

## Python Course Recap Notebook

Welcome to the recap of all sessions (1–12).

This notebook covers definitions, examples, assignments, and practice problems.

Let's quickly revise everything we've learned step by step 🔄.

### ✓ Module 1: Getting Started

#### ✓ Session 1: Getting Started with Python

- What is Python?
- Features and applications
- Interpreter vs. Script
- Writing and running your first Python program

```
# Example: A classic first program
print("Hello, Python!")
```

#### ✓ Assignment

Write a program that prints your name and a personal greeting.

#### ✓ Session 2: Variables, Data Types & I/O

- Variables, constants, identifiers
- Primary data types: int, float, str, bool
- Taking input and printing output

```
# Example
user_name = input("What's your name? ")
user_age = int(input("How old are you? "))
print(f"Hello, {user_name}! You are a {type(user_age)}.")
```

#### ✓ Assignment

Ask the user for their favorite food and a quantity. Print: `You want to eat 5 units of pizza`.

### ✓ Module 2: Control Flow

#### ✓ Session 3: Conditionals

- if, elif, else
- Nested if
- Shorthand if

```
# Example
score = 50
if score >= 90:
    print("Excellent!")
elif score >= 80:
    print("Good job!")
else:
    print("Keep practicing.")
```

Keep practicing.



Check if a number is positive, negative, or zero.



- For loop
- While loop

```
# Example  
for i in range(3):  
    print(f"For loop count: {i}")  
  
count = 0  
while count < 1000000000000000000:  
    print(f"While loop count: {count}")  
    count += 1
```



```
nums = [1, 2, 3, 4, 5]
total = 0

for n in nums:
    total += n

print("Final Total:", total)
```

Step	n (current loop value)	total (before)	Operation ( $\text{total} += n$ )	total (after)
Start	-	0	-	0
1st loop	1	0	$0 + 1$	1
2nd loop	2	1	$1 + 2$	3
3rd loop	3	3	$3 + 3$	6
4th loop	4	6	$6 + 4$	10
5th loop	5	10	$10 + 5$	15



Use a for loop to print every other number from 0 to 20.



- Loop control: break, continue, pass
- Nested loops
- Pattern printing

```
# Example
for i in range(5):
    if i == 3:
        continue
    print(i)
```



Use nested loops to print a pyramid pattern.



- Lists: creation, indexing, slicing, methods, comprehensions

- Tuples: immutable sequences

```
# Example
my_list = [1, 2, 3, 4]
my_list.append(5)
print(my_list[1:3])

my_tuple = (10, 20, 30)
print(my_tuple)
```

### ✓ Assignment

Create a list of numbers.

Use a comprehension to create a new list with squares of even numbers.

## ✓ Session 7: Dictionaries & Sets

- Dictionaries: key-value pairs
- Sets: operations, uniqueness

```
# Example
student = {"name": "Bob", "grade": 92}
print(student["name"])

colors = {"red", "green", "blue"}
colors.add("yellow")
print(colors)
```

### ✓ Assignment

Create a dictionary of 5 countries and capitals.

Ask the user for a country and print its capital.

## ✓ Session 8: Iteration & Practice

- Iterating collections: zip, enumerate, items
- Practice with problems

```
# Example
data = {"apples": 3, "bananas": 5}
for item, count in data.items():
    print(f"I have {count} {item}.")
```

### ✓ Assignment

Solve the "Runner-Up Score" problem on HackerRank.

## ✓ Module 4: Functions

## ✓ Session 9: Defining Functions

- Defining and calling functions
- Arguments: positional, keyword, default
- Return values and multiple returns

```
# Example
def calculate_area(length, width):
    return length * width

area = calculate_area(5, 10)
print(area)
```

### ✓ Assignment

Write a function that takes a name and optional age.

If age is given, print both. Otherwise, only the name.

## ✓ 📖 Session 10: Advanced Functions

- Arbitrary arguments (\*args, \*\*kwargs)
- Scope and lifetime of variables
- Lambda, map, filter

```
# Example
nums = [1, 2, 3, 4, 5]
squared = list(map(lambda x: x**2, nums))
print(squared)
```

### ✓ Assignment

Write a function that accepts an arbitrary number of numbers and returns their sum.

## ✓ 💎 Module 5: Advanced Python Concepts

## ✓ 📖 Session 11: Exception Handling & Comments

- try, except, finally
- Comments and docstrings

```
# Example
try:
    num = int("abc")
except ValueError:
    print("Invalid input! Please enter a number.")
```

### ✓ Assignment

Write a program that asks for a number.

Use try-except to handle `ValueError`.

## ✓ 📖 Session 12: File Handling

- Reading and writing files
- The with statement

```
# Example
with open("example.txt", "w") as f:
    f.write("Hello, world!")

with open("example.txt", "r") as f:
    content = f.read()
    print(content)
```

### ✓ Assignment

Create a text file with a list of names.

Write a program to read and print each name line by line.