

[Previous: Python String](#)[Next: Python Boolean and Conditional Programming](#) >[Home](#) » [Introduction to Python](#) » Output Data To Your Screen With Python's print() Function

PYTHON PRINT

```
print(1234)
```

```
print('My age is:', 12)
```

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

PYTHON.LAND

Output Data To Your Screen With Python's print() Function

May 7, 2022

Before we continue, I need to tell you about the Python `print` function. We've been using the [REPL](#) so far, and the REPL automatically prints anything that is evaluated to the screen. A regular Python program will not do this though. For this reason, Python offers us the `print` function.

What is a function?

We haven't talked about [functions](#) yet, but we will cover functions in-depth soon. For now, it suffices to know that a function is something you can call. It will do something specific for us, and the name often implies what it does. Functions often accept arguments, which they can use to do their specific job.

The Python `print` function

The Python `print()` function can `print text` to our `screen`. We can feed one or more arguments, and these arguments will be `printed` on the `screen`. Let's try some print calls. The following code example is interactive, meaning you can edit it and run it. Read the code, press the run button, and inspect the result:

A couple of observations that you might have caught yourself:

- Python's `print` is called by entering its name followed by parentheses.
- The optional arguments are listed between the parentheses, separated by commas.
- You can `print both` numbers and [strings](#). In fact, most [data types](#) in Python can be printed.
- You can mix argument types, like numbers and strings.
- In its most basic form, we can call `print` without any argument. In this case, it prints an empty `new line`.
- The `print()` function will convert special characters into something meaningful. E.g., the newline character `\n` turns into an actual newline, and the tab character `\t` will get converted into an actual tab.
- `print()` won't escape strings like the REPL does, because we don't want to show escaped strings to the users of our software. It's just a convenience thing the REPL offers us, the programmers.

You're learning about `print` this early because you'll encounter it in almost every Python program and example on the web. Another reason is that the interactive code examples can only show data by using `print`, like the one above. Now that you know `print`, I can include a lot more of these interactive examples which are a great learning aid!

Get certified with our courses

Learn Python properly through small, easy-to-digest lessons, progress tracking, quizzes to test your knowledge, and practice sessions. Each course will earn you a downloadable course certificate.

[Beginners Python Course \(2023\)](#)[Modules, Packages, And Virtual Environments \(2023\)](#)[NumPy Course: The Hands-on Introduction To NumPy \(2023\)](#)[◀ Previous: Python String](#)[Next: Python Boolean and Conditional Programming](#)