Day 4 of Training at Ansh Info Tech

Topics Covered

Strings in Python

- String Methods
- String Formatting
- String Slicing and Concatenation
- 30 Practice Questions on Strings

Operators in Python

- Arithmetic Operators
- Bitwise Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Identity Operators
- Membership Operators
- 30 Practice Questions on Operators in Python

Precedence of Operators in Python

- 20 Practice Questions on Precedence of Operators in Python

Summary

Strings in Python

Strings are sequences of characters. Python provides various methods for manipulating strings, such as `.upper()`, `.lower()`, `.replace()`, and `.split()`. String formatting includes techniques like f-strings and the `.format()` method. String slicing and concatenation involve extracting and combining substrings.

Practice Questions on Strings

1. Convert a string to uppercase and lowercase.
2. Remove whitespace from a string.
3. Replace all occurrences of a substring in a string.

4. Find the index of the first occurrence of a substring.
5. **And many more...**

Operators in Python

Operators are symbols that perform operations on variables and values. Types include:

- **Arithmetic Operators**: Perform basic math operations.
- **Bitwise Operators**: Operate on binary representations.
- **Relational Operators**: Compare values.
- **Logical Operators**: Combine Boolean expressions.
- **Assignment Operators**: Assign values to variables.
- **Identity Operators**: Check if two variables reference the same object.
- **Membership Operators**: Test membership in sequences.

Practice Questions on Operators in Python

1. Perform basic arithmetic operations.
2. Use bitwise operators to manipulate binary values.
3. Compare two values using relational operators.
4. **And many more...**

Precedence of Operators in Python

Operator precedence determines the order in which operations are evaluated in expressions. Higher precedence operators are evaluated first. Understanding this helps in correctly interpreting and writing complex expressions.

Practice Questions on Precedence of Operators in Python

1. Evaluate expressions with mixed operators.
2. Determine the result of expressions with different precedence.
3. **And many more...**

Strings in Python

String Methods

Note: id() gives memory location of variable

- str.upper()
- str.lower()

- str.capitalize() ("Change first Letter to upper)
- str.title() ("Change first letter of every word to upper")
- str.swapcase()
- str.endswith()
- str.center()
- str.isnumeric()
- str.isalnum()
- str.expandtabs()
- str.isalpha()
- str.islower()
- str.index()
- str.find()
- str.casefold()

String Manipulation:

Write a Python program that reverses a given string. Given a string, write a function to count the number of vowels in it. Write a program to check if a string is a palindrome. Implement a function to capitalize the first letter of each word in a sentence.

```
str = "Hello World"
print(str[::-1])
vowelCount = 0
for char in str:
  if char.lower() in ['a', 'e', 'i', 'o', 'u']:
    vowelCount+=1
print(vowelCount)
str2 = str[::-1]
print("String is a Palindrome" if str == str2 else "String is not a Palindrome")
str = "hello world"
print(str.title())
```

String Formatting:

Create a formatted string using f-strings that includes variables and their values. Write a program to left-align a list of strings with a specified width. Given a list of names, format them into a bulleted list using string concatenation or join method.

```
var1 = "Hello"
var2 = "World"
print(f"{var1} {var2}")

strings = ["Hello", "World", "This", "is", "a", "test"]
width = 10
aligned_strings = []
for string in strings:
  aligned_strings.append(string.ljust(width))
print(aligned_strings)

names = ["Alice", "Bob", "Charlie"]
formatted_list = "- " + "\n- ".join(names)
print(formatted_list)
```

```
Hello World
['Hello     ', 'World     ', 'This      ', 'is        ', 'a         ', 'test      ']
- Alice
- Bob
- Charlie
```

String Searching and Counting:

Write a function to find the index of the first occurrence of a substring in a given string. Count the occurrences of a specific word in a paragraph. Write a program to extract all email addresses from a given text.

```
str = "Hello World"
sub = "World"
if(sub in str):
  print(str.index(sub))
else:
  print("Substring not found")
str2 = "Write a function to find the index of the first occurrence of a substring in a giver
print(str2.count("a"))

str3 = "john.doe@example.com, asdfjk;lasdf, jane.smith@example.com, mary.johnson@example.com
emails = []
for email in str3.split(","):
  if "@" in email:
    emails.append(email)
print(emails)
```

```
6
15
['john.doe@example.com', ' jane.smith@example.com', ' mary.johnson@example.com']
```

String Slicing and Concatenation:

Given a sentence, extract the last three words. Concatenate two strings with a space in between. Write a program that removes leading and trailing whitespaces from a string.

```
str = "Given a sentence, extract the last three words"
l1 = str.split(" ")
print(l1[-3:])
str1 = "Hello"
str2 = "World"
print(str1 + " " + str2)
print(str.strip())
print(str.replace(" ", ""))
```

```
['last', 'three', 'words']
Hello World
Given a sentence, extract the last three words
Givenasentence,extractthelastthreewords
```

Operators

- Arithmetic Operators -> +, -, *, /, %, //, exp.
- Bitwise Operators -> &, |, ~, ^, >>, <<
- Assignment Operators -> =, +=, -=, /=, *=, %=, //=, &=, |=, ^=, <<=, >>=
- Comparison Operators -> ==, !=, >=, <=, >, <
- Logical Operators -> and, or, not
- Identity Operators -> is, is not
- Membership Operators -> in, not in

Operator Precedence in Python

- Parenthesis

- Exponentiation
- Unary -> +x, -x, ~x
- *, / , //, %
- +, -
- ▌ ▌ , <<
- &
- ^
- |
- Relational
- not
- and
- or
- walrus :=

```
import re

s = 'GeeksforGeeks: A computer science portal for geeks'

match = re.search(r'portal', s)
print(match)

print('Start Index:', match.start())
print('End Index:', match.end())
```

```
<re.Match object; span=(34, 40), match='portal'>
Start Index: 34
End Index: 40
```

1. Given a string containing a sentence, use a string method to convert the entire string to uppercase.
2. You have a string with extra spaces at the beginning and end. Use a string method to remove these extra spaces.
3. Given a string, find the first occurrence of the substring "Python" and print its position.
4. You have a string with words separated by commas. Use a string method to split the string into a list of words.
5. Given a string containing a sentence, replace all occurrences of the word "Java" with "Python".
6. You have a string containing a mix of uppercase and lowercase characters. Use a string method to convert the entire string to lowercase.

7. Given a string, check if it starts with the prefix "Hello".
8. You have a string with numbers and letters. Use a string method to check if the string is alphanumeric.
9. Given a string containing multiple words, use a string method to join the words with a hyphen ("-").
10. You have a string with different words separated by spaces. Use a string method to count the number of words in the string.
11. Given a string containing a sentence, use a string method to capitalize the first letter of each word.
12. You have a string containing a URL. Use a string method to check if the URL ends with ".com".
13. Given a string containing a list of names separated by commas, use a string method to sort the names alphabetically.
14. You have a string containing a sentence. Use a string method to find the number of times the word "data" appears in the string.
15. Given a string, use a string method to reverse the string.
16. You have a string containing multiple email addresses separated by semicolons. Use a string method to extract the domain of each email address.
17. Given a string containing a paragraph, use a string method to replace all newline characters ("\n") with spaces.
18. You have a string that contains both letters and digits. Use a string method to extract only the digits from the string.
19. Given a string containing a sentence, use a string method to find the last occurrence of the substring "Python".
20. You have a string containing a sentence. Use a string method to swap the case of each character in the string.
21. Given a string containing a date in the format "DD-MM-YYYY", use a string method to extract the year.
22. You have a string with multiple lines of text. Use a string method to split the string into a list of lines.
23. Given a string containing a sentence, use a string method to find the index of the first occurrence of the character "a".
24. You have a string with special characters. Use a string method to remove all special characters from the string.

25. Given a string, use a string method to check if all characters in the string are digits.
26. You have a string containing a sentence. Use a string method to split the sentence into individual words and then join them back together with spaces.
27. Given a string containing a file path, use a string method to extract the file name.
28. You have a string containing HTML tags. Use a string method to remove all HTML tags from the string.
29. Given a string, use a string method to check if it is a palindrome.
30. You have a string containing a sentence with multiple spaces between words. Use a string method to remove all extra spaces, leaving only a single space between words.

```
# Question 1
str = "Hello World"
print(str.upper())

# Question 2
str = "   Hello World   "
print(str.strip())

# Question 3
str = "I am Learning the Python Programming Language"
print(str.find("Python"))

#Question 4
str = "I,am,Learning,the,Python,Programming,Language"
print(str.split(","))

#Question 5
str = "I am Learning the Java Programming Language and Java is good"
print(str.replace("Java", "Python"))

#Question 6
str = "Hello World"
print(str.lower())

#Question 7
str = "Hello World"
print(str.startswith("Hello"))

#Question 8
str = "Hello World"
print(str.isalnum())

#Question 9
str = "I am Learning the Python Programming Language"
print(str.replace(" ", "-"))
print("-".join(str.split(" ")))

#Question 10
str = "I am Learning the Python Programming Language"
print(len(str.split(" ")))

#Question 11
str = "I am Learning the Python Programming Language"
print(str.title())

#Question 12
str = "https://www.google.com"
print(str.endswith(".com"))

#Question 13
str = "Alice,Bob,Charlie"
```

```
print(sorted(str.split(",")))

#Question 14
str = "I am Learning the Python Programming Language"
print(str.count("data"))

#Question 15
str = "Hello World"
print(str[::-1])

#Question 16
str = "john.doe@example.com;jane.smith@example.com;mary.johnson@example.com"
l1 = str.split(";")
l2 = []
for email in l1:
  index = email.find("@")
  l2.append(email[index+1:])
print(l2)

#Question 17
str = "asdkfjas;l \n asdjfl;kas \n asdfjk;laf\n asdfklj;a"
print(str.replace("\n", " "))

#Question 18
string = "abc123def456"
digits = ''.join(c for c in string if c.isdigit())
print(digits)

#Question 19
str = "I am Learning the Python Programming Language and Python is good"
print(str.rfind("Python"))

#Question 20
str = "Hello World"
print(str.swapcase())

#Question 21
str = "12-01-2023"
print(str[-4:])

#Question 22
str = "Hello\nWorld"
print(str.splitlines())

#Question 23
str = "I am Learning the Python Programming Language"
print(str.find("a"))

#Question 24
str = "Hell@#$@$%!@o World!"
for char in str:
```

```
  if not char.isalnum():
    str = str.replace(char, "")
print(str)

#Question 25
str = "Hello World"
str2 = "2341"
print(str.isdigit())
print(str2.isdigit())
print(str.isnumeric())
print(str2.isnumeric())

#Question 26
str = "I am Learning the Python Programming Language"
print(" ".join(str.split(" ")))

#Question 27
str = "C:\\Users\\user\\Desktop\\file.txt"
print(str.split("\\")[-1])

#Question 28
import re
str = "<html><body><h1>Hello World</h1></body></html>"
clean = re.compile('<.*?>')
clean2 = re.sub(clean, '', str)
print(clean2)

#Question 29
str = "Hello World"
print(str == str[::-1])

#Question 30
str = "I   am Learning the Python   Programming Language"
print(str.split(" "))
print(str.split())

print(' '.join(str.split()))
```

```
HELLO WORLD
Hello World
18
['I', 'am', 'Learning', 'the', 'Python', 'Programming', 'Language']
I am Learning the Python Programming Language and Python is good
hello world
True
False
I-am-Learning-the-Python-Programming-Language
I-am-Learning-the-Python-Programming-Language
7
I Am Learning The Python Programming Language
True
```

```
['Alice', 'Bob', 'Charlie']
0
dlroW olleH
['example.com', 'example.com', 'example.com']
asdkfjas;l   asdjfl;kas   asdfjk;laf  asdfklj;a
123456
50
hELLO wORLD
2023
['Hello', 'World']
2
HelloWorld
False
True
False
True
I am Learning the Python Programming Language
file.txt
Hello World
False
['I', '', '', 'am', 'Learning', 'the', 'Python', '', '', 'Programming', 'Language']
['I', 'am', 'Learning', 'the', 'Python', 'Programming', 'Language']
I am Learning the Python Programming Language
```

1. You have two variables, a and b, containing integer values. Use arithmetic operators to calculate and print the sum, difference, product, and quotient of these variables.
2. Given a variable x containing an integer value, use the modulus operator to check if x is even or odd.
3. You have two variables, a and b. Use comparison operators to check if a is greater than b, and print the result.
4. Given two boolean variables, p and q, use logical operators to evaluate and print the result of p AND q, p OR q, and NOT p.
5. You have a variable n containing an integer value. Use the bitwise AND, OR, and XOR operators to perform operations with another integer m.
6. Given a variable y containing a floating-point number, use the floor division operator to divide y by 2 and print the result.
7. You have two strings, str1 and str2. Use the concatenation operator to join these strings and print the result.
8. Given a variable z containing an integer, use the increment operator to increase its value by 1 and print the result.
9. You have a list my_list and a value val. Use the in operator to check if val is present in my_list and print the result.

10. Given two variables a and b, use the assignment operator to assign the value of b to a and print a.
11. You have two variables, x and y, containing integer values. Use the compound assignment operator to add y to x and print the result.
12. Given a variable num containing an integer value, use the bitwise left shift operator to shift num by 2 bits and print the result.
13. You have a string text and an integer n. Use the repetition operator to repeat text n times and print the result.
14. Given two variables, a and b, use the comparison operators to check if a is equal to b and if a is not equal to b. Print the results.
15. You have two variables, x and y, containing integer values. Use the bitwise right shift operator to shift x by 3 bits and print the result.
16. Given a boolean variable flag, use the logical operator to check if flag is True and print the result.
17. You have a variable price containing a float value. Use the floor division operator to calculate how many whole units you can get for a given amount and print the result.
18. Given two sets, set1 and set2, use the union operator to combine these sets and print the result.
19. You have a dictionary my_dict and a key key. Use the in operator to check if key is present in my_dict and print the result.
20. Given two variables a and b, use the comparison operators to check if a is less than or equal to b, and print the result.
21. You have a string s and a character c. Use the in operator to check if c is present in s and print the result.
22. Given a variable num containing an integer value, use the bitwise NOT operator to invert its bits and print the result.
23. You have a list numbers and a value x. Use the multiplication operator to multiply each element in the list by x and print the result.
24. Given two boolean variables, a and b, use the logical operator to evaluate a AND NOT b and print the result.
25. You have two variables, a and b, containing integer values. Use the modulus operator to find the remainder when a is divided by b and print the result.

26. Given a variable a, use the exponentiation operator to calculate a raised to the power of 3 and print the result.
27. You have a string str1 and an integer n. Use the slicing operator to get the first n characters of the string and print the result.
28. Given two lists, list1 and list2, use the concatenation operator to merge these lists and print the result.
29. You have a dictionary person with keys name and age. Use the indexing operator to access and print the value of name.
30. Given a variable x containing an integer, use the decrement operator to decrease its value by 1 and print the result.

```
#Question 1

a = 10
b = 5

sum_ab = a + b
difference_ab = a - b
product_ab = a * b
quotient_ab = a / b

print("Sum:", sum_ab)           # Output: 15
print("Difference:", difference_ab)   # Output: 5
print("Product:", product_ab)       # Output: 50
print("Quotient:", quotient_ab)     # Output: 2.0

#Question 2

x = 4

if x % 2 == 0:
    print("x is even")  # Output: x is even
else:
    print("x is odd")

#Question 3

a = 10
b = 5

is_a_greater_than_b = a > b
print("Is a greater than b?", is_a_greater_than_b)  # Output: True

#Question 4

p = True
q = False

print("p AND q:", p and q)  # Output: False
print("p OR q:", p or q)    # Output: True
print("NOT p:", not p)      # Output: False

#Question 5

n = 6  # 110 in binary
m = 3  # 011 in binary

bitwise_and = n & m
bitwise_or = n | m
bitwise_xor = n ^ m

print("Bitwise AND:", bitwise_and)  # Output: 2 (010 in binary)
```

```
print("Bitwise OR:", bitwise_or)    # Output: 7 (111 in binary)
print("Bitwise XOR:", bitwise_xor)  # Output: 5 (101 in binary)

#Question 6

y = 7.5

floor_division_result = y // 2
print("Floor division result:", floor_division_result)  # Output: 3.0

#Question 7

str1 = "Hello"
str2 = "World"

concatenated_string = str1 + " " + str2
print("Concatenated string:", concatenated_string)  # Output: Hello World

#Question 8

z = 10

z += 1
print("Incremented value:", z)  # Output: 11

#Question 9

my_list = [1, 2, 3, 4, 5]
val = 3

is_val_in_list = val in my_list
print("Is val in list?", is_val_in_list)  # Output: True

#Question 10

a = 5
b = 10

a = b
print("Value of a:", a)  # Output: 10

#Question 11

x = 5
y = 3

x += y
print("Value of x after addition:", x)

#Question 12
```

```
num = 4  # 100 in binary

shifted_num = num << 2
print("Left shifted num:", shifted_num)  # Output: 16 (10000 in binary)

#Question 13

text = "Hello"
n = 3

repeated_text = text * n
print("Repeated text:", repeated_text)  # Output: HelloHelloHello

#Question 14

a = 5
b = 3

is_a_equal_to_b = a == b
is_a_not_equal_to_b = a != b

print("Is a equal to b?", is_a_equal_to_b)  # Output: False
print("Is a not equal to b?", is_a_not_equal_to_b)  # Output: True

#Question 15

x = 10
shifted_x = x >> 3
print("Shifted x:", shifted_x)  # Output: 2 (010 in binary)

#Question 16

flag = True

if flag:
    print("Flag is True")  # Output: Flag is True

#Question 17

price = 10.5

amount = 100
whole_units = amount//price
print("Whole units:", whole_units)  # Output: 10

#Question 18

set1 = {1, 2, 3}
set2 = {3, 4, 5}

union_set = set1 | set2
```

```
print("Union of sets:", union_set)  # Output: {1, 2, 3, 4, 5}

#Question 19

my_dict = {"name": "Alice", "age": 25}
key = "name"

is_key_in_dict = key in my_dict
print("Is key in dictionary?", is_key_in_dict)  # Output: True

#Question 20

a = 3
b = 5

is_a_less_than_or_equal_to_b = a <= b
print("Is a <= b?", is_a_less_than_or_equal_to_b)  # Output: True

#Question 21

s = "Hello, world!"
c = "w"

is_c_in_s = c in s
print("Is c in s?", is_c_in_s)  # Output: True

#Question 22

num = 5  # 0101 in binary

inverted_num = ~num
print("Bitwise NOT num:", inverted_num)  # Output: -6 (in two's complement

# Task 23: Multiply List Elements
numbers = [1, 2, 3, 4]
x = 2
```