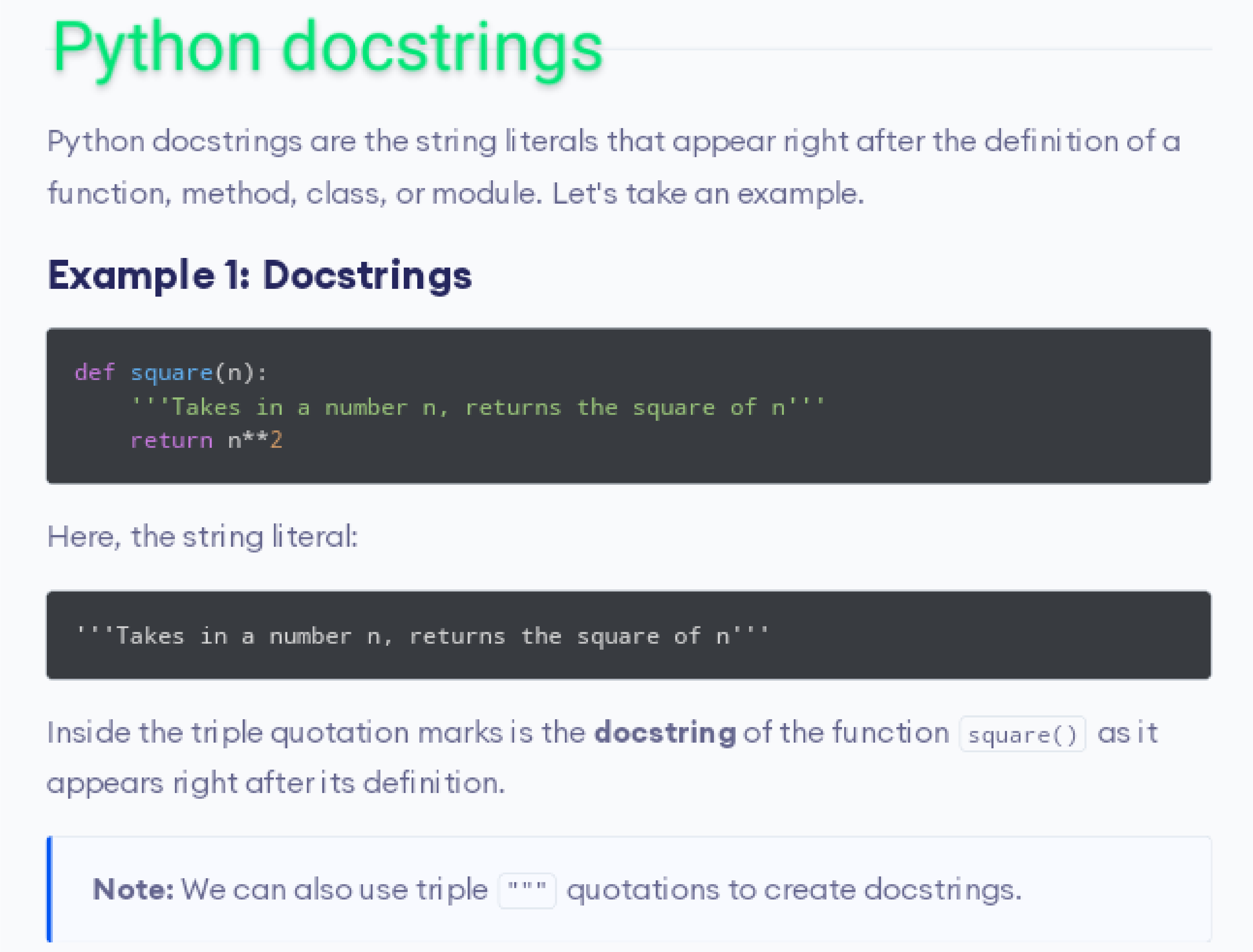
# Python

Python
Docstring



# Python docstrings

Python docstrings are the string literals that appear right after the definition of a function, method, class, or module. Let's take an example.

## Example 1: Docstrings

```
def square(n):
    '''Takes in a number n, returns the square of n'''
    return n**2
```

Here, the string literal:

```
'''Takes in a number n, returns the square of n'''
```

Inside the triple quotation marks is the **docstring** of the function `square()` as it appears right after its definition.

> **Note:** We can also use triple `"""` quotations to create docstrings.

## Python Comments vs Docstrings

### Python Comments

Comments are descriptions that help programmers better understand the intent and functionality of the program. They are completely ignored by the Python interpreter.

In Python, we use the hash symbol `#` to write a single-line comment. For example,

```
# Program to print "Hello World"
print("Hello World")
```

### Python Comments Using Strings

If we do not assign strings to any variable, they act as comments. For example,

```
"I am a single-line comment"

'''
I am a
multi-line comment!
'''

print("Hello World")
```

> **Note:** We use triple quotation marks for multi-line strings.

**Python docstrings**

As mentioned above, Python docstrings are strings used right after the definition of a function, method, class, or module (like in **Example 1**). They are used to document our code.

We can access these docstrings using the `__doc__` attribute.

## Python __doc__ attribute

Whenever string literals are present just after the definition of a function, module, class or method, they are associated with the object as their `__doc__` attribute. We can later use this attribute to retrieve this docstring.

### Example 2: Printing docstring

```python
def square(n):
    '''Takes in a number n, returns the square of n'''
    return n**2

print(square.__doc__)
```

**Output**

```
Takes in a number n, returns the square of n
```

Here, the documentation of our `square()` function can be accessed using the `__doc__` attribute.