# Untitled3

March 1, 2018

In [3]:
```
import math
import sympy as sy


x = sy.Symbol('x')
f = sy.exp(x)

# Uso de la funcion de taylor :
def taylor(funcionTaylor,x0,n):
    i = 0
    p = 0
    while i <= n:
        p = p + (funcionTaylor.diff(x,i).subs(x,x0))/(math.factorial(i))*(x-x0)**i
        i +=1
    return p


def evaluarTaylor(n,resultEva):
    funcionTaylor = str(taylor(f,0,n))
    x = resultEva
    return eval(funcionTaylor)

def aitken(x0,x1,x2,resultEva,Iteraciones):
    error = 1
    j = 3
    i = 1
#Formula de aitoken
    a0 = x2-((x2-x1)**2)/(x2-2*x1+x0)
    while(error >= Iteraciones):
        a1 = a0
        x0 = evaluarTaylor(j-2,resultEva)
        x1 = evaluarTaylor(j-1,resultEva)
        x2 = evaluarTaylor(j,resultEva)
        a0 = x2-((x2-x1)**2)/(x2-2*x1+x0)
        error = (abs(a0-a1)/abs(a0))
        j += 1
        i += 1
        print("\t R: ",a0,"   \t E:",error)
```

```
resultEva = 1
x0 = evaluarTaylor(0,resultEva)
x1 = evaluarTaylor(1,resultEva)
x2 = evaluarTaylor(2,resultEva)

E = 1.e-15

aitken(x0, x1, x2, resultEva,E)

 R:  2.7499999999999996          E: 0.09090909090909109
 R:  2.722222222222222          E: 0.010204081632653026
 R:  2.7187500000000004          E: 0.0012771392081734002
 R:  2.7183333333333333          E: 0.00015328019619883848
 R:  2.718287037037037          E: 1.70314229753378e-05
 R:  2.7182823129251696          E: 1.7379033240986566e-06
 R:  2.7182818700396827          E: 1.6292846295018792e-07
 R:  2.718281831765628          E: 1.4080237797666107e-08
 R:  2.7182818287037036          E: 1.1264190708260698e-09
 R:  2.7182818284759573          E: 8.378316917123542e-11
 R:  2.7182818284601415          E: 5.818305134519292e-12
 R:  2.718281828459112          E: 3.786946510310784e-13
 R:  2.718281828459049          E: 2.319872322968696e-14
 R:  2.718281828459045          E: 1.4703416131491758e-15
 R:  2.718281828459045          E: 0.0
```