



Write a function `mousePressed(data, event)` that if the mouse is pressed within the oval, will set it to a random color from `data["colors"]`.

### Problem 3: matplotlib practice

Consider two lists, `week` and `num_coffee`, where `num_coffee[i]` is the number of coffees consumed in `week[i]`

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Num_Coffee	0	4	10	18	20	29	34	2	45	57	63	70	75	84	109	200

Create a scatter plot of the coffee consumed against week number with red.

Now with blue, create a scatter plot only of weeks that had `num_coffee` greater than 10 cups away the average (you will need to compute this too) amount of coffee.

#### Problem 4: using modules

Consider a library called “OneTen” which has the functions:

**isUpperClassmen** - pass in the name of TA and return whether they’re an upperclassman

**getTA** - pass in the section letter and get the name of the TA back.

Now, write a function using the “OneTen” module (make sure to import OneTen!) that loops through sections A-L, gets the names of the TA’s and returns whether the TA is an UpperClassmen or not. Also, pick a name for your OneTen import to use in the rest of your functions to call **isUpperClassmen** and **getTA** from.

Extension: write a function **getRandomUpperClassman(TAs)** that takes in a list of TA’s and returns a random upperclassman TA in the list. Use the **isUpperClassmen** function from the OneTen module.

Starter Code:

```
import OneTen as _____

def getAllTAs():
    # Create a list of all the sections of 110 (letters A through P)

    # Use the OneTen module with whatever import name you gave it to
    # call getTAs for each section

    # Return a list of all TAs

def getUpperClassmen():
    # Find a list of all TA's (hint: use a function you wrote before!)

    # Call isUpperClassmen on the tas in the list of TAs and return a
    # list of TAs who are upperclassmen

def getRandomUpperclassman(TAs):
    # Use the OneTen module to get a random upperclassman
    # from the list of TAs
```