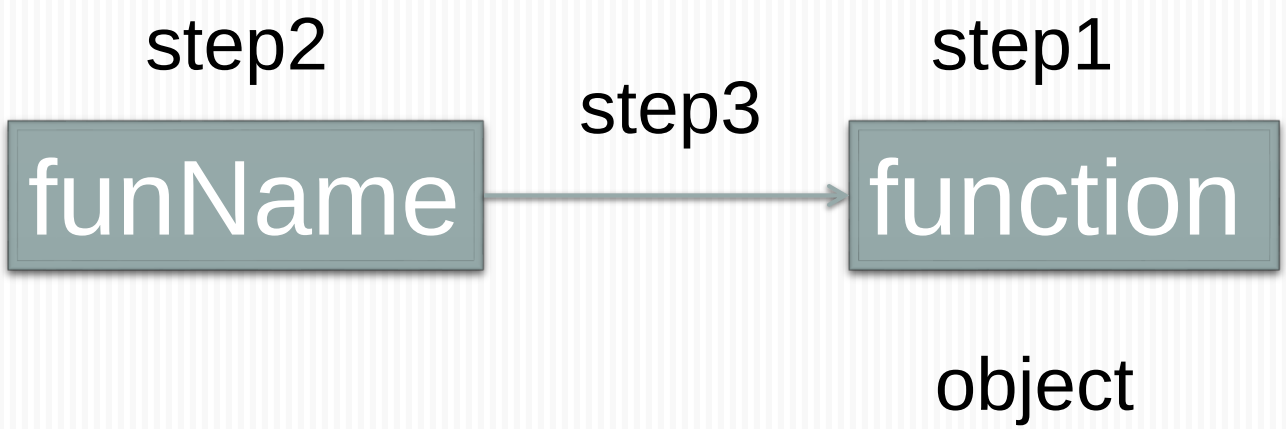
PYTHON FUNCTIONS

Lecturer: Terry

1 FUNCTION BASICS

```
def <funName>(arg1, arg2, ..., argN):
    <statements>
return <expression>
```



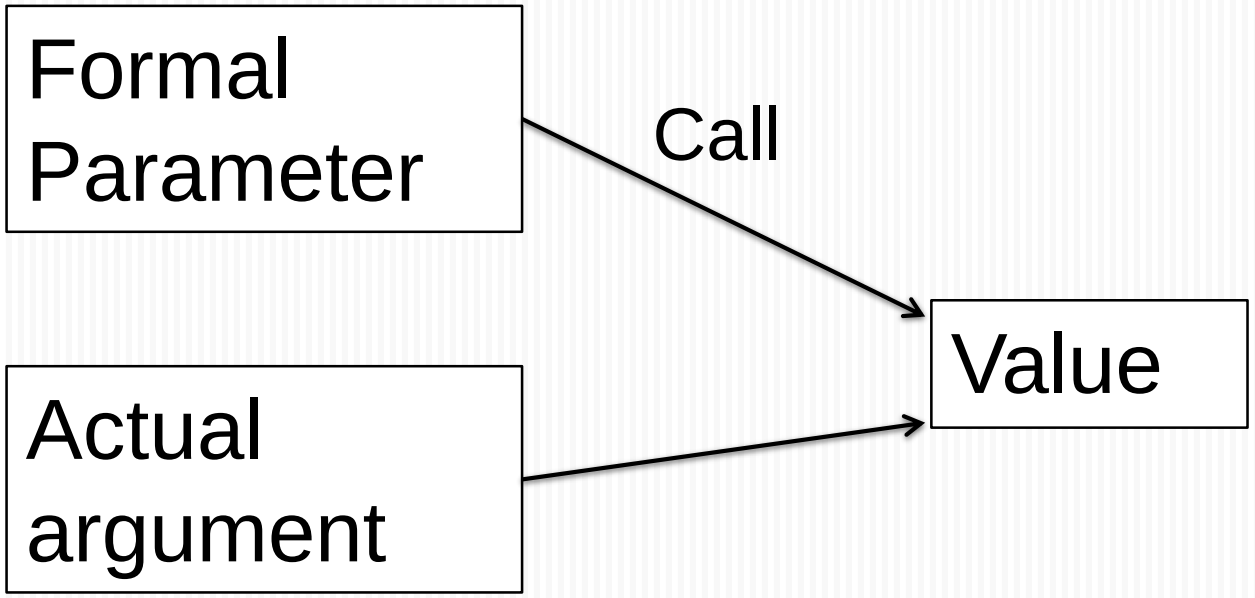
Syntax

Simple functions:

- If a function takes no arguments, it must still include the parentheses, but without anything in them.

Functions without arguments, statements, or return values.

Define



Define and call a function with arguments, statements and return value.

2 THE ARGUMENTS

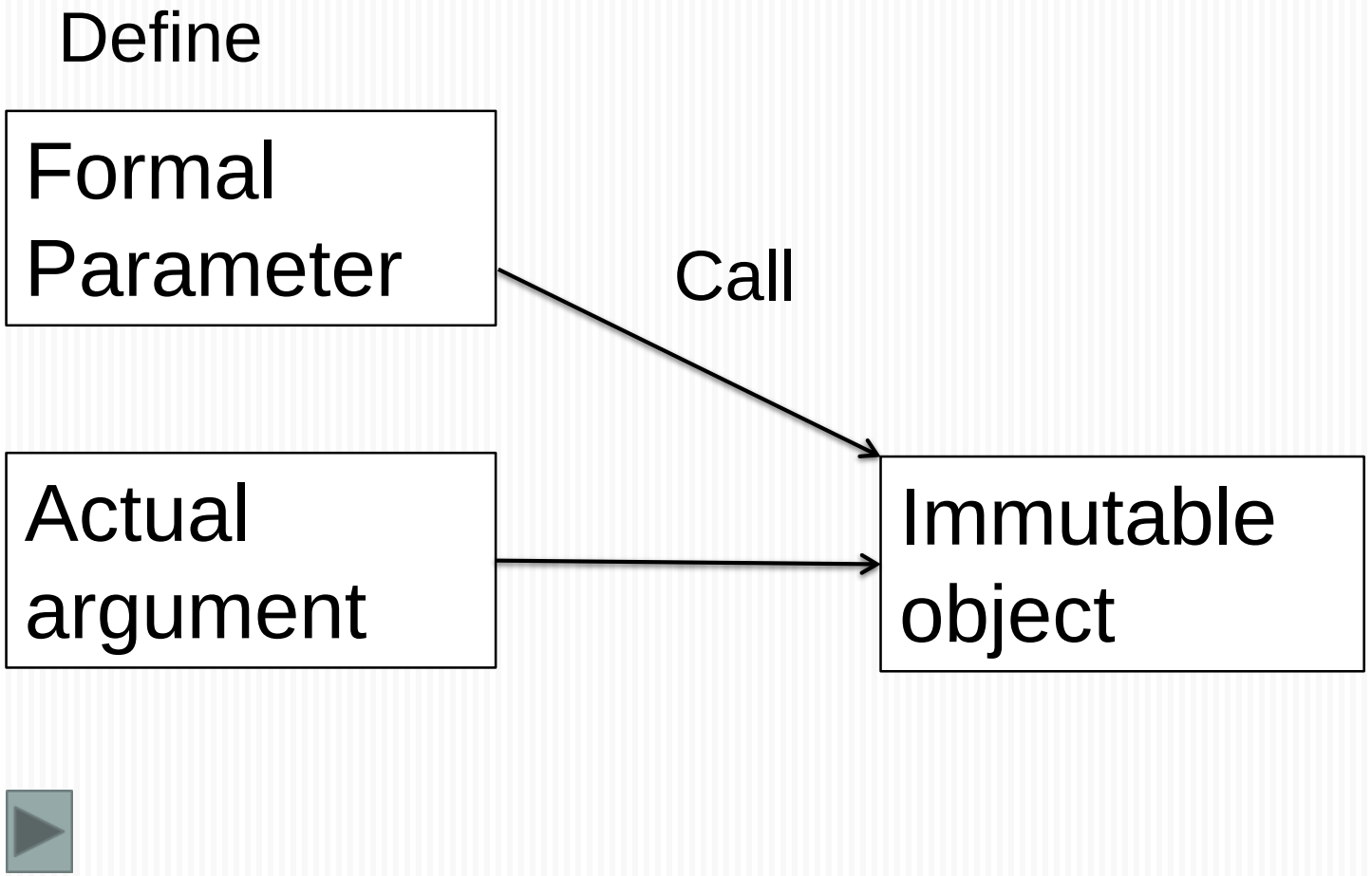
```
>>> def my_subtraction(minuend, subtrahend):
...     difference = minuend - subtrahend
...     return difference

>>> my_subtraction(7,2)
5

>>> my_subtraction(2,7)
-5
```

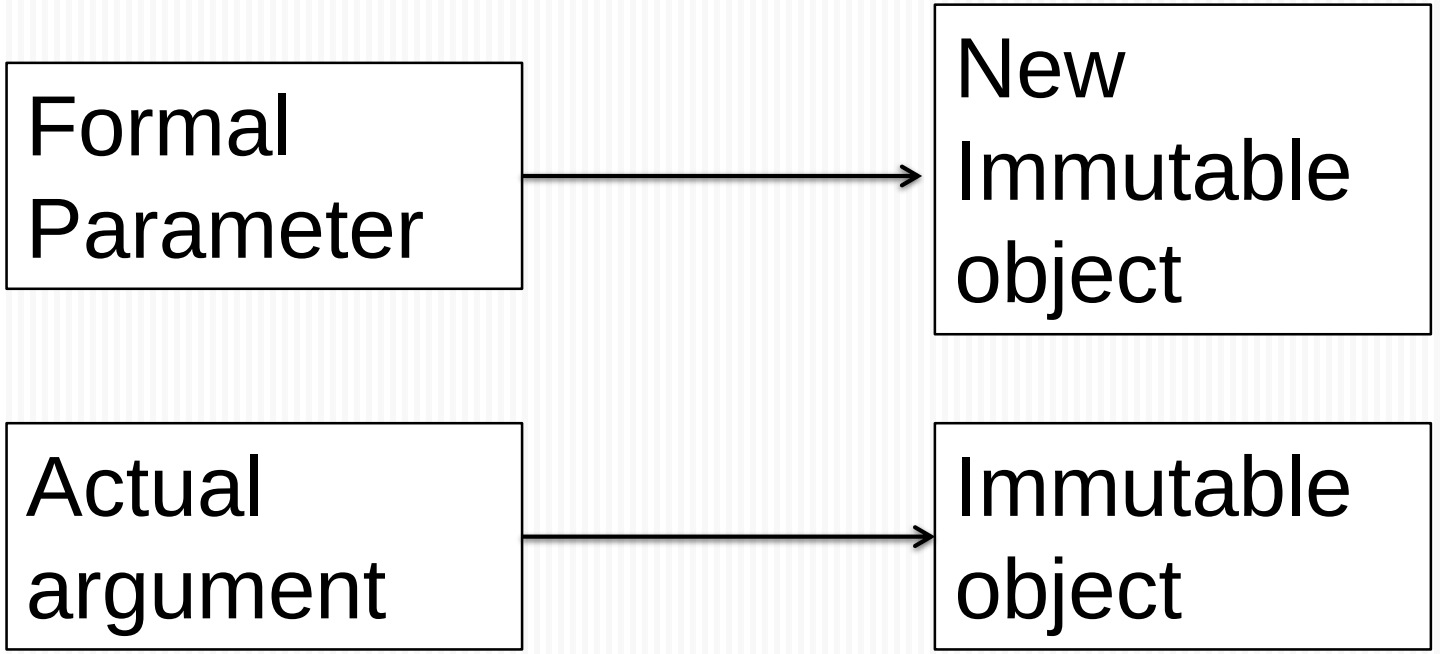
Positional Argument

```
>>> my_subtraction(minuend = 7,
subtrahend = 2)
5

>>> my_subtraction(subtrahend = 2,
minuend = 7)
5

>>> my_subtraction(subtrahend = 2, 7)
```

Keyword Argument

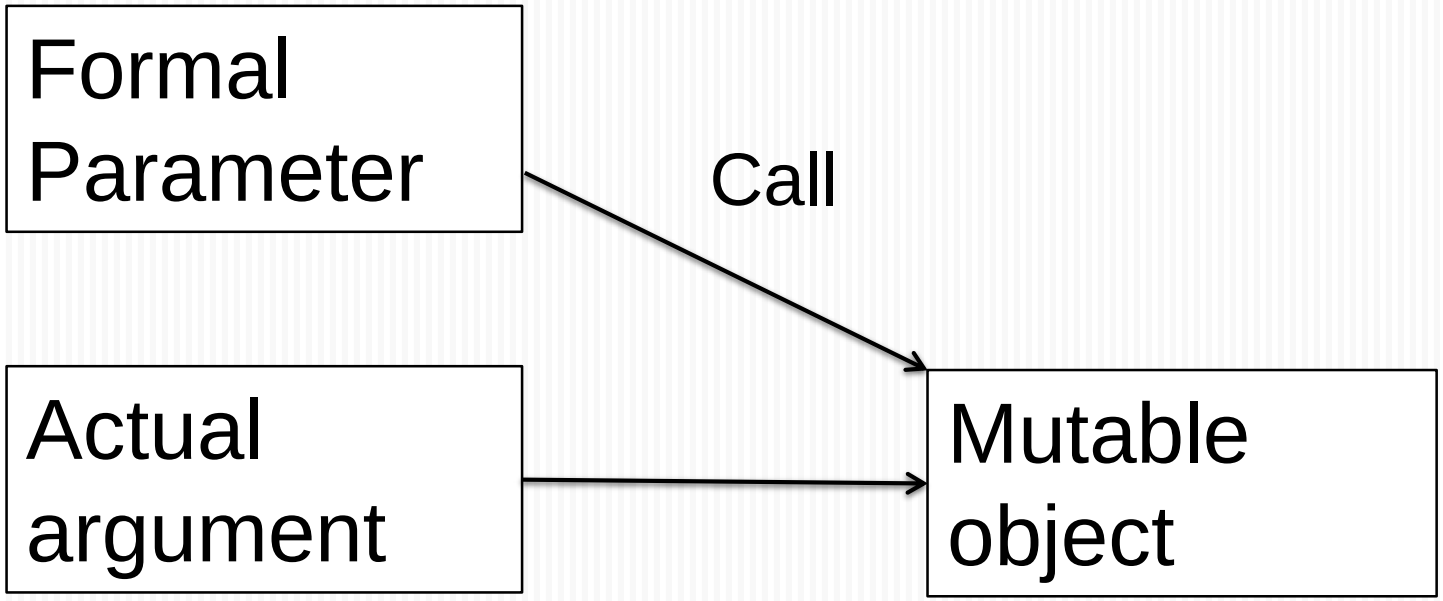
Immutable Arguments



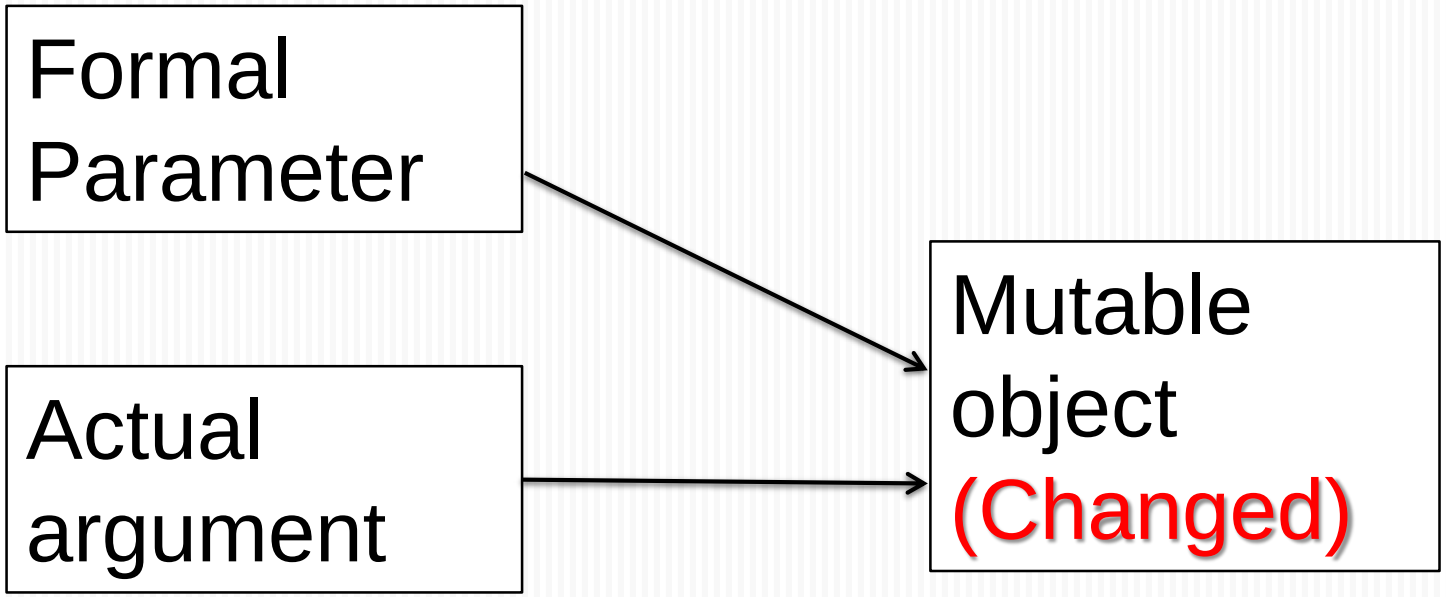


Immutable Arguments

Define



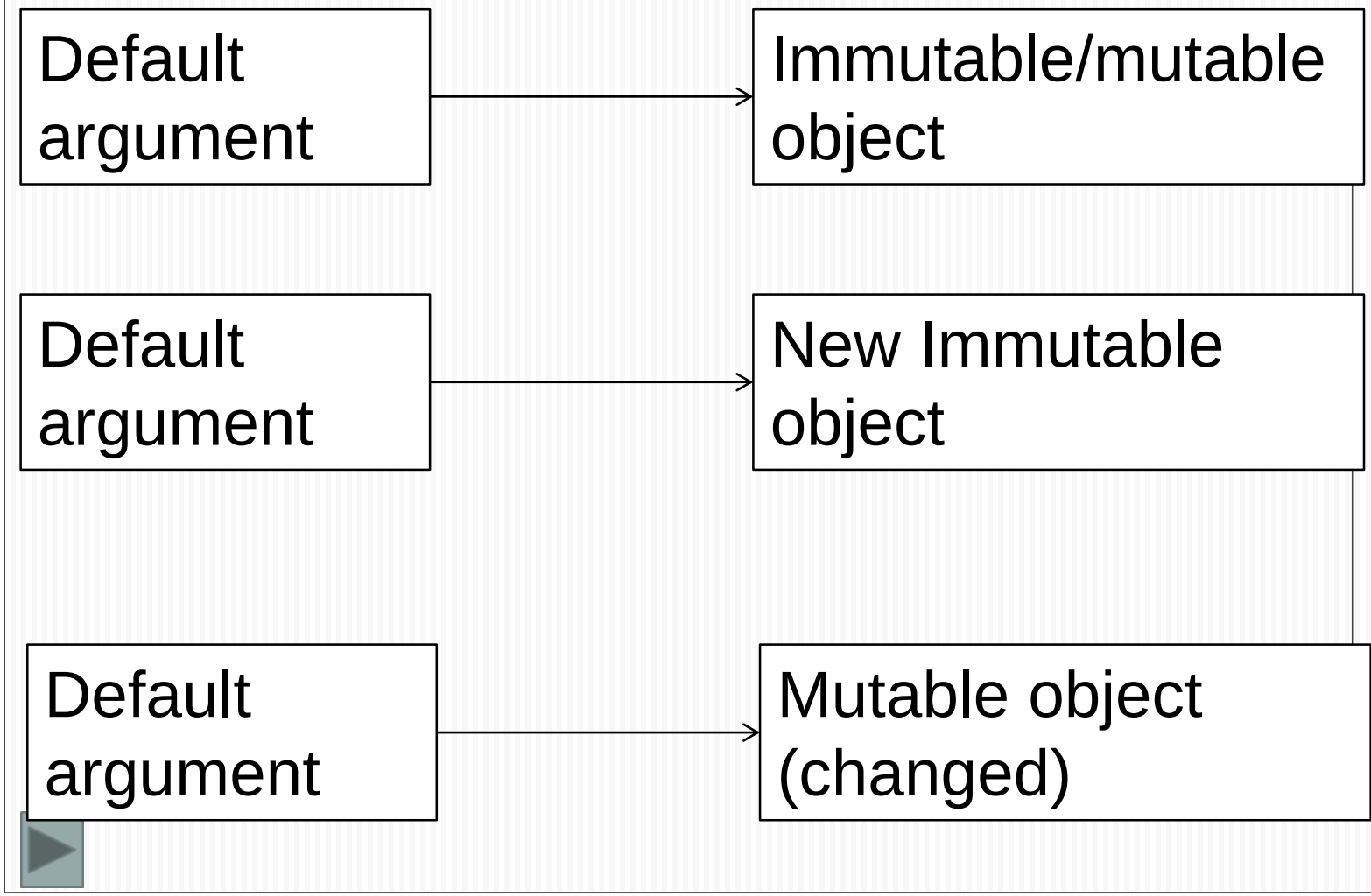
Mutable Arguments



Mutable Arguments

```
>>> def my_print(arg1, arg2 = 'World!'):
...     print(arg1, arg2)

>>> my_print('Hi')
Hi World!
```

Default Arguments



Default Immutable / Mutable Arguments

If we are not sure about the number of arguments passed to the function, what should we do?

*args: a non-keyworded variable length argument list
**kwargs: keyworded variable length of arguments (dict)

Variable Arguments

3 Anonymous Functions

lambda arg1, arg2, ... argN: expression

- A lambda function is a lightweight anonymous function.
  - any number of arguments
  - only a single expression.
  - (not statements)

Syntax