**Web개발\_Day41\_노트정리**

**JavaScript 반복문**

ㆍwhile문

* While (조건) {

반복해서 실행할 코드;

}

ㆍfor문

* For (초기값; 반복조건; 증가값){

반복해서 실행할 코드;

}

ㆍbreak문 : 반복작업을 중단시킬 때 사용한다.

ㆍcontinue문 : 반복을 지속되기 하되, 특정 코드를 건너뛴다.

**JavaScript 함수**

ㆍ함수란? : 함수(function)란 하나의 로직을 재실행 할 수 있도록 하는 것으로 코드의 재사용성을 높여준다.

ㆍ함수 생성

* Function 함수명 (인자) {

코드

Return 반환값

}

* Return : 함수 내에서 사용한 return은 return 뒤에 따라오는 값의 함수의 결과로 반환한다.
* 인자란? : 인자는 함수로 유입되는 입력값을 의미하는데, 어떤 값을 인자로 전달하느냐에 따라서 함수가 반환하는 값이나 동작방법을 다르게 할 수 있다.

ㆍJS에서 함수를 호출하는 또 다른 방법들

* Var 변수명 = function () { …내용… }
* 함수사용 : 변수명();
* 위와 같은 방법으로 함수를 변수에 담아서 변수를 함수화하여 사용할 수 있다.
* ( function() { …내용…} )();
* 위와 같은 방법으로 함수를 ( ) 에 묶는다.
* 이를 ‘익명함수’라고 부른다
* 일회성 호출에만 사용이 가능하고 재사용은 불가능하다.

**변수의 유효성 범위, Scope**

ㆍ함수 밖에서 변수를 선언하면 그 변수는 전역변수가 된다.

ㆍ전역변수는 어플리케이션 전역에서 접근이 가능한 변수이다.

* 즉, 어떤 함수 안에서도 그 변수에 접근 할 수 있다.

ㆍvar 변수 선언의 문제점

* 변수를 같은 이름으로 한번 더 선언했음에도 불구하고, 에러가 나오지 않고 각기 다른 값이 출력된다.
* 이를 보완하기 위해 추가된 변수 선언 방식이 let, const 이다.

ㆍlet, const 의 차이점

* Let : 선언, 재할당이 가능핟
* Const : 선언, 재할당이 불가능. 최초의 할당만이 가능
* 재할당이 필요가 없는 ‘상수’에 사용하는 것이 적합하다.

**Object Model**

ㆍ 웹 브라우저의 구성요소들은 하나하나가 객체화 되어있다. 자바스크립트로 이 객체를 제어해서 웹 브라우저를 제어할 수 있게 된다. 이 객체들은 서로 계층적인 관계로 구조화되어있다. BOM, DOM은 이 구조를 구성하고 있는 가장 큰 틀의 분류라고 할 수 있다.

ㆍObject Mode의 구성

* DOM : Document Object Model
* BOM : Browser Object Model
* JavaScript Core