



PYTHON COURSE

While students in large colleges study masses of theory, which is mostly obsolete and irrelevant to the vacancies offered, we, who still work simultaneously as interviewers and recruiters in companies, had the vision of Python classes which would **focus on and be adapted to the labor market**; classes that would be fresh and up to date, during which the students learn only what they need to know for their future jobs. This means: gaining more practice, paying less money. Two birds one stone.

Our knowledge, your future



Individuals

Our data courses focus on practical knowledge; in class exercises, homework assignments and learning in small groups which allows for personal attention and better understanding of the material.



Companies

We offer customized big data engineering courses and workshops according to your company needs. Course materials are suited to your everyday tasks and training requirements.



“Find Work”

We can provide career assistance by reviewing your resume, teaching social media networking and defining LinkedIn content for professional “branding” as well as refer you to relevant positions.

COMPANY DETAILS

HEADQUARTERS

Ramat Gan, Tel Aviv District

YEAR FOUNDED

2015

COMPANY TYPE

Educational Institution

COMPANY SIZE

40-50 employees

What Is Python?

Python is one of the most in-demand programming languages in the industry. It is favored by companies such as Instagram, YouTube, and Spotify. Python's diversity, adaptability and its elegant, easy-to-master basics set it apart from other programming languages, making it very popular. With python you can do everything from GUI development, Web application, System administration tasks, financial calculation, Data Analysis, Visualization, and list goes on.

COMPANIES USING
PYTHON


Google


NETFLIX


Expedia
 PayPal



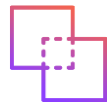
Easy to Learn

Known for its simplicity and elegant syntax. Python is much easier to read and write than other programming languages.



Free & Open-Source

Python is free to download and use commercially. You can even edit the source code and contribute to its community.



Highly Versatile

Python is used in a diverse range of applications and fields of study, which contributes to the growing demands for Python developers.



Fewer Lines of Code

Python is highly expressive and allows for greater functionality. You'll be amazed how much you can accomplish with only 500 lines of code.

THE INSTRUCTORS



David Rotenberg
Senior VP of R&D
Telefire



Alex Kuznetsov
DevOps Team Lead
Ceva



Doron Nuni
R&D Group Manager
Cognyte



Yuval Wilf
Sr. Director of Architecture
Imperva



Course Summary



Frontend

We teach you the language and tools you'll need to create what users see and how they could interact with a website or a mobile application.

Backend

You'll learn the logic on how software applications work behind the scenes to produce desired contents or results.

Data Base

You'll understand how user information is stored. You'll also learn other relevant information such as setting things up in the cloud.

OUR ALUMNI WORK WITH THE BEST



<i>Topic</i>	<i>Description</i>
PYTHON CORE	
Overview and setup	<ul style="list-style-type: none"> • Introduction • Python environment setup • PyCharm setup
Python basics	<ul style="list-style-type: none"> • Data types • Syntax • Operators • Statements • Conditions • Loops • Data input • Comments • Functions
Data structures	<ul style="list-style-type: none"> • Array • List • Tuple • Dictionary
Files (I/O)	<ul style="list-style-type: none"> • Reading • Writing • File manipulations • Directories
Error handling	<ul style="list-style-type: none"> • What are exceptions? • Python built-in exceptions • Try-Except • Finally • Raise exceptions • User defined exception • Assertions
Modules and tools	<ul style="list-style-type: none"> • Debug • Packages • PIP • Conventions • Unit testing

<p>Unit test</p>	<ul style="list-style-type: none"> • What is a testing framework? • PyTest • Decorators • Assertions • Run order • Soft & hard fails • Test run and suites
<p>Design patterns</p>	<ul style="list-style-type: none"> • What are Design patterns? • Anti-design patterns • Programming conventions • Coupling and de-coupling • Page object • Conventions • Refactoring
<p>Source control management</p>	<ul style="list-style-type: none"> • What is Git? • Understanding Git • Terminology • Setup & configuration • Configuring Git • Repository overview • Building your repository • Branching and Merging • Commits • Resolving conflicts • Pull Requests • Remote repository
<p>Python Project</p>	<ul style="list-style-type: none"> • Using Python file I/O, functions, and OOP to decrypt an encrypted file.

FRONTEND	
HTML, CSS	<ul style="list-style-type: none"> • Introduction • Elements • Attributes • CSS • Web Server • Web Client
Flask	<ul style="list-style-type: none"> • Flask as a micro web framework for Python • Flask pip installation • Virtual environment for Flask projects • application structure (app.py, templates, static) • modular code organization • Routes and views • Returning dynamic content • Jinja2 template engine • Templates and inheritance • Error handling and exceptions
Selenium WebDriver	<ul style="list-style-type: none"> • WebDriver Architecture • WebDriver object methods • Asserts and verifications • Switch and navigation • Synchronization • Reporting system • Page object model (POM) • Selenium keys • Alerts
WebDriver elements	<ul style="list-style-type: none"> • Locating Web elements • Web elements interaction • Locators types • Controllers • Web elements data • File uploads
Web Application Project	<ul style="list-style-type: none"> • Using Flask and SQLAlchemy to create a fully functional, interactive, DB-based web app.

BACKEND	
REST API Methods and codes	<ul style="list-style-type: none"> • Rest methods • Status codes • API example
REST API Requests & Responses	<ul style="list-style-type: none"> • Headers • Content types • Handling responses • Authorization • Python rest API libraries • Sending requests from Python code • Handling responses from Python code
JSON	<ul style="list-style-type: none"> • Overview • Structure • JSON rules • JSON parsing
REST hands-on	<ul style="list-style-type: none"> • Requests • REST client • REST CRUD operations • REST methods • Creating REST requests • Response processing • Working with a real API • Running automatic API tests • Selenium & API testing integration • API as Data Driven tests
Backend server	<ul style="list-style-type: none"> • Routes • Path params • Query params • Binding • Debug mode

Relational Database	<ul style="list-style-type: none"> • Overview • PyMySQL • SQLAlchemy • Setup and configuration • MySQL drivers • Running queries from our code • Creating and working with a remote database • Connections • Statements and prepared statements • Security • DB operations
Advanced Flask	<ul style="list-style-type: none"> • Static files and assets • Database integration with Flask-SQLAlchemy • User authentication and authorization • Flask-RESTful for API endpoints • Serializing and deserializing data with marshmallow • Deployment options for Flask applications
<i>CONTINUOUS DEVELOPMENT</i>	
CI Intro	<ul style="list-style-type: none"> • Overview • Why continuous integration is used for? • Jenkins introduction • Jenkins Architecture
Jenkins core	<ul style="list-style-type: none"> • Setup & configuration • Jenkins configurations • Plugins • Pipelines • Alerts and notifications • Authentication • Authorization • Creating users

<p>Build jobs</p>	<ul style="list-style-type: none"> • What are build Jobs • Building your build jobs • Build triggers • Scheduling • CRON expressions • Reporting • Disabling and deleting jobs • Jenkins - Maven Integration • Automating automations • Nightly runs
<p>Automation Project</p>	<ul style="list-style-type: none"> • Building an Automation on top of your web app project using Selenium and enabling PyTest runs and reporting with Jenkins.
<p>Cloud testing</p>	<ul style="list-style-type: none"> • Cloud introduction • SauceLabs • RemoteWebDriver • Browser pools • Parallel testing
<p>Using ChatGPT</p>	<ul style="list-style-type: none"> • GPT-x models and OpenAI API • API key and setup • OpenAI Python library • Authentication and rate limits • Parameters for requests (prompt, max tokens, temperature) • Receiving and parsing responses • Example use cases • Integrating ChatGPT into Flask