

Professional Master's Degree

Python Development



Professional Master's Degree Python Development

- » Modality: online
- » Duration: 12 months
- » Certificate: TECH Technological University
- » Dedication: 16h/week
- » Schedule: at your own pace
- » Exams: online

Website: www.techtute.com/us/information-technology/professional-master-degree/master-python-development

Index

01

Introduction

p. 4

02

Objectives

p. 8

03

Skills

p. 12

04

Course Management

p. 16

05

Structure and Content

p. 20

06

Methodology

p. 30

07

Certificate

p. 38

01

Introduction

Python stands out as a versatile and efficient programming language. First of all, its clear and readable syntax makes it easy to write and understand code, which speeds up the development process and reduces the likelihood of errors. In addition, the wide variety of available libraries and Frameworks gives developers access to powerful, pre-existing tools, enabling faster and more efficient development. Its open source nature and active developer community also contribute to the constant improvement and updating of the language, ensuring a dynamic and modern development environment. For this reason, TECH has developed a comprehensive program, aimed at professionals with the latest advances in Python Development, with a 100% online methodology.



“

Python is recognized for its adaptability, being used in applications ranging from web development to Artificial Intelligence. What are you waiting for to enroll?"

Python is a high-level programming language, widely used among computer scientists, as it has a wide range of libraries and frameworks that simplify common tasks, allowing developers to focus on the logic of their applications instead of wasting time on technical details. Moreover, its versatility is another of its significant advantages, as it can be used in a variety of environments, from web development to data analysis and machine learning.

This is the origin of this Professional Master's Degree in Python Development, a program that will cover advanced data and type handling in Python, exploring identifiers, keywords, integral and Boolean types, as well as advanced strings formatting and encodings. Additionally, collections such as tuples, lists and dictionaries will be examined, along with iteration techniques and lambda functions, providing a solid foundation in the fundamentals of the language.

Furthermore, you will delve into Python application development, with an emphasis on best practices and modern methodologies. From application architecture to deployment and maintenance, the course will cover aspects such as advanced design and modeling, dependency management, design patterns, testing and debugging, performance optimization, deployment and distribution strategies.

Similarly, the computer scientist will delve into web and mobile development with Python, covering frameworks such as Django and Flask, as well as the development of APIs and web services. In addition, it will focus on interface design and user experience (UI/UX), from the use of design tools to improving accessibility and usability. Finally, data management and analysis will be addressed, using Python and tools such as NumPy, Pandas and Matplotlib.

This program is presented as a unique opportunity, through a fully online and adaptable academic proposal. With this approach, professionals will enjoy greater freedom to manage their access time, allowing them to harmonize their daily personal and work commitments.

This **Professional Master's Degree in Python Development** contains the most complete and up-to-date program on the market. The most important features include:

- ♦ The development of practical cases presented by experts in Python Development
- ♦ The graphic, schematic and practical contents of the book provide theoretical and practical information on those disciplines that are essential for professional practice
- ♦ Practical exercises where self-assessment can be used to improve learning
- ♦ Its special emphasis on innovative methodologies
- ♦ Theoretical lessons, questions to the expert, debate forums on controversial topics, and individual reflection assignments
- ♦ Content that is accessible from any fixed or portable device with an Internet connection



A comprehensive and specialized qualification in Python Development, which will prepare you to face real challenges in the world of Software Development"

“

Through this 100% online Professional Master's Degree, you will delve into Frontend and Backend design, database management and app store publishing strategies"

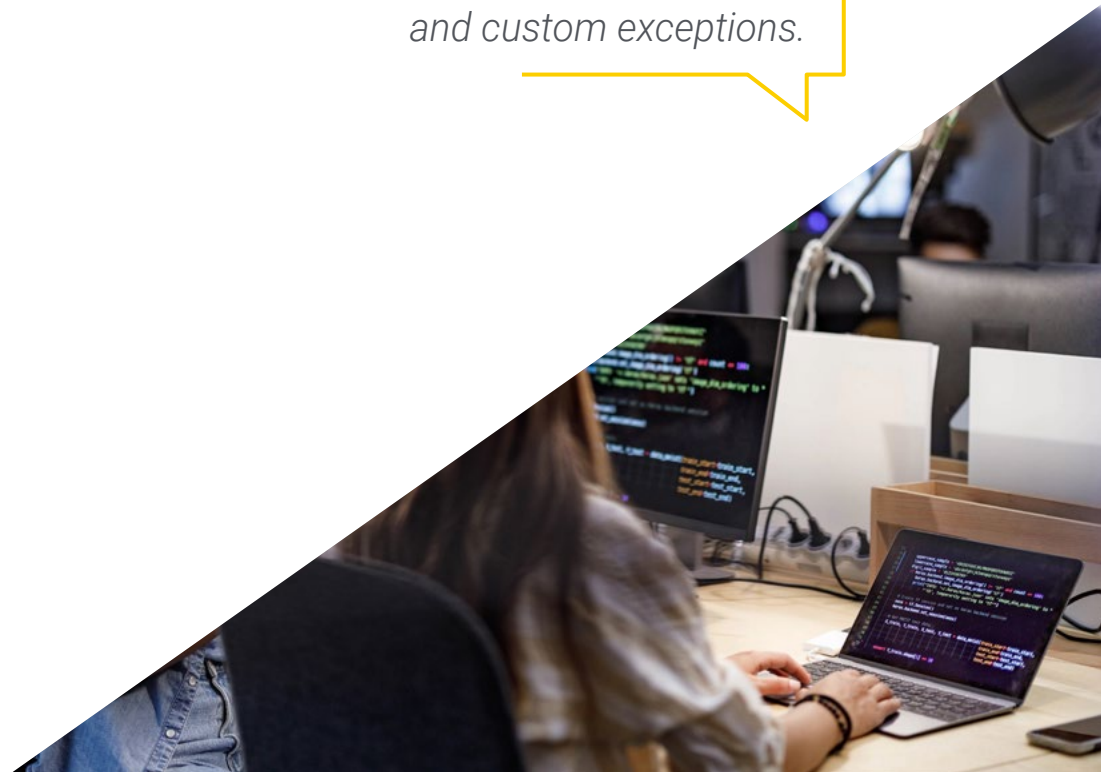
The program's teaching staff includes professionals from the field who contribute their work experience to this educational program, as well as renowned specialists from leading societies and prestigious universities.

The multimedia content, developed with the latest educational technology, will provide the professional with situated and contextual learning, i.e., a simulated environment that will provide immersive education programmed to learn in real situations.

This program is designed around Problem-Based Learning, whereby the professional must try to solve the different professional practice situations that arise during the academic year. For this purpose, the students will be assisted by an innovative interactive video system created by renowned and experienced experts.

You will cover everything from the basics such as variables and data types, to the most advanced data visualization techniques and performance and storage optimization strategies. Enroll now!"

You'll delve into Object Oriented Programming (OOP), as well as topics such as classes, inheritance, polymorphism, abstract class creation and custom exceptions.



02

Objectives

This Professional Master's Degree has been designed with the firm purpose of cultivating highly competent and specialized professionals in the vast universe of programming in this high-performance language. Through comprehensive modules, graduates will gain a deep immersion in Python syntax and functionalities, from advanced data management, to mastery in Object Oriented Programming (OOP) and the efficient design of web and mobile applications. With a rigorous and up-to-date pedagogical approach, this program will guarantee a unique educational experience that will position computer scientists as sought-after experts in the Software development industry. Get ready to push your limits and step into an educational journey that not only elevates your Python skills, but also gives you the tools to excel and lead in the dynamic world of technology development.



“

Become a leader in the field of Software development, equipping them with advanced knowledge and specialized skills to address contemporary technology challenges”



General Objectives

- Provide a comprehensive understanding of Python
- Enable advanced data and type handling in Python
- Apply the principles of Object Oriented Programming (OOP) in Python
- Encourage the use of best practices and modern methodologies in software development
- Provide comprehensive education in web and mobile development with Python
- Integrate UI/UX principles in software development
- Teach the configuration and use of data development tools and environments
- Delve into the use of data structures and functions in Python
- Learn advanced data visualization techniques with Matplotlib
- Learn performance optimization and data warehousing strategies



Thanks to the extensive library offered by this program, filled with the most innovative multimedia resources, you will gain the tools to stand out and lead in the dynamic world of technological development"





Specific Objectives

Module 1. Python Programming

- ◆ Enable the configuration and effective use of the Python development environment
- ◆ Understand advanced programming concepts

Module 2. Advanced Data and Flow Control with Python

- ◆ Master the conventions and practices for handling identifiers and keywords
- ◆ Apply complex data structures and their operations

Module 3. Object Oriented Programming (OOP) in Python

- ◆ Master the creation and use of classes and objects in Python
- ◆ Apply Inheritance and Polymorphism in Python

Module 4. Application Development in Python

- ◆ Specialize in the design and advanced modeling of applications
- ◆ Learn how to optimize, deploy and maintain applications

Module 5. Web and Mobile Development with Python

- ◆ Use popular Python web *Frameworks*
- ◆ Prepare for the development and publication of mobile applications

Module 6. User Interface and User Experience with Python

- ◆ Instruct in responsive and adaptive design techniques
- ◆ Prepare for usability testing and user behavior analysis

Module 7. Data Processing and *Big Data* with Python

- ◆ Handle flow control techniques and data handling functions
- ◆ Promote best practices for coding and error handling in Python

Module 8. Data Structures and Functions in Python

- ◆ Create and use advanced functions
- ◆ Read and write files and their processing

Module 9. Data Handling in Python with NumPy and Pandas

- ◆ Create and manipulate Arrays with NumPy
- ◆ Promote proficiency in data visualization with Matplotlib

Module 10. Advanced Techniques and Practical Applications in NumPy and Pandas

- ◆ Develop expertise in loading and storing data from and to various sources
- ◆ Instruct in advanced data cleansing and transformation strategies

03 Skills

The program will not only impart technical knowledge, but will also focus on cultivating competencies that will transform graduates into outstanding professionals in the field of software development. This will cover the ability to develop innovative and efficient solutions, through an expert command of Python, from advanced data management to the creation of high-performance web and mobile applications. In addition, computer scientists will be equipped with specialized skills in Object Oriented Programming (OOP), user interface design and user experience (UI/UX), and advanced data analysis with libraries such as NumPy and Pandas.



“

The development of crucial soft skills such as critical thinking, problem solving, team collaboration and adaptability will be encouraged”



General Skills

- Develop practical programming skills
- Equip yourself with advanced skills in program flow control
- Handle Object Oriented Software design and implementation
- Become proficient in comprehensive Python application development
- Master the design and management of web and mobile applications
- Master interface and user experience design using Python
- Develop skills in data management and analysis with Python
- Acquire skills in advanced file handling and modeling in Python
- Develop skills in advanced data management with NumPy and Pandas
- Delve into advanced data management with NumPy and Pandas

“

You will face real-world challenges and stand out in a competitive and constantly evolving job market. Bet on TECH!”





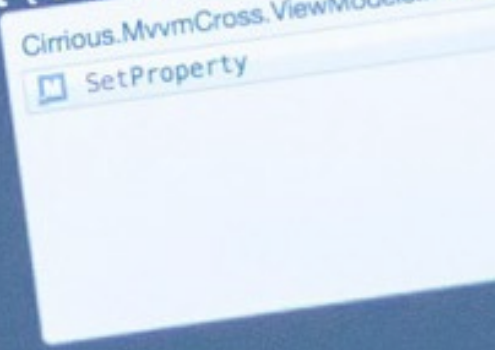
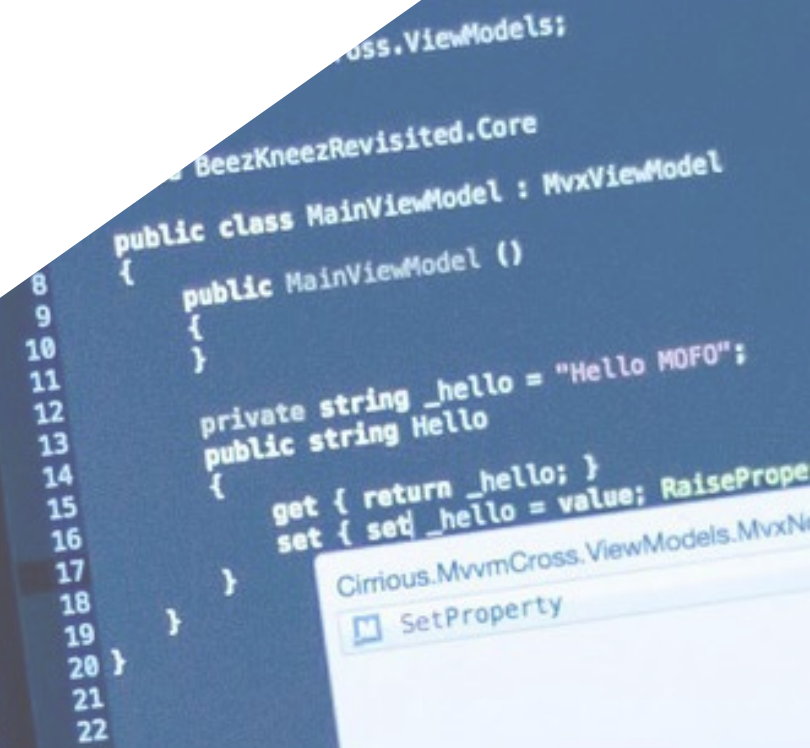
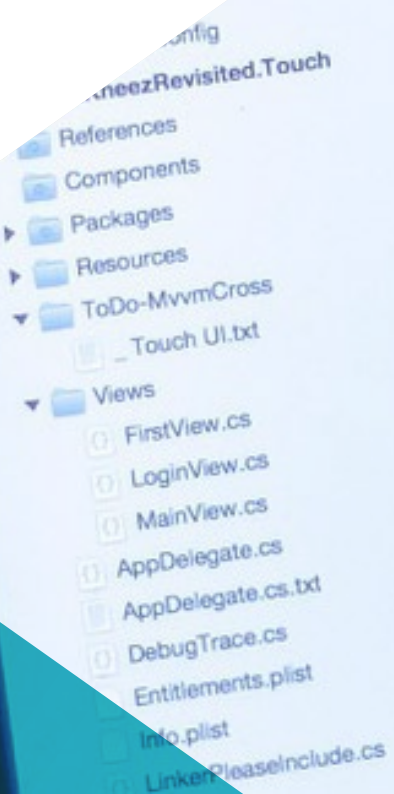
Specific Skills

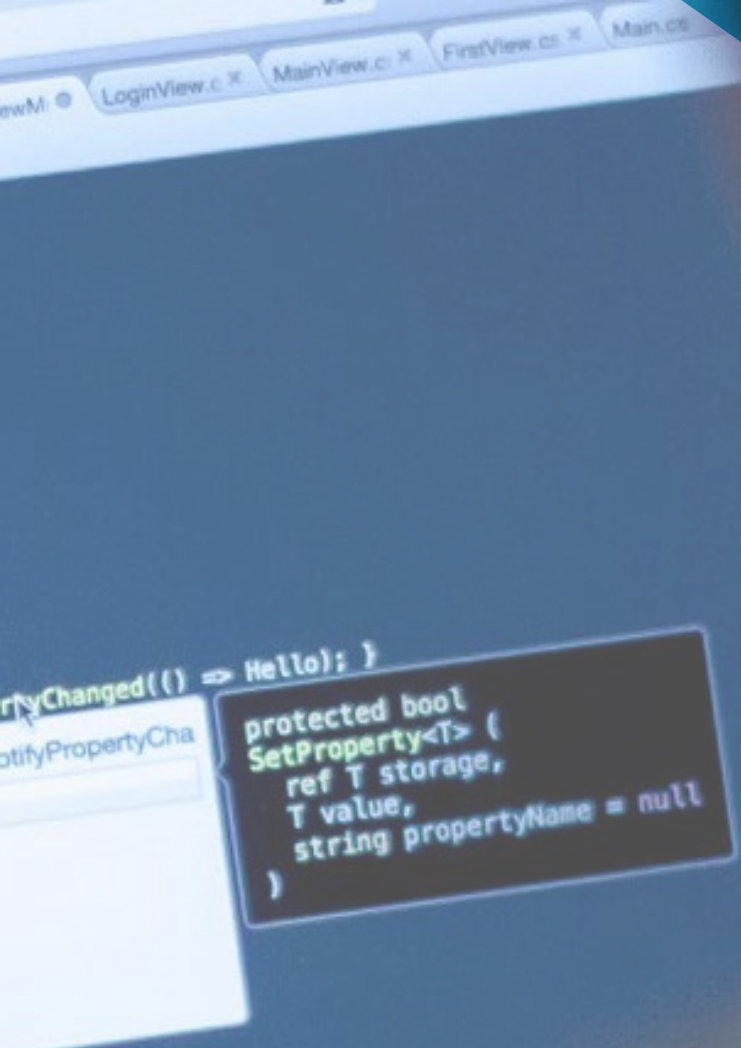
- ◆ Get qualified in advanced data handling in Python
- ◆ Handle advanced use of Python functions
- ◆ Implement advanced Object Oriented Programming (OOP) concepts such as abstract classes and custom exceptions
- ◆ Control testing and Debugging
- ◆ Develop APIs and web services
- ◆ Master the use of UI/UX design tools with Python
- ◆ Use essential Python data libraries
- ◆ Apply different data structures in a practical way
- ◆ Use Pandas for structured data handling
- ◆ Promote competencies in analysis and manipulation of time series and complex data

04

Course Management

The faculty of this Professional Master's Degree are industry experts with extensive hands-on experience in software development and a proven track record of successful projects. By marrying theory with practical application, these mentors will not only impart advanced Python knowledge, but will also share their real-world experience, providing students with valuable insight into current best practices and trends in the industry. In addition, its pedagogical approach will stand out for its commitment to providing graduates with the most up-to-date technical education to excel in the dynamic field of software development.





“

You will succeed hand in hand with the best and acquire the knowledge and skills you need to embark on the Python Development industry”

Management



Mr. Matos Rodríguez, Dionis

- Data Engineer at Wide Agency Sadexo
- Data Consultant at Tokiota
- Data Engineer at Devoteam
- BI Developer at Ibermática
- Applications Engineer at Johnson Controls
- Database Developer at Suncapital España
- Senior Web Developer at Deadlock Solutions
- QA Analyst at Metaconcept
- Professional Master's Degree in Big Data & Analytics by the EAE Business School
- Professional Master's Degree in Systems Analysis and Design
- Bachelor's Degree in Computer Engineering from APEC University

Professors

Mr. Villar Valor, Javier

- ♦ Director and Founding Partner of Impulsa2
- ♦ Chief Operations Officer (COO) at Summa Insurance Brokers
- ♦ Director of Transformation and Operational Excellence at Johnson Controls
- ♦ Professional Masters Degree in Professional Coaching
- ♦ Executive MBA from Emlyon Business School, France
- ♦ Professional Master's Degree in Quality Management from EOI, Spain
- ♦ Computer Engineering from the University Action Pro-Education and Culture (UNAPEC)

Mr. Gil Contreras, Armando

- ♦ Lead Big Data Scientist at Jhonson Controls
- ♦ Data Scientist-Big Data at Opensistemas S.A.
- ♦ Fund Auditor at Creatividad y Tecnología S.A. (CYTSA)
- ♦ Public Sector Auditor at PricewaterhouseCoopers Auditores.
- ♦ Professional Master's Degree in Data Science at University Center of Technology and Art
- ♦ Professional Máster Degree MBA in International Relations and Business from the Center for Financial Studies (CEF).
- ♦ Bachelor's Degree in Economics from the Technological Institute of Santo Domingo.

Ms. Gil Contreras, Milagros

- ♦ Content Creator at MPCTech LLC
- ♦ Project Manager
- ♦ Freelance IT Writer
- ♦ MBA from the Complutense University of Madrid
- ♦ Degree/Graduate in Business Administration from the Technological Institute of Santo Domingo.

Mr. Delgado Panadero, Ángel

- ♦ ML Engenieer at Paradigma Digital
- ♦ Computer Vision Engineer at NTT Disruption
- ♦ Data Scientist at Singular People
- ♦ Data Analyst at Parclick
- ♦ Specialist in Data Engineering on GPC
- ♦ Specialist in Deep Learning
- ♦ Degree in Physics at the University of Salamanca

Ms. Delgado Feliz, Benedit

- ♦ Administrative Assistant and Electronic Surveillance Operator for the National Drug Control Directorate (DNCD)
- ♦ Customer Service at Cáceres y Equipos
- ♦ Claims and Customer Service at Express Parcel Services (EPS)
- ♦ Microsoft Office Specialist at the National School of Informatics (Escuela Nacional de Informática).
- ♦ Social Communicator from the Catholic University of Santo Domingo.



Take the opportunity to learn about the latest advances in this field in order to apply it to your daily practice"

05

Structure and Content

The syllabus has been meticulously designed to offer a comprehensive experience for graduates. From a deep dive into the syntax and advanced functionalities of Python, to specialization in Object Oriented Programming (OOP), web and mobile application design, and expert data management with libraries such as NumPy and Pandas, each module will be carefully structured to provide solid knowledge and practical skills. In addition, crucial topics such as user interface and user experience (UI/UX) design, advanced data handling, and performance and storage optimization will be explored.



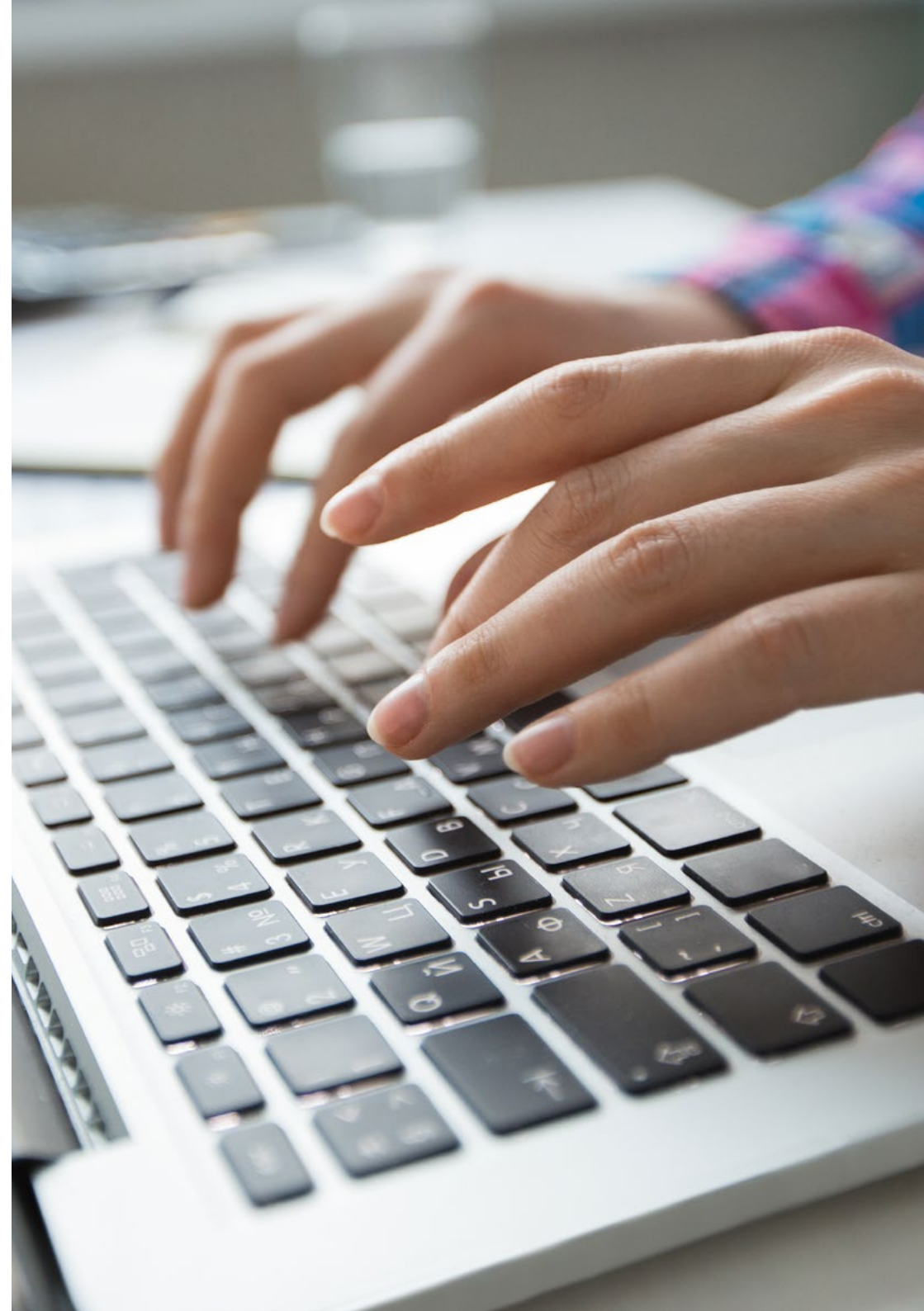


“

Not only will you gain complete mastery in Python, but you'll also prepare yourself to confidently tackle dynamic and tech industry challenges”

Module 1. Python Programming

- 1.1. Creation and Execution of Python Programs
 - 1.1.1. Configuration of the Development Environment
 - 1.1.2. Execution of Python Scripts
 - 1.1.3. Integrated Development Tools (IDEs)
- 1.2. Data in Python
 - 1.2.1. Primitive Types (int, float, str)
 - 1.2.2. Conversion and Casting of Data Types in Python
 - 1.2.3. Immutability and Data Storage in Python
- 1.3. References to Objects in Python
 - 1.3.1. References in Memory
 - 1.3.2. Identity vs. Equality
 - 1.3.3. Reference Management and Garbage Collection
- 1.4. Collection Data in Python
 - 1.4.1. Common Lists and Operations
 - 1.4.2. Tuples and their Immutability
 - 1.4.3. Dictionaries and Data Access
- 1.5. Logical Operations in Python
 - 1.5.1. Boolean Operators
 - 1.5.2. Conditional Expressions
 - 1.5.3. Short-Circuit Evaluation
- 1.6. Arithmetic Operators in Python
 - 1.6.1. Arithmetic Operations in Python
 - 1.6.2. Division Operators
 - 1.6.3. Precedence and Associativity
- 1.7. Input/output in Python
 - 1.7.1. Reading Data from Standard Input
 - 1.7.2. Writing Data to Standard Output
 - 1.7.3. File Handling
- 1.8. Creating and Calling Python Functions
 - 1.8.1. Function Syntax
 - 1.8.2. Parameters and Arguments
 - 1.8.3. Return Values and Anonymous Functions



- 1.9. Using Strings in Python
 - 1.9.1. Manipulating and Formatting Strings
 - 1.9.2. Common Strings Methods
 - 1.9.3. Interpolation and F-strings
- 1.10. Error and Exception Handling in Python
 - 1.10.1. Common Types of Exceptions
 - 1.10.2. Try-except Blocks
 - 1.10.3. Creating Custom Exceptions

Module 2. Advanced Data and Flow Control with Python

- 2.1. Identifiers and Keywords in Python
 - 2.1.1. Variable Naming Rules
 - 2.1.2. Python Reserved Words
 - 2.1.3. Naming Conventions
- 2.2. Integral and Boolean Types in Python
 - 2.2.1. Integral Types
 - 2.2.2. Boolean Specific Operations
 - 2.2.3. Conversions and Representations
- 2.3. Floating-point Types and Complex Numbers in Python
 - 2.3.1. Accuracy and Representation
 - 2.3.2. Floating Point Operations
 - 2.3.3. Use of Complex Numbers in Calculations
- 2.4. String Formatting and Encodings in Python
 - 2.4.1. Advanced Formatting Methods
 - 2.4.2. Unicode and UTF-8 Encodings
 - 2.4.3. Working with Special Characters
- 2.5. Collections: Tuples, Lists and Dictionaries in Python
 - 2.5.1. Comparing and Contrasting Types
 - 2.5.2. Type-specific Methods
 - 2.5.3. Efficiency and Selection of Suitable Type
- 2.6. Sets and Frozen Sets in Python
 - 2.6.1. Creation and Operations in Sets
 - 2.6.2. Frozen Sets
 - 2.6.3. Practical Applications and Performance

- 2.7. Iterating and Copying Collections in Python
 - 2.7.1. For Loops and List Comprehensions
 - 2.7.2. Shallow vs. Deep
 - 2.7.3. Iterators and Generators
- 2.8. Using Lambda Functions in Python
 - 2.8.1. Syntax and Creation of Lambda Functions
 - 2.8.2. Applications in Filters and Maps
 - 2.8.3. Limitations and Best Practices
- 2.9. Control Structures: Conditionals and Loops in Python
 - 2.9.1. If-else and Elif Structures
 - 2.9.2. While and For Loops
 - 2.9.3. Flow Control with Break, Continue and Else
- 2.10. Advanced Functions and Methods in Python
 - 2.10.1. Recursive functions
 - 2.10.2. Higher Order Functions
 - 2.10.3. Function Decorators

Module 3. Object Oriented Programming (OOP) in Python

- 3.1. Object Oriented Programming (OOP) in Python
 - 3.1.1. Classes and Objects
 - 3.1.2. Encapsulation and Abstraction
 - 3.1.3. Object Oriented Programming (OOP) in Python
- 3.2. Creation of Classes and Objects in Python
 - 3.2.1. Classes in Python OOP
 - 3.2.2. Instantiation and Initialization Methods
 - 3.2.3. Attributes and Methods
- 3.3. Attributes and Methods in Python
 - 3.3.1. Instance Attributes vs. Class
 - 3.3.2. Instance, Class and Static Methods
 - 3.3.3. Encapsulation and Information Hiding
- 3.4. Inheritance and Polymorphism in Python
 - 3.4.1. Single and Multiple Inheritance
 - 3.4.2. Overwriting and Method Extensions
 - 3.4.3. Polymorphism and Duck Typing

- 3.5. Properties and Attribute Access in Python
 - 3.5.1. *Getters and Setters*
 - 3.5.2. Decorator *@property*
 - 3.5.3. Access Control and Validation
 - 3.6. Custom Classes and Collections in Python
 - 3.6.1. Creating Collection Types
 - 3.6.2. Special Methods (*__len__*, *__getitem__*)
 - 3.6.3. Custom Iterators
 - 3.7. Aggregation and Composition in Python Classes
 - 3.7.1. Relationships Between Classes
 - 3.7.2. Aggregation vs. Composition
 - 3.7.3. Object Lifecycle Management
 - 3.8. Use of Decorators in Python Classes
 - 3.8.1. Use of Decorators in Python Classes
 - 3.8.2. Class Decorators
 - 3.8.3. Applications and Use Cases
 - 3.9. Abstract Classes and Methods in Python
 - 3.9.1. Abstract Classes
 - 3.9.2. Abstract Methods and Implementation
 - 3.9.3. Use of ABC (*Abstract Base Class*)
 - 3.10. Python OOP Exceptions and Error Handling
 - 3.10.1. Custom Exceptions in Classes
 - 3.10.2. Exception Handling in Methods
 - 3.10.3. Best Practices in Exceptions and OOP
 - 4.3. Dependency and Library Management in Python
 - 4.3.1. Package Management with Pip
 - 4.3.2. Use of Virtual Environments
 - 4.3.3. Resolving Dependency Conflicts
 - 4.4. Design Patterns in Python Development
 - 4.4.1. Creative, Structural and Behavioral Patterns
 - 4.4.2. Practical Application of Patterns
 - 4.4.3. Refactoring and Patterns
 - 4.5. Testing and *Debugging* in Python Applications
 - 4.5.1. *Testing* Strategies (Unitary, Integration)
 - 4.5.2. Use of Testing *Frameworks*
 - 4.5.3. *Debugging* Techniques and Tools
 - 4.6. Security and Authentication in Python
 - 4.6.1. Application Security
 - 4.6.2. Implementation of Authentication and Authorization
 - 4.6.3. Vulnerability Prevention
 - 4.7. Optimization and Performance of Python Applications
 - 4.7.1. Performance Analysis
 - 4.7.2. Code Optimization Techniques
 - 4.7.3. Efficient Resource and Data Management
 - 4.8. Deployment and Distribution of Python Applications
 - 4.8.1. Deployment Strategies
 - 4.8.2. Use of Containers and Orchestrators
 - 4.8.3. Distribution and Continuous Updates
 - 4.9. Maintenance and Updating in Python
 - 4.9.1. Software Lifecycle Management
 - 4.9.2. Maintenance and Refactoring Strategies
 - 4.9.3. System Upgrade and Migration
 - 4.10. Documentation and Technical Support in Python
 - 4.10.1. Creating Effective Documentation
 - 4.10.2. Documentation Tools
 - 4.10.3. Strategies for Supporting and Communicating with Users
- Module 4. Application Development in Python**
- 4.1. Python Application Architecture
 - 4.1.1. Software Design
 - 4.1.2. Common Architectural Patterns
 - 4.1.3. Requirements and Needs Assessment
 - 4.2. Design and Modeling of Python Applications
 - 4.2.1. Use of UML and Diagrams
 - 4.2.2. Modeling Data and Information Flow
 - 4.2.3. SOLID Principles and Modular Design

Module 5. Web and Mobile Development with Python

- 5.1. Web Development with Python
 - 5.1.1. Structure and Components of a Website
 - 5.1.2. Technologies in Web Development
 - 5.1.3. Trends in Web Development
- 5.2. Popular Web Frameworks with Python
 - 5.2.1. Django, Flask and Other Options
 - 5.2.2. Frameworks Comparison and Selection
 - 5.2.3. Frontend Integration
- 5.3. FrontEnd Development: HTML, CSS and JavaScript with Python
 - 5.3.1. HTML and CSS
 - 5.3.2. JavaScript and DOM Manipulation
 - 5.3.3. Frameworks and Frontend Libraries
- 5.4. Backend and Databases with Python
 - 5.4.1. Backend Development with Python
 - 5.4.2. Relational and Non-relational Database Management
 - 5.4.3. Backend-Frontend Integration
- 5.5. APIs and Web Services with Python
 - 5.5.1. Designing RESTful APIs
 - 5.5.2. Implementing and Documenting APIs
 - 5.5.3. API Consumption and Security
- 5.6. Mobile Development with Python
 - 5.6.1. Mobile Development Platforms (Native, Hybrid)
 - 5.6.2. Development Tools and Environments
 - 5.6.3. Adaptation of Applications for Mobile Devices
- 5.7. Mobile Development Platforms with Python
 - 5.7.1. Android and IOS
 - 5.7.2. Frameworks for Cross-development
 - 5.7.3. Testing and Deployment on Mobile Devices
- 5.8. Design and UX in Mobile Applications with Python
 - 5.8.1. Mobile Interface Design
 - 5.8.2. Usability and User Experience with Python
 - 5.8.3. Prototyping and Design Tools

- 5.9. Mobile Testing and Debugging with Python
 - 5.9.1. Testing Strategies on Mobile Devices
 - 5.9.2. Debugging and Monitoring Tools
 - 5.9.3. Test automation
- 5.10. Publishing in Application Stores with Python
 - 5.10.1. Publication Process in App Store and Google Play
 - 5.10.2. Apps Compliance and Policies
 - 5.10.3. Marketing and Promotion Strategies

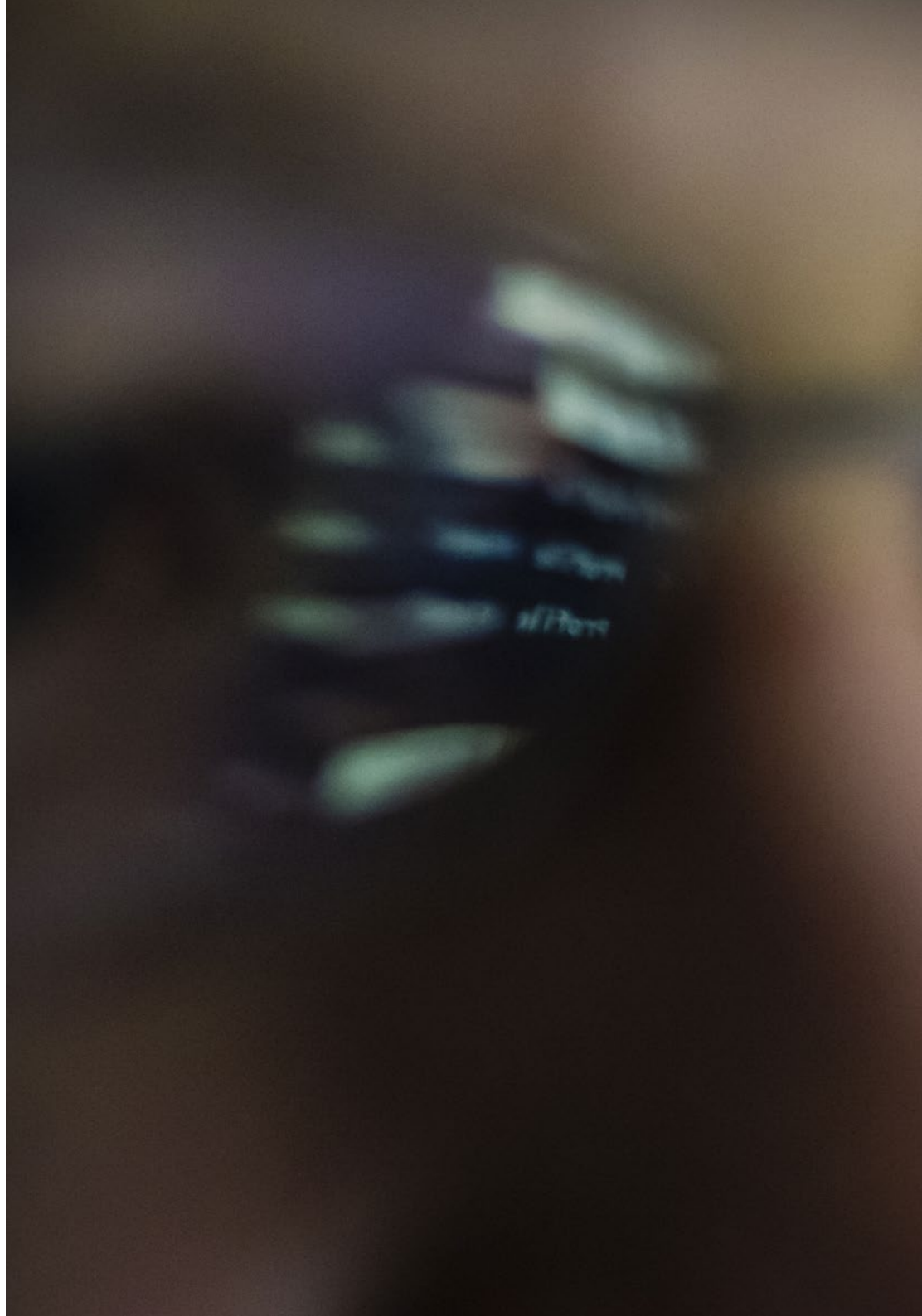
Module 6. User Interface and User Experience with Python

- 6.1. User Interface Design with Python
 - 6.1.1. UI Design with Python
 - 6.1.2. User-computer Interaction with Python
 - 6.1.3. User-centered Design with Python
- 6.2. UI/UX Design Tools with Python
 - 6.2.1. Design and Prototyping Software
 - 6.2.2. Collaboration and Feedback Tools
 - 6.2.3. Integration of Design into the Development Process
- 6.3. Responsive and Adaptive Design with Python
 - 6.3.1. Responsive Design Techniques
 - 6.3.2. Adaptation to Different Devices and Screens
 - 6.3.3. Testing and Quality Assurance
- 6.4. Animations and Transitions with Python
 - 6.4.1. Creating Effective Animations with Python
 - 6.4.2. Tools and Libraries for Animations
 - 6.4.3. Impact on UX and Performance
- 6.5. Accessibility and Usability with Python
 - 6.5.1. Web Accessibility
 - 6.5.2. Evaluation Tools and Techniques
 - 6.5.3. Implementation of Best Practices
- 6.6. Prototyping and Wireframes with Python
 - 6.6.1. Wireframes and Mockups Creation
 - 6.6.2. Rapid Prototyping Tools
 - 6.6.3. Usability *Testing* and *Feedback*

- 6.7. Usability Testing with Python
 - 6.7.1. Usability Testing Methods and Techniques
 - 6.7.2. Results-based Analysis and Improvements
 - 6.7.3. Usability Testing Tools
- 6.8. User Behavior Analysis with Python
 - 6.8.1. Analysis and Tracking Techniques
 - 6.8.2. Data Interpretation and Metrics
 - 6.8.3. Continuous Improvement Based on Data
- 6.9. Feedback-based Enhancements with Python
 - 6.9.1. Feedback Management and Analysis
 - 6.9.2. Feedback Cycles and Continuous Improvement
 - 6.9.3. Strategies for Implementing Effective Change
- 6.10. Future Trends in UI/UX with Python
 - 6.10.1. Innovations and Emerging Trends
 - 6.10.2. Impact of New Technologies on UI/UX
 - 6.10.3. Preparing for the Future of Design

Module 7. Data Processing and *Big Data* with Python

- 7.1. Using Python on Data
 - 7.1.1. Python in Data Science and Analysis
 - 7.1.2. Essential Libraries for Data
 - 7.1.3. Applications and Examples
- 7.2. Setting Up the Python Development Environment
 - 7.2.1. Python Installation and Tools
 - 7.2.2. Configuration of Virtual Environments
 - 7.2.3. Integrated Development Tools (IDE)
- 7.3. Variables, Data Types and Operators in Python
 - 7.3.1. Variables and Primitive Data Types
 - 7.3.2. Data Structures
 - 7.3.3. Arithmetic and Logical Operators
- 7.4. Flow Control: Conditionals and Loops
 - 7.4.1. Conditional Control Structures (if, else, elif)
 - 7.4.2. Loops (for, while) and Flow Control
 - 7.4.3. List Comprehensions and Generator Expressions



- 
- 7.5. Functions and Modularity with Python
 - 7.5.1. Use of Functions
 - 7.5.2. Parameters, Arguments and Return Values
 - 7.5.3. Modularity and Code Reuse
 - 7.6. Error and Exception Handling with Python
 - 7.6.1. Errors and Exceptions
 - 7.6.2. Exception Handling with Try-Except
 - 7.6.3. Creating Custom Exceptions
 - 7.7. Ipython Tool
 - 7.7.1. Ipython Tool
 - 7.7.2. Using IPython for Data Analysis
 - 7.7.3. Differences with the Standard Python Interpreter
 - 7.8. Jupyter Notebooks
 - 7.8.1. Jupyter Notebooks
 - 7.8.2. Use of Notebooks for Data Analysis
 - 7.8.3. Publication of Jupyter Notebooks
 - 7.9. Python Coding Best Practices
 - 7.9.1. Style and Conventions (WBS 8)
 - 7.9.2. Documentation and Comments
 - 7.9.3. Testing and Debugging Strategies
 - 7.10. Python Resources and Communities
 - 7.10.1. Online Resources and Documentation
 - 7.10.2. Communities and Forums
 - 7.10.3. Learning and Updating in Python

Module 8. Data Structures and Functions in Python

- 8.1. Sets in Python
 - 8.1.1. Operations and Methods
 - 8.1.2. Differences and Practical Application
 - 8.1.3. Iteration and Comprehensions
- 8.2. Dictionaries and their Use in Python
 - 8.2.1. Dictionary Creation and Manipulation
 - 8.2.2. Data Access and Management
 - 8.2.3. Patterns and Advanced Techniques

- 8.3. List and Dictionary Comprehensions in Python
 - 8.3.1. Syntax and Examples
 - 8.3.2. Efficiency and Readability
 - 8.3.3. Practical Applications
- 8.4. Functions on Data in Python
 - 8.4.1. Creating Functions
 - 8.4.2. Scope and Namespace
 - 8.4.3. Anonymous and Lambda Functions
- 8.5. Function Arguments and Return Values in Python
 - 8.5.1. Positional and Named Arguments
 - 8.5.2. Multiple Return Values
 - 8.5.3. Variable and Keyword Arguments
- 8.6. Lambda Functions and Higher-Order Functions in Python
 - 8.6.1. Use of Lambda Functions
 - 8.6.2. Map, Filter and Reduce Functions
 - 8.6.3. Data Processing Applications
- 8.7. File Handling in Python
 - 8.7.1. Reading and Writing Files
 - 8.7.2. Handling Binary and Text Files
 - 8.7.3. Best Practices and Exception Handling
- 8.8. Reading and Writing Text and Binary Files in Python
 - 8.8.1. File Formats and Encoding
 - 8.8.2. Handling Large Files
 - 8.8.3. Serialization and Deserialization (JSON, pickle)
- 8.9. Contexts and File Operations
 - 8.9.1. Using the Context Manager (with)
 - 8.9.2. File Processing Techniques
 - 8.9.3. Security and Error Handling
- 8.10. Python Modeling Libraries
 - 8.10.1. Scikit-learn
 - 8.10.2. TensorFlow
 - 8.10.3. PyTorch

Module 9. Data Handling in Python with NumPy and Pandas

- 9.1. Creating and Manipulating Arrays in NumPy
 - 9.1.1. NumPy
 - 9.1.2. Basic Operations with Arrays
 - 9.1.3. Arrays Manipulation and Transformation
- 9.2. Vectorized Operations with Arrays
 - 9.2.1. Vectorization
 - 9.2.2. Universal Functions (ufunc)
 - 9.2.3. Efficiency and Performance
- 9.3. Indexing and Segmentation in NumPy
 - 9.3.1. Access to Elements and Slicing
 - 9.3.2. Advanced and Boolean Indexing
 - 9.3.3. Reordering and Selection
- 9.4. Pandas Series and DataFrames
 - 9.4.1. Pandas
 - 9.4.2. Data Structures in Pandas
 - 9.4.3. DataFrames Manipulation
- 9.5. Indexing and Selection in Pandas
 - 9.5.1. Access to Data in Series and DataFrames
 - 9.5.2. Selection and Filtering Methods
 - 9.5.3. Use of loc e iloc
- 9.6. Operations with Pandas
 - 9.6.1. Arithmetic Operations and Alignment
 - 9.6.2. Aggregation and Statistics Functions
 - 9.6.3. Transformations and Application of Functions
- 9.7. Handling Incomplete Data in Pandas
 - 9.7.1. Detection and Handling of Null Values
 - 9.7.2. Filling and Elimination of Incomplete Data
 - 9.7.3. Strategies for Handling Incomplete Data
- 9.8. Strategies for Handling Incomplete Data
 - 9.8.1. Concatenation and Data Merging
 - 9.8.2. Grouping and Aggregation (groupby)
 - 9.8.3. Pivot Tables and Crosstabs

- 9.9. Visualization with Matplotlib
 - 9.9.1. Matplotlib
 - 9.9.2. Graphics Creation and Customization
 - 9.9.3. Integration with Pandas
- 9.10. Customizing Graphics in Matplotlib
 - 9.10.1. Styles and Settings
 - 9.10.2. Advanced Graphics (scatter, bar, etc.)
 - 9.10.3. Creating Complex Visualizations

Module 10. Advanced Techniques and Practical Applications in NumPy and Pandas

- 10.1. Loading Data from Different Sources
 - 10.1.1. Importing from CSV, Excel and Databases
 - 10.1.2. Reading Data from APIs and Web
 - 10.1.3. Big Data Management Strategies
- 10.2. Data Storage in Python
 - 10.2.1. Exporting to Different Formats
 - 10.2.2. Storage Efficiency
 - 10.2.3. Data Security and Privacy
- 10.3. Data Cleansing Strategies in Python
 - 10.3.1. Identification and Correction of Inconsistencies
 - 10.3.2. Data Normalization and Transformation
 - 10.3.3. Automation of Cleaning Processes
- 10.4. Advanced Data Transformation in Pandas
 - 10.4.1. Manipulation and Transformation Techniques
 - 10.4.2. Combining and Restructuring DataFrames
 - 10.4.3. Use of Regular Expressions in Pandas
- 10.5. Combination of DataFrames in Pandas
 - 10.5.1. Merge, Join and Concatenation
 - 10.5.2. Handling of Conflicts and Keys
 - 10.5.3. Efficient Combination Strategies
- 10.6. Advanced Transformation and Pivoting of Data in Pandas
 - 10.6.1. Pivot and Melt
 - 10.6.2. Reshaping and Transposition Techniques
 - 10.6.3. Applications in Data Analysis
- 10.7. Time Series in Pandas
 - 10.7.1. Handling of Dates and Times
 - 10.7.2. Resampling and Window Functions
 - 10.7.3. Trend and Seasonality Analysis
- 10.8. Advanced Index Management in Pandas
 - 10.8.1. Multilevel and Hierarchical Indexes
 - 10.8.2. Advanced Selection and Manipulation
 - 10.8.3. Query Optimization
- 10.9. Performance Optimization Strategies
 - 10.9.1. Speed and Efficiency Improvements
 - 10.9.2. Use of Cython and Numba
 - 10.9.3. Parallelization and Distributed Processing
- 10.10. Practical Data Manipulation Projects
 - 10.10.1. Development of Real Examples of Use
 - 10.10.2. Integration of Python Techniques
 - 10.10.3. Strategies for Solving Complex Data Problems



This innovative program will adapt to you to keep you up to date with the latest trends and technologies, guaranteeing you to be at the forefront of innovation in software development"

06

Methodology

This academic program offers students a different way of learning. Our methodology uses a cyclical learning approach: **Relearning**.

This teaching system is used, for example, in the most prestigious medical schools in the world, and major publications such as the **New England Journal of Medicine** have considered it to be one of the most effective.



“

Discover Relearning, a system that abandons conventional linear learning, to take you through cyclical teaching systems: a way of learning that has proven to be extremely effective, especially in subjects that require memorization"

Case Study to contextualize all content

Our program offers a revolutionary approach to developing skills and knowledge. Our goal is to strengthen skills in a changing, competitive, and highly demanding environment.

“

At TECH, you will experience a learning methodology that is shaking the foundations of traditional universities around the world”



You will have access to a learning system based on repetition, with natural and progressive teaching throughout the entire syllabus.



The student will learn to solve complex situations in real business environments through collaborative activities and real cases.

A learning method that is different and innovative

This TECH program is an intensive educational program, created from scratch, which presents the most demanding challenges and decisions in this field, both nationally and internationally. This methodology promotes personal and professional growth, representing a significant step towards success. The case method, a technique that lays the foundation for this content, ensures that the most current economic, social and professional reality is taken into account.

“ *Our program prepares you to face new challenges in uncertain environments and achieve success in your career”*

The case method has been the most widely used learning system among the world's leading Information Technology schools for as long as they have existed. The case method was developed in 1912 so that law students would not only learn the law based on theoretical content. It consisted of presenting students with real-life, complex situations for them to make informed decisions and value judgments on how to resolve them. In 1924, Harvard adopted it as a standard teaching method.

What should a professional do in a given situation? This is the question that you are presented with in the case method, an action-oriented learning method. Throughout the course, students will be presented with multiple real cases. They will have to combine all their knowledge and research, and argue and defend their ideas and decisions.

Relearning Methodology

TECH effectively combines the Case Study methodology with a 100% online learning system based on repetition, which combines different teaching elements in each lesson.

We enhance the Case Study with the best 100% online teaching method: Relearning.

In 2019, we obtained the best learning results of all online universities in the world.

At TECH you will learn using a cutting-edge methodology designed to train the executives of the future. This method, at the forefront of international teaching, is called Relearning.

Our university is the only one in the world authorized to employ this successful method. In 2019, we managed to improve our students' overall satisfaction levels (teaching quality, quality of materials, course structure, objectives...) based on the best online university indicators.



In our program, learning is not a linear process, but rather a spiral (learn, unlearn, forget, and re-learn). Therefore, we combine each of these elements concentrically.

This methodology has trained more than 650,000 university graduates with unprecedented success in fields as diverse as biochemistry, genetics, surgery, international law, management skills, sports science, philosophy, law, engineering, journalism, history, and financial markets and instruments. All this in a highly demanding environment, where the students have a strong socio-economic profile and an average age of 43.5 years.

Relearning will allow you to learn with less effort and better performance, involving you more in your training, developing a critical mindset, defending arguments, and contrasting opinions: a direct equation for success.

From the latest scientific evidence in the field of neuroscience, not only do we know how to organize information, ideas, images and memories, but we know that the place and context where we have learned something is fundamental for us to be able to remember it and store it in the hippocampus, to retain it in our long-term memory.

In this way, and in what is called neurocognitive context-dependent e-learning, the different elements in our program are connected to the context where the individual carries out their professional activity.



This program offers the best educational material, prepared with professionals in mind:



Study Material

All teaching material is produced by the specialists who teach the course, specifically for the course, so that the teaching content is highly specific and precise.

These contents are then applied to the audiovisual format, to create the TECH online working method. All this, with the latest techniques that offer high quality pieces in each and every one of the materials that are made available to the student.



Classes

There is scientific evidence suggesting that observing third-party experts can be useful.

Learning from an Expert strengthens knowledge and memory, and generates confidence in future difficult decisions.



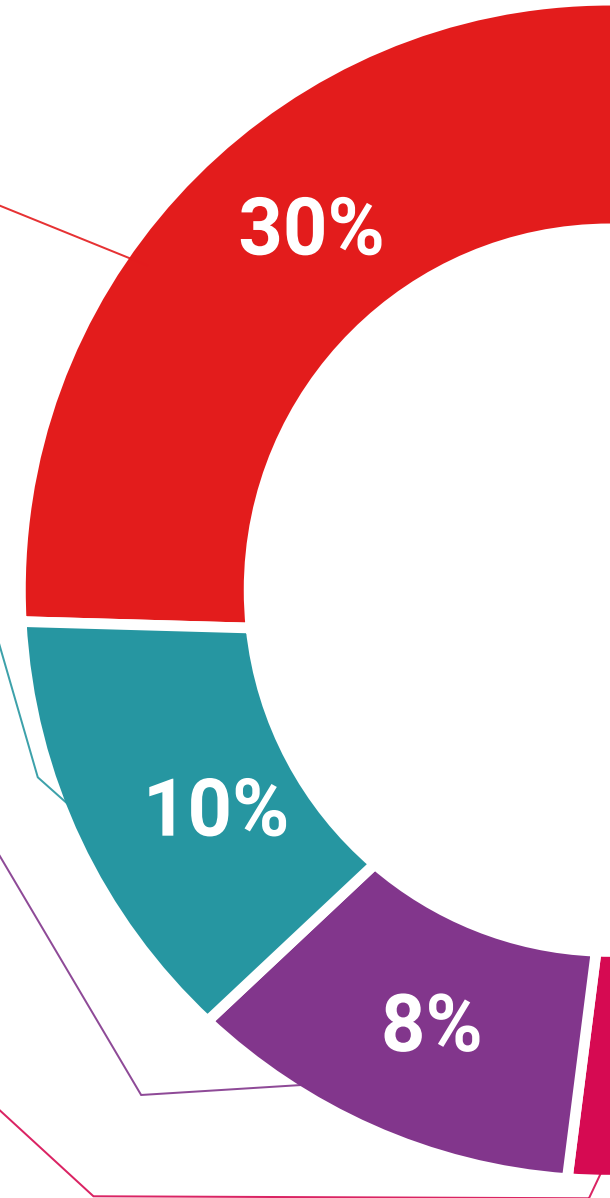
Practising Skills and Abilities

They will carry out activities to develop specific skills and abilities in each subject area. Exercises and activities to acquire and develop the skills and abilities that a specialist needs to develop in the context of the globalization that we are experiencing.



Additional Reading

Recent articles, consensus documents and international guidelines, among others. In TECH's virtual library, students will have access to everything they need to complete their course.





Case Studies

Students will complete a selection of the best case studies chosen specifically for this program. Cases that are presented, analyzed, and supervised by the best specialists in the world.



Interactive Summaries

The TECH team presents the contents attractively and dynamically in multimedia lessons that include audio, videos, images, diagrams, and concept maps in order to reinforce knowledge.

This exclusive educational system for presenting multimedia content was awarded by Microsoft as a "European Success Story".



Testing & Retesting

We periodically evaluate and re-evaluate students' knowledge throughout the program, through assessment and self-assessment activities and exercises, so that they can see how they are achieving their goals.



07

Certificate

The Professional Master's Degree in Python Development guarantees students, in addition to the most rigorous and up-to-date education, access to a Professional Master's Degree diploma issued by TECH Technological University.



“

Successfully complete this program and receive your university qualification without having to travel or fill out laborious paperwork"

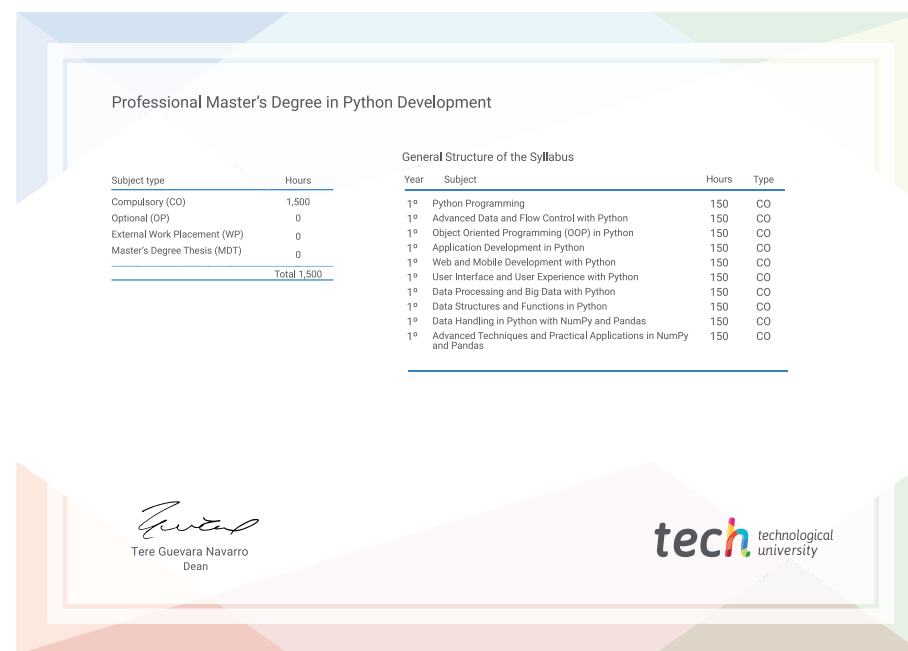
This **Professional Master's Degree in Python Development** contains the most complete and up-to-date program on the market.

After the student has passed the assessments, they will receive their corresponding **Professional Master's Degree** issued by **TECH Technological University** via tracked delivery*.

The diploma issued by **TECH Technological University** will reflect the qualification obtained in the Professional Master's Degree, and meets the requirements commonly demanded by labor exchanges, competitive examinations, and professional career evaluation committees.

Title: **Professional Master's Degree in Python Development**

Official N° of Hours: **1,500 h.**



*Apostille Convention. In the event that the student wishes to have their paper diploma issued with an apostille, TECH EDUCATION will make the necessary arrangements to obtain it, at an additional cost.



Professional Master's Degree Python Development

- » Modality: online
- » Duration: 12 months.
- » Certificate: TECH Technological University
- » Dedication: 16h/week
- » Schedule: at your own pace
- » Exams: online

Professional Master's Degree

Python Development