Functions

- Define a function with the **def** keyword, then a name, then parameters in brackets
- Don’t forget to put a **:** after the parameters
- Then you must indent with the **TAB** key for the function body
- Return values using the **return** keyword

```
def add_numbers(num_a, num_b):
    return num_a + num_b

result = add_numbers(2, 3)
print(result)
```

- Here we’re calling our function to add 2 and 3 together, then printing the result. Give it a try, and make some changes.

Comments, Variables and Scope

```
# everything after # is a comment

x = 1

def foo():
    y = 2

print(x) # this works
print(y) # this doesn't
```

```
def foo():
    global y
    y = 2

print(y) # this works now
```

- start comments with a **#**
- assigning a variable in Python is easy, just type the name, an =, and its value
- variables outside of functions have global scope (they can be used anywhere)
- variables inside functions have local scope (they can only be used inside the function)
- you can override this by defining a variable in a function with the **global** keyword (but don’t worry about this for now)