

삼성청년 SW·AI아카데미

Web 기초

<알림>

본 강의는 삼성청년SW·AI아카데미의 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.

Day2-1. CSS 심화

display

flex

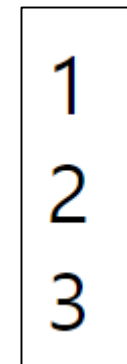
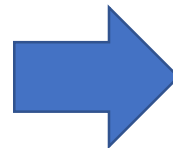
- 메인 축을 기반으로 정렬을 진행한다.
- 하나의 flex-container와 flex-item 들로 구성된다
- flex를 선언하면 부모요소들은 flex-container
- 자식 요소들은 자동으로 flex-item이 된다

flex-start

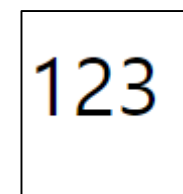
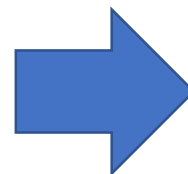
center

flex-end

```
<div class="container">  
  <div class="item">1</div>  
  <div class="item">2</div>  
  <div class="item">3</div>  
</div>
```



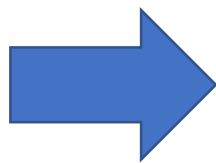
```
.container{  
  display: flex;  
}
```



Flex를 활용해서 메뉴 상단 바 만들어보기 1

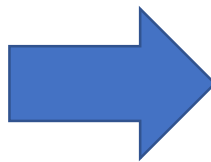
- 제한시간 6 분

- header 부분 border bottom에 gray 로 지정
- 메뉴1, 2, 3의 width는 120px height는 30px로 지정한다.



Flex를 활용해서 메뉴 상단 바 만들어보기

- 제한시간 10 분
 - Logo의 font는 36px로 지정한다
 - 메뉴왼쪽정렬, 로고 중앙정렬, 회원가입 로그인 오른쪽 정렬 (flex 속성 활용)
 - 각 메뉴와 로고, 회원가입 별 중앙 가운데 정렬하기



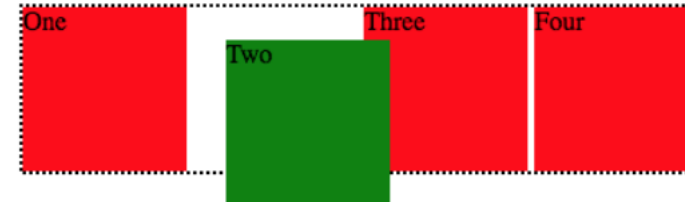
position

CSS Position 활용하기

position: static



position: relative

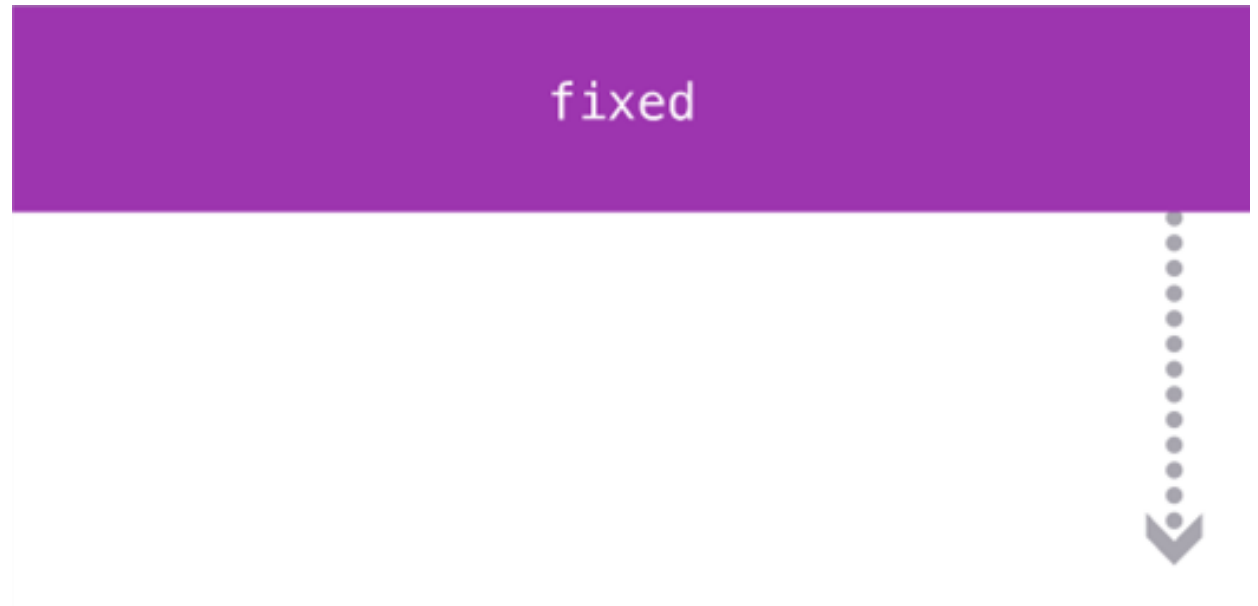


position: absolute



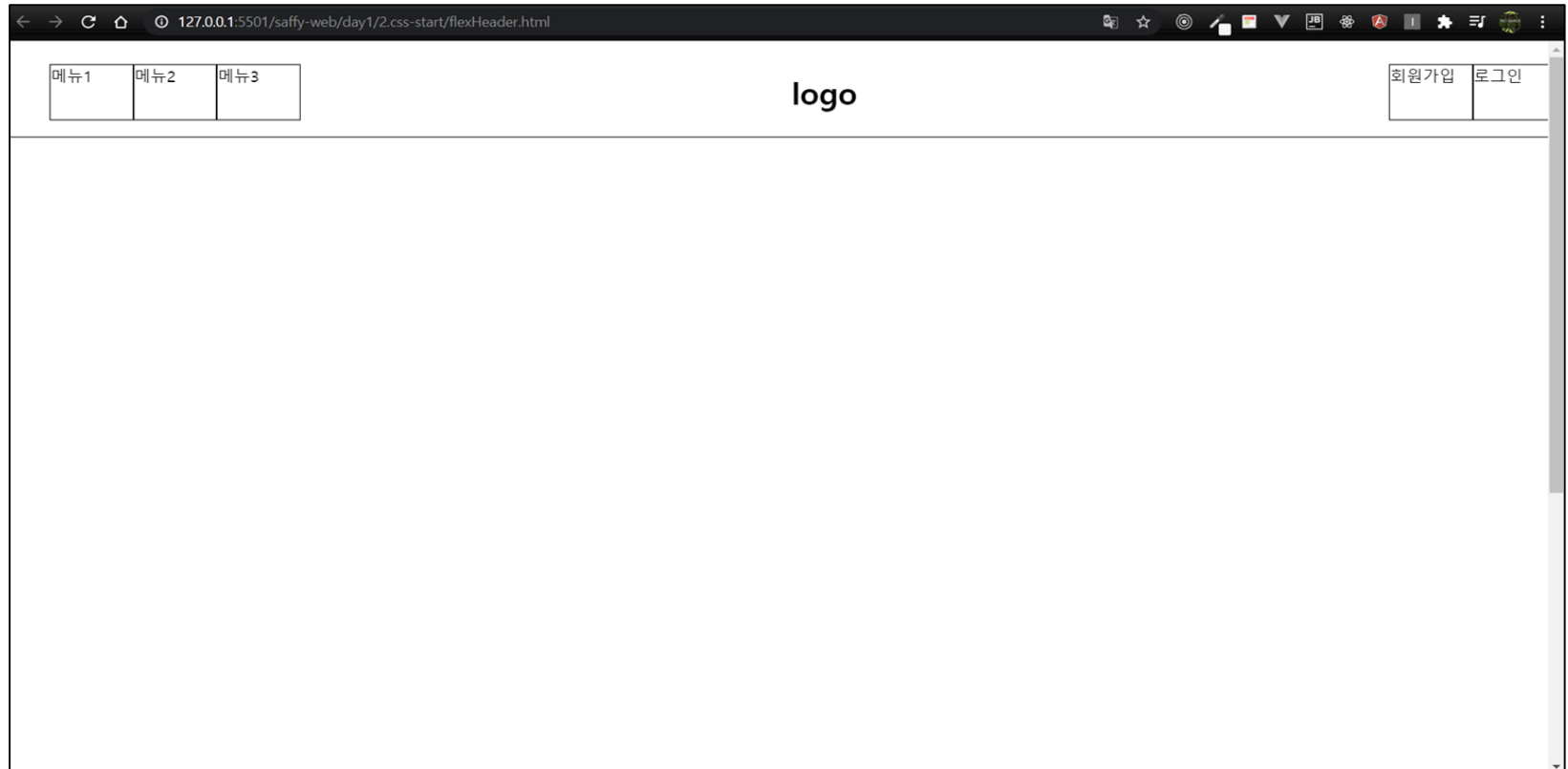
fixed

- 요소를 화면에 고정한다.
- 스크롤을 내려도 항상 같은 위치에 유지된다.
- 부모 요소의 위치와 무관하게 동작한다.
- top, right, bottom, left 값을 이용해 위치를 지정한다.



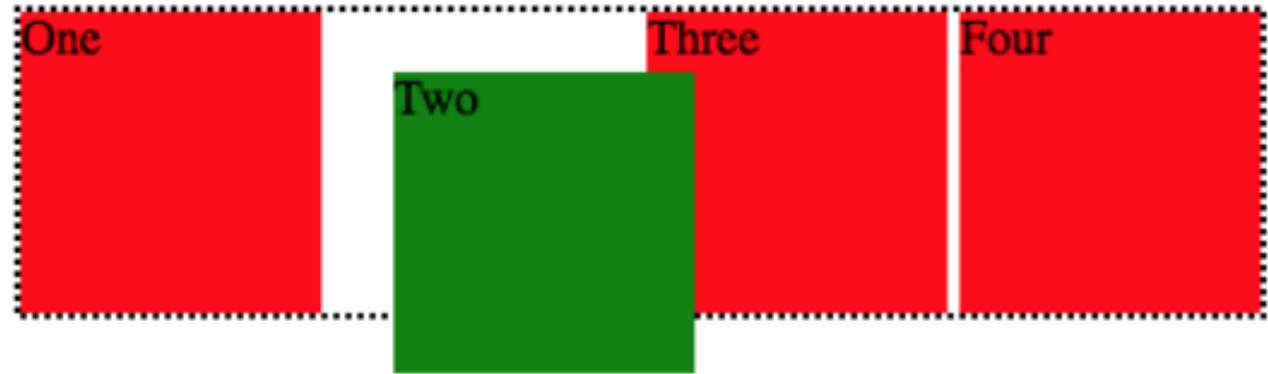
제한시간 : 3분

- Position을 활용해 메뉴 화면 상단 고정되게 만들기
 - (body {height : 1200px;}) 옵션을 주어 스크롤이 가능한 상태로 만들어 준다.



relative

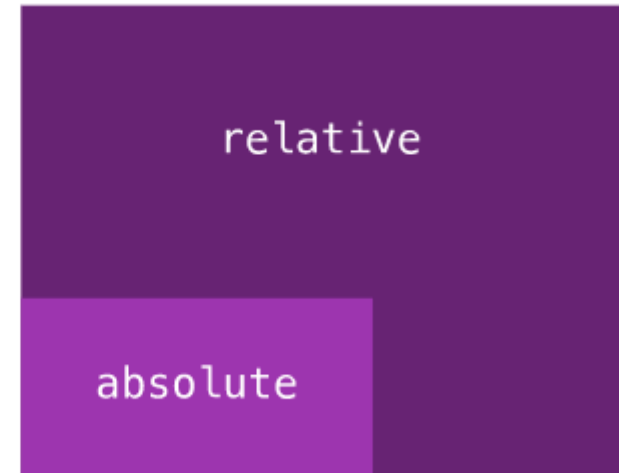
- 요소 자신이 원래 있어야 할 자리(현재 좌표)를 기준으로 이동한다.
- 시각적으로만 위치가 이동한다.
- top, left, right, bottom 값으로 이동을 지정할 수 있다.
- 기준점 역할을 하기 때문에, 자식 요소가 absolute일 때 기준이 되기도 한다.



position: relative

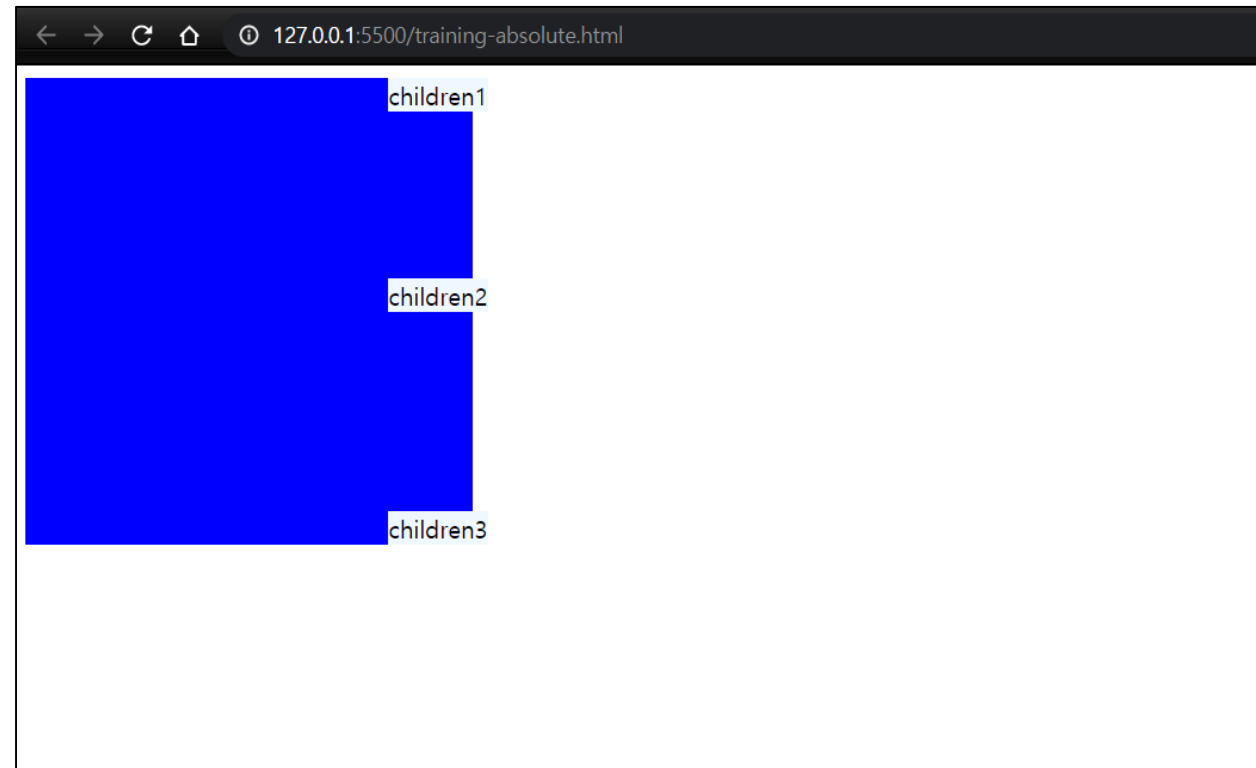
absolute

- position이 static이 아닌(relative, absolute, fixed) 부모 태그를 기반으로 움직인다.
- 다른 요소들과 겹칠 수 있다.
- 기준 요소가 없으면, body를 기준으로 위치가 정해진다.
- top, left, right, bottom 값을 사용해 정확한 위치를 지정할 수 있다.

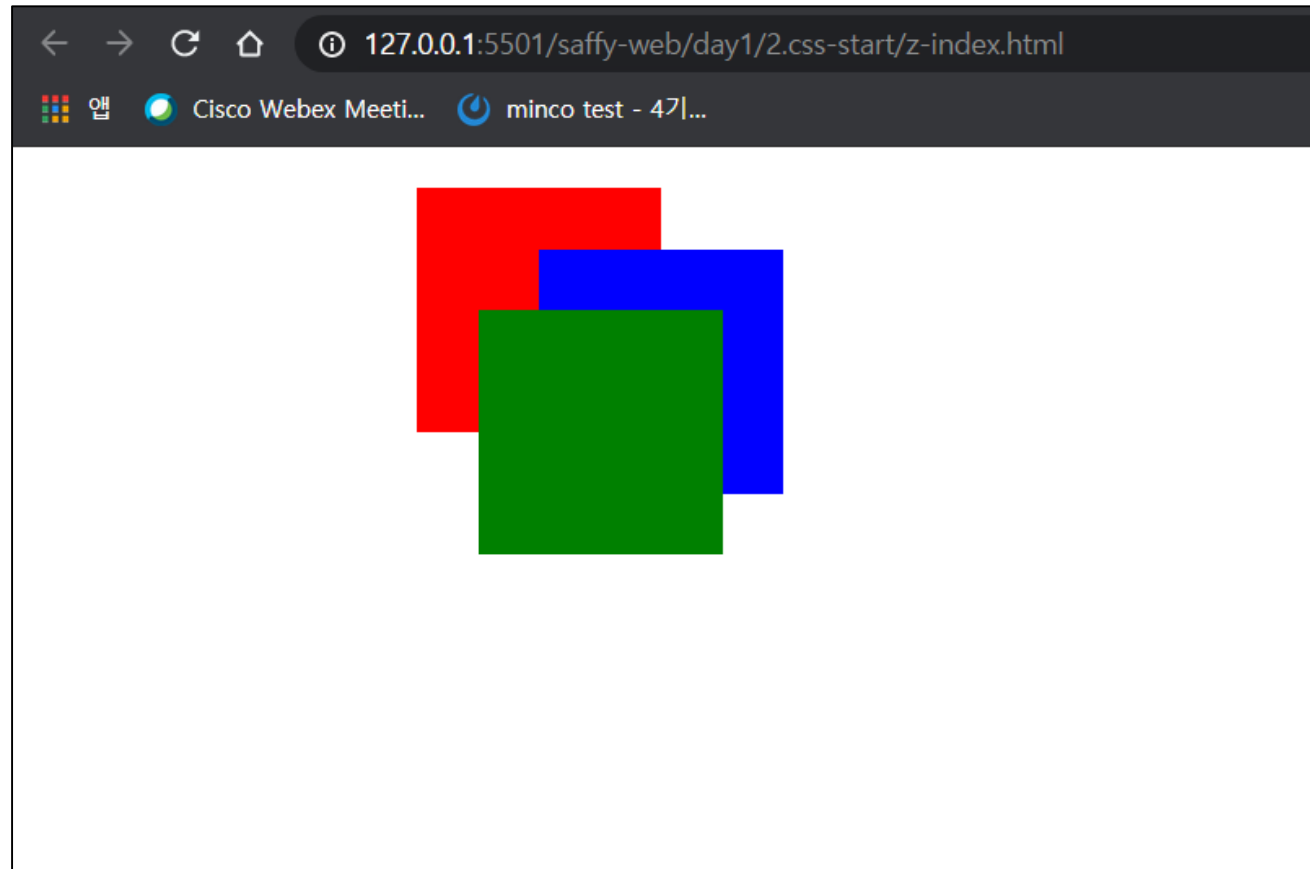


제한시간 : 10분

- Position을 relative와 absolute를 활용해
children 1 우측 상단 고정, children 2 우측 중앙 고정, children 3 우측 하단 고정 시키기
- Box는 width 300px height 300px로 지정한다
- 우측 고정으로부터 10px만큼 더 가야한다



각자 박스의 순서를 바꿔보기



네이버 상단바의 알림창 부분들을 만들어본다

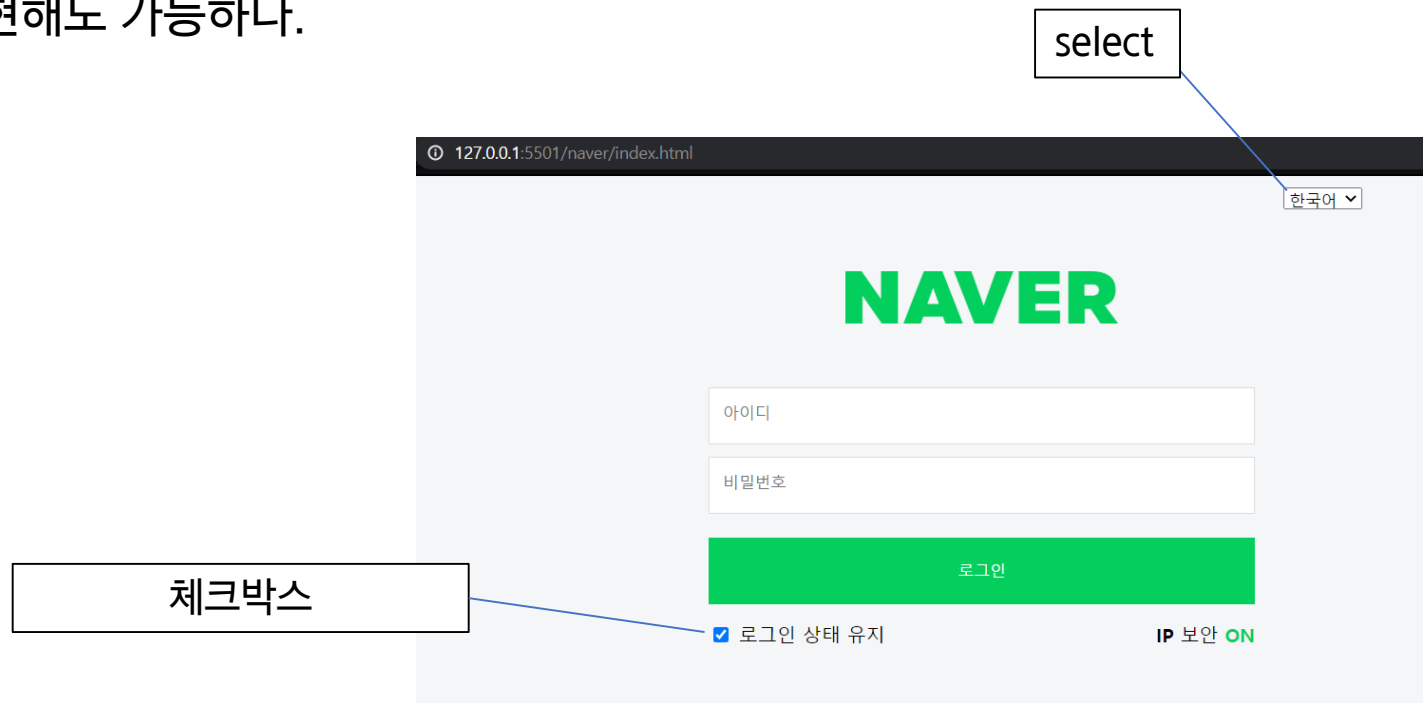
- Display Flex 활용
- Position relative, absolute 를 활용해서 상단바를 만들어본다

<https://www.flaticon.com/>



• 네이버 로그인창 만들어보기

- 금일 배운것을 모두 활용해서 네이버 로그인창을 구성하기
 - 내용이나 이미지, 아이콘은 다르게 구현해도 가능하다.
 - 네이버의 초록색: #04CF5C
 - 로고의 길이 : 자유롭게 지정



Day2-2. CSS 도전 미션

챕터의 포인트

- 미션 목표
- 실습 1. 카카오톡 프로필 사진
- 실습 2. 윈도우 이미지
- 실습 3. 동영상 이미지 만들기

미션 목표

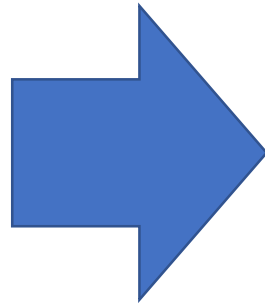
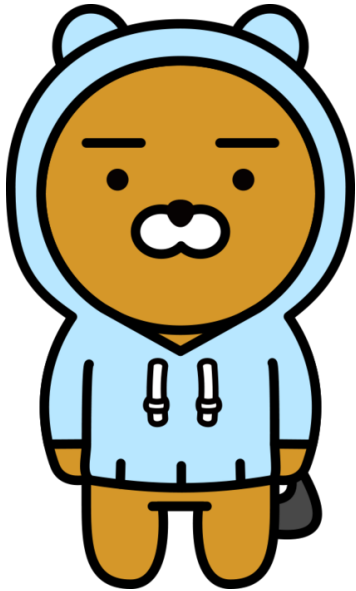
원하는 대로 CSS로 표현하기

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

실습 1. 카카오톡 프로필 사진

카카오톡 프로필 사진 형태 만들기

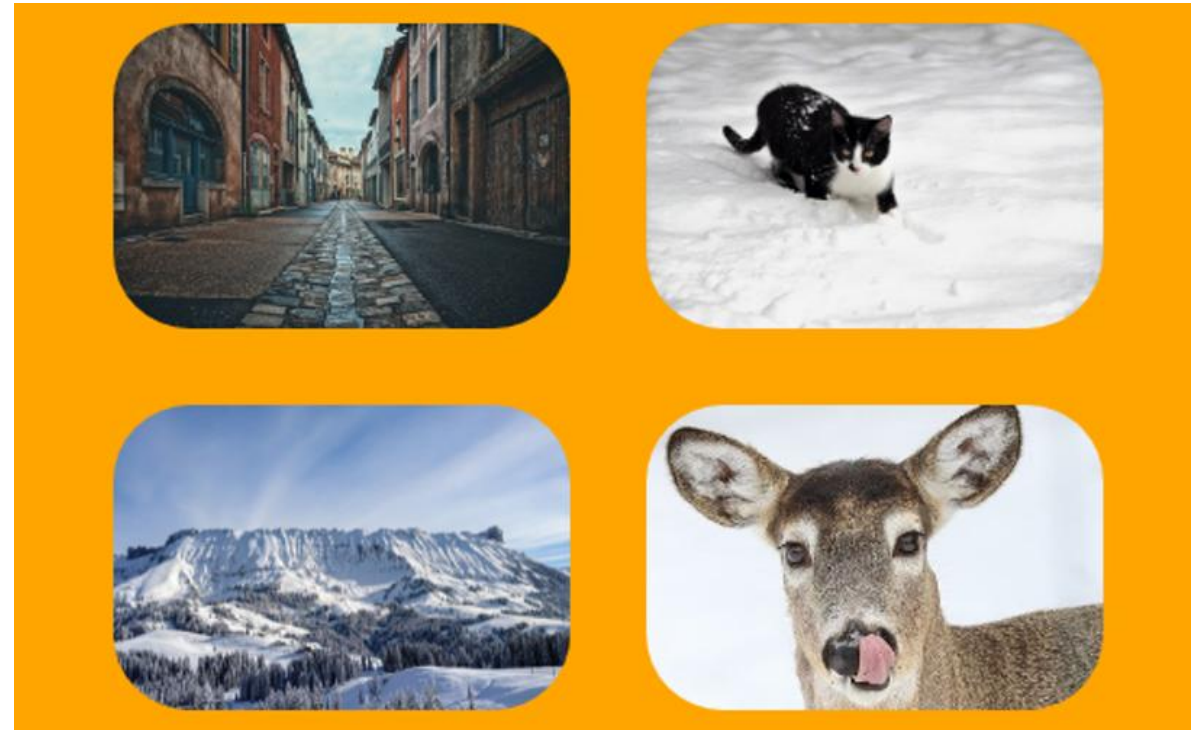
- 이미지 하나를 선정하여, 원형 태두리로 만든다.



실습 2. 윈도우 이미지

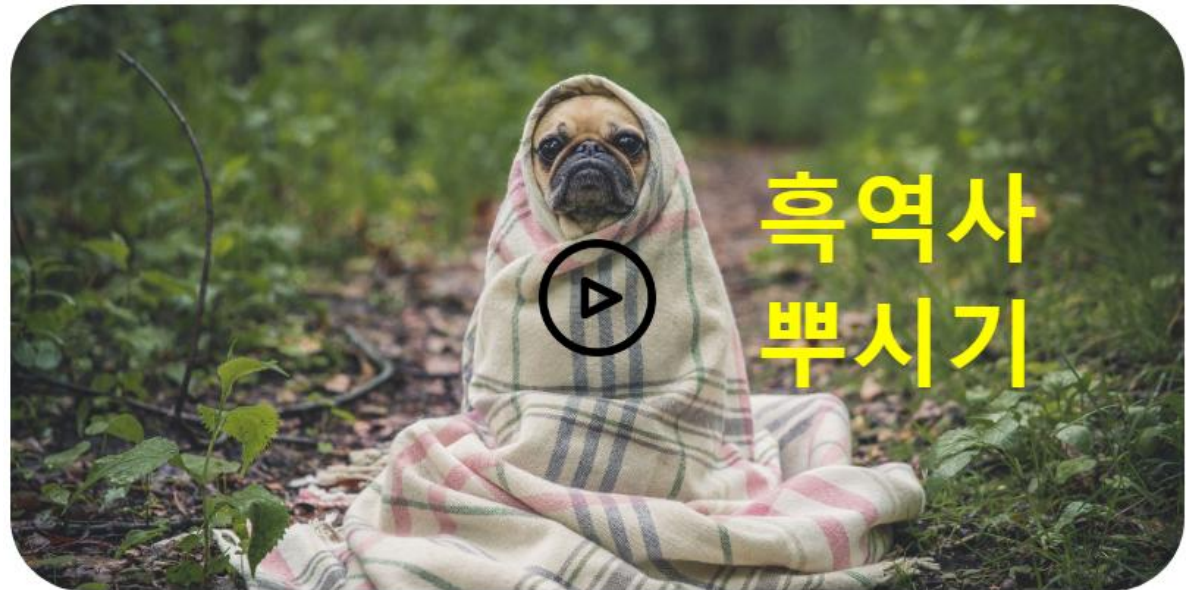
• 네 개의 사진을 둥근 사각 형태로 표현하기

- 이미지 4개 파일을 준비한다.
- Window 형태의 이미지를 구성한다.
- 간격은 일정한 간격을 유지한다.



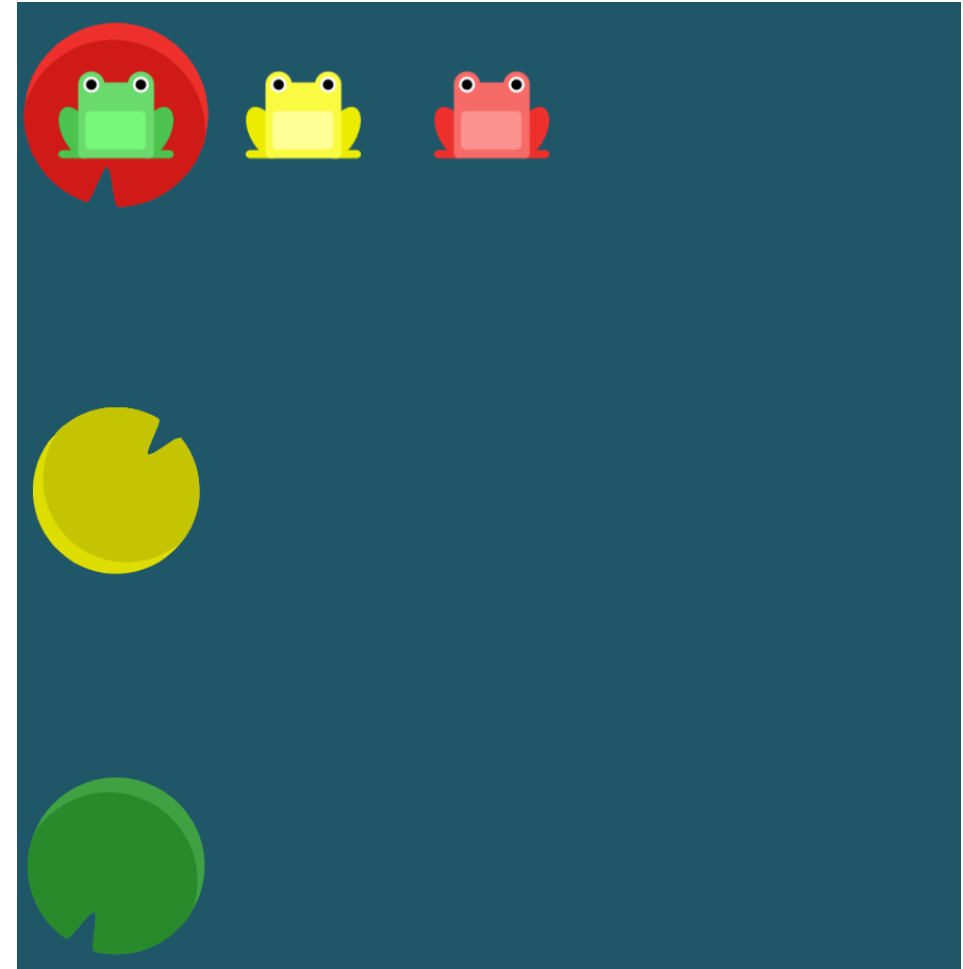
실습 3. 동영상 이미지 만들기

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



<https://flexboxfroggy.com/#ko>

- flex를 활용한 개구리 배치 게임 사이트



Day2-3. JavaScript 기초

챕터의 포인트

- JavaScript 개요
- JavaScript 기본 문법

JavaScript 개요

HTML 문서는 글, 표, 이미지 같은
정적인 내용만 넣을 수 있는 것이 아니다.

HTML 문서에 JavaScript 코드(소스코드)를 삽입할 수 있다.

- if / for / 변수 / 배열 / 함수 등 프로그래밍 기능을 구현할 수 있다.

소스코드 Run(실행)은 누가?

- 웹브라우저가 이 HTML을 여는 순간, 웹브라우저가 직접 수행한다.

다양한 기능을 HTML 문서에서 수행이 가능하다.

- Ex. 특정 버튼을 클릭했을 때 반응하도록 만들기
- Ex. 용자 입력에 따라 화면 내용을 바꾸기
- Ex. 로그인 기능 (ID/PASSWORD 확인 후 처리)

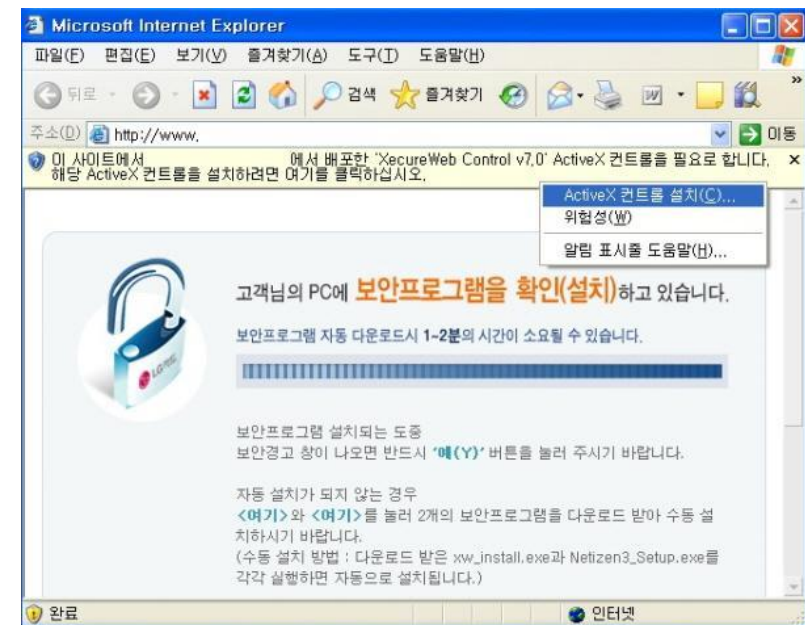
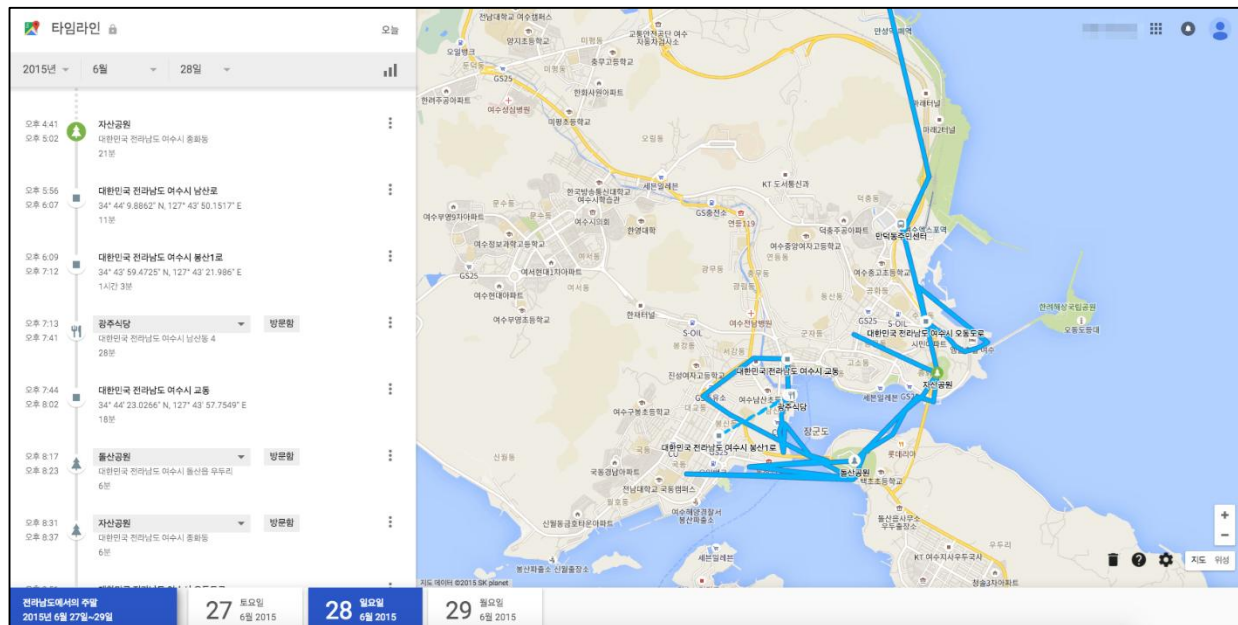
HTML 문서로 게임 / App 도 제작 가능하다.

JavaScript는 예전부터 있어왔다.

- 1990년대부터 대부분의 웹 브라우저에 JavaScript 인터프리터가 내장되어 있었다.
- 초기 사용처(기본 기능 중심)
 - 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



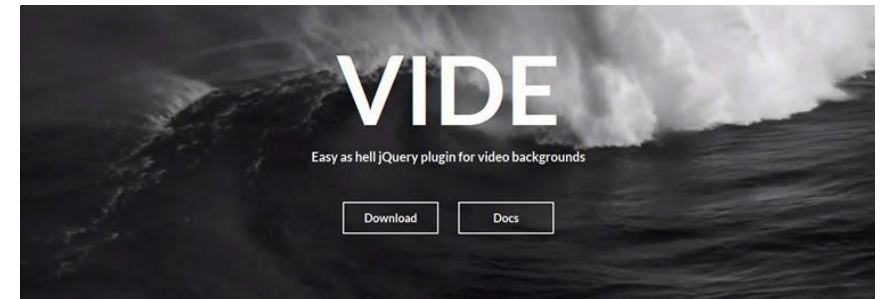
웹 페이지 새로고침 없이, 일부만 변경 가능

- 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 진리 값을 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>
```

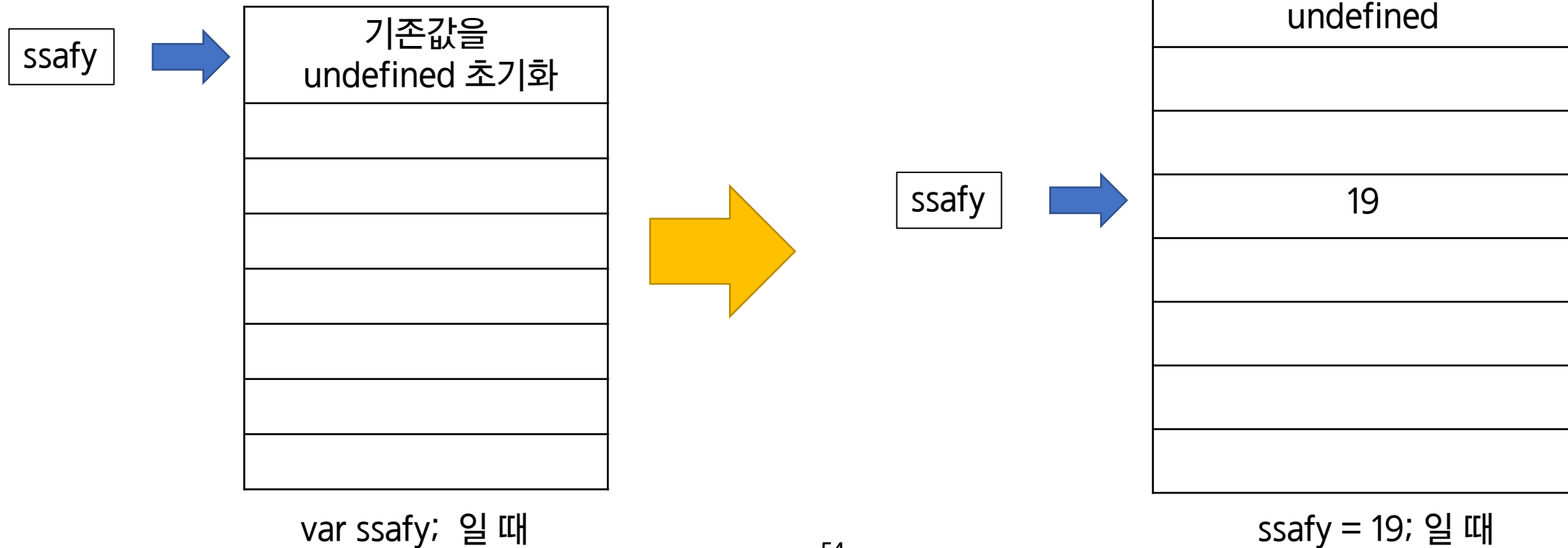
```
const result = { a: 5, b: 10 }
```

- 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;
- let과 const에서는 호이스팅이 동작하는걸 막는다.

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

undefined

undefined

var 사용 시 undefined

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

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

let 사용 시 error

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

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

const 사용 시 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
```

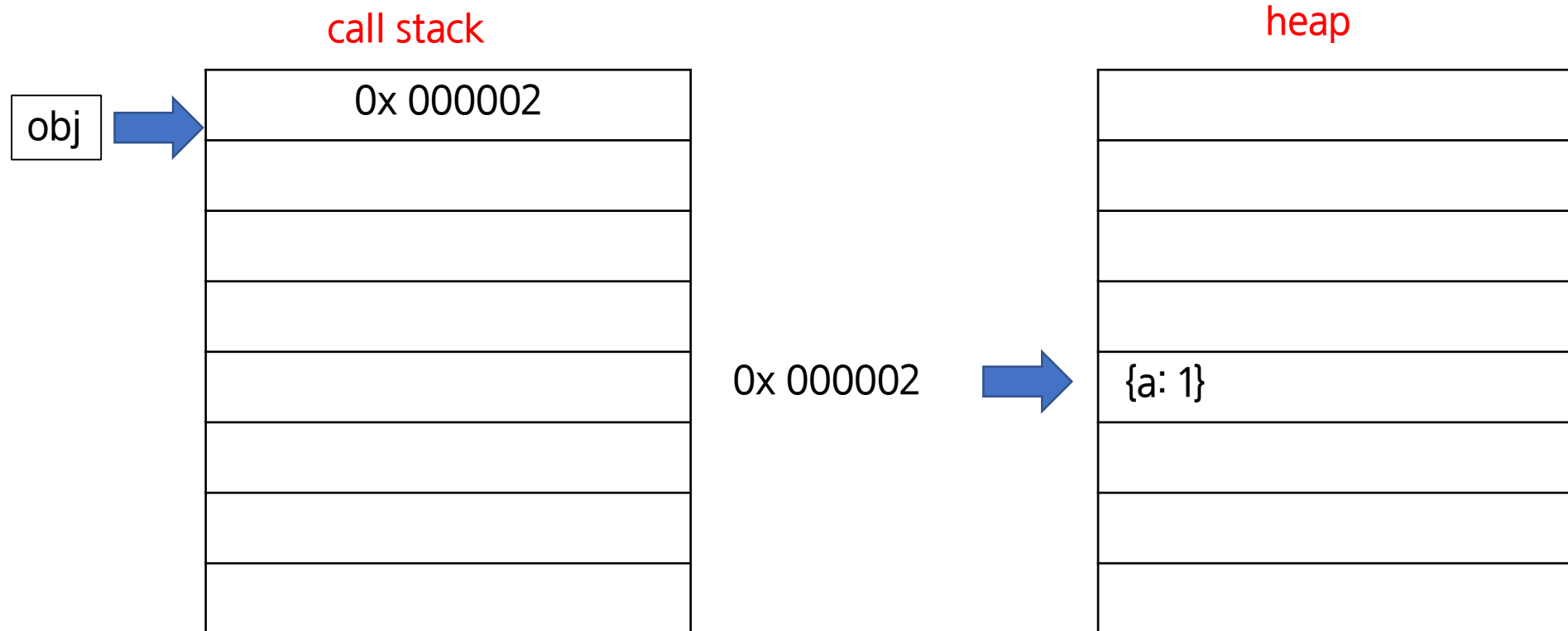
함수 선언식: 함수의 호이스팅

```
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 stack메모리의 값을 변경시키려 했기때문에 에러가 발생.

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

```
undefined
```

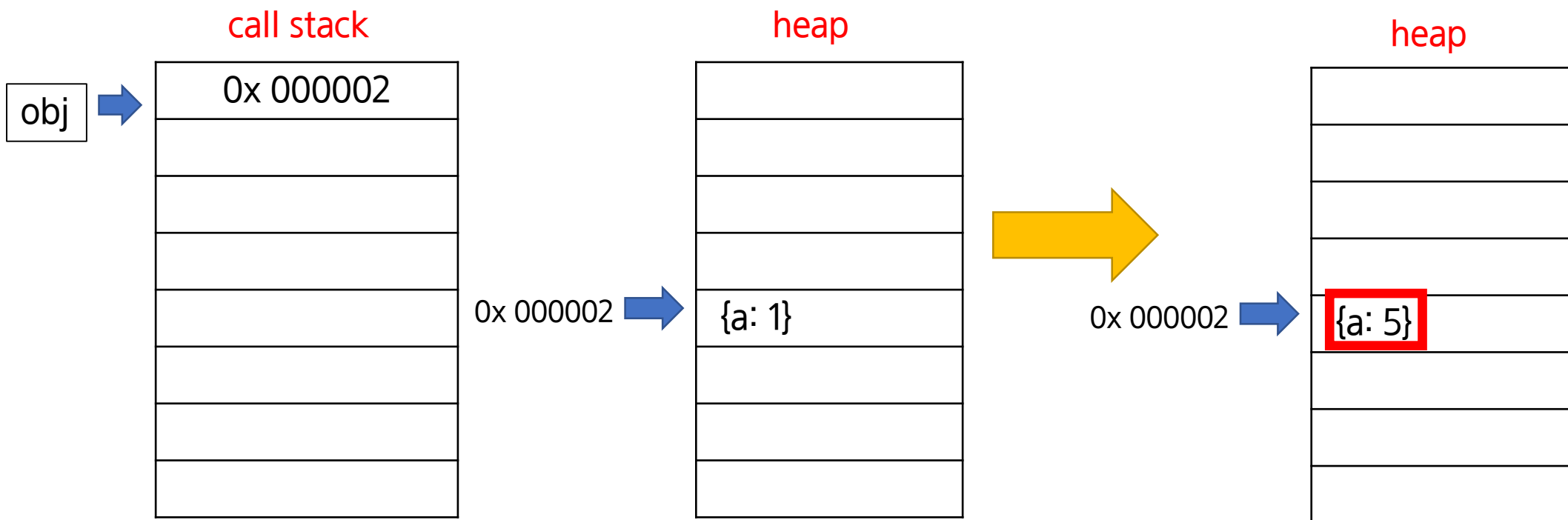
```
obj = { b: 1}
```

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

const 로 선언된 참조 자료형에서 값 바꾸기

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

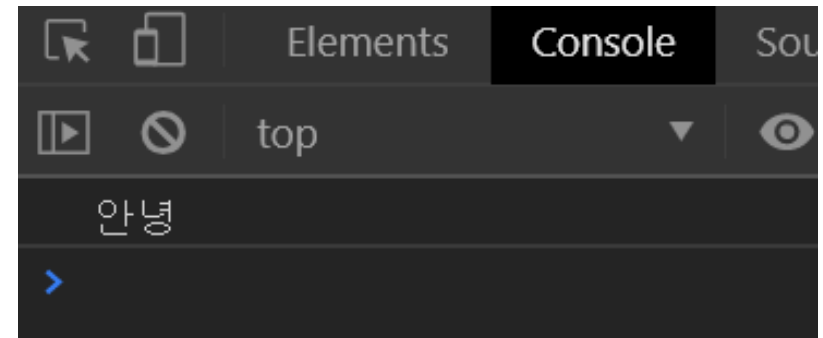
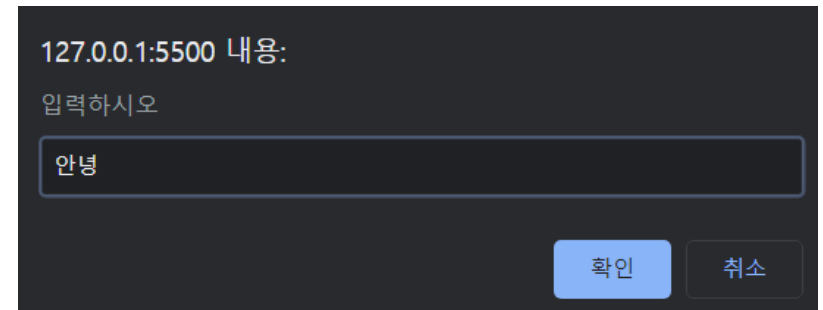
```
> const obj = { a : 1};  
< undefined  
> obj.a = 5;  
< 5  
> console.log(obj);  
  ▶ {a: 5}
```



prompt로 입력받기

- 문자열 입력 받고, 콘솔 출력하기

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



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)

JavaScript 에서 ES

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

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
2. 문자열 입력받고,
“BBQ” 이라면 1 ~ 30 까지 합 출력
“KFC” 이라면 1 ~ 30 까지 홀수 출력
그 외라면 “MOMS” 출력
3. 세 숫자 입력 받고, if문으로 가장 큰 값을 출력하기

- `let arr = new arr();`
- `let arr = new arr("MINCO", 1, 2, 3, "MINCO");`
- `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
2. ['A', 'E', 'W', 'Q', 'A']을 하드코딩하고, 'A'가 존재하는지
O 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 출력
하는 함수

second =

5 ~ -5 까지
출력하는 함수

내일 방송에서 만나요!

삼성청년SW·AI아카데미

과제1(8p)

Flex 활용 2

과제2(12p)

상단 고정 header 만들기

과제3(17p)

알림 표시 바 만들기

과제4(18p)

네이버 로그인 창 만들어 보기

과제5(24p)

카카오톡 프로필 사진

과제6(26p)

윈도우 이미지

과제7(28p)

동영상 이미지 만들기