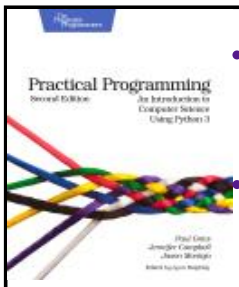
Type `str`: Strings in Python



- Chapter 4.1 Creating Strings of Characters
- Chapter 4.2 Using Special Characters in Strings

Optional reading

String Literal

A *string literal* is a sequence of characters. In Python, this type is called `str`. Strings in Python start and end with a single quotes (`'`) or double quotes (`"`). A string can be made up of letters, numbers, and special characters. For example:

```
>>> 'hello'
'hello'
>>> 'how are you?'
'how are you?'
>>> 'short- and long-term'
'short- and long-term'
```

If a string begins with a single quote, it must end with a single quote. The same applies to double-quoted strings. You can not mix the type of quotes.

Escape Sequences

To include a quote within a string, use an *escape character* (`\`) before it. Otherwise Python interprets that quote as the end of a string and an error occurs. For example, the following code results in an error because Python does not expect anything to come after the second quote:

```
>>> storm_greeting = 'wow, you're dripping wet.'
SyntaxError: invalid syntax
```

The *escape sequence* `\'` indicates that the second quote is simply a quote, not the end of the string:

```
>>> storm_greeting = 'Wow, you\'re dripping wet.'
"Wow, you're dripping wet."
```

An alternative approach is to use a double-quoted string when including a single-quote within it, or vice-versa. Single- and double-quoted strings are equivalent. For example, when we used double-quotes to indicate the beginning and end of the string, the single-quote in `you're` no longer causes an error:

```
>>> storm_greeting = "Wow, you're dripping wet."
"Wow, you're dripping wet."
```

String Operators

Expression	Description	Example	Output
`str1 + str2`	concatenate `str1` and `str1`	`print('ab' + 'c')`	`abc`
`str1 * int1`	concatenate `int1` copies of `str1`	`print('a' * 5)`	`aaaaa`
`int1 * str1`	concatenate `int1` copies of `str1`	`print(4 * 'bc')`	`bcbcbcbc`

Note: *concatenate* means to join together

The `*` and `+` operands obey by the standard precedence rules when used with strings.

All other mathematical operators and operands result in a `TypeError`.