

Python Basics

Topics

- Type conversion
- Representation of Variables in Memory
- Input and Output Functions
- Comments

Type conversion

Implicit type casting Python

```
# Python automatically converts  
# a to int  
a = 7  
print(type(a))
```

```
# Python automatically converts  
# d to float as it is a float multiplication  
d = a * b  
print(d)  
print(type(d))
```

```
# Python automatically converts  
# c to float as it is a float addition  
c = a + b  
print(c)  
print(type(c))
```

```
# Python automatically converts  
# b to float  
b = 3.0  
print(type(b))
```

The output of implicit type casting

```
<class 'int'>  
<class 'float'>  
10.0  
<class 'float'>  
21.0  
<class 'float'>
```

Explicit Type Conversion

1. `int()` : convert any type variable to the integer type.
2. `float()` : convert any type variable to the float type.
3. `complex()` : convert any type variable to the complex type.
4. `bool()` : convert any type variable to the bool type.
5. `str()` : convert any type variable to the string type.

Int type conversion

Casting float value to an integer

```
pi = 3.14 # float number
print(type(pi))
# Output class 'float'

# converting float integer
num = int(pi)
print("Integer number:", num)
# Output 3
print(type(num))
# Output class 'int'
```

Casting boolean value to an integer

```
flag_true = True
flag_false = False
print(type(flag_true))
# Output class 'bool'

# converting boolean to integer
num1 = int(flag_true)
num2 = int(flag_false)

print("Integer number 1:", num1)
# Output 1
print(type(num1))
# Output class 'int'

print("Integer number 2:", num2)
# 0
print(type(num2))
# class 'int'
```

Casting a string to an integer

```
string_num = "225"  
print(type(string_num))  
# Output class 'str'  
  
# converting str to integer  
num1 = int(string_num)  
  
print("Integer number 1:", num1)  
# Output 225  
print(type(num1))  
# Output class 'int'
```

Wrong string to an integer casting

```
string_num = 'Score is 25'
print(type(string_num))
# Output class 'str'

# ValueError: invalid literal for int() with base 10: 'Score is 25'
num = int(string_num)
print(num)
```

Float type conversion

Casting integer to float

```
num = 725
print(type(num))
# Output class 'int'

# converting float to integer
num1 = float(num)

print("Float number:", num1)
# Output 725.0
print(type(num1))
# Output class 'float'
```

Casting Boolean to float

```
flag_true = True
flag_false = False
print(type(flag_true))
# Output class 'bool'

# converting boolean to float
num1 = float(flag_true)
num2 = float(flag_false)

print("Float number 1:", num1)
# Output 1.0
print(type(num1))
# class 'float'

print("Float number 2:", num2)
# Output 0.0
print(type(num2))
# class 'float'
```

Casting string to float

```
string_num = "725.535"
print(type(string_num))
# Output class 'str'

# converting str to float
num1 = float(string_num)

print("Float number:", num1)
# Output 725.535
print(type(num1))
# class 'float'
```

Complex type conversion

Two forms of complex function

```
complex(x)
```

```
complex(x,y)
```

Casting integer type to complex

```
r_num = 135
print(type(r_num)) # class 'int'

# converting int to complex(x)
c_num = complex(r_num)

print("Complex number:", c_num)
# Output (135+0j)
print(type(c_num))
# class 'complex'

# converting int to complex(x, y)
r_num, i_num2 = 135, 235
c_num = complex(r_num, i_num2)

print("Complex number:", c_num)
# Output (135+235j)
print(type(c_num)) # class 'complex'
```

Casting float type to complex

```
r_num = 53.250
print(type(r_num)) # class 'float'

# converting float to complex(x)
c_num = complex(r_num)

print("Complex number:", c_num)
# Output (53.25+0j)
print(type(c_num))
# class 'complex'

# converting float to complex(x, y)
r_num, i_num2 = 53.250, 350.750
c_num = complex(r_num, i_num2)

print("Complex number:", c_num)
# Output (53.25+350.75j)
print(type(c_num))
# class 'complex'
```

Casting Boolean type to complex

```
boolean_true = True
print(type(boolean_true))  # class 'bool'

# converting boolean to complex(x)
c_num = complex(boolean_true)

print("Complex number:", c_num)
# Output (1+0j)
print(type(c_num))
# class 'complex'

# converting boolean to complex(x, y)
r_bool, i_bool = False, True
c_num = complex(r_bool, i_bool)

print("Complex number:", c_num)
# Output 1j
print(type(c_num))
# class 'complex'
```

Bool type conversion

Cating integer to Boolean

```
num1 = 10
num2 = 0
print(type(num1))  # class 'int'

# Convert into to bool
b1 = bool(num1)
b2 = bool(num2)

print(b1)
# Output True
print(b2)
# Output False

print(type(b1))
# class 'bool'
```

Casting float to Boolean

```
f_num1 = 25.35
f_num2 = 0.0
print(type(f_num1)) # class 'float'

# Convert float into to bool
b1 = bool(f_num1)
b2 = bool(f_num2)

print(b1)
# Output True

print(b2)
# Output False

print(type(b1))
# Output class 'bool'
```

Casting complex type to boolean

```
c1 = 33 + 9j
c2 = 0 + 0j
print(type(c1)) # class 'complex'

# Convert complex value into to bool
b1 = bool(c1)
b2 = bool(c2)

print(b1) # True
print(b2) # False
print(type(b1)) # class 'bool'
```

Casting string to boolean

```
s1 = "False"
s2 = "True"
s3 = "812"
s4 = ""
print(type(s1)) # class 'str'

# Convert string into to bool
b1 = bool(s1)
b2 = bool(s2)
b3 = bool(s3)
b4 = bool(s4)

print(b1) # True
print(b2) # True
print(b3) # True
print(b4) # False
print(type(b1)) # class 'bool'
```

String type conversion

Casting integer to string

```
num = 15
print(type(num))  # class 'int'

# converting int to str type
s1 = str(num)
print(s1)
# Output '15'
print(type(s1))
# Output class 'str'
```

Cast float to string

```
num = 75.35
print(type(num)) # class 'float'

# converting float to str type
s1 = str(num)
print(s1)
# Output '75.35'
print(type(s1))
# Output class 'str'
```

Casting complex type to string

```
complex_num = 15 + 5j
print(type(complex_num)) # class 'complex'

# converting complex to str type
s1 = str(complex_num)
print(s1)
# Output '(15+5j)'

print(type(s1))
# class 'str'
```

Casting boolean to string

```
b1 = True
b2 = False
print(type(b1)) # class 'bool'

# converting bool to str type
s1 = str(b1)
s2 = str(b2)
print(s1)
# Output 'True'
print(s2)
# Output 'False'
print(type(s1)) # class 'str'
```

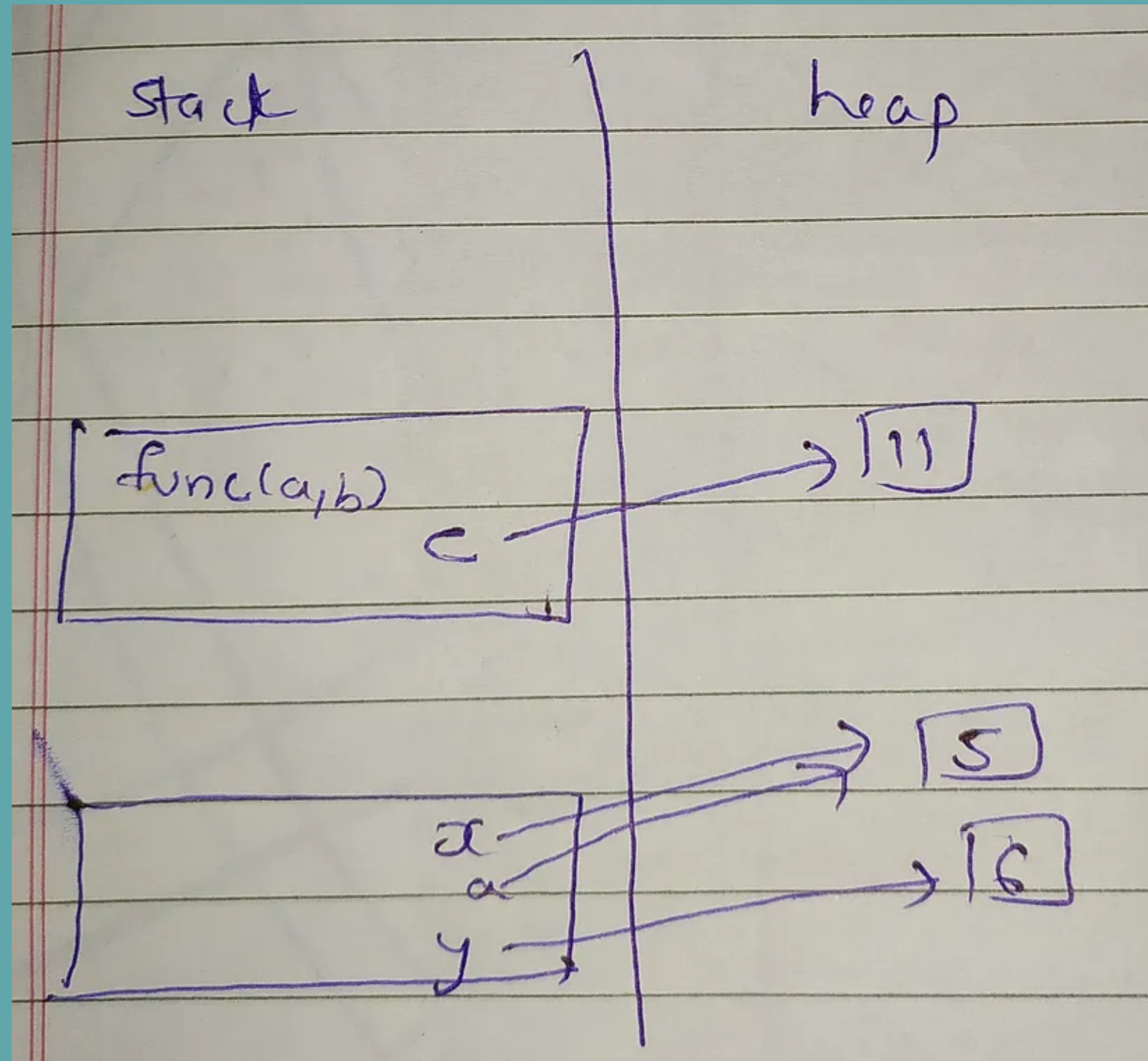
Representation of variables in memory

Let`s start with the theory

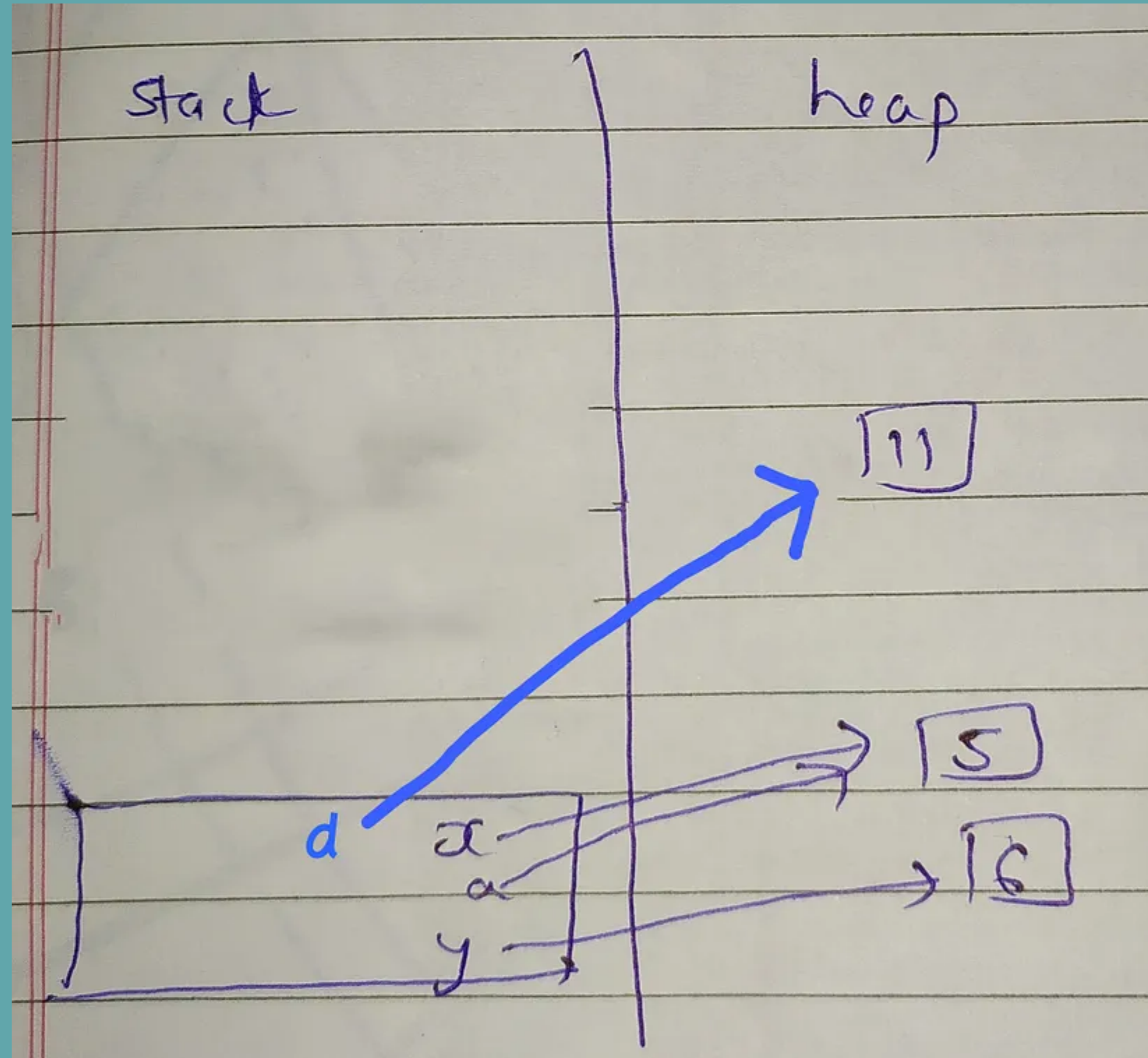
A simple Python program

```
1
2     x = 5
3     y = 6
4     a = 5
5
6     ✓ def func(a,b):
7         |     c = a+b
8         |     return c
9
10    d = func(x,y)
11
12
```

Memory representation



Memory representation



Code example

```
x = 5
```

```
y = 6
```

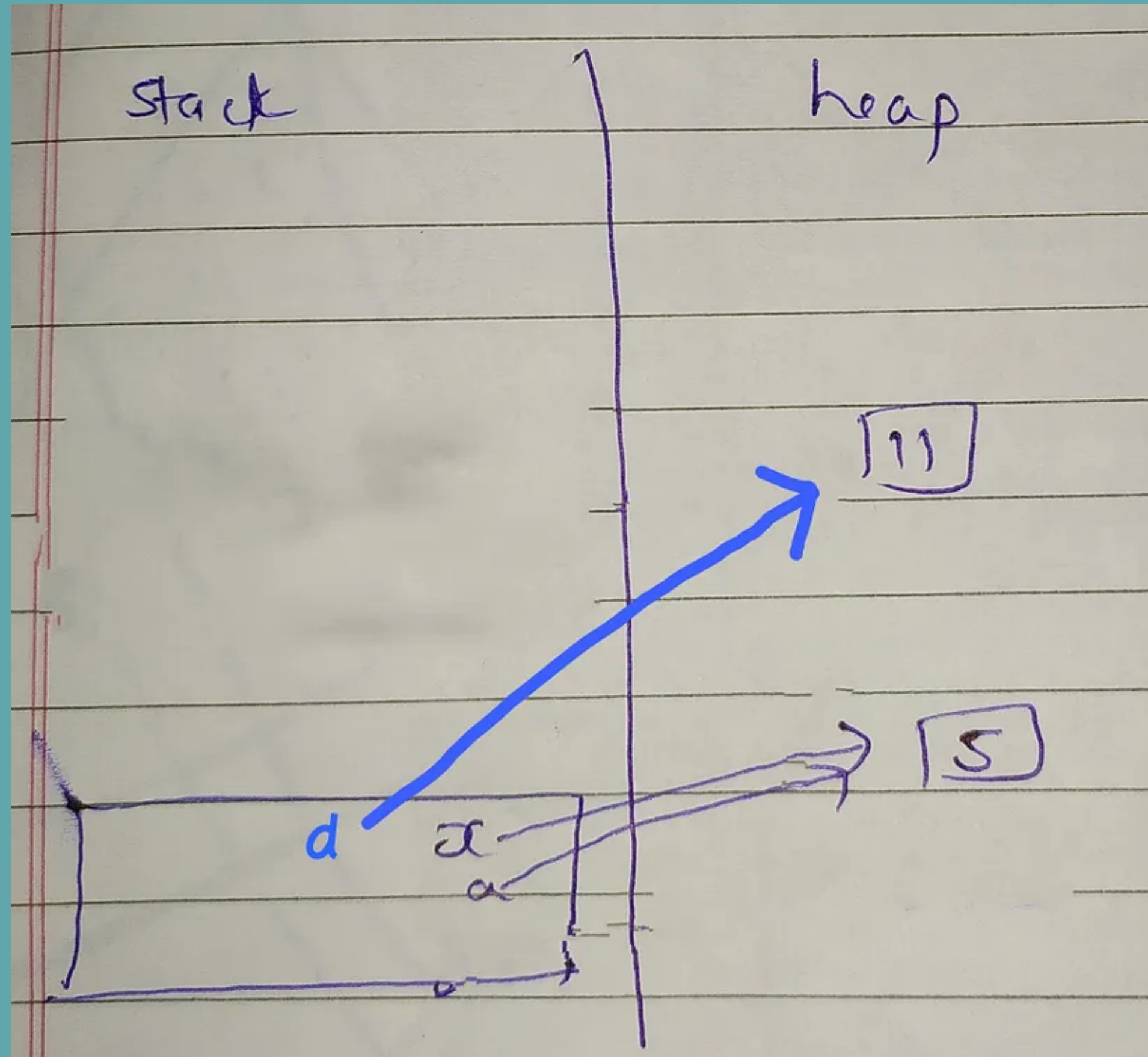
```
a = 5
```

```
✓ def func(a,b):  
    c = a+b  
    return c
```

```
d = func(x,y)
```

```
y = None
```

Memory representation



Input and output functions

Print function

```
print('Python is powerful')
```

```
# Output: Python is powerful
```

Syntax of print

```
print(object= separator= end= file= flush=)
```

- object - value(s) to be printed
- sep (optional) - allows us to separate multiple objects inside print().
- end (optional) - allows us to add add specific values like new line "\n", tab "\t"
- file (optional) - where the values are printed. It's default value is sys.stdout (screen)
- flush (optional) - boolean specifying if the output is flushed or buffered.
Default: False

Example 1

```
print('Good Morning!')  
print('It is rainy today')
```

```
Good Morning!  
It is rainy today
```

Example 2

```
# print with end whitespace  
print('Good Morning!', end= ' ')  
  
print('It is rainy today')
```

Good Morning! It is rainy today

Example 3

```
print('New Year', 2023, 'See you soon!', sep= '. ')
```

```
New Year. 2023. See you soon!
```

Print python Variables and Literals

```
number = -10.6

name = "Programiz"

# print literals
print(5)

# print variables
print(number)
print(name)
```

```
5
-10.6
Programiz
```

Print concatenated Strings

```
print('Programiz is ' + 'awesome.')
```

```
Programiz is awesome.
```

Output formatting

```
x = 5  
y = 10  
  
print('The value of x is {} and y is {}'.format(x,y))
```

Python input

```
input(prompt)
```

Example of Python User input

```
# using input() to take user input
num = input('Enter a number: ')

print('You Entered:', num)

print('Data type of num:', type(num))
```

```
Enter a number: 10
You Entered: 10
Data type of num: <class 'str'>
```

Casting input type

```
num = int(input('Enter a number: '))
```

Comments

Creating a Comment

```
#This is a comment  
print("Hello, World!")
```

Comments in the end of the line

```
print("Hello, World!") #This is a comment
```

Commenting code

```
#print("Hello, World!")  
print("Cheers, Mate!")
```

Multiline comments

```
#This is a comment  
#written in  
#more than just one line  
print("Hello, World!")
```

Multiline comments

```
"""  
This is a comment  
written in  
more than just one line  
"""  
  
print("Hello, World!")
```

Multiline comments

```
'''  
This is a comment  
written in  
more than just one line  
'''  
print("Hello, World!")
```

Thank you all)

Practical Tasks

1. Hello, Python!

- Write a Python program that prints "Hello, Python!" to the console.

2. Simple Calculator:

- Create a basic calculator program that asks the user to input two numbers and performs addition on them. Display the result.

3. Convert Temperature:

- Write a program that converts a temperature from Celsius to Fahrenheit. Ask the user to enter a temperature in Celsius and display the equivalent temperature in Fahrenheit.

4. Check Even or Odd:

- Write a Python program that takes an integer input from the user and determines whether it is even or odd.

5. Personal Information:

- Create a program that asks the user to input their name, age, and favorite color. Display a message that includes this information.

Easy Home Work Assignment

Assignment 1: Creating a Personal Chatbot

Create a simple chatbot in Python that responds to questions and executes user commands.

Requirements:

- Write a program that greets the user and asks for their name.
- The program should respond to questions like "How are you?" or "What's your name?"
- Include at least three questions and responses to make the chatbot more interesting.

Hint: Use conditional statements (if, elif, else) to determine the bot's response to different questions.

Submission: Share the Python code for your chatbot and a screenshot of a conversation with it.

Slightly More Challenging Homework Assignment

Assignment 2: Random Joke Generator

Write a Python program that generates random jokes or anecdotes and displays them to the user.

Requirements:

- Create a list of different jokes or anecdotes in your program.
- Use the random module to select a random joke from the list.
- The program should ask the user if they want to hear another joke after displaying one.
- If the user agrees, the program should present another random joke.

Hint: You can use the `random.choice()` function to pick a random item from a list.

Submission: Share the Python code for your joke generator and screenshots of random jokes displayed to the user.

These assignments will allow students to practice working with conditional statements, the random module, and user interaction while adding a touch of fun to their Python learning experience.