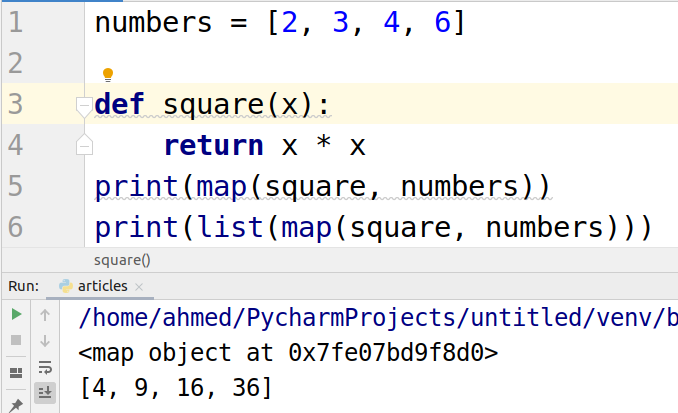
Python map() function with EXAMPLES

Sometimes you have a list of inputs that need to be passed to a single function. The map function helps us do this task. For example, there is a list of numbers like [ 2, 3, 4, 6] and you want to find the square of all the numbers. There are two approaches to this. One is you iterate over the list and pass the number one by one to the function and then save the output one by one in the result list. And the other way is to use the map function. Let’s take a look at code.

|  |
| --- |
| numbers = [2, 3, 4, 6]  def square(x):  return x \* x print(map(square, numbers)) print(list(map(square, numbers))) |

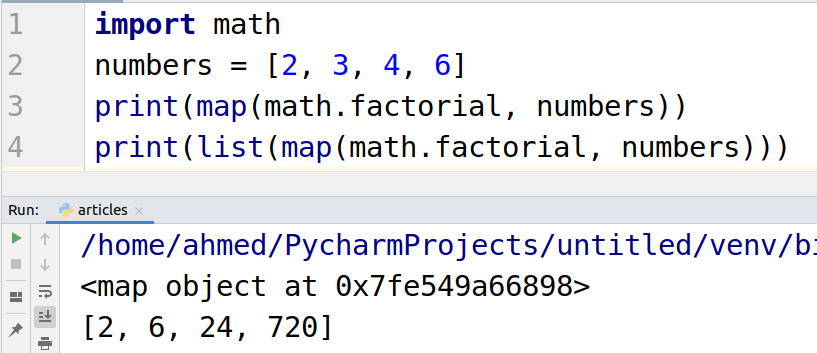
The following is the output.



### Using map function with built-in functions

In the above example, we created our own square functions. Now let’s take a look at how map function works with built-in functions. Let’s take the factorial of all the numbers in a list.

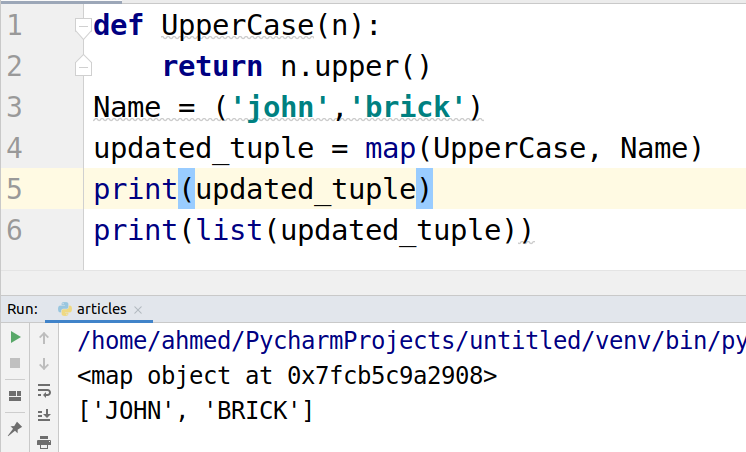
|  |
| --- |
| import math numbers = [2, 3, 4, 6] print(map(math.factorial, numbers)) print(list(map(math.factorial, numbers))) |



### Using map() with Tuple

Let us apply our map function on tuples.

|  |
| --- |
| def UpperCase(n):  return n.upper() Name = ('john','brick') updated\_tuple = map(UpperCase, Name) print(updated\_tuple) print(list(updated\_tuple)) |



### Using map() with Lambda function

Let’s use the lambda function inside the map. These programming practices make it really easy to code.

|  |
| --- |
| numbers = [2, 3, 4, 6] square = map(lambda x: x \* 2, numbers) print(square) print(list(square)) |

