



PRODDEC PYTHON DAY 2

Using python shell in command prompt.

```
In [ ]: # python -> windows  
# python3 -> Linux and Mac  
# OR  
# python2 -> Linux and Mac
```

Run python file in command prompt (windows)

```
In [ ]: # python your_python_file_name.py
```

For using linux online

```
In [ ]: # Go to https://bellard.org/jslinux/  
# select Alpine Linux 3.12.0 -> click here
```

Running python on Android

```
In [ ]: # Download and install Pydroid 3 from playstore.
```

For using python without installation

```
In [ ]: # Use https://www.onlinegdb.com  
# Use Desktop mode if using in android.  
# Select the Language as python3.  
# Use run button to run the program.  
# Use the share option to share the code.
```

Using the comments in the program

```
In [ ]: # Comments are used for readability of the code.  
# Two type of comments are used in python.  
# Single line comment  
# Multiline comment
```

Single line comment

```
In [1]: # This is my code.  
# If you change this code, You will destroy your computer.  
  
print("My name is peter")  
print("My age is 50") # This is not my actual age  
  
name=22121 # Please use a decent name for your variable like num1  
  
My name is peter  
My age is 50
```

Multiline comment

```
In [ ]: """  
Author: Peter davidson  
Date: 12/21/2050  
If you have any doubt, clear by yourself.  
Dont change the variables name used below.  
Below are the examples of the bool datatype.  
Please check this link :https://www.python.org  
"""  
doYouUnderStand=True  
isCoffeeAvailable=False
```

Use of comments

```
In [3]: # For comment out a python code  
# print("Hello World")  
  
# Adding small description in code  
print(1,12,23,1212) # example of multiple argument in print()  
  
# For understanding code by others and by ourself.  
# You can run some portion of the code with help of comment.  
# For adding large details -> use multiline comment.
```

1 12 23 1212

Shebang line

```
In [ ]: #!/usr/bin/python3  
  
# If you run this file (your python file) in Linux/mac without using the python.  
# It will be run by the python3 interpreter in this path.
```

Variables

```
In [ ]: """  
Variables are used to store information to be referenced and  
manipulated in a computer program.  
They also provide a way of labeling data with a descriptive name,  
so our programs can be understood more clearly by the reader and ourselves.  
It is helpful to think of variables as containers that hold information.  
Their sole purpose is to label and store data in memory.  
This data can then be used throughout your program.  
"""
```

Operations using variables

```
In [13]: num1=12151  
num2=51521521  
  
# Print the type of data in the variables.  
print(type(num1))  
  
# Arithmetic operation  
print(num1 + num2)  
print(num2 / num1)  
  
print("@*10") # Multiplication using the string  
  
my_symbol="$"  
print(my_symbol*10) # Multiplication using the string
```

```
<class 'int'>  
51533672  
4240.105423421941  
$$$$$$$$$  
$$$$$$$$$
```

Type Error

```
In [15]: name="peter"  
num=21151  
print(name+str(num)) # This Line works  
print(name+num) # This Line cause error  
  
peter21151  
  
-----  
TypeError: must be str, not int  
Traceback (most recent call last)  
<ipython-input-15-d9ac81d144b7> in <module>  
      2 num=21151  
      3 print(name+str(num))  
----> 4 print(name+num)  
  
TypeError: must be str, not int
```

String Concatenation (Joining String)

```
In [17]: first_name="Peter"  
last_name="David"  
  
# String concatenation  
full_name=first_name + " " +last_name  
  
print(full_name)
```

Peter David

Convention for naming a variable

```
In [1]: # Name should be related to the type and value in the variable.  
my_college = "CEC"  
  
# Dont use space in the variable name.  
# my name = "Peter" -> cause an error.  
  
# Two words in the variable should be separated by _  
my_age=52  
  
# Use camel casing  
myLaptop="Dell Laptop"  
  
# Use pascal casing  
MeraWatch="Sonata" # ALL starting letter, capital  
  
# Variable name should not start with number  
# 2name="Peter and David" -> cause error  
  
# Variables name can start with _  
_name="Hari"  
  
# Avoid using "function in python" for variable name  
min="Dont use it, it is terrible!!!"
```

Declaring variables

```
In [23]: num1=12145  
num2=5115  
num3=5151  
  
print(num1,num2,num3)  
  
# Declaring variables in single Line using ;  
phone1="0477"; phone2="0481"; phone3="0488"  
print(phone1,phone2,phone3)
```

```
# Multiple declaration in single Line  
food1,food2,food3="idli","dosa","chutney"  
print(food1,food2,food3)
```

```
# Adding same value to all variables  
student1=student2=student3="Btech"  
print(student1,student2,student3)
```

12145 5115 5151

0477 0481 0488

idli dosa chutney

Btech Btech Btech

Getting user input

```
In [2]: print("Enter your name: ")  
name=input()  
print("Hello",name," How may i help you?")
```

Enter your name:
Peter

Hello Peter , How may i help you?

```
In [7]: age=input("Enter your age: ")  
print("Hey old!","Your age is",age)
```

Enter your age: 56

Hey old! Your age is 56

Using int() function in input()

```
In [9]: # Program to print the date of year  
age=int(input("Enter your age: "))  
name=input("Enter your name\n") # For cursor to next Line  
DateOfYear=2020-age  
print("Hey",name,"!","Your date of year is",DateOfYear)
```

Enter your age: 21

Enter your name

Peter

Hey Peter ! Your date of year is 1999

Datatypes

```
In [29]: # Data that can be stored in a variable is of different types.  
# These are called datatype.  
# eg: str,int,bool  
  
# Printing the data type of True  
value=True  
print(type(value))
```

<class 'bool'>

Comparison operator

```
In [10]: # Output of a comparison operator will be always bool datatype (True or False)  
print(12>32213)  
print(2 num=21151)  
print(len("peter") >= len("Haris"))  
print(165<=4324)  
print("Peter"=="peter")  
print("teacher" != "student")
```

False

False

True

False

True

```
-----  
TypeError: must be str, not int  
Traceback (most recent call last)  
<ipython-input-15-d9ac81d144b7> in <module>  
      2 num=21151  
      3 print(len("peter"))  
----> 4 print(len("Haris"))  
  
TypeError: must be str, not int
```

TypeError: must be str, not int

nested if statement

```
In [13]: # if inside if -> nested if  
name="hari"  
age=12  
if name=="hari":  
    print("you are hari and your age is 12")
```

you are hari and your age is 12

Good to see you

```
In [17]: age=int(input("Enter your age: "))  
if age>60:  
    print("Hey you are older")
```

else:
 print("You are a baby!")

Enter your age: 12

You are a baby!

if else elif statement

```
In [16]: age=int(input("Enter your age: "))  
if age>60:  
    print("Hey you are older")  
elif 18<age<60:  
    print("You are not old")
```

else:
 print("You are a baby!")

Enter your age: 30

You are not old

Enter your age: 12
You are a baby!

nested if statement

```
In [30]: # if inside if -> nested if  
name="hari"  
age=12  
if name=="hari":  
    if age==12:  
        print("you are hari and your age is 12")
```

you are hari and your age is 12

Good to see you

you are hari and your age is 12

Good to see you

Online Visualization tool for python

```
In [2]: # visit this website -> http://pythontutor.com/visualize.html
```

```
In [3]: 
```