



Lecture # 2.3

Variables, Number Data Type and Operators

**DATA
SCIENCE**





Today's Agenda

- Variables in Python
- Overview of Python Data Types
- Numeric Data Types
- Operators
 - Arithmetic & Assignment operators
 - Comparison, logical and bitwise operators
 - Identity and Membership Operator
 - Operator Precedence
- Python print() Function





Variables



Variable in Python

- A statement in Python is something that results in an action or execution of a command. An expression, however, always results in a value
- A variable in Python is a reference rather a container (More on notebook)
- Python is dynamically typed (carries out type-checking at runtime.), which means you don't have to associate a type with a variable name
- So we don't have to explicitly declare a variable, rather the variable is created in the same statement, where we assign an object to the variable

```
apple = 200
oranges = 150
banana = 70
total = apple + oranges + banana
print(total)
```



200



150



70



Variable Naming Rules

- An identifier/variable name may only begin with A-Z, a-z, or an underscore(_).
- This may be followed by letters, digits, and underscores- zero or more.
- A number/digit can appear in the name, but not at the beginning.
- Python is case-sensitive. Name and name are two different identifiers.
- Spaces are not allowed. Instead, we must use snake_case to make variable names readable.
- The name of the variable should be something meaningful that describes the value it holds, instead of being random characters.
- Python Keywords:

`and, def, False, import, not, True, as, del, finally, in, or, try, assert, elif, for, is, pass, while, break, else, from, lambda, print, with, class, except, global, None, raise, yield, continue, exec, if, nonlocal, return`



Three Properties Associated with every Variable

- **Identity:** It can be considered as the address of the memory where the object is stored. An object's identity never changes after it is created. In Python the `id()` function returns the type of an object.
- **Type:** An object's type defines the possible values and operations that the type support. For example the integer type in Python support arithmetic operations, while string type support concatenation operation. An object's type never changes after it is created. In Python the `type()` function returns the type of an object.
- **Value:** It is the actual data stored in the object on which we perform various operations.



Variables in Python

<https://github.com/arifpucit/data-science.git>



[*View/Download from Jupyter nbviewer*](#)



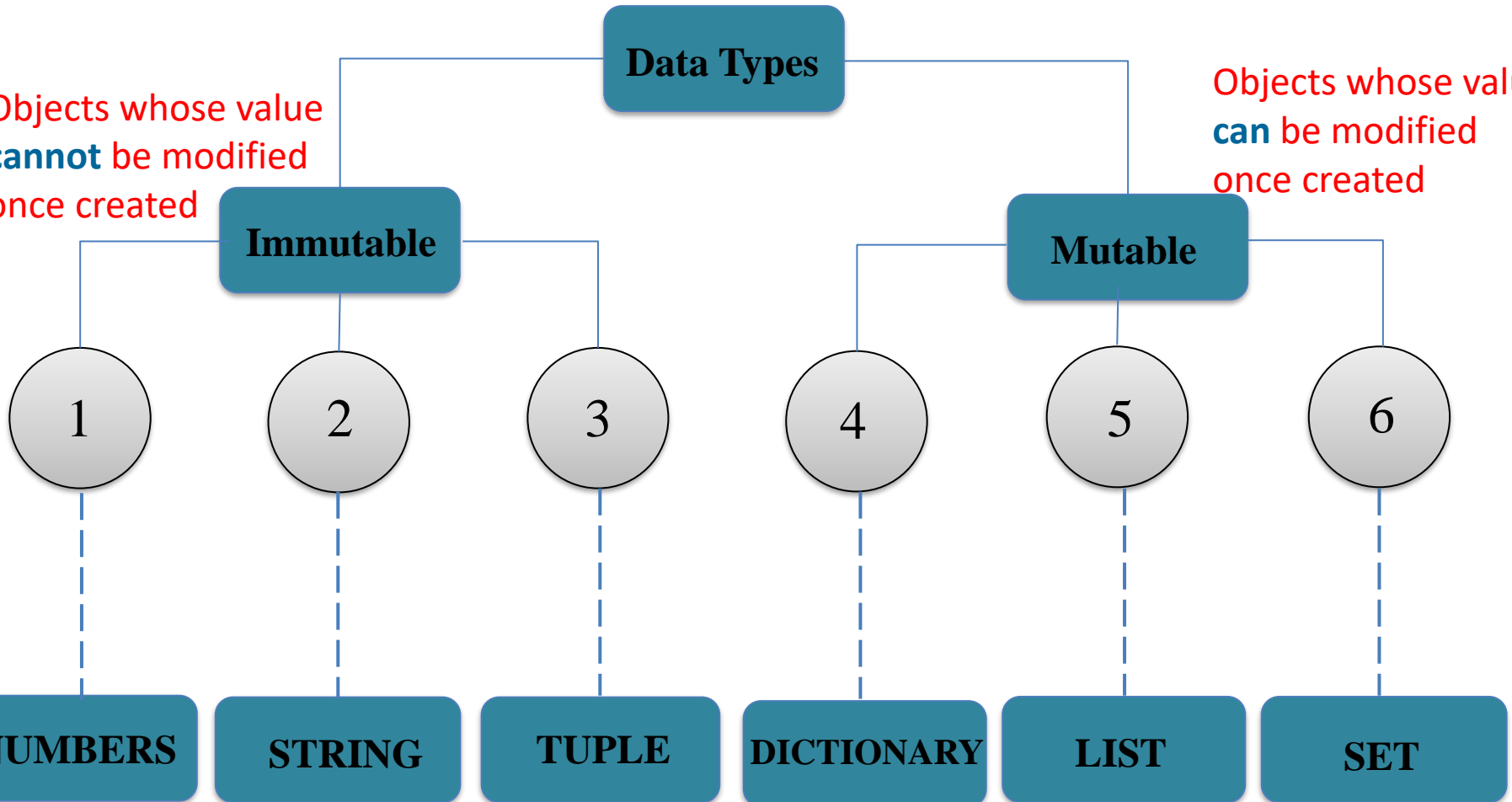
Overview of Data Types



Overview of Data Types

Objects whose value **cannot** be modified once created

Objects whose value **can** be modified once created



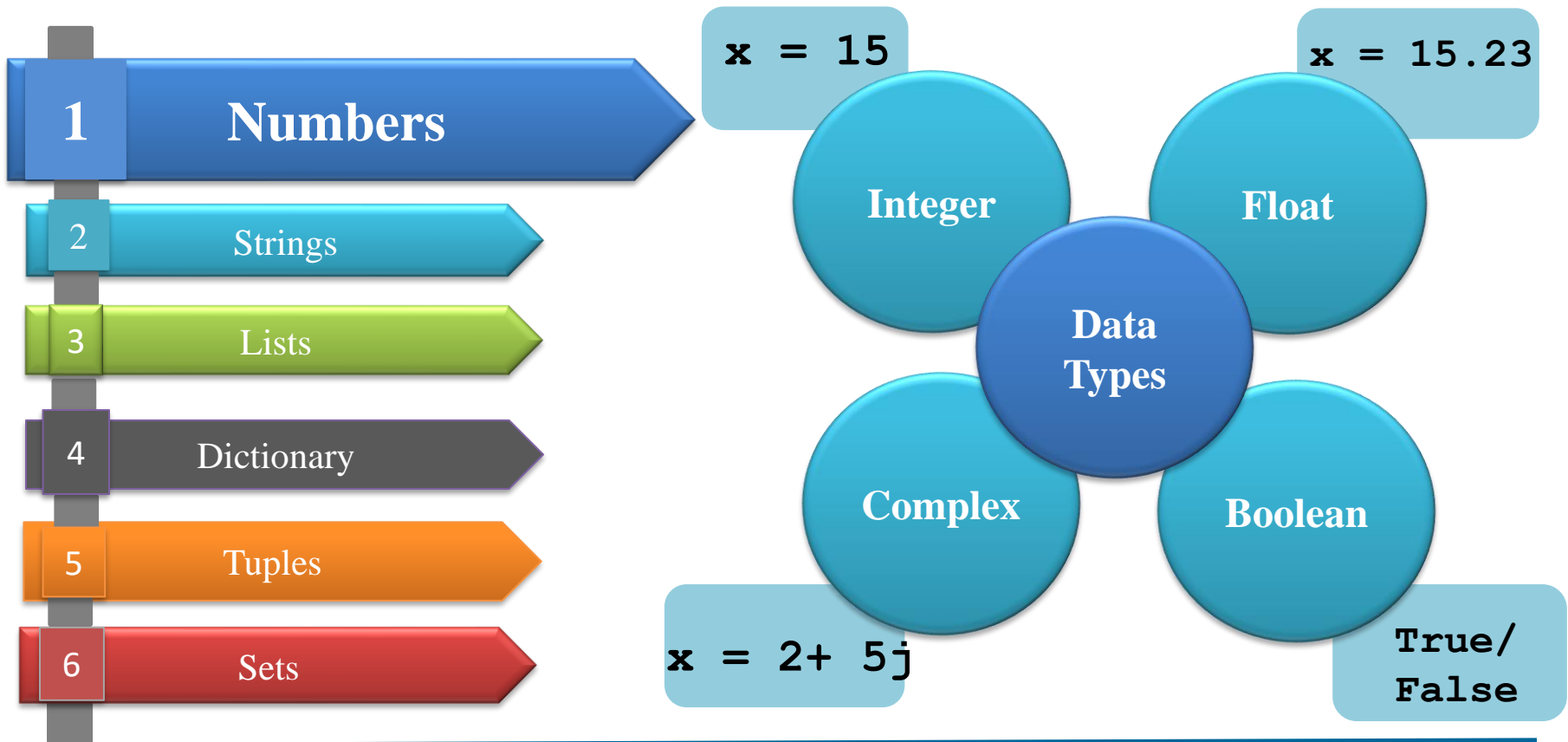


Numeric Data Types



Numeric Data Type or Numbers

- Numeric data types are a broad classification of objects in Python. There are three main types of numeric data types (Integer, Float, Complex). Some text also consider Boolean objects as part of Numeric data type





Numeric Data type

<https://github.com/arifpucit/data-science.git>



[*View/Download from Jupyter nbviewer*](#)



Overview of Operators



Operators





Arithmetic Operators

1 Arithmetic Operators

2 Assignment Operators

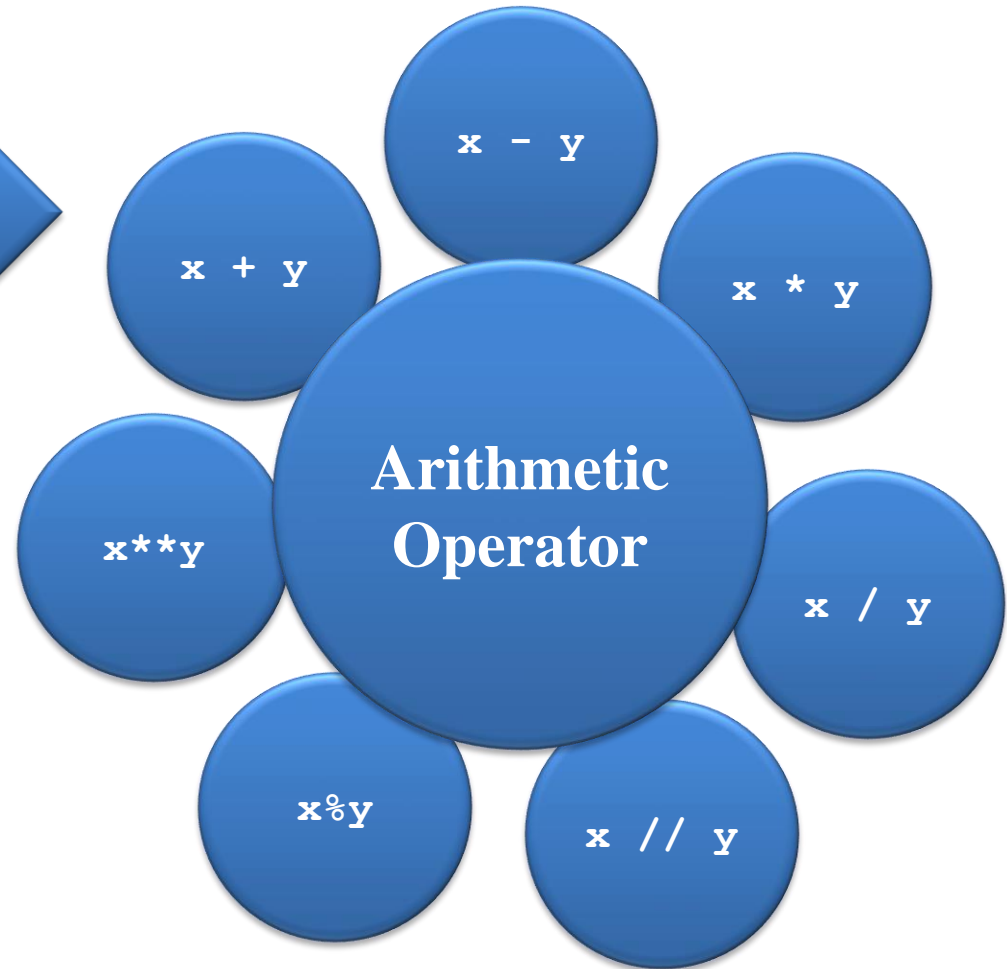
3 Comparison Operators

4 Logical Operators

5 Bitwise Operators

6 Identity Operators

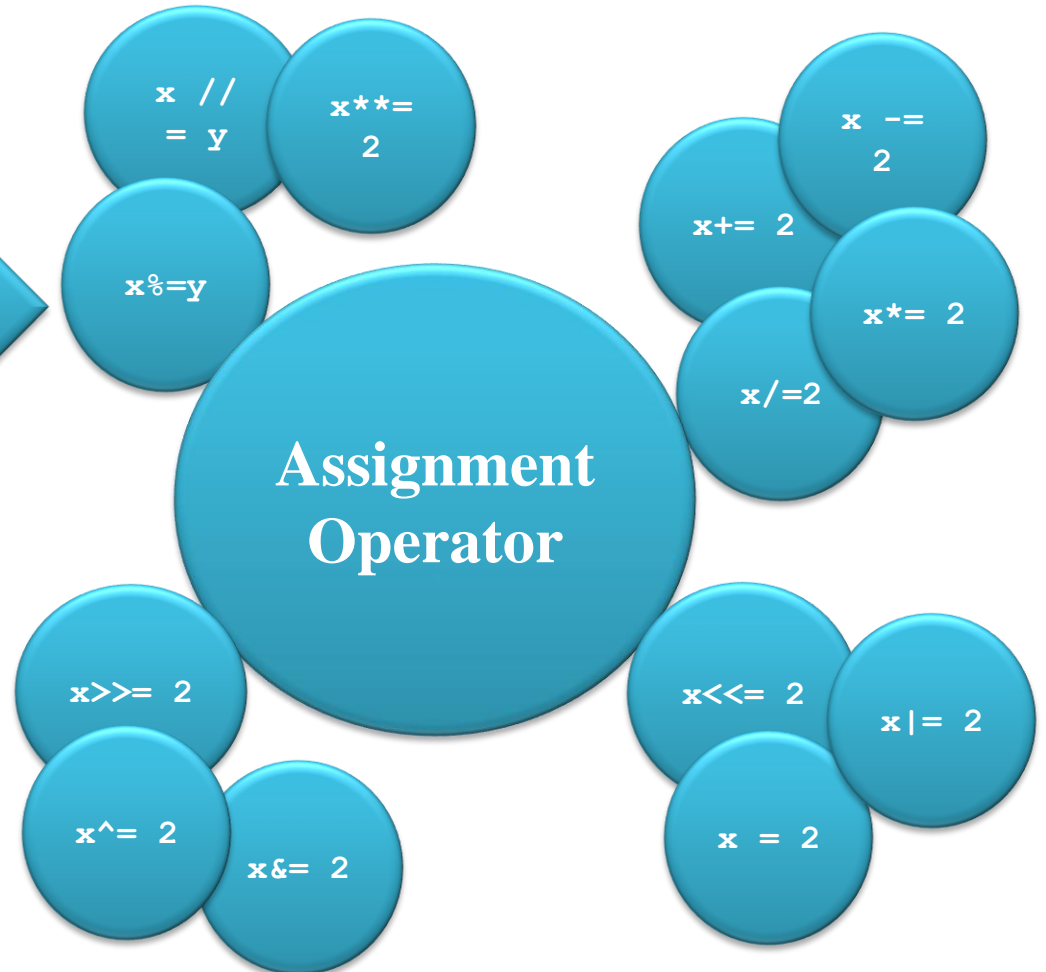
7 Membership Operators





Assignment Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators
- 4 Logical Operators
- 5 Bitwise Operators
- 6 Identity Operators
- 7 Membership Operators





Comparison Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators**
- 4 Logical Operators
- 5 Bitwise Operators
- 6 Identity Operators
- 7 Membership Operators

$x > y$
True if x is greater than y

$x < y$
True if x is less than y

$x == y$
True if both are equal

$x != y$
True if both are not equal

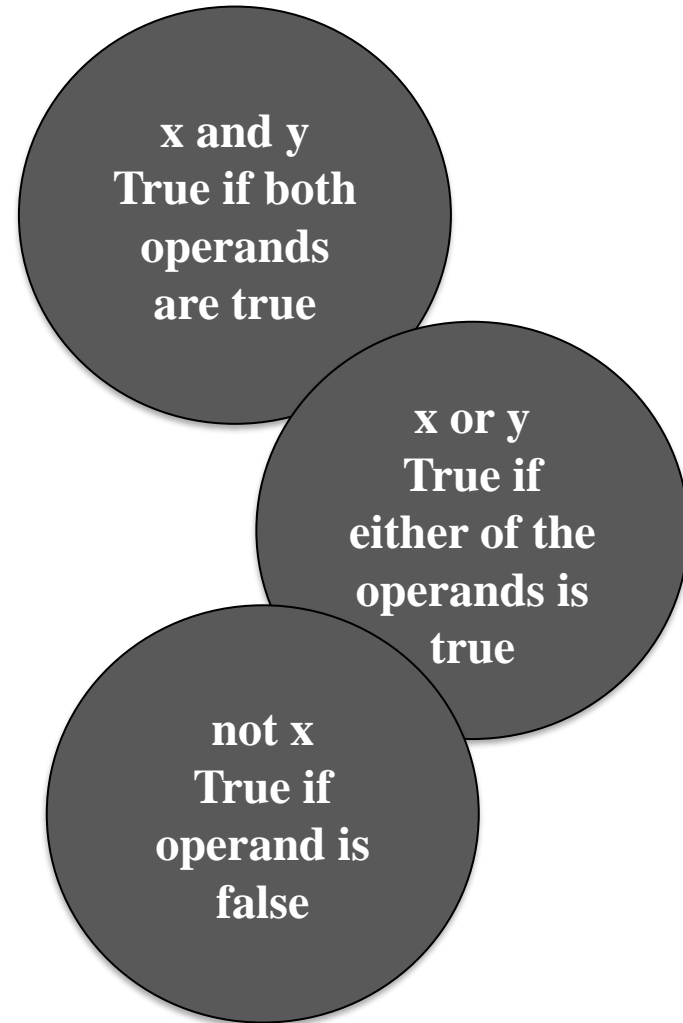
$x >= y$
True if x is greater than or to equal to y

$x <= y$
True if x is less than or to equal to y



Logical Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators
- 4 Logical Operators**
- 5 Bitwise Operators
- 6 Identity Operators
- 7 Membership Operators





Bitwise Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators
- 4 Logical Operators
- 5 Bitwise Operators**
- 6 Identity Operators
- 7 Membership Operators

$x \& y$
Bitwise AND

$x | y$
Bitwise OR

$\sim x$
Bitwise NOT

$x \wedge y$
Bitwise XOR

$x \ll y$
Bitwise left
shift

$x \gg y$
Bitwise right
shift



Identity Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators
- 4 Logical Operators
- 5 Bitwise Operators
- 6 Identity Operators**
- 7 Membership Operators

“Is”
True if
operands
are identical

“Is not”
True if
operands
are not
identical



Membership Operators

- 1 Arithmetic Operators
- 2 Assignment Operators
- 3 Comparison Operators
- 4 Logical Operators
- 5 Bitwise Operators
- 6 Identity Operators
- 7 **Membership Operators**

“In”
True if
value is
found in
sequence

“not in”
True if
value is not
found in
sequence



Operators in Python

<https://github.com/arifpucit/data-science.git>



[View/Download from Jupyter nbviewer](#)



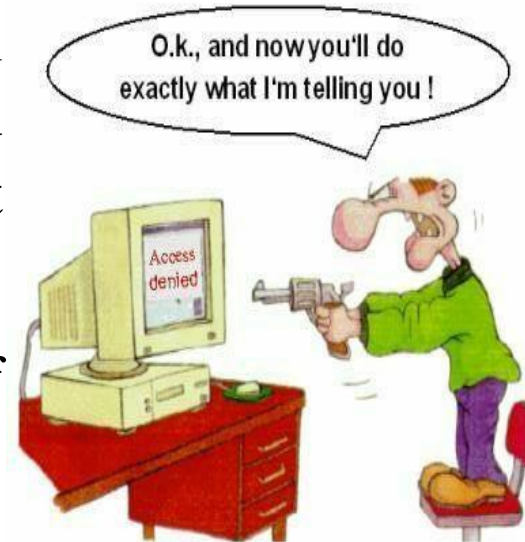
Operator Precedence

Operator	
()	Parentheses
**	Exponent
~ + -	Bitwise NOT, Unary plus, Unary minus
* / % //	Multiplication, Division, Modulus Floor division
+ -	Addition, Subtraction
>> <<	Bitwise shift operators
&	Bitwise AND
^	Bitwise XOR, Bitwise OR
<= >=	Comparisons Operator
== !=	Comparisons Operator
= %= /= //= -= += *= **=	Comparisons Operator
is is not	Identity
in not in	Membership operators
OR NOT AND	Logical Operator



Things To Do

- Play around with the basic data types of Python discussed in class. Do check out the mentioned links in the practice notebooks files to learn at your own pace.
- Practice code to have a clear understanding of different operators discussed in class
- This practice will of course help you in later part of the course



Coming to office hours does NOT mean you are academically weak!