In [48]:
```
from scipy.integrate import quad
import numpy as np
import scipy as sp
import sympy as smp
from scipy.integrate import quad
from scipy.integrate import cumulative_trapezoid
x=smp.symbols('x')
```

In [9]:
```
def integrand(x):
    return x
B_1, err = quad(integrand, 0, 1)
print ("Value of B_1 is",B_1)
```

```
Value of B_1 is 0.5
```

In [31]:
```
#we know φ0=1 and φ1=x-B
ϕ0=1
ϕ1= x- 0.5
print(ϕ1)
```

```
x - 0.5
```

In [55]:
```
# find φ2

#First let's find B_2
t=smp.integrate((x*(ϕ1**2)),(x,0,1))#to find to numerator part integral
s=smp.integrate(((ϕ1**2)),(x,0,1))#deno integral.
#B_2
B_2=t/s
#C_2
nc=smp.integrate(((x*ϕ1*ϕ0)),(x,0,1))#to find to numerator part integral
dc=smp.integrate(((ϕ0**2)),(x,0,1))#deno integral.
C_2=nc/dc

ϕ2=(((x-B_2)*ϕ1)-(C_2*ϕ0))
ϕ2
```

Out[55]:
$$(x - 0.5000000000000001)\,(x - 0.5) - 0.0833333333333333$$

In [41]:
```
#find φ3

#First find B_3
nb3=smp.integrate((x*(ϕ2**2)),(x,0,1))#to find to numerator part integral
db3=smp.integrate(((ϕ2**2)),(x,0,1))#deno integral.
#B_3
B_3=nb3/db3
#C_3
nc3=smp.integrate(((x*ϕ2*ϕ1)),(x,0,1))
dc3=smp.integrate(((ϕ0**2)),(x,0,1))
C_3=nc3/dc3

ϕ3=(x-B_3)*ϕ2-(C_3*ϕ1)
ϕ3
```

Out[41]:
$$-0.00555555555555559x + (x - 0.500000000000003)\,((x - 0.500000000000001)\,(x - 0.5$$