String creation

Strings are enclosed in single quotes '...', double quotes "...", or triple quotes '''...''' / """...""" for multi-line strings.

Strings are immutable, meaning once created, they cannot be changed.

```
In [2]: str="hello world"
        print(str)
```

```
hello world
```

```
In [3]: str1='hello world'
        print(str1)
```

```
hello world
```

str2=''' hello world good morning hope u all doing good''' print(str2)

```
In [10]: str3=('happy'
               'new'
               'year')
         print(str3)
```

```
happynewyear
```

```
In [29]: str4=("hello guys")
         s=str4*5
         s
```

```
Out[29]: 'hello guyshello guyshello guyshello guyshello guys'
```

```
In [26]: len(str4)
```

```
Out[26]: 10
```

Methods of string

```
In [3]: str1="hello world, Good Morning"
        str1
```

```
Out[3]: 'hello world, Good Morning'
```

```
In [5]: str1.capitalize() #Capitalizes the first character of the string
```

```
Out[5]: 'Hello world, good morning'
```

```
In [6]: str1.upper()
```

```
Out[6]: 'HELLO WORLD, GOOD MORNING'
```

```
In [7]: str1.isupper()
```

```
Out[7]:  False
```

```
In [8]:  str1
```

```
Out[8]:  'hello world, Good Morning'
```

```
In [9]:  str1.isupper()
```

```
Out[9]:  False
```

```
In [10]:  str1.lower()
```

```
Out[10]:  'hello world, good morning'
```

```
In [11]:  str1.islower()
```

```
Out[11]:  False
```

```
In [14]:  str1.count("o")
```

```
Out[14]:  5
```

```
In [105…  str1.casefold() #aggressive case normalization, especially useful when comparing
```

```
Out[105…  'hello world, good morning'
```

```
In [19]:  str1.center(10)
```

```
Out[19]:  'hello world, Good Morning'
```

```
In [20]:  str1.encode()
```

```
Out[20]:  b'hello world, Good Morning'
```

```
In [22]:  str1.endswith('g')
```

```
Out[22]:  True
```

```
In [23]:  str1.endswith('d')
```

```
Out[23]:  False
```

```
In [25]:  str1.expandtabs(10)
```

```
Out[25]:  'hello world, Good Morning'
```

```
In [26]:  str1.find("hello")
```

```
Out[26]:  0
```

```
In [27]:  str1.find("Morning")
```

```
Out[27]:  18
```

```
In [30]:  str1.find("w")
```

Out[30]:
```
6
```

In [31]:
```
str1.format()
```

Out[31]:
```
'hello world, Good Morning'
```

In [33]:
```
str1.format_map(1)
```

Out[33]:
```
'hello world, Good Morning'
```

In [34]:
```
str1.index("o")
```

Out[34]:
```
4
```

In [35]:
```
str1.isalnum()
```

Out[35]:
```
False
```

In [36]:
```
str1.isalpha()
```

Out[36]:
```
False
```

In [37]:
```
st="hello"
st.isalpha()
```

Out[37]:
```
True
```

In [38]:
```
str1.isascii()
```

Out[38]:
```
True
```

In [39]:
```
str1.isdecimal()
```

Out[39]:
```
False
```

In [40]:
```
str1.isdigit()
```

Out[40]:
```
False
```

In [41]:
```
str1.isidentifier()
```

Out[41]:
```
False
```

In [42]:
```
str1.islower()
```

Out[42]:
```
False
```

In [43]:
```
st1="good morning"
st1.islower()
```

Out[43]:
```
True
```

In [44]:
```
str1.isnumeric()
```

Out[44]:
```
False
```

In [45]:
```
str1.isprintable()
```
Out[45]:
```
True
```

In [46]:
```
str1.isspace()
```
Out[46]:
```
False
```

In [47]:
```
st.isspace()
```
Out[47]:
```
False
```

In [55]:
```
s="hlo guys""hmm"
s.isspace()
s
```
Out[55]:
```
'hlo guyshmm'
```

In [57]:
```
str1.join(s) #Joins elements of an iterable into a single string with the string
```
Out[57]:
```
'hhello world, Good Morninglhello world, Good Morningohello world, Good Morning
hello world, Good Morningghello world, Good Morninguhello world, Good Morningyh
ello world, Good Morningshello world, Good Morninghhello world, Good Morningmhe
llo world, Good Morningm'
```

In [60]:
```
str1.ljust(205)
```
Out[60]:
```
'hello world, Good Morning
'
```

In [61]:
```
str1.istitle()
```
Out[61]:
```
False
```

In [62]:
```
s2="Goood Morning"
s2.istitle()
```
Out[62]:
```
True
```

In [63]:
```
str1.lstrip()
```
Out[63]:
```
'hello world, Good Morning'
```

In [66]:
```
str1.partition("good")
```
Out[66]:
```
('hello world, Good Morning', '', '')
```

In [69]:
```
s2.partition("goood")
```
Out[69]:
```
('Goood Morning', '', '')
```

In [71]:
```
str1.replace("hello","hi")
```
Out[71]:
```
'hi world, Good Morning'
```

In [72]:
```
str1.rfind("hi")
```

```
Out[72]: -1
```

```
In [75]: str1.rfind("good")
```

```
Out[75]: -1
```

```
In [79]: str1.rindex("hello") # used with strings to find the last (rightmost) occurrence
```

```
Out[79]: 0
```

```
In [81]: str1.rindex("hi")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[81], line 1
----> 1 str1.rindex("hi")

ValueError: substring not found
```

```
In [83]: str1.rjust("hello")# method in Python is a handy way to right-align a string by
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[83], line 1
----> 1 str1.rjust("hello")

TypeError: 'str' object cannot be interpreted as an integer
```

```
In [84]: str1.split()
```

```
Out[84]: ['hello', 'world,', 'Good', 'Morning']
```

```
In [85]: str1.swapcase()
```

```
Out[85]: 'HELLO WORLD, gOOD mORNING'
```

```
In [86]: str1.upper()
```

```
Out[86]: 'HELLO WORLD, GOOD MORNING'
```

```
In [88]: str1.translate(table)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[88], line 1
----> 1 str1.translate(table)

NameError: name 'table' is not defined
```

```
In [89]: table = str.maketrans("aeiou", "12345")
         text = "hello world"
         print(text.translate(table))
```

```
h2ll4 w4rld
```

```
In [95]: str1.startswith("hello") # method in Python is a simple yet powerful way to chec
```

```
Out[95]: True
```

```
In [99]: s3="python"  #
         s3="python"
         s3="python"
         s3="python"
         s3.zfill(15)

Out[99]: '000000000python'
```

```
In [102… s3.removeprefix("p")

Out[102… 'ython'
```

```
In [103… s3.removesuffix("n")

Out[103… 'pytho'
```

```
In [ ]:
```