

```
#Dylan Wanser
#Simple outline of all concepts learned from COP 1500
#w3schools
#pynative
#learnpython.org
```

```
#THE FOLLOWING SECTION INSTRUCTS THE USER ON OPTIONS FOR WORKING FUNCTIONS
print("Welcome user. This will instruct you on how to open and run all.")
print("functions this program has to offer, and what they do.")
print("\nPlease input 'intro()' for the introductory portion of this program.")
print("Please input 'arithmetic()' for the math related portion.")
print("Please input 'string()' for ")
print("Please input 'conditionalStructures()' for comparing values.")
print("Please input 'relationalOps()' for comparison between numbers.")
print("Please input 'booleanOps()' for comparison between multiple numbers.")
print("Please input 'iterativeStructures()' for more complex functions.")
#This section allows the user to pick any function created in this program
print("\nOnce the previous program stops prompting for answers you may input")
print("any of the options previously mentioned again and it will execute")
print("that section.")
#This informs the user they may execute any number of functions and when to
```

```
def intro():
    print("Hello World")
    print("Welcome to the program, user.")

    wordInput = input("Please input a word of your choosing to be displayed. ")
    print(wordInput)
    #Word input to demonstrate input functions
    intInput = input("Now please input a number to be displayed. ")
    print(intInput)
    #Numeric input to demonstrate a robust input function that will not crash
    #the program if incorrect input it entered
    print("\n")
    #this generates space to make each segment of the program easier to read
```

```
def arithmetic():
    print("2 + 3 =", 2+3)
    #simple addition
    print("5 - 1 =", 5-1)
    #simple subtraction
    print("2 * 5 =", 2*5)
    #multiplication
    print("10 / 2 =", 10/2)
    #division
    print("2 ** 5 = ", 2**5)
    #exponentation or This to the That power
    print("11 // 3 =", 11//3)
    #floor division, no remainder
    print("11 % 2 =", 11%2)
    #modulus, shows remainder
    print("2.2 + 1.9 =", 2.2+1.9)
    #addition with decimals
    print("4.38 - 1 =", 4.38 - 1)
    #subtraction between a number with and without a decimal
    print("4.1 * 2 =", 4.1 * 2)
    print("4 * .5 =", 4 * 0.5)
    #multiplication with decimals
    print("2 / .5 =", 2/0.5)
    print("5.9 / 3 =", 5.9/3)
    #division with decimals
    print("\n")
    #this generates space to make each segment of the program easier to read
```

```
def string():
    print("The next line shows string operators being used to combine words")
    print("and to show individual words in a sentence")
```

```

stringWordOne = input("What's your favorite pie flavor? ")
stringAsterisk = 2
print("My favorite kind of pie is: " + stringWordOne)
#Adds the word stored in the variable to the end of the sentence without space
print("Repeat this twice. " * stringAsterisk)
#Changes how many times a string is displayed based on the saved number
print("\n")
#this generates space to make each segment of the program easier to read

```

```

def conditionalStructures():
    ifStatementVar = 3
    if(2 < ifStatementVar):
        print("2 < 3 is true")
    #If statement with no else or elif conditions
    if(15 < ifStatementVar):
        print("This statement will not print")
    else:
        print("15 < 3 is false")
    #If statement with else condition
    if(0 < ifStatementVar):
        print("0 < 3 is true.")
    elif(0 > ifStatementVar):
        print("This statement will not print")
    #If statement with an elif and no else condition
    if(3 == ifStatementVar):
        print("3 = 3 is true.")
    elif(3 < ifStatementVar):
        print("This statement will not print")
    else:
        print("This statement will not print")
    #If statement with an elif and an else condition
    print("\n")
    #this generates space to make each segment of the program easier to read

```

```

def relationalOps():
    if(3 == 3):
        print("3 is equal to 3")
    #equal
    if(5 != 3):
        print("5 is not equal to 3")
    #not equal
    if(9 > 3):
        print("9 is greater than 3")
    #greater than
    if(5 >= 5):
        print("5 is greater than or equal to 5")
    #greater than or equal to
    if(1 < 7):
        print("1 is less than 7")
    #less than
    if(11 <= 11):
        print("11 is equal to or less than 11")
    #less than or equal to
    print("\n")
    #this generates space to make each segment of the program easier to read

```

```

def booleanOps():
    booleanVar = 16
    notVar = "true"
    if(16 == booleanVar and booleanVar < 100):
        print("16 is less than 100 and equal to 16")
    #boolean operator and
    if(16 != booleanVar or booleanVar < 100):
        print("16 is less than 100 or is not equal to 16")
    #boolean operator or
    print(not notVar)
    #boolean operator not

```

```

print("\n")
#this generates space to make each segment of the program easier to read

def iterativeStructures():
    whileVar = 26
    while whileVar <30:
        print(whileVar)
        whileVar += 1
    #simple while structure that will count to just before 30
    forVar = "onomatopoeia"
    for x in forVar:
        print(x)
    #this for loop will print the word onomatopoeia with every letter on a new line
    for i in range(5):
        print(i, end=', ')
    #this will print all whole numbers before 5 starting at 0
    inVarList = ["Charge", "Table", "Pine", "Rome"]
    if("Table" in inVarList):
        print("\nTable is a word in the variable list for this structure.")

```