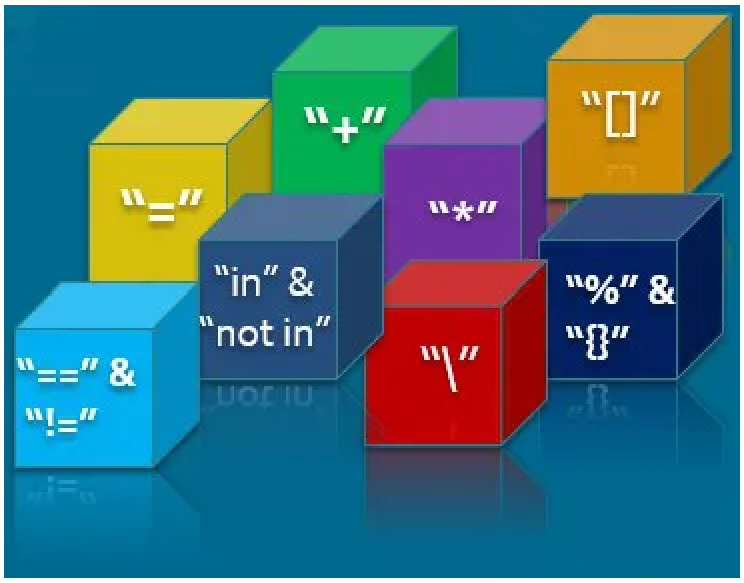
String Operations



Assignment Operator =

We can bind or assign a string to another variable:

In [1]:
```
# Assign string to variable
Name = "Big Data Analytics"
Name
```

Out[1]:
```
'Big Data Analytics'
```

Concatenate Operator +

In [1]:
```
Name1 = "Big Data"
Name2 = " Analytics"
print(Name1)
print(Name2)

Name = Name1 + Name2
print(Name)
```

```
Big Data
 Analytics
Big Data Analytics
```

Repetition Operator *

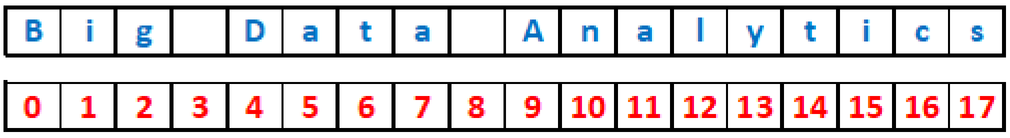
In [2]:
```
print(Name1 * 4)
```

```
Big DataBig DataBig DataBig Data
```

Indexing

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

It is helpful to think of a string as an ordered sequence. Each element in the sequence can be accessed using an index represented by the array of numbers:



B	i	g		D	a	t	a		A	n	a	l	y	t	i	c	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

In [1]:
```
Name = "Big Data Analytics"
```

In [7]:
```
print(Name)
```

```
Big Data Analytics
```

In [8]:
```
print(Name[6])
```

```
t
```

The first index can be accessed as follows:

[Tip]: Because indexing starts at 0, it means the first index is on the index 0.

Slicing operator [ ]

In [4]:
```
# Print the first element in the string
print(Name[0])
```

```
B
```

We can access index 6:

In [5]:
```
# Print the element on index 6 in the string
print(Name[6])
```

```
t
```

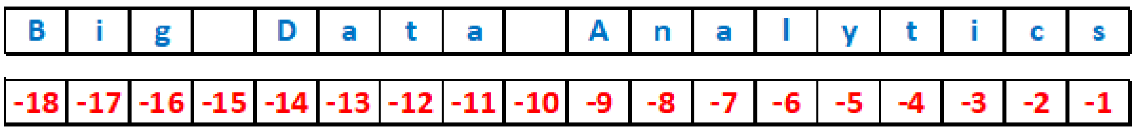
Moreover, we can access the 13th index:

In [6]:
```
# Print the element on the 13th index in the string
print(Name[13])
```

```
y
```

Negative Indexing

We can also use negative indexing with strings:



B	i	g		D	a	t	a		A	n	a	l	y	t	i	c	s
-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Negative index can help us to count the element from the end of the string.

In [2]:
```python
len(Name)
```

Out[2]:
```
18
```

The last element is given by the index -1:

In [7]:
```python
# Print the last element in the string
print(Name[-1])
```

```
s
```

The first element can be obtained by index -17:

In [8]:
```python
# Print the first element in the string
print(Name[-17])
```

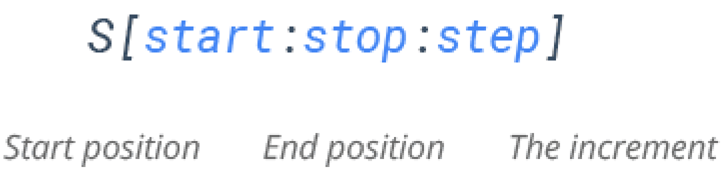
```
i
```

We can find the number of characters in a string by using `len`, short for length:

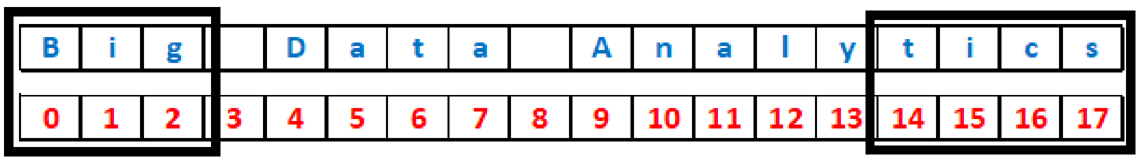
In [9]:
```python
# Find the length of string
len(Name)
```

Out[9]:
```
18
```

Slicing



We can obtain multiple characters from a string using slicing, we can obtain the 0 to 2nd and 14th to the 17th element:



[Tip]: When taking the slice, the first number means the index (start at 0), and the second number means the length from the index to the last element you want (start at 1)

In [7]:
```python
Name[4:8]
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Default Values --> start = 0 : stop = len(S) : step = 1

```
In [1]: Name = 'Big Data Analytics'
```

```
In [2]: Name[0:15:2]
Out[2]: 'BgDt nlt'
```

```
In [10]: Name[::]      # Name[0:18:1]
Out[10]: 'Big Data Analytics'
```

```
In [12]: Name[0:3]
Out[12]: 'Big'
```

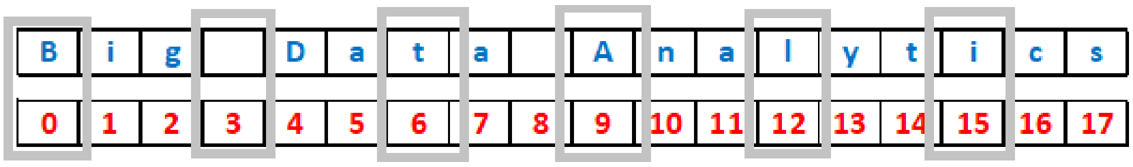
```
In [13]: Name[:3]
Out[13]: 'Big'
```

```
In [14]: Name[4:]    # 4:18:1
Out[14]: 'Data Analytics'
```

```
In [18]: Name[4::1] # 4:18:1
Out[18]: 'Data Analytics'
```

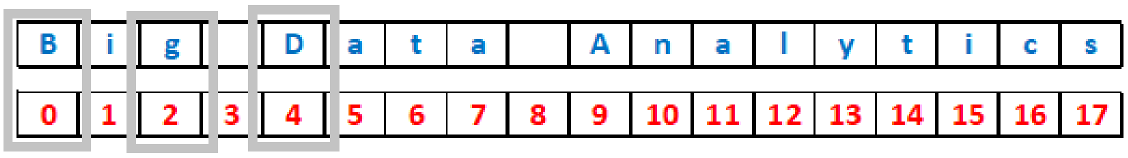
Stride

We can also input a stride value as follows, with the '2' indicating that we are selecting every second variable:



```
In [21]: # Get every second element. The elments on index 1, 3, 5 ...
         Name[::2]    # [0:18:2]
Out[21]: 'BgDt nltc'
```

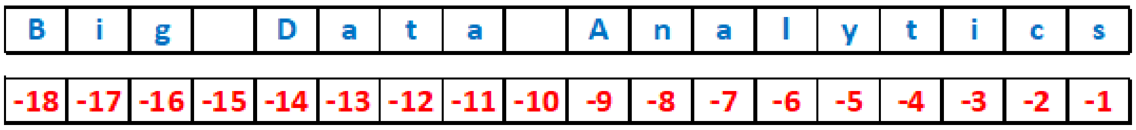
We can also incorporate slicing with the stride. In this case, we select the first five elements and then use the stride:



```
In [22]: Name[0:5:2]
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[22]:
```
'BgD'
```



B	i	g		D	a	t	a		A	n	a	l	y	t	i	c	s
-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Default Values for -ve indexing --> start = -len(S) : stop = -1 : step = 1

In [23]:
```python
Name[:-1]
```
Out[23]:
```
'Big Data Analytic'
```

In [24]:
```python
Name[-18:-1:1]
```
Out[24]:
```
'Big Data Analytic'
```

In [8]:
```python
Name[-18:0:1]   # wrong usage
```
Out[8]:
```
''
```

In [5]:
```python
Name[-18::1]
```
Out[5]:
```
'Big Data Analytics'
```

In [25]:
```python
Name[::-1]
```
Out[25]:
```
'scitylanA ataD giB'
```

In [9]:
```python
Name[::-2]
```
Out[9]:
```
'siyaAaa i'
```

In [26]:
```python
Name[0:18:-1]
```
Out[26]:
```
''
```

In [29]:
```python
Name[17:0:-1]
```
Out[29]:
```
'scitylanA ataD gi'
```

In [10]:
```python
Name[17::-1]
```
Out[10]:
```
'scitylanA ataD giB'
```

In [11]:
```python
print(Name[17::-1])
```
```
scitylanA ataD giB
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js