DİFERANSİYEL DENKLEMLER

In [1]:

```
import sympy as sm
import IPython.display as ipd
f, x, x0, h = sm.symbols('f, x, x0, h')
f = sm.Function('f')

def ft(f, x, h=0, n=3):
    x0 = sm.symbols('x0')
    theSeries = sm.series(f(x), x, x0, n).doit()
    if h == 0:
        ss = theSeries
    else:
        ss = theSeries.subs({x:x0+h, x0:x})
    return ss

def printf(pattern, values):
    valuesLatex = tuple([sm.latex(i).replace(r'\rightarrow', r'\rightarrow ') for i i
n values])
    ipd.display(ipd.Markdown(pattern%valuesLatex))


fxph = ft(f, x, h)
fxmh = ft(f, x, -h)
fx_left = f(x+h) - f(x-h)
fx_right = fxph - fxmh

solwO = sm.expand(sm.solve(fx_left-fx_right, sm.Derivative(f(x), x))[0])
sol = solwO.removeO()
OO = solwO - sol
solO = sm.simplify(sol) + OO
```

In [2]:

```
pattern = "$\Large %s = %s$"
equations = [(f(x+h), fxph),
             (f(x-h), fxmh),
             (fx_left, fx_right),
             (sm.Derivative(f(x), x),solO)
            ]

for eqi in equations:
    printf(pattern, eqi)
```

$$f(h+x) = f(x) + h\frac{d}{dx} f(x) + \frac{h^2 \frac{d^2}{dx^2} f(x)}{2} + O\left(h^3\right)$$

$$f(-h+x) = f(x) - h\frac{d}{dx} f(x) + \frac{h^2 \frac{d^2}{dx^2} f(x)}{2} + O\left(h^3\right)$$

$$-f(-h+x) + f(h+x) = 2h\frac{d}{dx} f(x) + O\left(h^3\right)$$

$$\frac{d}{dx} f(x) = \frac{-f(-h+x)+f(h+x)}{2h} + O\left(h^2\right)$$

In [3]:
```python
from sympy import Matrix, solve_linear_system
a,b,c,d,e = sm.symbols('a,b,c,d,e')
bir = sm.Rational(1)
system = Matrix((
        #(1,1,0), # f(x) hesaba katılmayacak
        (1,-1,1),
        (1,1,0)
       ))

coeffs = solve_linear_system(system, a, b)
lcm_of_coeffs = sm.lcm([coeffs[key].q for key in coeffs])
norm_coeffs = {key:lcm_of_coeffs*coeffs[key] for key in coeffs}

print norm_coeffs
```

```
{b: -1, a: 1}
```

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$$

In [4]:
```python
import sympy as sm
import IPython.display as ipd
f, x, x0, h = sm.symbols('f, x, x0, h')
f = sm.Function('f')

def ft(f, x, h=0, n=3):
    x0 = sm.symbols('x0')
    theSeries = sm.series(f(x), x, x0, n).doit()
    if h == 0:
        ss = theSeries
    else:
        ss = theSeries.subs({x:x0+h, x0:x})
    return ss

def printf(pattern, values):
    valuesLatex = tuple([sm.latex(i).replace(r'\rightarrow', r'\rightarrow ') for i i
n values])
    ipd.display(ipd.Markdown(pattern%valuesLatex))

n=5
fxph = ft(f, x, h, n)
fxmh = ft(f, x, -h, n)
fxp2h = ft(f, x, 2*h, n)
fxm2h = ft(f, x, -2*h, n)


fx_left = f(x-2*h) - 8*f(x-h) + 8*f(x+h) - f(x+2-h)
fx_right = fxm2h - 8*fxmh + 8*fxph - fxp2h

solwO = sm.expand(sm.solve(fx_left-fx_right, sm.Derivative(f(x), x))[0])
sol = solwO.removeO()
OO = solwO - sol
solO = sm.simplify(sol) + OO
```

In [5]:

```python
pattern = "$\Large %s = %s$"
equations = [(f(x+h), fxph),
             (f(x-h), fxmh),
             (f(x+2*h), fxp2h),
             (f(x-2*h), fxm2h),
             (fx_left, fx_right),
             (sm.Derivative(f(x), x),solO)
            ]

for eqi in equations:
    printf(pattern, eqi)
```

$$f(h+x) = f(x) + h\frac{d}{dx}f(x) + \frac{h^2 \frac{d^2}{dx^2} f(x)}{2} + \frac{h^3 \frac{d^3}{dx^3} f(x)}{6} + \frac{h^4 \frac{d^4}{dx^4} f(x)}{24} +$$

$$f(-h+x) = f(x) - h\frac{d}{dx}f(x) + \frac{h^2 \frac{d^2}{dx^2} f(x)}{2} - \frac{h^3 \frac{d^3}{dx^3} f(x)}{6} + \frac{h^4 \frac{d^4}{dx^4} f(x)}{24}$$

$$f(2h+x) = f(x) + 2h\frac{d}{dx}f(x) + 2h^2\frac{d^2}{dx^2}f(x) + \frac{4h^3 \frac{d^3}{dx^3} f(x)}{3} + \frac{2h^4 \frac{d^4}{dx^4}}{3}$$

$$f(-2h+x) = f(x) - 2h\frac{d}{dx}f(x) + 2h^2\frac{d^2}{dx^2}f(x) - \frac{4h^3 \frac{d^3}{dx^3} f(x)}{3} + \frac{2h^4}{}$$

$$f(-2h+x) - 8f(-h+x) + 8f(h+x) - f(-h+x+2) = 12h\frac{}{d}$$

$$\frac{d}{dx}f(x) = \frac{f(-2h+x)-8f(-h+x)+8f(h+x)-f(-h+x+2)}{12h} + O\left(h^4\right)$$

In [6]:

```python
from sympy import Matrix, solve_linear_system
a,b,c,d,e = sm.symbols('a,b,c,d,e')
bir = sm.Rational(1)
system = Matrix((
        #(1, 1, 1, 1, 0),
        (1, -1, 2, -2, 1),
        (bir/2, bir/2, 2, 2, 0),
        (bir/6,-bir/6, 4*bir/3, -4*bir/3,0),
        (bir/24, bir/24, 2*bir/3, 2*bir/3, 0)
       ))

coeffs = solve_linear_system(system, a, b, c, d)
lcm_of_coeffs = sm.lcm([coeffs[key].q for key in coeffs])
norm_coeffs = {key:lcm_of_coeffs*coeffs[key] for key in coeffs}

print norm_coeffs
```

```
{c: -1, b: -8, a: 8, d: 1}
```

In [7]:
```python
import sympy as sm
import IPython.display as ipd
f, x, x0, h = sm.symbols('f, x, x0, h')
f = sm.Function('f')

def ft(f, x, h=0, n=3):
    x0 = sm.symbols('x0')
    theSeries = sm.series(f(x), x, x0, n).doit()
    if h == 0:
        ss = theSeries
    else:
        ss = theSeries.subs({x:x0+h, x0:x})
    ssl = sm.latex(ss).replace(r'\rightarrow', r'\rightarrow ')
    return ss

def printf(pattern, values):
    valuesLatex = tuple([sm.latex(i).replace(r'\rightarrow', r'\rightarrow ') for i in values])
    ipd.display(ipd.Markdown(pattern%valuesLatex))

n=6
fxph = ft(f, x, h, n)
fxmh = ft(f, x, -h, n)
fxp2h = ft(f, x, 2*h, n)
fxm2h = ft(f, x, -2*h, n)


fx_left = -f(x-2*h) + 16*f(x-h) - 30*f(x) + 16*f(x+h) - f(x+2-h)
fx_right = -fxm2h + 16*fxmh - 30*f(x) + 16*fxph - fxp2h

solwO = sm.expand(sm.solve(fx_left-fx_right, sm.Derivative(f(x), x, x))[0])
sol = solwO.removeO()
OO = solwO - sol
solO = sm.simplify(sol) + OO
```

In [8]:

```python
pattern = "$\Large %s = %s$"
equations = [(f(x+h), fxph),
             (f(x-h), fxmh),
             (f(x+2*h), fxp2h),
             (f(x-2*h), fxm2h),
             (fx_left, fx_right),
             (sm.Derivative(f(x), x, x),solO)
            ]

for eqi in equations:
    printf(pattern, eqi)
```

$$f(h+x) = f(x) + h\frac{d}{dx}f(x) + \frac{h^2\frac{d^2}{dx^2}f(x)}{2} + \frac{h^3\frac{d^3}{dx^3}f(x)}{6} + \frac{h^4\frac{d^4}{dx^4}f(x)}{24} +$$

$$f(-h+x) = f(x) - h\frac{d}{dx}f(x) + \frac{h^2\frac{d^2}{dx^2}f(x)}{2} - \frac{h^3\frac{d^3}{dx^3}f(x)}{6} + \frac{h^4\frac{d^4}{dx^4}f(x)}{24}$$

$$f(2h+x) = f(x) + 2h\frac{d}{dx}f(x) + 2h^2\frac{d^2}{dx^2}f(x) + \frac{4h^3\frac{d^3}{dx^3}f(x)}{3} + \frac{2h^4\frac{d^4}{dx}}{3}$$

$$f(-2h+x) = f(x) - 2h\frac{d}{dx}f(x) + 2h^2\frac{d^2}{dx^2}f(x) - \frac{4h^3\frac{d^3}{dx^3}f(x)}{3} + \frac{2h^4}{}$$

$$-30f(x) - f(-2h+x) + 16f(-h+x) + 16f(h+x) - f(-h+$$

$$\frac{d^2}{dx^2}f(x) = \frac{-30f(x) - f(-2h+x) + 16f(-h+x) + 16f(h+x) - f(-h+x+2)}{12h^2} + O\left(h^4\right)$$

In [9]:

```python
from sympy import Matrix, solve_linear_system
a,b,c,d,e = sm.symbols('a,b,c,d,e')
bir = sm.Rational(1)
system = Matrix((
        #(1, 1, 1, 1, 1),
        (1, -1, 2, -2, 0),
        (bir/2, bir/2, 2, 2, 1),
        (bir/6,-bir/6, 4*bir/3, -4*bir/3,0),
        (bir/24, bir/24, 2*bir/3, 2*bir/3, 0),
        (bir/120, -bir/120, 4*bir/15, -4*bir/15, 0)
       ))

coeffs = solve_linear_system(system, a, b, c, d, e)
print coeffs
```

```
{c: -1/12, b: 4/3, a: 4/3, d: -1/12}
```

In [10]:

```
lcm_of_coeffs = sm.lcm([coeffs[key].q for key in coeffs])
norm_coeffs = {key:lcm_of_coeffs*coeffs[key] for key in coeffs}

print norm_coeffs
```

```
{c: -1, b: 16, a: 16, d: -1}
```

In [ ]: