String Slicing

In this notebook we take a look at a very useful feature in Python called slicing.

Slicing enables us to extract subsets of a sequence. In this notebook we study slicing as it applies to extracting portions of a string.

The general form is `str_variable[start:end:step]`

`str_variable` : is any string variable or string literal

`start` : is an integer representing the starting index

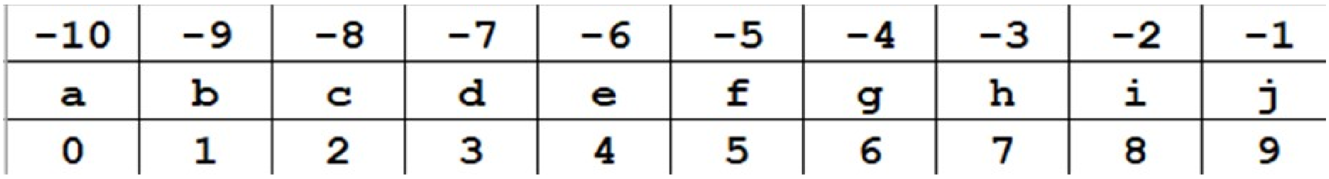
`end` : is an integer representing the end index. The last valid integer before the end index will be the last value returned

`step` : is an integer representing the step value (the increment)

In [4]:

```
my_str = 'abcdefghij'
print('my_str', my_str)
```

```
my_str abcdefghij
```



-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
a	b	c	d	e	f	g	h	i	j
0	1	2	3	4	5	6	7	8	9

In [5]:

```
print('my_str[0:3]=', my_str[0:3])
```

```
my_str[0:3]= abc
```

In [6]:

```
print('my_str[4:8]=', my_str[4:8])
```

```
my_str[4:8]= efgh
```

In [7]:

```
print('my_str[2:5]=', my_str[2:5])
```

```
my_str[2:5]= cde
```

In [8]:

```
print('my_str[-5:-3]=', my_str[-5:-3])
```

```
my_str[-5:-3]= fg
```

In [9]:

```
print('my_str[:3]=', my_str[:3])
```

```
my_str[:3]= abc
```

In [10]:

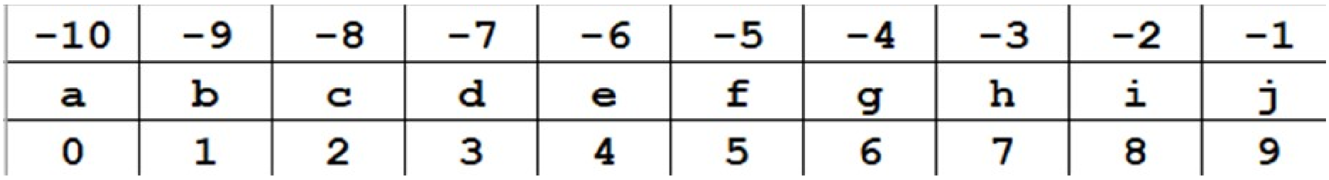
```
print('my_str[4:]=', my_str[4:])
```

```
my_str[4:]= efghij
```

In [11]:

```
print('my_str[:]=', my_str[:])
```

```
my_str[:]= abcdefghij
```



-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
a	b	c	d	e	f	g	h	i	j
0	1	2	3	4	5	6	7	8	9

In [12]:

```
#if the character at the start index comes after (or is) the character at the end index and the step is positive,
#nothing is printed
print('my_str[-8:2]=', my_str[-8:2])
```

```
my_str[-8:2]=
```

In [13]:

```
#if the start index is less than -n and step is positive, slicing starts at index 0
print('my_str[-18:3]=', my_str[-18:3])
```

```
my_str[-18:3]= abc
```

In [14]:

```
#if the start index is less than -n and step is negative, nothing is printed
print('my_str[-18:3:-1]=', my_str[-18:3:-1])
```

```
my_str[-18:3:-1]=
```

In [15]:

```
# if the end value is greater than the length of the string, all the characters to the end are displayed
print('my_str[4:100]=', my_str[4:100])
```

```
my_str[4:100]= efghij
```

In [16]:

```
print('my_str[2:-2:2]=', my_str[2:-2:2])
```

```
my_str[2:-2:2]= ceg
```

In [17]:

```
print('my_str[::2]=', my_str[::2])
```

```
my_str[::2]= acegi
```

In [18]:

```
print('my_str[::-1]=', my_str[::-1]) # This prints the string is reverse
```

```
my_str[::-1]= jihgfedcba
```

In [19]:

```
print('my_str[2:2]=', my_str[2:2])
```

```
my_str[2:2]=
```