

# UI 화면구현

JS vs Java

Java - 기본적인 프로그래밍이기 때문에 중요한 편(JSP, Spring으로 파생이 됨)

JS - 프론트엔드에서 중요한 프로그래밍(Jquery.js, Vue.js, React.js등이 있음)

JS자체만 적으로 구현하는 것을 Vanilla JS라고도 부름.

Node.js - 서버에서 사용하는 JS

AJAX와 같은 기술로 JSON, XML을 웹서버에서 이용함.

<변수와 연산자>

변수 : number, string, boolean, undefined, object, array, null(참조 - 가리키는 값의 개념)

변수 vs 함수(Function)

Function 형태의 데이터 타입도 있음.

변수와 함수는 차이점이 있음.

array와 object

[ , , , ] : 배열(array)

{ , , , } : 연관배열, JSON이라고 의미

전역 변수 vs 지역 변수

<연산자>

%, > , ==, !=

값은 True or False가 나온다.

=== 자료형까지 같아야 참 값이 된다.

<if, if~else, switch~case>와 <for, while, do~while> 조건문

switch~case는 제약 조건이 있는 조건문

<반복문>

document.write()를 10줄 사용하면 10번 사용 가능하긴 하다.

for( ; ; ){}

for문 내에서 초기값, 종료식, 증가식 순으로 사용을 한다.

관례적으로 i를 사용하여 반복문을 나타낸다.

값을 입력할 때에 값의 시작을 0으로 시작한다.

for(var i=0 > i <=9; I=i+1

i++, ++i도 같은 표현이다.

while(i < 10)

break; - 중지(반복문에서 쓰인다)

continue; - 계속

go to 현재에는 사용하지 않는 방법

클래스와 값이 같은 함수 생성자 함수라고 한다.

전체적으로는 객체라고 한다.

메서드 형태는 객체.함수로 나타낸다.

<함수 선언 = 함수 정의>

함수를 만들고 사용하여 호출한다.

JS는 유연하기 때문에 함수 선언 전에 함수를 호출해도 오류가 나지 않음.

일반적인 프로그래밍 언어에서는 순서가 맞지 않음.

“ ”, “ ” 구분을 하지 않는 편이다.

무명 함수 : 이름이 없는 함수

한 번만 사용하여 실행시키는 함수에 주로 사용하는 것이 무명함수이다.

무명 함수를 변수에 담아서 사용을 한다.

만들어주고 사용을 해야 컴파일러가 인식을 한다.

무명함수를 우선해서 사용을 한다.

변수 없이 반환값을 출력할 수 있다.

함수형이라는 변수 add변수

함수를 만들어서 쓰기에 함수형 변수를 사용한다.

<매개변수 = 중계변수 = 매개 인수 = parameter (argument는 특수한 매개체를 의미)>

매개변수의 개수가 다르거나 자료형이 다르면 다른 함수

자바에서 메소드 오버로딩이 있어 모든 함수가 작동하지만 JS에서는 메소드 오버로딩이 불가

-> Method overloading은 객체지향 프로그래밍에서 하나의 클래스 내에 같은 이름의 메소드를 여러 개 정의할 수 있는 기능

arguments : 매개변수, 매개인자, 매개인수 = parameter 중에서 독특한 parameter를 지칭  
배열과 반복문 세트로 나온다.

arguments 자체가 배열(array)이다.

<배열의 특징>

순서대로 나열된 데이터를 저장할 수 있습니다.

항목의 인덱스는 0부터 시작합니다.

Array의 길이는 동적으로 변할 수 있습니다.

배열 내에 다른 자료형의 데이터를 함께 저장할 수 있습니다.

배열은 객체이며, 객체의 일종으로 각각의 인덱스는 프로퍼티 이름으로 취급됩니다.

배열 메소드를 사용하여 다양한 작업을 수행할 수 있습니다. 예를 들어, push() 메소드를 사용하여 배열의 끝에 새로운 항목을 추가하거나, splice() 메소드를 사용하여 배열의 특정 위치에 있는 항목을 삭제할 수 있습니다.

Array는 다양한 용도로 사용되며, 데이터의 추가, 삭제, 변경, 검색 등에 유용합니다.

<Literal Array>

Literal - 문자 그대로

자료형과 관련해서 나타낸다.

“3”과 3의 차이 string과 num의 차이다.

JS에서는 리터럴 활용함으로 자료형을 문자 그대로 나타내는 것을 볼 수 있음.

널 = 오브젝트형

배열 - 1)만들고 2)사용

본디 배열은 개수를 정해주고 만든다.

배열 + 반복문 : 한 세트로 쓴다.

같은 유형만 가능.

<NaN(Not a Number)>

데이터가 Undefined 형태여서 NaN으로 나타냄.

숫자가 아닌 값을 나타내는 JavaScript의 내장된 값 중 하나입니다.

보통 연산 중에 유효한 숫자가 아닌 결과값이 생성될 때 NaN 값이 반환됩니다.

if~continue 또는 if~break 사용시 특정 조건에서 작동시킬 때 사용됨.

parseInt()나 parseFloat()를 사용 문자열에서 숫자열로 바꾸기 가능

<배열 객체(Array Object)로 생성>

클래스 + 생성자 = 객체

<Object(객체형)>

숫자형(Number - Num)

글자형(String - Str)

배열(Array)형

<Array 속성 확인하기>

typeof - 연산자

isArray() - Array Object(객체)의 Method - 클래스명.Method형태

instanceof() - Array 생성자의 연산자