DebuggingBasicsClass2-Completed

February 9, 2022

[1]:
```
import pandas as pd
import numpy as np
```

1 Examples

[2]:
```
def squareMyNumber(n):
    square = n*n

    ###############
    return(square)
    ###############
    # if you don't have a return statement, your function will return NONE,␣
 ↪which causes issues


k = squareMyNumber(6)

# If I have 6 friends, and we all bring enough treats for everyone, we'll have␣
 ↪k treats!
# If each treat costs $1.50, how much did we spend?

spent = k * 1.5
```

[3]:
```
mySentenceIsTheBestSentence = "You know, like a liar."

for letter in mySentenceIsTheBestSentence:#### careful of mispellings if the␣
 ↪error says a var is not defined!
    if letter == "l":
        print("It's an l!!!")
```

```
It's an l!!!
It's an l!!!
```

[4]:
```
# 1. get raw URL
my_data_frame = "https://raw.githubusercontent.com/cmparlettpelleriti/
↪CPSC392ParlettPelleriti/master/Data/BreastCancer.csv"

# 2. read_csv
d = pd.read_csv(my_data_frame)

d.head()

#3. potential: if you didn't run the imports at the top of the notebook, you'd␣
↪need to do that here


# hint, look at line 1, and think about ALL the steps you need to do to open a␣
↪url from github in pandas
```

[4]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
0	0.11840	0.27760	0.3001	0.14710	
1	0.08474	0.07864	0.0869	0.07017	
2	0.10960	0.15990	0.1974	0.12790	
3	0.14250	0.28390	0.2414	0.10520	
4	0.10030	0.13280	0.1980	0.10430	

	…	texture_worst	perimeter_worst	area_worst	smoothness_worst	\
0	…	17.33	184.60	2019.0	0.1622	
1	…	23.41	158.80	1956.0	0.1238	
2	…	25.53	152.50	1709.0	0.1444	
3	…	26.50	98.87	567.7	0.2098	
4	…	16.67	152.20	1575.0	0.1374	

	compactness_worst	concavity_worst	concave points_worst	symmetry_worst	\
0	0.6656	0.7119	0.2654	0.4601	
1	0.1866	0.2416	0.1860	0.2750	
2	0.4245	0.4504	0.2430	0.3613	
3	0.8663	0.6869	0.2575	0.6638	
4	0.2050	0.4000	0.1625	0.2364	

	fractal_dimension_worst	Unnamed: 32
0	0.11890	NaN
1	0.08902	NaN

```
2                   0.08758          NaN
3                   0.17300          NaN
4                   0.07678          NaN

[5 rows x 33 columns]
```

[5]:
```
names = ["Chelsea", "Dan", "Craig", "Tony", "Blake", "Cynthia", "Julie"]

for name in names: #####
    print(name + " gets an A+!!!")
```

```
Chelsea gets an A+!!!
Dan gets an A+!!!
Craig gets an A+!!!
Tony gets an A+!!!
Blake gets an A+!!!
Cynthia gets an A+!!!
Julie gets an A+!!!
```

[6]:
```
mySentence = "Can a bee be said to be or not to be an entire bee if half the␣
 ↪bee is not a bee due to an ancient injury?"

words = mySentence.split(" ")
words
```

[6]:
```
['Can',
 'a',
 'bee',
 'be',
 'said',
 'to',
 'be',
 'or',
 'not',
 'to',
 'be',
 'an',
 'entire',
 'bee',
 'if',
 'half',
 'the',
 'bee',
 'is',
 'not',
 'a',
 'bee',
```

```
'due',
'to',
'an',
'ancient',
'injury?']
```