High-Level Program Design

Python Programming Accelerator

GENERAL ASSEMBLY

For Data Practitioners

Python Programming Accelerator

Instructor-Led Onsite or Remote ~40 Hours

Overview:

Data-skilled employees will learn how to use Python — the preferred language for data scientists — to analyze advanced data sets, create accessible web applications, and complete data projects for your business.

Business Outcomes:

- Advance the skills of analysts throughout your organization.
- Increase the productivity of your analysts, allowing them to work with larger data sets.
- Kickstart data initiatives through applied projects.

~40 Hours of Python Training

Python Fundamentals	Intermediate Python	Python for Analytics	Python for Web Development
Learn programming fundamentals and practice coding in Python.	**Level up your Python skills** by adding scripting, modules and APIs to your programming toolkit.	**Practice using Python and Pandas f**or data analysis, and understand data workflow and terminology.	**Use Flask** to develop your own APIs and power user interactions with your data

Why Python Programming?

- **Validated tools and approaches** for programming, data analysis, and web applications, developed in partnership with top organizations.
- **Built with subject matter experts** with experience in Data Analysis and Web Development.
- 40 hours of expert-led, hands-on learning, including:
  - **Projects, labs, and assignments** that mimic real-world tasks and workflows.
  - **Case studies and examples** that demonstrate how businesses use the concepts.
  - **Regular feedback and touchpoints** with instructors and peers to ensure students meeting learning goals.

Learner Personas for Python Programming

This product is specifically designed for the following audiences:

- **Aspiring programmers.** Receive a foundation in Python that can be applied to any project, be it app development, game development, web development, data science, or more.
- **Aspiring data scientists.** Learn Python, the main tool of the trade for data analysis, and Pandas, one of the main data science libraries. Walk away with a finished data analysis application.
- **Aspiring web developers.** Learn Python and get an introduction to Flask, the basics of web applications. Walk away with a finished web application (we'll provide the HTML / CSS).

Projects in Python Programming

What to know about the **projects** for this course:

- Python Fundamentals Project: Build an interactive budget tracker or study flashcard management application
- Data Analysis Project: Conduct in-depth analysis of a data set and communicate your findings with data visualizations
- Web Development Project: Create an web application that accepts incoming user requests and manages a data set

Anatomy of Python Programming

Week	Activities	What's Covered
1	Introduction to Python	Explain the value of Python. Use Jupyter Notebook to execute basic Python programs. Use operators to define and manipulate variables. Differentiate between data types in Python.
	Data Structures	Use lists and list methods to manage collections of data. Use index-based retrieval to access and manipulate list items. Use dictionaries to represent data with multiple properties.
2	Conditionals	Define conditional statements in Python to create logic-driven programs. Use logical operators to enhance conditional statements.
	Loops	Use loops to iterate over code blocks repeatedly. Differentiate between while and for loops. Use list comprehension to generate lists using loop syntax.

Anatomy of Python Programming

Week	Activities	What's Covered
3	Functions	Define functions to encapsulate blocks of code. Use parameters in a function. Understand how to return a value from a function. Create functions that include loops and conditional logic to generate specific return values.
	Modules and Scripting	Understand how libraries, packages, and modules relate to one another. Use import statements to access Python libraries. Use Python scripts to automate tasks. Create Python scripts to read and write to files.
4	Object-Oriented Programming	Define a class. Instantiate an object from a class. Create classes with default instance variables.
	Error Handling and De-Bugging	Identify common errors in Python Read error messages for guidance to fix errors Implement try/except error handling

Anatomy of Python Programming

Week	Activities	What's Covered
5	Python Fundamentals Unit Project	Apply fundamental Python concepts to complete the unit lab.
	Python Fundamentals Review	Solidify and review concepts learned in the Python fundamentals unit
6	Exploratory Data Analysis	Use Pandas to read in a data set. Use DataFrame attributes and methods to investigate a data set's integrity. Apply filters and sorting to DataFrames.
	Data Visualization with Pandas	Explain the characteristics of a great data visualization. Identify when to use a bar chart, pie chart, line chart, scatterplot, or histogram. Use Pandas to implement line charts, bar charts, scatterplots, and histograms.

Anatomy of Python Programming

Week	Activities	What's Covered
7	Cleaning and Combining Data	Use Pandas to handle missing or problematic data values. Identify appropriate cleaning strategies for specific types of data. Use groupby() and JOIN statements to combine data with Pandas. Create insights from data by splitting and combining data segments.
	Data Analysis Lab	Apply Data Analysis skills to a specific data project
8	Data Analysis Review and Next Steps	Solidify and review concepts learned in the Data Analysis unit Identify areas for immediate, further development in Data Analysis
	APIs	Make HTTP requests for data to external API sources. Evaluate API documentation to determine data contents, accessibility, and request formatting.

Anatomy of Python Programming

Week	Activities	What's Covered
9	Server Development with Flask	Create a Flask application that accepts user requests Use routes and route parameters to enable complex user requests
	Flask Templates	Use Flask to render templates with local variables Understand and implement RESTful routes
10	Web Development Unit Lab	Apply Web Development concepts to a specific project
	Web Development Review and Next Steps	Solidify and review concepts learned in the Web Development unit Identify areas for immediate, further development in Web Development

Tools Used in the Course

- Jupyter Notebook
- Google Slides
- A text editor such as VS Code or Sublime

Python Programming

What You'll Learn

Module Overview

Python Fundamentals

20 hours 

Why?

Explore the fundamentals of programming using Python, such as variables, data structures, control flow, functions, and libraries. Learn to create Python scripts that can automate processes and save time!

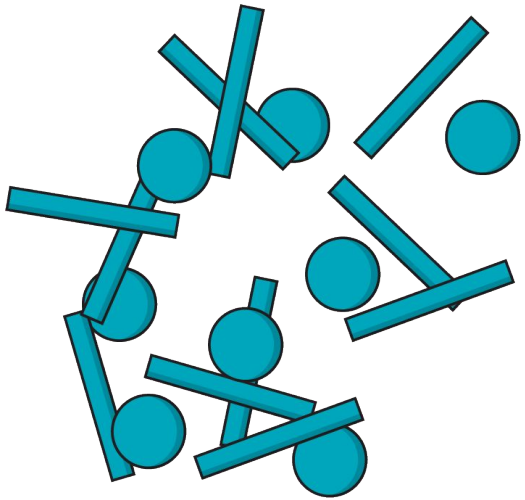
Learning Objectives

- Create Python programs that leverage common programming techniques, such as functions, control flow, and data structures
- Use Python libraries to perform specific tasks such as managing csv files

Our Sad, Unstructured Data

So far, our variables have only stored a single piece of information, or data. Imagine trying to represent a data set using this method:

```
customer_one = "Anees Rosario"
customer_two = "Alya Pham"
customer_three = "Marc Wormald"
customer_four = "Ellie-Mai Muir"
```



You won't get very far using a new variable for every new piece of data!

Putting It All Together

Armed with the foundational building blocks of Python, we can start creating genuinely useful scripts to automate specific tasks, like:

- Filing analysis.
- Sending reporting emails.
- Formatting files.
- Processing data.

Module Overview

Data Analysis

10 hours 

Why?

One of the most common reasons new programmers are learning Python is to perform data analysis with libraries such as Pandas. Learn to generate and communicate insights from any data set using Python.

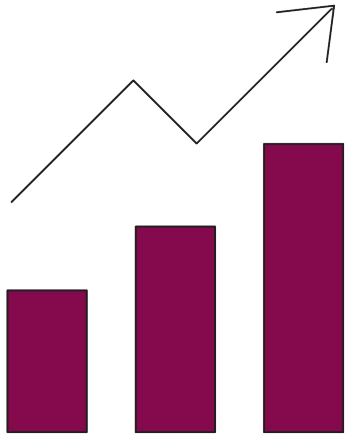
Learning Objectives

- Perform exploratory data analysis using the Pandas library
- Generate and communicate insights using Python for Data Analysis

Exploratory Data Analysis: Definition

In a nutshell, exploratory data analysis (EDA) means "**getting to know" a data set**. This can include:

- **Checking data types** to make sure data is stored properly.
- **Calculating summaries for columns**, like the average, minimum, or maximum.
- Evaluating your data set for **missing data**.
- Identifying potential **trends or outliers**.
- **Basic visualization** of your data.



Criteria for Good Visualizations

1. **Simplified**
   Distill your insight to its essential components.

2. **Easy to Interpret**
   Use logical ordering and understandable metrics.

3. **Clearly Labeled**
   Use clearly distinguishable colors and avoid cluttering text.

Module Overview

Web Development

10 hours 

Why?

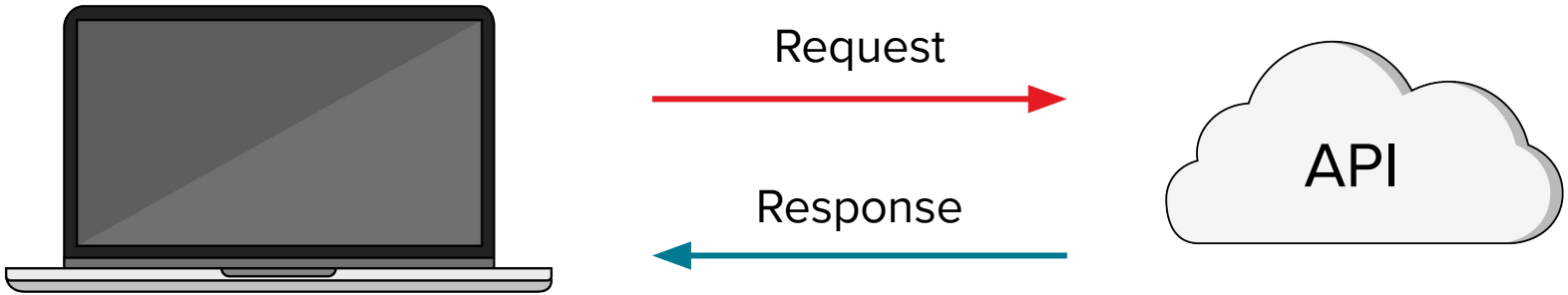
Using the Flask library, we can create dynamic web applications using Python. Web applications allow you to open up your applications to any user with an internet connection, without asking them to run Python code directly.

Learning Objectives

- Create web servers using the Flask library
- Create applications that display data and interfaces to the user with HTML

Welcome to the Neighborhood!

APIs, or **application programming interfaces**, are data sources that can give you access to information from all sorts of places on the internet. Simply put, an API is a system that receives **requests** and returns **responses**.



Group Exercise:
API Routes

30 minutes

Create the seven routes listed in table for a restaurant reviews application. Responses should be strings that describe what the route does.

"/reviews"	GET	index	Show the home page for this collection
"/reviews"	POST	create	Add a new item to a collection
"/reviews/<id>"	GET	show	Show details of a specific item
"/reviews/<id>"	PUT	update	Update details of a specific item
"/reviews/<id>"	DELETE	delete	Delete an item
"/reviews/new"	GET	new	Show the form to create a new item
"/reviews/<id>/edit"	GET	edit	Show the form to update a specific item


GA