

PythonTip 01 - Functions (after-class-version)

January 26, 2026

1 Python Tip #1: Functions

Functions are separately defined code snippets that you can then use in your main code.

```
[1]: def double(number): # arguments = input
      new_number = 2*number
      return new_number
```

```
[2]: double(5)
```

```
[2]: 10
```

```
[3]: def add_up_list(L):
      total = 0
      for l in L:
          total += l
      return total
```

```
[4]: add_up_list([3,7,1])
```

```
[4]: 11
```

```
[ ]:
```

```
[5]: def print_hello():
      print("hello")
```

```
[6]: print_hello()
```

```
hello
```

```
[7]: x = print_hello()
```

```
hello
```

```
[8]: print(x)
```

```
None
```

```
[9]: def double(number=7): # number will default to 7 if you don't specify it
      new_number = 2*number
      return new_number
```

```
[10]: double()
```

```
[10]: 14
```

```
[11]: double(5)
```

```
[11]: 10
```

```
[12]: double(number=5)
```

```
[12]: 10
```

```
[ ]:
```

“Lambda Functions” sound very fancy, but they are just a quicker way to define very simple functions.

```
double = lambda x : 2*x
```

```
[name] = lambda [inputs] : [outputs]
```

```
[13]: new_double = lambda number : 2*number
      new_double(5)
```

```
[13]: 10
```

```
[14]: combine = lambda x, y: 2*x + 3 * y**2
```

```
[15]: combine(5,2)
```

```
[15]: 22
```

```
[ ]:
```

They are often useful (as we’ll see later) for extracting one component of a tuple or list.

```
[16]: second_component = lambda r : r[1]
```

```
[17]: second_component([5, -8, 1])
```

```
[17]: -8
```

This is totally equivalent to:

```
def second_component(r):
    return r[1]
```

This is mostly useful when you just want to use the function in one spot, and not define it forever.

When sorting a list, you can give it a “key” function to tell it what to sort by.

```
[18]: L = [-5, 1, 0, 7, -10]
      print(L)
      L.sort()
      print(L)
```

```
[-5, 1, 0, 7, -10]
[-10, -5, 0, 1, 7]
```

```
[20]: L.sort(key=lambda x : abs(x))
      print(L)
```

```
[0, 1, -5, 7, -10]
```

```
[ ]:
```

```
[ ]:
```

```
[23]: L = [(0, 3), (-1, 7), (0, 1), (2, 5)]
```

```
[24]: sorted(L)
```

```
[24]: [(-1, 7), (0, 1), (0, 3), (2, 5)]
```

```
[26]: second_component = lambda x : x[1]
```

```
[27]: sorted(L, key=second_component)
```

```
[27]: [(0, 1), (0, 3), (2, 5), (-1, 7)]
```

```
[28]: sorted(L, key=lambda x : x[1])
```

```
[28]: [(0, 1), (0, 3), (2, 5), (-1, 7)]
```

```
[ ]:
```