

JavaScript 배우기 전, 프로그래밍 기본

1. 들어가기전

Q1. 프로그래밍과 코딩은 같나?

코딩: 코드(명령어) 작성하는 일

프로그래밍: 코드를 모아놓아서, 내가 하고자 하는 일을 컴퓨터한테 시키는 것

Q2. 왜 코딩을 하지?

컴퓨터는 정해진 문법에 맞춰 똑바로 말해줘야 알아듣기 때문

Q3. 프로그래밍 언어는 각각 다를까?

비슷비슷하다. 서울 사람이 경상도말 어느정도 알아듣는 것과 같다.

여기선 다른 언어들보다 좀 쉬운 언어인 파이썬과,

파이썬에 없는 개념들을 슈도코드를 통해 설명한다. (예를 들면 변수선언, 배열 등)

슈도코드는 실제 프로그래밍 언어는 아니지만 간단하게 설명하기 위한 코드라고 이해하면 된다.

2. 주석(Comment)

왜 쓸까?

1. 니 코드가 어려워서 일주일 후에 보면 너도 모름
2. 다른 사람은 니 코드를 아예 모름

그래서 설명이 필요한데 그 설명이 주석이다.

주석이 시작되고 끝나는 부분은 컴퓨터가 해석하지 않는다.

```
# 이건 주석임  
print("hi")
```

이 경우, # 으로 시작하는 주석은 해석 안함.

여러줄 주석도 보통의 언어는 제공한다.

'''

이건 여러줄 주석

'''

주석은 언어마다 문법이 다르므로, VSCode 의 `ctr + /` 기능을 이용할 것
이미 HTML, CSS 를 통해서 주석 쓰는 법을 우린 알고 있으므로 넘어가자!

3. 변수(Variable)

수학적 변수는 머릿속에서 지워버리자.
먼저 메모리의 개념을 알아야 한다.

-	
1668	
1669	
1670	
1671	
1672	
-	

이것이 RAM 이라고 가정하고, 1669번에 jony 라는 이름을 집어넣는다고 생각하자.

-	
1668	
1669	jony
1670	
1671	
1672	
-	

근데 우리가 1669 라고 기억하고 다니긴 어렵다!
그래서 알아듣기 쉽게 이 방에 이름을 name 이라고 붙여줄 것이다.

...	
1668	
name 1669	jony
1670	
1671	
1672	
...	

즉, 변수는 메모리의 주소다.

변수가 주소라면, 해당 주소에 들어가는 값이 바뀔수도 있다는 뜻이다.

...	
1668	
name 1669	david
1670	
1671	
1672	
...	

jony에서 david로 바뀌었다. 즉, 변수는 변한다.

이제 코드로 변수를 설명해보자.

이것을 변수 선언문이라고 한다.

변수는 반드시 선언하자마자 값을 넣어주어야 한다.

이것을 초기화라고 한다.

int number = 3;

여기서,

int

변수의 타입(자료형)이다. 다음장에서 상세히 설명한다.

number

변수의 이름이다.

이름은 뭐하는놈인지 알아볼수있게 지어야한다.

=

같다는뜻이 아니다! 집어넣는다는 뜻이다.

3

변수의 값이다.

;

세미콜론이라고 읽는다. 코드 문장 마칠 때 쓴다.

여기서, = 는 같다는 뜻이 아니라 집어넣는다는 뜻이라고 했다.
즉, 다음과 같이 읽어야 한다. ‘3을 number에 넣어라’

int number = 3;



오른쪽에서 왼쪽으로 읽어라

이름짓기, 제일 중요하다.

변수든 함수든 객체든 이름짓는 게 개발자에게 가장 중요한 일이다.

개발자는 이름짓는사람이다.

제발 이름은 길게 지어도 좋으니깐 알아볼 수 있도록 짓자.

4. 자료형(Type)

자료형은 타입이라고도 한다.

정확하게는 Data Type 이다.

프로그래밍에서 여러분이 쓸 수 있는 데이터의 “종류”다.

int number = 3;

여기서, number 라는 이름을 가진 변수는 오로지 int 타입만 담을 수 있다.

int 는 integer 의 약자로, “정수”를 말한다.

1, 0, -2, 954, -217, ...

흔히 프로그래밍 언어들에서 쓰는 대표적 타입을 알아보면,

int	정수
float, double	실수 (27.831...)
string	문자열 ("조교의 웹강의", "짱", "a")
boolean	참 또는 거짓. (true, false)
array	배열
...	

```
int a = 3;  
float b = 27.831;  
String c = "조교의 웹";  
boolean d = true;
```

참고로 문자열(string)은 따옴표로 감싸줘야만한다.

즉, 다음은 문자열이지 불리언이 아니다.

"true"

5. 연산자(Operator)

컴퓨터는 계산기다. 여러분은 거대한 계산기를 만든다고 생각하면 된다.

연산 = 계산



연산자는 연산을 하기 위한 기호다. 여기선 + 가 쓰였다.

피연산자는 연산을 **당하는 것들**이다. 숫자만 말하진 않는다.

종류는 다음과 같다.

사칙연산 + - * /

곱하기는 * 다.

나누기는 / 다.

a = 1

b = 2

result1 = a + b

result2 = a - b

result3 = a * b

result4 = a / b

여기서 특수한것들이 있다.

증가 a++ 1증가

감소 a-- 1감소

```
a = 1
b = 1
a++
b--
```

결과: a는 2, b는 0

6. 배열(Array)

데이터를 모아두어 하나의 변수로 만든 것이다.

```
a = [0, 11, 31, -12];
b = ["조교", "시현", "준수"];
print(a[3]);
print(b[0]);
```

결과:

a[3] 은 -12

b[0] 은 “조교”

여기서, 같은 타입만 모아놓은것에 주의하자. a는 int만, b는 string만 모았다.

JavaScript에선 다른 타입들도 함께 모아서 배열을 만들 수 있지만,

그렇게 하지 마라

배열의 요소(element, 엘리먼트) 를 가져올 땐 인덱스(index)를 통해 가져온다.

보면 알겠지만, 배열변수 이름 다음에 대괄호 [] 써서 그 안에 인덱스를 넣는다.

근데 이상하다! a[3] 은 31 아니야?

아니야. **프로그래밍에서 숫자는 0부터 센다!**

배열의 요소도 인덱스를 통해 바꿀 수 있다.

```
a = [0, 11, 31, -12];  
a[1] = 50;  
print(a);
```

결과:

배열 a는 [0, 50, 31, -12]

7. 조건문(Conditional statements)

상황에 따라 다르게 동작시키고 싶을 때 사용한다.

```
// 내가 좋아하는 사람  
myFavorite = "엄마";  
  
if(myFavorite == "엄마") {  
    print("엄마가 좋아");  
} else if(myFavorite == "아빠") {  
    print("아빠가 좋아");  
} else {  
    print("다 싫어");  
}
```

결과: “엄마가 좋아”

if 하고 소괄호 () 안에 들어가는 게 조건(condition) 이다.

이 조건이 일치하면 중괄호 { } 안에 있는 게 동작한다.

== 이라고 = 을 두 개 썼는데, 이것이 바로 프로그래밍에서 “같다” 를 의미하는 기호이다.

즉,

= 집어넣는다

== 같다

추가로, 이것도 있다.

!= 다르다

else if 는 if 가 아닐 경우에 다른 조건을 비교할 때 쓴다.

else 는 if 든 else if 든 조건을 만족하는 게 없을 때 실행된다.

else if 와 else 는 상황에 따라 필요없을수도 있다. 다음을 보자.

```
// 내가 좋아하는 사람
myFavorite = "아빠";

if(myFavorite == "엄마") {
    print("엄마가 좋아");
}
```

이 경우, if 는 조건을 만족시키지 못하기 때문에 아예 실행되지 않으므로, 프로그램은 아무 결과도 내놓지 않는다.

다음 코드는 어떨까?

```
// 내가 좋아하는 사람
myFavorite = "아빠";

if(myFavorite != "엄마") {
    print("엄마가 좋아");
}
```

결과: 엄마가 좋아

이 경우, == 대신 != 가 쓰였다. myFavorite 이 엄마가 아니므로 “엄마가 좋아” 가 출력된다.

if else 는 여러번 쓸 수도 있다.

```
// 내가 좋아하는 사람
myFavorite = "아빠";

if(myFavorite == "엄마") {
    print("엄마가 좋아");
} else if(myFavorite == "아빠") {
    print("아빠가 좋아");
} else if(myFavorite == "누나") {
    print("누나가 좋아");
} else {
    print("다 싫어");
}
```

결과 : “아빠가 좋아”

이 경우, else if 가 두 번 쓰인 경우이다. 즉, 여러번 쓸 수 있다.

조건문에서 조건으로 흔히 쓰이는 게 바로 boolean, 즉 true 와 false 이다.
다음을 보자.

```
isFavorite = true;
if(isFavorite) {
    print("나 좋아하는 사람 있어");
}
```

결과: “나 좋아하는 사람 있어”

즉, 안에 있는 조건이 참(true) 이기 때문에 if 문이 실행된 것이다.