# Python Syllabus

**Python Basics:**

* **Introduction to Python:** History, features, where it's used.
* **Environment Setup:** Installing Python, IDEs (like PyCharm, VSCode).
* **Basic Syntax:** Keywords, identifiers, indentation, comments.
* **Variables and Data Types:** Numbers, strings, lists, tuples, dictionaries.
* **Operators:** Arithmetic, comparison, assignment, logical, bitwise, membership, identity operators.
* **Control Flow:** if-else, for loops, while loops, break, continue, pass statements.

**Intermediate Python:**

* **Functions:** Defining functions, arguments, return values, lambda functions.
* **Modules and Packages:** Importing modules, exploring standard modules, creating packages.
* **File Handling:** Opening, reading, writing, and closing files.
* **Exception Handling:** Try, except, finally, raising exceptions.
* **Object-Oriented Programming:** Classes, objects, inheritance, polymorphism, encapsulation, and abstraction.

**Advanced Python:**

* **Iterators and Generators:** Creating iterators, using generators.
* **Decorators:** Understanding decorators, applying them.
* **Context Managers:** With statement, creating context managers.
* **Regular Expressions:** Pattern matching, parsing data.
* **Testing:** Introduction to unit testing with unittest.

**Python Libraries (Focusing on your interests and goals):**

* **NumPy:** Arrays, array operations, mathematical computations.
* **Pandas:** DataFrames, reading and writing data, data manipulation.
* **Matplotlib and Seaborn:** Data visualization, creating plots and charts.
* **Scikit-Learn:** Basic machine learning - regression, classification, clustering.
* **TensorFlow/Keras:** Introduction to deep learning, building simple neural networks.

**Web Development with Django:**

* **Getting Started with Django:** Installation, creating a project, app structure.
* **Models:** Defining models, migrations, querying the database.
* **Admin Interface:** Configuring admin, customizing admin interface.
* **Views and URLs:** Understanding URLconf, writing views, HttpRequest, HttpResponse.
* **Templates:** Django template language, template inheritance.
* **Forms:** Django forms, validation, displaying forms in templates.
* **Authentication:** User authentication, permissions, custom user models.
* **Class-Based Views:** Using generic views, advantages over function-based views.
* **RESTful APIs with Django Rest Framework:** Serializers, viewsets, routers.
* **Testing:** Writing tests for models, views, and forms.
* **Deployment:** Deploying Django applications to servers or platforms like Heroku.