############## Strings #############

Strings 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:

Get your own Python Server

print("Hello")

print('Hello')

Quotes Inside Quotes:

You can use quotes inside a string, as long as they don't match the quotes surrounding the string:

#Example:

print("It's alright")

print("He is called 'Johnny'")

print('He is called "Johnny"')

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 incididunt
ut 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 incididunt
ut 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])
```

Looping Through a String:

Since strings are arrays, we can loop through the characters in a string, with a for loop.

#Example:

Loop through the letters in the word "banana":

```
for x in "banana":
  print(x)
```

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))
```

Check String:

To check if a certain phrase or character is present in a string, we can use the keyword in.

#Example:

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

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

Use it in an if statement:

#Example:

Print only if "free" is present:

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

Check if NOT :

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

#Example:

Check if "expensive" is NOT present in the following text:

```
txt = "The best things in life are free!"
print("expensive" not in txt)
# Use it in an if statement:
```

#Example:

print only if "expensive" is NOT present:

```
txt = "The best things in life are free!"
if "expensive" not in txt:
  print("No, 'expensive' is NOT present.")
```

########## Python - Slicing Strings ############

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.

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

```
b = "Hello, World!"
print(b[2:5])
```

Note: The first character has index 0.

Slice From the Start:

By leaving out the start index, the range will start at the first character:

```
# Example
```

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

```
b = "Hello, World!"
```

```
print(b[:5])
```

ADVERTISEMENT

Slice To the End:

By leaving out the end index, the range will go to the end:

```
# Example
```

Get the characters from position 2, and all the way to the end:

```
b = "Hello, World!"
print(b[2:])
```