

# Theory: Errors

⌚ 8 minutes    7 / 8 problems solved

Start practicing

6568 users solved this topic. Latest completion was about 1 hour ago.

The first thing you need to know about programming is how to print `"Hello, world!"`. The second one is that this might be a challenging task, as even such a tiny script can contain various errors. Here you are:

```
1 | print("Hello, world!")
```

If you run this code, you'll get this:

```
1 | Traceback (most recent call last):
2 |   File "FULL/PATH/TO_YOUR/SCRIPT.PY", line 1
3 |     print("Hello, world!")
4 |           ^
5 | SyntaxError: unexpected EOF while parsing
```

**Traceback** is a stack trace that appears when your code causes an error and it reports detailed information on that particular error, indicating the **specific files** in which the error occurred. Nonetheless, the lines that are the most informative for us right now are the last two. They point out the mistake in your code.

This might seem a little bit frustrating, but generally what errors are designed for is to allow Python to communicate with you. Whenever you see those red threatening lines – don't panic! Just read carefully what they are saying.

## §1. Syntax errors

In the example above, we can clearly see the magic word `SyntaxError` that is likely to haunt you throughout the period of getting used to Python. A large variety of different errors are referred to as syntax errors. What they usually mean is that Python has encountered a problem while trying to **compile** your program and it can't even execute it.

If you read carefully the text of a traceback, it will help you to find mistakes and correct them quite easily, as you can see an **arrow** pointing to the exact place where Python found the mistake in your code. Every syntax error has an **associated value**. It describes an error in detail. In the example, the message `"SyntaxError: unexpected EOF while parsing"` means that something else had been expected after your statement, but you didn't pass it to the interpreter. In our case, there should've been a closing round bracket `)`.

Mistakes won't be so obvious all the time. It is quite likely that the message you'll get as an associated value will be the most common and obscure `"invalid syntax"`, which isn't really helpful. Well, anyway, to locate the problem it's enough to know that the error is in the syntax.

## §2. Common errors for beginners

Some of the most common syntax errors are:

- **wrong spelling** of keywords and function names, e.g. `While` instead of `while`, `pint` instead of `print`;
- the wrong number of **parentheses** in function calls, e.g. `print "just one round bracket"`;
- **indents** are also the fertile soil for errors, therefore, use spaces and tabs carefully;
- **quotes**. Don't forget to wrap a string in quotes of the same type: triple quotes for multi-line strings, double or single quotes for ordinary strings.

Current topic:

✓ [Errors](#)

Stage 13★...

Topic depends on:

✓ [Program execution](#)

Stage 13★...

Topic is required for:

✓ [Exceptions](#)

Stage 13★...

[Debugging in shell](#)

Stage 2...

Table of contents:

- [↑ Errors](#)
- [§1. Syntax errors](#)
- [§2. Common errors for beginners](#)
- [Feedback & Comments](#)

Modern IDEs tend to check everything for you and kindly highlight the places where you have made a mistake or typo, but don't rely on this too much and be ready to read the traceback yourself.


Mind that Python stops compiling your program after finding the **first** Syntax error, so it might take a while to fix every single mistake.

Check the following piece of code, for example. It looks like a Petri dish of syntax errors:

```
1 missing_quote = "this is a mistake!  
2 another_string = this is not a string!"  
3 parted_string = 'I'd like to be highlighted, but'  
4     prnit("I am not")
```

As you can see, there are plenty of syntax errors in this tiny piece of code. If you've checked and corrected everything from the list above and yet you encounter those error messages – don't worry! Once again, it's just Python trying to tell you that something went wrong. Deep breath, reread the article, and continue perfecting your programming skills!

 Report a typo

 Thanks for your feedback!

Start practicing

This content was created over 1 year ago and updated 6 days ago. [Share your feedback below in comments to help us improve it!](#)

[Comments \(9\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)