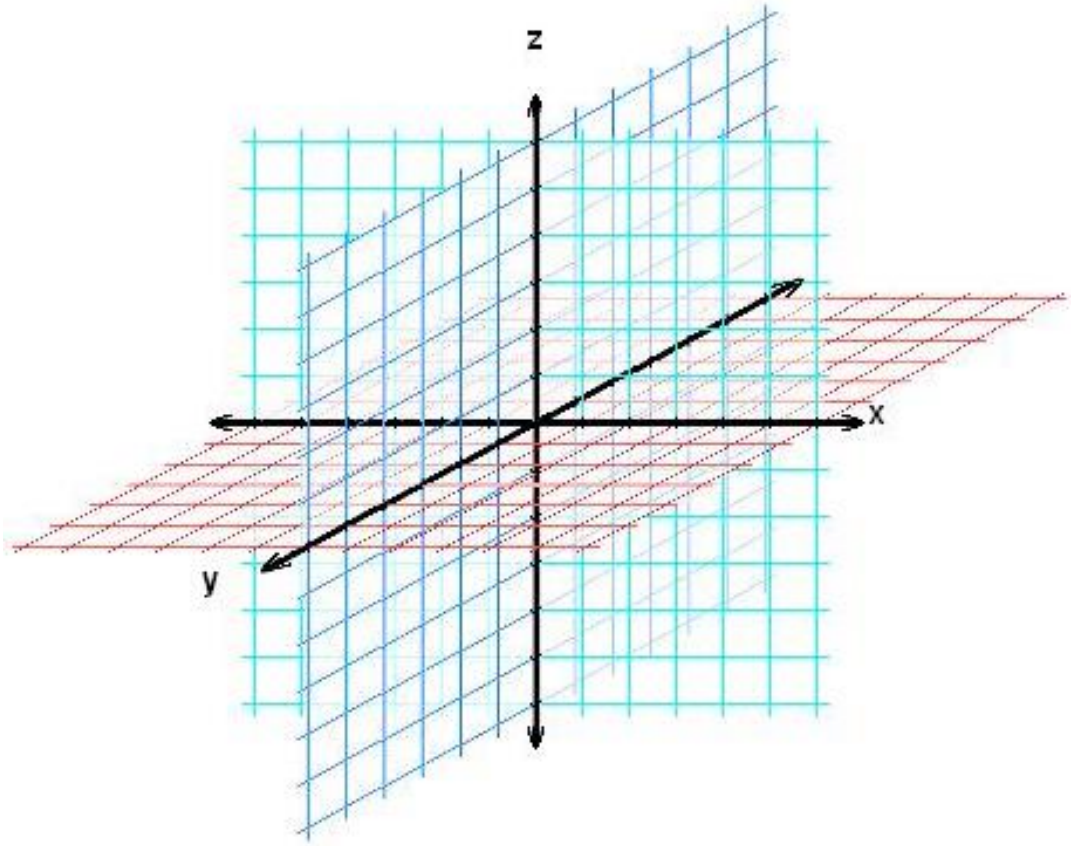
CSS Transform 을 활용한 박스 움직이기

- transform 속성을 활용해서 현재 위치에서 이동 시키기
  - X축 80px Y축 60px 이동
  - 30도 만큼 회전 시키기
  - 1.5배 확대시키기



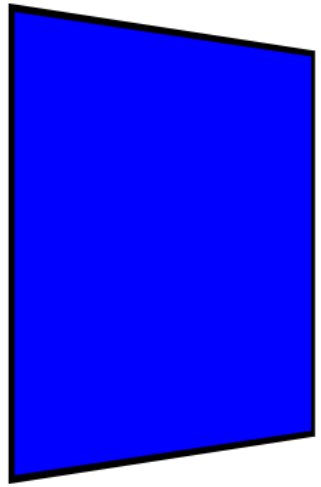
Transform : perspective

- 원근감을 부여해 좌표공간을 변형한다.
- 3D 공간에서의 회전, 확대, 이동, 비틀기 등이 가능해 진다.



CSS Transform 을 활용한 3D 요소 만들기

- transform perspective 와 transform 요소들 활용하기
- :hover시에 요소 변화시키기
  - width: 300px; height: 300px
  - background-color: blue
  - Y축으로 45도만큼 회전
  - Z축으로 – 50px 만큼 이동



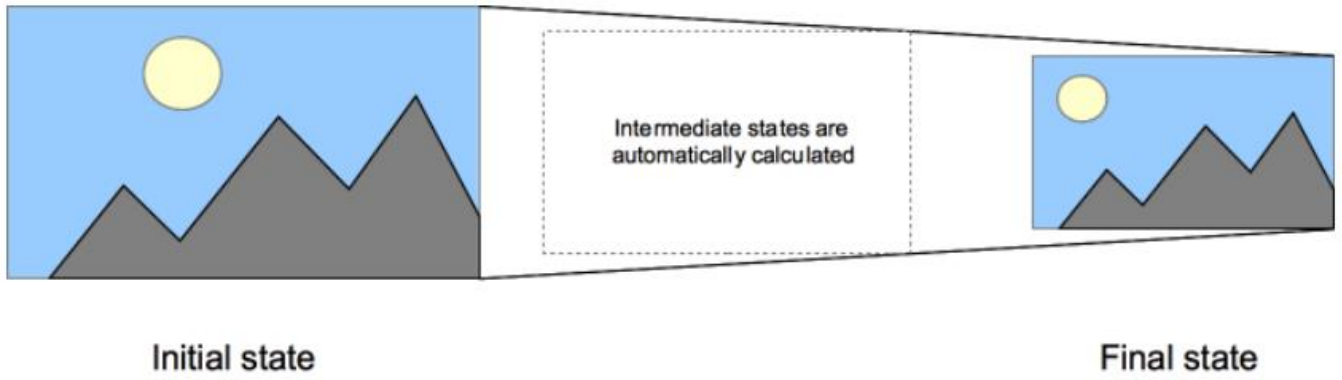
SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

Transition

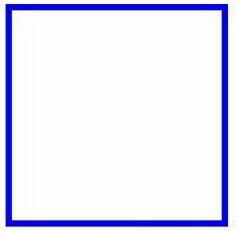
Transition

- 한가지 상태에서 다른 상태로 변화하는 것
- 값이 변화할 때 일정 시간에 걸쳐 일어나게 한다.
- 애니메이션의 속도 조절



Transition

- transition-property
  - transition을 적용해야 하는 CSS속성의 이름을 명시
- transition-duration
  - 애니메이션의 재생 시간
- transition-timing-function
  - 가속도 값
- transition-delay
  - transition 의 동작시간을 delay 이후에 진행



돌아가는 원반 만들기

- pixabay에서 이미지 다운
- transition 속성을 활용해서 원판 회전시키기

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

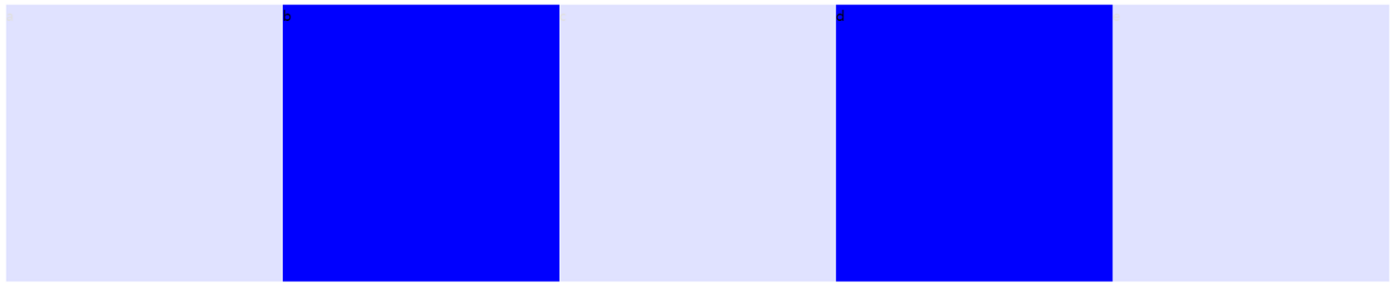
Animation

CSS Animation

- 구간을 퍼센트별로 지정해서 특정 CSS 이벤트들을 지정해줄 수 있다.
- 구간 반복이 가능하다.

[도전] 사라지는 애니메이션 구현하기

- 사라지는 애니메이션 구현하기
  - .box-wrapper > .box * 6 구조
    - .box-wrapper는 display : flex로 정의
    - .box의 width:300px; height:300px; background : blue;
    - animation을 활용하여 보였다가 보이지않게 변하는 애니메이션 작성하기



SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

7장. JavaScript 기초

챕터의 포인트

- JavaScript 개요
- JavaScript 기본 문법

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

JavaScript 개요

JavaScript가 쓰이는 이유

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.

소스코드를 HTML 넣으면 장점

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.

JavaScript가 지금 각광받는 이유

JavaScript는 예전부터 있어왔다.

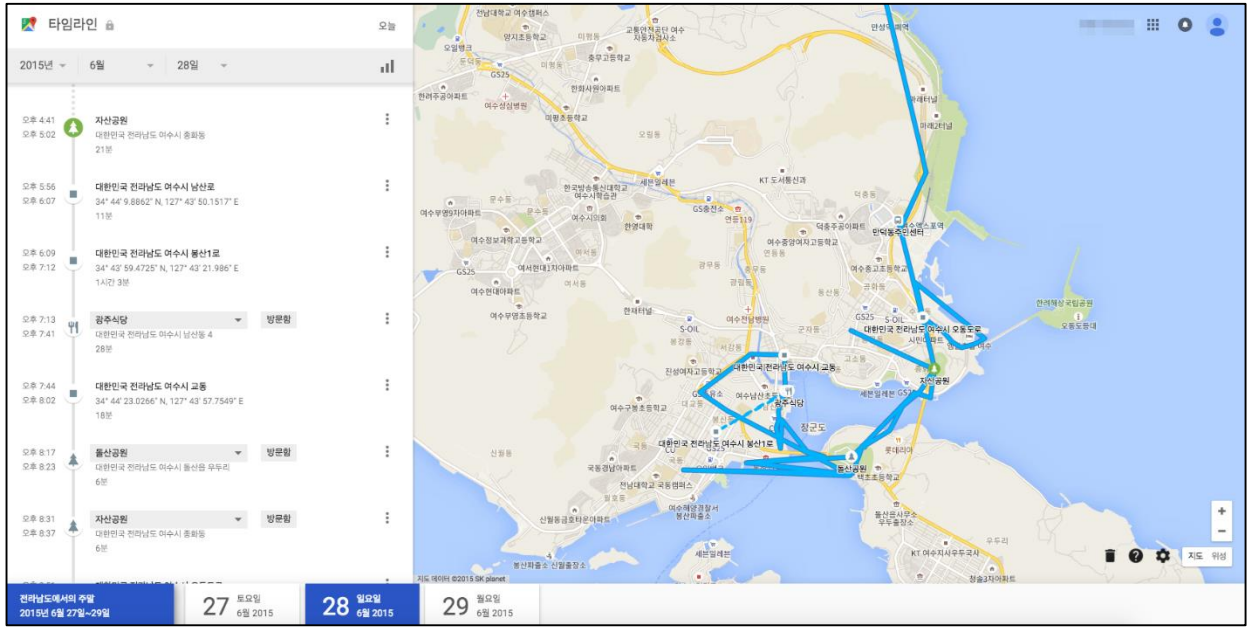
- 1990년대부터 다양한 웹브라우저에 JavaScript 인터프리터를 내장하였다.
- 사용처
  - HTML에서 클릭시 Form 내용 조건 검사
  - ActiveX 실행
  - if / for / 변수 / 배열 / for문 수행 가능
  - 특정 영역 값 변경 가능

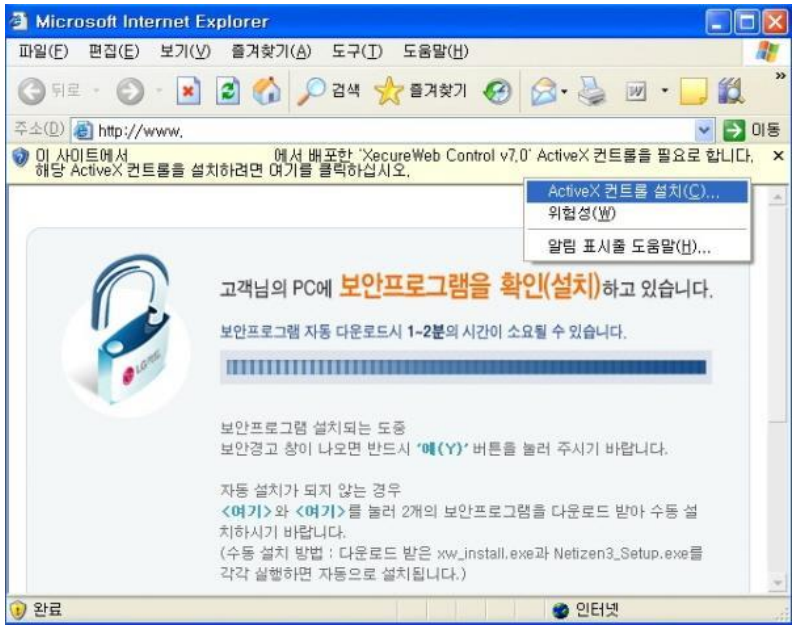
JavaScript의 인기 원인 1 : AJAX

Confidential

Asynchronous JavaScript and XML

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





AJAX, 비통기통신이란?

Confidential

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

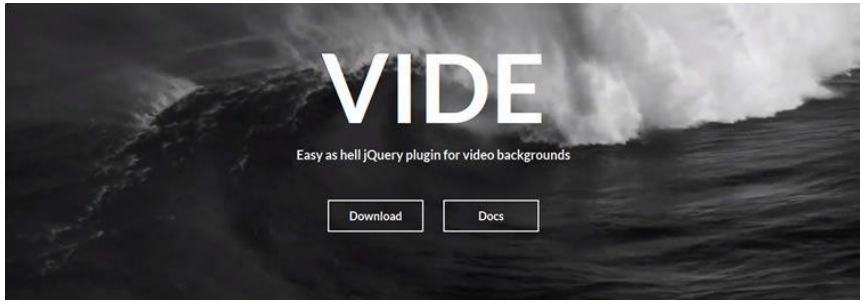
- 서버의 부담을 줄일 수 있다.

JavaScript의 인기 원인 2 : jQuery

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 인기 원인 3 : Node.js

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

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

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

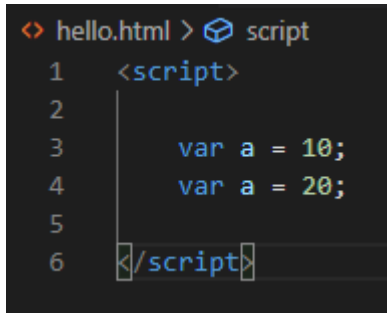
var는 사장되었다.

var 대신 사용하는 문법

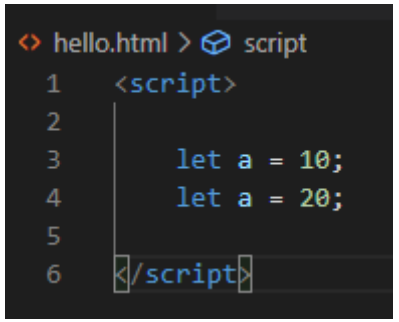
- var
  - 재 할당이 가능하다
  - 함수 레벨 스코프

- let : C언어 변수와 비슷
  - 재 할당이 가능하다
  - 블록 레벨 스코프

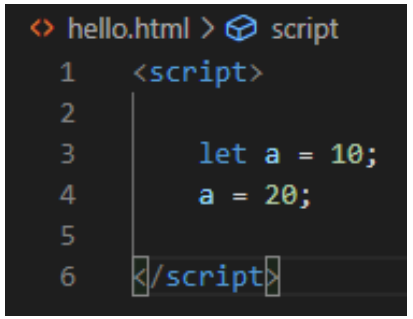
- const : C언어 const와 비슷
  - 재 할당 불가
  - 블록 레벨 스코프



가능



불가능



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
```

JavaScript 자료형

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

자바스크립트 원시 자료형 - Number

Number

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

```
typeof(100)
'number'
typeof(-300.1)
'number'
```

자바스크립트 원시 자료형 - Boolean

Boolean

- true/false 로 나누는 진리값을 뜻한다.

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

null 과 undefined

- null
  - 값이 비어있다고 명시하는것

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

자바스크립트 참조 자료형 - Array

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
```

자바스크립트 참조 자료형 1 : 함수 선언 식

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 > ...
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>
```

8 출력

```
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 > ...

```
<script>

let go = function() {   alert("#");   };

go();

</script>
```

test.html > ...

```
<script>

function ABC() {
    console.log("##");
}

bbc = ABC;
bbc();

</script>

```

```
<script>

let run = function() {

    for (let i = 0; i < 3; i++) {
        console.log("#");
    }

};

run();

</script>
```

자바스크립트 참조 자료형 - Object

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

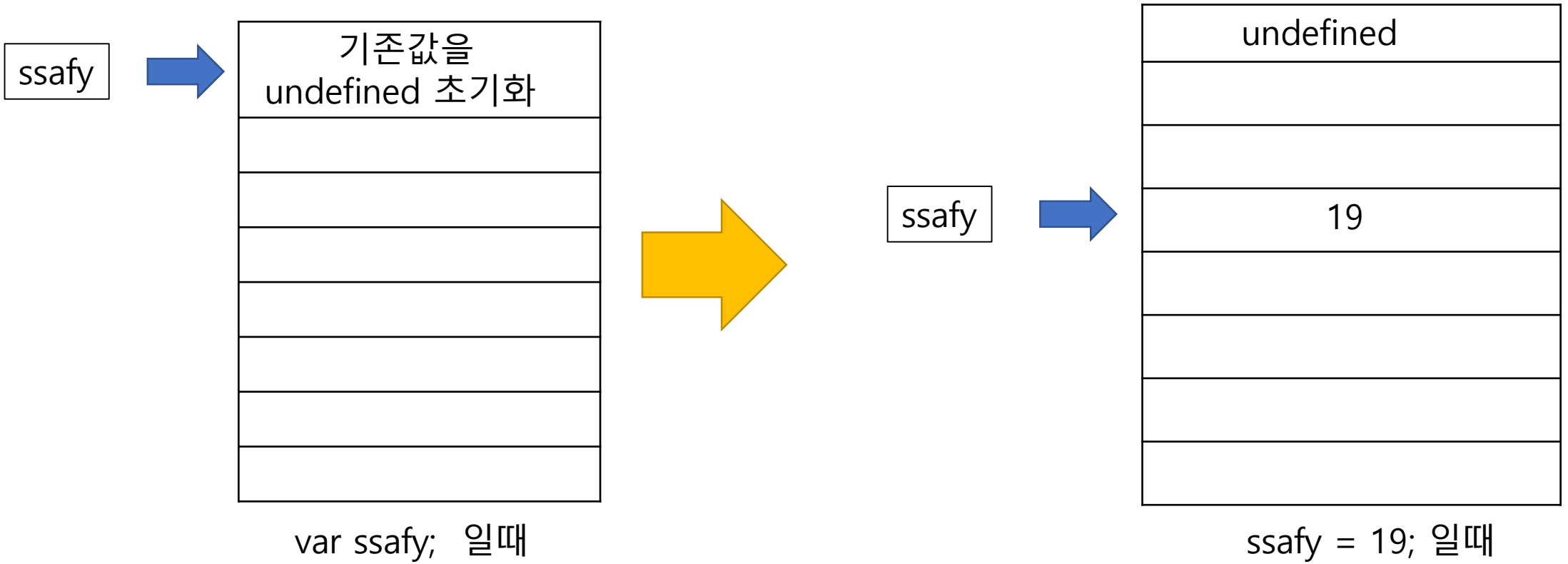
- 객체 값에 접근하기 위한 방법
  - result['a']
  - result.a 로 접근 가능하다.

```
const result = {a: 5, b: 10}
undefined
result['a']
5
result.a
5
```

원시 자료형의 메모리 할당 과정


```
var ssafy = 19;
```

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



변수의 호이스팅

런타임 이전에 변수의 “선언문“들 만 먼저 끌어져 올라오는 행위

- var ssafy; let ssafy; const ssafy;
- let과 const에서는 호이스팅이 동작하는걸 막는다.

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

var 사용시 undefined

```
> console.log(ssafy)
  let ssafy = 19;
✕ ▶Uncaught                VM1747:1
  ReferenceError: ssafy is
  not defined
      at <anonymous>:1:13
```

let 사용시 error

```
> console.log(ssafy)
  const ssafy = 19
✕ ▶Uncaught                VM2548:1
  ReferenceError: ssafy is
  not defined
      at <anonymous>:1:13
```

const 사용시 error

함수의 호이스팅

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");
}
```

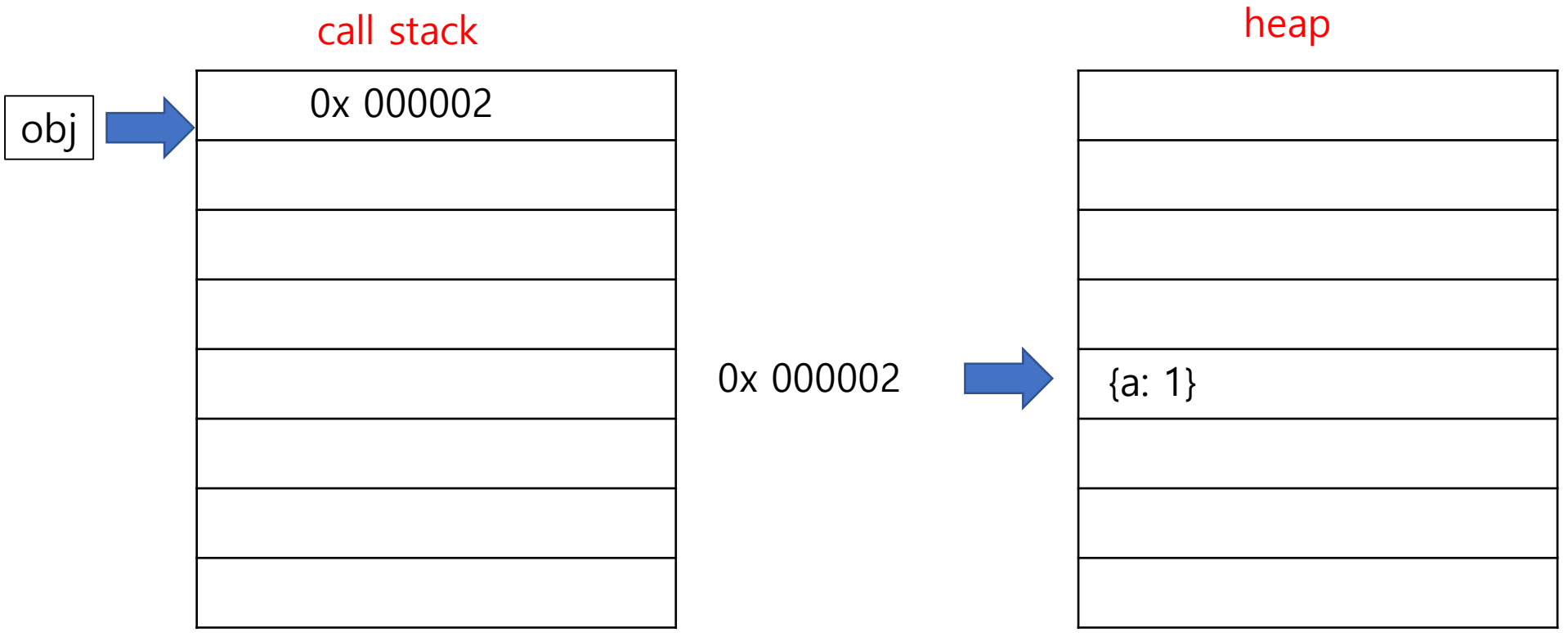
```
▶Uncaught ReferenceError: hello is not defined
    at <anonymous>:1:1
```

함수 표현식: 변수의 호이스팅

참조 자료형의 메모리 할당 과정


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

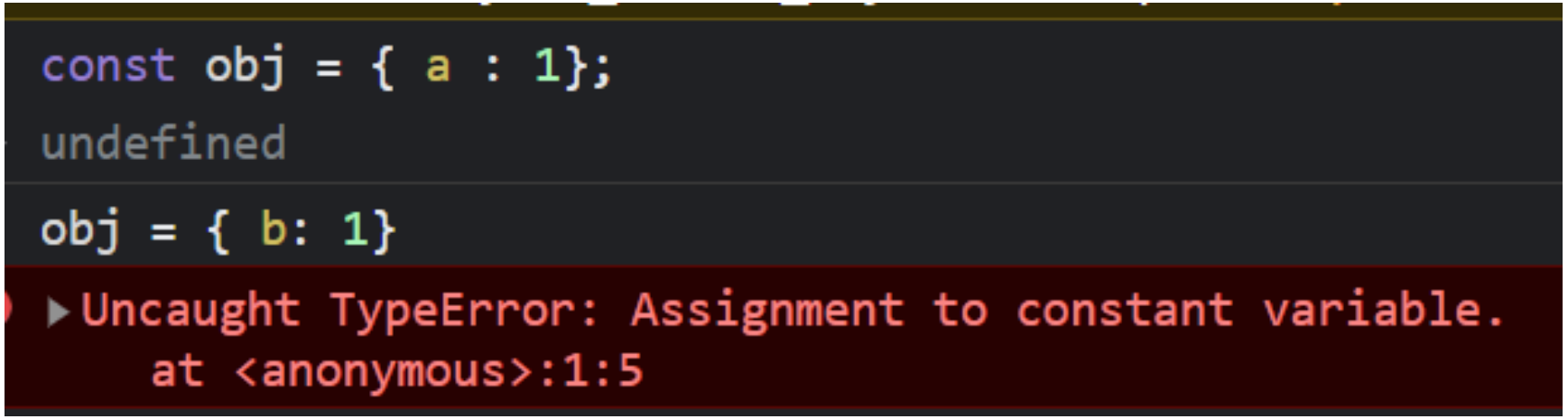
- 변수는 call stack메모리에 저장(원시값)
- 객체는 heap 메모리에 저장



참조 자료형에서 값 바꾸기

const 로 선언된 참조 자료형에서 값 바꾸기

- call stack메모리의 값을 변경시키려 했기때문에 에러가 발생.

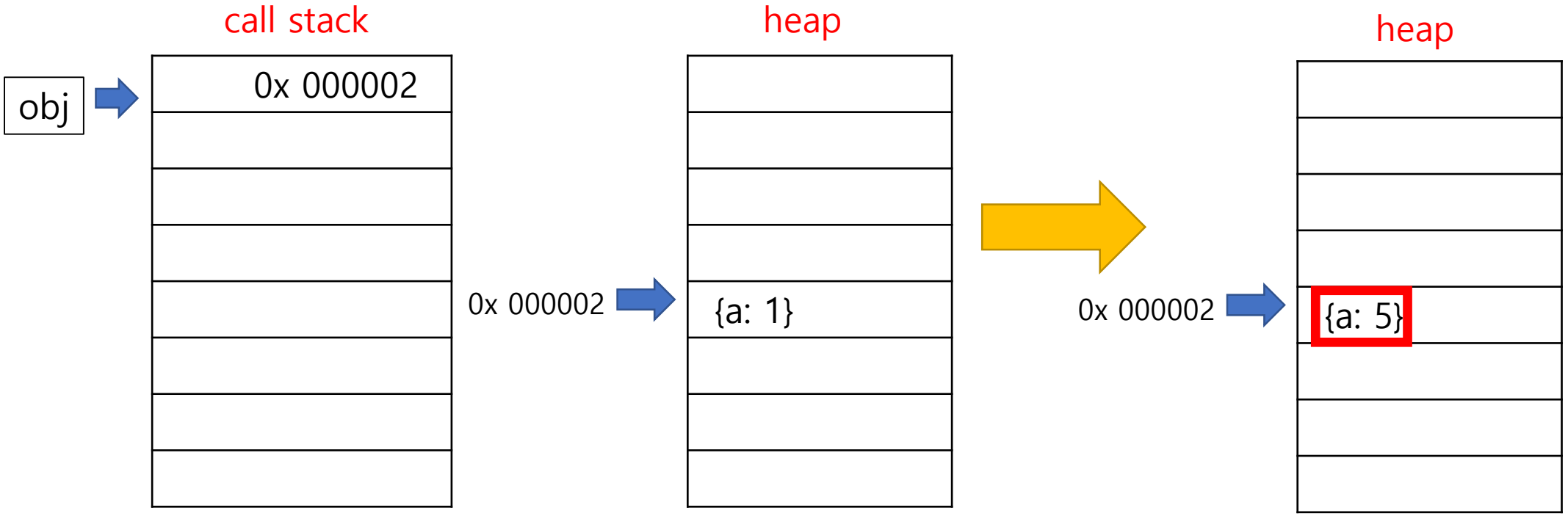


참조 자료형에서 값 바꾸기

const 로 선언된 참조 자료형에서 값 바꾸기

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

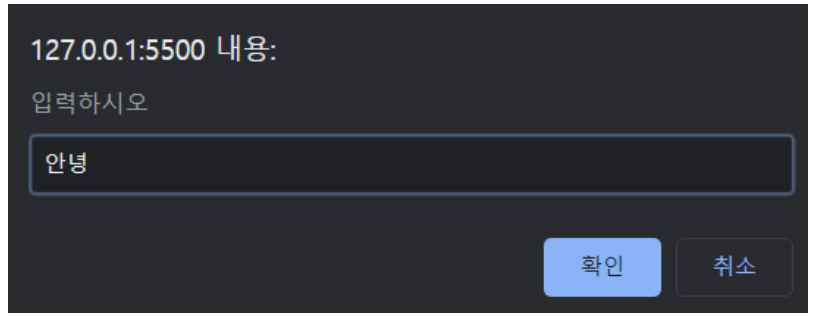
```
> const obj = { a : 1};
< undefined
> obj.a = 5;
< 5
> console.log(obj);
  ▶ {a: 5}
```

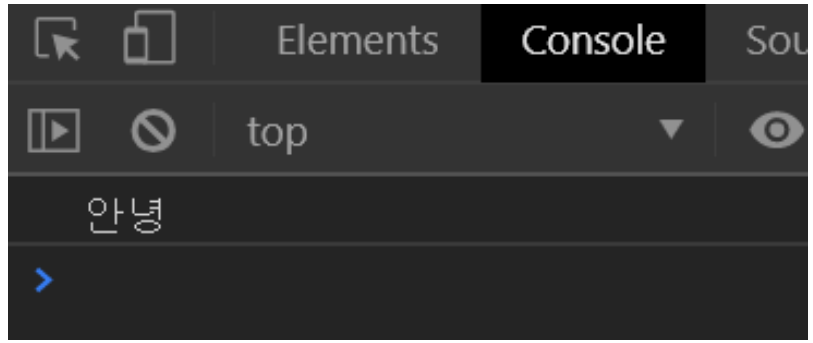


prompt로 입력받기

- 문자열 입력 받고, 콘솔 출력하기

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

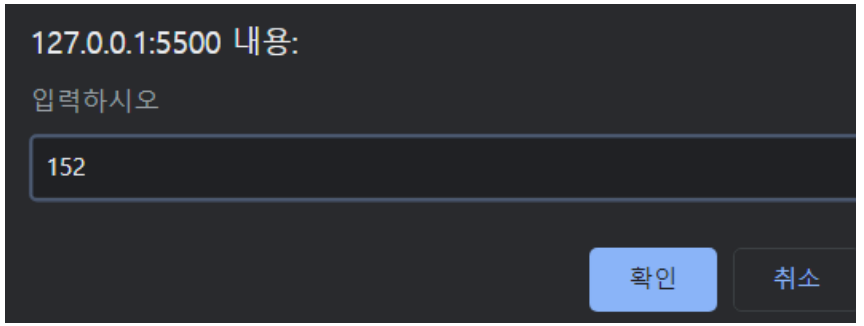





alert로 출력하기

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

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

[참고] ECMAScript

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.

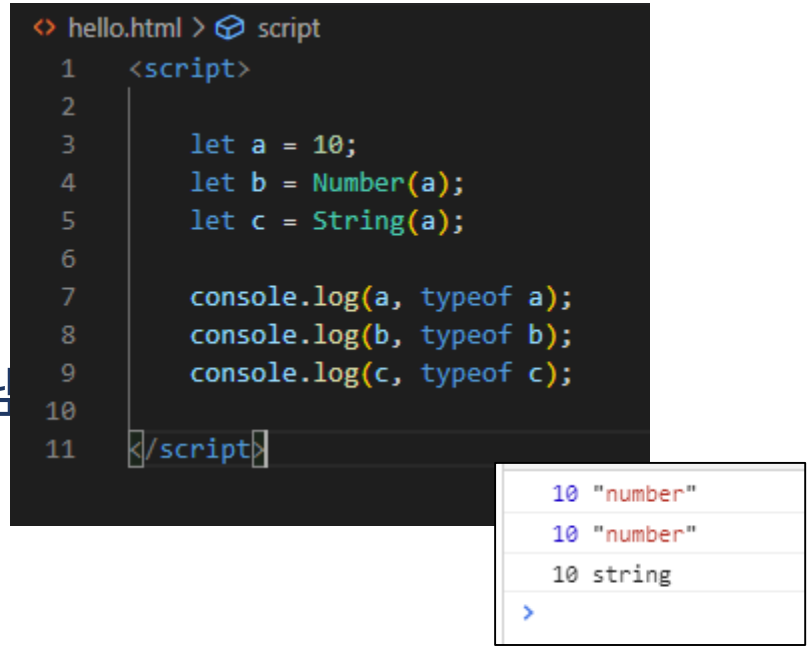
Type 변환

Number와 String 형 변환이 자유롭다.

- Number(변수)
- String(변수)

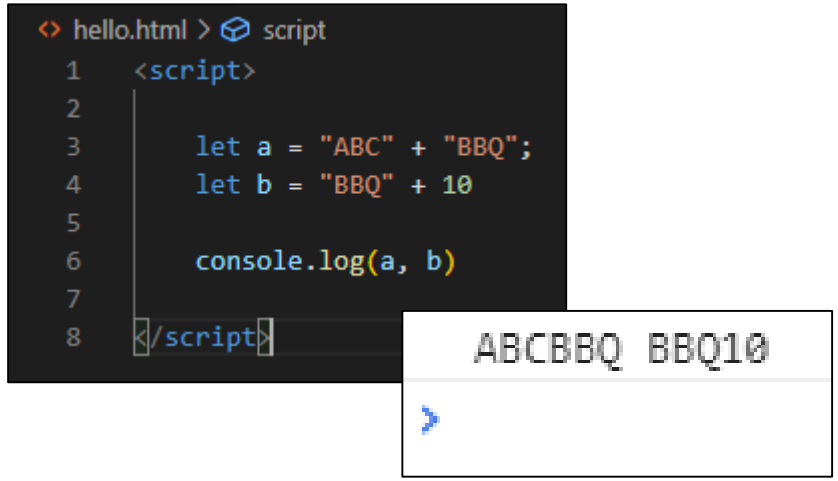
[참고] 세미콜론

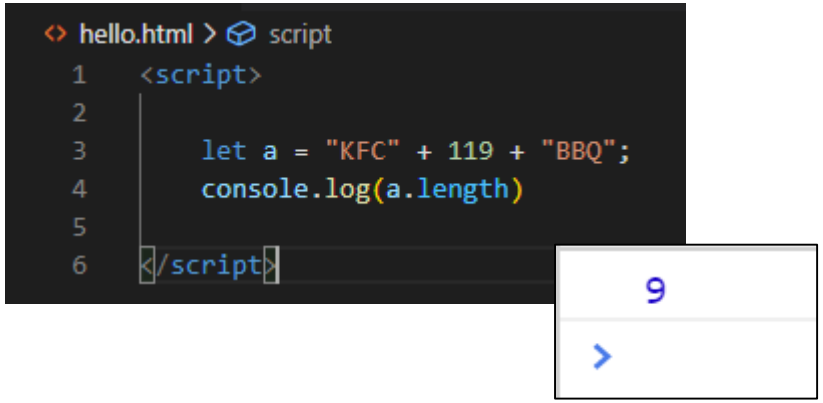
- 파이썬 : 세미콜론 없는 것이 기본 (있어도 에러 안남)
- JavaScript : 세미콜론 있는 것이 기본 (없어도 에러 안남



파이썬 처럼 '+' 연산자로 문자열 붙이기 가능

- 문자열.length : 문자열 길이가 저장된 속성





if 문

C언어와 동일하다.

- >, >=
- <, <=

같다와 다르다 주의

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

JavaScript 에서는
같다와 다르다를 === / !== 를 사용하자

hello.html

```
<script>

    let a = 5;
    let b = 10;

    if (a < b) {
        alert("B BIG");
    }
    else if (a === b) {
        alert("A == B");
    }
    else {
        alert("A BIG");
    }

</script>
```

while과 for 문

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
>
```

[도전] 다음 세 문제를 구현할 것

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

array 구현

C++ 스러운 예시

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

Python 스러운 예시

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

hello.html

```
<script>

let a = [];

a.push(3);
a.push(4);
a.push(5);

console.log(a);

</script>
```

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

array 객체 예시

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>
```

[도전] 다음 두 문제를 구현 할 것

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
   O 또는 X 출력하기

함수를 만드는 방법 1 : 함수 선언식

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 > ...
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>
```

8 출력

```
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

함수를 만드는 방법 2 : 함수 표현식

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>
```

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>
```

[도전] 함수표현식 사용하기

first / second 변수에 무명함수를 저장한다.

- first와 second를 각각 한번씩 호출하기

first =

second =

SAMSUNG SW ACADEMY FOR YOUTH

함께가요 미래로!
Enabling People

내일 방송에서 만나요!

삼성 청년 SW 아카데미