Module 2

Lab 2 : Python Basic Programming for IoT

Assc. Prof. Ts. Dr. Ahmad Shukri Mohd Noor

WIJDAN SKILLS ENTERPRISE

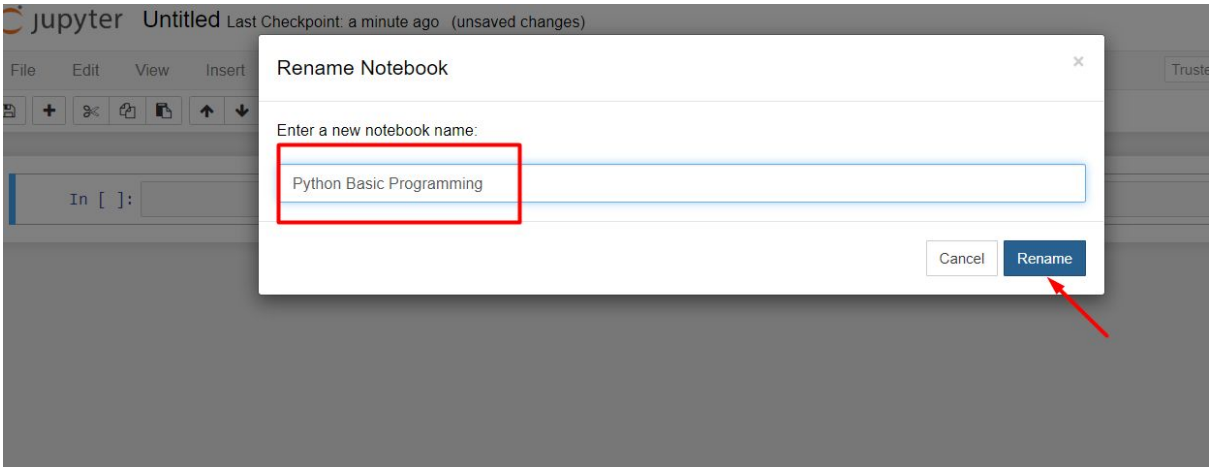
LAB 2: PYTHON BASIC PROGRAMMING

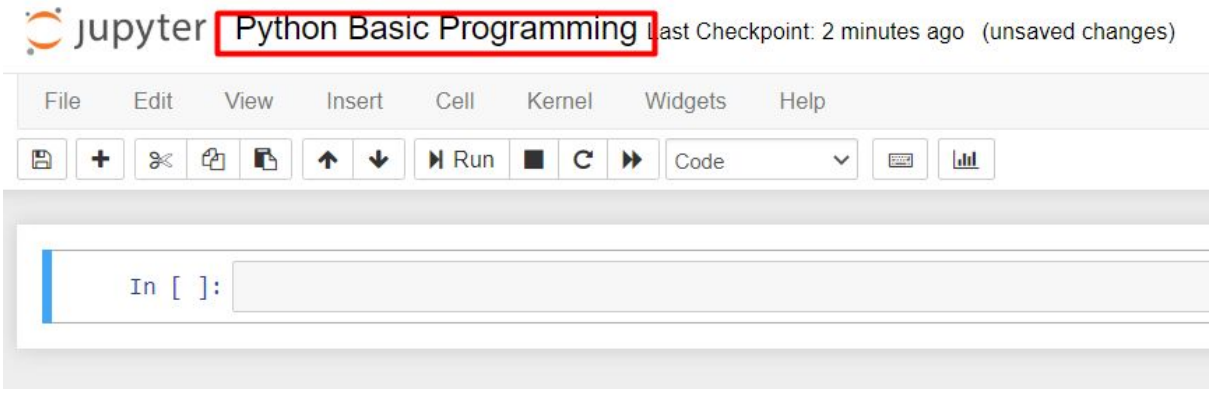
Objective:
In this lab we are going to code using Jupyter Notebook. Throughout this lab, we will cover python syntax, element, comment, variable, data types and basic operators.

Steps:

Install WinPython

1. Start Jupyter Notebook.
2. Create a new python 3 file.
3. Change the title to “Python Basic Programming” and start code.





4. Write our first code to familiarise with the python syntax and jupyter notebook. Press “Enter” to see the result.

```
In [1]: print("Hello world")
        Hello world
```

5. The code below shows the importance of indentation in python.

```
In [2]: if 5>3:
        print("wrong indentation")
```

```
  File "<ipython-input-2-ed14205c5890>", line 2
    print("wrong indentation")
        ^
IndentationError: expected an indented block
```

```
In [3]: if 5>3:
            print("right indentation")
```

```
right indentation
```

6. The code below using # for comment

```
In [9]: # this is comment
        print("comment test")
```

```
comment test
```

```
In [8]: # this is comment
        # python do not have multiline comment
        print("comment test")
```

```
comment test
```

7. The code below creates a variable in python.

```
In [11]: a = 5
         b = 6.0098
         _car = 'BMW' # can use '' or " "

         print(_car)
         print(b)
         print(a)
```

```
BMW
6.0098
5
```

In [13]:
```
#value assigned to multiple variables

car1,car2,car3 = "BMW",'Mercedes',"Volkswagen"
print(car1)
print(car2)
print(car3)
```

```
BMW
Mercedes
Volkswagen
```

In [16]:
```
#same value assigned to multiple variables

car1=car2=car3 = "Viva"
print(car1)
print(car2)
print(car3)
```

```
Viva
Viva
Viva
```

In [18]:
```
nama = 'Dan'
print('My name is ' + nama)
```

```
My name is Dan
```

In [19]:
```
nama = 'Dan'
ayat = 'My name is '
print( ayat + nama)
```

```
My name is Dan
```

In [20]:
```
number1 = 20
number2 = 2020
print(number1 + number2)
```

```
2040
```

8. Variables created outside of a function are called global variables. We may use the Global Keyword to create a global variable within a function.

In [22]:
```
location = 'Sungai Petani'

def function1():
    print(location)

function1()
```

```
Sungai Petani
```

In [23]:
```
#global variable
location = 'Sungai Petani'

def function1():
    #local variable
    location = 'Jitra'
    print(location)

function1()
print(location)
```

```
Jitra
Sungai Petani
```

In [25]:
```
#global variable
location = 'Sungai Petani'

def function1():
    #global variable created
    #inside function
    global location

    location = 'Jitra'
    print(location)

function1()
print(location)
```

```
Jitra
Jitra
```

9. The code below is to assign variables to a particular data type.

```
In [39]: e = ["wij","dan","mohamad"]
         print(e)
         print(type(e))

['wij', 'dan', 'mohamad']
<class 'list'>
```

```
In [40]: f = ("wij","dan","mohamad")
         print(f)
         print(type(f))

('wij', 'dan', 'mohamad')
<class 'tuple'>
```

```
In [41]: e = ["wij","dan","mohamad"]
         print(e)

         e[2] = "ariff"
         print(e)

['wij', 'dan', 'mohamad']
['wij', 'dan', 'ariff']
```

```
In [42]: f = ("wij","dan","mohamad")
         print(f)
         f[2] = ariff
         print(f)

('wij', 'dan', 'mohamad')
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-42-07d2fee7e8f1> in <module>
      1 f = ("wij","dan","mohamad")
      2 print(f)
----> 3 f[2] = ariff
      4 print(f)

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

In [45]:
```
e = ["wij","dan","mohamad"]
f = ("wij","dan","mohamad")

print(e.__sizeof__())
print(f.__sizeof__())
```

```
64
48
```

In [51]:
```
h = {'name': 'wijdan', 'age': 20}

print("his name is",h['name'])

print("his age is",h['age'])
```

```
his name is wijdan
his age is 20
```

10. The code below is to use Input and Output in python.

```
In [ ]: print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)
```

value to be printed

separator used between values

printed after all values are printed. default is new line

sys.stdout is screen as default

```
In [1]: print(1,2,3,4)
        print(1,2,3,4, sep='#', end='.')
```

```
1 2 3 4
1#2#3#4.
```

```
In [2]: x = 10
        y = 2020

        print("I am {} years old in {}".format(x,y))
```

```
I am 10 years old in 2020
```

```
In [3]: print("i love {0} and {1}".format("roti canai","teh tarik"))
        print("i love {1} and {0}".format("roti canai","teh tarik"))
```

```
i love roti canai and teh tarik
i love teh tarik and roti canai
```

```
In [5]: z = input('Enter a number :')
        z
```

```
Enter a number :200
```

```
Out[5]: '200'
```

```
In [20]: name,age = input('enter your name:'),int(input('enter your age:'))
```

```
enter your name:wijdan
enter your age:20
```

11. The code below to introduce basic operators in python.
    a. Arithmetic

```
In [8]: x = 2
        y = 4
        print("x + y = ", x+y)
        print("x - y = ", x-y)
        print("x * y = ", x*y)
        print("x / y = ", x/y)
        print("x // y = ", x//y)
        print("x ** y = ", x**y)
```

```
x + y =  6
x - y =  -2
x * y =  8
x / y =  0.5
x // y =  0
x ** y =  16
```

    b. Comparison

```
In [10]: x = 2
         y = 4
         print("x > y = ", x>y)
         print("x < y = ", x<y)
         print("x == y = ", x==y)
         print("x != y = ", x!=y)
         print("x >= y = ", x>=y)
         print("x <= y = ", x<=y)
```

```
x > y =  False
x < y =  True
x == y =  False
x != y =  True
x >= y =  False
x <= y =  True
```

c. Logical

```
In [11]: x = True
         y = False
         print("x and y = ", x>y)
         print("x or y = ", x<y)
         print("x not y = ", x==y)
```

```
x and y =  True
x or y =  False
x not y =  False
```

d. Bitwise

```
In [13]: x = 8
         y = 4

         print(x&y) #and
         print(x|y) #or
         print(~x) #not
         print(x^y) #exclusive or
         print(x>>2) #bitwise right shift
         print(x<<2) #bitwise left shift
```

```
0
12
-9
12
2
32
```

12. To allow flexibility we might want to take the input from the user. In Python, we have the input() function to allow this. It is save in string data type. Use a cast to take numeric data.

```
In [5]: z = input('Enter a number :')
        z
        Enter a number :200
Out[5]: '200'
```

```
In [20]: name,age = input('enter your name:'),int(input('enter your age:'))
         enter your name:wijdan
         enter your age:20
```

LAB 2: PYTHON BASIC PROGRAMMING

References:

1. https://github.com/winpython
2. https://jupyter.org/
3. https://www.w3schools.com/python/default.asp