assignment02

September 27, 2018

0.0.1 This is script demonstrates the first order Taylor expansion of a given function

0.0.2 Name : Jee-Hye Yang

0.0.3 Student ID : 20145708

0.0.4 import packages for plottion graphs and manipulating data:

```
In [10]: import numpy as np
         import matplotlib.pyplot as plt
```

0.0.5 define my function :

$$f(x) = xsinx$$

```
In [11]: def myFunction(x) :
             f = np.sin(x) * x
             return f
```

0.0.6 define the derivative of my function:

$$f'(x) = xcosx + sinx$$

```
In [12]: def myDerFunction(x) :
             DF = np.cos(x) * x + np.sin(x)
             return DF
```

0.0.7 define the domain of the function

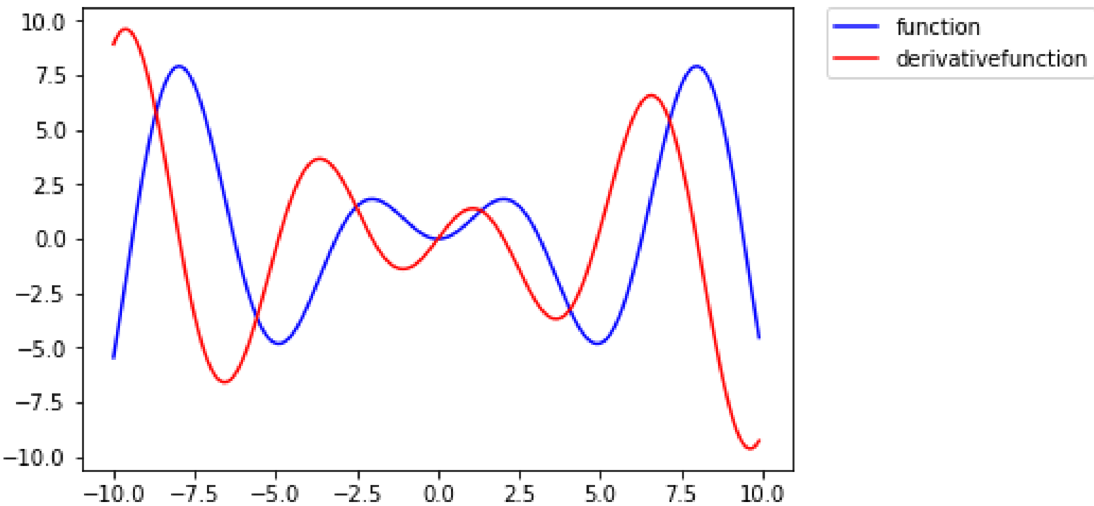
```
In [13]: x = np.arange(-10,10,0.1)
```

0.0.8 computing the graph

```
In [14]: f = myFunction(x)
         DF = myDerFunction(x)
```

0.0.9 plot the graphs for the function and its derivative

```
In [15]: plt.figure(1)
         plt.plot(x,f,'b',label="function")
         plt.plot(x,DF,'r',label="derivativefunction")
         plt.legend(bbox_to_anchor=(1.05, 1), loc =2,borderaxespad=0 )
         plt.show()
```



0.0.10 define the Taylor approximation function :

$$\widehat{f}(x) = f'(c)(x - c) + f(c)$$

```
In [37]: def myTaylorapprxiFunction1(x) :
             TF1 = myDerFunction(a) * (x-a) +  myFunction(a)
             return TF1

         def myTaylorapprxiFunction2(x) :
             TF2 = myDerFunction(b) * (x-b) +  myFunction(b)
             return TF2

         def myTaylorapprxiFunction3(x) :
             TF3 = myDerFunction(c) * (x-c) +  myFunction(c)
             return TF3
```

0.0.11 select the three point that we want

```
In [50]: a = -1
         b = 0
         c = 5
```

0.0.12 computing the graph

```
In [51]: TF1 = myTaylorapprxiFunction1(x)
         TF2 = myTaylorapprxiFunction2(x)
         TF3 = myTaylorapprxiFunction3(x)
```

0.0.13 plot the graphs for the Taylor aproximation function with original function and its derivative

```
In [52]: plt.figure(1)
         plt.plot(x,f,'b',label="function")
         plt.plot(x,DF,'r',label="derivativefunction")
         plt.plot(x,TF1,'g',label = "Taylorapprxifunction1")
         plt.plot(x,TF2,'y',label = "Taylorapprxifunction2")
         plt.plot(x,TF3,'',label = "Taylorapprxifunction3")
         plt.legend(bbox_to_anchor=(1.05, 1), loc =2,borderaxespad=0 )
         plt.show()
```