python troubleshooting

Contents

error messages

There are a great number of different `Error`s that can be raised in python. Most of the time, the type of `Error` (e.g., `NameError`, `KeyError`, etc.) plus the associated error message should be enough to help you understand what went wrong, though this is not always the case - especially when you're just starting out.

The list compiled here is not exhaustive, but it should help you get an idea for some of the "most common" error messages that you'll encounter when starting out.

NameError

If you see a message like the following:

Skip to main content

```
NameError: name '<name>' is not defined
```

It means that you're trying to use a variable that has not yet been defined. In a script, you'll need to check that you've actually defined the variable in question before trying to use it. In a jupyter notebook, you'll most likely need to check that you've run the cell where that variable is defined.

ModuleNotFoundError

If you see a message like the following:

```
ModuleNotFoundError: No module named '<module>'
```

It most likely means that you have launched jupyter/python from the wrong conda environment (usually **base** instead of **egm722**).

It may also mean that you haven't installed `<module>` into the current environment, but I would first check what conda environment you've launched from.

AttributeError

If you see a message like the following:

```
AttributeError: '<object>' object has no attribute '<attribute>'
```

Skip to main content

It means that you are attempting to access an attribute of an object that does not exist. Most likely, it means you either have a typo (e.g., `xmas` instead of `xmax`), but there can be other causes as well.

KeyError

If you see a message like the following:

```
KeyError: '<key>'
```

it means that you have tried to use a **Key** in a **dict** (or **pandas.DataFrame**, **geopandas.GeoDataFrame**, or similar object types) that does not exist. Most likely, it means that you haven't added that particular column to the **DataFrame** yet.

warnings

Unlike `Error`s, `Warning`s are not critical. This doesn't mean that you can safely ignore them, but it does mean that your program managed to run without raising an `Exception` that stopped its execution.

There are a great many kinds of `Warning` messages that you might encounter, but these are probably the most common.

FutureWarning

If you see a message like this:

```
FutureWarning: <X> is deprecated; in a future version this will <Y>. Do <Z> instead.
```

Skip to main content

It usually means that the default behavior of something is going to change in a future version. In order for your code to continue working as expected, you'll need to modify it in the way suggested (`<Z>`).

UserWarning

If you see a message like this:

```
UserWarning: The default value for <argument> to <function> will change from <X> to <Y> after <version>
```

it means that in a future version of the package you're using, the default behavior for <function> is going to change. In order to make sure that things continue working as expected, you should explicitly set the value of `<argument>` to `<X>`.

‹ Previous
git troubleshooting

Next ›
programming project