Python Basics

Lesson 03 – Operators

Operators

These are special symbols that represent computations.

Types of Operators:

- + Addition
- Subtraction
- Multiplication
- / Division
- % Modulus
- ** Exponent
- // Floor Division

Operators are applied to values, which are called operands.

Arithmetic Operators

The following table contains the complete list of operators, their descriptions, and examples:

Operator	Description	Example
+ Addition	Adds values on either side of the operator	a + b = 30
- Subtraction	Subtracts the right hand operand from the left hand operand	a – b = -10
• Multiplication	Multiplies values on either side of the operator	a * b = 200
/ Division	Divides the left hand operand by the right hand operand	b / a = 2
% Modulus	Divides the left hand operand by the right hand operand and returns a reminder	b % a = 0
** Exponent	Returns an exponential (power) calculation on operators	a**b = 10 to the power 20
// Floor Division	Represents the division of operands where the result is the quotient; the digits after the decimal point are removed	9//2 = 4 and 9.0//2.0 = 4.0

Arithmetic Operation on Numeric values

All arithmetic operations are applicable on numeric values using operators.

Example:

```
>>>a= 10
>>>b= 15
>>>multiplication_value= a*b
>>>sum= a+b
>>>division= a/b
>>>square_value= a**2
```

Order of Operands

What should be done when an expression has multiple operands?

Example:
2*3+1

In these situations, the following precedence order is followed:

- Parentheses: top precedence ((1+1)*8 = 16)
- Exponentiation: next highest precedence ((1+1) ** (5-2) = 8)
- Multiplication and Division: same precedence (25/ 2 * 3.14 = 39.25)

Operators with the same precedence are evaluated from left to right (except exponentiation); Remember the order of operands as "PEMDAS".

Operators On Strings

You cannot apply all arithmetic operations on strings.

/ and -	Not used with strings
+ and *	Used with strings

"+" is used to concatenate strings.

Example:

```
>>> a = "Hello"
>>> b = "World !!!"
>>> a + b
Output: "Hello world !!!"
```

"*"performs repetition in strings.

Example:

```
>>>a = "Spam"
>>> a * 3
Output: "SpamSpamSpam"
```

Logical Operator

Logical Operators	
And	return true if all the condition are true
Or	return false if all condition are false
not	return true for false and false for true

Logical Operator

Logical variables are used with Boolean expressions or variables. In an “and” statement, all variables should be True for overall True result.

And	Or	Not

Example:

```
>>> a= True
>>> b= True
>>> c = False
>>> a and b
Output True
>>> a and c
Output: False
```

Logical Operator

In an “or” statement, only one variable should be True for overall True result.

And	Or	Not

Example:

```
>>> a= True
>>> b= True
>>> c = False
>>> a or b
Output True
>>> a or c
Output: True
```

Logical Operator

In a “not” statement, the opposite of a value is returned.

And	Or	Not

Example:

>>> a= True

>>> not a

Output: False

Relational Operator

Relational Operators	
==	Equal to Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Variable Comparison

You can check whether there is equality/inequality between two variables. Some examples are:

Equality (==)

Example:

```
>>> a = 10
>>> b = 10
>>> a == b
True
>>> c = 11
>>> a == c
False
```

Inequality (!= or <>)

Examples:

-
```
>>> a = 10
>>> b = 10
>>> a != b
False
>>> c = 11
>>> a != c
True
```
-
```
>>> a <> b
False
```

Variable Comparison

It is also possible to check if a variable is greater than or less than the other variable:

Greater (>)

Example:

```
>>> a = 10
>>> b = 5
>>> a > b
True
>>> c = 11
>>> a > c
False
```

Lesser (<)

Example:

```
>>> a = 10
>>> b = 5
>>> a < b
False
>>> c = 11
>>> a < c
True
```

Variable Comparison

Additionally, you can also check if a variable is greater than equal to or lesser than equal to the variable at its right side:

Greater than equal to (>=)

Example:

```
>>> a =10
>>> b = 10
>>> a>=b
True
```

Lesser than or equal to (<=)

Example:

```
>>> a = 10
>>> b = 10
>>> a <= b
True
```

Relational Operators

- >< it's a relational operators
- Every time we use relational operators we get Boolean value back either true or false.
- 6 relational operator available (<,>,>=,<=,==, !=)

```
>>> 3 > 7
False
>>> 4 < 8
True
>>> 3 < 1
False
>>> 4 >= 4
True
>>> 5 <= 6
True
>>> 5 <= 5
True
>>> if 5 = 5:

SyntaxError: invalid syntax
>>> apple = 5
>>> if 5 == 5:
        print("hello")


hello
>>> 4 != 6
True
>>> 4 != 4
False
>>>
```

Assignment Operators

Assignment Operators	
=	C = A + B will assign value of A + B into C
+=	B += A is equivalent to B = B + A
-=	B -= A is equivalent to B = B – A
*=	B *= A is equivalent to B = B * A
/=	B /= A is equivalent to B = B / A
**=	B **= A is equivalent to B = B ** A
//=	B //= 2 is same as B = B // 2

Membership Operators

Membership Operator	
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise.
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.

Print (“p” in “python”)

The above statement return **True**

Print (“a” not in “python”)

The above statement return **True**

Identity Operators

Identity Operator	
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise.
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise.

Print (type(7) is int)

The above statement return **True**

Print (type(7.2) is not int)

The above statement return **True**

Quiz

Quiz 1

Which of the following statements is/are true?

Select all that apply.

- [ ] a. "= =" is used for assignment in Python.
- [ ] b. "=" is used for assignment in Python.
- [ ] c. Logical "or" is true only when all operands are true.
- [ ] d. "**" is an exponential operator.

Quiz 2

What will be the output of “hello” +3?

- a. hello 3
- b. Not allowed
- c. hellohellohello
- d. True

Quiz 3

Which of the following operators has the highest precedence?

- a. +
- b. **
- c. ( )
- d. /

Quiz 4

Which function is used to return the datatype of a variable?

- a. data_type
- b. type_value
- c. type
- d. value

Quiz 5

Evaluate (2+2)**3+6.

- a. 24
- b. 70
- c. 72
- d. 64

Exercise

Exercise 1: *Create a List of your favorite songs. Then create a list of your favorite movies. Join the two lists together (Hint: List1 + List2). Finally, append your favorite book to the end of the list and print it.*

Exercise 2: *There were 20 people on a bus. 10 got off. 3 came on. 15 came on. 5 came off. If there were 3 more buses, and each had 15 people throughout the bus ride, how many people were on all the buses at the last stop? (Should be done via one equation in Python Shell)*

Exercise 3: *Create a small program that will take in a User's name and last name (Hint: varName = input("Enter your name")), store both in two variables. And then print out a message saying ("Hello there, FirstName LastName") (Using the %s symbols)*

Answers

Answer 1:

```
favSongs = ["LA Story, Lost Stars, Grenade"]
favMovies = ["Avengers", "Iron Man", "Hulk"]
favorites = favSongs + favMovies
favorites.append("NYPD Red") #Amazing Book ;)
print(favorites)
```

Answer 2:

```
>>> 20-10+3+15-5 + (3 * 15)
68
```

Answer 3:

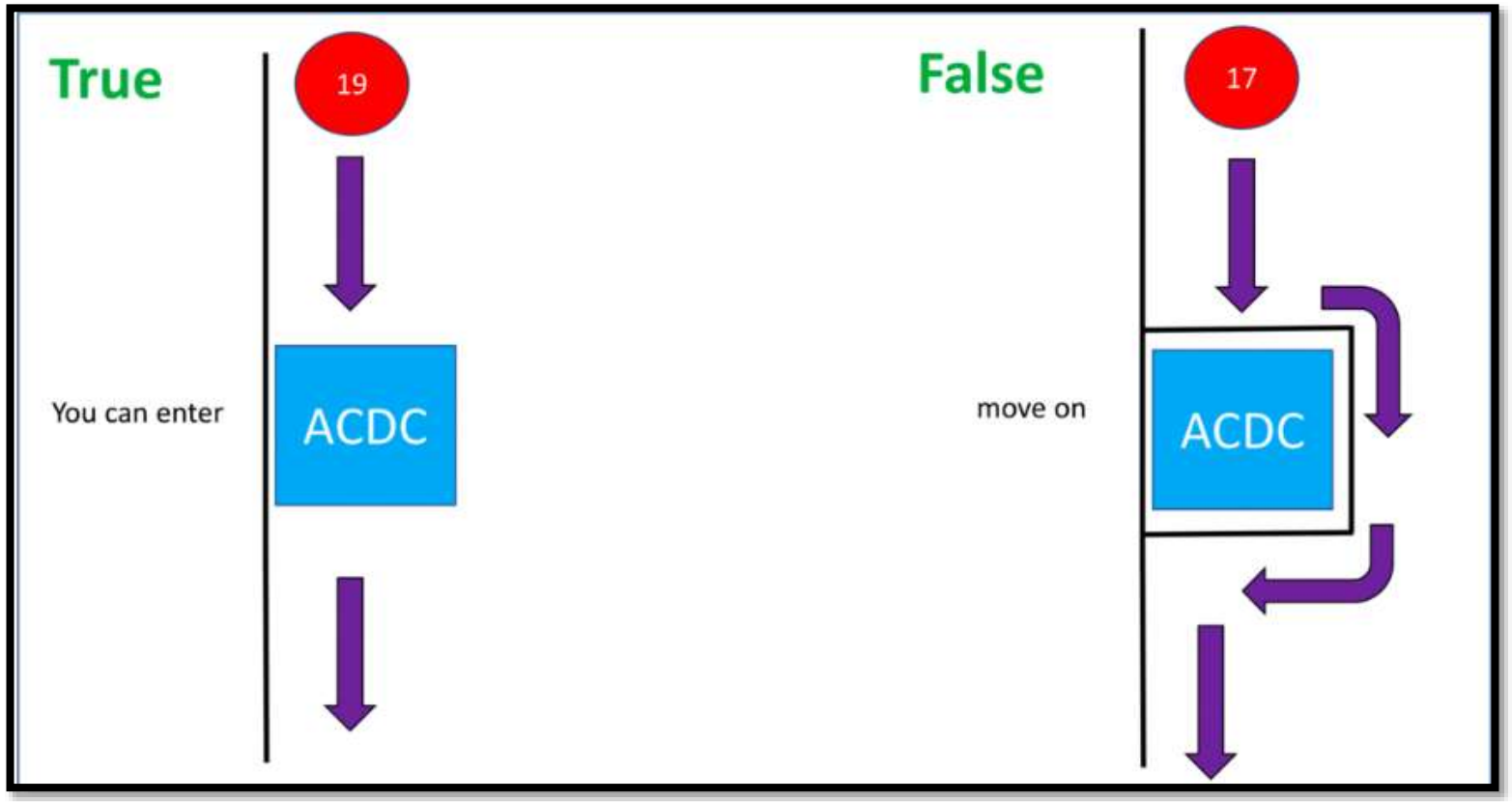
```
First = input("Enter your first name: ")
Last = input("Enter your last name: ")
greeting = ("Hello there, %s %s")
print(greeting %(First,Last))
```

Python Basics

Lesson 04—Control Statements

Branching

Branching allows us to run different statements for different inputs. It is helpful to think of an **if statement** as a locked room, if the statement is true you can enter the room and your program will run some predefined task, but if the statement is false your program will skip the task.



Conditional Statements

Control or Conditional statements allow to check the conditions and change the behavior of a program.

Abilities:

- Run a selected section
- Control the flow of a program
- Cover different scenarios

Example:

```
if x>0:
        print "x is positive"
```

Types of Conditional Statements

There are five types of conditional statements:

- If
- If...else
- If...else if (If...elif)
- If...else if...else (if...elif...else)
- Nested if

If Statements

These are used to run conditional statements.

If Condition = True	The code runs.
If Condition = False	Nothing happens.

Example:

```
if x > 0:
        print "x is positive"
```

If...Else Statements

These are used to control the flow of a program and run conditional blocks.

Example 1

```
age = 20
if age == 20:
        print "age is 20 years"
else:
        print "age is not 20"
```

Example 2

```
if age > 18:
        print "person is adult"
else:
        print "person is not adult"
```

If-Else Statements

```
>>> if 5 > 2:
        print("My name is BOB")

My name is BOB
>>> if 5 > 2:
        print("My name is BOB")
else :
        print("The earth has collapsed")

My name is BOB
>>> if 2 > 5:
        print("My name is BOB")
else :
        print("The earth has collapsed")

The earth has collapsed
>>>
```

Elif Statements

```
>>> day = "Tueday"
>>> if day == "Monday":
        print("Sunny!")
else"
SyntaxError: EOL while scanning string literal
>>> if day == "Monday":
        print("Sunny!")
else"
SyntaxError: EOL while scanning string literal
>>> if day == "Monday":
        print("Sunny!")
else:
        print("Rainy")


Rainy
>>> if day == "Monday":
        print("Sunny!")
elif day == "Tuseday":
        print("Cloudy!")
else:
        print("R")
```

Logical Operators

```
>>> age = 14
>>> if age > 13:
        print("You are elligible for facebook")


You are elligible for facebook
>>> if age > 13 and name == "Avi":
        print("Elligible")


Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    if age > 13 and name == "Avi":
NameError: name 'name' is not defined
>>> name = "Bob"
>>> if age > 13 or name == "Avi":
        print("Elligible")


Elligible
>>>
```

If...Else If Statements

These are used to combine multiple If statements. These statements:

- Execute only one branch
- Can be innumerable

Example:

```
marks= 95
If marks > 90:
        print “A grade”
elif marks >= 80:
        print “B grade”
elif marks >= 60
        print “C grade”
```

If…Else If…Else Statements

These are combined statements. The else statement executes, when none of the conditions are met.

Example:

```
marks= 95
If marks > 90:
        print “A grade”
elif marks >= 80:
        print “B grade”
elif marks >= 60
        print “C grade”
Else:
        print “fail”
```

Nested If Statements

When these are used, one condition can also be nested within another.

Example:

```
If location == 'US':
        if age > 18:
                print “person is adult from US”
else:
        print “person is not from US"
```

Nested If/Else

```
>>> age = 14
>>> if age > 13:
        print("You are elligib le for facebook")


You are elligib le for facebook
>>> if age > 13:

        print("You are elligib le for facebook")


You are elligib le for facebook
>>> name = "Bob"
>>> if age > 13:
        if name == "Bob":
            print("You get special privileges")
        else:
            print("You are elligible for facebook")


You get special privileges
>>>
```

In Clause

When using conditional statements, sometimes, you need to know if an element exists in a list or not. In such cases, use the “in clause”.

Example:

```
name_list = ['harry', 'john','andy' ,'rose' ]
name = 'harry'
If name in name_list:
        print “name is in list”
```

Quiz

Quiz 1

Which keyword should be used to keep the if or else block empty?

- a. break
- b. pass
- c. continue
- d. type

Quiz 2

What is the Python keyword for else if statements?

- a. else if
- b. else
- c. elif
- d. nested

Exercise

Exercise 1: *Using if statements, create a variable called day, set it to "Tuesday". Check to see if day is equal to "Monday"* ***or*** *"Tuesday", and if it is, print, "Today is sunny". If it is not, print "Today it will rain"*

Answer 1:

```
day = "Tuesday"
if day == "Tuesday" or day == "Monday":
        print("Today is sunny")
else:
        print("Today it will rain")
```

Exercise List

Q. Find the greatest number among two

Q. Check whether a number is +ve, -ve or zero

Q. Check whether a number is even or odd

Q. Find the grade of a student

marks	>	80	A	grade
marks	>	60	B	Grade
marks	>	40	C	grade
marks	<	40	D	grade

Q. Find the greatest number among three

Q. Check whether a year is leap year or not

if a year is divisible by 400 its a leap year

or

if a year is divisible by 4 and not divisible by 100 its a leap year

Q. Write a Python program to convert temperatures to and from celsius, fahrenheit.

60°C is 140 in Fahrenheit

45°F is 7 in Celsius

$$C = (5 (F - 32)) / 9$$

$$F = (9C + (32 * 5)) / 5$$