Method

+2
Published Oct 21, 2021•**Updated Oct 26, 2022**

Contribute to Docs

Classes can be broken into two core parts:

- The data that is attributed to a class's members or properties.
- The behaviors that are defined or inherited in the class.

Methods are the "behavior" part of the class. When an instance variable is created from a class, it has access to the class's associated methods. Methods can accept parameters (sometimes they're called "arguments") and can return a result.

In object-oriented programming, methods promote reusability and keep functionality encapsulated inside an object.

Example

In the Python example below, a class for a character in a game, `Character`, is defined with certain behaviors. The character can:

- Introduce themselves via `.introduceSelf()`.
- Move left given an integer amount via `.moveLeft()`.

- Move right given an integer amount via `.moveRight()`.

```
class Character:
 def __init__(self, name, movex):
  self.name = "Player"  # Character's name
  self.movex = 0       # Character's starting position

 def introduceSelf(self):
  # Print out an introduction phrase
  print(f"Hello! I'm {self.name}.")

 def moveLeft(self, x):
  # Move the character left by x pixels
  self.movex -= x

 def moveRight(self, x):
  # Move the character right by x pixels
  self.movex += x
```

Now, when an instance of `Character` is created, the game character can introduce themselves, move left, or move right.

Methods in Different Languages

- Methods in C++
- Methods in Java
- Methods in JavaScript
- Methods in Python