enumerate()

In this lecture we will learn about an extremely useful built-in function: enumerate(). Enumerate allows you to keep a count as you iterate through an object. It does this by returning a tuple in the form (count,element). The function itself is equivalent to:

```
def enumerate(sequence, start=0):
    n = start
    for elem in sequence:
        yield n, elem
        n += 1
```

Example ¶

In [1]:

```
lst = ['a','b','c']

for number,item in enumerate(lst):
    print(number)
    print(item)
```

```
0
a
1
b
2
c
```

enumerate() becomes particularly useful when you have a case where you need to have some sort of tracker. For example:

In [2]:

```
for count,item in enumerate(lst):
    if count >= 2:
        break
    else:
        print(item)
```

```
a
b
```

enumerate() takes an optional "start" argument to override the default value of zero:

In [3]:

```
months = ['March','April','May','June']

list(enumerate(months,start=3))
```

Out[3]:

```
[(3, 'March'), (4, 'April'), (5, 'May'), (6, 'June')]
```

Great! You should now have a good understanding of enumerate and its potential use cases.