



Python Workshop



InnoGators

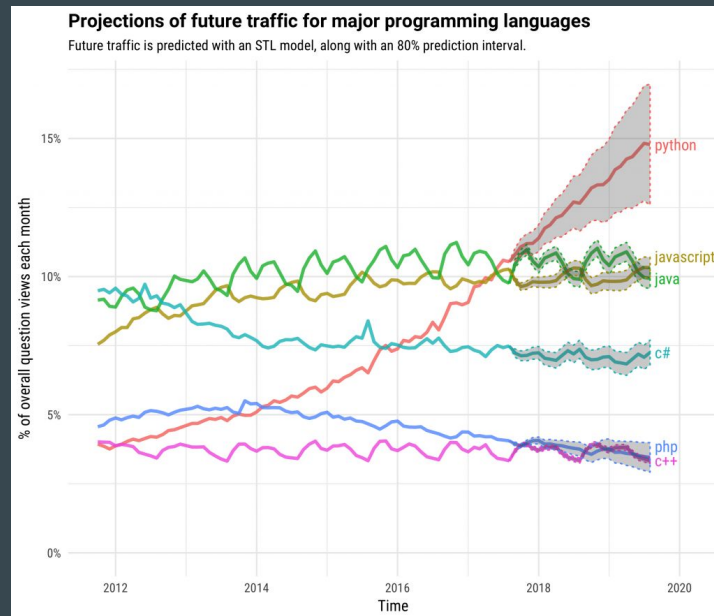
- Design Organization
 - Help members cultivate their technical skills through design experience and collaboration.
 - Working with professors and companies to make their ideas come true!
- Design Projects:
 - 3D Smart Filament Recycler
 - Radiation Plume Tracking Drone
- www.innogators.weebly.com
 - Go to the New Members tab if you're interested!

S.I.F.T.

- Consulting Organization
 - Work with real businesses
 - Case Competitions
- Product Management, coding, business principles
- We will be recruiting this coming spring
 - Mentors
 - Project Managers
 - Analysts
- Questions?
 - Email networking.sift@gmail.com

What is Python used for?

- Scripting and Automating Tasks
- Data Science
- Machine Learning and Artificial Intelligence
- Web and Game Development



Let's get into the basics!

Open up this website on your browser:

<https://repl.it/languages/python3>

Data Types

- String: 'camel' → str
- Numeric
 - Integer: 6 → int
 - Float: 3.2 → float
- List (sequenced): ("a","b") <> ("b","a") → list
- Set (unordered): {1,3,2} = {1,2,3} → set
- Tuple (immutable/unchangeable): (1, 2, 3) → tuple
- Range (range of numbers): Range(5) is Integers from 0-4 → range
- Boolean: True or False → bool

There are others, but these are the most important for today.

Math/Operations

- Python also implements the basic math operations: Add (+), Subtract (-), Multiply (*), Divide (/), Integer Division (//), as well as Modulus/Remainder (%) and Exponent (**)
- Add $\rightarrow x + y$
- Subtract $\rightarrow x - y$
- Multiply $\rightarrow x * y$
- Divide $\rightarrow x / y$
- Integer Division $\rightarrow x // y$
- Modulus/Remainder $\rightarrow x \% y$
- Exponent $\rightarrow x ** y = x^y$

Boolean Operators

- In addition to Math Operations, Python also implements Boolean/Logical Operations: And, Or, Not, Equal, Not Equal, Less Than, Greater Than
- And \rightarrow x and y (x, y must be booleans)
 - True and True is True
 - True and False is False
- Or \rightarrow x or y (x, y must be booleans)
 - True or False is True
 - False or False is False
- Not \rightarrow not x (x must be boolean)
 - not True becomes False
- Equality \rightarrow $x == y$
 - $2 == 3$ is False
 - $2 == 2$ is True
- Not Equal \rightarrow $x != y$
 - $2 != 3$ is True
 - $2 != 2$ is False
- Less Than / Less than or equal to \rightarrow $x < y$ / $x <= y$
 - $2 < 3$ is True
- Greater Than / Greater than or equal to \rightarrow $x > y$ / $x >= y$
 - $2 > 5$ is False
 - $2 >= 2$ is True

Variables and Printing

- Variables are containers for storing data values
- Can set your variable equal to a basic variable value such as 0 or 'guru99' as listed below
- Print command will output given value to the screen

```
1 # Declare a variable and initialize it
2 f = 0
3 print(f)
4 # re-declaring the variable works
5 f = 'guru99'
6 print(f)
7
8
9
10
11
12
```

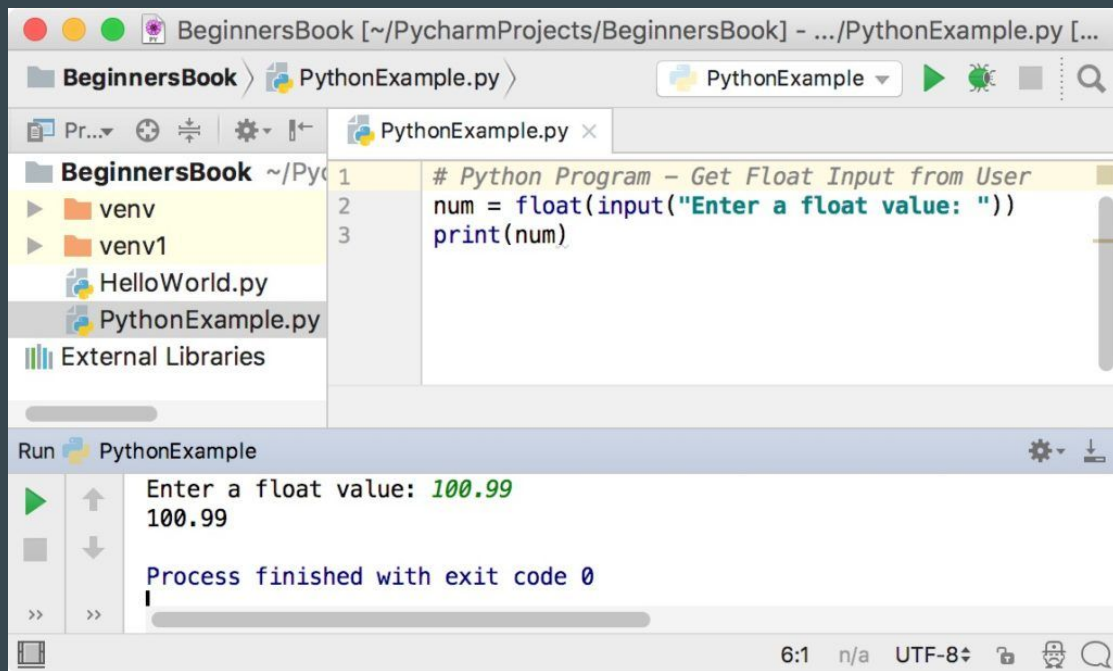
you can re-declare the variables, even-after if it is declared once. it works fine

```
Run Python5.1
0
guru99
```

A screenshot of a Python IDE window. The editor shows a Python script with two lines of variable assignment and two print statements. The first line is a comment, the second is 'f = 0', the third is 'print(f)', the fourth is another comment, the fifth is 'f = 'guru99'', and the sixth is 'print(f)'. The values '0' and 'guru99' are highlighted with red boxes. A red arrow points from the second 'print(f)' to a callout box containing the text 'you can re-declare the variables, even-after if it is declared once. it works fine'. Below the editor, the Run console shows the output '0' and 'guru99', with 'guru99' highlighted by a red box.

User Input

- Use casting if you want it to be a specific data type



The screenshot shows a Python IDE window titled "BeginnersBook [~/PycharmProjects/BeginnersBook] - .../PythonExample.py [...". The main editor displays the following code:

```
1 # Python Program - Get Float Input from User
2 num = float(input("Enter a float value: "))
3 print(num)
```

The left sidebar shows the project structure with folders "venv" and "venv1", and files "HelloWorld.py" and "PythonExample.py".

The bottom panel shows the output of the program:

```
Run PythonExample
Enter a float value: 100.99
100.99
Process finished with exit code 0
```

The status bar at the bottom indicates the time is 6:1, the encoding is n/a, and the character set is UTF-8.

If Statements

- Code that executes only if Logical condition is evaluated to True
- Conditional statements must use the == when equating two values

```
Test.py ×  
1 num = 10  
2 if (num == 0):  
3     print("Number is Zero")  
4  
5 elif (num > 5):  
6     print("Number is greater than 5")  
7  
8 else:  
9     print("Number is smaller than 5")  
10 |
```

For-while Loops

- Loops allow execution of same code multiple times without repeating
- Notice how indenting is **critical** to the language
- Remember not to make an indefinite loop when using while

```
In [7]: my_list = [1, 5, 12, 91, 102]
my_list_length = len(my_list)
for i in range(0,my_list_length):
    print(i, my_list[i] * my_list[i])

0 1
1 25
2 144
3 8281
4 10404
```

```
while-loop.py x
1
2     def print_msg(count, msg):
3         while count > 0:
4             print(msg)
5             count -= 1
6
7
8     print_msg(3, "Hello World")
9

Run: while-loop x
/Users/pankaj/Documents/PycharmProje
Hello World
Hello World
Hello World

Process finished with exit code 0
```

Practice: Calculator

- Write a program to mimic a calculator
- Ask the user which operation to perform (addition, subtraction, multiplication, division, and exponentiation)
 - Take two inputs
 - Specify type of operation
 - Perform the operation
 - Present the result
 - Start again

Resources

If you would like to continue your journey with Python, check out these websites for practice problems, syntax help, and more.

- <https://coderbyte.com/>
- <https://www.learnpython.org/>
- https://www.youtube.com/watch?v=_uQrJ0TkZlc&vl=en
- <https://www.guru99.com/python-tutorials.html>
- <https://www.codecademy.com/catalog/language/python>
- <https://automatetheboringstuff.com/>

Poll: Thoughts on an intermediate Python workshop?

Topics to be covered: Functions and Classes