Strings & String Methods

Strings are a data type used to represent text

In Python, the string type is str

In [1]:
```
type("Hello World")
```
Out[1]:
```
str
```

To find the length of a string, use the len() function

In [2]:
```
greeting = "Hello, World"
len(greeting)
```
Out[2]:
```
12
```

String Concatenation:

Strings can be combined using the + operator

In [3]:
```
string1 = "abra"
string2 = "cadabra"
magic_word = string1 + string2
magic_word
```
Out[3]:
```
'abracadabra'
```

Another example of string concatenation using a string literal

In [4]:
```
first_name = "Tyrion"
last_name = "Lannister"
full_name = first_name + " " + last_name
full_name
```
Out[4]:
```
'Tyrion Lannister'
```

String Indexing

Each character in a string can be accessed by it's index

In [5]:
```
car_brand = "Maruti Suzuki"
car_brand[4]
```
Out[5]:
```
't'
```

Trying to access an index beyond the length of the string, you will get an IndexError

```
In [6]: car_brand[17]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call las
t)
<ipython-input-6-45d96039c72b> in <module>
----> 1 car_brand[17]

IndexError: string index out of range
```

A good way to access the last character in a string is to use a negative index

```
In [ ]: car_brand[-1]
```

String Slicing

Strings can be sliced to get a portion of a string (a substring). The substring will start at the first index and end before the second index

```
In [ ]: dessert = "vanilla ice cream"
        dessert[8:11]
```

You can omit the first index of a slice, and Python will assume the substring starts from the beginning of the string

```
In [ ]: dessert[:7]
```

You can omit the second index of a slice, and Python will assume the substring end at the last character of the string

```
In [ ]: dessert[12:]
```

Omitting both indexes in a slice will return the whole string

```
In [ ]: dessert[:]
```

Unlike string indexing, python will not throw an index error if you enter an index that is beyond the length of the string

```
In [ ]: dessert[5:19]
```

Some String Methods

Strings can be converted to all lowercase letters using the .lower() method

```
In [ ]: college_name = "RNSIT"
        college_name.lower()
```

However, this does not change the string itself

```
In [ ]: college_name
```

You'll have to assign the lowercase string to another variable

```
In [ ]: lower_college_name = college_name.lower()
        lower_college_name
```

The .upper() method will convert a string to all uppercase letters

```
In [ ]: lower_college_name.upper()
```

Removing whitespace from a string

There are times you need to remove whitespace from the beginning or end of a string (e.g. accidental user input, creating strings from filenames)

There are 3 string methods you can use

.rstrip() removes whitespace from the right side of a string

In [ ]:
```
accidental_user_input = "Toast       "
accidental_user_input = accidental_user_input.rstrip()
accidental_user_input
```

.lstrip() removes whitespace from the left side of a string

In [ ]:
```
accidental_user_input = "       Butter"
accidental_user_input = accidental_user_input.lstrip()
accidental_user_input
```

.strip() removes whitespace from the left and right side of a string

In [ ]:
```
accidental_user_input = "       Jam       "
accidental_user_input = accidental_user_input.strip()
accidental_user_input
```

See if a string starts or ends with a certain substring

To see if a string starts with a certain substring use the .startswith() method

In [ ]:
```
country = "India"
country.startswith("In")
```

To see if a string ends with a certain substring use the .endswith() method

In [ ]:
```
country = "India"
country.endswith("dia")
```

These methods are case sensitive

In [ ]:
```
country.startswith("in")
```

Working with User Input

You can get user input as a string using the .input() function

In [10]:
```python
question = "Where are you from? "
user_input = input(question)
print(user_input)
```

```
Where are you from? Bengaluru
Bengaluru
```

Converting between Strings and Numbers

Sometimes you'll want to convert a string to a number, you can do that with the int() or float() functions.

The user input is always returned as a string so to do math with it, you'll need to convert it to a number first

In [9]:
```python
num = input("Enter a number to be tripled: ")
tripled_num = int(num) * 3
print(tripled_num)
```

```
Enter a number to be tripled: 3
9
```

The float() function will add/preserve the decimal

In [14]:
```python
float(num)
```

Out[14]:
```
3.0
```

To convert a number to a string use the str() function

In [18]:
```python
num_mangos = 7
phrase = "I am going to eat " + str(num_mangos) + " mangos"
print(phrase)
```

```
I am going to eat 7 mangos
```

f-strings

f-strings let you include non-string variables in strings in a more readable way

In [21]:
```python
num_legs = 8
phrase = f"A spider has {num_legs} legs"
print(phrase)
```

```
A spider has 8 legs
```

Finding a substring in a string

Use the .find() method to find the index of the first substring in a string

```
In [22]: phrase = "A needle in a haystack"
         phrase.find("needle")

Out[22]: 2
```

If the substring is not present a -1 will be returned

```
In [23]: phrase.find("nail")

Out[23]: -1
```

Replacing substrings

Use the .replace() method to replace all occurences of one substring with another ¶

```
In [29]: phrase = "I'm telling lies"
         new_phrase = phrase.replace("lies", "the truth")
         print(new_phrase)

         I'm telling the truth
```