MAT335 homework0-exercises

January 12, 2020

1 Homework 0

Do the programing part of Homework 0 in this notebook. Predefined are function *stubs*. That is, the name of the function and a basic body is predefined. You need to modify the code to fulfil the requirements of the homework.

```python
[6]: # import numpy and matplotlib
     import numpy as np
     import matplotlib.pylab as plt
     # We give the matplotlib instruction twice, because firefox sometimes gets␣
      ↪upset if we don't.
     # note these `%`-commands are not actually Python commands. They are␣
      ↪Jupyter-notebook-specific commands.
     %matplotlib notebook
     %matplotlib notebook
```

```python
[2]: def hi(firstname, lastname):
         print("Hi " + firstname + "." + " What is the origin of " + "'" + lastname␣
      ↪+ "'" + "?")

     hi("Fariha", "Shakeel")
```

```
Hi Fariha. What is the origin of 'Shakeel'?
```

```python
[4]: squared_numbers =[]
     def square_the_list(number_list):
         for number in number_list:
             number = squared_numbers.append(number ** 2)
         return squared_numbers

     square_the_list([-1, 0, 1, 2, 3])
```

```
[4]: [1, 0, 1, 4, 9]
```

1.1 The graphics part

```python
[9]: # Define y=x^2
     def f0(x):
         return x**2
     # Define derivative function
     def f1(x):
         return 2*x
     # Define data range of x for parabola
     x = np.linspace(-5,5,100)
     # Plot tangent line at x=1
     x0 = 1
     y0 = f0(x0)
     # Define tangent line of y=x^2
     def tan_line(x, x0, y0):
         return f1(x0)*(x - x0) + y0
     # Define the range for tangent line of y=x^2
     tan_range = np.linspace(x0-3, x0+3, 10)
     # Plot figure
     plt.figure()
     plt.plot(x, f0(x))
     plt.scatter(x0, y0, color='C1', s=50)
     plt.plot(tan_range, tan_line(tan_range, x0, y0), 'C1--', linewidth = 1)
```

```
[9]: [<matplotlib.lines.Line2D at 0x7f006ca32320>]
```
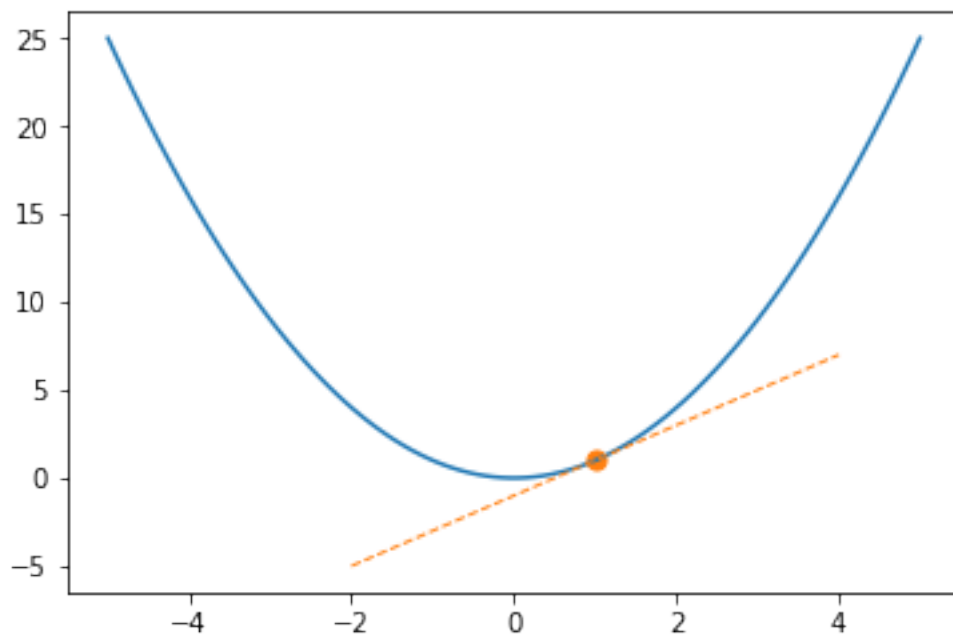
```
[36]: def is_prime(i):
          if i == 0:
              return 0
          elif i == 1:
              return 0
          elif i > 1:
              for number in range(2, i):
                  if i % number == 0:
                      return 0
                      break
              else:
                  return 1


      for i in [19,25,12,4,7,0,3,11,64,2,1,35,13]:
          print(i, "is a prime?", is_prime(i))
```

```
19 is a prime? 1
25 is a prime? 0
12 is a prime? 0
4 is a prime? 0
7 is a prime? 1
0 is a prime? 0
3 is a prime? 1
11 is a prime? 1
64 is a prime? 0
2 is a prime? 1
1 is a prime? 0
35 is a prime? 0
13 is a prime? 1
```

```
[37]: prime_grid = np.array(range(50*50))
      # Use previous function is_prime() to inspect primality of numbers
      for i in range(len(prime_grid)):
          if is_prime(i) == 1:
              prime_grid[i] = 1
          else:
              prime_grid[i] = 0
      prime_grid = prime_grid.reshape(50, 50)
      # plot figure
      fig = plt.figure()
      ax = fig.add_subplot(1, 1, 1)
      ax.imshow(prime_grid, cmap ='Greys')
```

```
[37]: <matplotlib.image.AxesImage at 0x7f006c664a90>
```