PYTHON WORKBOOK – SECTION 2

Variables, Data Types & Operators

Programmer's Hub – by CodeWithVivek
https://www.youtube.com/@code-with-vivek

2.1 Variables in Python

Quick Explanation

A **variable** is a name that stores a value in memory.

Examples:

name = "Vivek"

age = 30

is_student = True

Variables can change during program execution.

Python variables:

- do **not** need a declared type
- are **case-sensitive**
- must start with a **letter or underscore**

Try This:

Write THREE valid Python variable names and assign values to them:

1. _____

2. _____

3. _____

Debug This:

Why is this invalid?

2name = "Aisha"

Your explanation: _____

2.2 Data Types

Core Python Data Types

Type	Example	Description
int	10	Whole numbers
float	3.14	Decimal numbers
str	"Hello"	Text
bool	True	Logic values True/False
NoneType	None	No value
complex	3+4j	Mathematical complex numbers

Use type() to check any value:

```
print(type(10))      # int
print(type("Python"))  # str
```

Your Turn:

Write the type of each value:

1. "123" → _____
2. 123 → _____
3. 12.0 → _____
4. True → _____

Do You Understand?

✓ What is the difference between "50" and 50?
✓ Why is Python called “dynamically typed”?

2.3 Type Conversion (Typecasting)

Explanation

Use these functions to convert types:

int()

float()

str()

bool()

Example:

num = "15"

num = int(num)

Try This:

Convert user input into a number and multiply it by 10:

Write your code here:

Debug This:

Why does this code fail?

age = int("twenty")

Hint: _____

2.4 Arithmetic Operators

Operators in Python

Operator	Meaning	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division (float)	a / b
//	Floor division	7 // 2 = 3
%	Remainder	7 % 3 = 1
**	Exponent	2 ** 3 = 8

Your Turn:

Calculate:

1. 17 // 3 = _____
2. 17 % 3 = _____
3. 2 ** 5 = _____

Do You Understand?

✓ What is the difference between / and //?

2.5 Relational & Logical Operators

Relational Operators

== != > < >= <=

Example:

```
5 > 3   # True
```

Logical Operators

and or not

Example:

```
age > 18 and age < 60
```

Try This:

Write expressions that evaluate to:

1. **True:** _____
2. **False:** _____

Debug This:

Why does this show an error?

```
print(5 > "3")
```

2.6 f-Strings (Formatted Strings)

Explanation

f-strings let you embed variables inside text:

```
name = "Vivek"
print(f"My name is {name}.")
```

You can also calculate inside f-strings:

```
print(f"5 + 3 = {5 + 3}")
```

Your Turn:

Write an f-string that prints:

"My name is __ and I am __ years old."

```
# Write here:
```

Do You Understand?

✓ Why are f-strings better than string concatenation?

Section Summary

✓ Variables store data using names
✓ Python has multiple built-in data types
✓ Use typecasting to convert values
✓ Arithmetic operators perform calculations
✓ Relational & logical operators compare values
✓ f-strings make formatted output easy

Mini Assignment

Create a program:

1. Ask the user's name
2. Ask their marks in 3 subjects
3. Calculate the **total** and **average**
4. Print the report using an **f-string**
5. Use at least one logical operator

Write your draft here: