

## Python Destructuring


Review the topic of destructuring and unpacking in Python with this e-book. Check out the table of contents to navigate to each topic.

### Contents

- 1: Unpacking
- 2: The 'enumerate' function
- 3: The 'zip' function


## Unpacking

Used to assign values from an iterable to variables in a single statement.



```
x, y, z = collection
```

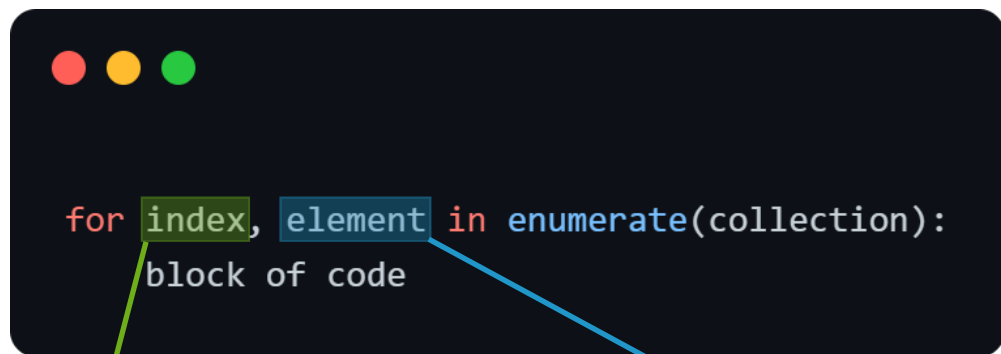
## Example



```
example_list = [1, 2, 3]  
a, b, c = example_list  
print(a, b, c)
```

## The enumerate() function

Used to retrieve an object containing both the element and its index inside the iterable.

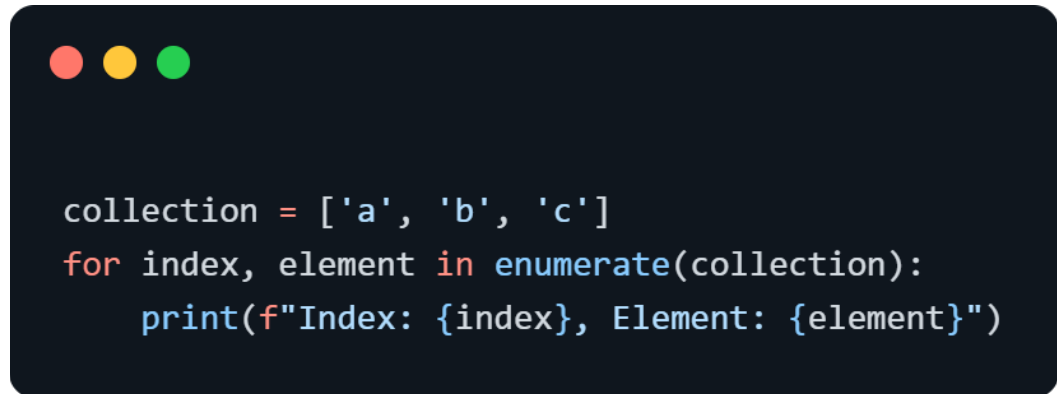


```
for index, element in enumerate(collection):  
    block of code
```

Index of an element from collection

The element from collection

## Example



```
collection = ['a', 'b', 'c']  
for index, element in enumerate(collection):  
    print(f"Index: {index}, Element: {element}")
```

## Output

```
Index: 0, Element: a  
Index: 1, Element: b  
Index: 2, Element: c
```

## The zip() function

Used to combine multiple iterables into a single one.

The new iterable is created by joining i-th element of each iterable until the end of the shortest one is reached.

The `zip()` function returns a `zip` object.

We'll talk more about this later in the course!

## Example

```
numbers = [1, 2, 3, 4, 5]
letters = ['a', 'b', 'c']
combined = list(zip(numbers, letters))
print(combined)
```

OUTPUT

[1, 2, 3, 4, 5] ['a', 'b', 'c']



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