

The `range()` function

This notebook introduces the `range()` function. The `range()` consists of three parts. `range(start, end, step)`. Of these only the `end` is required. The default value of `start` is 0, and the default value of `step` is 1. The iteration variable iterates from the `start` value to one less than the `end` value in increments specified by the `step` variable. The iteration always stops just before the `end` variable. If the `step` value is negative, the iteration will stop at one more than the `end` variable. If the `step` variable is positive then the iteration will stop at one less than the `end` variable.

In [20]:

```
for i in range(0,5,1):  
    print(i)
```

0
1
2
3
4

In [21]:

```
'''  
The example above can also be written as  
'''  
for i in range(5):  
    print(i)
```

0
1
2
3
4

In [22]:

```
'''  
Note also, the slight variation in the print statement.  
The default end character when using a print statement is the newline character. But we can change it to  
something else of our preference. In this code, we have changed it to a ','. Hence each time the  
print statement  
is executed, it will end with a comma  
'''  
for i in range(5, 10):  
    print(i,end = ',') # Note the variation  
print('Done') #We now add this regular print statement to ensure that a newline character is printed at the end of the sequence
```

5,6,7,8,9,Done

In [23]:

```
'''  
The iteration variable starts at 5 and ends at 19 (one less than 20) in increments of 3.  
'''  
for i in range(3, 20, 3):  
    print(i, end = ':')  
print('Done')
```

3:6:9:12:15:18:Done

In [24]:

```
'''  
In this example, the iteration goes backwards. It begins at 20 and works down to 6 (one more than
```

```
In this example, the iteration goes backwards. It begins at 20 and works down to 0 (one more than 5) decreasing each time by 3.
'''
for i in range(20, 5, -3):
    print(i, end = ' ') #Note that in this example, we have set the end value in the print to ' '
    (empty space)
print('Done')
```

20 17 14 11 8 Done

In [25]:

```
'''
In this example, since the step value is omitted, it will take on a default of 1. Therefore, since the start is greater than the end nothing is printed. When the step value is positive, the start must always be less than the end for something to be printed.
'''
for i in range(10, 5):
    print(i, end = ' ')
print('Done')
```

Done

In [26]:

```
for i in range(0,12,2):
    print(i, end = ' ')
print('Done')
```

0 2 4 6 8 10 Done

In [27]:

```
# This range expression has a negative step value. In this case, the start value must always be larger than the end value
for i in range(100, 0, -10):
    print(i, end = ',')
print('Done')
```

100,90,80,70,60,50,40,30,20,10,Done

In [28]:

```
for i in range(10, -5, -3):
    print(i, end = ';')
print('Done')
```

10;7;4;1;-2;Done

In [29]:

```
for i in range(-3, 4):
    print(i, end = ' ')
print('Done')
```

-3 -2 -1 0 1 2 3 Done

In [30]:

```
for i in range(0, 10, 1):
    print(i, end = ':')
print('Done')
```

0:1:2:3:4:5:6:7:8:9:Done

