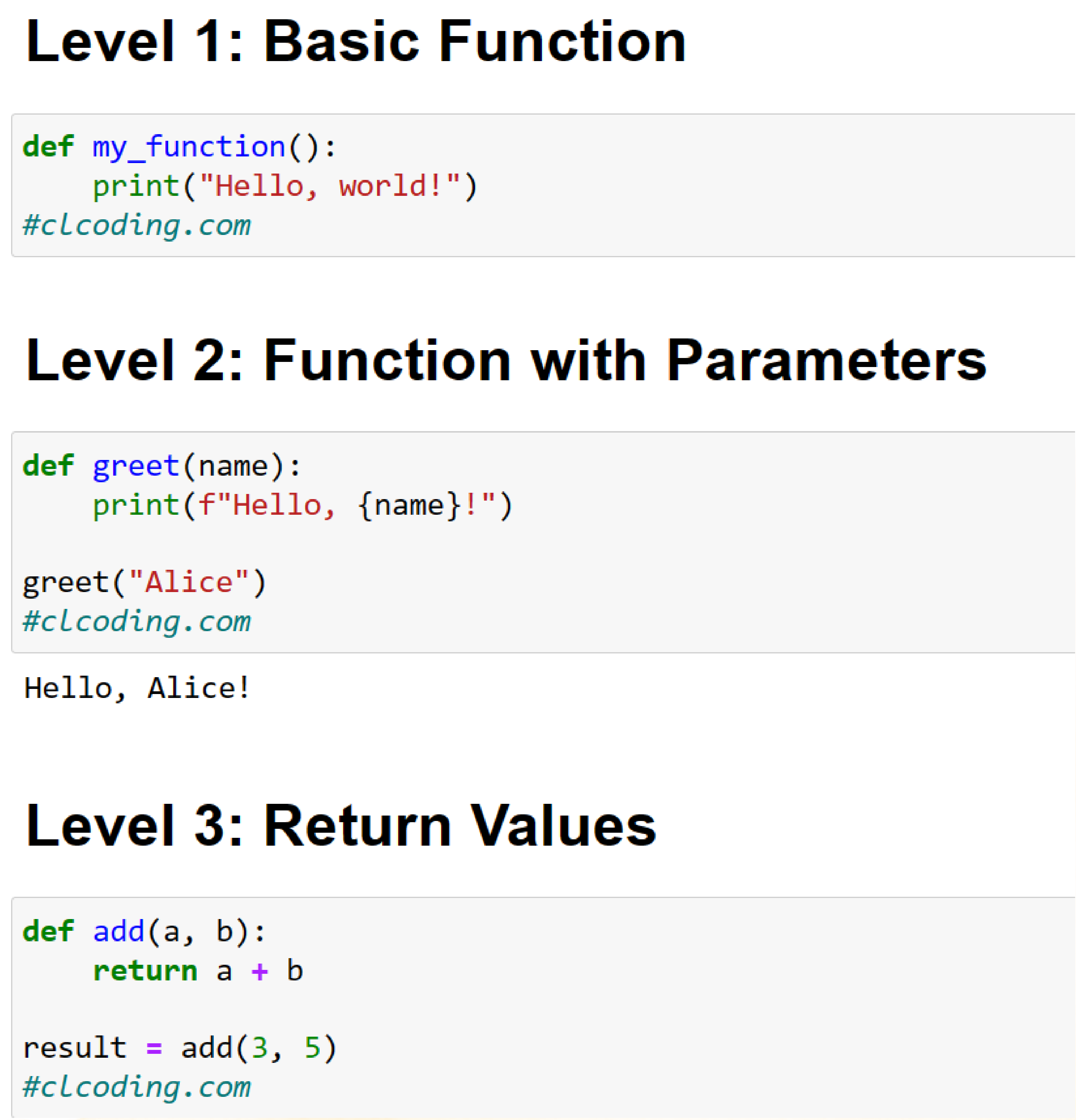
10 Levels of Writing Python Functions



Level 1: Basic Function

```
def my_function():
    print("Hello, world!")
#clcoding.com
```

Level 2: Function with Parameters

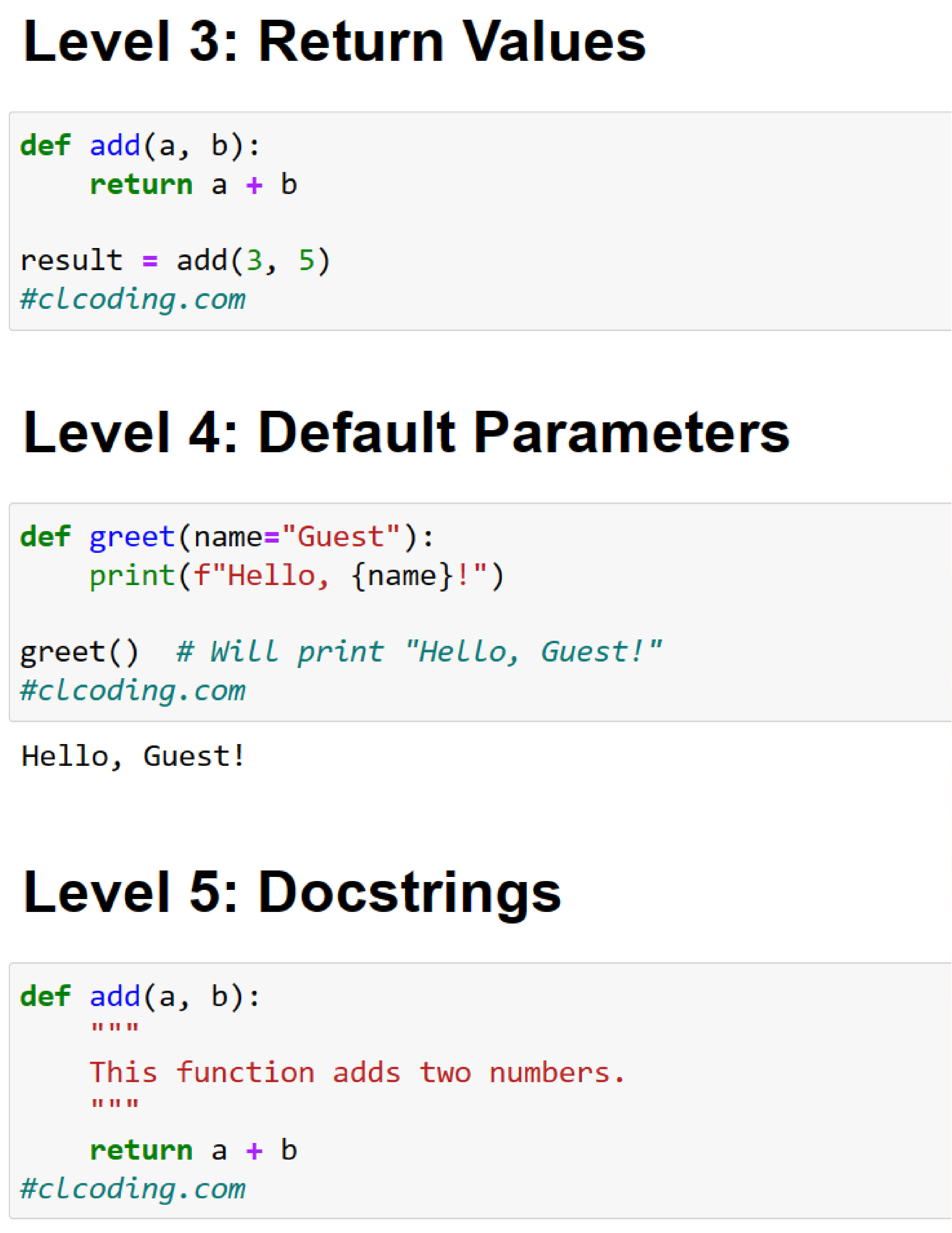
```
def greet(name):
    print(f"Hello, {name}!")

greet("Alice")
#clcoding.com
```

```
Hello, Alice!
```

Level 3: Return Values

```
def add(a, b):
    return a + b

result = add(3, 5)
#clcoding.com
```

10 Levels of Writing Python Functions



Level 3: Return Values

```python
def add(a, b):
    return a + b

result = add(3, 5)
#clcoding.com
```

Level 4: Default Parameters

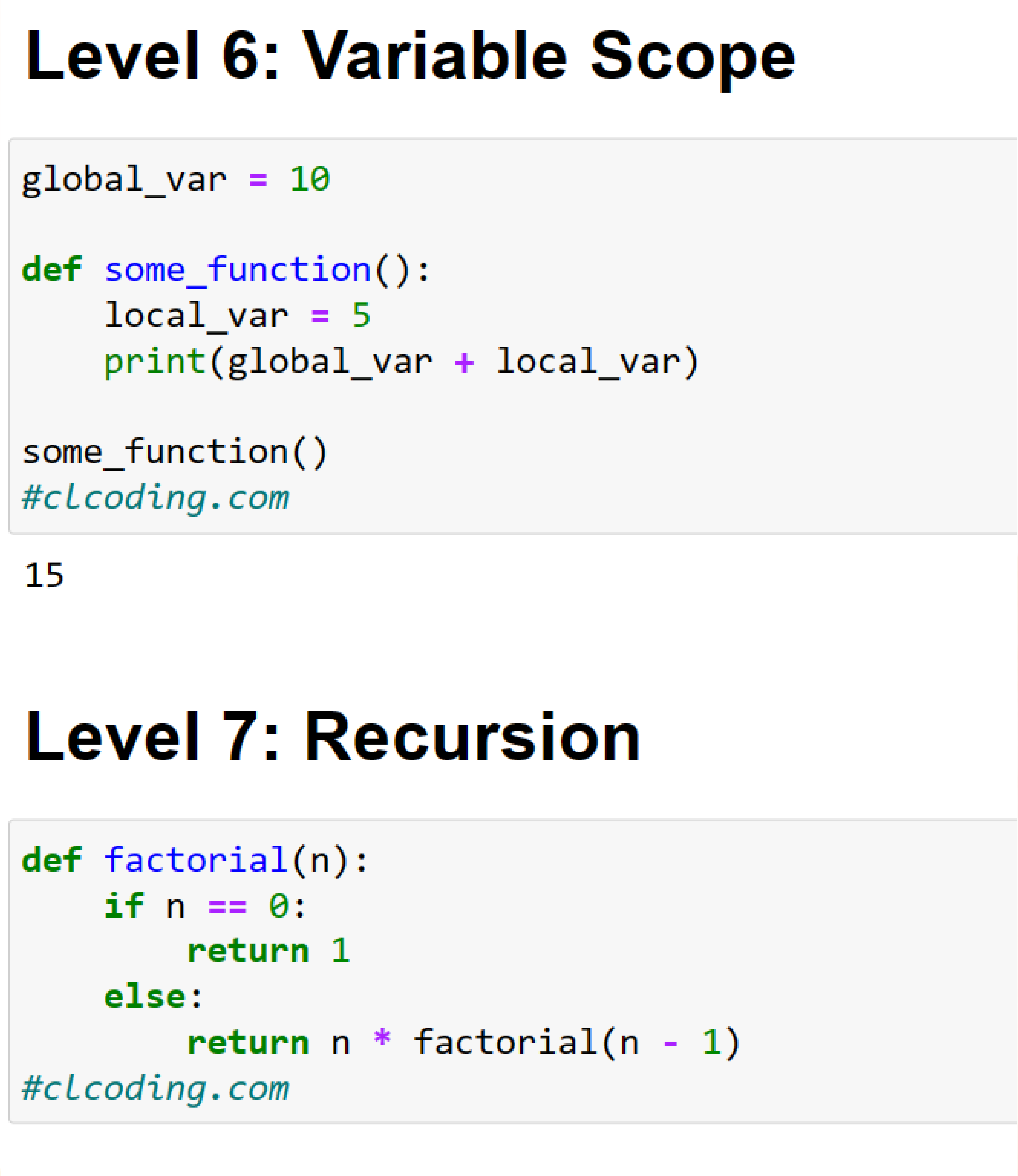
```python
def greet(name="Guest"):
    print(f"Hello, {name}!")

greet()  # Will print "Hello, Guest!"
#clcoding.com
```

```
Hello, Guest!
```

Level 5: Docstrings

```python
def add(a, b):
    """
    This function adds two numbers.
    """
    return a + b
#clcoding.com
```

10 Levels of Writing Python Functions



Level 6: Variable Scope

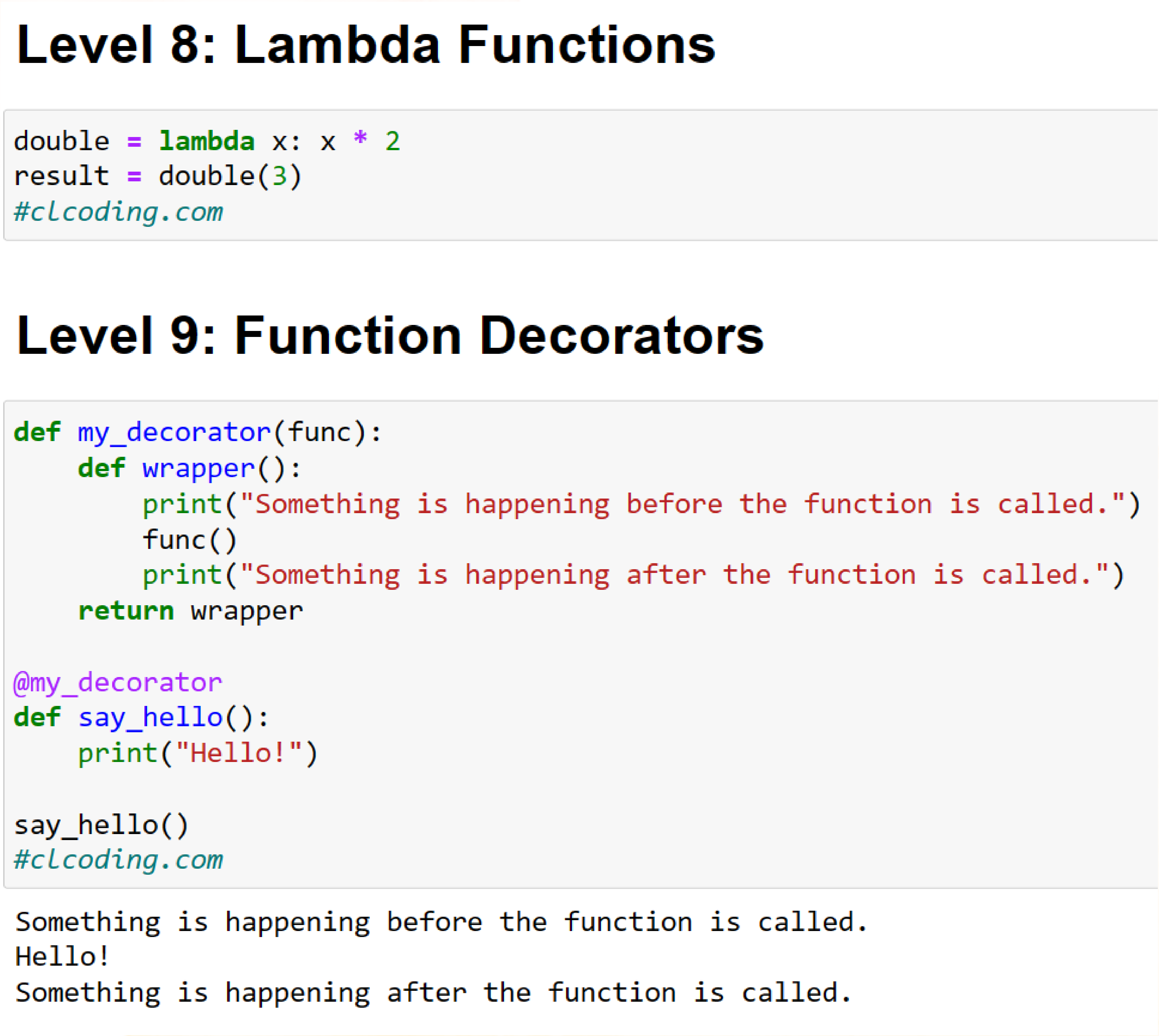
```python
global_var = 10

def some_function():
    local_var = 5
    print(global_var + local_var)

some_function()
#clcoding.com
```

```
15
```

Level 7: Recursion

```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
#clcoding.com
```

/Pythonclcoding /Pythoncoding /clcoding



Level 8: Lambda Functions

```
double = lambda x: x * 2
result = double(3)
#clcoding.com
```

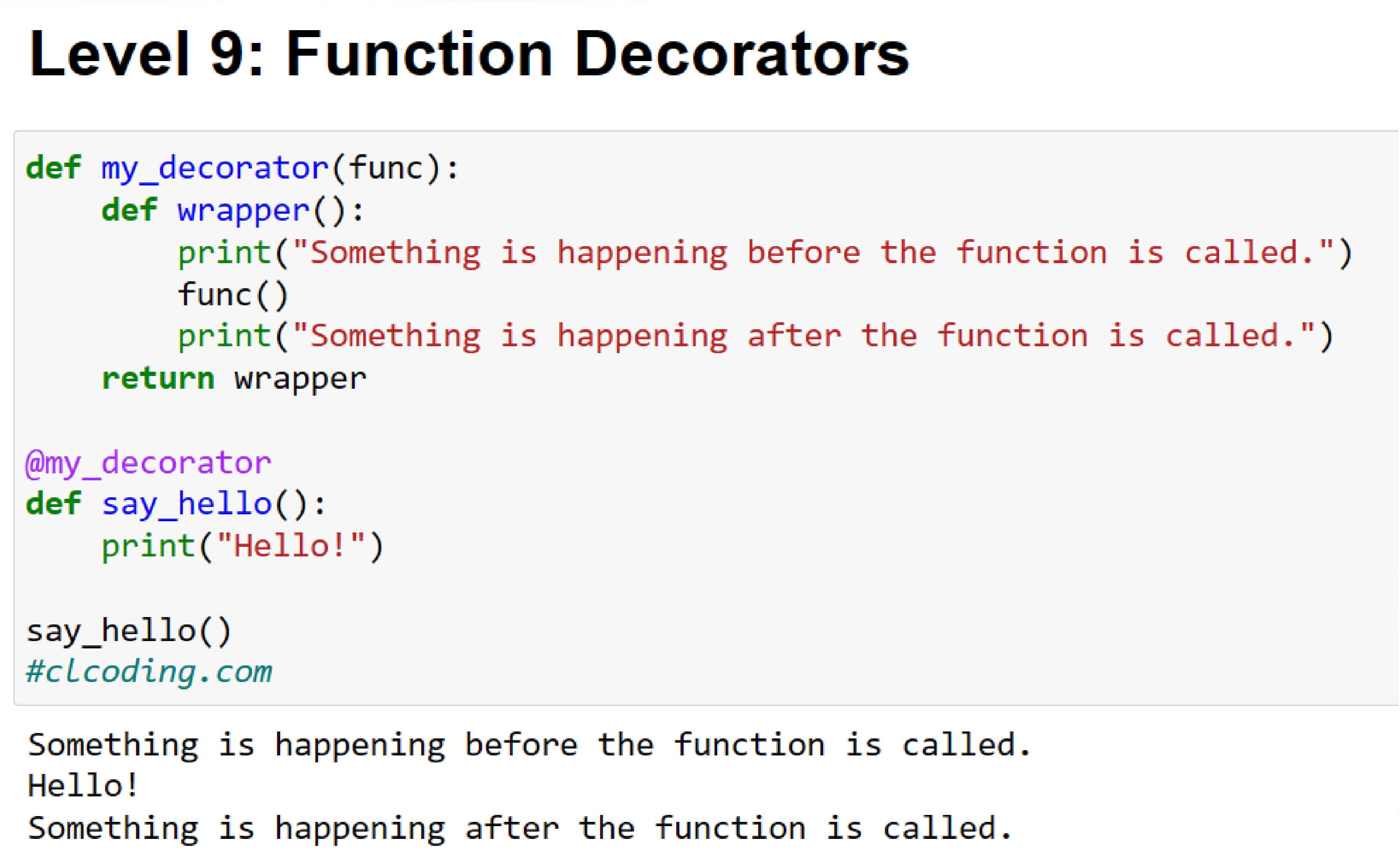
Level 9: Function Decorators

```
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")

say_hello()
#clcoding.com
```

```
Something is happening before the function is called.
Hello!
Something is happening after the function is called.
```

/Pythonclcoding   /Pythoncoding   /clcoding

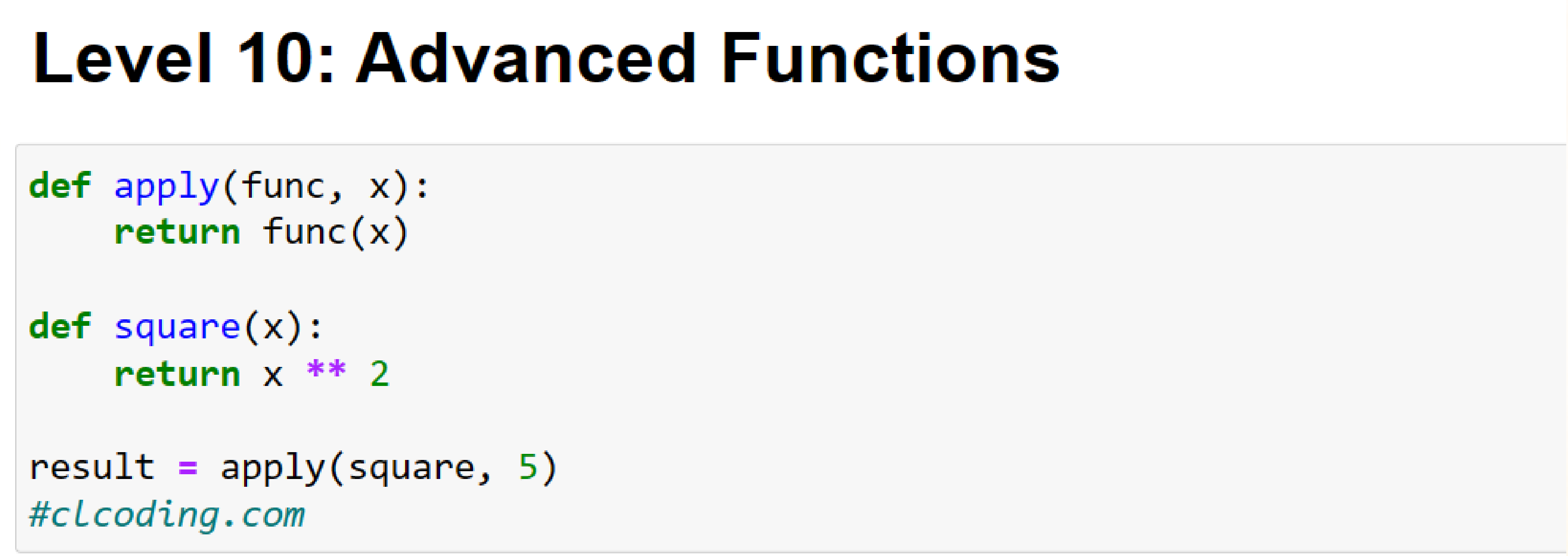
10 Levels of Writing Python Functions



Level 9: Function Decorators

```python
def my_decorator(func):
    def wrapper():
        print("Something is happening before the function is called.")
        func()
        print("Something is happening after the function is called.")
    return wrapper

@my_decorator
def say_hello():
    print("Hello!")

say_hello()
#clcoding.com
```

```
Something is happening before the function is called.
Hello!
Something is happening after the function is called.
```



Level 10: Advanced Functions

```python
def apply(func, x):
    return func(x)

def square(x):
    return x ** 2

result = apply(square, 5)
#clcoding.com
```