Common Jupyter error messages

Adapted from interactivepython.org

Parse error

Parse errors happen when you make an error in the syntax of your program.

Type error

Type Errors occur when you you try to combine two objects that are not compatible.

Name error

Name errors almost always mean that you have used a variable before it has a value.

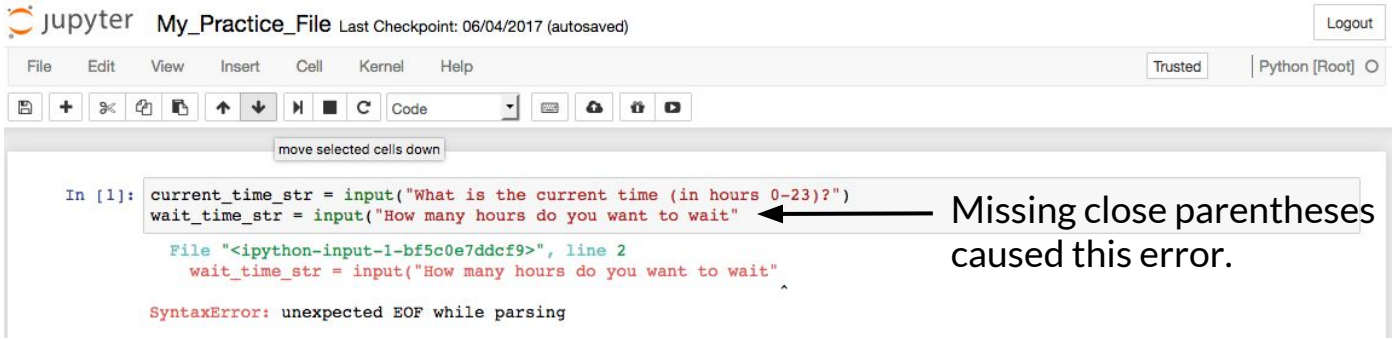
Value error

Value errors occur when the variable expects the value to be a certain data type but it gets something different.

Parse error

Parse errors happen when you make an error in the syntax of your program.

Syntax errors are like making grammatical errors in writing. If you don't use periods and commas in your writing then you are making it hard for other readers to figure out what you are trying to say. Code also has grammatical rules that must be followed or else the program can't figure out what you are trying to say. Usually Parse Errors can be traced back to missing punctuation characters, such as parentheses, quotation marks, or commas.



Type error

Type Errors occur when you you try to combine two objects that are not compatible.

For example you try to add together an integer and a string. Usually type errors can be isolated to lines that are using mathematical operators, and usually the line number given by the error message is an accurate indication of the line.

```
In [19]: number = str(input("What's your favorite number? "))
         What's your favorite number? 7

In [20]: print(number)
         7

In [21]: print(number + 1)
         ---------------------------------------------------------------------------
         TypeError                                 Traceback (most recent call last)
         <ipython-input-21-3ad155d8df06> in <module>()
         ----> 1 print(number + 1)

         TypeError: Can't convert 'int' object to str implicitly
```

Name error

Name errors almost always mean that you have used a variable before it has a value.

Often Name Errors are simply caused by typos in your code. They can be hard to spot if you don't have a good eye for catching spelling mistakes. Other times you may simply mis-remember the name of a variable or even a function you want to call. Or you didn't give the value a variable name yet.

```
In [12]: my_lucky_number = 7
         your_lucky_number = four
         our_lucky_numbers = my_lucky_number + your_lucky_number
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-293e70a8be42> in <module>()
      1 my_lucky_number = 7
----> 2 your_lucky_number = four
      3 our_lucky_numbers = my_lucky_number + your_lucky_number

NameError: name 'four' is not defined
```

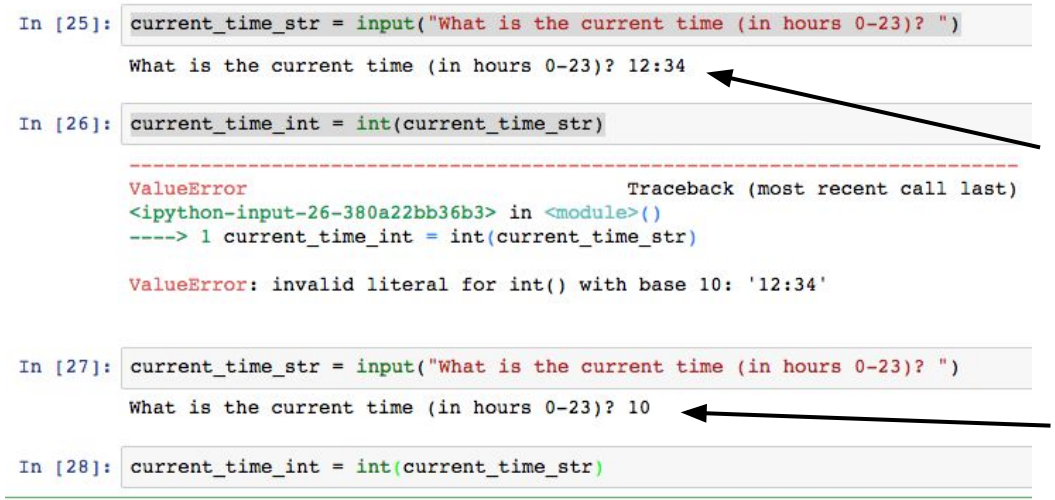
The value here is missing the quote marks so it is not written correctly as a string.

But watch out! The next line will also throw an error. The code is setup to add the two numbers. Strings and ints are different data types and can't be added.

Value error

Value errors occur when the variable expects the value to be a certain data type but it gets something different. This tends to happen when you're asking for user input but did not include specific enough directions about how to enter the data.

Be sure to comment your code, printing very explicit instructions and/or being mindful of how you name the variables which will remind you about what types of values to expect.



This input is correct for how we think about time but does not follow the prompt.

This input does follow the prompt, so it works.