Python - Map Function

The `map()` function is a built-in function.

map() Signature:
```
map(function, iterable [, iterable2, iterable3,...iterableN]) --> map object
```

The `map()` function calls the specified function for each item of an iterable (such as string, list, tuple or dictionary) and returns a list of results.

Consider the following simple square function.

```
def square(x):
    return x*x
```

Now, we can call the map function with the list of numbers to get the list of results, as shown below.

```
>>> numbers=[1, 2, 3, 4, 5]
>>> sqrList=map(square, numbers))
>>> next(sqrList)
1
>>> next(sqrList)
4
>>> next(sqrList)
9
>>> next(sqrList)
16
>>> next(sqrList)
25
```

In the above example, the `map()` function applies to each element in the numbers[] list. This will return a map object which is iterable and so, we can use the `next()` function to traverse the list.

Map with Lambda Expression

The `map()` function passes each element in the list to the built-in function, a lambda function or a user-defined function, and returns the mapped object. The following `map()` is used with the lambda function.

```
>>> sqrList = map(lambda x: x*x, [1, 2, 3, 4])
>>> next(sqrList)
1
>>> next(sqrList)
4
```

```
>>> next(sqrList)
9
>>> next(sqrList)
16
>>> next(sqrList)
25
```

Map with Built-in Function

In the following example, a built-in function pow() is given to map two list objects, one for each base and index parameter. The result is a list containing the power of each number in bases raised to the corresponding number in the index.

```
>>> bases=[10, 20, 30, 40, 50]
>>> index=[1, 2, 3, 4, 5]
>>> powers=list(map(pow, bases, index))
>>> powers
[10, 400, 27000, 2560000, 312500000]
```