

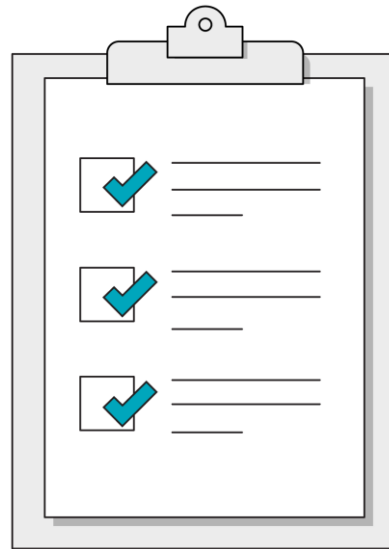
General Assembly



Intro to Python

Our Goals for Today

- Use Python functions to encapsulate problem-solving logic.
- Begin building Python skills through code challenges.



Intro to Python

Functions & Code Challenges



Python Syntax



Data Structures



Control Flow



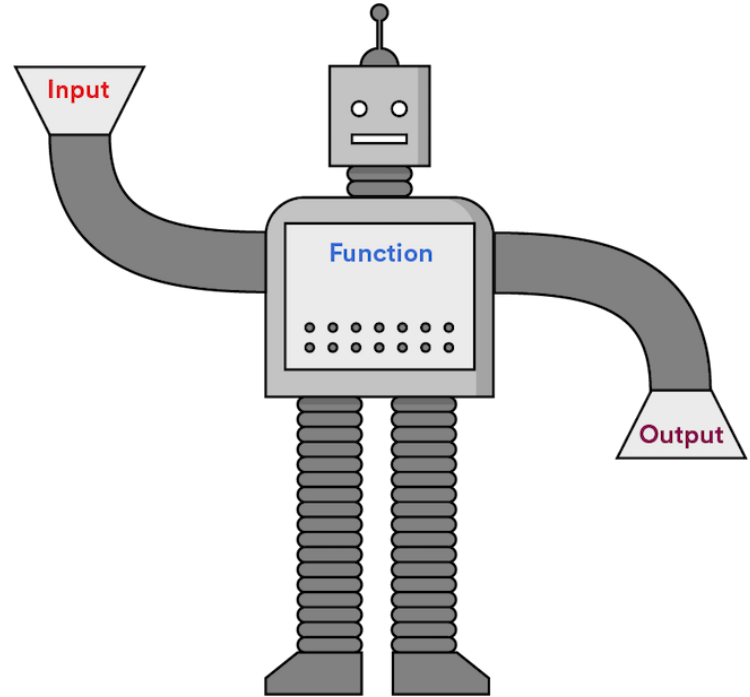
Functions &
Code
Challenges



Functions

Functions are chunks of code that are grouped and will execute together, like a modular program within a program.

A function takes input, performs logic, and returns output.



Defining Functions

`def` required

**Any name
(snake_case)**

**Parameter list
(This function does not
take parameters.)**

**Colon
required**

```
def display_greeting():  
    print('Welcome!')
```

**Indentation
required**

Invoking Functions

Defining a function simply sets things up. For anything to happen, you must **invoke** the function at some point.

```
def greetings():
```

```
    print("hello!")
```


```
# Invoke the function using its name and more parentheses.
```

```
greetings()
```

Function Parameters & Arguments

```
get_total_price(100., 0.1)
```

```
def get_total_price(list_price, tax_rate):  
    total_price = list_price * (1. + tax_rate)  
    return total_price
```

Two red arrows originate from the arguments '100.' and '0.1' in the function call above. One arrow points to the parameter 'list_price' in the function definition, and the other points to the parameter 'tax_rate'.

Arguments

Values passed in to the function.

Parameters

Variable names within the function.

Inside the function body:

```
list_price = 100.
```

```
tax_rate = 0.1
```

parameter = argument



Keyword & Positional Arguments

When providing arguments to a function, you can either rely on the position or on keywords to define the specific inputs.

```
def calculate_area(length, width):
```

```
    return length * width
```

```
calculate_area(8,9)
```

```
calculate_area(width=9, length=8)
```




Discussion:

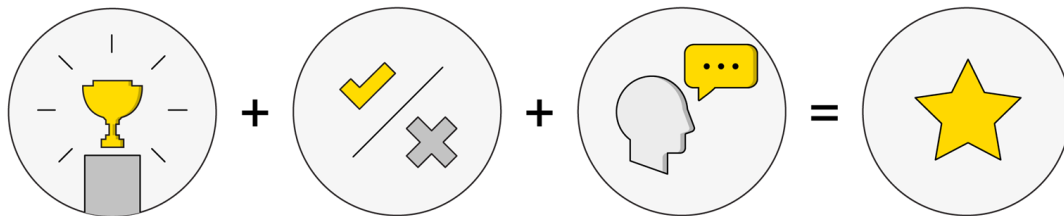
What does this function do?

```
def mystery_function(list_to_check, value):  
    for i in range( len(list_to_check) ):  
        if(list_to_check[i] == value):  
            return i  
  
    return -1
```



Most code challenges you come across will involve writing a function that provides a solution to a given problem.

Section 16 has some good initial code challenges to help you practice writing functions.





Your Journey From Here

To continue building your programming skills in Python, we recommend creating an account at [CodeWars.com](https://www.codewars.com) for code challenges. Let's take a few minutes to set that up before wrapping up for the day.

These challenges will help you learn new techniques through application, so don't be afraid to Google new methods and features of the Python language as you go!

—
Questions?

