

본 자료의 무단전재와 복사, 배포, 상업적인 용도의 이용을 금지합니다.

Taewon Kang (서울교육대학교 과학영재교육원 튜링반 13기)

Undergraduate Student

Korea University, Department of Computer Science and Engineering

Mar 19, 2022

# What is Computer Science?

파이썬의 기본 문법

# 파이썬이란?

- \* 컴퓨터와 사람이 쉽게 대화할 수 있도록 도와주는 징검다리와 같은 프로그래밍 언어
- \* 스크래치와 같은 블록 코딩이 아닌 텍스트 코딩 형태의 프로그래밍 언어
- \* 현업(SW, AI 개발)에서 널리 사용됨.

# 파이썬 출력 (문자열)

```
>>> print('Hello World!')  
Hello World!
```

(1) 문자열을 출력할 때, 따옴표로 감싸서 출력을 합니다.  
따옴표로 감싸면, 자료형이 문자열(**str**) 이 됩니다.  
'Hello World' 라는 문자열을 출력합니다.

```
>>> print("Hello", "SNUE!")  
Hello SNUE!
```

(2) 이를 응용해서, 여러 개의 자료를 출력할 수 있습니다.  
여러 개의 자료는 쉼표(,)로 구분합니다!

## 파이썬 출력 (문자열)

```
print("Hello")
print("World")
=====
```

실행 결과:

```
Hello
World
```

```
print("안녕"*2)
print("안녕"+“하세요”)
=====
```

실행 결과:

```
안녕안녕
안녕하세요
```

(3) `print()` 를 여러 번 사용하면,  
한 줄에 하나씩 자료를 출력할 수 있습니다.

(4) 문자열 자료형(str) 에서의 덧셈과 곱셈은 각각 결합, 반복 연산으로 표현됩니다.  
즉, 서로 다른 자료형을 서로 결합하거나, 안녕을 n번 곱하는 연산이 가능합니다.

# 파이썬 출력 (숫자) 및 연산

```
>>> print(12)  
12
```

(1) 숫자를 출력할 때 따옴표를 감싸지 않습니다.  
따옴표를 감싸지 않으면, 자료형이 정수(int)로 표현됩니다.  
단순 12라는 숫자를 출력합니다.

```
>>> print(12+4)  
16
```

(2) 따옴표를 감싸지 않으면, 문자열(str)이 아닌 숫자(int)로 취급되므로,  
사칙연산이 가능합니다. (+, -, \*, / 등)

```
>>> print(12-4)  
8
```

12 빼기 4를 Python으로 표현

```
>>> print(12*4)  
48
```

12 곱하기 4를 Python으로 표현

```
>>> print(12/4)  
3
```

```
>>> print(12//4)  
3
```

12 나누기 4의 연산결과,  
12 나누기 4의 몫을 Python으로 표현

```
>>> print(12%4)  
0
```

12 나누기 4의 나머지를 Python으로 표현

# 파이썬 입력

```
a = input()  
print(a)  
=====  
12  
12
```

(1) 컴퓨터에게 자료를 전달하는 코드는 **input()** 입니다.  
입력받은 자료는 어디에 보관해야 하는데, 보관하는 공간을 변수라고 합니다.  
변수는 변하는 수라는 뜻으로, 자료를 저장하는 공간을 직접 만들어 줄 수 있습니다.  
변수 = **input()** 의 형식으로 사용자의 키보드 입력을 변수에 담아보세요.

```
a = int(input())  
print(19-a)  
=====  
12  
7
```

(2) 기본적으로 **input()** 으로 입력받은 값은 문자열(str)로 저장됩니다.  
숫자를 입력받아 사칙연산을 하고 싶다면 자료형 변환을 해 주어야 합니다.  
str() 자료형을 int() 로 변환하기 위하여, int() 로 **input()**을 감싸주세요.

왼쪽 예시는 사용자의 나이를 입력받아 성인(만 19세) 이 되려면 몇년이 남았는지  
구하는 코드입니다.

```
a = 3 # 1을 a에 저장  
b = "ABC" # "ABC" 를 b에 저장  
c = b * a # "ABCABCABC" 를 c에 저장
```

(Tip) **=** 는 대입연산자입니다.  
오른쪽에 있는 값을 왼쪽 변수에 저장해 줍니다.  
왼쪽은 변수 자리이므로, 숫자를 왼쪽에 배치할 수 없습니다.

# 파이썬 조건문

```
if a > 5:  
    print(a)
```

조건을 지정합니다. 만약 a가 5보다 크다면?

콜론을 꼭 넣습니다.

```
if a > 5:  
    print(a)
```

4칸 들여쓰기(indent) 합니다. PyCharm에서 <tab> 키를 누르면 자동으로 생깁니다.

```
A > B # A가 B보다 큰지?  
A < B # A는 B보다 작은지?  
A == B # A와 B는 같은지?  
A != B # A와 B는 다른지?  
A >= B # A가 B보다 크거나 같은지?  
A <= B # A가 B보다 작거나 같은지?
```

(1) python에서는 if를 사용하여 조건을 정합니다.

if 뒤에 조건을 써주고, 조건이 참(True)인 경우 아래 명령을 실행합니다.  
그 아래줄에 들여쓰기를 해서 명령을 써 줍니다.

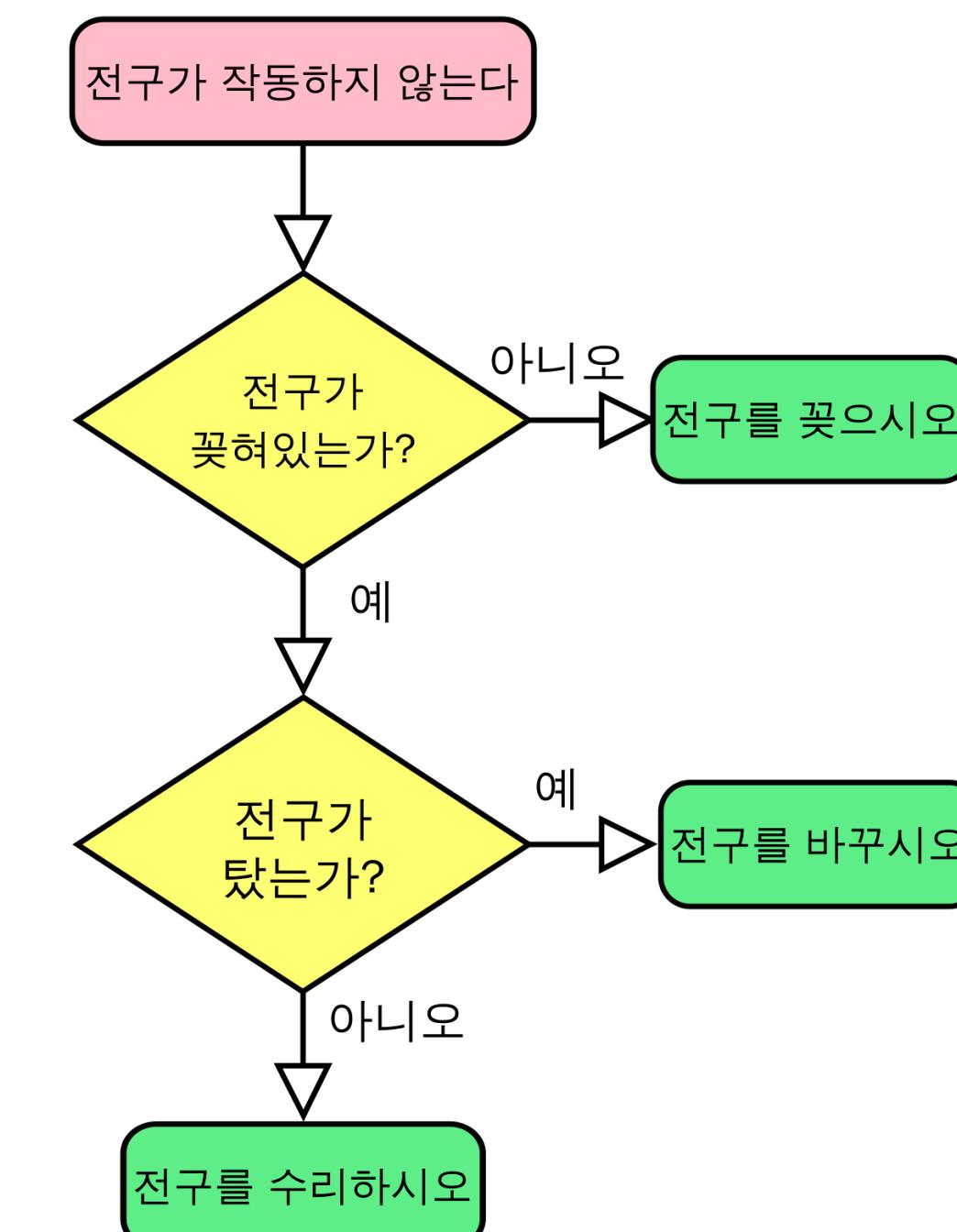
(Tip) == 를 포함해 왼쪽에 있는 연산자는 비교연산자입니다.

왼쪽에 명시된 연산자는 if문에 사용이 가능한 비교연산자로,  
왼쪽 질문에 대한 결과를 참(True) 또는 거짓(False)로 알려줍니다.

따라서, a가 3인지를 if문의 조건으로 쓰고 싶다면,

if a == 3: 으로 표현해 주셔야 에러 없이 코드를 만들 수 있습니다.

절대 if a = 3: 으로 착각하지 않도록 합니다!!!



# 파이썬 조건문

```
if a > 0:  
    print(a)  
else:  
    print(-1*a)
```

## (2) 조건에 해당되지 않아도 명령을 실행할 수 있습니다.

조건에 해당되지 않는지 검사하는 명령이 else 문입니다.

Else문은 위에 명시한 if문의 조건에 해당하지 않을 때 실행됩니다.

ex) 왼쪽의 코드는 절대값을 구하는 코드의 예시입니다.

```
if score >= 95:  
    print("A+")  
elif score >= 90:  
    print("A0")  
else:  
    print("B+")
```

## (2) 꼭 if문의 조건에 해당되지 않아도 다른 조건을 검사할 수 있습니다.

if문 조건에 해당되지 않았을 때 다른 조건으로 검사하는 명령이 elif 문입니다.

elif문은 위에 명시한 if문의 조건에 해당하지 않을 때 실행되고,

만약 모든 조건에 해당되지 않는 경우 else문을 추가할 수 있습니다. (선택)

# 파이썬 반복문

python의 반복문에는 일정한 범위 내에서 반복하는 for문과, 조건이 참인 동안 반복하는 while문이 있습니다.

```
for i in [1,2,3]:  
    print(i)  
=====
```

실행 결과:

```
1  
2  
3
```

(1) 리스트의 범위를 조절하여 반복 횟수를 결정할 수 있습니다.

여러 개의 자료를 담는 자료형을 리스트라 하고, 다음과 같이 구성됩니다.

```
a = [1, 2, 3, 4, 5]
```

리스트에는 인덱스가 있으며, 0부터 시작합니다. ex) a[1] # 2

현재 이 코드에서는 1, 2, 3을 순서대로 리스트에서 빼내 i 변수에 넣고,  
3번 반복하며 i를 각 반복에서 출력하고 있습니다.

```
i = 0  
while i < 5:  
    print(i)  
    i += 1
```

(2) while문은 조건이 참인 동안 계속 반복됩니다.

i가 5보다 작은 경우까지 반복되므로,

i값은 0, 1, 2, 3, 4까지 출력되고 반복문을 빠져나오게 됩니다.