* **What is Python?**

Python is a high-level, general-purpose programming language known for its readability and simplicity. It uses clear syntax similar to English, making it easier to learn and write compared to some other languages.

* **Key Features of python:**
  + **Easy to learn and read:** Encourages rapid development and prototyping.
  + **Interpreted language:** No need for compilation, allowing for faster testing and debugging cycles.
  + **Extensive standard library:** Provides built-in modules for various functionalities.
  + **Large and active community:** Offers abundant resources and support.
* **Use Cases:**
  + **Web development:** Frameworks like Django and Flask power many web applications.
  + **Data science and machine learning:** Popular libraries like NumPy, Pandas, and TensorFlow make it a go-to choice for data analysis and manipulation.
  + **Automation and scripting:** Python automates repetitive tasks, improving efficiency.
  + **Scientific computing:** Libraries like SciPy and Matplotlib support scientific calculations and visualizations.

**Installing Python:**

The installation process varies slightly depending on your operating system. Steps of installing:

1. Go to <https://www.python.org/downloads/>.
2. Download the appropriate installer for your OS.
3. Run the installer and follow the on-screen instructions.

**Verification:**

1. Open a terminal or command prompt.
2. Type python --version and press Enter.   
   If Python is installed correctly, you'll see the installed version number.

**Setting up virtual Environments:**

Virtual environments isolate project dependencies, preventing conflicts between different projects. Here are some popular tools:

* **Windows:** venv module.

**Python Syntax and Semantics:**

A simple Python program that prints "Hello, World!":

Python

Print("Hello, World!")

**Explanation:**

* print: This is a built-in function that outputs data to the console.
* (): Parentheses are used to enclose arguments passed to the function.
* "Hello, World!": This is a string literal enclosed in double quotes.

**Data Types and Variables:**

Python supports various data types to represent different kinds of information:

* **Integers:** Whole numbers (e.g., 10, -5).
* **Floats:** Numbers with decimal points (e.g., 3.14, -1.2).
* **Strings:** Textual data enclosed in quotes (e.g., "Hello", 'World').
* **Booleans:** Represent True or False values.
* **Lists:** Ordered collections of items enclosed in square brackets [] (e.g., [1, 2, 3, "apple"]).
* **Dictionaries:** Unordered collections of key-value pairs enclosed in curly braces {} (e.g., {"name": "Alice", "age": 30}).

Here's a script demonstrating different data types and variables:

Python

# Integer

age = 25

# Float

pi = 3.14159

# String

name = "Bob"

# Boolean

is\_active = True

# List

fruits = ["apple", "banana", "cherry"]

# Dictionary

person = {"name": "Alice", "city": "New York"}

print(f"Age: {age}")

print(f"Pi: {pi}")

print(f"Name: {name}")

print(f"Active: {is\_active}")

print(f"Fruits: {fruits}")

print(f"Person: {person}")

This script creates variables of different data types, assigns values, and then uses f-strings (formatted string literals) to print them with descriptive labels.

**Control Structures:**

* **Conditional Statements (if-else):**

These statements execute code based on certain conditions.

Python

age = 18

if age >= 18:

print("You are eligible to vote.")

else:

print("You are not eligible to vote.")

* **Loops (for):**   
  Loops allow you to repeat a block of code a specific number of times or until a condition is met.

Python

fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

print(fruit)

**Functions in Python:**

Functions are reusable blocks of code that perform specific tasks. They promote code modularity and maintainability.

Python

def sum\_numbers(num1, num2):

"""This function adds two numbers and returns the sum."""

return num1 + num2

result = sum\_numbers(5, 3)

print(f"The sum is: {result}")

**Lists and Dictionaries:**

* **Lists:** Ordered, mutable collections of elements enclosed in square brackets []. You can access elements by index (starting from 0).
* **Dictionaries:** Unordered collections of key-value pairs enclosed in curly braces {}. You can access values using their keys.

Python

# List

numbers = [1, 2, 3, 4, 5]

print(numbers[2]) # Access third element (index 2)

# Dictionary

person = {"name": "Charlie", "age": 30}

print(person["name"]) # Access value using key "name"

**Exception Handling:**

Exception handling allows you to gracefully handle errors that might occur during program execution.

Python

try:

# Code that might raise an exception

num = int("hello") # This will raise a ValueError

except ValueError:

print("Invalid input. Please enter a number.")

finally:

# Code that always executes, regardless of exceptions

print("This block always runs.")

**Modules and Packages:**

* **Modules:** Python files containing reusable functions, variables, and classes. You can import specific functionalities from modules.
* **Packages:** Hierarchies of modules organized into directories. You use import statements to import modules or entire packages.

Python

import math

print(math.pi) # Access the value of pi from the math module

# Example with a package (assuming a package named 'geometry' exists with a module 'rectangle.py')

from geometry.rectangle import calculate\_area

length = 5

width = 3

area = calculate\_area(length, width)

print(f"Area of rectangle: {area}")

**File I/O:**

* **Reading from Files:**

Python

with open("data.txt", "r") as file:

content = file.read()

print(content)

* **Writing to Files:**

Python

data\_to\_write = [" line 1", " line 2", " line 3"]

with open("output.txt", "w") as file:

file.writelines(data\_to\_write)