Academy Xi

Data Analytics Elevate

DATA ANALYTICS

DAE2-E

Week 1 : Topic 1: Python Essentials
Session 1: 30th August

Welcome Everyone!

- You can download these slides from the shared Student folder on Google Drive:
  https://drive.google.com/drive/folders/15NLyoIDW-EObb5SPow7GLQsqQo0Kx3oL?usp=sharing

- The link to join each Zoom meeting on Monday and Wednesday evening is:

  https://academyxi.zoom.us/j/89382521382

  Every live session will be recorded on Zoom and uploaded to the Student Folder under 'Live Sessions'.

- Pre-reading: each week please complete the readings or labs before class those marked with ☆

- 80% Student Attendance

- Housekeeping rules:

  - Please engage in the live session and turn on your camera so that we can see your friendly face ☺
  - Turn your mobile phone to silent
  - Please turn your mic on mute and use chat unless you are speaking

Academy Xi

AGENDA

1. Introduction
2. The Data Science Process
3. Jupyter Notebook
4. Variable Assignment
5. Data Types Part I:

a) **Numbers:**
   - Integers
   - Floating-point numbers

b) **Booleans**

c) **Sequence:**
   - Strings
   - Lists
   - Tuple

Introductions

- Task: Introduce yourself to the class
  - Your Name
  - 2 interesting facts about you
  - Why are you here?
  - What would you like to learn?

Course Outcomes

- **2 Projects** –Build your portfolio in Python and Data Modelling
- **An industry credential**
- Get to know your peers who will be valuable to your new career !
- Develop competencies in:
  - ✓ Python for Data Analysis
  - ✓ SQL
  - ✓ APIs
  - ✓ Webscraping
  - ✓ Statistics
  - ✓ Linear Regression
  - ✓ Multiple Linear Regression and Evaluating a model

Mining for Gold: Analytics

"… helps clients identify and capture the most value and meaningful insights from data, and turn them into competitive advantages." – McKinsey Analytics

We deliver insight and impact for clients through a wide range of flexible support models, providing ad hoc, deeply transformational, and ongoing analytics architecture and solutions.

Importantly, we are fully technology agnostic and able to work with our clients' preferred technologies and platforms.

Source: https://www.mckinsey.com/business-functions/mckinsey-analytics/how-we-help-clients

Academy Xi

Analytics: begins with a problem
CRISP-DM

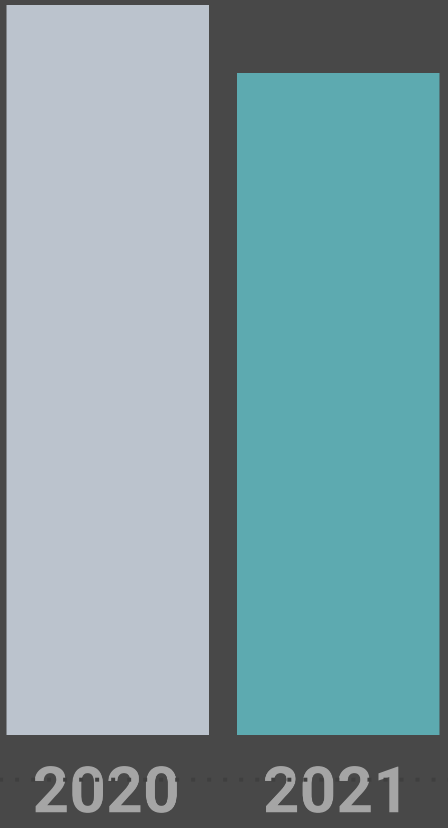
Step 1: Define the business problem

- Gather requirements from your boss, business stakeholders, SMEs, client
- Develop a series of additional business questions
- Define the scope of your problem
- Turn your business problem into an analytical task
- Collaborate: with peers, workshop with your clients

Data Science workflow: Steps to solve the problem

Cross-Industry Standard process for data mining

1: Business Understanding	define the business problem
2: Data Understanding	obtain quality data, get more data
3: Data Preparation	clean data, statistics, missing values
4: Modeling	build a model
5: Evaluation	which metrics to evaluate the model?
6: Deployment	share Python scripts, jupyter notebooks



Source: https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

Why learn Python programming language?

- General purpose programming language
- https://www.python.org/
- Manipulating, cleaning and crunching data

What is Python?

- Python is an open-source
- Python is free
- Python was created by Guido Van Rossum in 1991
- Object-oriented language can execute code that includes data attributes

Benefits of Python

- Easy to learn coding language
- It's fast
- Kids can learn it
- Readable code that can be understood when working in large project teams
- Easy to write code for production

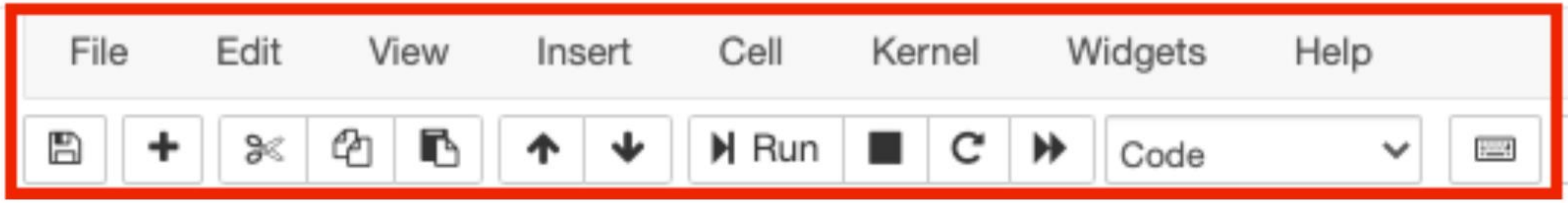
Why is Python popular?

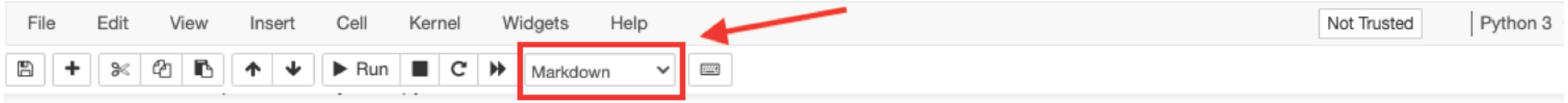
- Webscraping
- Data Visualization
- Data Analysing
- Deploying apps for machine learning
- Artificial Intelligence
- Building recommendation engines

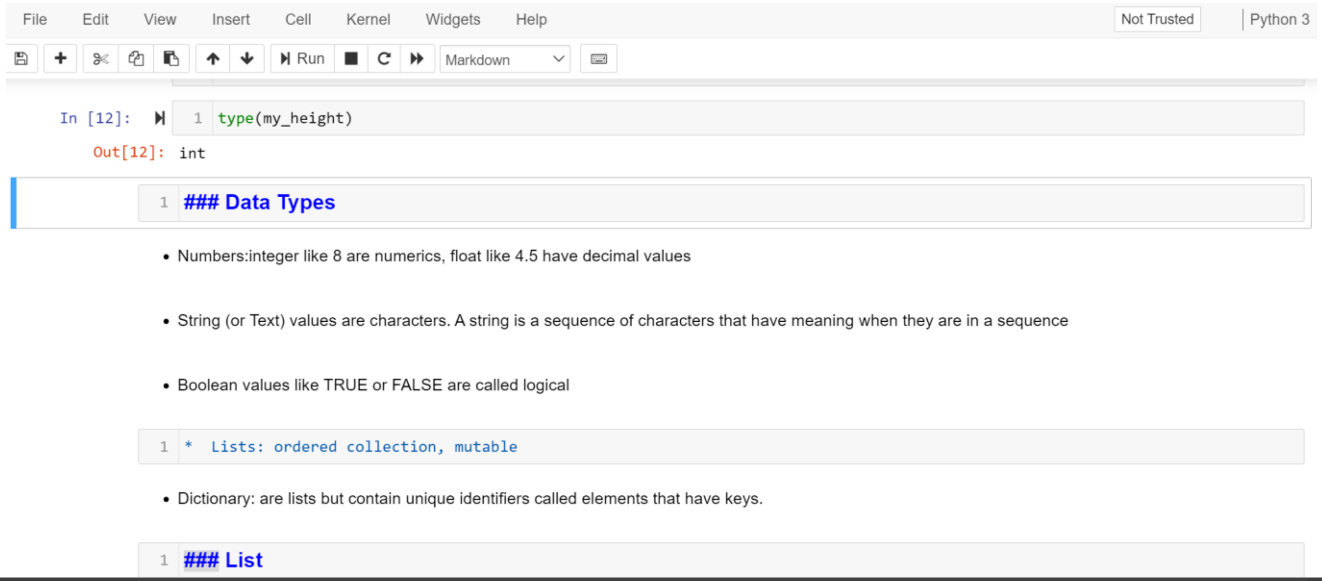
Who uses Python?

- Data Analysts
- Data Scientists
- Machine Learning Practitioners
- Software Engineers
- Roboticists
- Developers
- Researchers
- Consultants
- Programmers

Jupyter Notebook







```
In [12]:  type(my_height)
Out[12]:  int
```

```
### Data Types
```

- Numbers:integer like 8 are numerics, float like 4.5 have decimal values
- String (or Text) values are characters. A string is a sequence of characters that have meaning when they are in a sequence
- Boolean values like TRUE or FALSE are called logical

```
*  Lists: ordered collection, mutable
```

- Dictionary: are lists but contain unique identifiers called elements that have keys.

```
### List
```

# Header
## Header
### Header
#### Header

*** Bullet Point**

- https://jupyter.org/

Academy Xi

Python Data Types

Built-In Data Types:

Numbers

Booleans (TRUE, FALSE)

Sequence (String, Lists, Tuple)

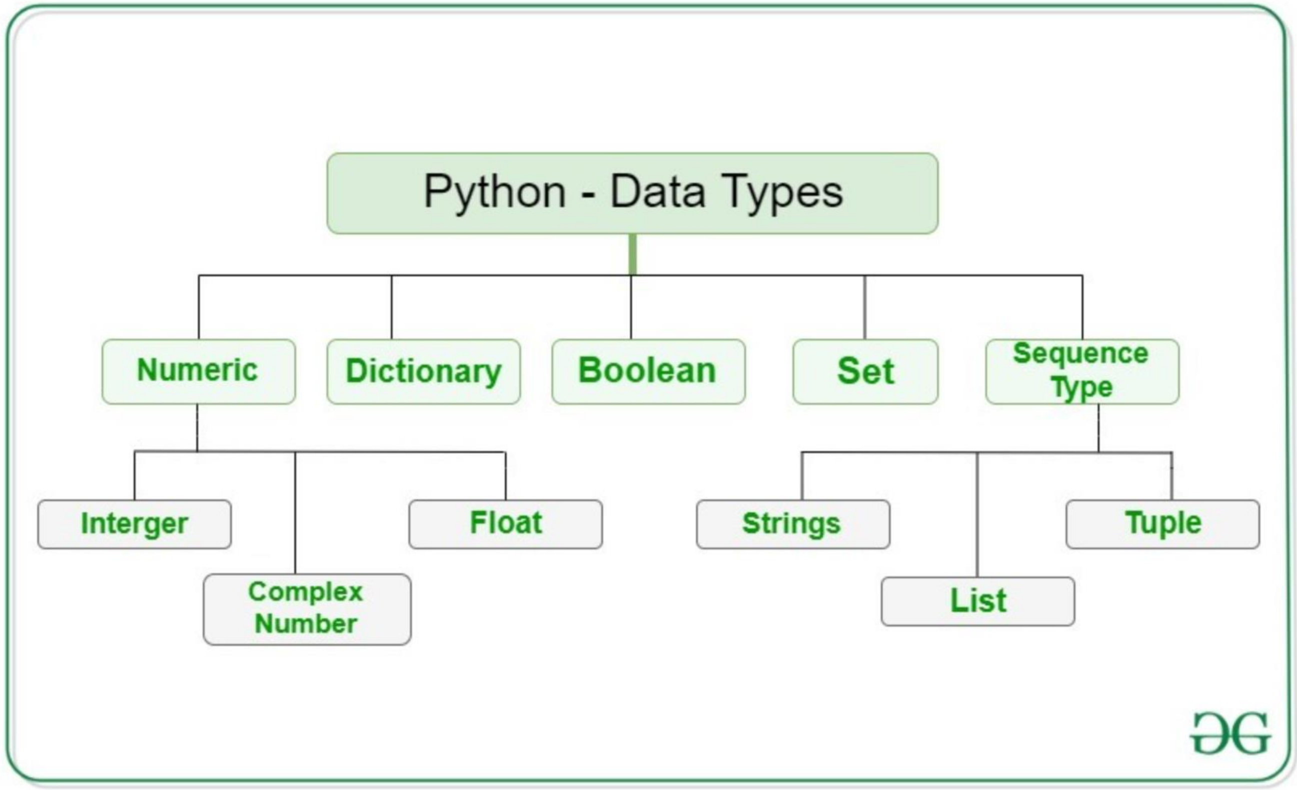
Set

Dictionary

Academy Xi

Data Types

- Variables can store different data types
- Data types are classes of data items
- Variables are objects of these classes
- Everything is an **object** in Python programming
- Check the data type with the function: type (variable_name)

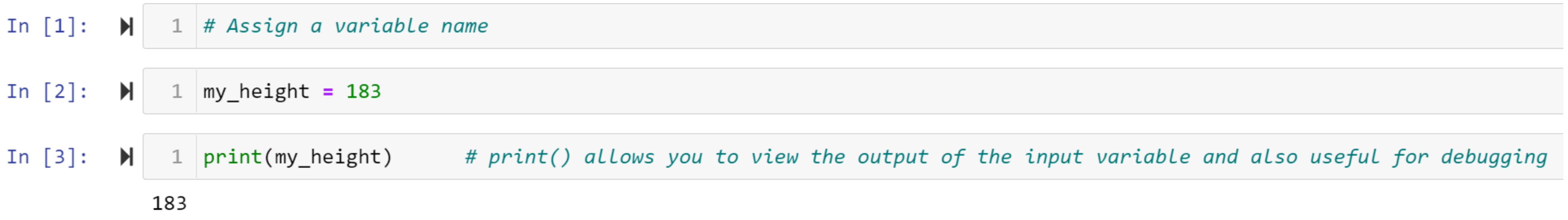


https://www.geeksforgeeks.org/python-data-types/

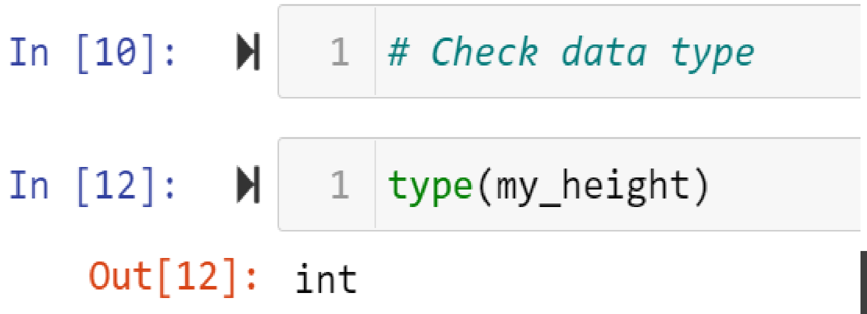
Variable Assignment

- Python variable names are **case-sensitive**
- Variable names can only use letters, underscore and numbers
- Variable name start with a letter or an underscore
- Variable names <u>cannot</u> use Python's **reserved words** (https://realpython.com/lessons/reserved-keywords/)

Left hand-side assigns the variable to the Right hand-side. In Python = is variable assignment



```
In [1]:  # Assign a variable name

In [2]:  my_height = 183

In [3]:  print(my_height)        # print() allows you to view the output of the input variable and also useful for debugging

183
```



```
In [10]:  # Check data type

In [12]:  type(my_height)

Out[12]:  int
```

Function: type ()

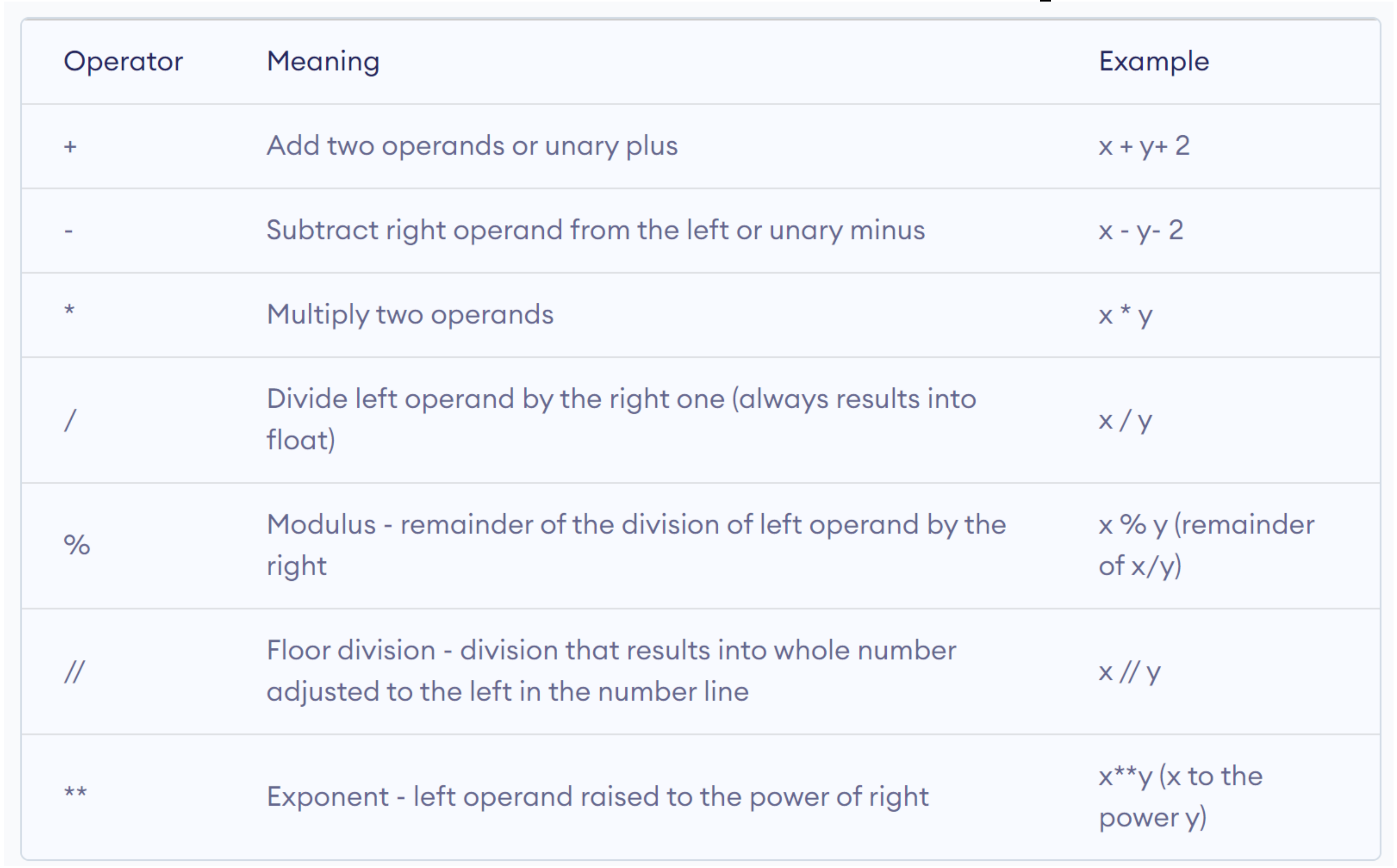
Data Type: Numeric

- **Integer:** like 8 are numerics. They are positive or negative whole numbers
- **Float:** like 4.5 are decimal or fraction

```
x = 5                  # creates an object (i.e. variable)
type (5)               # int
type (5.0)             # float
```

```
c = 7e-5               # creates an object (i.e. variable)
c
print(“Type of c is: ”,type(c))

Type of c is : <class, ‘float’>
```

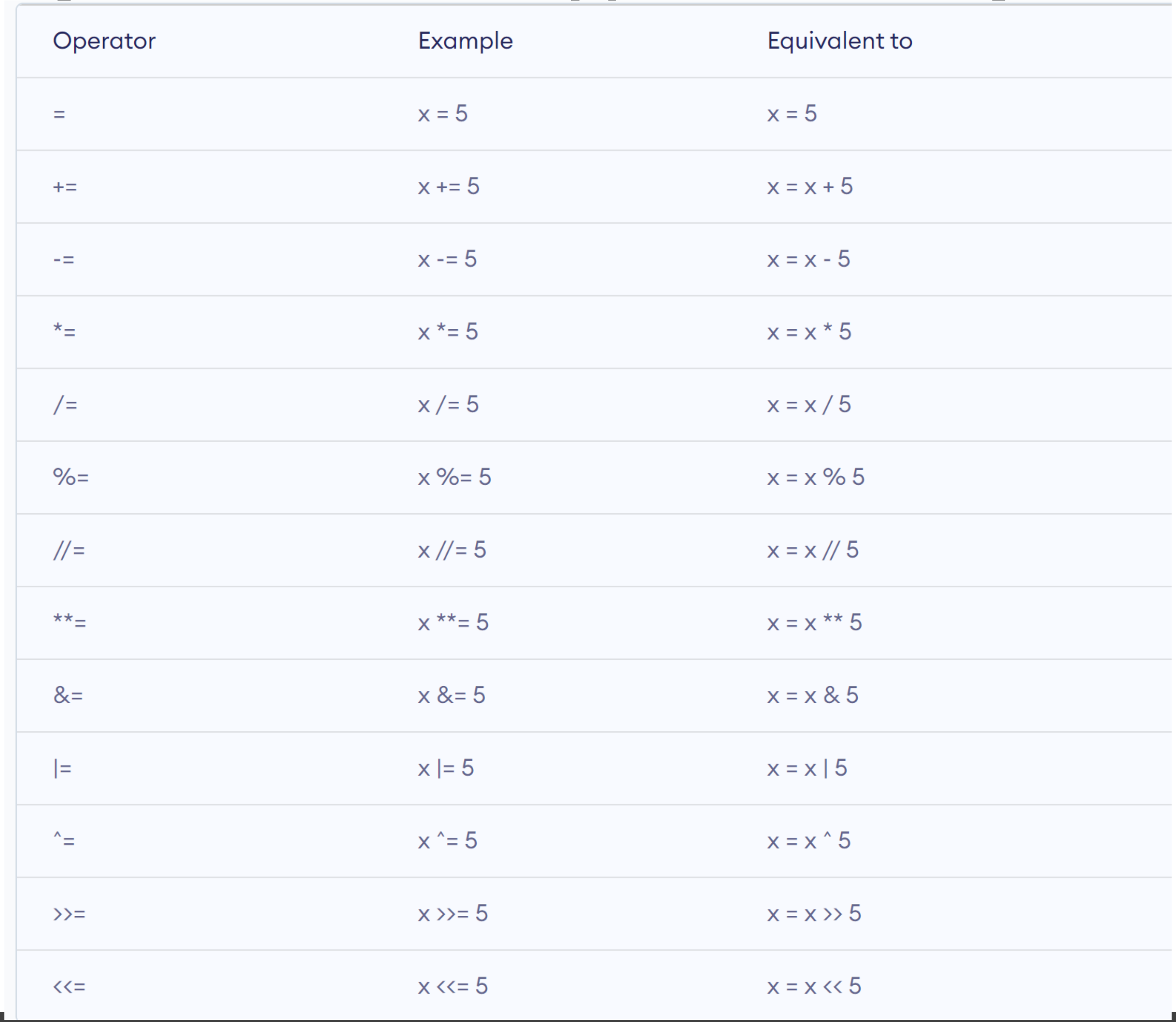
Calculator: Arithmetic Operators



Operator	Meaning	Example
+	Add two operands or unary plus	x + y+ 2
-	Subtract right operand from the left or unary minus	x - y- 2
*	Multiply two operands	x * y
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Source: https://www.programiz.com/python-programming/operators

Important: Assignment Operators



Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5
<<=	x <<= 5	x = x << 5

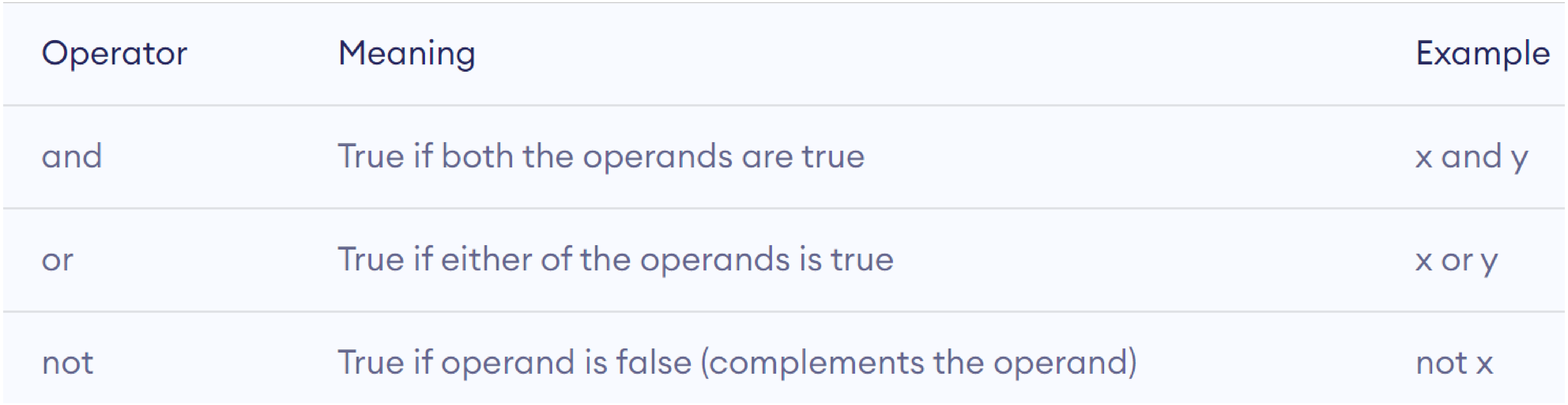
Source: https://www.programiz.com/python-programming/operators

Data Type: Booleans

- Boolean values like TRUE or FALSE are called logical
- or
- and
- not
- == (equivalent)

```
 x = 5                                  # creates an object (i.e. variable)
type (TRUE)                             # bool

profitable = TRUE



# make a list of ingredients for your breakfast and dinner
breakfast_list = [ “milk”, “ scrambled egg”, “toast”, “cereal”, “coffee”, “yoghurt”, “french toast”,
“toast”]
dinner_list = [“steak”, “broccoli”, “pasta”, “salmon”]

# find out if milk is in the breakfast or dinner list
print(“milk” in breakfast_list or dinner_list)
```

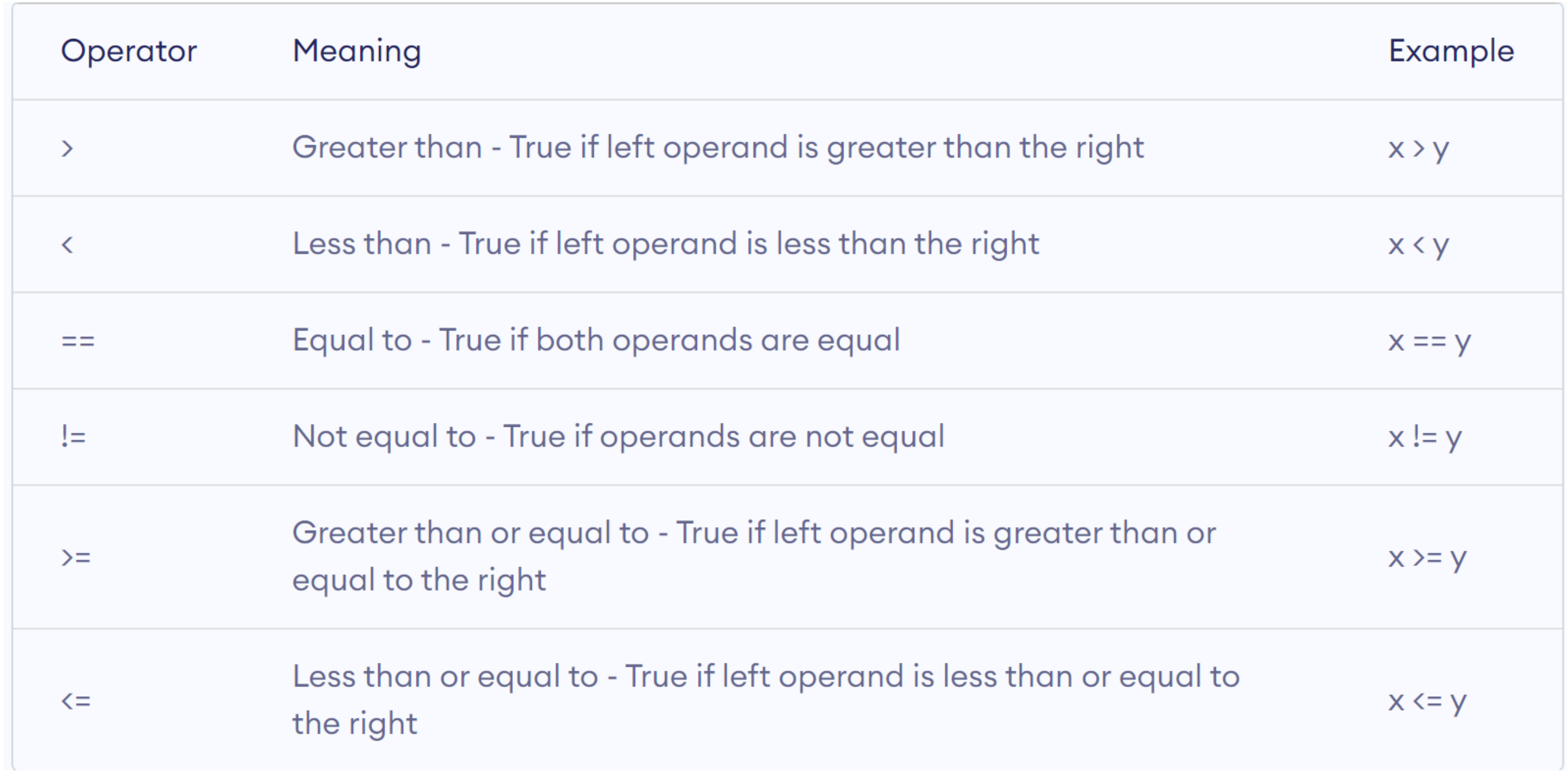
Logical Operators: AND, OR, NOT



Operator	Meaning	Example
and	True if both the operands are true	x and y
or	True if either of the operands is true	x or y
not	True if operand is false (complements the operand)	not x

Source: https://www.programiz.com/python-programming/operators

Comparison Operators



Operator	Meaning	Example
>	Greater than - True if left operand is greater than the right	x > y
<	Less than - True if left operand is less than the right	x < y
==	Equal to - True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to - True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to - True if left operand is less than or equal to the right	x <= y

Source: https://www.programiz.com/python-programming/operators

Data Type: Sequence

Sequence:

- **Ordered collection** of similar or different data types in Python
- Sequences allows to store multiple values in an organized and efficient fashion
- Sequence in Python:
    - String
    - List
    - Tuple

Data Type – Sequence - String

- A string (or text) is a collection of one or more characters put in:
  - Single quote ‘ ‘, Double quote “ “
- Str class

```
# Create a string with double quotes
String1 = “ Welcome to Data Analytics Elevate”
print(“ \nString with the use of Double Quotes: “)
print(String1)
print(type(String2))
```

```
String with the use of Double Quotes:
Welcome to Data Analytics Elevate
<class,’str’>
```

```
# Printing First character
print(“\nFirst character of String is:”, String1[0])
```

```
First character of String is:  W
```

```
# Printing Last character
print(“\nLast character of String is:”, String1[-1])
```

```
Last character of String is:  e
```

Data Type – Sequence - Lists [ ]

- **Ordered collection** of objects in Python
- **Mutable** or flexible
- Lists can be altered after they are created
- Lists can include **different data types** such as numbers, strings, other lists, tuples
- In lists you access individual values with consecutive integers calls **indices** or **index values**
- **List methods** append, remove

```
# Create a list
List = [“ Academy”, “Xi”, “Data”, “Analytics”]

# Accessing elements of a list with negative indexing
# Print the last element of the list
print(List[-1])

Analytics

# Accessing elements of a list
# Print third last element of the list
print(List[-3])

Xi
```

Data Type – Sequence - Tuple

- **Ordered collection** of objects in Python
- Tuples are similar to lists except the elements inside the list **cannot be modified** (i.e. add or removed)
- **Immutable**
- Class is tuple
- Tuples are enclosed in parentheses

Example: thislist = (“python”, “r”, “sql)

Data Type - Set

- A collection of string or integer with no specific order or index, like a list
- Automatically sorts values in alphabetical order

```
# Obtain unique values from breakfast list
breakfast_list = [ “milk”, “ scrambled egg”, “toast”, “cereal”, “coffee”, “yoghurt”, “french toast”, “toast”]
print(set(breakfast_list))

[ “milk”, “ scrambled egg”, “toast”, “cereal”, “coffee”, “yoghurt”, “french toast”]

 len(s)
```

Data Type - Dictionary { }

- Dictionaries combine **keys** with **values** in paired with a unique identifier
- Sequence data is stored in **unordered** manner
- A dictionary itself is mutable, but each of its individual keys are **immutable**
- Like in a dictionary, you can search the keys to obtain their corresponding value
- In dictionaries you access individual values using integers, strings or other python objects called keys

Example: e-commerce data

```
gov_dict = {"dine & discover": 1.0, "quarantine": 2.0, "business grant":3.0}
print(type(gov_dict))
print(gov_dict)
```

Example: covid exposure areas

```
epo_dict = {‘canterbury-bankstown": 25.0, “liverpool": 30.0, “bondi":55.0}
print(type(epo_dict ))
print(epo_dict)

bondi_expo = epo_dict[“bondi”]
```

Data Type Conversion

- Variables to booleans

bool()

- Variables to string

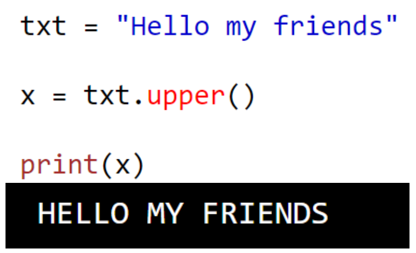
str()

Variables to integers

int()

Changing Data with Python Built In String Methods

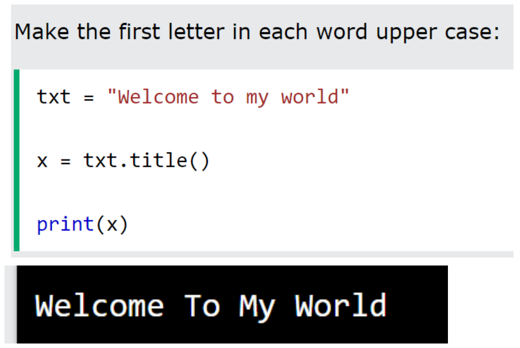
- .upper()



```
txt = "Hello my friends"

x = txt.upper()

print(x)
```

```
HELLO MY FRIENDS
```

- .title()

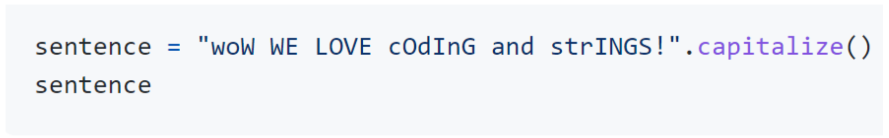


Make the first letter in each word upper case:

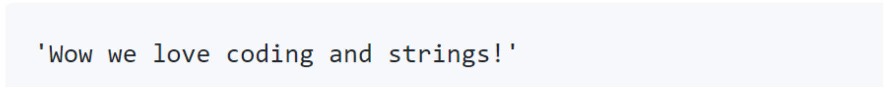
```
txt = "Welcome to my world"

x = txt.title()

print(x)
```

```
Welcome To My World
```

- .capitalize()



```
sentence = "woW WE LOVE cOdInG and strINGS!".capitalize()
sentence
```



```
'Wow we love coding and strings!'
```

https://www.w3schools.com/python/python_ref_string.asp

String Concatenation

a = String1
b = String2

a + b = String1 + String2

Quiz 1: Variable Names and Data Types

- Are Python variable names case sensitive?
- Can you have spaces in variable names?
- Is a dictionary an ordered sequence?
- Does a dictionary use square bracket or curly braces?
- Is a float a text or number data type?

Homework – Week 1 Labs

- Introduction to Variables: Variable Assignment – Lab
- Introduction to Variables: Strings - Lab
- Working with Lists – Lab
- Working with Dictionaries – Lab
- Built-in Python Operators, Functions and Methods – Lab
- Control Flow: Conditionals - Lab

Reading:

- The Data Science Process
- Jupyter Notebook

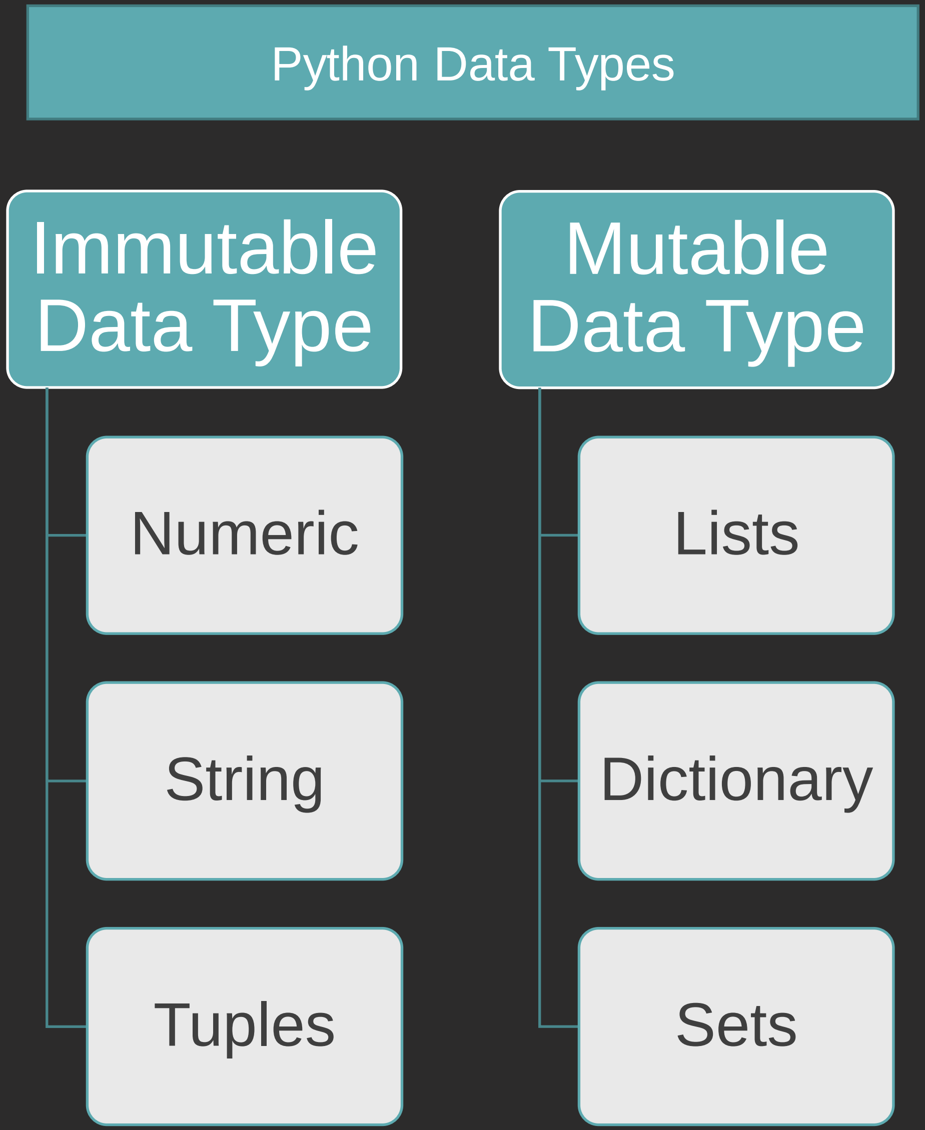
Re-Cap

Data Types Part 1:

- Numerics
- Booleans
- Sequence (String, List, Tuple)
- Dictionary

Session 2: Data Types Part 2

- Conditionals
- Built-In Python Operators, Functions and Methods



Session 2: Python Essentials – Data Types Part 2

- Re-cap: Python Essentials Session 1
- Control Flow
- Built-in Python Operators, Functions and Methods
- Project 1 in Week 6
- Mentoring: You may book mentoring sessions in my calendar from Week 1:

  https://calendly.com/da_mentor2

Thank You

Stay Connected

Join Slack channel: da-e-2.slack.com

# general
# questions
# random

Academy Xi