Python for Data Science, AI & Development

Table of Contents

[MODULE 1 Python Basics 3](#_Toc180778063)

[First program 3](#_Toc180778064)

[Types 3](#_Toc180778065)

[Expressions and Variables 3](#_Toc180778066)

[Format Strings in Python 3](#_Toc180778067)

[String Operations 3](#_Toc180778068)

[Module 1 Summary: Python Basics 3](#_Toc180778069)

[Cheat Sheet: Python Basics 4](#_Toc180778070)

[Module 1 Glossary: Python Basics 4](#_Toc180778071)

[MODULE 2 Python Data Structures 4](#_Toc180778072)

[Tuples 4](#_Toc180778073)

[Tuples 4](#_Toc180778074)

[Cheat Sheet: Lists and Tuples 4](#_Toc180778075)

[Dictionaries 5](#_Toc180778076)

[Sets 5](#_Toc180778077)

[Module 2 Summary: Python Data Structures 5](#_Toc180778078)

[CheatSheet: Dictionaries & Sets 6](#_Toc180778079)

[Module 2 Glossary: Python Data Structures 6](#_Toc180778080)

[MODULE 3 Python Programming Fundamentals 6](#_Toc180778081)

[Conditions and Branching 6](#_Toc180778082)

[Conditions and Branching 6](#_Toc180778083)

[Introduction to Loops in Python 6](#_Toc180778084)

[Loops 6](#_Toc180778085)

[Functions 6](#_Toc180778086)

[Functions 7](#_Toc180778087)

[Exception Handling 7](#_Toc180778088)

[Exception Handling 7](#_Toc180778089)

[Objects and Classes 7](#_Toc180778090)

[Objects and Classes 7](#_Toc180778091)

[Practise lab 7](#_Toc180778092)

[Module 3 summary 7](#_Toc180778093)

[Cheat sheet: Python programming fundamentals 9](#_Toc180778094)

[Module 3 glossary: Python programming fundamentals 9](#_Toc180778095)

[MODULE 4 Working with Data in Python 10](#_Toc180778096)

[Reading files with open 10](#_Toc180778097)

[Reading files with open 10](#_Toc180778098)

[Writing files with open 11](#_Toc180778099)

[Writing files with open 11](#_Toc180778100)

[Pandas 11](#_Toc180778101)

[Selecting Data in a Data Frame 11](#_Toc180778102)

[Loading Data with Pandas 11](#_Toc180778103)

[One Dimensional NumPy 11](#_Toc180778104)

[NumPy Matrix Mathematics 11](#_Toc180778105)

[Two Dimensional NumPy 12](#_Toc180778106)

[Beginners Guide to NumPy 12](#_Toc180778107)

[Some contexts on API 12](#_Toc180778108)

[Module 4 Summary: Working with Data in Python 12](#_Toc180778109)

[Cheat Sheet: Working with Data in Python 13](#_Toc180778110)

[Module 4 Glossary: Working with Data in Python 13](#_Toc180778111)

[MODULE 5 API and Data Collection 13](#_Toc180778112)

[Introduction to API 13](#_Toc180778113)

[Practice Project: GDP Data Extraction and Processing 13](#_Toc180778114)

[Web Scraping and HTML Basics 14](#_Toc180778115)

[Access REST APIs & Request HTTP 14](#_Toc180778116)

[API Examples 14](#_Toc180778117)

[Web Scraping - A Key Tool in Data Science 14](#_Toc180778118)

[Web Scraping Tables using Pandas 14](#_Toc180778119)

[Web Scraping 14](#_Toc180778120)

[Working with different file formats 14](#_Toc180778121)

[Module 5 Summary: APIs and Data Collection 14](#_Toc180778122)

[Cheat Sheet: APIs and Data Collection 16](#_Toc180778123)

[Module 5 Glossary: APIs and Data Collection 16](#_Toc180778124)

[Python Cheat Sheet: The Basics 16](#_Toc180778125)

## **MODULE 1 Python Basics**

## First program

<https://cf-courses-data.static.labs.skills.network/_r1fp-EZBXapbhESR3kRKA.jupyterlite.ipynb.html>

## Types

<https://cf-courses-data.static.labs.skills.network/TxVoPcArWtWQobeKhTB61g.jupyterlite.ipynb.html>

## Expressions and Variables

<https://cf-courses-data.static.labs.skills.network/bVLEXIWjo5syKtT_pFOu-Q.jupyterlite.ipynb.html>

## Format Strings in Python

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/mod_1/format_Strings.md.html>

## String Operations

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/jupyterlite/latest/lab/index.html?notebook_url=https://cf-courses-data.static.labs.skills.network/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/jupyterlite/files/Module_1/PY0101EN-1-2-Strings-20230410-1681084800.jupyterlite.ipynb>

## Module 1 Summary: Python Basics

* Python can distinguish among data types such as integers, floats, strings, and Booleans.
* Integers are whole numbers that can be positive or negative.
* Floats include integers as well as decimal numbers between the integers.
* You can convert integers to floats using typecasting, but you cannot convert a float to an integer.
* You can convert integers and floats to strings.
* You can convert an integer or float value to True (1) or False (0).
* Expressions in Python are a combination of values and operations used to produce a single result.
* Expressions perform mathematical operations such as addition, subtraction, multiplication, and so on.
* We use"//" to round off integer divisions, resulting in float values.
* Python follows the order of operations (BODMASS) to perform operations with multiple expressions.
* Variables store and manipulate data, allowing you to access and modify values throughout your code.
* The assignment operator "=" assigns a value to a variable.
* ":" denotes the value of the variable within the code.
* Assigning another value to the same variable overrides the previous value of that variable.
* You can perform mathematical operations on variables using the same or different variables.
* While performing operations with various variables, modifying a value in one variable will lead to changes in the other variables.
* Python string operations involve manipulating text data using tasks such as indexing, concatenation, slicing, and formatting.
* A string is usually written within double quotes or single quotes, including letters, white space, digits, or special characters.
* A string attaches to another variable and is an ordered sequence of characters.
* Characters in a string identify their index numbers, which can be positive or negative.
* We use strings as a sequence to perform sequence operations.
* You can input a stride value to perform slicing while operating on a string.
* Operations like finding the length of the string, combining, concatenating, and replicating, result in a new string.
* You cannot modify an existing string; they are immutable.
* You can perform escape sequences using " " to change the layout of the string.
* In Python, you perform tasks such as searching, modifying, and formatting text data with its pre-built string methods functions.
* You apply a method to a string to change its value, resulting in another string.
* You can perform actions such as changing the case of characters in a string, replacing items in a string, finding items in a string, and so on using pre-built string methods.

## Cheat Sheet: Python Basics

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Cheat_Sheet_Week-1.md.html?origin=www.coursera.org>

## Module 1 Glossary: Python Basics

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/200336.032_Module_1_Glossary.md.html>

## **MODULE 2 Python Data Structures**

## Tuples

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%202/PY0101EN-2-2-Lists.ipynb.html>

## Tuples

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-9de47e2e959a41d0ec6b31fcd757db7aa7d0e33b>

## Cheat Sheet: Lists and Tuples

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/handouts/Cheat_Sheet_Week-2.md.html?origin=www.coursera.org>

## Dictionaries

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%202/PY0101EN-2-4-Dictionaries.ipynb.html>

## Sets

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-68e17f621537380de1da07f34d46af9f32b57a38>

## Module 2 Summary: Python Data Structures

* In Python, we often use tuples to group related data together.
* Tuples refer to ordered and immutable collections of elements.
* Tuples are usually written as comma-separated elements in parentheses “()".
* You can include strings, integers, and floats in tuples and access them using both positive and negative indices.
* You can perform operations such as combining, concatenating, and slicing on tuples.
* Tuples are immutable, so you need to create a new tuple to manipulate it.
* Tuples, termed nesting, can include other tuples of complex data types.
* You can access elements in a nested tuple through indexing.
* Lists in Python contain ordered collections of items that can hold elements of different types and are mutable, allowing for versatile data storage and manipulation.
* A list is an ordered sequence, represented with square brackets "[]".
* Lists possess mutability, rendering them akin to tuples.
* A list can contain strings, integers, and floats; you can nest lists within it.
* You can access each element in a list using both positive and negative indexing.
* Concatenating or appending a list will result in the modification of the same list.
* You can perform operations such as adding, deleting, splitting, and so forth on a list.
* You can separate elements in a list using delimiters.
* Aliasing occurs when multiple names refer to the same object.
* You can also clone a list to create another list.
* Dictionaries in Python are key-value pairs that provide a flexible way to store and retrieve data based on unique keys.
* Dictionaries consist of keys and values, both composed of string elements.
* You denote dictionaries using curly brackets.
* The keys necessitate immutability and uniqueness.
* The values may be either immutable or mutable, and they allow duplicates.
* You separate each key-value pair with a comma, and you can use color highlighting to make the key more visible.
* You can assign dictionaries to a variable.
* You use the key as an argument to retrieve the corresponding value.
* You can make additions and deletions to dictionaries.
* You can perform an operation on a dictionary to check the key, which results in a true or false output.
* You can apply methods to obtain a list of keys and values in a dictionary.
* Sets in Python are collections of unique elements, useful for tasks such as removing duplicates and performing set operations like union and intersection. Sets lack order.
* Curly brackets "{}" are helpful for defining elements of a set.
* Sets do not contain duplicate items.
* A list passed through the set function generates a set containing unique elements.
* You use “Set Operations” to perform actions such as adding, removing, and verifying elements in a set.
* You can combine sets using the ampersand "&" operator to obtain the common elements from both sets.
* You can use the Union function to combine two sets, including both the common and unique elements from both sets.
* The sub-set method is used to determine if two or more sets are subsets.

## CheatSheet: Dictionaries & Sets

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/handouts/Cheat_Sheet_Week-2_Part-2.md.html?origin=www.coursera.org>

## Module 2 Glossary: Python Data Structures

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/200336.011_Module_2_Glossary.md.html>

## **MODULE 3 Python Programming Fundamentals**

## Conditions and Branching

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-3f03b401aa0f6e13b1791c7f3e699ea5cb855d60>

## Conditions and Branching

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module_3/Conditional_branching_Reading.md.html>

## Introduction to Loops in Python

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/lab/Module_3/loop_reading.md.html?origin=www.coursera.org>

## Loops

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-efcb1af3d2aa577028b4b2b2e5dd33e3365eadbc>

## Functions

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module_3/function_reading.md.html?origin=www.coursera.org>

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%203/Python_reference_sheet.pdf?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0101ENSkillsNetwork19487395-2022-01-01>

## Functions

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-a5a8d8789b355898b21009c714e0fd4d92cb7b7d>

## Exception Handling

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Exception_handling_Reading.md.html?origin=www.coursera.org>

## Exception Handling

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-2fcc068a3becb301a4b5d1375d599209edbfd1dc>

## Objects and Classes

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/classes_and_object_Reading.md.html>

## Objects and Classes

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-f709c39cbfa87cdea1c2d2b730b8ae06f765f256>

## Practise lab

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-20b56826c7763dad482ffed3be40aff0dcb2b1ba>

## Module 3 summary

* Python conditions use “if” statements to execute code based on true/false conditions created by comparisons and Boolean expressions.
* Comparison operations require using comparison operators equal to "=", greater than ">", less than "<".
* An exclamation mark "!" is used to define inequalities of a variable.
* You can compare integers, strings, and floats.
* Python branching directs program flow by using conditional statements (for example, if, else, elif) to execute different code blocks based on conditions or tests.
* You can use the "if" statement with conditions to define actions if true.
* To perform actions based on true or false output, you can use the "else" statement with conditions.
* The elif statement allows for additional checks only if the initial condition is false.
* To execute various operations on Boolean values, we use Boolean logic operators.
* Python loops are control structures that automate repetitive tasks and iterate over data structures like lists or dictionaries.
* The range() function generates a sequence of numbers with a specified start, stop, and step value for loops in Python.
* A for loop in Python iterates over a sequence, such as a list, tuple, or string, and executes a block of code for each item in the sequence.
* A while loop in Python executes a block of code as long as a specified condition remains true.
* Python functions are reusable code blocks that perform specific tasks, take input parameters, and often return results, enhancing code modularity and reusability.
* You may or may not have written the codes that are often included in functions.
* Python has a set of built-in functions such as "len" to find the length of a sequence or "sum" to find the total sum of a sequence.
* The "sorted" function creates a new sorted list, while "sort" sorts items in the original list.
* You can also create your own functions in Python.
* To ensure clarity and organization and facilitate understanding and maintenance of the code, developers must document functions using a documentation string enclosed in three quotes.
* The help command will return the documentation defined for a particular function.
* A function can have multiple parameters.
* “No return” statement in the function means that the function will return nothing.
* The "No work" function does not execute any task. You can use the "pass" keyword to meet the requirement of a non-empty body.
* A function will usually perform more than one task.
* In Python, the scope of a variable determines where you can access or modify that variable. Global scope allows access from anywhere, while local scope restricts it to a block or function.
* In Python, a programmer defines a local variable within a specific block or function, which can only be accessed or modified within that block or function.
* In Python, a global variable is a variable defined at the top level of a program that any part of the code can access or modify.
* Exception handling in Python is a mechanism for managing and responding to errors and exceptions that may occur during program execution, preventing them from crashing the program.
* In Python, you use the "try-except" statement to attempt a block of code and specify alternative actions to execute if an error occurs, allowing you to handle exceptions.
* In Python, you use the "try-except-else" statement to attempt a block of code, handle exceptions in the "except" block, and execute code in the "else" block when no exceptions occur.
* Python developers use the "try-except-else-finally" statement to attempt a block of code, catch exceptions in the "except" block, execute code in the "else" block when no exceptions occur, and ensure that the "finally" block always runs, regardless of whether an exception raised or not.
* In Python, objects are instances of classes that encapsulate data and behavior, serving as the foundation for creating and working with various data types and custom data structures.
* To determine the type of an object in Python, you can use the `type()` command.
* Any changes made within the method of the object may result in a change in object type.
* Classes in Python are blueprints for creating objects, defining their attributes and methods, enabling code organization, and object-oriented programming.
* Function "init" is a special method used to initialize data attributes.
* We can create instances of a class in Python.
* Data attributes consist of the data defining the objects.
* Methods are functions that interact and change the data attributes.
* The method has a function that requires the self as well as other parameters.

## Cheat sheet: Python programming fundamentals

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/handouts/Cheat_Sheet_Week-3.md.html?origin=www.coursera.org>

## Module 3 glossary: Python programming fundamentals

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Analogy | Refers to a concept or comparison outside the scope of the programming language itself, used to explain or relate one concept to another in a more understandable way. |
| Attributes | Attributes in Python refer to the characteristics or properties of an object, and they can be accessed using dot notation. |
| Branching | Branching in Python is a process of altering the flow of a program based on conditions, typically using if, elif, and else statements. |
| Comparison operators | Comparison operators in Python are used to compare values and return Boolean results (True or False), including operators like == (equal),!= (not equal), < (less than), > (greater than), <= (less than or equal to), and >= (greater than or equal to). |
| Conditions | Conditions in Python are used to make decisions in code, executing specific blocks of code based on whether a given expression evaluates to True or False. |
| Enumerate | In Python, "enumerate" is a built-in function that adds a counter to an iterable, allowing you to loop through both the elements and their corresponding indices. |
| Exception handling | Exception handling in Python is a mechanism for gracefully managing and responding to errors or exceptional conditions that may occur during program execution. |
| Explicitly | In Python, the term "explicitly" refers to performing an action or specifying something in a clear, unambiguous, and direct manner. |
| For loops | For loops in Python are used for iterating over a sequence (such as a list, tuple, or string) or other iterable objects, executing a set of statements for each item in the sequence. |
| Global variable | Global variables in Python are variables defined outside of any function or block and can be accessed and modified from any part of the code. |
| Incremented | "Incremented" in Python means to increase the value of a variable by a specified amount, typically done using the += operator or by adding a fixed value. |
| Indent | In Python, "indent" refers to the use of whitespace at the beginning of a line to signify the structure and scope of code blocks, such as loops and functions. |
| Indices | In Python, "indices" refer to the position or location of elements in a sequence, like a string, list, or tuple, starting with 0 for the first element. |
| Iterate | In Python, "iterate" means to repeatedly perform a set of operations or steps on each item in a collection, such as a list, tuple, or dictionary, typically using loops or iterators. |
| Local variables | Local variables in Python are variables defined within a specific function or block of code and are only accessible within that function or block. |
| Logic operators | Logic operators in Python are used to perform logical operations on Boolean values, including operators like and (logical AND), or (logical OR), and not (logical NOT). |
| Loops | Loops in Python are constructs for repeating a block of code, enabling the execution of the same code multiple times. |
| Parameters | Parameters in Python are placeholders in a function definition, used to accept and work with values provided to the function when it is called. |
| Programming Fundamentals | Programming fundamentals in Python involve variables, control structures, functions, data structures, input/output, and error handling for building software. |
| Range function | The range function in Python generates a sequence of numbers that can be used for iterating in a loop and is typically used as range (start, stop, step), where it creates numbers from start to stop-1 with the given step increment. |
| Scope of function | The "scope of a function" in Python refers to the region of code where a variable defined within that function is accessible or visible. |
| Sequences | Sequences in Python are ordered collections of items that can include data types like strings, lists, and tuples, allowing for indexing and iteration. |
| Syntax | In Python, "explicitly" means to state something clearly and directly, leaving no room for ambiguity or implicit interpretation. |
| While loops | While loops in Python are used to repeatedly execute a block of code as long as a specified condition is true. |

## **MODULE 4 Working with Data in Python**

## Reading files with open

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/lab/Module_4/Read_file_with_open_Reading.md.html?origin=www.coursera.org>

## Reading files with open

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-3481987e7e258049751b8fa1864bf21ea56a2904>

## Writing files with open

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/lab/module_4/write_file_with_open_reading.md.html?origin=www.coursera.org>

## Writing files with open

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstUFkwMTAxRU4tU2tpbGxzTmV0d29yay9sYWJzL2xhYi9tb2R1bGVfNC93cml0ZV9maWxlX3dpdGhfb3Blbl9yZWFkaW5nLm1kIiwidG9vbF90eXBlIjoiaW5zdHJ1Y3Rpb25hbC1sYWIiLCJhdGxhc19maWxlX2lkIjoxMDg0NSwiYWRtaW4iOmZhbHNlLCJpYXQiOjE3MjkxMDE1ODN9.oyadsz29PJwjMF2qQzqRo-tGLphd7I65NKmBm1hLj4I>

## Pandas

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module_4/Reading_Pandas.md.html?origin=www.coursera.org>

## Selecting Data in a Data Frame

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-826b414eaa32d6037739ed3f76e35c735b213ac6>

## Loading Data with Pandas

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-6ef12b311613461a781af1e12e2ec127a4e37e63>

## One Dimensional NumPy

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/jupyterlite/latest/lab/index.html?notebook_url=https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%205/PY0101EN-5-1-Numpy1D.ipynb>

## NumPy Matrix Mathematics

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/mod_4/matrix_maths.md.html>

## Two Dimensional NumPy

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-ac43516ffb5f3a6f2cbfe9842574692793fe2152>

## Beginners Guide to NumPy

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module_4/numpy_reading.md.html>

## Some contexts on API

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/mod_5/API_relevance.md.html?origin=www.coursera.org>

## Module 4 Summary: Working with Data in Python

* Python uses the open() function and allows you to read and write files, providing access to the content within the file for reading. It also allows overwriting it for writing and specifies the file mode (for example, r for reading, w for writing, a for appending).
  + To read a file, Python uses an open function along with *r.*
  + Python uses the**open with** function to read and process a file attribute, that is, from open to close.
  + In Python, you use the **open** method to edit or overwrite a file.
  + To write a file, Python uses the **open** function along with *w.*
  + In Python, "a" indicates that the program has appended to the file.
  + In Python, “\n” signifies that the code should start on a new line.
  + Python uses various methods to print lines from attributes.
* Pandas is a powerful Python library for data manipulation and analysis, providing data structures and functions to work with structured data like data frames and series.
  + You import the file (panda) by using the import command followed by the file name.
  + In Python, you use the **as** command to provide a shorter name for the file.
  + In Pandas, you use a data frame (df) to specify the files to read.
  + DataFrames consist of rows and columns.
  + You can create new DataFrames by using the column or columns of a specific DataFrame.
  + We can work with data in a DataFrames and save the results in different formats.
  + In Python, you use the **Unique** method to determine unique elements in a column of the DataFrames.
  + You use the inequality operator along with df to assign a Boolean value to the selected column in DataFrames.
  + You save a new DataFrame as a different DataFrame, which may contain values from an earlier DataFrame.
* NumPy is a Python library for numerical and matrix operations, offering multidimensional array objects and a variety of mathematical functions to work with data efficiently.
  + NumPy is a basis for Pandas.
  + A NumPy array or ND array is similar to a list, usually of a fixed size with the same kind of element.
* A one-dimensional NumPy array is a linear sequence of elements with a single axis, like a traditional list, but optimized for numerical computations and array operations.
  + You can access elements in a NumPy using an index.
  + You use the attribute **dtype** to get the data type of the array elements.
  + You use **nsize** and **ndim** to get the size and dimension of the array, respectively.
  + You can use indexing and slicing methods in NumPy.
  + Vector additions are widely used operations in Python.
  + Representing vector addition with line segments or arrows is useful.
  + NumPy codes work much faster, which is helpful with lots of data.
  + You perform vector subtraction by replacing the addition sign with a negative sign.
  + Multiplying an array by a scalar in Python entails multiplying each element of the array by the scalar value, leading to a new array in which each element scales by the scalar.
  + Hadamard product refers to the element-wise multiplication of two arrays of the same shape, resulting in a new array where each element is the product of the corresponding elements in the input arrays.
  + The dot product in Python is the sum of the element-wise products of two arrays, often used for vector and matrix operations to find the scalar result of multiplying corresponding elements and summing them.
  + When working with NumPy, it is common to utilize libraries like Matplotlib to create graphs and visualizations from numerical data stored in NumPy arrays.
* A two-dimensional NumPy array is a grid-like structure with rows and columns suitable for representing data as a matrix or a table for numerical computations.
  + In NumPy, "shape" refers to an array's dimensions (number of rows and columns), indicating its size and structure.
  + You use the attribute "size" to obtain the size of an array.
  + You use rectangular attributes to access the various elements in an array.
  + You use a scalar to multiply elements in NumPy.

## Cheat Sheet: Working with Data in Python

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/handouts/Cheat_Sheet_Week-4.md.html?origin=www.coursera.org>

## Module 4 Glossary: Working with Data in Python

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/200336.025_Module_4_Glossary.md.html>

## **MODULE 5 API and Data Collection**

## Introduction to API

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-2a96c5751a445900997e9b5966eb386c0651bd9e>

## Practice Project: GDP Data Extraction and Processing

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-293571b9d67673c0e6e499716401d89380f58918>

## Web Scraping and HTML Basics

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/lab/module_5/WebScraping_reading.md.html>

## Access REST APIs & Request HTTP

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-87a589b34505945dd0b33079ec07059499d15a46>

## API Examples

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-a1e7e02885c09c0ac95800b28ad5db4ac6e74580>

## Web Scraping - A Key Tool in Data Science

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/mod_5/additional_webscraping.md.html>

## Web Scraping Tables using Pandas

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/mod_5/webscraping_using_pandas.md.html>

## Web Scraping

<https://labs.cognitiveclass.ai/v2/tools/jupyterlab?ulid=ulid-f37bec6fa10ab127c2d4b3c02d4043fcd12424ea>

## Working with different file formats

<https://labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-d282380e893ba3137046abaf92f07fe0c46f4ecb>

## Module 5 Summary: APIs and Data Collection

* Simple APIs in Python are application programming interfaces that provide straightforward and easy-to-use methods for interacting with services, libraries, or data, often with minimal configuration or complexity.
  + An API lets two pieces of software talk to each other.
  + Using an API library in Python entails importing the library, calling its functions or methods to make HTTP requests, and parsing the responses to access data or services provided by the API.
  + Pandas API processes the data by communicating with the other software components.
  + An Instance forms when you create a dictionary and then use the DataFrames constructor to create a Pandas object.
  + Method “head()” will display the mentioned number of rows from the top (default 5) of DataFrames, while method “mean()” will calculate the mean and return the values
* Rest APIs allow you to communicate through the internet, taking advantage of resources like storage, access more data, AI algorithms, and so on.
  + HTTP methods transmit data over the internet.
  + An HTTP message typically includes a JSON file with instructions for operations.
  + HTTP messages containing JSON files are returned to the client as a response from web services.
  + Dealing with time series data involves using the Pandas time series function.
  + You can get data for daily candlesticks and plot the chart using Plotly with the candlestick plot.
* The HTTP (HyperText Transfer Protocol) transfers data, including web pages and resources, between a client (a web browser) and a server on the World Wide Web.
  + The HTTP protocol is commonly used for implementing various types of REST APIs.
  + An HTTP response includes information like the type of resource, length of resource, and so on
  + Uniform resource locator (URL) is the most popular way to find resources on the web.
  + URL is divided into three parts: scheme, internet address or base URL, and route
  + The GET method is one of the popular methods of requesting information. Some other methods may also include the body.
  + Response method contains the version and body of the response.
  + POST submits data to the server, PUT updates data already on the server, DELETE deletes data from the server
* Requests is a Python library that allows you to send HTTP/1.1 requests easily
  + You can modify the results of your query with the GET method.
  + You can obtain multiple requests from a URL like name, ID, and so on with a Query string.
* Web scraping in Python involves extracting and parsing data from websites to gather information for various applications, using libraries like Beautiful Soup and requests.
  + HTML comprises text surrounded by blue text elements enclosed in angular brackets called tags.
  + You can select an HTML element on a web page to inspect the webpage.
  + Web pages may also contain CSS and JavaScript along with HTML elements.
  + Each HTML document is like an HTML Tree, which may contain strings and other tags.
  + Each HTML table is comprised of table tags and is structured with elements such as rows, headers, body and so on.
* Tabular data can also be extracted from web pages using the `read\_html` method in Pandas.
* Beautiful Soup in Python is a library for parsing and navigating HTML and XML documents, making extracting, and manipulating data from web pages more accessible.
* To parse a document, pass it through the Beautiful Soup constructor to get a beautiful soup object representing the document as a nested data structure.
* Beautiful soup represents HTML as a set of tree-like objects with methods to parse the HTML.
* Navigable string is like a Python string that supports beautiful soup functionality.
* find\_all is a method used to extract content based on the tag’s name, its attributes, the text of a string, or some combination of these.
* The find\_all method looks through a tag’s descendants and retrieves all descendants that match your filters.
* The result is a Python iterable like a list.
* File formats refer to the specific structure and encoding rules used to store and represent data in files, such as .txt for plain text or .csv for comma-separated values.
* Python works with different file formats such as CSV, XML, JSON, xlsx, and so on
* The extension of a file name will let you know what type of file it is and what it needs to open with.
* To access data from CSV files, we can use Python libraries such as Pandas.
* Similarly, different methods help parse JSON, XML, and other files.

## Cheat Sheet: APIs and Data Collection

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/handouts/Cheat_Sheet_Week-5.md.html?origin=www.coursera.org>

## Module 5 Glossary: APIs and Data Collection

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/200336.031_Module_5_Glossary.md.html>

## Python Cheat Sheet: The Basics

<https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/handouts/Python%20Cheat%20Sheet%20-%20The%20Basics%20Coursera.pdf>