Python Built-in Functions: zip(), lambda, map()

1. zip() Function

The `zip()` function takes two or more iterables (like lists, tuples) and returns an iterator of tuples, where the i-th tuple contains the i-th element from each iterable.

It stops when the shortest input iterable is exhausted.

Example:

```
list1 = [1, 2, 3]
list2 = ['a', 'b', 'c']
zipped = zip(list1, list2)
print(list(zipped))
```

Output:

```
[ (1, 'a'), (2, 'b'), (3, 'c') ]
```

Explanation: Each element of list1 is paired with the element at the same index in list2.

2. lambda Function

A `lambda` function is an anonymous, single-expression function.

Syntax: `lambda arguments: expression`

Used when a short function is needed without formally defining it with `def`.

Example:

```
add = lambda x, y: x + y
print(add(5, 3))
```

Output:

```
8
```

Explanation: The lambda function returns the sum of x and y.

3. map() Function

The `map()` function applies a given function to all the items in an iterable (like list).

It returns a map object, which can be converted to a list or other iterables.

Example:

```
nums = [1, 2, 3, 4]
doubled = map(lambda x: x * 2, nums)
print(list(doubled))
```

Output:

```
[2, 4, 6, 8]
```

Explanation: The lambda function is applied to each element of `nums`, doubling them.