

1_기초문법

출력

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <script>
      console.log("hello world");
      document.write("HELLO WORLD");
    </script>
  </body>
</html>
```

주석처리

```
<!DOCTYPE html>
<html>
  <head>
    <title>Document</title>
  </head>
  <body>
    <script>
      //x라는 변수에 3+5의 결과를 대입 한다.

      var x = 3 + 5;
    </script>
  </body>
</html>
```

변수 선언자(ES6)

개념/문법

- 변수
 - `let`
 - 재 할당이 가능한 변수
 - `const`
 - 재 할당이 불가능한 변

참고 Naming Convention

- 변수의 이름을 짓는 방법
- 카멜 케이스(camelCase)
 - 첫글자는 소문자, 이후 단어의 첫글자는 대문자
- 스네이크 케이스(snake_case)
 - 소문자로 작성하되 띄어쓰기는 '_' 로 대체

```
//Naming Convention
const add_one_to_number ( bad )
addOneToNumber ( good )
```

- 식별자에 주의하여 변수 할당

```
let 2number = 2;    // [X]
let $number = 2;    // [0]
let _number = 2;    // [0]
let number2 = 2;    // [0]
```

let

- 재할당 가능한 변수

```
let x1 = 5; // x1 5을 저장
let y1 = 6; // y1 6을 저장
let z1 = x1 + y1; // z1 11을 저장

// let x1 = 7; // 에러
x1 = 7;
```

const

- 선언된 변수명에 재선언 불가능
- 값 재할당은 가능

```
//const => 상수 변하지않는값

const PI = 3.14;
PI = 2.141; // 정의되지 않음
```

var

- ES6 이전에는 변수를 선언할 때는 `var` 키워드를 사용해 변수 할당
- 하지만 `var` 로 선언한 변수는 다양한문제를 일으키며, 현재 `var` 키워드 사용은 지양하고 있다. 대신 `let`, `const` 사용!
- **안좋은 이유**
 - 선언 전에 사용할 수 있다.
 - 선언을 여러번 할 수 있다.
 - 스코프에 영향을 받지 않기 때문에 암묵적 전역 변수를 양산할 가능성이 크다.

```

var x = 5; //변수 x에 5을 저장
var y = 6; //변수 y에 6을 저장
var z = x + y; // 변수 z에는 11을 저장

console.log("x = ", x);
console.log("y = ", y);
console.log("z = ", z);

var x = 7; // 재선언
z = x + y; // 이미 선언된 변수에 새로운 값을 할당
console.log("x = ", x);
console.log("z = ", z);

```

스코프 (Scope)

- 변수나 함수를 사용할 수 있는 영역

전역 스코프

- 자바스크립트 최상단에 선언한 변수의 영역은 **전역 스코프** 이다.
- **전역 스코프** 에 선언된 변수를 **전역 변수** 라고 한다.

```

/* 전역스코프 */
let name = '오민엽';    // 전역변수
let age = 39;           // 전역변수

function printPerson() {
    console.log(`name: ${name}, age: ${age}`);
}

```

블록 스코프

- 블록 **{ }** 으로 묶인 영역을 블록 스코프라 한다
- 블록안에 선언된 변수는 블록 안에서만 접근 가능

```

{
    let a = 'parent';
}

```

```

{
  let a = '재선언';
  let b = 'child';
  console.log('a: ', a);
  console.log('b: ', b);
}
console.log('a: ', a);
console.log('b: ', b);
}

```

PoimaWeb


웹 프로그래밍 튜토리얼

 <https://poimaweb.com/es6-block-scope>



변수와 상수

대다수의 자바스크립트 애플리케이션은 사용자나 서버로부터 입력받은 정보를 처리하는 방식으로 동작합니다. 아래와 같이 말이죠. 온라인 쇼핑몰 - 판매 중인 상품이나 장바구니 등의 정보 채팅 애플리케이션 - 사용자

 <https://ko.javascript.info/variables>

