



Python Workshop

Go to: https://www.onlinegdb.com/online_python_compiler

Variables

- No actual variable types
- Other than lists or dictionaries, any variable can be assigned anything
- Integers, Floats, Strings, Characters, etc.

ex:

```
v1 = "test"  
v1 = 24  
v1 = False
```

Parsing

- If a number is first made as an integer rather than a float, you would need to parse it to that data type
- Needed when expecting decimal points when dividing integers

```
x1 = 4  
x = float(x1)  
  
print(x)
```

Output: 4.0

Arithmetic

- PEMDAS
- If something isn't surrounded by parentheses, it will run in sequential order

```
x1 = 4**2*3+4  
x2 = 4** (2*3+4)  
  
print("%f\n %f" % (x1, x2))
```

Output:

52.000000

1048576.000000



Getting Inputs

- Allows you to get user input and have more dynamic code
- Can take in any variable type
- May need to be parsed to that data type

```
a = input("Enter a value: ")  
b = input("Enter b value: ")  
c = input("Enter c value: ")
```

Formatted Prints

- Allows you to have dynamic print statements
- Using “\n” is the same as pressing enter to go to a new line

ex:

```
x1 = 1.23456789
x2 = 7568.576456

print("The value of x is %.4f and %.4f\n" % (x1, x2))
```

output: The value of x is 1.2345 and 7568.5764



Numpy Library

- Python doesn't support most mathematical functions
- A library is needed to do this (unless you want to do it yourself)
- The online compiler doesn't require any download
- Needs to be imported as shown below

```
import numpy as np  
  
d = np.sqrt(4)  
print(d)
```

Output: 2.0



If Statements

```
if(boolean_expression):  
    statement(s)  
elif(boolean_expression):  
    statement(s)  
else:  
    statement(s)
```

Ex:

```
if (4 != 3):  
    print("good")
```



While Loops

```
while(condition):
    statement(s)
```

ex:

```
while (4 != 5):  
    print("this is printed an infinite number of times")
```

[illegible]

Methods

- Contain a set algorithm or function
- Can be called at any point by the program
- Can return a value or run something without returning a value

Output:

Hello
Hello
Hello

```
def main():  
    print("Hello")  
  
main()  
main()  
main()
```

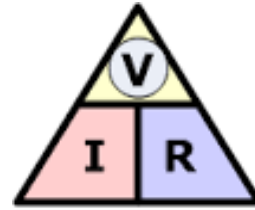


Workshop (pt. 1)

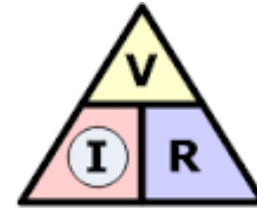
Ohm's Law:

Create a method that can

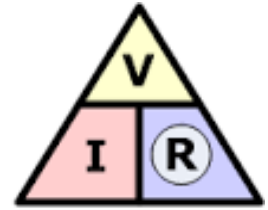
- ask for what part of ohm's law you are searching for
- calculate wanted value
- print out the wanted value



$$V = I \times R$$



$$I = \frac{V}{R}$$



$$R = \frac{V}{I}$$

```
def ohms_law():  
    V = input("Enter voltage: ")  
    C = input("Enter current: ")  
  
    R = float(V) / C  
  
    print("The resistance needed is %.4f\n" % (R))
```

Go to: https://www.onlinegdb.com/online_python_compiler

Workshop (pt. 2)

Quadratic Formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Create a method that can

- take in values for a, b, and c
- calculate the two solutions of it (+/-)
- print out the two solutions

Go to: https://www.onlinegdb.com/online_python_compiler

Workshop (pt. 3)

Parallel Circuit:

Create a method that can

- take in resistance for any number of branches of a parallel circuit
- calculate actual resistance for the branch
- print out the total resistance

$$\frac{1}{R_{\text{Total}}} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \dots + \frac{1}{Rn}$$

Go to: https://www.onlinegdb.com/online_python_compiler

Workshop (pt. 4)

Final Part:

Create a main method that allows any method to be run at any time by taking in a user's input and matching it to an if statement.

Go to: https://www.onlinegdb.com/online_python_compiler