Aproximación

Luis Eduardo Robles Jimenez

0224969

Input ¶

In [4]:
```
#expression = "2*x**3 - 4*log(x)"
#expression = "x**(1/3)"
#expression = "x**3*log(x)"
expression = "(E**x+E**(-x))/2"
```

Method

In [17]:
```
f = Taylor(expression, 3, 0)
print("\ng(x) =", f, "\n")
#print("f(x) ≈", N(f.subs(x, 2.1)))
```

```
        f(x) =   exp(x)/2 + exp(-x)/2
        f'(x) =  exp(x)/2 - exp(-x)/2
        f''(x) =        exp(x)/2 + exp(-x)/2
        f'''(x) =       exp(x)/2 - exp(-x)/2

g(x) = 1.00000000000000*(x - 0)**0/(0!) + 0*(x - 0)**1/(1!) + 1.00000000000000*(x - 0)**2/(2!) + 0*(x - 0)**3/(3!)

g(x) = 0.5*x**2 + 1.0
```

Taylor

In [16]:
```
def Taylor(function, order, a = 0):
    d, fS = [parse_expr(expression)], ""
    for i in range(order + 1):
        if i > 0:
            d.append(diff(d[i-1], x))
            fS += " + "
        print("\tf", end="")
        for j in range(i):
            print("'", end="")
        print("(x) =\t", d[i])
        fS += str(N(d[i].subs(x, a))) + "*(x - " + str(a) + ")**" + str(i) + "/(" + str(i) + "!)"
    print("\ng(x) =", fS)
    return collect(expand(parse_expr(fS)), x)
```

Run First

In [2]:
```
from sympy import *
x = symbols("x")
```