

## Python Basics – Core

Introduction to Python Language	
First Encounter with Python Installation on Computer (OS Windows, Mac)	Options for executing commands. Moving to Visual Studio Code, PyCharm, Anaconda, Jupyter Notebook
Basic Language Constructs	
Variables, assignment operator, type and id functions	print() and input() functions
Numeric types, arithmetic operations	Converting strings to int() and float() numbers
Mathematical functions and working with math module	Boolean type bool. Comparison operators and and, or, not operators
Strings and Lists	
Introduction to strings. String operations	String formatting: F-strings
Introduction to string indexes and slices	Lists - operators and functions
Basic string methods	List slices and list comparison
Special characters, char escaping, r-strings	Basic list methods
String formatting: % and format method	Nested lists, multidimensional lists
Conditional statements, loops, list comprehensions	
Conditional statement if. If-else construct	for loop statement, enumerate() function
Nested conditions and multiple choice.	Iterator and iterable objects. iter() and next() functions
Ternary conditional operator. Nested ternary condition while loop statement	Nested loops. Examples with nested loops
Loop control statements break, continue, and else	Example of working with nested loops. List comprehensions
for loop statement, range() function	Nested list comprehensions
Dictionaries, Tuples, and Sets	
Introduction to dictionaries (dict). Basic dictionary operations	Sets (set) and their methods
Dictionary methods, iterating over dictionary elements in a loop	Set operations, set comparison
Tuples (tuple) and their methods	Set and dictionary comprehensions
Functions	
Functions: first acquaintance, def definition and their call	Recursive functions
return statement in functions. Functional programming	Anonymous (lambda) functions
Euclidean algorithm for finding the greatest common divisor (GCD)	Variable scope. global and nonlocal keywords
Named arguments. Actual and formal parameters	Closures in Python

Functions with arbitrary number of parameters *args and **kwargs	Introduction to function decorators
Operators * and ** for packing and unpacking collections	Decorators with parameters. Preserving properties of decorated functions
<b>Modules and Packages. Working with Files</b>	
Importing standard modules. import and from commands	open() function. Reading data from a file
Importing your own modules	Handling FileNotFoundError and context manager (with) for files
Installing third-party modules (pip install). Batch installation	Writing data to a file in text and binary modes
Packages (package) in Python. Nested packages	Modules pickle, shelve, os
<b>Generators. Some useful functions</b>	
Generator expressions	Sorting through sort() and sorted() functions' peculiarities
Generator function. yield statement	key argument for sorting collections by key
map() function. Examples of its usage	isinstance and type functions for checking data types
filter() function for filtering values of iterable objects	all() and any() functions. Examples of their usage
zip() function. Examples of usage	Lambda and generator functions
<b>Additional</b>	
Extended number representation. Number systems	Annotations of types at class level
Bitwise operations AND, OR, NOT, XOR. Shift operators	match/case construct. First acquaintance
random module of the standard library	match/case construct with tuples and lists
Annotation with basic types	match/case construct with dictionaries and sets
Annotations of collection types	match/case construct. Examples and peculiarities of usage