

<!- 개발 관련 웹사이트 및 커뮤니티 ->

1) 디자인

- awwwards : <https://www.awwwards.com/>
- YouTube Channels : <https://cssauthor.com/best-youtube-channels-for-uiux-designers/>

2) 커뮤니티

- 생활코딩 : <https://www.facebook.com/groups/codingeverybody>
- 하코사 : <http://cafe.naver.com/hacosa>

3) 개발

- freeCodeCamp : <https://www.freecodecamp.org/>
- W3Schools : <https://www.w3schools.com/>
- NHN 엔터 : <https://github.com/nhnent/fe.javascript/wiki/FE-Weekly>
- Code Combat : <https://codecombat.com/>
- treehouse : <https://teamtreehouse.com>
- YouTube Channels : <https://cssauthor.com/best-youtube-channels-for-web-designers-and-developers/>

4) 기타

- statcounter : <http://gs.statcounter.com/>
- Can I Use : <http://caniuse.com/>
- Cloud9 : <https://c9.io/>
- 웹접근성 연구소 : <http://www.wah.or.kr/>
- 네이버 널리 : <http://nuli.navercorp.com/>
- HTML Validator : <https://validator.w3.org/>
- HTML 특수문자 : <http://egloos.zum.com/keep/v/1030642>
- 닷홈 : <http://www.dothome.co.kr/>

## /\* JavaScript 소개 \*/

- JavaScript 는 동적인 웹페이지를 만들기 위해서 사용되는 프로그래밍 언어로 최근에는 웹뿐만 아니라 하이브리드앱, 서버 개발, IoT 등 사용범위가 확대되고 있다.
- Syntax : 컴퓨터에서 사용되는 기본 문법을 의미하며, 외국어로 비유하자면 영어, 프랑스어, 중국어 등 국가별로 고유의 문장을 갖고 있듯이 프로그래밍 언어 또한 고유의 문장 구조를 갖고 있음. 즉, JavaScript 프로그래밍 언어를 배운다는 것은 그 언어의 문장 구조를 배우는 것과 같다고 할 수 있음
- <http://keepearthquakesweird.com/>
- <http://www.subaru.com/csr/environment.html>

## /\* JavaScript 적용 방법 \*/

### 1) html 문서 <script> 안에 JavaScript 기본 문법 삽입

```
<body>
  <script>
    document.write("Good Morning");
  </script>
</body>
```

### 2) 외부 파일 연결 방식

- html 문서 안에 <script src="파일명.js"></script> 삽입

```
/* main.js */
alert("Hello World");
```

- 단, <script>와 외부 파일 연결 방식을 동시에 사용할 수 없음

```
<script src ="main.js">
  alert("Hello World");    // 출력되지 않음
</script>
```

### 3) 구글 크롬 console 사용

- 크롬 요소검사로 들어간 후 Console 탭 안에서 JavaScript 문법 사용 가능

```
console.log("Start Point");

alert("Hello World");
document.write("Nice to meet you");

console.log("End Point");
```

## **`/* Variables ( 변수 ) 소개 */`**

- 변수는 문자나 숫자 등 어떤 값들을 담아내는 공간으로 변수명은 영어와 숫자 그리고 특수 문자를 혼합하여 지정할 수 있다. 단, 한글은 사용할 수 없다.
- 변수 선언과 초기화 : `var 변수명 = 값;`

```
// 자바스크립트 구문의 마지막은 ; 세미콜론으로 마침표를 표기해야 함
var msg;

// 방금 선언한 변수에 값을 할당하는 작업, 할당하기 전에는 undefined 값이 할당되어 있음
msg = "Hello World";

// 변수 선언과 값을 할당하는 작업을 동시에 진행할 수도 있음
var name = "inkweon Kim";

// 변수에 들어있는 값을 확인하고 싶다면 console.log 사용
console.log(msg);
console.log(name);
```

### **- 변수 이름 작성 요령**

- 캐멜 케이스 : `koreaSeoulHome`
- 스네이크 케이스 : `kore_seoul_home`

### **- 변수명 작성시 주의사항**

- 변수 이름은 문자나 밑줄(\_)이나 달러 기호(\$) 등으로 시작할 수 있다.
- 단, 변수 이름 작성시 숫자가 맨 앞에 사용될 수 없다 ex) `var 3point ( x )`
- 변수 이름은 최대한 자세하게 작성해야 한다. ex) `score`, `randomNumber`
- `tmax`, `n1` 등 의미가 불명확한 단어들의 조합은 피해야 한다.

**`/* 원시타입과 참조타입 */`**

**- 원시타입 : 문자열 ( String )**

```
var msg = "Hello World!";  
console.log(msg);
```

```
msg = 'Nice to meet you';  
console.log(msg);
```

```
// String 안에 작은 따옴표를 삽입하는 방법(1) : 큰 따옴표 안에 작은 따옴표를 사용  
var single_quotation_marks_1 = "She's a great person!";  
console.log(single_quotation_marks_1);
```

```
// String 안에 작은 따옴표를 삽입하는 방법(2) : \ 사용  
var single_quotation_marks_2 = 'She\'s a great person!';  
console.log(single_quotation_marks_2);
```

## - 원시타입 : 숫자 ( Number )

- Integers ( 정수 ) : 5, 0, -100, 999
- Float ( 실수 ) : 3.14, -9.8888, 0.0000
- 단, 문자열 안에 숫자를 넣는 방식은 지양해야 함 ex) var aString = '10' ( x )

```
var int = 10;
var flo = 3.14;
console.log(int + flo);
```

## /\* 연산자 (1) \*/

- 연산자는 값을 계산하거나 증가 또는 감소 그리고 문자열을 이어 붙이는 등 다양한 연산을 처리

```
// 숫자 사칙연산 그리고 %
var a = 20;
var b = 10;

console.log(a + b);
console.log(a - b);
console.log(a * b);
console.log(a / b);
console.log(a % b);

// 문자열 연산
var firstName = "Inkwon ";
var lastName = "Kim";

console.log(firstName + lastName);

// 숫자와 문자열 연산
var str1 = "현재 시간은 ";
var presentTime = 12;
var str2 = "시 입니다";

console.log(str1 + presentTime + str2);

// 문자열의 숫자와 Number의 숫자 차이
var num1 = "10";
var num2 = "20";

console.log(num1 + num2);
```

### **/\* 연산자 (2) \*/**

```
// ++, --
var num1 = 10;

console.log(--num1);
console.log(++num1);

// +=, -=, *=, /=, %=
var num2 = 20;
num2 += 40; // num2 = num2 + 40; 와 같은 의미

console.log(num2);
```

### **/\* 연산자 (3) \*/**

```
// 비교 연산자
var a = 10;
var b = 20;
var c = 30;

console.log( a === b );
console.log( a !== b );

console.log( a > b );
console.log( a < b );
console.log( a >= b );
console.log( a <= b );

// 논리 연산자
console.log( a === b && a === c )
console.log( a === b || a === c )
```

- 비교 연산자에서 ==, === 의 차이
- ( "3" == 3 ) true 값을 취하지만 ( "3" === 3 ) 경우에는 false 값을 취함

- 원시타입 : 참 또는 거짓 ( Boolean )

```
console.log(true);  
console.log(false);  
console.log(1 === 2);  
console.log(1 < 2);
```

- 원시타입 : null, undefined

```
var n = null;  
var u;  
  
console.log(n);  
console.log(u);
```

- null은 사용자가 임의로 빈 값을 삽입한 상태
- undefined 변수를 선언한 상태

- null과 undefined 차이

```
console.log(typeof null);  
console.log(typeof undefined);  
  
console.log(null == undefined);  
console.log(null === undefined);  
  
console.log(!null);  
console.log (!!null);  
  
console.log(!undefined);  
console.log (!!undefined);  
  
console.log(1 + null);  
console.log(1 + undefined);
```

- NaN ( Not-a-Number ) : 숫자가 아니라는 의미



- 참조타입 : 함수 ( Function )

- 함수란 한번 만들어 두면 재사용이 가능한 형태를 가진 코드를 의미한다.

```
// 함수 선언
function foo() {
    console.log('call foo');
}

// 함수 호출
foo();
```

- 함수에 정보 입력하기

- Parameter ( 매개변수 ) : 전달된 값을 받아들이는 변수
- Argument ( 인자 ) : 어떤 함수를 호출시 전달되는 값

```
function yourAge( age ) {    // Parameter : age
    console.log(age + "살입니다.");
}

yourAge(20);    // Argument : 20

function rectangle(width, height) {    // Parameter와 Argument는 1개 이상 삽입이 가능
    var area = width * height;
    console.log(area);
}

rectangle(200, 100);
```

- return : 함수 자체가 값을 넣고자 할 때 사용된다.

```
function yourAge( age ) {
    return age + "살입니다.";
}

var age = yourAge(20);
console.log(age);
```

- 참조타입 : 배열 ( Array )

```
var fruit = ["apple", "banana", "tomato"];

console.log(fruit[0]);
console.log(fruit[1]);
console.log(fruit[2]);
```

- 배열 안에 배열에게 접근하는 방법

```
var scores = [
    [100, 80, 90, 150],
    [75, 90, 65, 98],
    [43, 87, 59, 10]
];

// scores 배열 안에 있는 100을 선택하기 위해서는 scores[0][0] 으로 표시
console.log(scores[0][0]);
```

## - 참조타입 : 객체 ( Object )

- 객체란 변수와 함수를 모여 만든 하나의 집합체이다.
- name 은 key 또는 property , inkweon 은 value 로 불림

```
var student = {  
  name : "inkweon",  
  age : 31,  
  skills : ['JavaScript', 'HTML', 'CSS']  
};
```

## - 객체 프로퍼티에 접근하는 방법

```
var student = {  
  name : "inkweon",  
  age : 31,  
  skills : ["JavaScript", "HTML", "CSS"]  
};  
  
console.log(student.name);           // console.log(student['name']);  
console.log(student.age);            // console.log(student['age']);  
console.log(student.skills);         // console.log(student['skills']);
```

## - 객체에 프로퍼티 추가하는 방법

```
var student = {  
  name : "inkweon",  
  age : 31,  
  skills : ["JavaScript", "HTML", "CSS"]  
};  
  
student.gender = "male";
```

## - 객체의 Method ( 메서드 ) : 객체 안에 포함되어 있는 함수를 메서드라 얘기함

```
var student = {  
  .....  
  sum : function(num1, num2) {  
    return num1 + num2;  
  }  
};  
  
var total = student.sum(10, 20);
```

- 중요 내용
- 원시 타입과 참조 타입 차이점 : <https://brunch.co.kr/@brunch92ny/10>

```
// 원시타입
var str1 = "Hello World";
var str2 = str1;

str1 = "Nice to meet you";

console.log(str1);
console.log(str2);


// 참조 타입
var obj1 = { name: "Inkweon" };
var obj2 = obj1;

obj1.name = "Park";

console.log(obj1);
console.log(obj2);
```

- 배경 색상 변경 (1)

```
// 즉시실행함수
(function () {

    var colors = ['yellow', 'green', 'pink', '#DC143C', 'rgba(123, 123, 123, 0.5)'];
    var screen = document.getElementById('screen');
    var mouse = document.getElementById('mouse');

    mouse.addEventListener('click', function() {

        var random = Math.floor(Math.random() * colors.length);
        screen.style.background = colors[random];

    });

})();
```

- Math.random() : 0 ~ 1 사이의 랜덤한 실수를 출력하는 함수, 0은 포함 / 1을 불포함
- Math.floor() : 소수점 이하는 버리는 함수
- 배열.length : 배열 안에 있는 데이터의 개수를 출력하는 프로퍼티