

# UI 화면구현

## <Animation 효과>

transform, transition

scale, rotate, skew(밀림 변환), Matrix(선형 대수를 이용한 변환 - 행렬)

keyframe

속성 효과 - 가시화(Visibility), Display 2가지가 있음.

디스플레이가 중요도가 높음

display none : 안보이기

display block, display inline 속성값을 줘서 보이게 한다.

박스모델을 만들 때 사용한다.

Perspective(3차원 투영) ↔ Orthogonal (직교 투영)

보간 : 변화의 과정

## <JavaScript(JS)> - Programming Language

Java, Python과 같은 프로그래밍 언어

if, for, 변수, 함수 등을 활용하여 한다.

Java Python C++ C JS 등 다양한 프로그래밍 언어를 할 때 중요한 것

1) 변수와 상수에 대해서

변수 - 자료형(숫자형(Integer), 글자형(Character), 객체(날짜) 논리형)

2) 연산자

사칙연산, % , 부등호 등 (+, -, \*, %, >, <, >=, <=, ==, !=, &&, !!)

3) 제어문

분기문(Branch), 조건문 - If, Switch~Case

반복문 - For, While, do~while, foreach

4) Function(함수)

함수 또는 메소드(Method)

Method - 객체지향적인 프로그래밍 언어(Java, C++, Python)

절차 지향적인 언어(C언어)

객체지향적인 프로그래밍에서는 클래스, 객체, 상속을 사용함.

## <JS의 개요>

웹문서를 동적으로 제어하기 위하여 고안된 프로그래밍 언어이다.

제어 - 상태변환

프로그래밍으로 버튼을 눌렀을 때 상태

이벤트로 웹문서를 제어

HTML+CSS+JS = HTML5 : Front-end

HTML은 모델(데이터 구조) CSS 뷰(디자인) JS는 제어 담당을 한다.

JSP(서버) Back-end와의 연동의 역할을 하는데 있어서도 JS가 중요함.

<JS 적용하는 방법>

<head>또는 <body> 내에 작성

<body>에는 잘 쓰지 않음.

HTML 태그 안에 속성값으로 정의한다.

<button type....></button>

alert 과 같은 것으로 표현

JS 파일을 만들어서 참조시키는 방법이 있다.

<script src=...></script>로 표현

<데이터 타입>

number, string, boolean, array, object, null, undefined가 있다.

JS = Simple 단순하고 유연한(Flexible) 프로그래밍, 쉽다는 의미는 아님  
Strict(엄격한) 프로그래밍 언어 - 문법에 맞지 않는 것은 오류표시를 해준다.

자료형

변수 ↔ 상수

변수를 만들고 사용한다.

변수를 만들때는 1회 만들고 사용은 1번 이상을 써야 한다.

여러 번 쓸 가능성이 있거나 여러 번 쓸 때 변수를 사용한다.

값을 받고 저장한다는 의미가 변수

참조값으로 변수를 선언하기도 한다.

Undefined 값(비정의)

글자는 ""표시로 값을 표현

JS에서는 Var로 변수를 선언하고 처리한다.

Boolean - 논리형 T(True), F(False)의 두 개의 값만 있음

Array(배열)은 여러 개의 값을 넣을 수 있음.

Object - 객체

서로 다른 속성과 값을 하나의 집합으로 나타내는 것이다.

값을 초기화해야 함 0이나 빈 따옴표로 표기 하여 초기화한다.

null - object(객체)를 초기화시킨다.

배열 내에 Object가 포함됨.

연관 배열

JSON = JavaScript Object를 의미

변수명 작성규칙

예약어는 프로그래밍에서 쓰는 것이기 때문에 변수명으로 사용 하지 못함.

CSS에서 #id, .class로 식별자를 준다.

대소문자를 구분한다.

문자 \_ \$로 시작

한글 사용은 가능하지만 영문자 사용이 좋음.

<전역 변수와 지역 변수>

전역 변수(Global)

코드 내 어느 위치에서든 선언하여 전 영역에서 사용할 수 있는 변수

지역 변수(Local)

변수가 선언 된 해당 블록에서 선언하여 범위내에서 유효하게 사용할 수 있는 변수

지역 변수를 우선으로 하여 프로그래밍 작성하는 것이 좋음.

JavaScript ES6에서는 var, let(재선언 하지 않는 변수에 사용)을 사용하는 추세  
함수를 1회 만들고 여러 번 사용한다.

만들기 : function(입력값), 출력(return 값)형태로 지정 getGrade()로 만들

사용 : getGrade()를 호출하여 사용

return 값에 함수를 넣으면 함수 자체가 return 값이 된다.

return 값이 없을 때 사용하면 오류가 난다.

<함수>

4종류의 형태로 나온다.

in out 둘 다 없음 - 입력 값도 없고, 출력 값도 없음

f1();

document.write("");

in out 출력만 있는 것 - 입력 값은 없으나 출력 값은 있음

f2();

document.write(f2());

in out 입력만 있는 것 - 입력 값은 있는데 출력 값이 없음

f3(10);

document.write((f1()\*3);

in out 둘 다 있는 것 - 입력 값, 출력 값 모두 있음

f4(20);

document.write(f4() + 100); - 값을 출력

★ 여러 줄이 반복되는 경우 코드를 줄이기 위해서 함수를 사용한다.

디버깅 : 프로그래밍 랭귀지(Java, Python)등에서 에러를 잡을 때 사용한다.

프론트엔드는 웹브라우저에서 사용

<디버깅 방법>

F12(DevTool)에서 Source탭에서 자바스크립트를 볼 수 있음

1) break point(중단 점을 찍는다.)

2) F5를 누르면 중단점 찍은 것 에서 멈춘다.

프로그래밍에서는 결과값 보다는 과정이 중요하다.

디버깅을 통해서 프로그래밍의 로직을 볼 수 있음.

공부하는 과정에서 디버깅 해보는 것도 좋음.

프로그래밍은 기억력 또는 지능으로 하는 것이 아니라 눈으로 하는 것이 많음.  
디버깅 많이 하는 편이니 알아두면 좋음.

### <연산자(Operator)>

좌항과 우항 사이에 들어가는 것

단항, 이항, 삼항 등(다항)

단항 :  $i++$ ,  $!a$  (자기 자신이 값이 변할 때 주로 사용하는 편)

이항 : 일반적인 연산 방식,  $a+b$  형태

삼항 :  $x>y ? z=x : x=y$  형태로 복잡한 형태

앞의 조건식이 맞은 경우 다음것 아닌 경우는 그 다음것 실행할 때 사용한다.

### <연산자의 종류>

$+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ (몫을 제외한 나머지) : 산술연산

$>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $==$ (같다),  $!=$  : 비교연산

$===$  (자바 스크립트에만 있는 것) : 자료형 체크, 자료형이 같으면 True

`var a=3;`

`a == 3, a === 3` 모두 true

`""` 표시할 경우 String 표시

$\&\&$ ,  $!!$ ,  $!$ ,  $||$  논리연산

`a>3 b<7` a에 7, b에 3입력 시 둘 다 True가 나옴

`True&&True`

논리값과 논리값을 연산

`T && T - True`

`T && F - False`

`F && T - False`

`F && F - False`

비교 연산의 결과를 나타낼 때 사용

`||` 는 둘 중 하나만 True여도 True

### 대입 연산자

`a=3`은 `a=a+3`(`a+=3`으로도 표현 가능)과 같다.

`a+=1`, `a=a+1`, `a++`(코드 실행시 다음줄에서 값이 변환됨), `++a`(코드 실행시 즉시 값이 변환됨) 모두 같은 표현

### 제어문(Control Statement)

분기문 = 조건문(Branch)

`if`, `if~else`문, `switch~case`문

반복문(Loop, Riterator)

`for`문, `while`문

제어문이 한 줄일때는 중괄호를 쓰지 않아도 무방함.

`switch~case`와 `while`은 중괄호 생략하지 않음.