SCIENTIFIC COMPUTING WITH PYTHON

PYTHON BASIS

Course Coordinator: Dr. R. Mariyal Jebasty
Assistant Professor,
Department of Physics
Wavoo Wajeeha College of Arts & Science
Kayalpatnam.

Course Instructors

1. Mrs. Pushpa, Assistant Professor in Physics, Wavoo Wajeeha Women's College of Arts & Science, Kayalpatnam.
2. Dr. S. Usharani , Assistant Professor in Physics, Wavoo Wajeeha Women's College of Arts & Science, Kayalpatnam.

AGENDA

- Introduction to python
- Characteristics of python
- Features of python
- Installation on python in ubuntu system
- Setting up virtual environment
- Anaconda installation

PYTHON INTRODUCTION

CHARACTERISTICS OF PYTHON

- ✓ Supports functional, structured, and OOPs methods
- ✓ Scripting language
- ✓ Dynamic data types
- ✓ Automatic garbage collection
- ✓ Works on different platforms
- ✓ simple syntax
- ✓ fewer lines of code

FEATURES OF PYTHON

- Free and open source
- Easy-to-learn, read and maintain
- A broad standard library
- Interactive Mode
- Portable
- Databases
- GUI Programming
- Scalable
- Embeddable
- Object-oriented language
- High level language
- Dynamic memory allocation

What can you do with Python?

- ✓Build a website using Python
- ✓Develop a game in Python
- ✓Perform Computer Vision (Facilities like face-detection and color-detection)
- ✓Implement Machine Learning (Give a computer the ability to learn)
- ✓Enable Robotics with Python
- ✓Perform Data Analysis using Python
- ✓Automate a web browser
- ✓Perform Scripting in Python
- ✓Perform Scientific Computing using Python
- ✓Build Artificial Intelligence

Install Python 3 and pip on Ubuntu

- Step 1 Setting Up Python 3
  - sudo apt update
  - sudo apt -y upgrade
  - python3 –V

Output - Python 3.8.10

  - sudo apt-get install python3.x

Step 2 pip installation

  - sudo apt install -y python3-pip
  - pip3 install package_name

Eg: sudo apt-get install python3-pip

sudo pip install pandas or sudo pip3 install pandas

sudo pip install numpy or sudo pip3 install numpy

Set Up a Virtual Environment on Ubuntu

- sudo apt install -y build-essential libssl-dev libffi-dev python3-dev

For virtual environment

- sudo apt install -y python3-venv
- mkdir environments
- cd environments
- python3 -m venv my_env
- ls my_env

Output

- bin include lib lib64 pyvenv.cfg share

Activate a Virtual Environment

- To use this environment, you need to activate it, which you can achieve by typing the following command that calls the activate script:
- source my_env/bin/activate
- (my_env) user@hostname:~/environments$

To deactivate

- (my_env) user@hostname: deactivate

Anaconda installation

- https://www.anaconda.com/products/individual
- sha256sum Anaconda3-2020.11-Linux-x86_64.sh
- bash Anaconda3-2020.11-Linux-x86_64.sh
- source ~/.bashrc
- conda info
- python3
- conda deactivate

Variables & Data Types

Scenario

Salary List

Scenario

Salary List

₹1M

₹1.5M

₹1.8M

₹0.9M

₹1.6M

₹2M

Scenario

Salary List

₹1M

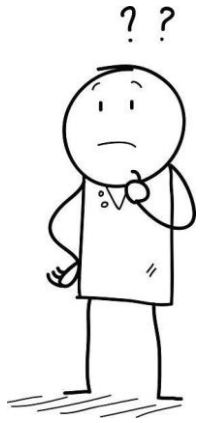
₹1.5M

₹1.8M

₹0.9M

₹1.6M

₹2M



Scenario

Salary List

Ramesh -- 1M

Suresh – 1.5M

Krishna – 0.9M

Rahul – 1.6M

Swati – 1.8M

Sam – 2M

Variables

- Ramesh = 1M
- Suresh = 1.5M
- Krishna = 0.9M
- Rahul = 1.6M
- Swati = 1.8M
- Sam = 2M

What are Variables?

Variables are names bounded to objects.

Variables

Ramesh = 1M

Suresh = 1.5M

Krishna = 0.9M

Rahul = 1.6M

Swati = 1.8M

Sam = 2M

What are Variables?

Variables are names bounded to objects.

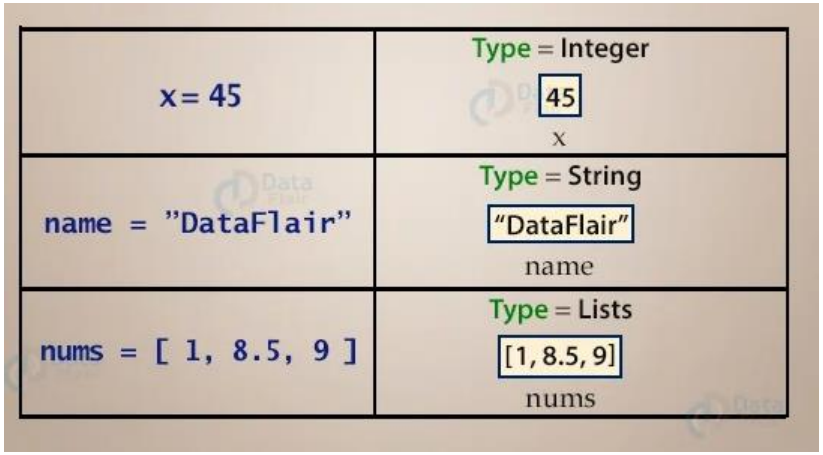
Variables

Variables

Ramesh	= 1M
Suresh	= 1.5M
Krishna	= 0.9M
Rahul	= 1.6M
Swati	= 1.8M
Sam	= 2M

What are Variables?

Variables are names bounded to objects.



Variables in Python

Variable Assignment

Variables in Python

Variable Assignment

- **Variable_name = Value**

```
A = 5
B = 10
```

Variables naming rules in Python

- **Python is case-sensitive**

A=5 is different from a=5

Variables naming rules in Python

- **Python is case-sensitive**

A=5 is different from a=5

- **Variable name cannot start with special character except underscore (_)**

_sam=5 is valid

Variables naming rules in Python

- **Python is case-sensitive**

  A=5 is different from a=5

- **Variable name cannot start with special character except underscore (_)**

  _sam=5 is valid

  @sam=5 is invalid

Variables naming rules in Python

- **Python is case-sensitive**

  A=5 is different from a=5

- **Variable name cannot start with special character except underscore (_)**

  _sam=5 is valid

  @sam=5 is invalid

- **Variable name cannot start with a number**

  9sam =5 is invalid

Variables naming rules in Python

- **Python is case-sensitive**

A=5 is different from a=5

- **Variable name cannot start with special character except underscore (_)**

_sam=5 is valid
@sam=5 is invalid

- **Variable name cannot start with a number**

9sam =5 is invalid sa9m =5 is valid

Reserved keywords cannot be used as variable names.

and	def	False	import	not	True
as	del	finally	in	or	try
assert	elif	for	is	pass	while
break	else	from	lambda	print	with
class	except	global	None	raise	yield
continue	exec	if	nonlocal	return	

How do variables work?

```
a = 10
b = a
a = 6
```

What is the value of a and b?

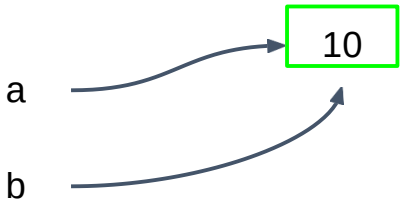
How do variables work?

```
a = 10
b = a
a = 6
```

What is the value of a and b?

How do variables work?

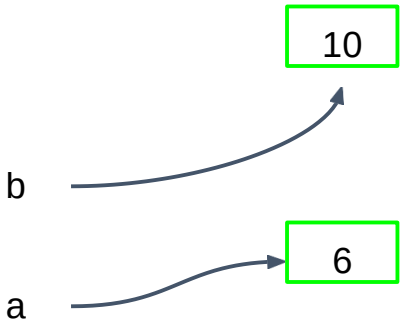
```
a = 10
b = a
a = 6
```



What is the value of a and b?

How do variables work?

```
a = 10
b = a
a = 6
```



What is the value of a and b?

Ans. a is 6 and b is 10

- **NOTE**
  - It's considered best practice (PEP8) that names are lowercase.
  - Variable name should make sense.
  - Avoid using words that have special meaning in Python like "list" and "str"

```
my_dogs = 2
```

```
my_dogs = [ “Sammy” ,  “Frankie” ]
```

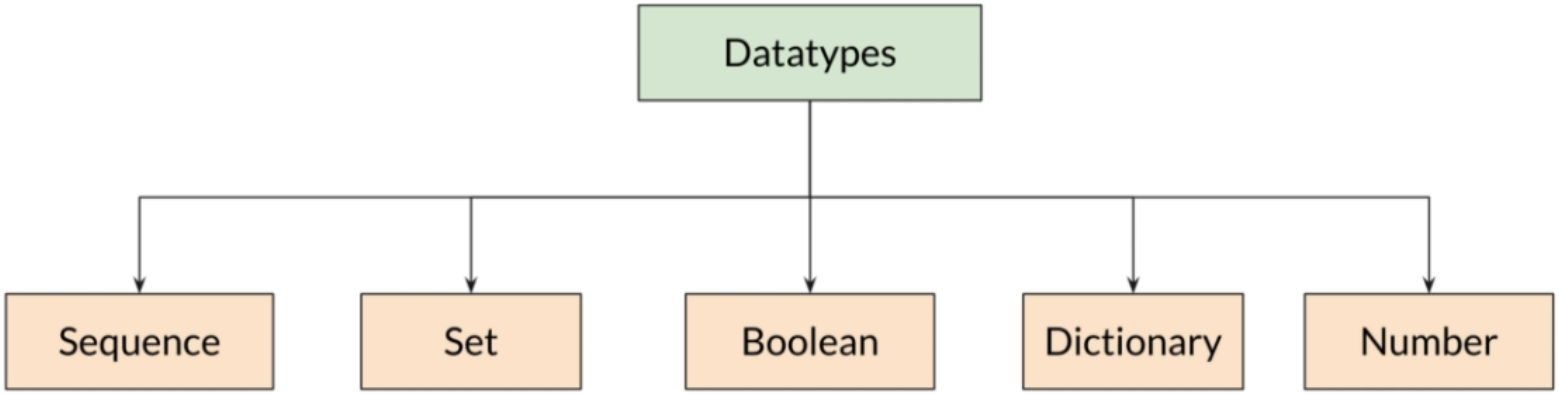
This is okay in Python!

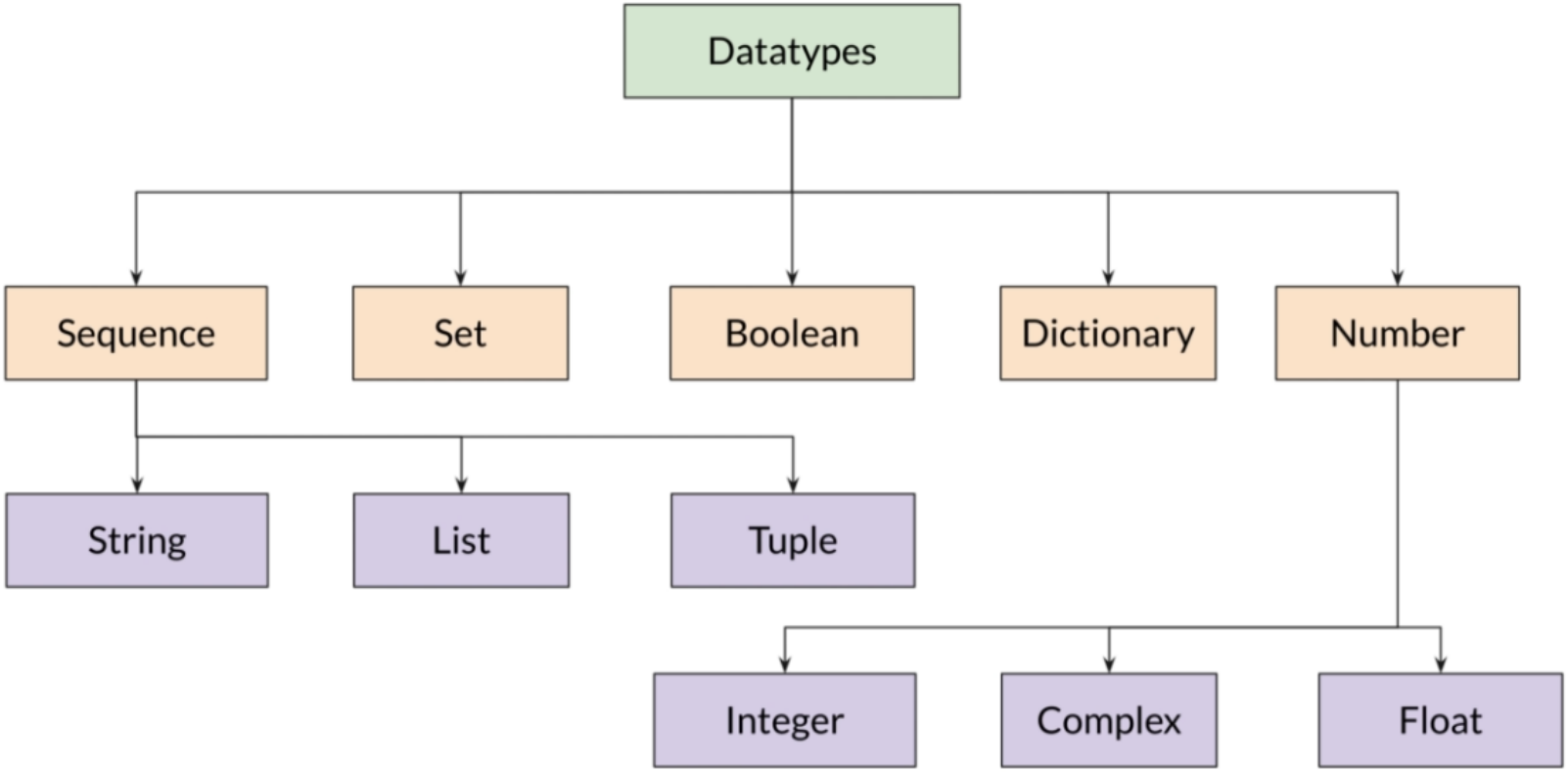
```
my_dogs = 2
```

```
my_dogs = [ “Sammy” ,  “Frankie” ]
```

ERROR in other Languages!

BASIC DATA TYPES


Datatypes
Sequence
Set
Boolean
Dictionary
Number


Datatypes
Sequence
Set
Boolean
Dictionary
Number
String
List
Tuple
Integer
Complex
Float

Name	Type	Description
Integers	int	Whole numbers, such as: **3 300 200**
Floating point	float	Numbers with a decimal point: 2.3 **4.6 100.0**
Strings	str	Ordered sequence of characters: **"hello" 'Sammy' "2000" "楽しい"**
Lists	list	Ordered sequence of objects: **[10,"hello",200.3]**
Dictionaries	dict	Unordered Key:Value pairs: **{"mykey" : "value" , "name" : "Frankie"}**
Tuples	tup	Ordered immutable sequence of objects: **(10,"hello",200.3)**
Sets	set	Unordered collection of unique objects: **{"a","b"}**
Booleans	bool	Logical value indicating **True** or **False**

- **Numbers**: Store numerical information and come in two forms:
  - Integers - Whole Numbers
  - Floating Point - Numbers with a decimal
- **Strings**: Ordered sequence of characters
- **Lists**: Ordered sequence of objects (mutable)
- **Tuples**: Ordered sequence of objects (immutable)
- **Dictionary**: Key-Value pairing that is unordered.
- **Sets** : unordered collections of unique elements

For More Details

- https://data-flair.training/blogs/features-of-python/
- https://data-flair.training/blogs/install-python-windows/
- https://data-flair.training/blogs/python-variables-and-data-types/

Basic datatypes in Python

integer (int)

```
integer_number = 20
print(type(integer_number))
```

```
<class 'int'>
```

floating point number

```
float_number = 20.55
print(type(float_number))
```

```
<class 'float'>
```

complex number

```
complex_value = 20+3j
print(type(complex_value))
```

```
<class 'complex'>
```

sequence data type

list data type

```
fruits = ['orange',"apple", 'mango']
print(type(fruits))
print(fruits)

combined = [1, 'mariya', 23.4, 5+4j]
print(combined)

empty_list = []
empty_list1 = list()
print(empty_list)
print(empty_list1)
```

```
<class 'list'>
['orange', 'apple', 'mango']
[1, 'mariya', 23.4, (5+4j)]
[]
[]
```

```
listinlist = [1,[2,3],3]
len(listinlist)
```

```
3
```

tuple datatype

```
tup1 = (2,3)
print(type(tup1))

(first,second) = tup1
```

```
<class 'tuple'>
```

```
first
```

```
2
```

```
second
```

```
3
```

keywords with first letter caps

True, False, None

Assigning and Reassigning Python Variables

```
Age = 18
print(Age)
print(type(Age))
```

```
18
<class 'int'>
```

```
Age = 'Mariya'
print(Age)
print(type(Age))
```

```
Mariya
<class 'str'>
```

Multiple Assignment

```
Age, Name = 18, 'Mariya'
print(Age,Name)
```

```
18 Mariya
```

Swapping Variables

```
x, y = 'orange', 'Apple'
x,y = y,x
print(x,y)
```

```
Apple orange
```

Deleting Variables

```
Name = 'Mariya'
del Name
print(Name)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-16-a024e7ce78f8> in <module>
      1 Name = 'Mariya'
      2 del Name
----> 3 print(Name)

NameError: name 'Name' is not defined
```

SEARCH STACK OVERFLOW

String Formatter

```
decipline = 'doctor'
x = 10
print(f"He is a {decipline}  with {x} years of experience")
```

```
He is a doctor  with 10 years of experience
```

Indexing

```
listOfNumbers = [1,2,3,4,5,6,7,8,9,10]
len(listOfNumbers)
```

```
10
```

Slicing list

```
list = listOfNumbers[3:7]
print(list)
```

```
[4, 5, 6, 7]
```

```
print the indexed value 0 to 6
```

✓ 0s completed at 12:23 PM