

LECTURE 1

DOHYUNG KIM

WELCOME

- This is the Python Programming course
- Today
 - Introduction to the course
 - Preliminaries

COURSE INTRODUCTION

- What is this course about?
 - Learn how to make an application using python language

ADMINISTRATIVE INFORMATION

- Class Hour & Room
 - Online classes for both Lecture and Lab session
- Office Hours
 - With prior reservation via email : d.kim@kangwon.ac.kr

INSTRUCTOR

- Dohyung Kim
 - Assistant professor in Department of Computer Science & Engineering
 - #401 in Hanbit bldg.
 - E-mail: d.kim@kangwon.ac.kr
 - sites: <https://icn.kangwon.ac.kr>

TEXTBOOK & REFERENCES

- Textbook & References
 - No official textbook
- However, you may refer to any kinds of materials
 - e.g., <https://docs.python.org/ko/3/tutorial/index.html>

GRADING POLICY (TENTATIVE)

- Attendance: 10%
- Lab Session: 30%
- Midterm & Final: 60%
- You will fail this course when 1) missing more than 4 class hours or 2) cheating in assignment or projects

PYTHON

- Created by Guido van Rossum, and released in 1991
- Used for
 - Web development (server-side)
 - Software development
 - Mathematics
 - System scripting
- The latest version is python 3, which we will use in this course
 - Python 2 is still popular

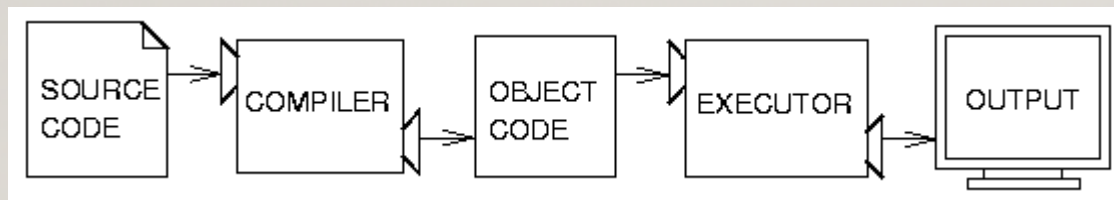
WHY PYTHON?

- Python **works on a variety of platforms** (Windows, Mac, Linux, Raspberry pi, etc.)
- Python has a **simple syntax** similar to the English language
- Using python, programs could be developed with **fewer lines** than some other languages
- Runs on an **interpreter system**



Reading lines and perform computation

cf) compiler language



Read program and translates it completely before program starts running

DEVELOPMENT ENVIRONMENT

- You can install python 3 in your machine and use a variety of IDEs
 - Python 3
 - <https://www.python.org/downloads/>
 - IDEs
 - Wing IDE - <http://www.wingware.com>
 - pyDev - <https://www.pydev.org>

OR

- Online Python IDE
 - <https://repl.it/languages/Python3>
 - We will use this IDE in the lecture class

TWO MODES

- Command-line mode

\$ python

>> print (1+1)

2

```
Python 3.8.1 (default, Feb  2 2020, 08:37:37)
> print("Welcome to the Python Programming Class")
Welcome to the Python Programming Class
>
> def greeting():
...     print("Hello, everyone!")
...
> greeting()
Hello, everyone!
> 
```

- Script mode (write a program in a file named oooo.py)

\$ python oooo.py

Online Python IDE, and online Python RE...
in Python, and host your programs and apps ...

save 

run 

share 

+ new repl

talk 

main.py  saved 

1 print("Welcome to Python Programming Class")

2 print("I hope we will have wonderful time")

Welcome to Python Programming Class
I hope we will have wonderful time
>

PYTHON COMMENT

- Comment out a single line
 - Add a '#' at the start of each line
- Comment out multiple lines
 - Put a treble quotes('') at the start and end of the block

```
'''  
This is a comment block.  
We can comment out multiple lines.  
'''  
  
print("Welcome to the Python Programming Class")
```

ERRORS IN PYTHON

- Syntax errors (compile-time errors)

```
print("Syntax error!")
```

```
File "main.py", line 1
    print("Syntax error!")
                                ^
SyntaxError: EOL while scanning string literal
> |
```

- Runtime errors

```
print(1/0)
```

```
Traceback (most recent call last):
  File "main.py", line 1, in <module>
    print(1/0)
ZeroDivisionError: division by zero
> |
```

- Logical errors (compiles and runs, but wrong)

```
print("1+1 = 3")
```

```
1+1 = 3
> |
```

BASIC CONSOLE OUTPUT

- Basic print function

```
print("Hello")  
print("everyone")
```

```
Hello  
everyone  
> |
```

- Print on the same line

```
print("Hello", end=" ")  
print("everyone")  
print("everyone")
```

```
Hello everyone  
> |
```

- Print multiple lines

```
print("Hello", "everyone")  
print()  
a = 3  
b = 4  
c = (a**2 + b**2)**0.5  
print("side a : ", a)  
print("side b : ", b)  
print("hypotenuse : ", c)
```

```
Hello everyone  
  
side a : 3  
side b : 4  
hypotenuse : 5.0  
> |
```


BASIC CONSOLE INPUT

- Input a string

```
name = input("Input your name: ")  
print("Your name is", name)
```

```
Input your name: Albert  
Your name is Albert  
➤ █
```

- Input a number

```
val = input("Enter a number:")  
print("one half of", val, "=", val/2)
```

```
Enter a number:4  
Traceback (most recent call last):  
  File "main.py", line 2, in <module>  
    print("one half of", val, "=", val/2)  
TypeError: unsupported operand type(s) for /: 'str' and 'int'  
➤ █
```

- Input a number with int()

```
val = int(input("Enter a number:"))  
print("one half of", val, "=", val/2)
```

```
Enter a number:4  
one half of 4 = 2.0  
➤ █
```

IMPORT MODULES

- Call without importing

```
print(math.factorial(5))
```

```
Traceback (most recent call last):  
  File "main.py", line 1, in <module>  
    print(math.factorial(5))  
NameError: name 'math' is not defined  
> |
```

- Call with importing

```
import math  
print(math.factorial(5))
```

OR

```
from math import factorial  
print(factorial(5))
```

```
120  
> |
```

IMPORT MODULES

- When using math functions open, you can use shorter names

```
import math
f = math.factorial
print(f(5))

sin = math.sin
pi = math.pi
degree = float(input("input degree:"))
radian = degree*2*pi/360
print(sin(radian))
```

```
120
input degree:90
1.0
> 
```

QUESTION?
