Basic Python programs, functions

What is python

- Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

Python features

- Object oriented language
- Interpreted language
- Supports dynamic data type
- Independent from platforms
- Simple and easy grammar
- High-level language
- Automatic memory management
- It’s free (open source)!

Why learn python? (cont.)

- Reduce development time
- Reduce code length
- Easy to learn and use as developers
- Easy to understand codes
- Easy to extend to other languages

Where to use python?

Python is huge collection of standard library which can be used for the following:

- Machine Learning
- GUI Applications
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV,)
- Multimedia
- Scientific computing
- Text processing and many more..

Installing Python

Windows:

- Download Python from http://www.python.org
- Install Python.
- Run **Idle** from the Start Menu.

Mac OS X:

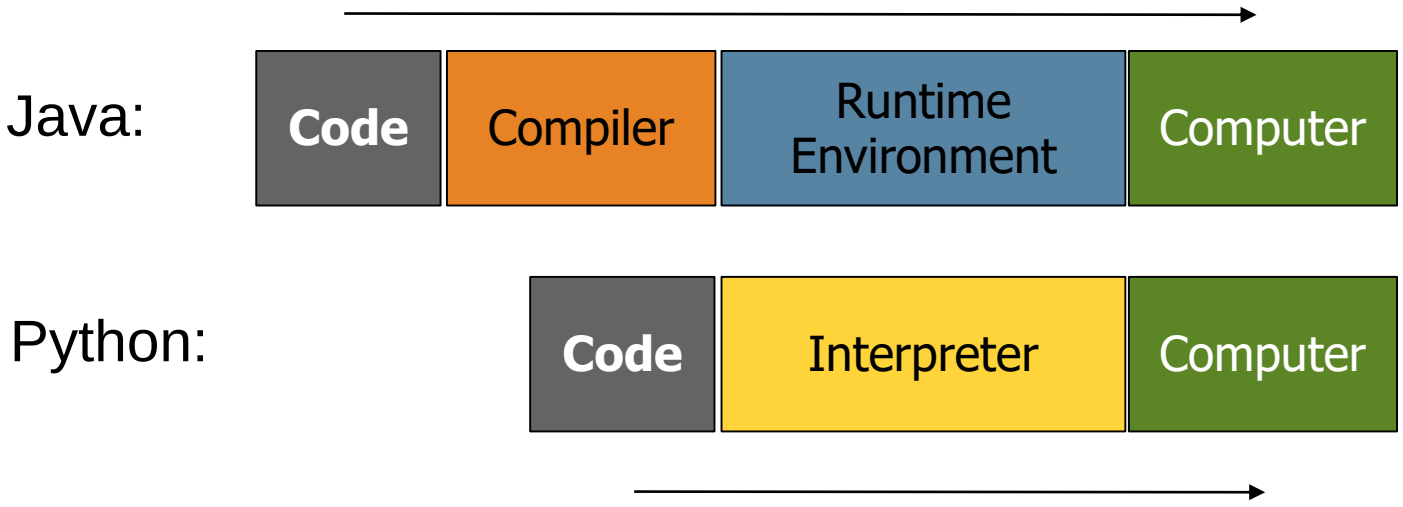
- Python is already installed.
- Open a terminal and run `python` or run Idle from Finder.

Linux:

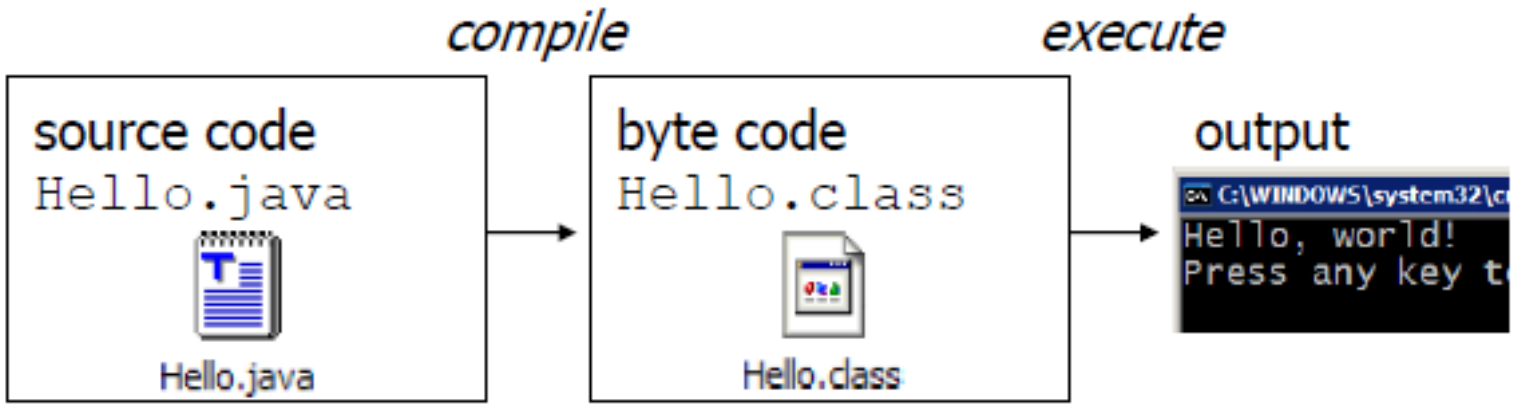
- Chances are you already have Python installed. To check, run `python` from the terminal.
- If not, install from your distribution's package system.

Interpreted Languages

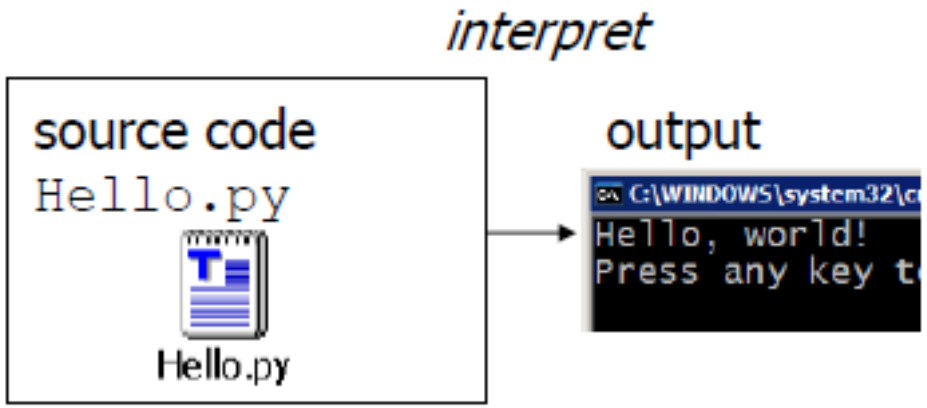
- **interpreted**
  - Not compiled like Java
  - Code is written and then directly executed by an **interpreter**
  - Type commands into interpreter and see immediate results



- Many languages require you to *compile* (translate) your program into a form that the machine understands.



- Python is instead directly *interpreted* into machine instructions.



Python Syntax

- **Much of it is similar to C syntax**
- **Exceptions:**
  - missing operators: `++`, `--`
  - no curly brackets, `{ } ,` for blocks; uses whitespace
  - different keywords
  - lots of extra features
  - no type declarations!

Python Indentation

- Indentation refers to the spaces at the beginning of a code line.
- Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.
- Python uses indentation to indicate a block of code.

Python Comments

Comments starts with a #, and Python will ignore them:

Syntax

#This is a comment
print(“hello level 14 welcome to python program")

Comments can be placed at the end of a line, and Python will ignore the rest of the line:

print(“hello levle 14 welcome to python program") #first program

Python Program

- Python does not have a main method like Java
  - The program's main code is just written directly in the file
- Python statements do not end with semicolons

hello.py

```
1 print("Hello, world!")
```

Print statement

The Python print() function is often used to output variables or text

Syntax:

```
print ("Message“)
 Print ( Expression)
```

In the print() function, you output multiple variables, separated by a comma:

- print (Item1, Item2, ..., ItemN)

Examples:

```
print (‘Hello, world!’)
 x = 35
print (x)
```

Output:

```
Hello, world!
35
```

Print statement……..con’t

Notes

- Elements separated by commas print with a space between them
- A comma at the end of the statement will not print a newline character:

```
print ('hello', 'there'  )
```

output:

```
hello there
```

input statement

input : Reads a number from user input.

You can assign (store) the result of input into a variable.

Example:

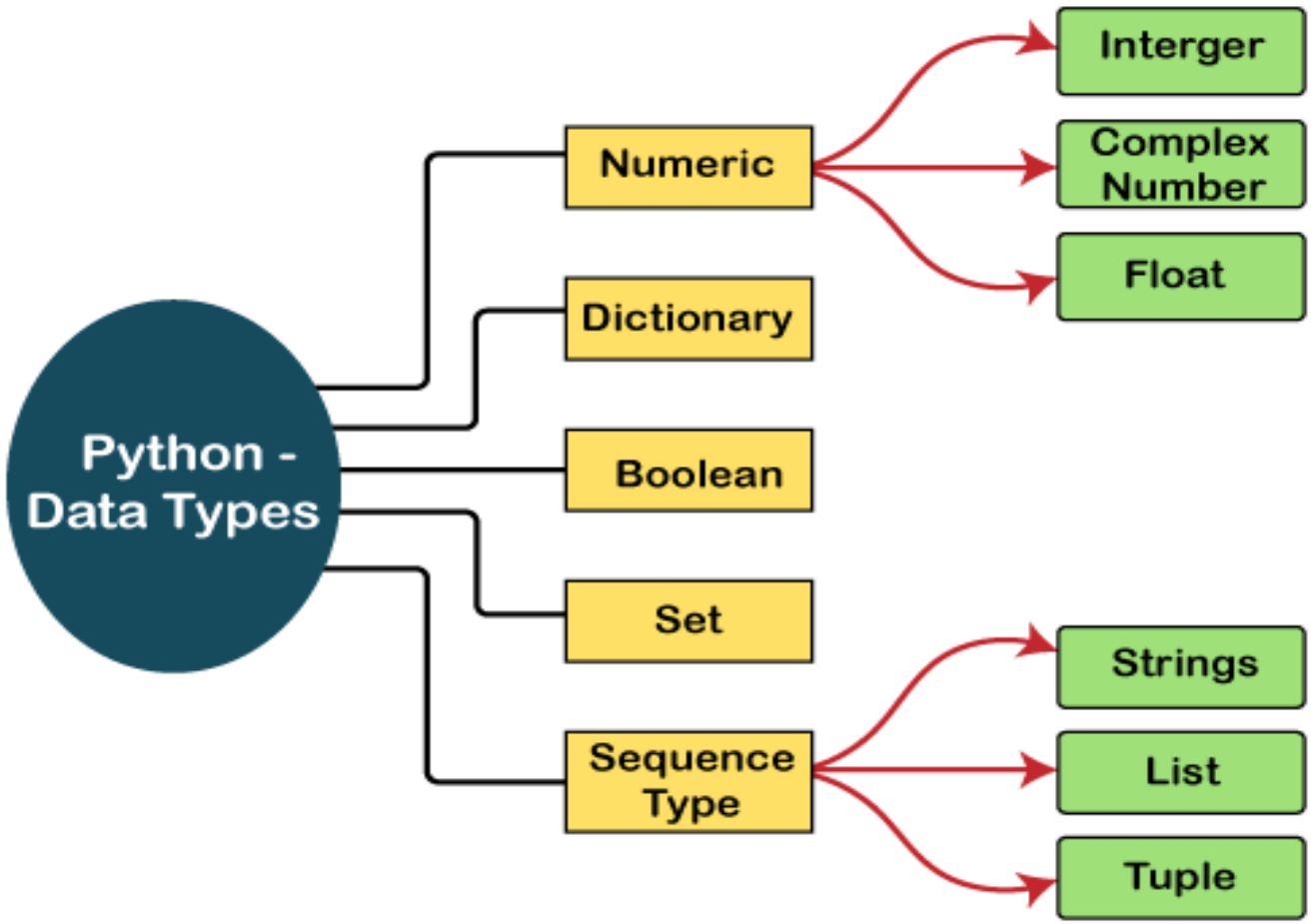
```
age = input("How old are you? ")
print ("Your age is", age)
```

Output:

```
How old are you? 22
Your age is 22
```

Variable Names

- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (age, Age and AGE are three different variables)


Python -
Data Types
Numeric
Interger
Complex
Number
Float
Dictionary
Boolean
Set
Sequence
Type
Strings
List
Tuple

Data type

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple
Set Types:	set
Boolean Type:	bool

Creating Variables

- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.

A= 5
B= “Sara"
Variables do not need to be declared with any particular *type*, and can even change type after they have been set.

Assignment value to the variables

Python allows you to assign values to multiple variables in one line:

Example:

x, y, z = “math", “Art", “science“

A,b=3,6

assign the *same* value to multiple variables in one line:

Example:

X,y=4

Casting and Get the Type

- If you want to specify the data type of a variable, this can be done with casting.

```
x = str(3)    # x will be '3'
y = int(3)    # y will be 3
z = float(3)  # z will be 3.0
```

you can get the data type of a variable with the type() function:

```
x = 5
y = "John"
print(type(x))
print(type(y))
```

```
int
 str
```

Python Expression

Arithmetic operators we will use:

+ - * / addition, subtraction/negation, multiplication, division

% modulus, a.k.a. remainder

** exponentiation

precedence: **Order in which operations are computed.**

* / % ** have a higher precedence than + - 1 + 3 * 4 is 13

Parentheses can be used to force a certain order of evaluation.

(1 + 3) * 4 is 16

Conditional Statements

The syntax of the if-statement is given below.

```
if expression:
    statement
```

Example:

```
num = int(input("enter the number?"))
if num%2 == 0:
    print("Number is even")
```

if-else statement

- Compare two integers and print the min.

```
if  x < y:
        print (x)
else:
        print (y)
print ('is the minimum')
```

1. Check if x is less than y.
2. If so, print x
3. Otherwise, print y.

The elif statement

Syntax:

```
if expression 1:
    # block of statements

elif expression 2:
    # block of statements

elif expression 3:
    # block of statements

else:
    # block of statements
```

Example

```
number = int(input("Enter the number?"))
if number==10:
    print("number is equals to 10")
elif number==50:
    print("number is equal to 50");
elif number==100:
    print("number is equal to 100");
else:
    print("number is not equal to 10, 50 or 100");
```