Map

Map

- Takes the form

  map(function_to_apply, list_of_inputs)

- Instead of iteration

  items = [1, 2, 3, 4, 5]

  squared = []

  for i **in** items: squared.append(i**2)

  squared = list(map(**lambda** x: x**2, items))

In line lambda expression

```
: values = [1, 2, 3, 4, 5]
  mapped_values = map(lambda x: x + 10, values)
  print(list(mapped_values))
```

```
[11, 12, 13, 14, 15]
```

Another example – multiple arguments

```
: divide = lambda x, y : x / y
  values = [(8,2)]
  mapped = map(lambda x : divide(x[0], x[1]), values)
  print(list(mapped))

  [4.0]
```

Handling the exception

```python
divide = lambda x, y : x / y

values = [(8,0)]
mapped = map(lambda x : divide(x[0], x[1]), values)
print(list(mapped))
```

ZeroDivisionError   Traceback (

Use a def in this case

```python
def divide(x, y):
    try: return x/y
    except ZeroDivisionError: return 0

values = [(8,2)]
mapped = map(lambda x : divide(x[0], x[1]), values)
print(list(mapped))

values = [(8,0)]
mapped = map(lambda x : divide(x[0], x[1]), values)
print(list(mapped))
```

```
[4.0]
[0]
```