

## Lambda's in Python

Lambda's are not barn animals, but are small anonymous functions that can be written in Python. They have any number of arguments and a single expression.

Here is lambda syntax:

```
lambda arguments : expression
```

You can create a number multiplier with a lambda as follows:

```
my_lamb = lambda x : x * 1000
```

To test this function you can call it with a print statement:

```
print(my_lamb(5))
```

The displayed result will be 5000.

Multiple arguments are applied as follows:

```
my_lamb = lambda x, y, z : x * y * z  
print(my_lamb(5, 5, 5))  
result: 125
```



[www.ScottMacri.com/vpn](http://www.ScottMacri.com/vpn)

Now you can recreate the one we built above with a more powerful technique.

```
def my_lamb(x):  
    return lambda y : y * x  
  
lamb_multiplier_1k = my_lamb(1000)  
print(lamb_multiplier_1k(5))
```

The answer again will be 5000. The parameters in the function definition (`my_lamb`) are used in the expression portion of the lambda. The lambda parameters (`y`) are used when we create a function from the lambda (ie - when we set a variable = to a call to the lambda function, then execute the function). The lambda is essentially an anonymous function within a function.

You could easily reuse the same function to do something like:

```
lamb_multiplier_25 = my_lamb(25)  
print(lamb_multiplier_25(4))  
result: 100
```

These are nice when you have a specific use case for them. Very convenient!!!