In [1]:

```
# Styling notebook
from IPython.core.display import HTML
def css_styling():
    styles = open("./styles/custom.css", "r").read()
    return HTML(styles)
css_styling()
```

Out[1]:

### Simple formulas or arithmetic/geometric sums

sumArith(start,d,n,startIndex) $\leftarrow \sum_{i=\text{startIndex}}^{n} a_i$,
where $a_{\text{startIndex}} = \text{start}$ and $a_i$ is arithmetic with $d$.

In [1]:

```
def sumArith1(start,d,n) :
    return ((n+1)/2)*(start + (start + n*d))

def sumArith(start,d,n,startIndex) :
    return sumArith1(start,d,n-startIndex)
```

In [6]:

```
sumArith(2,6,10,0)
```

Out[6]:

```
352.0
```

sumGeom(start,r,n,startIndex) $\leftarrow \sum_{i=\text{startIndex}}^{n} a_i$,
where $a_{\text{startIndex}} = \text{start}$ and $a_i$ is geometric with $r$.

In [6]:

```
def sumGeom1(start,r,n) :
    # Since r can be a floating point, don't ever compare
    # it to zero without a "zero" interval - here it's 1e-6 (ymmv)
    if abs(r-1.0) < 1e-6 : return (n + 1)*start
    return start*(1 - r**(n+1))/(1 - r)

def sumGeom(start,r,n,startIndex) :
    return sumGeom1(start,r,n-startIndex)
```

In [7]:

```
sumGeom(-2,2,7,0)
```

Out[7]:

```
-510.0
```

### Closed formulas for sums of linear, quadratic and cubic terms

In [7]:

```
def sumLinears(n) :
    return (n*(n + 1)/2)
```

In [8]:

```
sumLinears(100)
```

Out[8]:

```
5050.0
```

In [2]:

```
def sumSquares(n) :
    return (n*(n + 1)*(2*n + 1)/6)
```

In [3]:

```
sumSquares(10)
```

Out[3]:

```
385.0
```

In [4]:

```
def sumCubes(n) :
    return (n**2*(n + 1)**2/4)
```

In [3]:

```
sumCubes(20)
```

Out[3]:

```
44100.0
```