KHWARIZMI SCIENCE SOCIETY

CODING WITH PYTHON

A COURSE FOR ABSOLUTE BEGINNERS

iDRACK

KHWARIZMI
SCIENCE SOCIETY

iDRACK

https://idrack.org/

https://idrack.org/forum/community/

contact@idrack.org

COURSE OUTLINE

KHWARIZMI
SCIENCE SOCIETY

5 WEEKS, SATURDAYS, 2-5 PM

Week 1:

- An introduction to Python
- Installing Python on Windows
- Python Shell

Week 2:

- Saving & Running scripts
- Operators & Variables
- Working with Strings

Week 3:

- Python Collections
- Condition Blocks

Week 4:

- Writing Loops
- Functions in Python

Week 5:

- Introducing Modules
- In-class Assessment

iDRACK

INSTRUCTOR CONTACT

ROOP OMAR

(roop.omar@gmail.com)

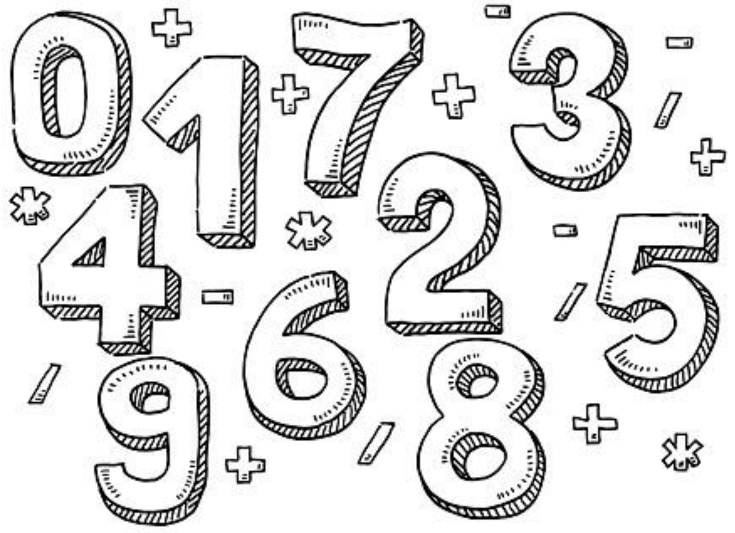
We will try to cater to all queries within class timings, but if you feel there is something you need help with later, or were not able to ask during the session, please feel free to drop me an email, and I will get back to you as soon as I can.

KHWARIZMI SCIENCE SOCIETY

DATA-TYPES IN PYTHON

iDRACK

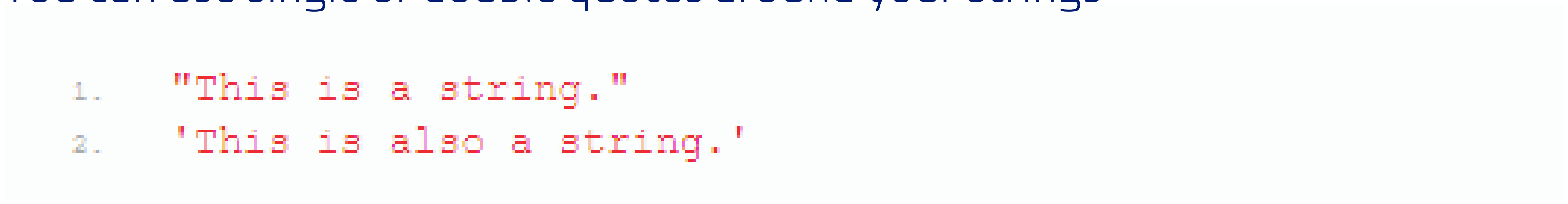
PYTHON NUMBERS



- These include:
  - ✓ Integer
  - ✓ Floats
  - ✓ Complex number
- they are represented as int, float and complex.
- **Integers** - can hold a value of any length, the only limitation being the amount of memory available.
- **Float, or "floating point number"** - is only accurate up to 15 decimal places. After that, it rounds the number off.
- **Complex numbers** - are written with a "j" as the imaginary part.

STRINGS IN PYTHON

- A string is simply a series of characters.
- Anything inside quotes is considered a string in Python
- You can use single or double quotes around your strings



```python
"This is a string."
'This is also a string.'
```

- This flexibility allows you to use quotes and apostrophes within your strings

LISTS IN PYTHON

- A list is a collection of items in a particular order.
- It can include
  - ✓ the letters of the alphabet
  - ✓ the digits from 0–9
  - ✓ the names of all the people in your family.
- You can put anything you want into a list
- The items in your list don't have to be related.

DICTIONARIES IN PYTHON

- A dictionary in Python is a collection of **key-value pairs**.
- Each key is connected to a value
- You can use a key to access the value associated with that key.
- A key's value can be a number, a string, a list, or even another dictionary.

```
fruits = {'value':tomato,'key':red}
```

KHWARIZMI SCIENCE SOCIETY

VARIABLES IN PYTHON

iDRACK

VARIABLES ARE LIKE LABELS

- Each *variable* is connected to a *value*
- This is the information stored in that variable
- It could be a number, text, or a list of the same.
- You can change the value assigned to a variable at any time in your program, and Python will always keep track of the current value.
- If you re-assign a new value to the same variable, it will be replaced.

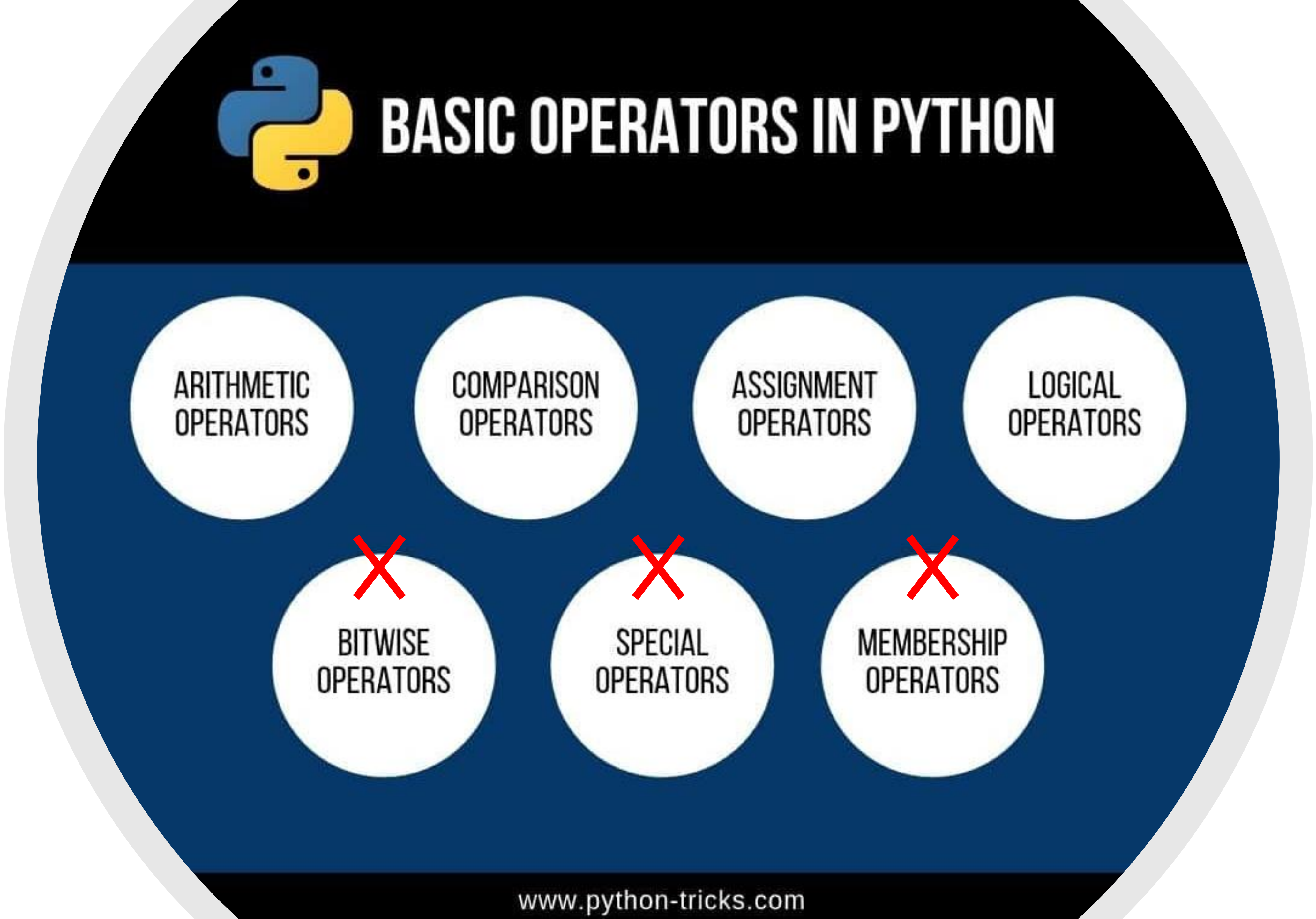
NAMING VARIABLES

- There are some conventions and rules for naming variables.
- Some may cause errors, and others simply make your code more readable.
  - ✓ Variable names can contain only letters, numbers, and underscores
  - ✓ They can start with a letter or underscore, but not a number
  - ✓ Spaces are not allowed, but underscores can be used to separate words
  - ✓ Avoid using python keywords or function names as variable names
  - ✓ Best to keep names short and descriptive
  - ✓ It's best to use lowercase letters in the names.

KHWARIZMI SCIENCE SOCIETY

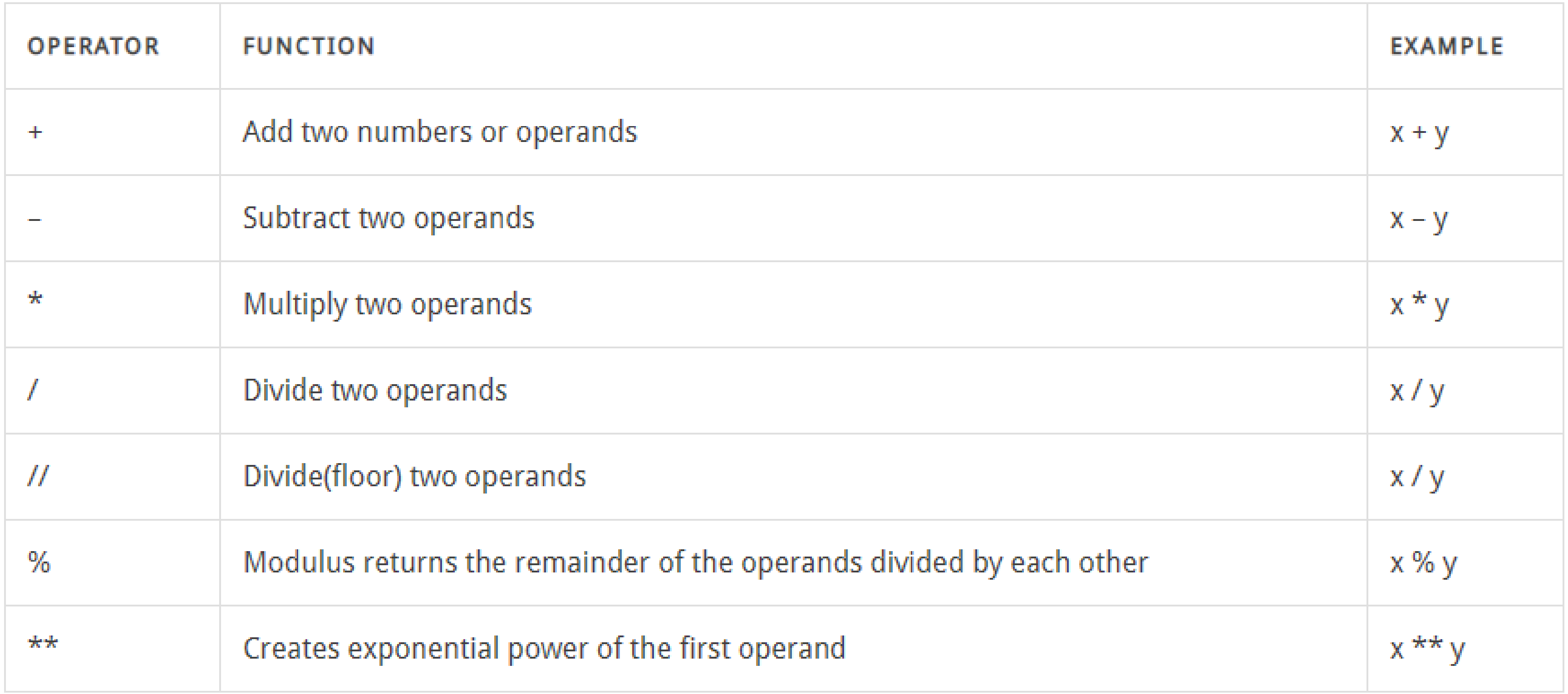
OPERATORS IN PYTHON

iDRACK


BASIC OPERATORS IN PYTHON
ARITHMETIC OPERATORS
COMPARISON OPERATORS
ASSIGNMENT OPERATORS
LOGICAL OPERATORS
BITWISE OPERATORS
SPECIAL OPERATORS
MEMBERSHIP OPERATORS
www.python-tricks.com

ARITHMETIC OPERATORS

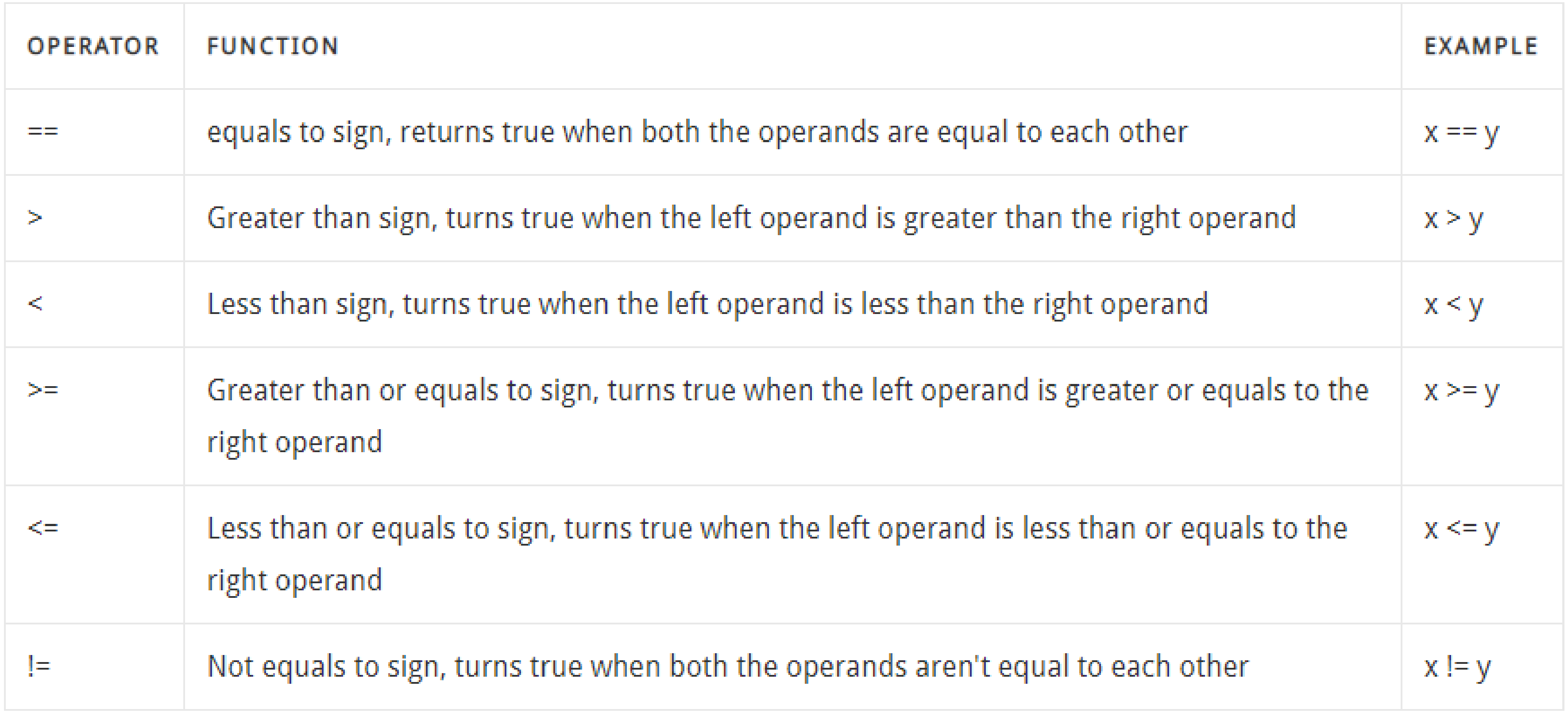
KHWARIZMI SCIENCE SOCIETY



OPERATOR	FUNCTION	EXAMPLE
+	Add two numbers or operands	x + y
–	Subtract two operands	x – y
*	Multiply two operands	x * y
/	Divide two operands	x / y
//	Divide(floor) two operands	x / y
%	Modulus returns the remainder of the operands divided by each other	x % y
**	Creates exponential power of the first operand	x ** y

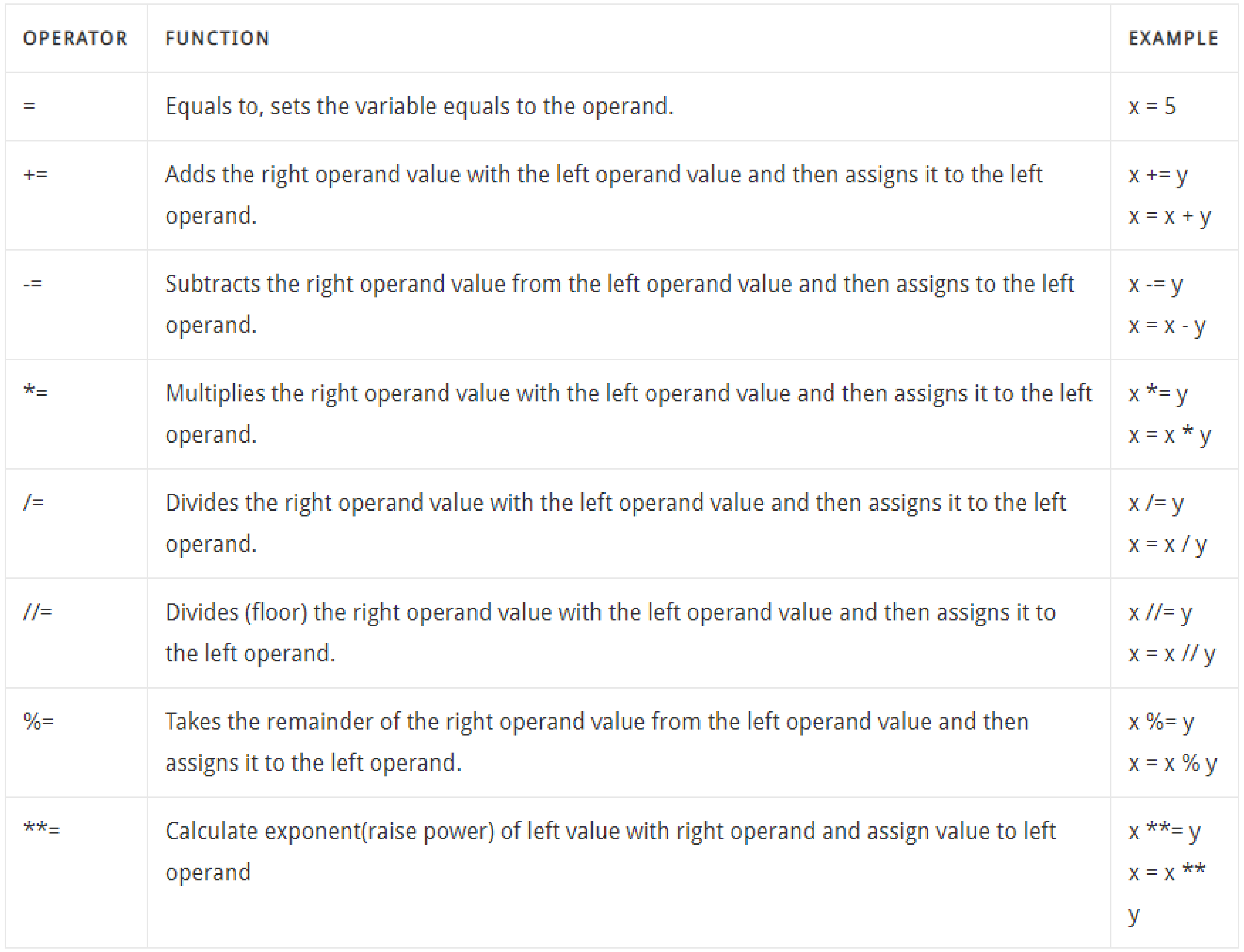
iDRACK

COMPARISON OPERATORS



OPERATOR	FUNCTION	EXAMPLE
==	equals to sign, returns true when both the operands are equal to each other	x == y
>	Greater than sign, turns true when the left operand is greater than the right operand	x > y
<	Less than sign, turns true when the left operand is less than the right operand	x < y
>=	Greater than or equals to sign, turns true when the left operand is greater or equals to the right operand	x >= y
<=	Less than or equals to sign, turns true when the left operand is less than or equals to the right operand	x <= y
!=	Not equals to sign, turns true when both the operands aren't equal to each other	x != y

ASSIGNMENT OPERATORS



OPERATOR	FUNCTION	EXAMPLE
=	Equals to, sets the variable equals to the operand.	x = 5

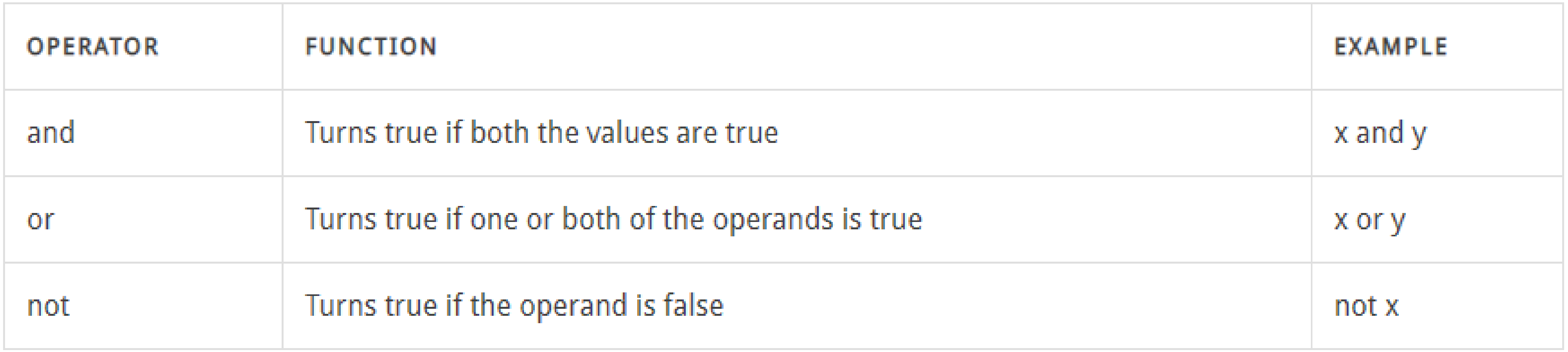
+=	Adds the right operand value with the left operand value and then assigns it to the left operand.	x += y
		x = x + y

-=	Subtracts the right operand value from the left operand value and then assigns to the left operand.	x -= y
		x = x - y

*=	Multiplies the right operand value with the left operand value and then assigns it to the left operand.	x *= y x = x * y
/=	Divides the right operand value with the left operand value and then assigns it to the left operand.	x /= y x = x / y
//=	Divides (floor) the right operand value with the left operand value and then assigns it to the left operand.	x //= y x = x // y
%=	Takes the remainder of the right operand value from the left operand value and then assigns it to the left operand.	x %= y x = x % y
**=	Calculate exponent(raise power) of left value with right operand and assign value to left operand	x **= y x = x ** y

LOGICAL OPERATORS

KHWARIZMI
SCIENCE SOCIETY



OPERATOR	FUNCTION	EXAMPLE
and	Turns true if both the values are true	x and y
or	Turns true if one or both of the operands is true	x or y
not	Turns true if the operand is false	not x

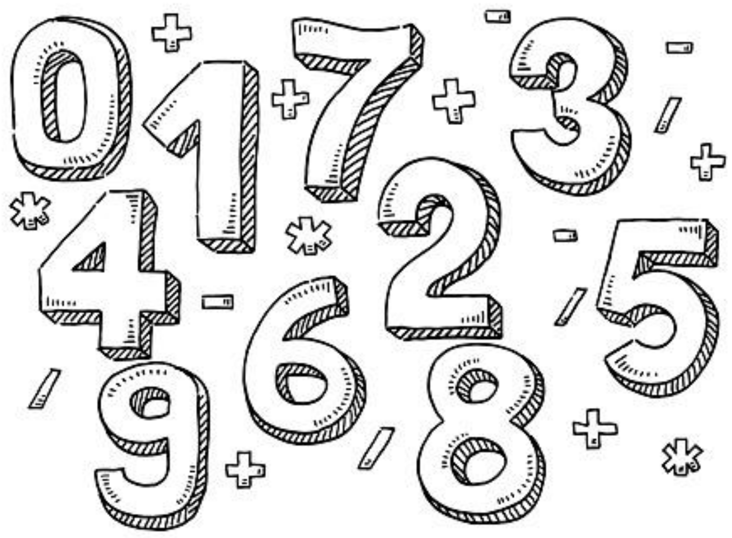
iDRACK

NUMBERS IN PYTHON

KHWARIZMI SCIENCE SOCIETY

iDRACK

PYTHON NUMBERS



- These include:
  - ✓ Integer
  - ✓ Floats
  - ✓ Complex number
- they are represented as int, float and complex.
- **Integers** - can hold a value of any length, the only limitation being the amount of memory available.
- **Float, or "floating point number"** - is only accurate up to 15 decimal places. After that, it rounds the number off.
- **Complex numbers** - are written with a "j" as the imaginary part.

COMMON MATHEMATICAL FUNCTIONS

Function	What it does ?	Example
`abs(number)`	Returns the absolute value of the number. In other words, the `abs()` function just returns the number without any sign.	`abs(-12)` is `12`, `abs(112.21)` is `112.21`.
`pow(a, b)`	Returns `a^b`.	`pow(2, 3)` is `8`, `pow(10, 3)` is `1000`
`round(number)`	Rounds the number to the nearest integer.	`round(17.3)` is `17`, `round(8.6)` is `9`
`round(number, ndigits)`	Rounds the `number` to `ndigits` after decimal point	`round(3.14159, 2)` is `3.14`, `round(2.71828, 2)` is `2.72`
`min(arg1, arg2, ... argN)`	Returns the smallest item among `arg1`, `arg2`, ... `argN`	`min(12, 2, 44, 199)` is `2`, `min(4, -21, -99)` is `-99`
`max(arg1, arg2, ... argN)`	Returns the largest item among `arg1`, `arg2`, ... `argN`	`max(991, 22, 19)` is `991`, `max(-2, -1, -5)` is `-1`

KHWARIZMI
SCIENCE SOCIETY

iDRACK

MORE NUMBER FUNCTIONS

- **type() -** takes one argument and returns which class it belongs to.
- **int() -** converts another numeric type into an int.
- **float()** - converts another numeric type into a float.
- int(), float() can't convert a complex number.
- **complex()** - converts another numeric type into a complex number.
- **abs()** – returns the absolute value of a number
- **log()** – returns natural logarithm of a number
- **round()** – number is rounded to "n" digits from the decimal point.
- **sqrt()** – returns the square root of a number

```
round(number, number of digits)
```

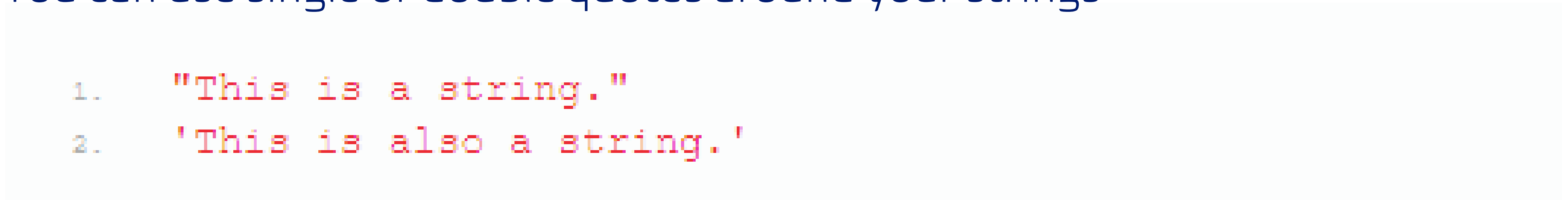
STRINGS IN PYTHON

KHWARIZMI SCIENCE SOCIETY

iDRACK

STRINGS IN PYTHON

- A string is simply a series of characters.
- Anything inside quotes is considered a string in Python
- You can use single or double quotes around your strings



```
"This is a string."
'This is also a string.'
```

- This flexibility allows you to use quotes and apostrophes within your strings

- To display a single character from a string, we put its index in square brackets.
- Remember that indexing begins at 0.

```
>>> a[1]
```

- Sometimes, we may want to display only a part of a string.
- For this, we use the slicing operator [].

```
>>> a[3:8]
```

- There are many variations of this command, we'll look at some of them.

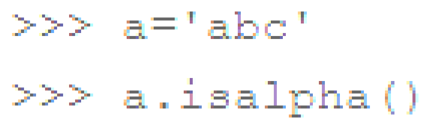
- Concatenation is the operation of joining stuff together.
- In python strings can be joined using the concatenation operator "+".
- Similarly, you can use the "*" to print multiple copies of a string

- Also, you may want to put a tab, or a newline, or other characters in your string.
- We can do this using an escape sequence.
- An escape sequence is a backslash followed by a character.

- **\n –** newline
- **\t –** tab
- **\\ –** backslash
- **\' –** single quote
- **\" –** double quote

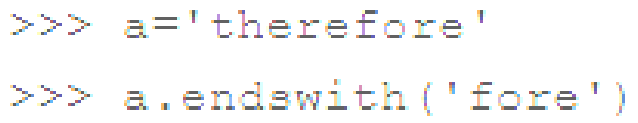
MORE STRING FUNCTIONS

- len() - returns the length of a string.
- strip() - removes whitespaces from the beginning and end of the string.
- isdigit() - returns True if all characters in a string are digits.
- isalpha() - returns True if all characters in a string are alphabets.



```
>>> a='abc'
>>> a.isalpha()
```

- startswith() - takes a string as an argument and returns True if the string begins with the string in the argument.
- endswith() - takes a string as an argument and returns True if the string ends with the string in the argument.



```
>>> a='therefore'
>>> a.endswith('fore')
```

- **find()** - takes an argument and searches for it in the string. Then returns the index of the substring.
- If the string doesn't exist in the main string, then the index it returns is 1.

```
>>> 'homeowner'.find('meow')
```

- **replace()** - takes two arguments. The first is the substring to be replaced. The second is the substring to replace with.

```
>>> 'banana'.replace('na','ha')
```

- **split()** - takes one argument. The string is then split around every occurrence of the argument in the string.

```
>>> 'No. Okay. Why?'.split('.')
```

- In Python, *lower()* is a built-in method used for string handling.
- The *lower()* methods returns the lowercased string from the given string.
- It converts all uppercase characters to lowercase.

```
string = 'GEEKSFORGEEKS'
print(string.lower())

string = 'GeeksforGeeks'
print(string.lower())
```

Output:

```
geeksforgeeks
geeksforgeeks
```

- In Python, *upper()* is a built-in method used for string handling.
- The *upper()* methods returns the uppercased string from the given string.
- It converts all lowercase characters to uppercase.

```
string = 'geeksforgeeks'
print(string.upper())

string = 'My name is ayush'
print(string.upper())
```

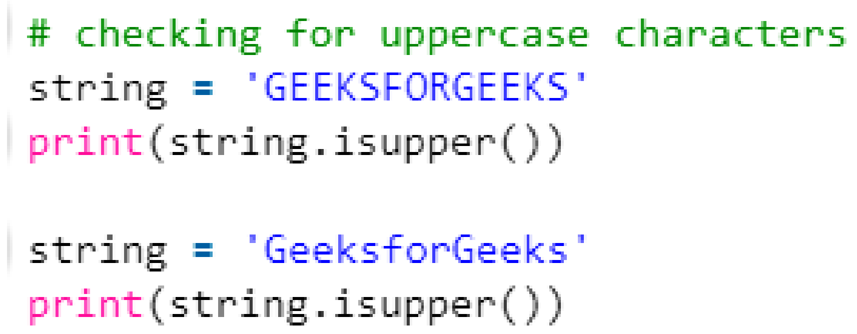
Output:

```
GEEKSFORGEEKS
MY NAME IS AYUSH
```

KHWARIZMI
SCIENCE SOCIETY

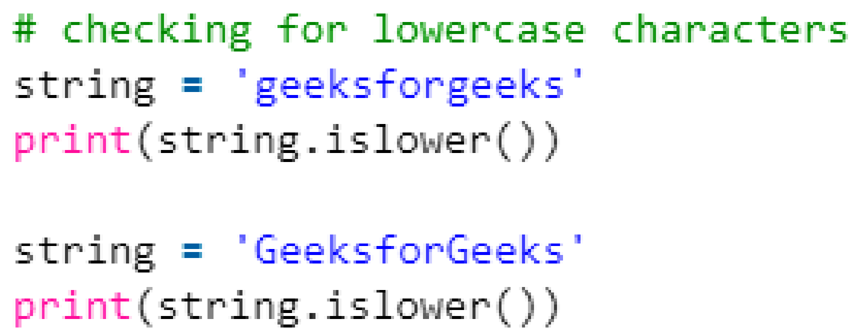
iDRACK

- In Python, *isupper()* is a built-in method used for string handling.
- The *isupper()* methods returns "True" if all characters in the string are uppercase. Otherwise, It returns "False".



```
# checking for uppercase characters
string = 'GEEKSFORGEEKS'
print(string.isupper())

string = 'GeeksforGeeks'
print(string.isupper())
```

- In Python, *islower()* is a built-in method used for string handling.
- The *islower()* methods returns "True" if all characters in the string are lowercase, Otherwise, It returns "False".



```
# checking for lowercase characters
string = 'geeksforgeeks'
print(string.islower())

string = 'GeeksforGeeks'
print(string.islower())
```

KHWARIZMI SCIENCE SOCIETY

SOME ERRORS YOU MIGHT COME ACROSS

iDRACK

SYNTAX ERRORS

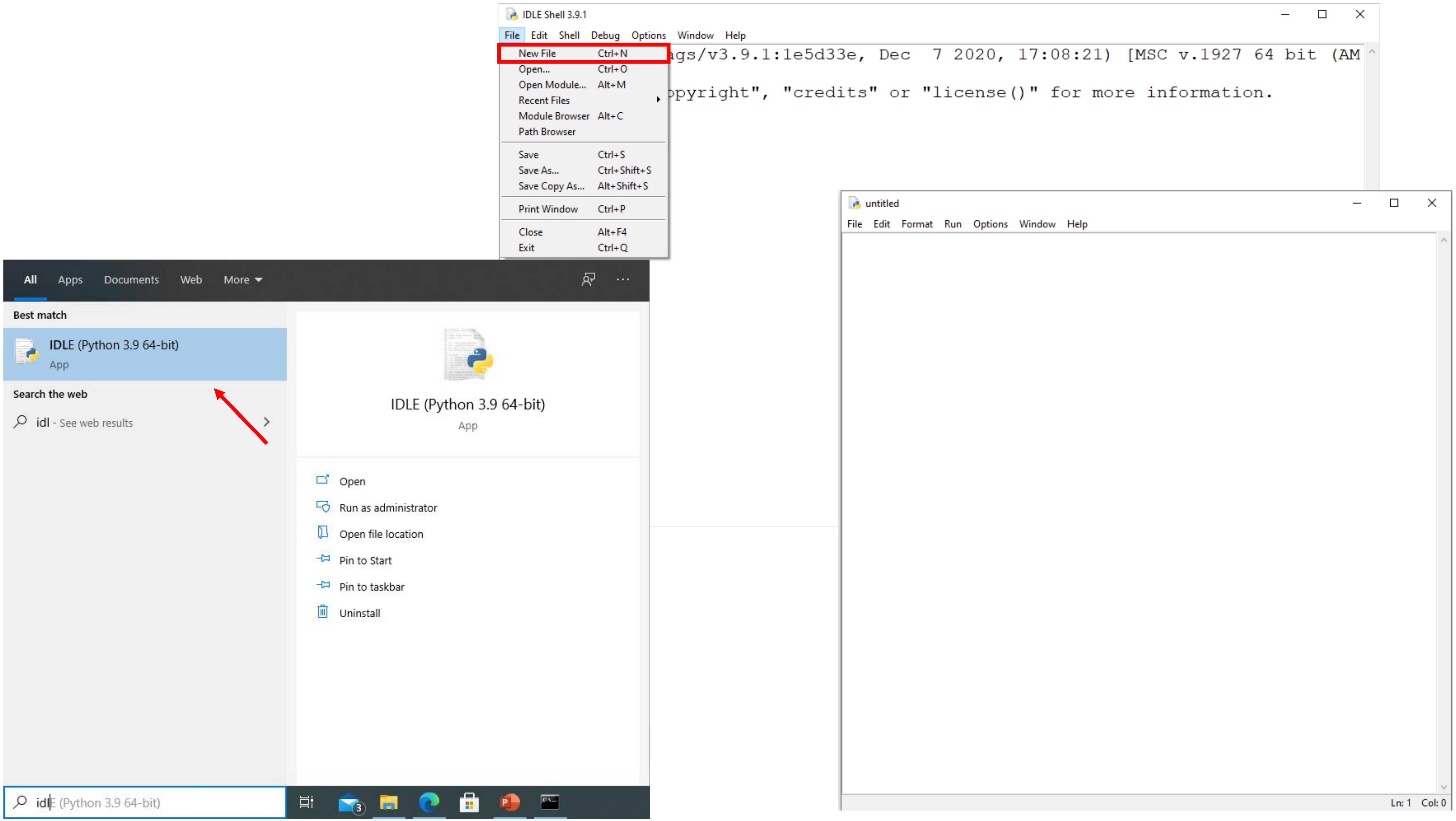
- Frequent errors include:
  - ✓ Having a different number of open and close brackets or parenthesis
  - ✓ Forgetting to place quotes around a string
  - ✓ Misspelled variable or function names. Python is case sensitive.
  - ✓ Using = instead of == or vice versa.
- In Python, "= "is used for assigning a value to a variable, and "== "is used for comparing.
- A typical input-output error is attempting to open a file that does not exist, or does not exist in the location that you specified.
- It is safest to load a data file in the same folder as the Python code that reads or writes to it. Python looks to the same folder by default.

KHWARIZMI SCIENCE SOCIETY

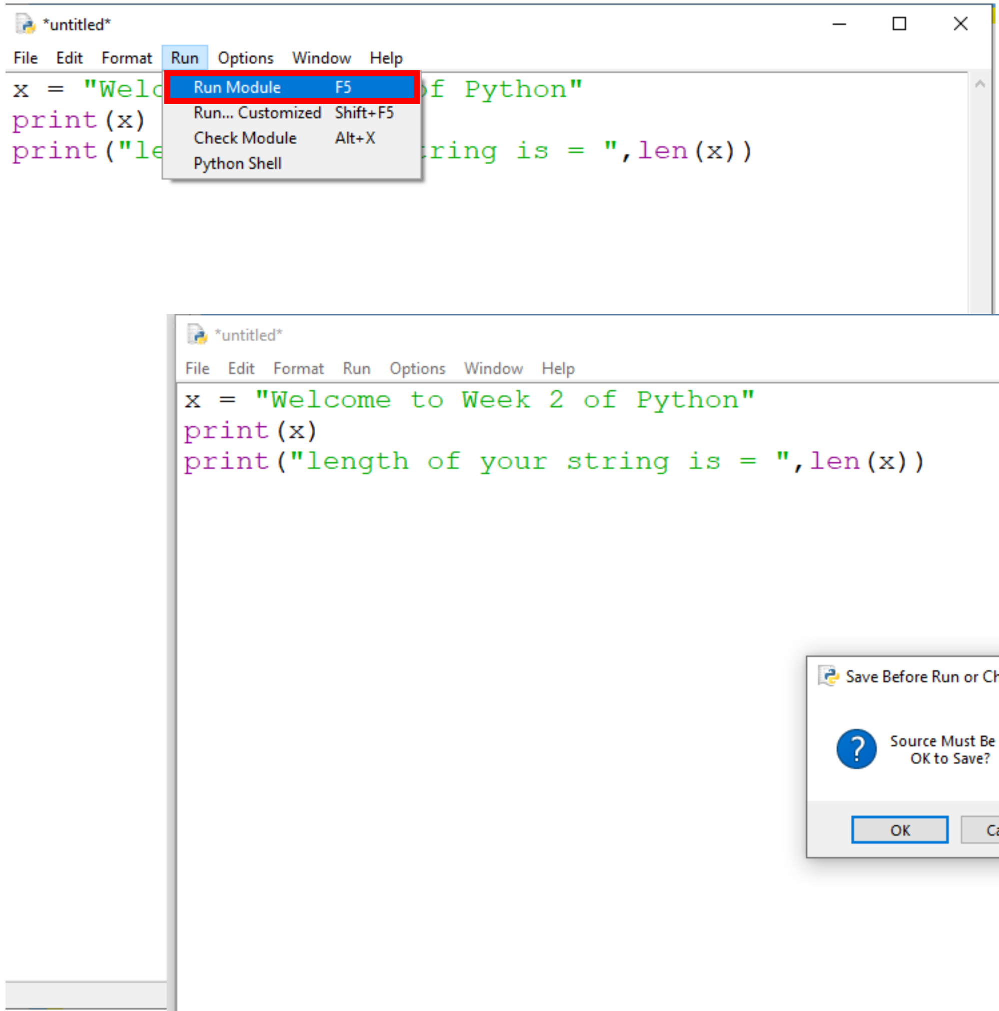
SAVING AND RUNNING PYTHON SCRIPTS

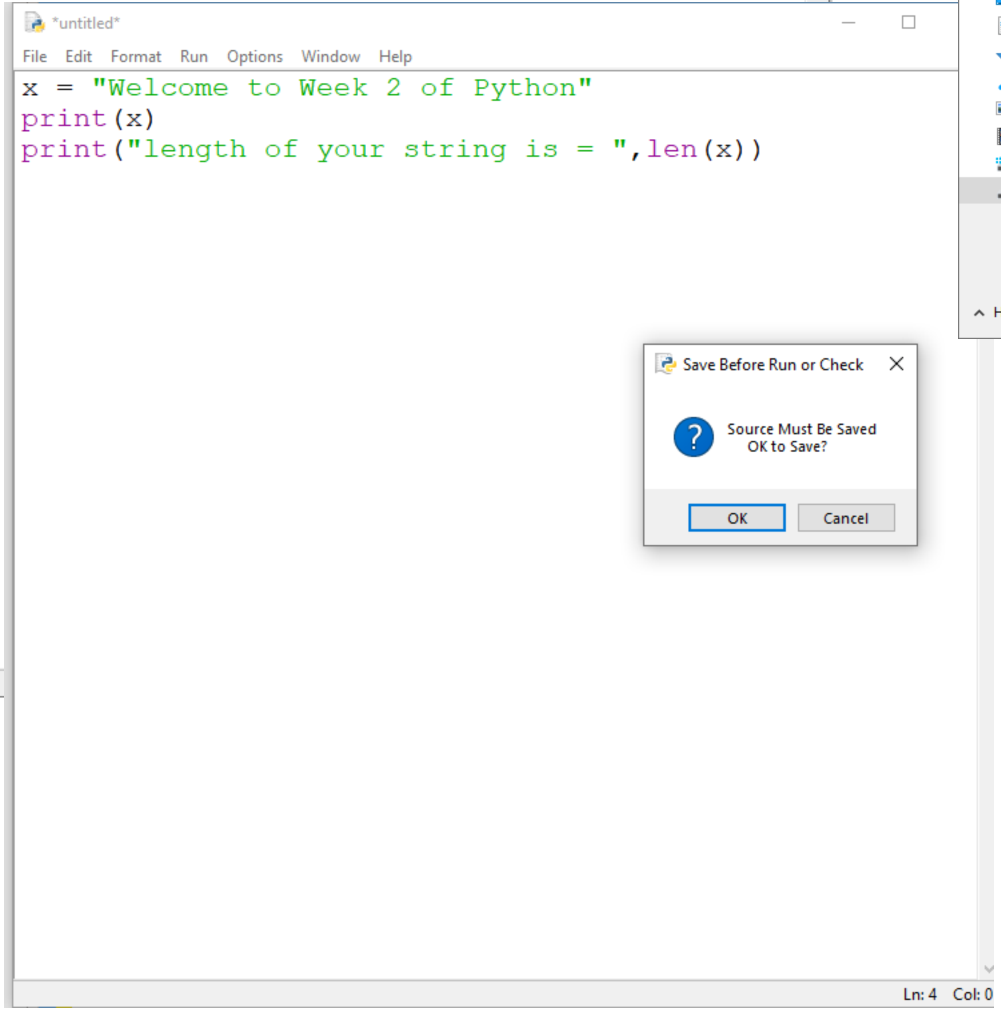
iDRACK

- Entering commands at the prompt is just the beginning.
- IDLE has two modes
  - ✓ *Interactive*
  - ✓ *Script*
- We have been using the interactive mode so far.
- It immediately returns the results of commands we enter into the shell.
- In script mode, you will write a script and then run it.
- Remember to always save your scripts using the **.py** extension !



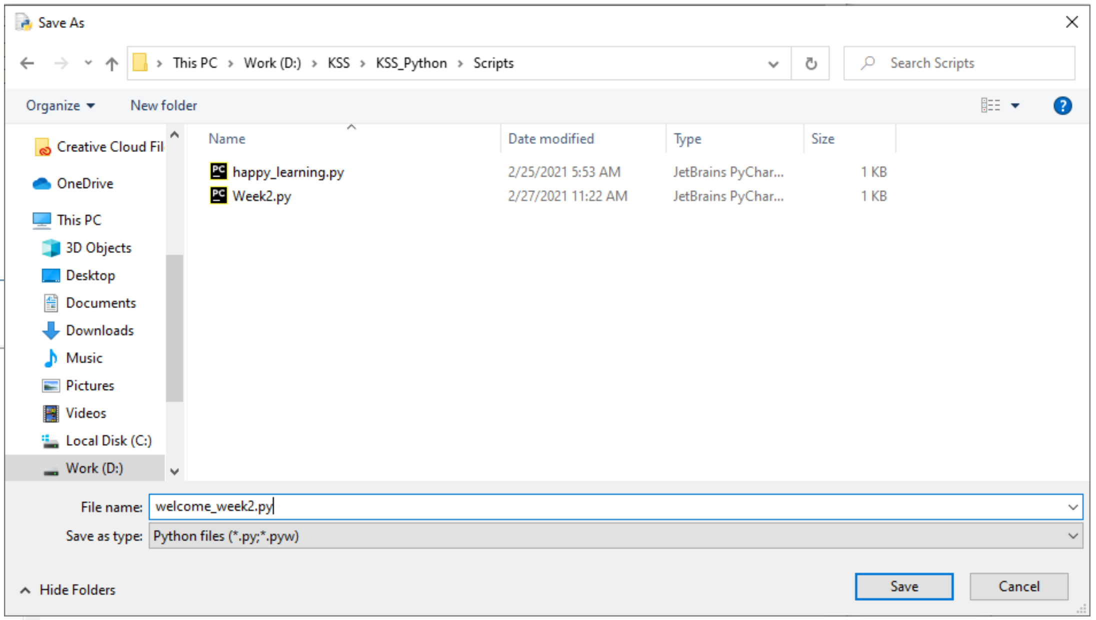
```
ags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AM

opyright", "credits" or "license()" for more information.
```


untitled
File Edit Format Run Options Window Help
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell



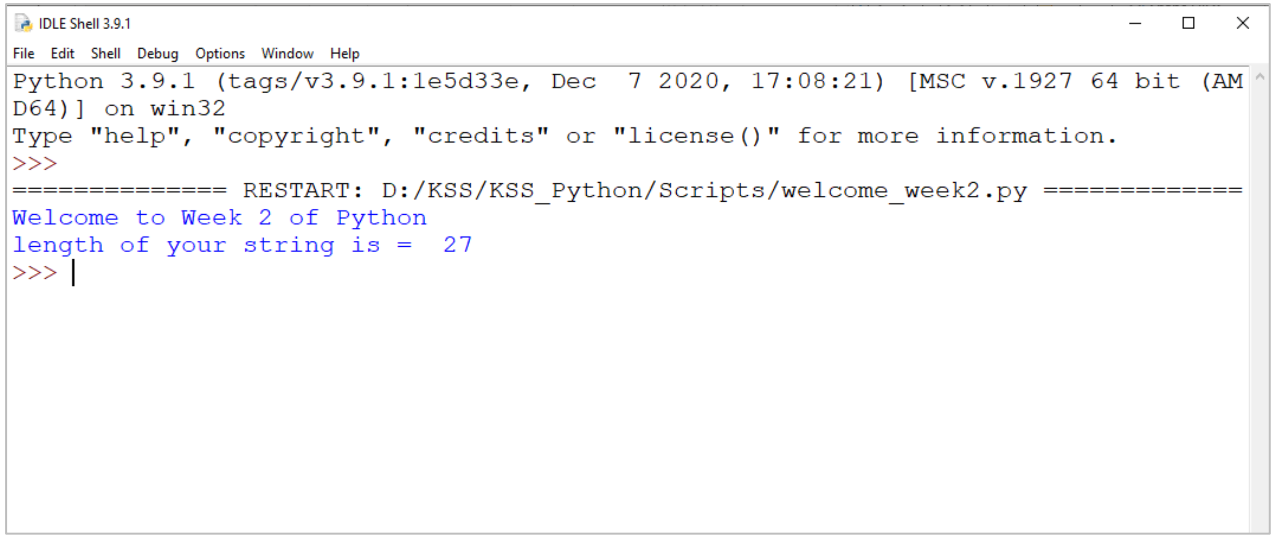
```
x = "Welcome to Week 2 of Python"
print(x)
print("length of your string is = ",len(x))
```

Source Must Be Saved
OK to Save?



Name	Date modified	Type	Size
happy_learning.py	2/25/2021 5:53 AM	JetBrains PyChar...	1 KB
Week2.py	2/27/2021 11:22 AM	JetBrains PyChar...	1 KB

File name: welcome_week2.py
Save as type: Python files (*.py;*.pyw)



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec  7 2020, 17:08:21) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: D:/KSS/KSS_Python/Scripts/welcome_week2.py ============
Welcome to Week 2 of Python
length of your string is =  27
>>> 
```

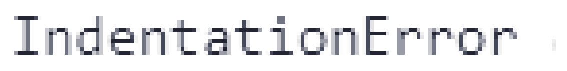
KHWARIZMI
SCIENCE SOCIETY

Python Comments, Indentation, and Statements

iDRACK

INDENTATION IN PYTHON

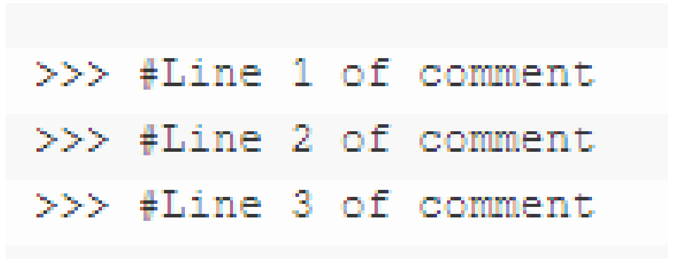
- Unlike C++ or Java, Python does not use curly braces for indentation.
- Instead, Python makes use of indentation.
- Python indentation is a way of telling a Python interpreter that the group of statements belongs to a particular block of code.
- A block is a combination of all these statements.
- There are no strict rules on what kind of Python indentation you use.
- But it must be consistent throughout the block.
- Without proper indentation, you will get an error, and the code will not be compiled.



```
IndentationError
```

COMMENTS IN PYTHON

- We use comments to explain the code, and the interpreter ignores them.
- You never know when a programmer may have to understand code written by another and make changes to it.
- Its also good to help yourself remember these details when you look at the code later.
- For these purposes, good code will hold comments in the right places.
- To declare a Python comment, use the hash (#) sign.

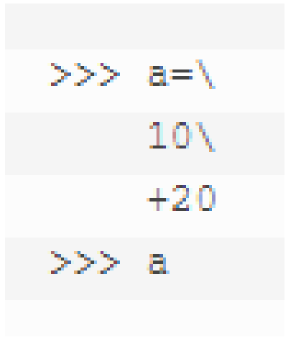


```
>>> #Line 1 of comment
>>> #Line 2 of comment
>>> #Line 3 of comment
```

MULTI-LINE PYTHON STATEMENTS

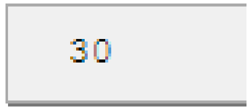
KHWARIZMI SCIENCE SOCIETY

- In Python, every statement ends with a newline character.
- But it is possible to span a statement over multiple lines.
- We do this using the '\' character.



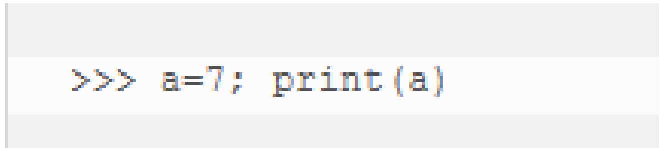
```
>>> a=\
    10\
    +20
>>> a
```

Output



```
30
```

- You can also easily put multiple statements in python on one line.



```
>>> a=7; print(a)
```

iDRACK

Some Exercises

KHWARIZMI
SCIENCE SOCIETY

1. Assign a message to a variable, and then print it.
2. Use a variable to represent a person's name, and print a message to that person. Something like, "Hi Ayesha, how are you feeling today?"
3. Find a famous quote. Print the quote with the name of the person.
4. Apply any 4 of the string functions you've learnt today on your name.
5. Add atleast 2 lines of comment to your code. Your name and current date, and what your code does.

Take a screenshot of your code and output for
record. We will be reviewing this in the next session

iDRACK