

# CoderZ Summer Assignment #3

Shaurya Singh

## Contents

<b>1</b>	<b>Operators</b>	<b>2</b>
1.1	Types of Operator . . . . .	2
1.2	Arithmetic Operators . . . . .	2
1.3	Comparison Operators . . . . .	3
1.4	Assignment operators . . . . .	4
1.5	Logical Operators . . . . .	5
<b>2</b>	<b>Descision making</b>	<b>5</b>
2.1	If statement . . . . .	6
2.2	If else statement . . . . .	6
2.3	Nested if statements . . . . .	7
<b>3</b>	<b>Loops</b>	<b>7</b>
3.1	While loop . . . . .	7
3.2	For loop . . . . .	8
3.3	Loop Control Statements . . . . .	9
<b>4</b>	<b>Assignment</b>	<b>11</b>

Hello everyone! This is the third assignment you will receive this summer. By this point, all of you should have received and submitted Assignment #1 as well as Assignment #2, which had you learn the basics of python and variables . If you still need help, feel free to email me at [shaunsingho207@gmail.com](mailto:shaunsingho207@gmail.com).

In Assignment #3, you will learn how to use operators, descision making, and loops

# 1 Operators

Operators are constructs which manipulate the value of operands. Operands are the numbers or values you are acting on, and the operator is what you're doing to them

Consider the expression  $6 + 10 = 16$ . Here, 6 and 10 are called the operands and + is called the operator.

## 1.1 Types of Operator

You will need to know the following operators from the python language:

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators

## 1.2 Arithmetic Operators

Arithmetic Operators are the ones you commonly use in math. This consists of addition, subtraction, multiplication, division, modulus, exponents, and floor division. An example is below:

```
#!/usr/bin/python3

a = 21
b = 10
c = 0

c = a + b
print ("Line 1 - Value of c is ", c)

c = a - b
print ("Line 2 - Value of c is ", c )

c = a * b
print ("Line 3 - Value of c is ", c)

c = a / b
print ("Line 4 - Value of c is ", c )
```

```

c = a % b
print ("Line 5 - Value of c is ", c)

a = 2
b = 3
c = a**b
print ("Line 6 - Value of c is ", c)

a = 10
b = 5
c = a//b
print ("Line 7 - Value of c is ", c)

```

```

Line 1 - Value of c is 31
Line 2 - Value of c is 11
Line 3 - Value of c is 210
Line 4 - Value of c is 2.1
Line 5 - Value of c is 1
Line 6 - Value of c is 8
Line 7 - Value of c is 2

```

### 1.3 Comparison Operators

Comparison operators compare values. These consist of equal, not equal, greater, less, greater/equal, less/equal. An example is below:

```

#!/usr/bin/python3

a = 21
b = 10

if ( a == b ):
    print ("Line 1 - a is equal to b")
else:
    print ("Line 1 - a is not equal to b")

if ( a != b ):
    print ("Line 2 - a is not equal to b")
else:
    print ("Line 2 - a is equal to b")

if ( a < b ):
    print ("Line 3 - a is less than b" )
else:
    print ("Line 3 - a is not less than b")

```

```

if ( a > b ):
    print ("Line 4 - a is greater than b")
else:
    print ("Line 4 - a is not greater than b")

a,b = b,a #values of a and b swapped. a becomes 10, b becomes 21

if ( a ≤ b ):
    print ("Line 5 - a is either less than or equal to b")
else:
    print ("Line 5 - a is neither less than nor equal to b")

if ( b ≥ a ):
    print ("Line 6 - b is either greater than or equal to b")
else:
    print ("Line 6 - b is neither greater than nor equal to b")

```

Line 1 - a is not equal to b  
 Line 2 - a is not equal to b  
 Line 3 - a is not less than b  
 Line 4 - a is greater than b  
 Line 5 - a is either less than or equal to b  
 Line 6 - b is either greater than or equal to b

## 1.4 Assignment operators

Assignment operators assign values. These consist of assign, add and assign, subtract and assign, multiply and assign, divide and assign, modulus and assign, exponent and assign. The only one you need to know is assign, the rest are a combination of the arithmetic operators and assign. An example is below

```

#!/usr/bin/python3

a = 21
b = 10
c = 0

c = a + b
print ("Line 1 - Value of c is ", c)

c += a
print ("Line 2 - Value of c is ", c )

c *= a

```

```

print ("Line 3 - Value of c is ", c )

c /= a
print ("Line 4 - Value of c is ", c )

c = 2
c %= a
print ("Line 5 - Value of c is ", c)

c **= a
print ("Line 6 - Value of c is ", c)

c //= a
print ("Line 7 - Value of c is ", c)

```

```

Line 1 - Value of c is 31
Line 2 - Value of c is 52
Line 3 - Value of c is 1092
Line 4 - Value of c is 52.0
Line 5 - Value of c is 2
Line 6 - Value of c is 2097152
Line 7 - Value of c is 99864

```

## 1.5 Logical Operators

A logical operator is a symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expressions and on the meaning of the operator. These consist of AND, OR, and NOT.

## 2 Descision making

Decision-making is the anticipation of conditions occurring during the execution of a program and specified actions taken according to the conditions.

Decision structures evaluate multiple expressions, which produce TRUE or FALSE as the outcome. You need to determine which action to take and which statements to execute if the outcome is TRUE or FALSE otherwise

Python assumes any non-zero and non-null values are true, and any zero or null values as false. Python has the following types of decision making statements, as do many other languages:

## 2.1 If statement

An if statement consists of a boolean expression followed by one or more statements. It can execute code depending on the result of the statement

```
#!/usr/bin/python3

var1 = 100
if var1:
    print ("1 - Got a true expression value")
    print (var1)

var2 = 0
if var2:
    print ("2 - Got a true expression value")
    print (var2)
print ("Good bye!")
```

```
1 - Got a true expression value
100
Good bye!
```

## 2.2 If else statement

An if statement can also be followed by an else statement, which executes if the expression returns FALSE

```
#!/usr/bin/python3

amount = int(100)

if amount < 1000:
    discount = amount * 0.05
    print ("Discount", discount)
else:
    discount = amount * 0.10
    print ("Discount", discount)

print ("Net payable:", amount - discount)
```

```
Discount 5.0
Net payable: 95.0
```

## 2.3 Nested if statements

You can use if statements (and if else statements) within other if and if else statements too!

```
# !/usr/bin/python3

num = int(100)
if num%2 == 0:
    if num%3 == 0:
        print ("Divisible by 3 and 2")
    else:
        print ("divisible by 2 not divisible by 3")
else:
    if num%3 == 0:
        print ("divisible by 3 not divisible by 2")
    else:
        print ("not Divisible by 2 not divisible by 3")
```

divisible by 2 not divisible by 3

## 3 Loops

In general, statements are executed sequentially – The first statement in a function is executed first, followed by the second, and so on. There may be a situation when you need to execute a block of code several number of times.

Programming languages provide various control structures that allow more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times.

Like If statements, we can also use loops inside other while or for loops!

In python, we have the following types of loops

### 3.1 While loop

While loops repeat a statement or a group of statements while a given condition is true. An example is

```
#!/usr/bin/python3

count = 0
while (count < 4):
    print ('The count is:', count)
    count = count + 1

print ("Good bye!")
```

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
Good bye!
```

## 3.2 For loop

For loops execute a sequence of statements multiple times and abbreviate the code that manages the loop variable. An example is traversing a sequence of strings or a list sequence

```
#!/usr/bin/python3

for letter in 'Python':    # traversal of a string sequence
    print ('Current Letter :', letter)
print()
fruits = ['banana', 'apple', 'mango']

for fruit in fruits:       # traversal of List sequence
    print ('Current fruit :', fruit)

print ("Good bye!")
```

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n

Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```



### 3.3 Loop Control Statements

Loop control statements change the execution of a loop. We can break, continue, or pass something to a loop. Examples of these are below:

Break: Terminates the loop

```
#!/usr/bin/python3

for letter in 'Python':    # First Example
    if letter == 'h':
        break
    print ('Current Letter :', letter)

var = 10                    # Second Example
while var > 0:
    print ('Current variable value :', var)
    var = var -1
    if var == 5:
        break

print ("Good bye!")
```

```
Current Letter : P
Current Letter : y
Current Letter : t
Current variable value : 10
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Good bye!
```

Continue: Causes the loop to skip the rest of its body and retest

```
#!/usr/bin/python3

for letter in 'Python':    # First Example
    if letter == 'h':
        continue
    print ('Current Letter :', letter)

var = 10                    # Second Example
while var > 0:
    var = var -1
    if var == 5:
        continue
```

```
        continue
    print ('Current variable value :', var)
print ("Good bye!")
```

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : o
Current Letter : n
Current variable value : 9
Current variable value : 8
Current variable value : 7
Current variable value : 6
Current variable value : 4
Current variable value : 3
Current variable value : 2
Current variable value : 1
Current variable value : 0
Good bye!
```

Pass:

```
#!/usr/bin/python3

for letter in 'Python':
    if letter == 'h':
        pass
    print ('This is pass block')
    print ('Current Letter :', letter)

print ("Good bye!")
```

```
Current Letter : P
Current Letter : y
Current Letter : t
This is pass block
Current Letter : h
Current Letter : o
Current Letter : n
Good bye!
```

## 4 Assignment

This week, your assignment is to create a program which

1. Asks the user for input on three variables (x, y, and z)
2. Multiplies the first two variables
3. Compares the product of step 2 with the third value (if statement)
4. Prints `Variable x+y is greater than z` if its greater, else it should print `Variable z is greater than x+y` if its less