Python Strings

String Literals

String literals in python are surrounded by either single quotation marks, or double quotation marks.

'hello' is the same as "hello".

You can display a string literal with the print() function:

Example

print("Hello")  
print('Hello')

Assign String to a Variable

Assigning a string to a variable is done with the variable name followed by an equal sign and the string:

Example

a = "Hello"  
print(a)

Multiline Strings

You can assign a multiline string to a variable by using three quotes:

Example

You can use three double quotes:

a = """Lorem ipsum dolor sit amet,consectetur adipiscing elit,sed do eiusmod tempor incididuntut labore et dolore magna aliqua."""

print(a)

Or three single quotes:

Example

a = '''Lorem ipsum dolor sit amet,consectetur adipiscing elit,sed do eiusmod tempor incididuntut labore et dolore magna aliqua.'''

print(a)

**Note:** in the result, the line breaks are inserted at the same position as in the code.

Strings are Arrays

Like many other popular programming languages, strings in Python are arrays of bytes representing unicode characters.

However, Python does not have a character data type, a single character is simply a string with a length of 1.

Square brackets can be used to access elements of the string.

Example

Get the character at position 1 (remember that the first character has the position 0):

a = "Hello, World!"  
print(a[1])

Slicing

You can return a range of characters by using the slice syntax.

Specify the start index and the end index, separated by a colon, to return a part of the string.

Example

Get the characters from position 2 to position 5 (not included):

b = "Hello, World!"  
print(b[2:5])

Negative Indexing

Use negative indexes to start the slice from the end of the string:

Example

Get the characters from position 5 to position 1, starting the count from the end of the string:

b = "Hello, World!"  
print(b[-5:-2])

String Length

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

Example

The len() function returns the length of a string:

a = "Hello, World!"  
print(len(a))

String Methods

Python has a set of built-in methods that you can use on strings.

Example

The strip() method removes any whitespace from the beginning or the end:

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

Example

The lower() method returns the string in lower case:

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

Example

The upper() method returns the string in upper case:

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

Example

The replace() method replaces a string with another string:

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

Example

The split() method splits the string into substrings if it finds instances of the separator:

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

Learn more about String Methods with our [String Methods Reference](https://www.w3schools.com/python/python_ref_string.asp)

Check String

To check if a certain phrase or character is present in a string, we can use the keywords in or not in.

Example

Check if the phrase "ain" is present in the following text:

txt = "The rain in Spain stays mainly in the plain"  
x = "ain" in txt  
print(x)

Example

Check if the phrase "ain" is NOT present in the following text:

txt = "The rain in Spain stays mainly in the plain"  
x = "ain" not in txt  
print(x)

String Concatenation

To concatenate, or combine, two strings you can use the + operator.

Example

Merge variable a with variable b into variable c:

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

Example

To add a space between them, add a " ":

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

String Format

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

Example

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

But we can combine strings and numbers by using the format() method!

The format() method takes the passed arguments, formats them, and places them in the string where the placeholders {} are:

Example

Use the format() method to insert numbers into strings:

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

The format() method takes unlimited number of arguments, and are placed into the respective placeholders:

Example

quantity = 3  
itemno = 567  
price = 49.95  
myorder = "I want {} pieces of item {} for {} dollars."  
print(myorder.format(quantity, itemno, price))

You can use index numbers {0} to be sure the arguments are placed in the correct placeholders:

Example

quantity = 3  
itemno = 567  
price = 49.95  
myorder = "I want to pay {2} dollars for {0} pieces of item {1}."  
print(myorder.format(quantity, itemno, price))