Day 7

Strings in Python

Creating Strings

In [8]:

```
a = 'Hello'
b = "Python"
c = '''This is a
multiline
string'''

print(a)
print(type(a))
print(b)
print(type(b))
print(c)
print(type(c))
```

```
Hello
<class 'str'>
Python
<class 'str'>
This is a
multiline
string
<class 'str'>
```

Indexing of Strings

In [11]:

```
a = "Hello everyone"
print(a[0])
print(a[1])
print(a[2])
print(a[3])
print(a[4])
print(a[5])
print(a[6])
```

```
H
e
l
l
o

e
```

In [15]:

```
b = "AI is the Future"
print(b[-1], b[0])
print(b[-2], b[1])
print(b[-3], b[2])
print(b[-4], b[3])
print(b[-5])
print(b[-6])
print(b[-7])
```

```
e A
r I
u
t i
u
F
```

Slicing of Strings

positive slicing

In [21]:

```
a = "AI is the future of tech"
print(a[0:6])
```

```
AI is
```

In [23]:

```
a = "AI is the future of tech"
print(a[0:])
```

```
AI is the future of tech
```

In [25]:

```
a = "AI is the future of tech"
print(a[:])
```

```
AI is the future of tech
```

Slicing with Jump

In [28]:

```
a = "AI is the future of tech"
print(a[0:10:2])
```

```
A ste
```

In [30]:

```
a = "AI is the future of tech"
print(a[::2])
```

```
A steftr ftc
```

In [32]:

```
a = "i am learning datascience"
print(a[::3])
```

```
imengasee
```

Negative Slicing

In [46]:

```
a = "i am learning datascience"
print(a[-6:-1])
```

```
cienc
```

In [48]:
```
a = "i am learning datascience"
print(a[:-1])
```

```
i am learning datascienc
```

practice question

In [ ]:
```
Write a python program to reverse the string, "There is no shortcut for hardwork
```

In [52]:
```
str = "There is no shortcut for hardwork"
print(str[::-1])
```

```
krowdrah rof tuctrohs on si erehT
```

Finding the Length of a string

In [56]:
```
s = "theiscale"
length = len(s)
print(length)
```

```
9
```

empty string

In [63]:
```
s = ""
length = len(s)
print(length)
```

```
0
```

In [65]:
```
s = " "
length = len(s)
print(length)
```

```
1
```

In [67]:
```
s = "  "
length = len(s)
print(length)
```

```
2
```

In [71]:
```
s = "hello\nworld\nwhatsupp"
length = len(s)
print(s)
print(length)
```

```
hello
world
whatsupp
20
```

In [73]:
```
s = """hello
world"""
length = len(s)
```

```
print(s)
print(length)
```

```
hello
world
11
```

Python - Modify Strings

upper method

In [76]:

```
s = "this is the iscale"
upper_s = s.upper()
print(upper_s)
```

```
THIS IS THE ISCALE
```

In [78]:

```
a = "HeLLo WoRld"
upper_s = a.upper()
print(upper_s)
```

```
HELLO WORLD
```

lower method

In [82]:

```
text = "The Iscale"
lower_text = text.lower()
print(lower_text)
```

```
the iscale
```

In [84]:

```
sentence = "India is great "
lower_sentence = sentence.lower()
print(lower_sentence)
```

```
india is great
```

Replace String

In [87]:

```
text = "Hello, World! Hello, Python!"
new_text = text.replace("Hello", "Hi")
print(new_text)
```

```
Hi, World! Hi, Python!
```

In [91]:

```
tweet = "OMG, NLP is so cool! BTW, I'm learning NLP."
print(tweet)
clean_tweet = tweet.replace("OMG", "Oh my god").replace("BTW", "By the way")
print(clean_tweet)
```

```
OMG, NLP is so cool! BTW, I'm learning NLP.
Oh my god, NLP is so cool! By the way, I'm learning NLP.
```

In [93]:

```
text_with_missing = "NLP is an [missing] field of study."
clean_text = text_with_missing.replace("[missing]", "emerging")
```

```
print(clean_text)
```

```
NLP is an emerging field of study.
```

find() method

In [96]:

```
a = "python is great"
b = a.find("g")
print(len(a))
print(b)
```

```
15
10
```

In [98]:

```
a = "python is great"
b = a.find("thon")
print(len(a))
print(b)
```

```
15
2
```

Python String index() Method

In [101…

```
txt = "I am learning python for data science"

x = txt.index("i")
print(x)
```

```
10
```

String Concatenation

In [104…

```
a = "Hello"
b = "World"
c = a + " "+ b
print(c)
```

```
Hello World
```

In [106…

```
a = "data science"
print(a*3)
```

```
data sciencedata sciencedata science
```

sorted method

In [110…

```
a = "swati"
print(sorted(a))
```

```
['a', 'i', 's', 't', 'w']
```

practice questions

In [ ]:

```
Two strings are called anagrams if they contain the same characters in a differe
 ◆  Example: "listen" and "silent" are anagrams.
 ◆  Write a function to check if two given strings are anagrams.
```

In [114…

```
def are_anagrams(str1, str2):
    return sorted(str1) == sorted(str2)
```

```
True
False
```

In [116…

```
print(are_anagrams("listen", "silent"))
```

```
True
```

In [118…

```
print(are_anagrams("hello", "world"))
```

```
False
```

Question 2

In [ ]:

```
Write a Python function to find the longest word in a given sentence.
 ◆  Input: "Python is a powerful programming language"
 ◆  Output: "programming"
```

In [122…

```
def longest_word(sentence):
    words = sentence.split()
    print(words)# Split sentence into words
    return max(words, key=len)  # Find the word with the maximum length

# Test case
print(longest_word("Python is a powerful programming language"))
# Output: "programming"
```

```
['Python', 'is', 'a', 'powerful', 'programming', 'language']
programming
```

In [ ]:

```
Find the most frequently occurring character in a given string.
 ◆  Input: "mississippi"
 ◆  Output: "i"
```

In [125…

```
def most_frequent_char(s):
    max_count = 0
    max_char = ""

    for char in s:
        if s.count(char) > max_count:  # Count occurrences of each character
            max_count = s.count(char)  # Update max count
            max_char = char  # Update most frequent character

    return max_char

# Test case
print(most_frequent_char("mississippi"))  # Output: "i"
```

```
i
```

```
In [ ]:
```