

Python 101

An introductory Python workshop hosted by the Robotics and Automation Society

Goals:

Throughout this workshop you will learn the fundamentals of Python, along with the basics of using a Terminal for file navigation, and gain experience using a Raspberry Pi.

Note that the entirety of Python cannot be learned in a single day. The purpose of this workshop is to create a foundation that will make using and learning more about Python easier.

Terminal Basics:

```
$ ls // lists everything in your current Directory(folder)
$ cd Desktop // changes current Directory to Destination
$ cd .. // move back one Directory
$ cd Desktop
$ mkdir python_workshop // makes a directory with the given name
$ cd python_workshop
$ python
>>>
```

Python FAQ:

Q: What is Python?

A: Python is a high-level programming language that emphasizes simplicity and code readability.

Q: What fields and industries use Python?

A: Web Development, Machine Learning, Data Analysis, Scientific Computing, and Robotics are all areas that use Python.

Q: How Difficult is Python to learn?

A: Python is often regarded as one of the easiest programming languages to learn, due to its easy to remember syntax, readability, and focus on simplicity.

Variables:

```
>>>a=20.25
>>>a2=20
>>>first_user_input
>>>summation
>>>result
>>>number_of_counts
```

1. Rules to naming
 - a. Start with lower case letter
 - b. Include an underscore if there are multiple words
 - c. Be as specific as possible for long and complicated programs

Printing:

```
>>>a=20.25
>>>print(a)
20.25
```

1. Good way to follow value changes in the program
2. Makes debugging easier

Arithmetic

```
>>>a=20 + 20
>>>b=20-20
>>>c=20*20
>>>d=20/2
>>>e=2**2
>>>f=20%3
>>>g=(20+2)/2 * 3
>>>print(a, | b, | c, | d, | e, | f, | g )
40 | 0 | 400 | 10 | 4 | 2 | 14
```

1. Parentheses matter!
2. Still left to right, and parentheses take precedence

Reading user input

```
>>> x= input()
>>>25
>>>print(x)
25
>>>y=input("Enter number:")
Enter number: 30
>>>print(y)
30
```

If Statements (conditionals)

```
>>> x=100
>>>if x>99:
    Print("true")
true
>>>y=
Enter number: 30
>>>print(y)
30
>>>z=200
>>>if z<199:
    Print(z)
>>>print(x):
100
```

- When the condition is not met, the next line of code is executed, and the program continues running
- Don't forget the colon at the end of an if condition!

If-Else Statements

```
>>> x=100
>>> if x>=99:
    Print("true")
>>> else:
    Print("False")
true
>>> y=100
>>> if y<=99:
    print(y-100)
```

- When the first condition is not met, then the else statement is executed

Elif Statements

```
>>> x=105
>>> if x<=100:
    print("x less than or equal to 100 ")
>>> elif x>=100:
    print("x greater than or equal to 100")
>>> else:
    print("WTF")
x greater than or equal to 100
>>> y=100
>>> if y>=101:
    print(">=101")
>>> elif y<=99:
    print("<=99")
>>> else:
    print("WTF")
WTF
```

Practice Exercise: Calculator

Let's use what we've learned so far to build a calculator! Take a few minutes to use what you've learned and create a simple calculator script.

Hints:

- When writing programs, it helps to break the problem down into tasks that can be solved using a few lines of code.
- If you want to save your code, you can create a .py file and write your code in it using a text editor.
 1. Use the following command to create the file:
`$ touch [your_file_name].py`
 2. Open the file by double-clicking it in the File Explorer
 3. Run the code you saved by using the command:
`$ python [your_file_name].py`