

Arguments

Function Arguments

You can include as many arguments as you want in your function definition. Inside the function, these arguments can be treated as variables.

```
[21]: def arg3(arg1, arg2, arg3):  
      print(arg1)  
      print(arg2)  
      return arg3
```

```
[22]: arg3(1, 2, 3)
```

```
1  
2
```

```
[22]: 3
```

You may use variables or statements (anything that resolves to a value or object) as arguments:

```
[24]: var1 = 45  
  
      arg3(var1, 3*4, 7)
```

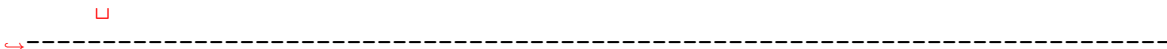
```
45  
12
```

```
[24]: 7
```

Positional Arguments

The arguments defined above are called **positional arguments**. In order to set them correctly, you need to parse them in the order they are defined in the function. You must also provide a value for each argument:

```
[23]: arg3('a', 'b')
```



```
TypeError                                Traceback (most recent call_
↳last)
```

```
<ipython-input-23-964cf24ab02e> in <module>
----> 1 arg3('a', 'b')
```

```
TypeError: arg3() missing 1 required positional argument: 'arg3'
```

Keyword Arguments

If you want to set optional arguments with a default value, you can use keyword arguments. The syntax is:

```
def function_name(keyword_arg = default_value):
```

For example:

```
[1]: def hello(name = 'World', time = 'today'):
      return f'Hello {name}! How are you {time}?'
```

(If you are unfamiliar with f-strings `f''`, see the section [String Formatting](#))

This function can be called with no arguments, in which case the default values will be used:

```
[3]: hello()
```

```
[3]: 'Hello World! How are you today?'
```

We can also parse the arguments like positional arguments:

```
[4]: hello('reader', 'feeling')
```

```
[4]: 'Hello reader! How are you feeling?'
```

```
[5]: hello('reader')
```

```
[5]: 'Hello reader! How are you today?'
```

Keyword arguments can be referred to by name, and out of order:

```
[6]: hello(time = 'this morning')
```

```
[6]: 'Hello World! How are you this morning?'
```

Combining Positional and Keyword Arguments

If you define a function with both positional and keyword arguments, the positional arguments must appear **before** the keyword arguments.

For example:

```
[4]: def hello_hello(num, name = 'World', time = 'today', weather = 'good'):

      return f"Hello {num*'hello'} {name}! How are you {time}?"
```

Here the positional argument `num` must be provided

```
[3]: hello_hello()
```

```

      □
↪ -----

      TypeError                                Traceback (most recent call□
↪ last)

      <ipython-input-3-260a0f5fbf0a> in <module>
      ----> 1 hello_hello()

      TypeError: hello_hello() missing 1 required positional argument: 'num'
```

As before, the keyword arguments can be provided like positional arguments or by name:

```
[5]: hello_hello(1, 'there', 'doing')
```

```
[5]: 'Hello hello there! How are you doing?'
```

```
[6]: hello_hello(0, time = 'awake this early', name = 'sleepy head')
```

```
[6]: 'Hello  sleepy head! How are you awake this early?'
```