

Javascript 문

예시)

```
let x, y, z;      // 자바스크립트 문-1
x = 5;            // 자바스크립트 문-2
y = 6;            // 자바스크립트 문-3
z = x + y;        // 자바스크립트 문-4
```

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Statements</h2>

<p><b>자바스크립트 프로그램</b>은 컴퓨터에서 실행되는 <b>문</b>의 목록이다.</p>

<p id="demo"></p>

<script>
let x, y, z; // 자바스크립트 문 1
x = 5;       // 자바스크립트 문 2
y = 6;       // 자바스크립트 문 3
z = x + y;   // 자바스크립트 문 4

document.getElementById("demo").innerHTML =
"z의 값은 " + z + "입니다.";
</script>

</body>
</html>
```

◆ 프로그램

- 컴퓨터 프로그램은 컴퓨터에 의해 "실행"되는 "명령"의 목록이다.
- 프로그래밍 언어에서는 이러한 프로그래밍 명령을 명령문이라고 한다.
- JavaScript 프로그램은 프로그래밍 명령문의 목록이다.
- HTML에서 JavaScript 프로그램은 웹 브라우저에서 실행된다.

◆ 자바스크립트 문 (Statements)

- JavaScript 문은 다음으로 구성된다.
- 값, 연산자, 표현식, 키워드 및 주석.
- 아래 예시 문은, 브라우저에 id="demo"인 HTML 요소 내부에 "안녕! 자바스크립트는 처음이지?"를 쓰도록 지시한다.

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Statements</h2>

<p>HTML에서 JavaScript 문은 브라우저에서 실행됩니다.</p>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML = "안녕! 자바스크립트는 처음이지?";
</script>

</body>
</html>
```

- 대부분의 JavaScript 프로그램에는 많은 JavaScript 문이 포함되어 있다.
- 명령문은 작성된 순서대로 하나씩 실행된다.
- JavaScript 프로그램 = JavaScript 문 = JavaScript 코드

◆ 세미콜론 ;

- 세미콜론은 JavaScript 문을 구분하는 역할을 한다.
- 각 실행 가능한 문의 끝에는 세미콜론을 추가하자.

예시)

```
let a, b, c; // a, b, c 를0 변수로 선언
a = 5;      // 값 5를 a에 할당
b = 6;      // 값 6을 b에 할당
c = a + b;   // a와 b의 값을 합하여 c에 할당
```

- 세미콜론으로 구분하면 한 줄에 여러 문(statements)이 허용된다.

예시)

```
a = 5; b = 6; c = a + b;
```

- 웹에서 세미콜론이 없는 예제를 볼 수 있기는 하다.
- 세미콜론으로 끝나는 문장은 필수는 아니다. 하지만 적극 사용하는 것이 유리하다.

◆ 자바스크립트 공백 (White Space)

- JavaScript는 두 칸 이상의 공백은 한 칸 공백으로 처리한다.
- 스크립트에 적절하게 공백을 사용하여 가독성을 높일 수 있다.
- 아래의 두 예시는 각각 다른 공백을 사용했지만 결과는 동일하다.

예시)

```
let person = "홍길동";  
let person="홍길동";
```

- 공백을 사용하기 위한 적절한 위치는 연산자 (= + - * /) 앞과 뒤에 공백을 사용하는 것이다.

예시)

```
let x = y + z;
```

◆ JavaScript 줄길이 및 줄바꿈

- 최고의 가독성을 위해 프로그래머는 긴 코드를 한 줄에 길게 쓰는 것을 좋아하지 않는다.
- JavaScript 문을 줄 바꿈 하고 싶을 때, 가장 좋은 시점은 연산자 다음이다.

예시)

```
document.getElementById("demo").innerHTML =  
"홍길동";
```

◆ JavaScript 코드블록 (Code Blocks)

- JavaScript 문은 **중괄호 {...}** 안에 있는 코드 블록으로 그룹화 할 수 있다.
- 코드 블록의 목적은 **함께 실행할 명령문**을 정의하는 것이다.
- 블록으로 함께 그룹화 된 명령문을 찾을 수 있는 곳은 **JavaScript 함수**이다.

예시)

```
function myFunction() {  
    document.getElementById("demo1").innerHTML = "안녕!";  
    document.getElementById("demo2").innerHTML = "자바스크립트는 처음이지?";  
}
```

※ 들여쓰기를 적절히 사용하면 가독성을 향상시켜서 디버깅/수정에 도움을 준다.

◆ JavaScript 키워드 (Keywords)

- JavaScript 문은 수행할 JavaScript 작업을 식별하는 키워드로 시작하곤 한다.
- **변수 이름으로 사용 불가**: 예약어, 객체, 속성, 메서드, 자바 예약어, 기타 예약어, HTML 이벤트 핸들러
- 자바스크립트 키워드는 예약어이다. 다음은 키워드 (예약어) 목록의 극히 일부이다.

키워드	설명
var	변수 선언
let	블록 변수 선언
const	블록 상수 선언
if	조건에서 실행할 명령문 블록을 표시합니다.
switch	다른 경우에 실행할 명령문 블록을 표시합니다.
for	루프에서 실행할 명령문 블록을 표시합니다.
function	함수 선언
return	기능 종료
try	명령문 블록에 대한 오류 처리를 구현합니다.