

· HTML

- 정적
- 한번 화면에 출력되면 언제든 그 모습 그대로

· JavaScript

- 동적
- 사용자와의 상호작용

→ HTML의 정보 + JavaScript의 기능

<script> 태그 안에는 JS가 의로 약속 → web browser가 <script> 안쪽 코드를 JS로 해석함

ex) <script>

```
document.write('hi');  
</script>
```

document.write(); 웹페이지에 글씨 출력, html과 달리 동적임 (html: 1+1 → 1+1
JS: 1+1 → 2)

이벤트(Event): 웹 브라우저 위에서 사용자와 상호작용하다 일어나는 일 ex) onclick, onchange, onkeydown ...

· 이벤트 속성 값으로는 반드시 JS가 와야함

· 이벤트의 속성 값은 web browser가 기억하고 있다가 사용자가 이벤트가 위치하고 있는 태그를 접근하면 기억하고 있던 JS코드를 JS문법에 따라 해석해서 웹 브라우저가 동작한다

콘솔(Console): 파일을 만들지 않고도 JS코드 즉시 실행 가능

· Console은 통해 JS 실행 → 지금 보고 있는 웹페이지를 대상으로 해서 JS가 실행된다

datatype(자료형): String(문자형), number(숫자), Boolean, Null, Undefined, Symbol

number(숫자) - 연산 산술연산자: +, -, *, / ...

string(문자형) - " " ' '

variable(변수): 바뀔수 있는 어떤 값

변수선언: var 변수명 = ~

constant(상수): 바뀌지 않는 값 ex) 1, 'hi' ...

대입연산자(=): 우항의 값을 왼쪽의 변수에 대입함
(좌항)

웹 브라우저 제어

① JS를 이용하여 제어하고자 하는 태그 선택

JavaScript select tag by CSS selector → `document.querySelector(____)`
selector

② 선택한 태그에 style 속성 적용

JavaScript element style → `.style`

HTML - 컴퓨터 언어

JS - 컴퓨터 프로그래밍 언어

program : 순서가 존재

• 비교연산자

일치연산자(==)

불일치연산자(!=)

초과연산자(>)
(이상) (>=)

미만연산자(<=)
(이하) (<=)

• boolean

true

false

• 조건문

if { }

else { }

배열 (Array) : []

반복문 (Loop) : while, for

while (조건문) {

}

for ([조건문1]; [조건문2]; [조건문3]) {

}

함수 (function)

함수 { 입력: parameter (매개 변수), argument (인자)
출력: return

```
function 함수명 (매개변수) {  
    ;  
}  
함수명 (인자)
```

} 선언
) 호출

객체 (Object)

서로 연관된 함수, 변수는 같은 이름으로 그룹화하여 정리정돈하기 위한 도구

Python 의 dictionary 처럼 { key: value } 형태이다

value에 배열, 숫자, 심지어 함수도 담을 수 있음

객체에 소속된 함수 = method

객체에 소속된 변수 = property

더러 웹페이지에서 반복되는 JS코드는 파일로 묶을 수 있다.

ex) colors.js → 각 웹페이지에 <script src="colors.js"></script> → 효율성 ↑ (1:캐시)

library VS framework

library: 프로그램에 사용한 부품 자체 ex) jquery jquery with CDN by <script src=""></script>
framework: 반제품 (깎은 톳)

UI VS API

UI (User Interface): 사용자가 시스템을 제어하기 위해서 사용하는 조작 장치
API (Application Programming Interface): 애플리케이션을 만들기 위해서 프로그래밍 할 때 사용하는 조작 장치