INTRODUCTION TO PROGRAMMING USING PYTHON

Prepared by
Josephine Boles

Outline

- Course Objectives
- History & features
- Why Python?
- How does python work?
- Install & Hello World
- Variables & Data Types
- Operators
- Numbers
- Strings
- Exercises

Course Objectives

Learn about Python, its uses and really understand it

History

- The implementation of Python was started in December 1989 by Guido Van Rossum at CWI in Netherland

- In February 1991, Guido Van Rossum published the code (labeled version 0.9.0)

- In 1994, Python 1.0 was released with new features like lambda, map, filter, and reduce.

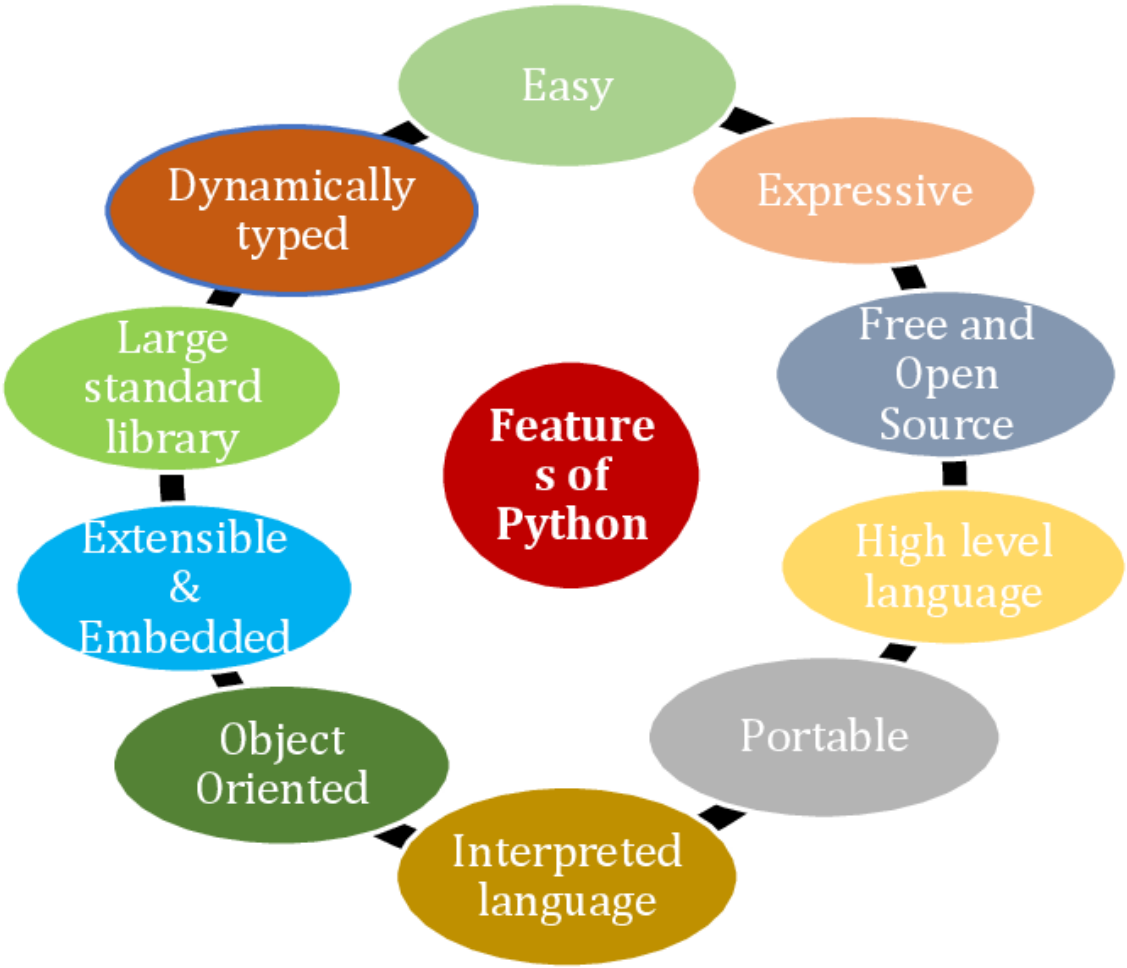
History

- Python 2.0 added new features such as list comprehensions, garbage collection systems.

- On December 3, 2008, Python 3.0 (also called "Py3K") was released. It was designed to rectify the fundamental flaw of the language.

- comedy series "Monty Python's Flying Circus". It was late on-air 1970,he select a name which unique, sort, and little-bit mysterious.
- used in every technical field, such as Machine Learning, Artificial Intelligence, Web Development, Mobile Application, Desktop Application, Scientific Calculation, etc.

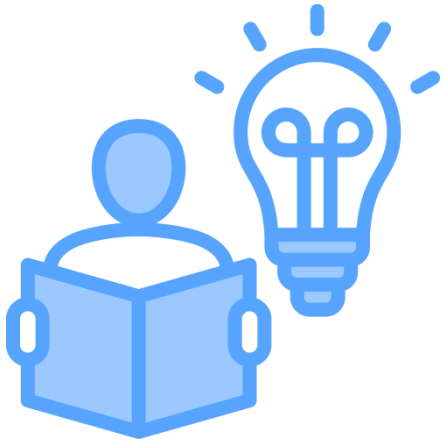
Features


Easy
Expressive
Free and Open Source
High level language
Portable
Interpreted language
Object Oriented
Extensible & Embedded
Large standard library
Dynamically typed
Features of Python

Why Python

Rapid Development

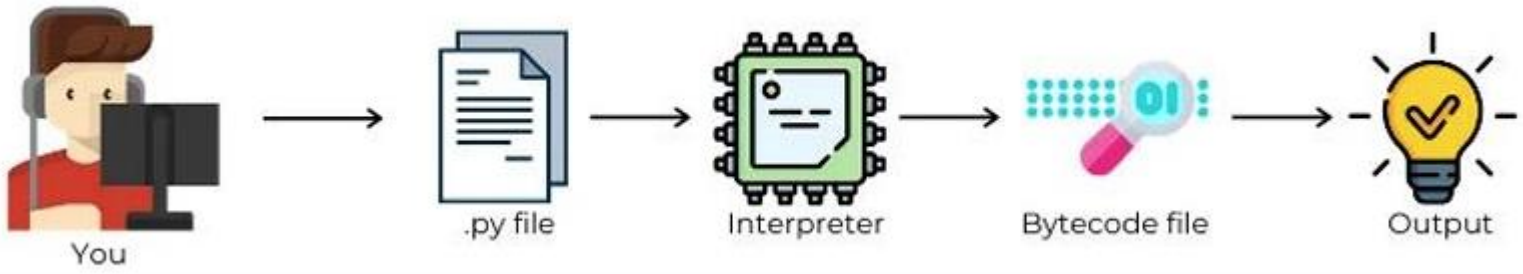
General Purpose Language



Easy To learn

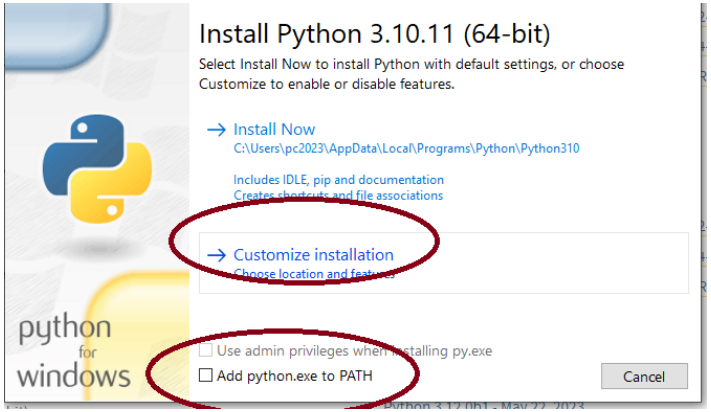
How does python work?

- Python is an interpreted language



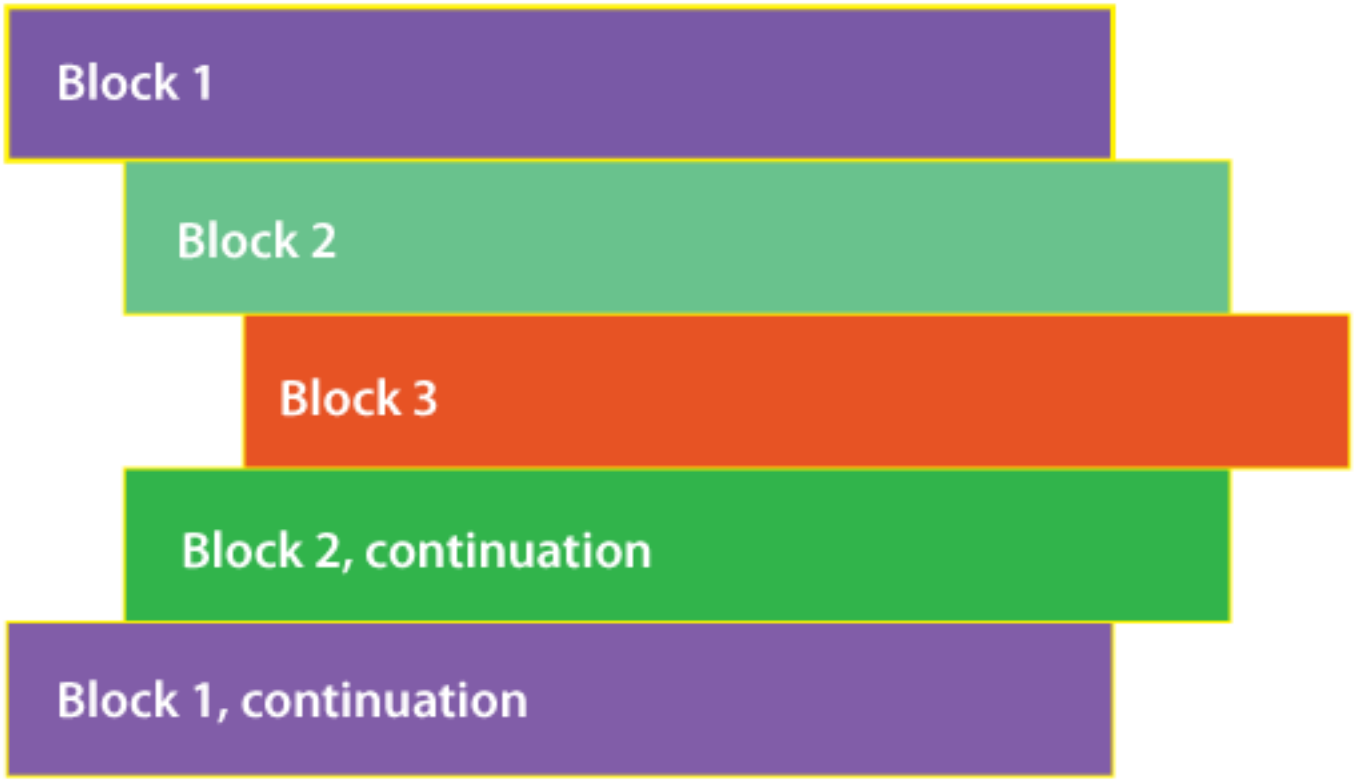
Install & Hello World

- Install python
  - https://www.python.org/downloads/
  - In windows
    - check add python to environment path



- Install pycharm community edition editor
  - https://www.jetbrains.com/pycharm/download/?section=windows

Line Indentations



Line Indentations

```
statement
if condition:
        if condition:
                statement
        else:
                statement
statement
```

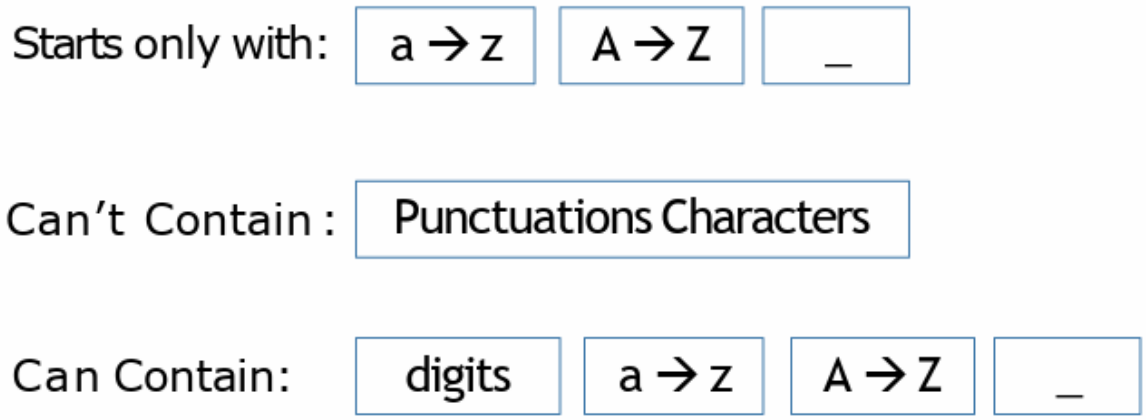
```
code block 1
code block 1
        code block 2
                code block 3
        code block 2
                code block 3
code block 1
```

Quotes …1.. 2 ...3

```
#singlelinecomment
singlestring='hello world'
#or
singlestring="hello world"

paragraph='''hello
world'''
#or
paragraph="""hello
world"""
```

Variables & Data Types

- **Variable is a name that is used to refer to memory location.**
- **Python variable or identifier and used to hold value.**
- **Identifier can be used with variables, functions ,classes and modules**



Python is Case Sensitive Language

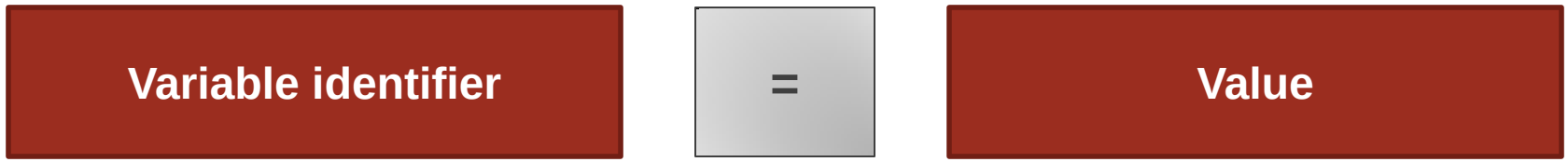
Variables & Data Types

- **Identifier doesn't be one of these word**
- **Reserved Words:**

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Variables & Data Types

- **Python is loosely typed language.**
  - **○ No need to define the variable, the interpreter will do everything.**
- **To define a variable**




```
name = 'jhone'
age = 17
isStudent = True
age = 'seventeen'
```

Variables & Data Types

- **Primitive**
  - **The primitive or basic data structures are the building blocks for data manipulation.**
  - **They contain pure and simple values of data. In Python.**

Numbers **String** **Boolean**

- **Non-Primitive Data Types**
  - **Non-primitive not just store a value, but rather a collection of values in various formats.**

Tuples **Lists** **Dictionaries** **Sets**

Variables & Data Types

- Type Conversion

```
age = 17.5
age=int(age)
print(type(age))
age=float(age)
print(type(age))
age=str(age)
print(type(age))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

Operators (Arithmetic)

+	addition Op	`2 + 3`	`#output: 5`
-	Subtraction Op	`4 - 2`	`#output: 2`
*	Multiplication Op	`4 * 5`	`#output: 20`
/	Division Op	`16 / 5`	`#output: 3.2`
%	Modulus Op	`16 % 5`	`#output: 1`
//	Division without Fractions	`16 // 5`	`#output: 3`
**	Exponent Op	`2 ** 4`	`#output: 16`

Operators (Assignment)

=	assign	`x = 4`	`#output: 4`
+=	add and assign	`x += 3`	`#output: 7`
-=	subtract and assign	`x -= 2`	`#output: 5`
*=	multiply and assign	`x *= 6`	`#output: 30`
/=	divide and assign	`x /= 2`	`#output: 15`
%=	get modulus and assign	`x %= 8`	`#output: 7`
//=	floor divide and assign	`x //= 3`	`#output: 2`
**=	get exponent and assign	`x **= 4`	`#output: 16`

Operators (Comparison)

a **<op>** b

==	return True if a equals b
>=	return True if a equals or greater than b
<=	return True if a equals or lesser than b
!=	return True if a not equals b
<>	return True if a not equals b
>	return True if a greater than b
<	return True if a lesser than b

Operators (Comparison)

When using == Python assumethat:

True = 1, **False** = 0

```
2 == "2"                    #output: False

True == "True"              #output: False

False == 0                  #output: True

True == 1                   #output: True

True == 2                   #output: False
```

Operators (Logic Gate)

`and`	AND Logic Gate
`or`	OR Logic Gate
`not`	Not Logic Gate

```
True and False          #output: False

True or False           #output: True

not False               #output: True

not (True == 2)         #output: True
```

Falsy Values

- **None**, **False**, 0 , Empty collections: **""**, (), [], {}

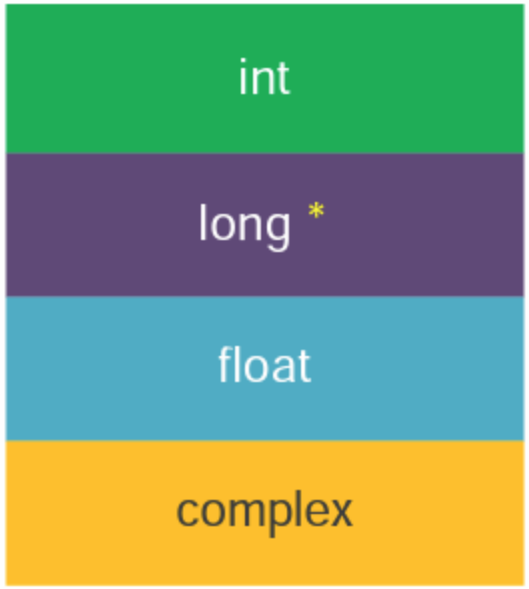
```
2 and 1                                  #output: 1

2 or 1                                   #output: 2

not 4                                    #output: False

not 0                                    #output: True

2 and 0                                  #output: 0

0 and 2                                  #output: 0

“Google” and 1                           #output: 1

“” and “Go”                              #output: “”

False or 0                               #output: 0
```

Numbers

- Long removed from python3

```
import sys
print(sys.maxsize)
```

```
9223372036854775807
```



Numbers(Type Conversion)

int	`int('18')`
float	`float(15)`
complex	`complex(4,5)`

Numbers(functions)

```
w, x, y, z = 4, 4.4, 4.6, 15

round(x)                    #output: 4

round(y)                    #output: 5

min(x,y,z)                  #output: 4.4

max(x,y,z)                  #output: 15
```

Strings

```
name = "Ahmed"
```

---or---

```
name = 'Ali'
```

Strings

```
name = "Ahmed "

print(name) # Ahmed

fullName = "Mohamed " + name * 3 + " Ali";

print(fullName) # Mohamed Ahmed Ahmed Ahmed Ali

nameIntro = ( "I'm " fullName );

print(nameIntro) # I'm Mohamed Ahmed Ahmed Ahmed Ali

print(name[4])    # d

print(name[1:3]) # hm

print(name[:4])   # Ahme
```

Strings

```
name = "information technology institute"

name.capitalize() # Information Technology Institute

len(name) #32

order = "Go read info about his work info in " + name

order.replace("info", "",2)

# Go read about his work in information technology institute

digits,containDigits = "0102002932", "Tel0102002932"

digits.isDigit() # True

containDigits.isDigit() # False
```

Strings

```
statment='    hello world    '
print('=',statment,'=')
print('=',statment.strip(),'=')
print('=',statment.lstrip(),'=')
print('=',statment.rstrip(),'=')
```

- **OUTPUT**

```
=     hello world     =
= hello world =
= hello world     =
=     hello world =
```

Strings (Formatting)

```
intro = "My Name is {0}"
intro.format('Ahmed')
# My Name is Ahmed
intro = "My Name is {1}, I work at {0}"
intro.format('ITI','Ali')
# My Name is Ali, I work at ITI
intro = "My Name is {name}, I work at {place}"
intro.format(name='Ahmed', place='ITI')
# My Name is Ahmed, I work at ITI
```

Tips and Trick

Swap Variables

Traditional Way

```
x = 4
y = 5
temp = x
x = y
y = temp
```

Python Way

```
x,y = 4,5
x,y = y,x
```

Tips and Trick

```
print("I'm", end=" ")
print("Ahmed", end=". ")
print("I", "love", "python")
```

Output:

```
I'm Ahmed. I Love Python
```

Exercises