

## Variables

### Naming and Using Variables

Be sure to keep the following rules in mind when working with variables:

- Variable names can contain only letters, numbers and underscores. They can start with a letter or an underscore, but not with a number.
- Spaces are not allowed in variable names, but underscores can be used to separate words in variable names.
- Avoid using Python keywords and function names as variable names.
- Variable names should be short but descriptive.
- Be careful when using the lowercase letter `l` and the uppercase letter `O` because they could be confused with the numbers 1 and 0.

### Variables are Labels

It's much better to think of variables as labels that you can assign to values.

You can also say that a variable references a certain value.

## Strings

A *string* is a series of characters.

Anything inside quotes is considered a string in Python, and you can use single or double quotes around your string

```
1 "This is a string"
2 'This is also a string'
```

This flexibility allows you to use quotes and apostrophes within your strings

```
1 'I told my friend, "Python is my favorite language!'"
2 "The language 'Python' is named after Monty Python, not the snake."
3 "One of Python's strengths is its diverse and supportive community."
```

### Changing Case in a String with Methods

```
1 name = "ada loveLace"
2 print(name.title())
3 ---> Ada Lovelace
```

A *method* is an action that Python can perform on a piece of data.

The dot(.) after `name` in `name.title()` tells Python to make the `title()` method act on the variable `name`.

Every method is followed by a set of parentheses, because methods often need additional information to do their work. That information is provided inside the parentheses.

The `title()` function doesn't need any additional information, so its parentheses are empty.

```
1 name = "ada lovelace"
2 print(name.upper())
3 print(name.lower())
4 ---> ADA LOVELACE
5 ---> ada lovelace
```

## Using Variables in Strings

```
1 first_name = "ada"
2 last_name = "lovelace"
3 full_name = f"{first_name} {last_name}"
4 print(full_name)
5 ---> ada lovelace
```

These strings are called *f-strings*.

The *f* is for *format*, because Python formats the string by replacing the name of any variable in braces with its value.

## Adding Whitespace to Strings with Tabs or Newlines

In programming, *whitespace* refers to any nonprinting characters, such as spaces, tabs, and end-of-line symbols.

```
1 print("Python")
2 ---> Python
3 print("\tPython")
4 --->     Python
```

## Stripping Whitespace

Extra whitespace can be confusing in your programs.

```
1 favorite_language = 'python '
2 favorite_language.rstrip()
3 ---> 'python'
```

```
4 favorite_language
5 ---> 'python '
6
7 favorite_language = ' python'
8 favorite_language.lstrip()
9 ---> 'python'
10 favorite_language
11 ---> ' python'
12
13 favorite_language = ' python '
14 favorite_language.strip()
15 ---> 'python'
16 favorite_language
17 ---> ' python '
```

## Removing Prefix

When working with strings, another common task is to remove a prefix.

```
1 nostarch_url = 'https://nostarch.com'
2 nostarch_url.removeprefix('https://')
3 ---> 'nostarch.com'
4
5 simple_url = nostarch_url.removeprefix('https://')
```

removesuffix

## Avoiding Syntax Errors with Strings

## Numbers

### Integers

```
1 2 + 3
2 ---> 5
```

-, \*, /, \*\*

### Floats

Python calls any number with a decimal point a *float*

However, be aware that you can sometimes get an arbitrary number of decimal places in your answer

```
1 0.2 + 0.1
2 ----> 0.30000000000000004
```

## Underscores in Numbers

When you're writing long numbers, you can group digits using underscores to make large numbers more readable

```
1 universe_age = 14_000_000_000
2 print(universe_age)
3 ----> 14000000000
```

## Multiple Assignment

```
1 x, y, z = 0, 0, 0
```

## Constant

A *constant* is a variable whose value stays the same throughout the life of a program.

Python doesn't have built-in constant types, but Python programmers use all capital letters to indicate a variable should be treated as a constant

```
1 MAX_CONNECTIONS = 5000
```

## Comments

```
1 #
2 '''
3 '''
```