Introduction to Python

Python is a popular, versatile programming language that's easy to use and understand.

Key Facts:

- **What it is:** Python is a **high-level programming language** that can be used for many different tasks (general purpose).
- **Who made it:** It was created by **Guido van Rossum**.
- **When it started:** He started working on it in **1989** in the Netherlands.
- **Official Launch:** Python was officially released to the public on **February 20, 1991**.
- **For Beginners:** It's highly recommended as the **first programming language** for people who are just starting out.

Eg1: To print the sum of 2 numbers

Java Example:-

```
public class SumNumbers {

   public static void main(String[] args) {

      int number1 = 10;

      int number2 = 5;

      int sum = number1 + number2;

      System.out.println("The sum is: " + sum);

   }

}
```

Python Example

```
number1 = 10

number2 = 5

sum_result = number1 + number2

print("The sum is:", sum_result)
```

Where we can use Python:

- Web Development
- Data Science & Machine Learning
- Artificial Intelligence
- Automation & Scripting
- Game Development
- Cybersecurity
- Mobile & Software Development

Features of Python

---

1. Freeware and Open Source

- **Freeware:** You don't have to pay to use it.
- **Open Source:** Its source code is available to the public, meaning anyone can view, modify, and improve it.

---

2. Simple and Easy to Learn

- Python's code uses plain English keywords and a clear structure, making it much **easier to read and write** than many other languages.

---

3. Platform Independent

- Python code written on one operating system (like Windows) will run on another (like macOS or Linux) **without needing changes**. It's the same code everywhere.

---

4. High-Level Programming Language

- You focus on **what to do** (the logic) rather than **how to do it** (managing computer memory). It's close to human language and far from machine language.

---

5. Portability

- This is very similar to being Platform Independent. It means you can easily **move your Python program** from one machine to another, and it will still work.

---

6. Dynamically Typed

- You **don't have to specify the type** of a variable (like `integer` or `string`) when you declare it. Python figures out the type automatically during execution.
Example: You just write `x = 10`, not `int x = 10`.

---

7. Interpreted

- Python code is **executed line-by-line** by an interpreter, not compiled all at once into a machine-readable file like Java or C++. This makes debugging easier.

---

8. Procedure Oriented and Object Oriented

- Python supports **multiple programming styles**:
  - **Procedure Oriented:** Writing code using functions and procedures.
  - **Object Oriented:** Organizing code using real-world **objects** and classes.

---

9. Extensive Library

- Python has a **huge collection of pre-written code** (modules and libraries) for almost any task—from web development and data analysis to machine learning. This means you don't have to write everything from scratch.

Python Identifiers

An **Identifier** is simply a **name** given to an element in a Python program. This name is used to identify things like:

- **Variables** (e.g., `a = 10`)
- **Functions**
- **Classes**
- **Modules**

Rules for Defining Identifiers:

---

1. Allowed Characters

Only letters, numbers, and the underscore (_) are permitted.

Rule	Example	Status	Explanation
Correct	user_name_25	$\checkmark$	Uses only allowed characters.
Incorrect	user-name	$\times$	Hyphens (-) are not allowed.
Incorrect	data#set	$\times$	Special symbols like # are not allowed.

---

2. Cannot Start with a Digit

The name must start with a letter or an underscore.

Rule	Example	Status	Explanation
Correct	_max_value	$\checkmark$	Starts with an underscore.
Correct	score99	$\checkmark$	Starts with a letter.
Incorrect	7days	$\times$	Cannot start with the digit 7.

---

3. Case Sensitivity

Python treats capitalization differently.

Code	Value	Explanation
PI = 3.14	3.14	Commonly used for a constant value.
Pi = 3.0	3.0	A completely different variable from PI.
pi = 2.71	2.71	A third, separate variable.

- If you access PI, you get 3.14.
- If you access pi, you get 2.71.

Python Reserved Words (Keywords)

Reserved Words are special words in Python that have a fixed meaning and predefined functionality.

- They are reserved by the language for internal purposes and cannot be used as Identifiers (names for variables, functions, etc.).
- There are 33 reserved words in Python (as of recent versions, though the exact count can occasionally change with major updates).

Python Keywords (35 Total)

Keywords (35)				
False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield
match	case	:=		

Important Notes:

- Case Sensitivity: All keywords are in lowercase except for `True`, `False`, and `None`. You must use them with the correct capitalization.
- Recent Additions: `async` and `await` are used for asynchronous programming, and `match`, `case`, and `:=` are for structural pattern matching and assignment expressions, respectively.