

Basic Exceptions

Every time that you try and run your code, Python will first try to parse your code, if it encounters something that it does not recognise, it will raise an exception.

This is a warning to the coder that they have done something invalid and the program will not run. Please read the exception and try to understand it.

The three main exceptions you may encounter for now will be:

1. `SyntaxError`
2. `NameError`
3. `TypeError`

1. SyntaxError

A `SyntaxError` is perhaps the most common kind of complaint you get while you are still learning Python. This occurs when the Python interpreter tries to parse the Python code.

It means you have entered something that Python does not understand, this is commonly a spelling mistake or something you have missed.

NOTE: Always pay attention to the error message, it is telling you what is wrong with your code!

Type the following into the console:

```
print(hello world)
```

You will get the following `SyntaxError` because of some missing `""` around the string.

```
File "<stdin>", line 1
  print(hello world)
        ^
SyntaxError: invalid syntax
```

2. NameError

A `NameError` occurs when a local or global name is not found. This refers to variables, functions, and other things like modules and classes.

Basically, Python reserves particular words such as `print`.

Try the following in the **console**

```
printf("Hello World!")
```

You will get the following `NameError` because we have misspelled `print`.

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'printf' is not defined
```

3. TypeError

A `TypeError` occurs when the data types of objects in an operation are invalid. For example, trying to divide a number by a string.

Try the following in the **console**

```
100/"5"
```

You will get the following `TypeError` because we are trying to divide `/` an `int` by a `str`. These two types don't play together well!

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: unsupported operand type(s) for /: 'int' and 'str'
```

=== TASK ===

Copy the following code in `main.py` and run the code.

```
# You need to fix the following lines  
# Run the code and then use the error messages to fix each line  
  
# This line has a SyntaxError  
print(hello world)  
  
# This line has a NameError  
printf("Hello World!")  
  
# The following lines cause a TypeError  
int1 = 100  
str1 = "10"  
print(int1 / str1)
```

You will see a `SyntaxError` on line 5 because of some missing `"` around the string.

```
print(hello world)
```

Fix this. Once you have completed this run the code. You will see a `NameError` on line 8 because `printf` is not a valid name.

```
printf("Hello World!")
```

Fix this. Once you have completed this run the code. You will see a `TypeError` on line 13 because we are trying to divide / an `int` by a `string`.

```
int1 = 100  
int2 = "10"  
print(int1 / int2)
```

Fix the code so that we print `10.0`

References

[Python Exceptions](#)