Check if "free" is present in the following text:

txt = "The best things in life are free!"  
print("free" in txt)

txt = "The best things in life are free!"  
if "free" in txt:  
  print("Yes, 'free' is present.")

txt = "The best things in life are free!"  
print("expensive" not in txt)

a = "Hello, World!"  
print(a.upper())

Whitespace is the space before and/or after the actual text, and very often you want to remove this space.

a = " Hello, World! "  
print(a.strip()) # returns "Hello, World!"

a = "Hello, World!"  
print(a.replace("H", "J"))

The split() method returns a list where the text between the specified separator becomes the list items.

a = "Hello, World!"  
print(a.split(",")) # returns ['Hello', ' World!']

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

As we learned in the Python Variables chapter, we cannot combine strings and numbers like this:

age = 36  
txt = "My name is John, I am " + age  
print(txt)

To specify a string as an **f-string**, simply put an f in front of the string literal, and add curly brackets {} as placeholders for variables and other operations.

age = 36  
txt = f"My name is John, I am {age}"  
print(txt)

A placeholder can include a modifier to format the value.

A modifier is included by adding a colon : followed by a legal formatting type, like .2f which means fixed point number with 2 decimals:

price = 59  
txt = f"The price is {price:.2f} dollars"  
print(txt)

A placeholder can contain Python code, like math operations:

txt = f"The price is {20 \* 59} dollars"  
print(txt)

The escape character allows you to use double quotes when you normally would not be allowed:

txt = "We are the so-called \"Vikings\" from the north."

|  |  |  |
| --- | --- | --- |
| \' | Single Quote | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_escape2) |
| \\ | Backslash | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_backslash) |
| \n | New Line | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_newline) |
| \r | Carriage Return | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_r) |
| \t | Tab | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_t) |
| \b | Backspace | [Try it »](https://www.w3schools.com/python/showpython.asp?filename=demo_string_b) |
| \f | Form Feed |  |
| \ooo | Octal value | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_string_octal) |
| \xhh | Hex value |  |

Python - String Methods

<https://www.w3schools.com/python/python_strings_methods.asp>

The bool() function allows you to evaluate any value, and give you True or False in return,

Any string is True, except empty strings.

Any number is True, except 0

bool("abc")  
bool(123)  
bool(["apple", "cherry", "banana"])

In fact, there are not many values that evaluate to False, except empty values, such as (), [], {}, "", the number 0, and the value None. And of course the value False evaluates to False.

bool(False)  
bool(None)  
bool(0)  
bool("")  
bool(())  
bool([])  
bool({})

Check if an object is an integer or not:

x = 200  
print(isinstance(x, int))

Python Operators--- See its page

https://www.w3schools.com/python/python\_operators.asp

x = ["apple", "banana"]

y = ["apple", "banana"]

z = x

print(x is z)

# returns True because z is the same object as x

print(x is y)

# returns False because x is not the same object as y, even if they have the same content

print(x == y)

# to demonstrate the difference betweeen "is" and "==": this comparison returns True because x is equal to y

List

It is also possible to use the list() constructor when creating a new list.

List items are ordered, changeable, and allow duplicate values.

List items are indexed

thislist = ["apple", "banana", "cherry"]  
if "apple" in thislist:  
  print("Yes, 'apple' is in the fruits list")

thislist = ["apple", "banana", "cherry"]  
thislist[1] = "blackcurrant"  
print(thislist)

thislist = ["apple", "banana", "cherry"]  
thislist[1:3] = "blackcurrant"  
print(thislist)

thislist = ["apple", "banana", "cherry"]

thislist[1:3] = ["blackcurrant"]

print(thislist)

thislist = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]  
thislist[1:3] = ["blackcurrant", "watermelon"]  
print(thislist)

thislist = ["apple", "banana", "cherry"]  
thislist[1:2] = ["blackcurrant", "watermelon"]  
print(thislist)

<https://www.w3schools.com/python/python_lists_add.asp>

<https://www.w3schools.com/python/python_lists_remove.asp>

<https://www.w3schools.com/python/python_lists_loop.asp>

<https://www.w3schools.com/python/python_lists_comprehension.asp>