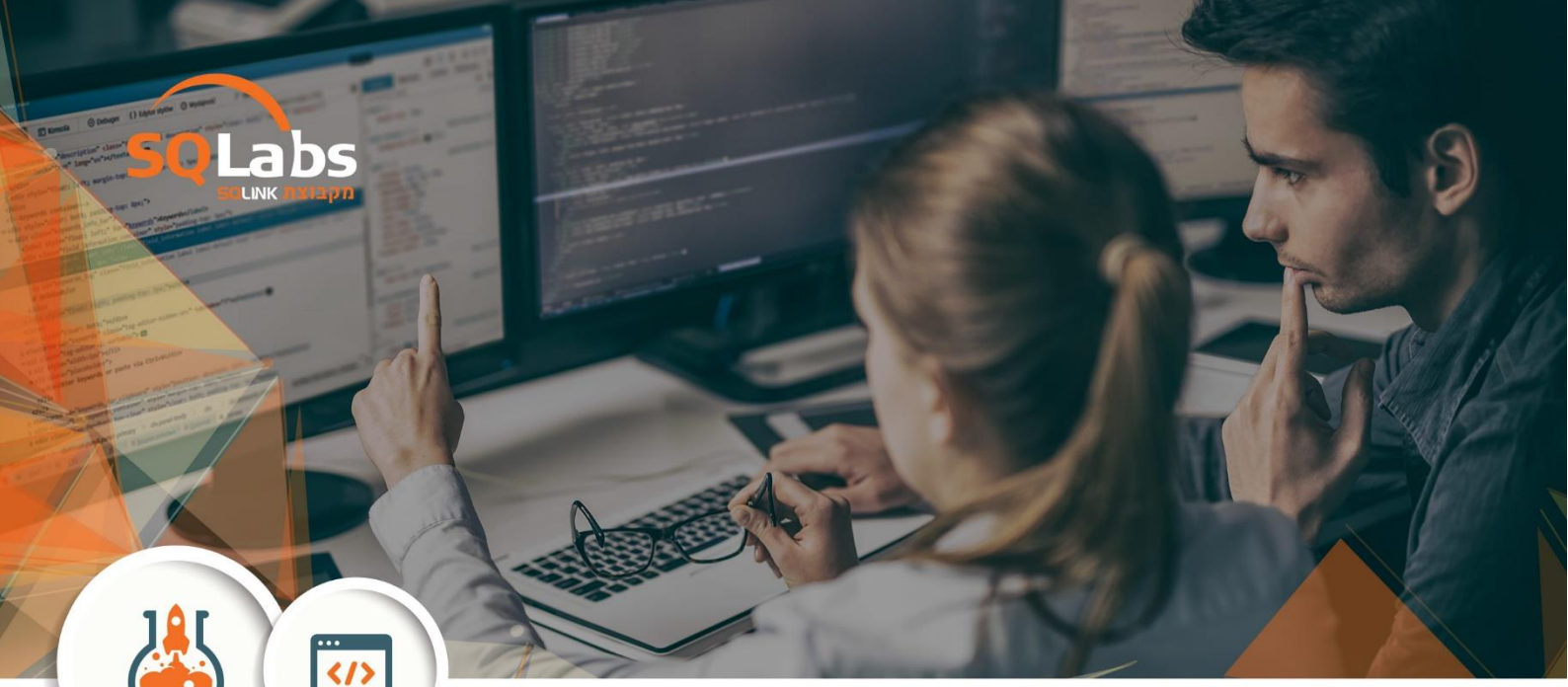


# Python for Beginners– 40 Hours

## Outline

A serious course for beginners. Target Audience Programmers with little or no knowledge in Python or other object oriented language. Exelent for QA engineers, DEV-OPS engineers, data analysts or anybody that is serious about learning the Python language. (we'll learn just the Python language, not the other tools these folks need in their job. So no Sellenium, no NumPy/Pandas..etc.)

## Target Audience

Programmers with little or no knowledge in Python or other object oriented language.
Exelent for QA engineers, DEV-OPS engineers, data analysts or anybody that is serious about learning the Python language.
(we'll learn just the Python language, not the other tools these folks need in their job. So no Sellenium, no NumPy/Pandas..etc.)

## Pre-requisites

- Basic experience in programming, in any programming language. (yes, Pascal from high-school is OK)
- Vauge knowledge of Object Oriented concepts.

## Syllabus

### Introduction and basic data types

- The uses of Python
- Static vs. Managed Vs. Dynamic Languages
- Development Tools
- python data: everything is an object
- Numbers: int, float
- Strings: the str type
- Boolean: the bool type
- Basics of lists
- Input & Output

### Control Flow structures

- if statement
- for loops
- range type and the range() function
- break and continue
- pass keyword
- while loops

### Functions

- defining functions with the def keyword
- default values for parameters
- keyword arguments
- arbitrary argument lists (*args)
- unpacking argument lists
- lambda expressions
- nested functions and closures
- Scopes : usign the global keyword

### Data Structures

- Python lists (The del statement)
- Looping with in
- Python dictionaries
- Python tuples
- Python sets

### Modules & Packages

- Creating modues in a separate file
- Importing complete modules
- Importing from modules
- Module search path
- “compiling” modules
- The standard library
- dir() function
- Packages

**Input/Output**

- Using formated printing
- opening files
- reading and writing text
- Reading and Writing JSON

**Errors and Exceptions**

- Errors & Exceptions
- Handling Exceptions
- Raising Exceptions
- User defined Exceptions
- Cleanup actions with finaly

**Classes**

- Python namespaces
- Class definition, class object
- Instance Objects
- Method Objects
- Class variables and instance variables
- Inheritance
- Iterators
- Generators

**Python Project (or..make sure we really learned something)**

- Project design
- PEP-8
- We'll use PyCharm
- ..and..Code Review !!!