



Arguments

An argument is a value we pass into the function as its input when we call the function

We use arguments so we can direct the function to do different kinds of work when we call it at different times

We put the arguments in parentheses after the name of the function

```
big = max('Hello world')
```



Argument



Parameters

A parameter is a variable which we use in the function definition. It is a “handle” that allows the code in the function to access the arguments for a particular function invocation.

```
>>> def greet(lang):  
...     if lang == 'es':  
...         print('Hola')  
...     elif lang == 'fr':  
...         print('Bonjour')  
...     else:  
...         print('Hello')  
...  
>>> greet('en')  
Hello  
>>> greet('es')  
Hola  
>>> greet('fr')  
Bonjour  
>>>
```



Return Values

Often a function will take its arguments, do some computation, and return a value to be used as the value of the function call in the calling expression. The return keyword is used for this.

```
def greet():  
    return "Hello"
```

```
print(greet(), "Glenn")  
print(greet(), "Sally")
```

Hello Glenn
Hello Sally



Return Value

A “fruitful” function is one that produces a result (or return value)

The return statement ends the function execution and “sends back” the result of the function

```
>>> def greet(lang):
...     if lang == 'es':
...         return 'Hola'
...     elif lang == 'fr':
...         return 'Bonjour'
...     else:
...         return 'Hello'
...
>>> print(greet('en'),'Glenn')
Hello Glenn
>>> print(greet('es'),'Sally')
Hola Sally
>>> print(greet('fr'),'Michael')
Bonjour Michael
>>>
```



Arguments, Parameters, and Results

```
>>> big = max('Hello world')  
>>> print(big)  
w
```

Argument → 'Hello world'

```
def max(inp):  
    blah  
    blah  
    for x in inp:  
        blah  
        blah  
    return 'w'
```

Parameter →

'w'
↑
Result



Multiple Parameters / Arguments

We can define more than one parameter in the function definition

```
def addtwo(a, b):  
    added = a + b  
    return added
```

We simply add more arguments when we call the function

```
x = addtwo(3, 5)  
print(x)
```

We match the number and order of arguments and parameters

```
8
```



Void (non-fruitful) Functions

When a function does not return a value, we call it a “void” function

Functions that return values are “fruitful” functions

Void functions are “not fruitful”

To function or not to function



Organize your code into “paragraphs” - capture a complete thought and “name it”

Don’t repeat yourself - make it work once and then reuse it

If something gets too long or complex, break it up into logical chunks and put those chunks in functions

Make a library of common stuff that you do over and over - perhaps share this with your friends...