

# Introduction to python

3

■ Contents 프로그래밍 언어란 변수 자료형 제어문 함수

# 프로그래밍 언어란?

처음 배워보는 프로그래밍 언어

python basic

## 프로그래밍 언어

00

컴퓨터 시스템을 구동 시키는 소프트웨어를 작성하기 위한 형식 언어

저급 언어

기계어 어셈블리어

저급 언어

베이직 C C# C++ 파이썬

session3.

# python 이란?

쉽고, 간결한 파이썬

python basic

**“Life is too short, You need python”**

01

파이썬은 가장 인간다운 언어이다

파이썬은 문법이 쉬워서 빠르게 배울 수 있다

파이썬은 프로그래밍을 즐길 수 있게 해준다?

# python 으로 할 수 있는 것들

웹 개발, 머신러닝, 딥러닝, 데이터사이언스 등등...

Tensorflow keras  
**Artificial Intelligence**

numpy pandas  
**Data Science**

# Web development

Data crawling

django flask

등등 너무 많지만..! 우리가 할 것은 장고와 크롤링

# python에서 조심해야하는 들여쓰기와 띄어쓰기..

python이 코드를 이해하는 방법

python basic

python은 중괄호를 작성하지 않아서 들여쓰기와 띄어쓰기를 잘해주어야 합니다. -(근본x)-

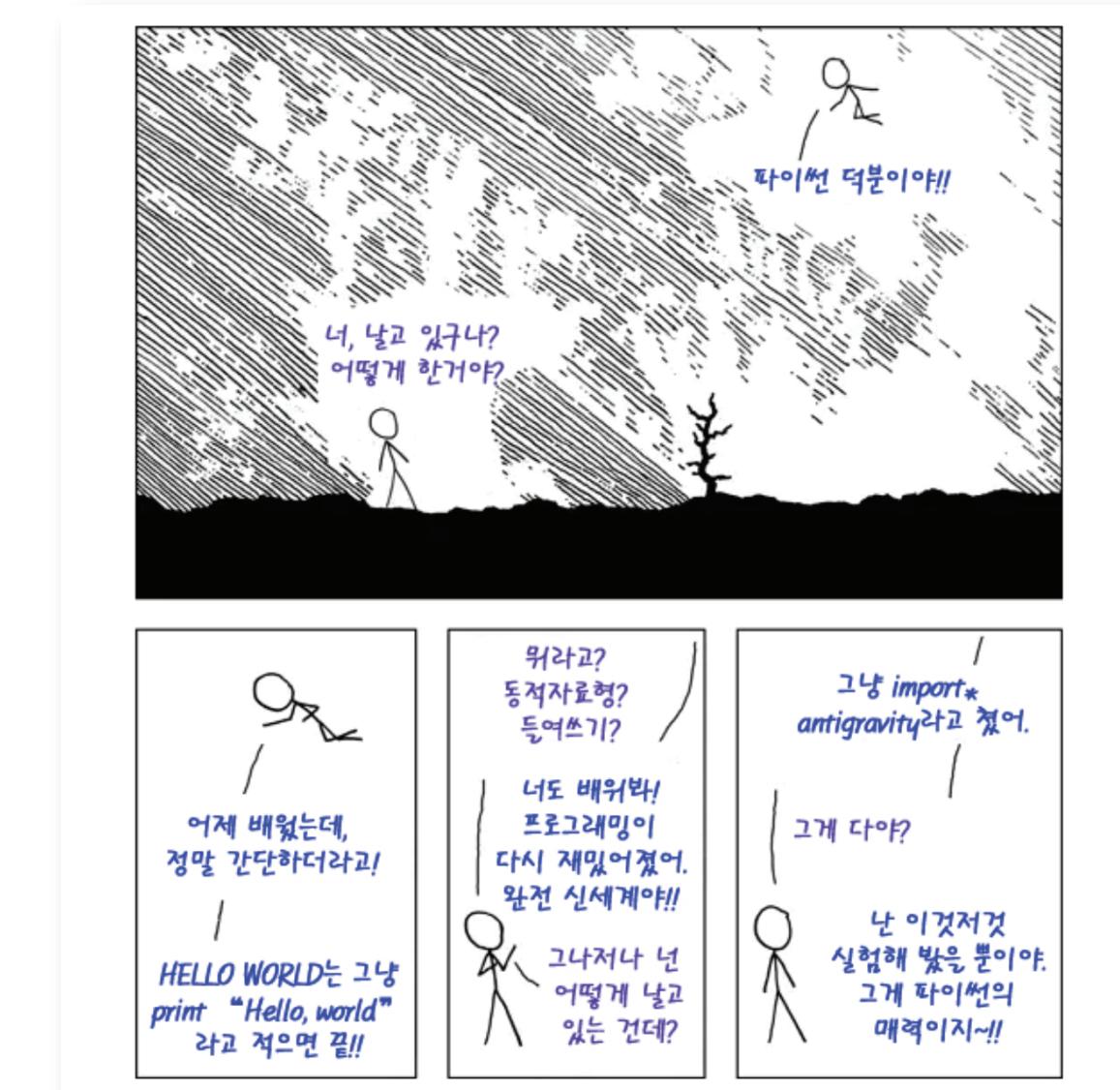
Python	JAVA	C
print("Hello, World");	public class Main { public static void Main (Srting[] args) { System.out.println("Hello, World"); } }	#include <stdio.h> int main() { printf("Hello, World"); return 0; }

03

session3.

# 그래도 좋은 언어이긴 합니다..!

python basic



04

session3.

# Variable & Data Type

# 변수와 자료형이란?

프로그래밍의 기초, 자료형

- 변수 변수란 값을 저장하기 위한 이름(자료형을 가리키는 것)
- = =은 같은 의미가 아닌 ‘대입’입니다.

**X = ????**

파이썬은 type을 지정해주지 않아도 되는 언어여서 어떤 자료형이든 받을 수 있습니다!

하지만....

# 그래도 어떠한 것들이 있는지 꼭 알고 있어야 합니다!

프로그래밍의 기초, 자료형

python basic

## python의 자료형

06

계산 프로그램을 만들려면 어떤 것을 계산할지부터 알아야 하고,  
데이터베이스 프로그램을 만들려면 어떤 자료를 저장할지부터 알아야 하는 것처럼

자료형      숫자 자료형      문자열 자료형      리스트 자료형      튜플 자료형      딕셔너리 자료형

집합 자료형      불 자료형

# 숫자형

## 숫자형을 활용하기 위한 연산자

사칙연산      +      -      \*      /

제곱연산자      \*\*

나눗셈 후 나머지 반환      %

나눗셈 후 몫을 반환      //

# 사칙 연산

+, -, \*, /

python basic

터미널에서 python 또는 python3 를 치면  
인터프리터가 나옴!

그냥 b 를 치는 것이나 print(b) 나 같아요!

```
>>> a = 1
>>> a
1
>>> b = a + 2
>>> b
3
>>> print(b)
3
>>> c = a * 2
>>> c
2
>>> d = c / 2
>>> d
1.0
```

# 제곱 연산자와 나눗셈

% -> 나머지를 반환  
// -> 몫을 반환

```
>>> a = 2 ** 2  
  
>>> a  
  
4  
  
>>> b = a % 1  
  
>>> b  
  
0  
  
>>> c = a // 1  
  
>>> c  
  
4
```

# 문자열

## 문자형을 활용하기 위한 문법들

기본	“ ”	‘ ’	“““ ”””	““ ””
줄바꿈	\n	\t		
연산자	+	*	len()	
인덱싱과 슬라이싱	a[0]	a[0:3]		
문자열 관련 함수들	count	find	index	join
	strip	split	replace	upper

# 문자열 작성하는법

과제 하셨으니 다 아시죠?

| 똑같으니깐 아무거나 쓰세요!  
| 단, 한 파일에서 일관되게만 해주세요.

```
>>> a = "likelion"  
>>> a  
'likelion'  
>>> b = 'likelion'  
>>> b  
'likelion'
```

# 문자열 연산 :: 더하기와 곱하기

| 곱하기는 문자열과 정수끼리만 가능!

```
>>> a = 'likelion'  
>>> b = 'good'  
>>> a + b  
'likeliongood'  
>>> a + ' ' + b  
'likelion good'  
>>> c = a * 2  
>>> c  
'likelionlikelion'  
>>> d = a * b  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: can't multiply sequence by non-int of type 'str'
```

# 문자열 길이 구하기 :: len()

정말 말 그대로 길이입니다....

len() 이라는 함수에 문자열을 던지면, 길이를 반환한다고 기억해주세요!

```
>>> a = 'likelion'  
>>> len(a)  
8
```

# 인덱싱

**파이썬은 숫자를 0부터 셉니다!**

a[0] 은 첫번째, a[-1] 은 마지막 으로 기억해주세요

```
>>> a = 'likelion'  
>>> a[0]  
'l'  
>>> a[2]  
'k'  
>>> a[-1]  
'n'
```

# 슬라이싱

a[0:3] index 0에서 index 3 전까지 슬라이싱 하겠다!

python basic

0 1 2 3 4 5 6 7  
l i k e l i o n

```
>>> a = 'likelion'  
>>> a[0:3]  
'lik'
```

# 문자 개수 세기 :: count()

python basic

| 문자열에서 세고자 하는 문자를 인자로!

```
>>> a = 'likelion'  
>>> count_l = a.count('l')  
>>> count_l  
2
```

16

# 문자 위치 찾기 :: find()

찾고자 하는 문자가 있으면 -> 처음 인덱스  
찾고자 하는 문자가 없으면 -> -1 을 반환

```
>>> a = 'likelion'  
>>> find_l = a.find('l')  
>>> find_l  
0  
>>> find_z = a.find('z')  
>>> find_z  
-1
```

# 문자열 삽입 :: join

| 각각의 문자 사이에 삽입!

```
>>> a = 'aaaaaaaa'  
>>> '!'.join(a)  
'a!a!a!a!a!a!a'
```

# 소문자를 대문자로 바꾸기 :: upper()

```
>>> a = 'likelion'  
>>> upper_a = a.upper()  
>>> upper_a  
'LIKELION'
```

# 양쪽 공백 지우기 :: strip()

| 나중에 크롤링할 때 유용해요~~

```
>>> a = ' likelion '
>>> strip_a = a.strip()
>>> strip_a
'likelion'
```

# 문자열 나누기 :: split()

python basic

```
>>> a = 'l*i*k*e*l*i*o*n'  
>>> split_a = a.split('*')  
>>> split_a  
['l', 'i', 'k', 'e', 'l', 'i', 'o', 'n']
```

21

| 인자로 아무것도 안넣으면 공백문자 기준!

# 문자열 바꾸기 :: replace()

python basic

```
>>> a = 'likilion'  
>>> a.replace('liki','like')  
'likelion'
```

22

# 즐거운 문자열 실습 시간 전에! vscode로 파일 출력하기

1. 실습 파일 다운 받고 원하는 위치에 옮기기!



2. 옮긴 위치를 터미널로 찾아서 이동해보기! (ls 했을 때 실습파일이 나와야해용)



3. python string\_practice.py 로 실습 파일 실행 시켜보기



모르겠으면 운영진 불러주세요~~

# vscode 세팅했으면, 즐거운 문자열 실습시간

python basic

**중훈의 방해를 이기고 문자열을 출력해주세요!!**

24

**문자열 새로만드는거 아니고 문자열 함수로 수정해주세요~**

**string\_practice.py**

# bool 자료형

| 참 / 거짓을 나타냅니다~

```
>>> 0!=1  
True  
>>> 2!=2  
False  
>>> |
```

# 리스트 자료형

다른 언어에서는 배열(array)라고 불러요!

python basic

| 어떠한 자료형이든지 올 수 있음!

```
>>> a = [1,2,3,4]
>>> b = ['1','2','3','4']
>>> c = ['a',1,'likelion',4]
>>>
>>> d = list()
>>>
>>> 
```

# 리스트의 인덱싱과 슬라이싱

| 문자열에서 한거랑 똑같아요~

```
>>> a = [1, 2, 3, 4]
>>> a[0]
1
>>> a[-1]
4
>>> a[0:3]
[1, 2, 3]
>>> a[1:3]
[2, 3]
>>>
```

# 리스트의 수정, 삭제, 삽입( append() )

append 와 del 잘 기억해두세요~

```
list.py
1 # 리스트의 수정, 삭제, 삽입
2
3 # 수정
4 list_a = ['a', 'b', 'c']
5 print('수정 전', list_a)
6
7 list_a[0] = 'z'
8 print('수정 후', list_a, '\n')
9
10 # 삭제
11 list_b = ['a', 'b', 'c']
12 print('삭제 전', list_b)
13 del list_b[1]
14 print('삭제 후', list_b, '\n')
15
16
17 # 삽입
18 list_c = ['a', 'b', 'c']
19 print('삽입 전', list_c)
20 list_c.append('d')
21 print('삽입 후', list_c)
22
23
```

# 튜플 자료형(그냥 리스트 쓰세요..)

수정, 삽입, 삭제 다 안됨.. 그냥 리스트 쓰세요

python basic

```
>>> tuple1 = (1,2,3)
>>> tuple1[0]
1
>>> tuple1[0] = 5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object does not support item assignment
>>>
>>> █
```

29

# 딕셔너리 자료형

python basic

## Key : value

30

이름 : 홍길동

나이 : 100

생년월일 : 1920-01-01

session3.

# 딕셔너리 자료형은 중요해요

딕셔너리는 key로 접근하여 value값을 가져올 수 있음!

```
dict.py
1  # 딕셔너리 -> 딕셔너리는 key : value의 페어다!
2
3  user_dictionary = {
4      'name': 'juwon',
5      'age': '24',
6      'school': 'korea university'
7  }
8  print('\n')
9  print(user_dictionary, '\n')
10
11 user_name = user_dictionary['name']
12 user_age = user_dictionary['age']
13 user_school = user_dictionary['school']
14
15 print('이름은 ? ', user_name, '\n')
16 print('나이는 ? ', user_age, '\n')
17 print('학교는 ? ', user_school, '\n')
18
```

# 셋 자료형..

정말 말그대로 set, 집합을 말함

집합과 마찬가지로 순서는 고려하지 않아요  
차집합, 합집합 등등 많은데 필요할 때 찾아보시면..

```
>>> set_of_num = set(['1', '2', '3', '4', '5'])  
>>> set_of_num  
{'5', '2', '3', '1', '4'}  
>>>
```

# Control Statement

# if else

만약에 말야..

python basic

```
#####
phone_call = 112

if phone_call == 112:
    print('112 입니다 무엇을 도와 드릴까요')
else:
    print('잘 못 거셨어요')
```

33

session3.

# 조건문 : in, not in

간단하게 list 에 있는지 조사할 수 있음!

python basic

```
user_list = ['주원', '중훈', '상하', '성우', '형제', '경연', '지환', '준태']

if '주원' in user_list:
    print('주원은 있어요')

if '현재' in user_list:
    print('현재는 있어요')
else:
    print('현재는 없어요ㅠ')
```

34

# while 문

~하는 동안 돌거야

python basic

```
tree_hit = 0

while tree_hit < 10:
    tree_hit = tree_hit + 1
    print("나무를 %d번 찍었습니다." % tree_hit)

    if tree_hit == 10:
        print("나무 넘어갑니다.")

# 위에랑 아래는 달라요.. 파이썬은 들여쓰기로 차별하거든요

# while tree_hit < 10:
#     tree_hit = tree_hit + 1
#     print("나무를 %d번 찍었습니다." % tree_hit)

# if tree_hit == 10:
#     print("나무 넘어갑니다.")
```

포매팅이 뭐죠...?

35

# 포매팅

숫자형 포매팅 : %d, 문자형 포매팅 : %s

```
print('몇번 : %d' % 10)

number = 99
print('몇번 : %d' % number)

print('이름: %s' % '주원')
name = '주원'
print('이름: %s' % name)
```

# while 문 탈출

break

python basic

```
# break

coffee = 10
money = 300

while money:
    print("돈을 받았으니 커피를 줍니다.")
    coffee = coffee - 1
    print("남은 커피의 양은 %d개입니다." % coffee)
    if coffee == 0:
        print("커피가 다 떨어졌습니다. 판매를 중지합니다.")
        break
```

37

session3.

# 밑에는 무시하고, 처음으로 돌아가기!

continue

python basic

```
# continue

a = 0
while a < 10:
    a = a + 1
    if a % 2 == 0:
        continue
    print(a)
```

38

session3.

# for 문

python basic

```
users = ['주원', '준태', '중훈', '지환', '성우', '형제', '경연']

# 기본 문법
for user in users:
    print(user)
```

39

# Intro

# break 와 continue

python basic

```
# break
for user in users:
    if user == '중훈':
        break
    print(user)
```

```
# continue
for user in users:
    if user == '중훈':
        continue
    print(user)
```

40

# range

python basic

```
# range
a = range(10)
print(a[9])

for i in range(10):
    print(i)
```

# 즐거운 제어문 실습 시간

python basic

## 문제 1 과 문제 2를 풀어주세요!

42

`control_practice.py`

# 과제 안내

python basic

4월 20일까지  
**파이썬 : 과제 1~4**

4월 27일까지  
**HTML / CSS : 페이스북 모바일 클론코딩**

첨부된 1, 2 이미지를 보면서 만들어주세요! 페이스북 웹은 개발자 도구로 볼 수도 있으니깐요 :)