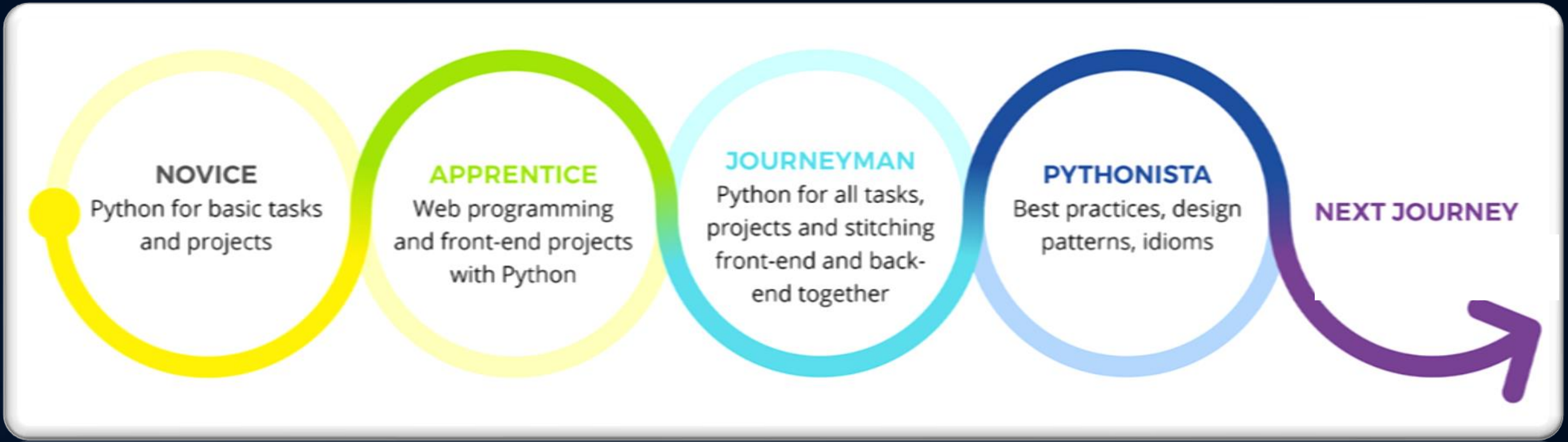
Novice Developer to Pythonista

A CAREER-MAPPED LEARNING JOURNEY

NOVAlearning

Novice Developer to Pythonista

Python continues to be one of the fastest-growing programming languages in the market today. Because of its ease of use and numerous supporting frameworks, it’s widely used in web development, writing scripts, automating tasks, data science, and even cybersecurity. The Python Novice to Pythonista journey takes learners on a path that starts with courses for those with basic math capabilities, Java, HTML, or CSS knowledge. By the end of the journey, the learner will have worked up to wrangling excel data with Python, Network programming, and continual testing strategies. Key technologies are covered such as Python 3.x, JSON, Flask, Anaconda, Jupyter Notebooks, Unittest, PyUnit, PyCharm, and more.



NOVICE
Python for basic tasks and projects

APPRENTICE
Web programming and front-end projects with Python

JOURNEYMAN
Python for all tasks, projects and stitching front-end and back-end together

PYTHONISTA
Best practices, design patterns, idioms

NEXT JOURNEY

This Learning Journey includes:

- More than 65 hours of learning
- Online Mentoring
- 365 days of access

Track 1: Python Novice

Focus areas:

- ✓ Getting Started with Python
- ✓ Linked Lists
- ✓ Arrays
- ✓ Constructs
- ✓ Language Basics
- ✓ Programming Style
- ✓ Frameworkd and Library
- ✓ Object-Oriented Concepts/Exploration
- ✓ Error Handling
- ✓ Basic Testing

Courses in this Track:

- Getting Started with Python: Introduction
- Complex Data Types in Python: Working with Lists & Tuples in Python
- Complex Data Types in Python: Working with Dictionaries & Sets in Python
- Complex Data Types in Python: Shallow & Deep Copies in Python
- Conditional Statements & Loops: If-else Control Structures in Python
- Conditional Statements & Loops: The Basics of for Loops in Python
- Conditional Statements & Loops: Advanced Operations Using for Loops in Python
- Conditional Statements & Loops: While Loops in Python
- Functions in Python: Introduction
- Functions in Python: Gaining a Deeper Understanding of Python Functions
- Functions in Python: Working with Advanced Features of Python Functions

Track 2: Python Apprentice

Focus areas:

- ✓ Working with Complex Data Types
- ✓ List, Strings as Lists
- ✓ Conditional Statements
- ✓ First Class Functions and Lambdas
- ✓ Deep and shallow copies of data
- ✓ Conditional Statements and Loops
- ✓ Loops for rangers, lists dictionaries
- ✓ Break and continue statements
- ✓ Unit Testing

Courses in this Track:

- Advanced Python Topics: File Operations in Python
- Advanced Python Topics: Exceptions & Command Line Arguments
- Advanced Python Topics: Python Modules & Virtual Environments
- Advanced Python Topics: Migrating from Python 2 to Python 3
- Python Classes and Inheritance: Introduction
- Python Classes & Inheritance: Getting Started with Classes in Python
- Python Classes & Inheritance: Working with Inheritance in Python
- Python Classes & Inheritance: Advanced Functionality Using Python Classes
- Data Structures & Algorithms in Python: Fundamental Data Structures
- Data Structures & Algorithms in Python: Implementing Data Structures
- Data Structures & Algorithms in Python: Sorting Algorithms
- Data Structures & Algorithms in Python: Implementing Sorting Algorithms
- Data Structures & Algorithms in Python: Trees & Graphs
- Data Structures & Algorithms in Python: Implementing Trees & Graphs

Track 3: Python Journeyman

Focus areas:

- ✓ Classes and Inheritance in Python
- ✓ Initialization/special methods in classes
- ✓ Parent, derived & abstract base classes
- ✓ Advanced Topics in Python
- ✓ Regx, APIs, Working with files
- ✓ Exception and debugging
- ✓ Member variables
- ✓ Automated Testing Strategies

Courses in this Track:

- Python Unit Testing: An Introduction to Python's unittest Framework
- Python Unit Testing: Advanced Python Testing Using the unittest Framework
- Python Unit Testing: Testing Python Code Using pytest
- Python Unit Testing: Testing Python Code Using doctest
- Python Requests: HTTP Requests with Python
- Python Concurrent Programming: Introduction to Concurrent Programming
- Python Concurrent Programming: Multithreading in Python
- Python Concurrent Programming: Multiprocessing in Python
- Python Concurrent Programming: Asynchronous Executions in Python

Track 4: Pythonista

Focus areas:

- ✓ Advanced Data Structures and Algorithms
- ✓ Complexity/Big-O natation
- ✓ Binary Trees
- ✓ Sorting, searching and algorithms
- ✓ Design Patterns, Python idioms
- ✓ Best Practices & APIs
- ✓ Graph & graph algorithms
- ✓ Continual Testing Strategies

Courses in this Track:

- Introduction to Using PyCharm IDE
- Excel with Python: Working with Excel Spreadsheets from Python
- Excel with Python: Performing Advanced Operations
- Excel with Python: Constructing Data Visualizations
- Socket Programming in Python: Introduction
- Socket Programming in Python: Advanced Topics
- Python Design Patterns: Principles of Good Design
- Python Design Patterns: Working with Creational Design Patterns
- Python Design Patterns: Working with Structural Design Patterns
- Python Design Patterns: Working with Behavioral Design Patterns

Extra Learning:

Productivity Tools for Blockchain Solutions Architect

- ✓ Confluence: Signing in & Navigating within Spaces
- ✓ Confluence: Setting Up & Managing Spaces
- ✓ Confluence: Working with Spaces
- ✓ Confluence: Working with Team Members
- ✓ Confluence: Configuring Spaces
- ✓ Slack Web: Signing in and Setting Up
- ✓ Slack Web: Using Channels
- ✓ Slack 2016: Private Messaging & Communication Tools
- ✓ Slack 2016: Creating, Finding, & Sharing Information
- ✓ Slack 2016: Configuring Slack
- ✓ Slack iOS: Using the iOS App

Business & Leadership for Blockchain Solutions Architect

- ✓ Encouraging Team Communication and Collaboration
- ✓ Using Strategic Thinking to Consider the Big Picture
- ✓ Getting to the Root of a Problem
- ✓ Unleashing Personal and Team Creativity
- ✓ Contributing as a Virtual Team Member
- ✓ Developing a Growth Mind-set
- ✓ Developing a Successful Team
- ✓ Reaching Sound Conclusions