

Day06_CheckIn_Homework_Answer

September 5, 2022

0.1 Answers to the coding check ins!

0.1.1 4.1 Check in answer

```
[ ]: import numpy as np

def get_area(r):
    """give radius, get area"""
    area = np.pi* r**2
    return(area)

get_area(5)
```

0.1.2 4.2 Check in answer (advanced)

```
[ ]: def get_area(r, type):
    """give radius of circle or square, get area"""

    if type == "circle":
        area = np.pi* r**2

    elif type == "square":
        area = r**2

    else:
        print(f"{type} is not a shape I know")
        area = 'no area available'

    return(area)

get_area(2, type = 'triangle')

[ ]: get_area(5, type = "square")
```

0.1.3 7.1 Coding check in answer

```
[ ]: # define our function
def get_hypotenuse (a,b):
    """Takes in the two sides of a right triangle and returns the hypotenuse"""
    hypot = (a**2 + b**2)**(0.5)
    return(hypot)

# define our dictionaries
triangle_1 = {"a": 1, "b":2}
triangle_2 = {"a": 15, "b":20}

# use function to get the hypotenuse
print(get_hypotenuse(**triangle_1))
print(get_hypotenuse(**triangle_2))
```

0.1.4 9.0 Coding check in answer

```
[ ]: # set your directory
loaddir = '../data/' #Make sure the paths end in '/'

# make a dictionary
teams_list = ['Cleveland Cavaliers', 'Chicago Bulls', 'Los Angeles Lakers']
filenames = ['Cavs.txt', 'Bulls.txt', 'Lakers.txt']

bball_dict = dict(zip(teams_list,filenames))

# define your basketball function
def get_bball_stats(sample, team_name):
    """Prints out the mean and max for each team """
    mean_val = round(np.mean(sample),2)
    max_val = round(np.max(sample),2)

    # make the variable name a thing

    print(f'The mean for The {team_name} is {mean_val}m and the tallest player_
↳is {max_val}m')

# loop through your data and use your function

for team in teams_list:
    # read in the data
    file = bball_dict.get(team)
```

```
data = np.loadtxt(loaddir+file, delimiter=',')

get_bball_stats(data, team_name = team)
```

0.1.5 11 Check in answer

```
[ ]: def get_square(*args):
      """Takes in any number of arguments and returns a list of the squares"""
      square = [] # empty list

      for val in args:
          square.append(val**2) # appending squared values for each value in args

      return(square) # return the list of squared values

print(get_square(1,2,3,5,10))
print(get_square(5,10,25))
```

0.2 13 Homework challenge

I'm still thinking about moving! This time I've narrowed down my locations by city! Not only do I not like sharing my beaches, I don't like being too hot or too cold! I've collected some temperature data for you to help me analyze so I can look at where I want to move. I've given you two data files named `city_temps_january.csv` and `city_temps_july.csv`. The columns for each are ordered as Anchorage, San Diego, Chicago, St. Louis, Cleveland, Seattle.

If you open up the `.csv` file on your computer, you'll see there are headers. Reading them in through `numpy` won't like the headers, so we can skip the first row to read in our array using the argument `skiprows = 1`. For more documentation on the `loadtxt` documentation, go [here](#).

```
[ ]: loaddir = '../data/'

jan_data = np.loadtxt(loaddir+'city_temps_january.csv', delimiter=',', skiprows=
    ↪ 1)
```

```
[ ]: ## part one answer

print(jan_data)
print(type(jan_data))
```

Write a function that takes in the `city_name`, `index month` and `data_set` and outputs an `f string` with the mean, min and max for a city. Then loop through each of the cities/indexes and calculate the metrics.

Hints: * You'll have to recall how to index 2d arrays in numpy * load in the july dataset just like we did january * You can pass a dataset to a function as an argument just like we did in sections 8 & 9 * Make a list of the city names (in order!) and index them that way

```
[ ]: ## function

def get_city_info(city_name, index, month, data_set):
    city_mean = np.mean(data_set[:,index]) # takes the mean
    city_max = np.max(data_set[:,index]) # the max
    city_min = np.min(data_set[:,index]) # the min

    print(f"{month} in {city_name}: \nmean is {round(city_mean)}, max is \n
    ↪{round(city_max)}, min is {round(city_min)}")
```

```
[ ]: ## for loop for january

cities = ['Anchorage', 'San Diego', 'Chicago', 'St. Louis', 'Cleveland', \
    ↪'Seattle']

for i in range(0,len(cities)):
    get_city_info(cities[i],i, "January", jan_data)
```

```
[ ]: ## for loop for july

july_data = np.loadtxt(loaddir+'city_temps_july.csv', delimiter=',',skiprows = \
    ↪1)

for i in range(0,len(cities)):
    get_city_info(cities[i],i, "July", july_data)
```

```
[ ]:
```