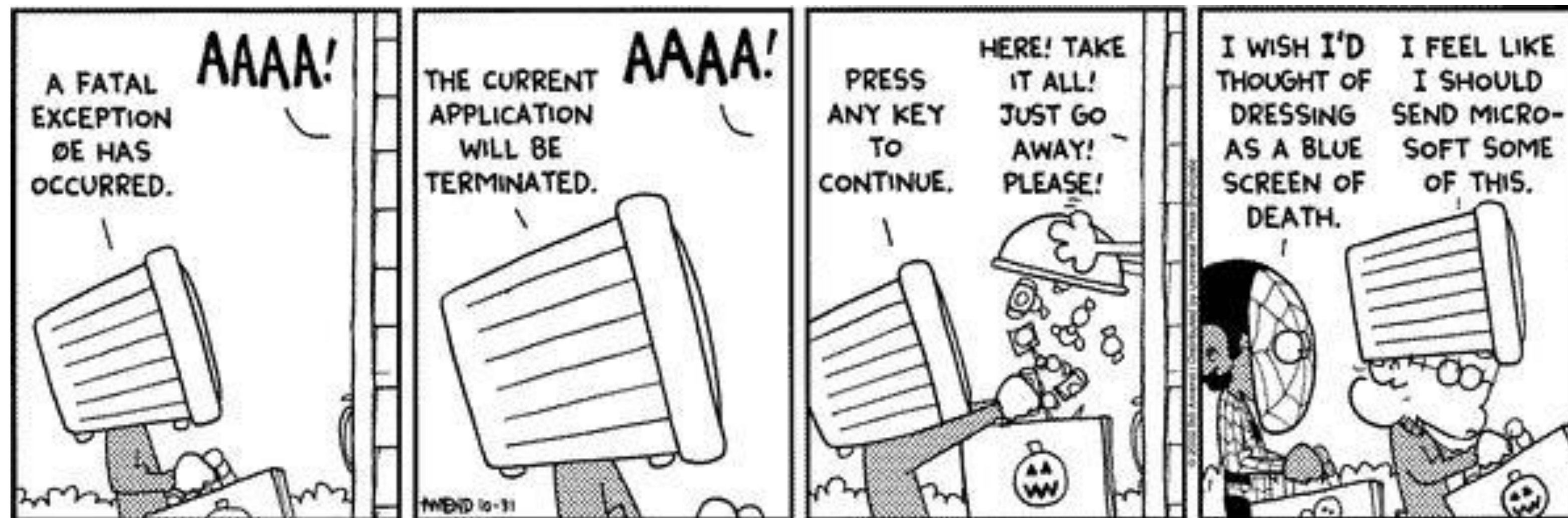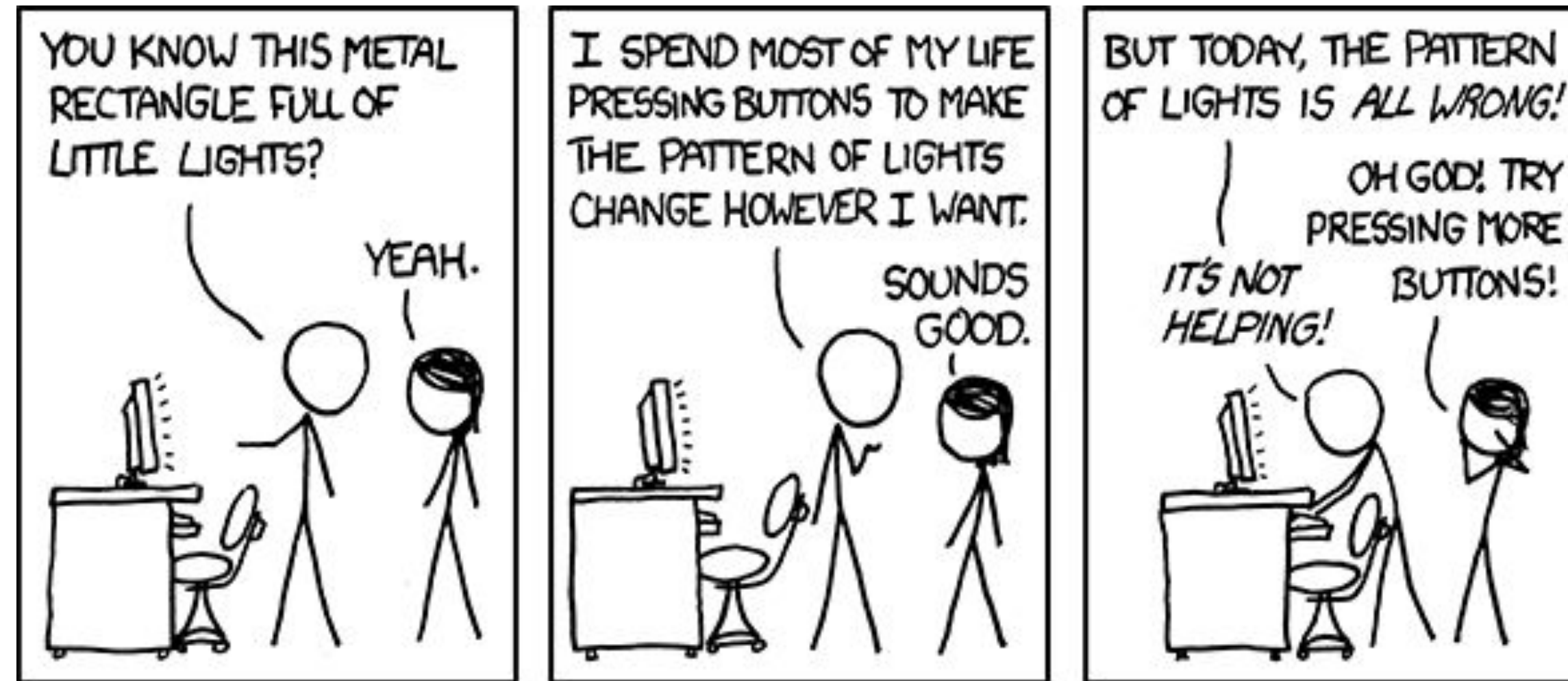# Python Part 5

Spring 2022
PCfB Class 8
March 4, 2022

Outline

- **`pandas`** module

- Types of errors

- Debugging tools/tips

pandas module

Series

```python
l = [1, "1", 2, 5.6]
s=pd.Series(l)
s
```

```
0      1
1      1
2      2
3    5.6
dtype: object
```

```
1  l = [1, "1", 2, 5.6]
2  s=pd.Series(l, index=["int1", "str1", "int2", "float1"])
3  s
```

```
int1          1
str1          1
int2          2
float1      5.6
dtype: object
```

```python
d = {"int1":1, "str1":"1", "int2":2, "float1":5.6}
s=pd.Series(d)
s
```

```
int1         1
str1         1
int2         2
float1     5.6
dtype: object
```

Accessing items in Series

```
3  s
```

```
int1            1
str1            1
int2            2
float1        5.6
dtype: object
```

```
1  s[3]
```

```
5.6
```

```
1  s["float1"]
```

```
5.6
```

```
1  l=list(range(10,50,10))
2  s=pd.Series(l)
3  s
```

```
0    10
1    20
2    30
3    40
dtype: int64
```

```
1  s2=s/10
2  s2
```

```
0    1.0
1    2.0
2    3.0
3    4.0
dtype: float64
```

```
1  s3=s+325
2  s3
```

```
0    335
1    345
2    355
3    365
dtype: int64
```

Dataframe

Convert file to dataframe

Optional arguments

sep

header

names

index_col

Accessing data

Operation	Syntax	Result
Select column	df[col]	Series
Select row by label	df.loc[label]	Series
Select row by integer location	df.iloc[loc]	Series
Slice rows	df[5:10]	DataFrame
Select rows by boolean vector	df[bool_vec]	DataFrame

Example dataset

City	Population	Elevation	Latitude	Longitude
Phoenix	1633000	1086	33.4484	-112.074
Flagstaff	74402	6909	35.1983	-111.6513
St. Louis	308174	466	38.627	-90.1994
Tulsa	402324	722	36.154	-95.9928

"cities.tsv"

```python
import pandas as pd
```

```python
df = pd.read_csv("cities.tsv", sep="\t", header=0, index_col=0)
```

```python
df
```

City	Population	Elevation	Latitude	Longitude
Phoenix	1633000	1086	33.4484	-112.0740
Flagstaff	74402	6909	35.1983	-111.6513
St. Louis	308174	466	38.6270	-90.1994
Tulsa	402324	722	36.1540	-95.9928

Access a single column

```
1  df["Population"]
```

```
City
Phoenix          1633000
Flagstaff          74402
St. Louis         308174
Tulsa             402324
Name: Population, dtype: int64
```

Access a single row - By name

```
1  df.loc["Flagstaff"]
```

```
Population        74402.0000
Elevation          6909.0000
Latitude             35.1983
Longitude          -111.6513
Name: Flagstaff, dtype: float64
```

Access a single row - By position

```
1  df.iloc[1]
```

```
Population        74402.0000
Elevation          6909.0000
Latitude             35.1983
Longitude          -111.6513
Name: Flagstaff, dtype: float64
```

Access specific item

```
1  df["Population"]["Phoenix"]
```

1633000

```
1  df["Elevation"]["Flagstaff"]
```

6909

Access specific item

```
1  df.loc["Phoenix"]["Population"]
```

1633000.0

```
1  df.loc["Flagstaff"]["Elevation"]
```

6909.0

Pandas can be very inefficient!

Pandas demo

Types of errors

Syntax Errors

- Detected before program is run

- Part of your code is not understood by the interpreter

```
File "./rev_comp_v1.py", line 10
    print (revseq.translate(trantab))
        ^
SyntaxError: invalid syntax
```

Runtime Errors (aka, Exceptions)

- Occur when program is executed

- 29 standard exception types
  (https://www.tutorialspoint.com/python/standard_exceptions.htm)

- Format = Traceback

```
Traceback (most recent call last):

  File "./rev_comp_v3.py", line 19, in <module>
    print_rev_comp(seq)

  File "./rev_comp_v3.py", line 10, in print_rev_comp
    print (revseq.translate(trantab))

AttributeError: 'builtin_function_or_method' object
has no attribute 'translate'
```

Tip #1: Look up!

- Problem not necessarily on the line where the error was detected

- Could be on a preceding line

Tool #1: `print()` function

- Prior to the error to check the status of important variables

- Within loops to check whether conditions have been met

Tool #2: Comments

- Temporarily remove sections of code to isolate problem

Use # to comment out a single line

Use "" "" to comment out multiple lines

Tip #2: Interactive interpreter

- Don't forget about the command line interface

- Easy way to test commands

Executing script in working directory

`./fasta2phy.py -f lassa_seqs.fasta`

Please install:

Adobe Illustrator

or

Inkscape

And:

Adobe Photoshop

or

GIMP