

</talentlabs>

CHAPTER 2

Strings Manipulation





</talentlabs>

AGENDA

- Revision on Strings
- Multiple Line Strings
- Composing Strings with Variables
- Escape Characters

Revision on Strings



Strings (aka Text)

- In programming, we use “Strings” to represent text (including words, phrases, sentences and even paragraphs)
- To define a string in Python, we can use **double quotes** or **single quotes**.



Single Quote

```
print('Hello, world!')
```

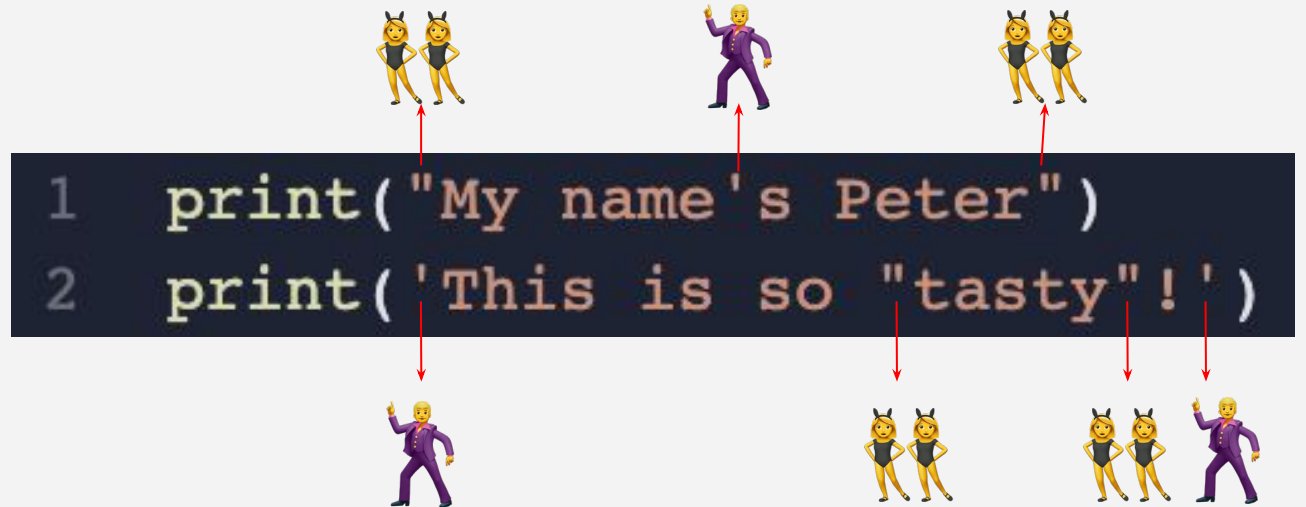


Double Quote

```
print("Hello, world!")
```

Including Single Quotes or Double Quotes

We need to mix single and double quotes so we can include single/double quotes in the string as we cannot include a single quote within single-quoted string.



```
1 print("My name's Peter")
2 print('This is so "tasty"!')
```

The diagram illustrates the use of single and double quotes in Python strings. Red arrows point from the quote characters in the code to corresponding cat and person emojis. In the first line, the opening double quote points to a pair of cat emojis, the single quote points to a person emoji, and the closing double quote points to another pair of cat emojis. In the second line, the opening single quote points to a person emoji, the double quote points to a pair of cat emojis, the single quote points to another pair of cat emojis, and the closing single quote points to a person emoji.

Joining Strings

The easiest way of joining strings
(i.e. combining 2 strings together)
is to use the “+” operator

Input:

```
1 print("Hello " + "World!")
```



Output:

```
Hello World!
```



Multiple Line Strings



Creating a Very Long Single Line String

- Sometimes the string we want to create is very long so it looks bad in the code
- We can split it into multiple lines using “\” in code (while the string is still actually one line.


Input:

(The two print statement is the same.)

```
print("Hello Peter, This is Mary. Nice to meet you.")
```

OR

```
print("Hello Peter, This is Mary. \nNice to meet you.")
```



Output:

```
Hello Peter, This is Mary. Nice to meet you.
```


Creating Multi-line String

Sometimes, you actually want multi-line strings:

- You can use triple quotes to create multi-line string.
- Both triple “single quotes” or triple “double quotes” would work

Input:

```
print('''this is line 1  
this is line 2  
this is line 3  
this is line 4  
''')
```

OR

```
print("""this is line 1  
this is line 2  
this is line 3  
this is line 4  
""")
```

Output:

```
this is line 1  
this is line 2  
this is line 3  
this is line 4
```

Composing Strings with Variables



Creating Dynamic String

- Consider the program here, we need to display the input user name in our greeting message
- The resulting string is different every time, depending on the variable's value (username)
- Simplest way is to use **the “+” operator**

```
1 username = input()  
2 print("Hello " + username)
```

Problem with Using “+”

- Sometimes, we need to include many variables in one string. For example, we want to ask users for multiple inputs, and summarize that to them.
- Using too many “+” operators is **hard to code and read**.

```
1 username = input()
2 age = input()
3 gender = input()
4
5 print("Hello " + username + " and you are " + age +
6      " years old. Your Gender is " + gender + ".")
```

👁👁 very hard to read...

Composing Strings

There is a smarter way of composing strings with variables using “.format()”



```
print("Hello " + username + " and you are " + age +  
" years old. Your Gender is " + gender + ".")
```



```
print("Hello {} and you are {} years old. \n  
Your Gender is {}".format(username, age, gender))
```

Smarter way using “.format()”

Escape Characters



Escape Characters

- Some special characters to help you in composing strings
- All of them start with “\”
(Be aware of direction, we are using “backslash”, not the usual “slash”)

Some of the Commonly Used Escape Characters

Escape Character	Usage
\n	new line
\"	using double quote in a double quoted string
\'	using single quote in a single quoted string

Example 1 - \n

These 2 strings are the same thing.

```
print("""This is line 1.  
This is line 2.""")
```

=

```
print("This is line 1.\nThis is line 2.")
```


Example 2 - \'

These 2 strings are the same thing.

```
print("My name's Peter")
```

=

```
print('My name\'s Peter')
```

Using single quote in a single-quoted string

Example 3 - \"

These 2 strings are the same thing.

```
print('This is so "tasty"!')
```

=

```
print("This is so \"tasty\"!")
```

Using double quotes in a double-quoted string