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

Name : Jun-ha Lee

Student ID : 2017220159

github : https://github.com/myosoo/Assignment02.git

numpy 와 matplotlib 패키지를 불러온다

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

My function을 정의 : $f(x) =$$\cos(x) \over 1 + e^x$

In [2]:

```
def myFunction(x):
    f = np.cos(x) / (1 + np.exp(x))
    return f
```

My derivative function을 정의 : $f'(x) =$$-\sin(x)(1+e^x-\cos(x)e^x) \over (1+e^x)^2$

In [3]:

```
def myDerivativeFunction(x):
    Df = (-np.sin(x)*(1+np.exp(x))-np.cos(x)*np.exp(x)) / ((1+np.exp(x))**2)
    return Df
```

First order Taylor Approximation, x=3

$\widehat{f}(x)\approx$ $f(3) + f'(3)(x-3)$

In [4]:

```
def myDerivativeFunction1(x):
    a = 3
    Df1 = myFunction(a) + myDerivativeFunction(a)*(x-a)
    return Df1
```

First order Taylor Approximation, x=0

$\widehat{f}(x)\approx$ $f(0) + f'(0)(x-0)$

In [5]:

```
def myDerivativeFunction2(x):
    a = 0
    Df2 = myFunction(a) + myDerivativeFunction(a)*(x-a)
    return Df2
```

First order Taylor Approximation, x=-2

$\widehat{f}(x)\approx$ $f(-2) + f'(-2)(x+2)$

In [6]:

```
def myDerivativeFunction3(x):
    a = -2
    Df3 = myFunction(a) + myDerivativeFunction(a)*(x-a)
    return Df3
```

함수의 범위를 지정한다 : $x = [-5 : 0.1 : 5]$

In [7]:

```
x = np.arange(-5, 5, 0.1)
```

함수 계산한다

In [8]:

```
f = myFunction(x)
Df = myDerivativeFunction(x)
Df1 = myDerivativeFunction1(x)
Df2 = myDerivativeFunction2(x)
Df3 = myDerivativeFunction3(x)
```

그래프로 함수 f와 미분함수 Df 그리고 3개의 점에서의 First order Taylor Approximation을 구한다

In [9]:

```
plt.figure(1)
plt.plot(x, f, 'b', label="function")
plt.plot(x, Df, 'r', label="derivative")
plt.plot(x, Df1, 'g', label="derivative at x=3")
plt.plot(x, Df2, 'g', label="derivative at x=0")
plt.plot(x, Df3, 'g', label="derivative at x=-4")
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.show()
```