CT60A0203
Introduction to Programming: Python
Week 2A

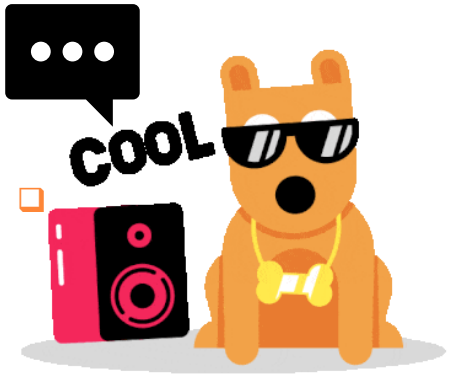
Ashok Kumar Veerasamy, PhD

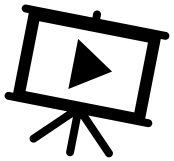
Learning objectives: Variables and Expressions

- To create, assign and use variables to store data in Python programs
- To learn variable naming conventions and in line commenting
- To know how to use arithmetic operators to write expressions
- How Concatenation of string and numbers works in Python
- How to obtain user input in Python: input()
- To know the use of type conversions str(), int() and float()
- To learn basic Python string operations

At the conclusion of this lecutre, students will be able to understand how to write code by using input(), naming conventions, handling arithmetic and scientific calculations in computational thinking way and differences between string and other data types.

- **Why do we need programs? Why should we learn to write programs?**
- First to understand how computer or related devices work
- Programs are written for various reasons
- For example: Game applications, automated machines such as coffee makers, fuel pump, Data analytics, and more….
- So basically, handle data that in the forms of numbers, text, images, pictures, audio and videos…





What programming language is used in your mobile phones ?
(Android / iPhone)

https://www.rovio.com/games/angry-birds-2/

- **So, what is data? And How do we handle data in Python programs)**
- Data is a piece of information stored in the computer for future retrieval
- So, programs are written to get, store, process and report.
- In python data is stored as objects.

- Example data:
- Integer → **0 4 8923423 -232 -45**
  - (The whole number which can be positive or negative)

We have one more:
True **False** (Boolean values)

- Float → **23.45 0.2342214 -1.00 0.23 -0.342342**
  - (The number with decimal values which can be positive or negative)
- String
  - (characters that are enclosed by " ….. “)
  - **"x" "Lahti" “India 2021" "早安中国” "9.12” "O+"**

- In computer all these data are stored as numbers(0 or 1: binary form)

- Attention! → **"23"** is different from **23**
- Similarly, **"23"** is different from **"23 "**

WHY?

But I heard Python supports some more data types

Refer:
https://www.w3schools.com/python/python_datatypes.asp

- **What is variable in programming and how do we define a variable in Python?**
- How do you store your friend/relative's phone numbers in your cell phone?
- Example: **policeNo** = 112 → Here, **policeNo** is a variable that reference to the value/object **112**

To store, compute, analyse and retrieve stored values we use variable. Variable is a reference to a memory address.

Example:

myName = "LUT" [here myName is string type variable]

my_Age = 35 [integer ..]

height123 = 1.67 [float..]

x = my_Age * height123 [expression]

So, the syntax is: <**Variable name**> = <**value**> or <**expression**>

Let's try the following statements in your IDE's shell and discuss with peers

>>> 12x = **1231**

>>> **_x = "Lahti@Finland"**

>>> **a&b = 45.124**

>>> **a_b = 45.124**

>>> **a, b, c = 12, 34.5, "Hello"**

>>> **totalMarks = 90 + 67 + 78 + 90**

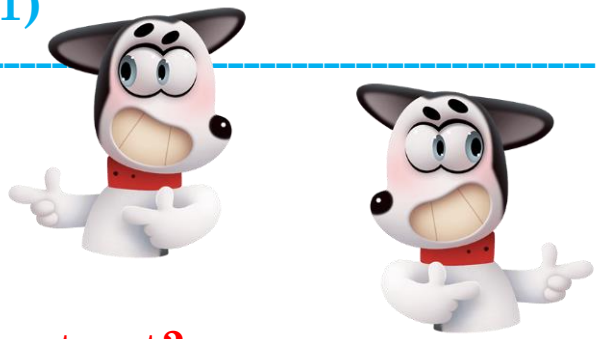
>>> **gradeAvg = totalMarks/4**

>>> **my@Villa = "Ashoka Palace"**

>>> **import = "angry birds"**

Some more…. [Try the following in the IDE]

```
name = "Hello World"
name1 = "Hello Python"
print (name)
print (name1)
```



What is the output?

```
name = "Hello World"
name1 = name
name1 = "Hi do you  know me"
print (name)
print (name1)
```

One more…

```
myname = "Ping Pong"
myName = “xyz"
Myname = " No more Ping Pong "
print( myname, Myname, myName)
```

Are myname and myName same or different?

Arithmetic operators and expressions

Of course, you are aware of these operators already: + - * /
But we have some more in Python to execute some interesting computations

a = 10
b = 3

%	→	print (a%b) → 1	print("Hi"*4)) → HiHiHiHi
**	→	print (2**5) → 32	print (2*5) → 10
//	→	print (a//b) → 3	print (a / b) → ? 3.33

So, the differences between a/b, a%b and a//b are……..

example 1.py

```
1  #use of %, // and /
2  #converting seconds into hours, minutes and seconds
3  sec = int(input("Enter number of seconds that you want to convert:"))
4  hour = (sec//3600)
5  minute = int((sec%3600)/60)
6  sec1 = (sec%3600)%60
7  print (hour,"hour",minute,"minutes",sec1,"seconds")
```

Ok, How to convert minutes in to year, month and days?

Some more arithmetic expressions, string concatenations and type() conversions

```
>>> 4* 2 + 10  → 18
>>> (4*2) + 10  → 18
>>> 4 * (2+10)  → 48
```

Well! do you want rounded brackets or ?

example 2.py

```
1  #string concatenation
2  firstName = "Ashok"
3  surName = "Kumar"
4  fullName = firstName + surName
5  print (fullName) # concatenation
6
```

```
a = "10"
b = "20"
print (a+b)
```

```
x = 5
y = "30"
#print(x+y)
print(x+int(y))
```

```
w1 = 12.55
w2 = "15.50"
tot_w = w1+float(w2)
print (tot_w)
```

```
x = 5
y = "30"
print(x+y)
```

Hmm! How to convert numbers into string?

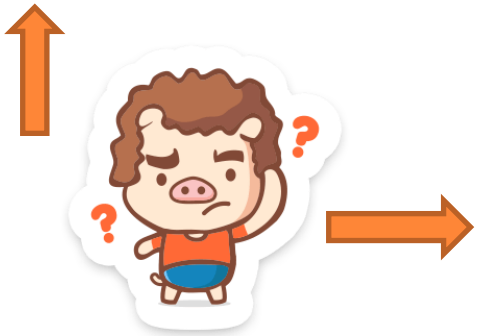
User input() in Python

Let's try the program given here and see what happens?

```
marks = input ("Enter your marks:")
bonusPoints = 10
tot_marks = marks + bonusPoints
```

Response to the input( ) is always a string. So how to fix it?

```
marks = input ("Enter your marks:")
bonusPoints = 10
tot_marks = int(marks) + bonusPoints
print(tot_marks)
```



```
marks = int(input ("Enter your marks:"))
bonusPoints = 10
tot_marks = marks + bonusPoints
print(tot_marks)
```

Which one is better?

Hold on! how to get float number as input?

comment line → # in Python

- To provide description about coding for readers to understand the program. That is coding standard clairty
  In Python, comment line is denoted (precedes with) by #

- How to include comment line in Python ?
  Check Weekly assignment Exercise 3 example program
  Did you do the practice exercises for week 1? If so, you know what comment line means.

```
#This program computes celcius to Fahrenheit
""" This program is written by me to compute
 celcius to Fahrenheit"""
```

Caution: Don't use comment lines too much in your program – which is not a good programming style.

Some exercises before move to next session of this week

1. **What is the output of the following code?**

```
1  name="XiaoboBi"
2  print(name+666)
```

a. XiaboBi666
b. name+666
c. TypeError: can only concatenate str (not “int”) to str
d. XiaboBi+666

2. **Which of the following are valid Python variable names or expressions?**
   a. xyz-23 b. 1stName c. x+y = 20 d. str = str(20) + "30" e. attended = True
   f. weight = int(56.89)

3. Try these to know what type() function is for…..

```
x = 10
y = 3.434
z = "LUT"
print (type(x)
print (type(y)
print (type(z))
```