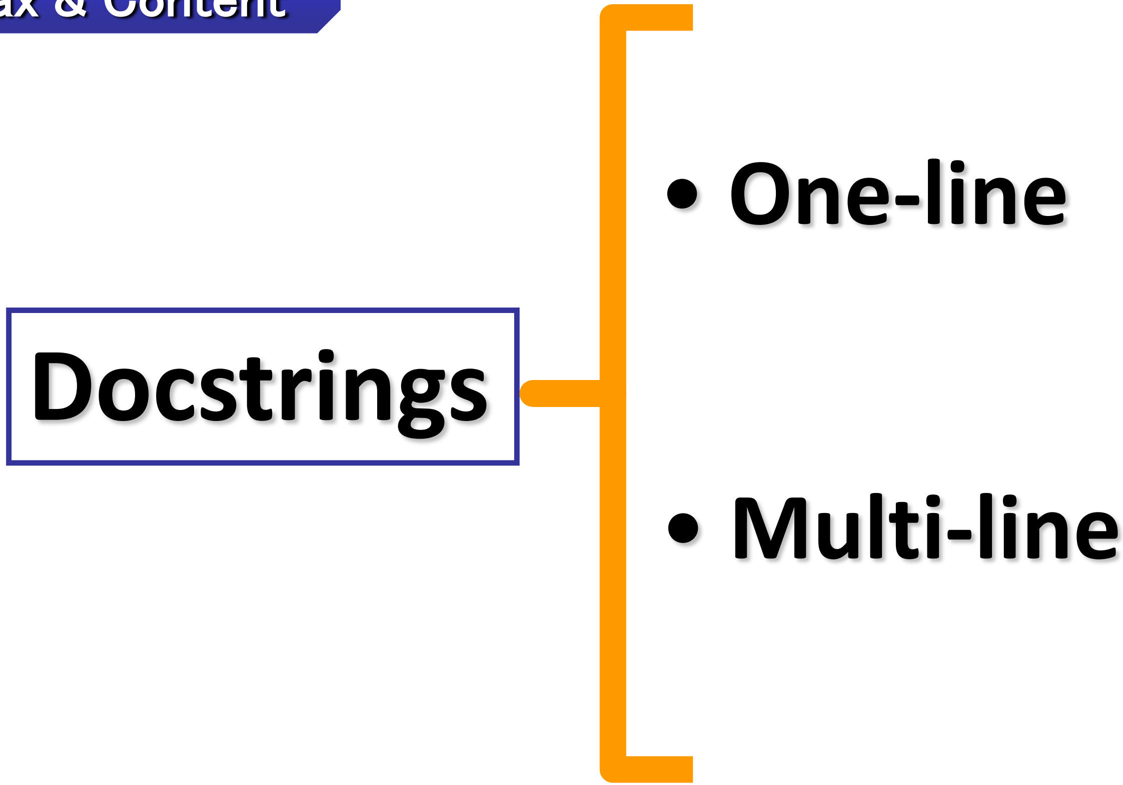
Lecture

Docstrings

Syntax & Content

Syntax & Content



Docstrings

- One-line
- Multi-line

Syntax & Content

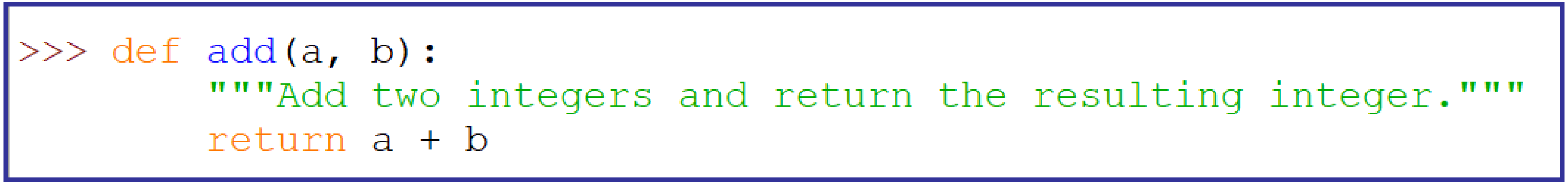
One-line Docstrings

Syntax & Content

One-line Docstrings

Methods or Functions

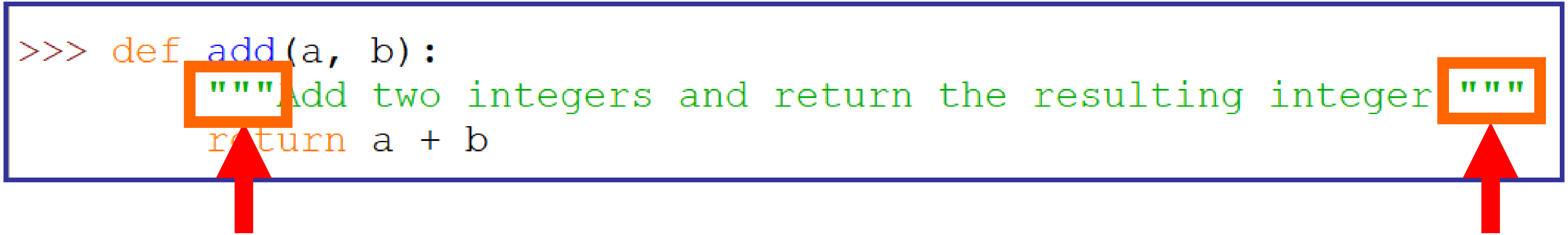
One-line Docstrings



```
>>> def add(a, b):
        """Add two integers and return the resulting integer."""
        return a + b
```

One-line Docstrings

- Triple quotes are used even though the string fits on one line. This makes it easy to later expand it.

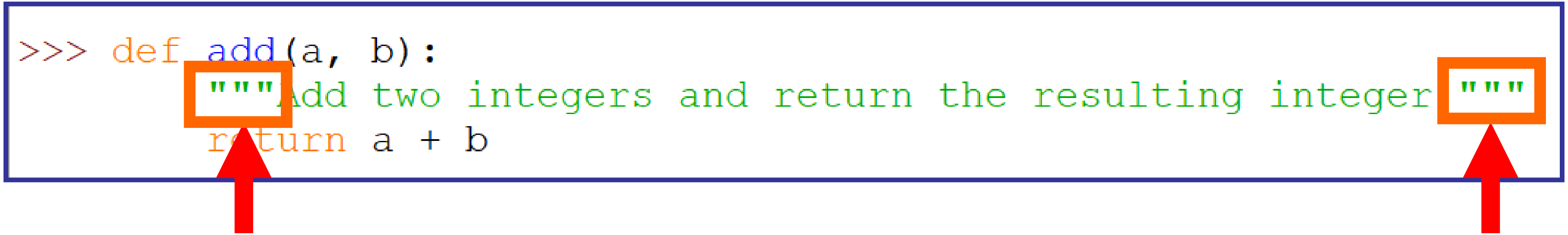


```
>>> def add(a, b):
        """Add two integers and return the resulting integer"""
        return a + b
```

Docs

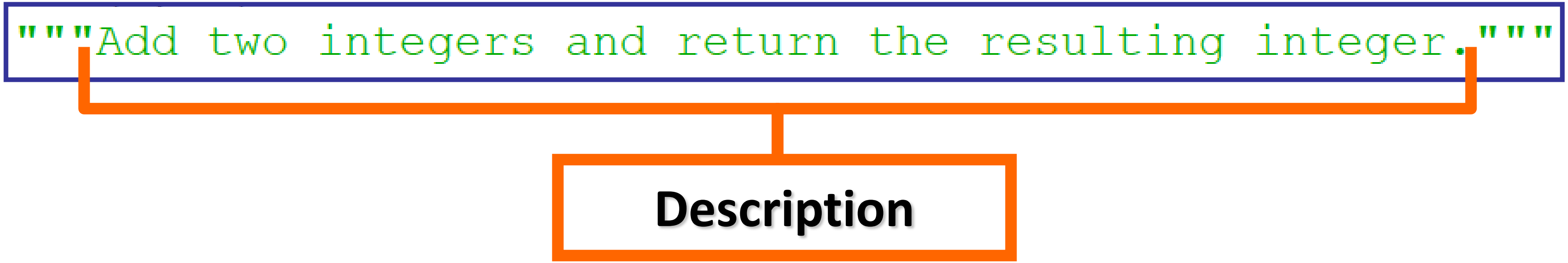
One-line Docstrings

- The closing quotes are on the same line as the opening quotes. This looks better for one-liners.



```
>>> def add(a, b):
        """Add two integers and return the resulting integer"""
        return a + b
```

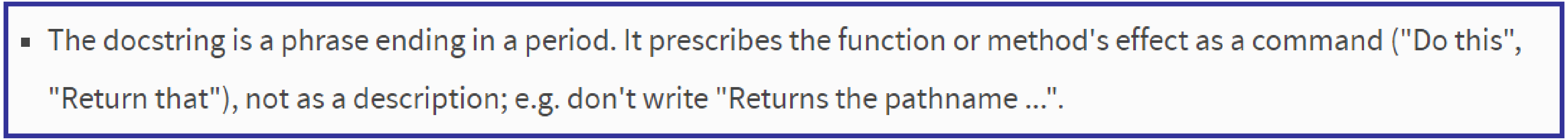
One-line Docstrings



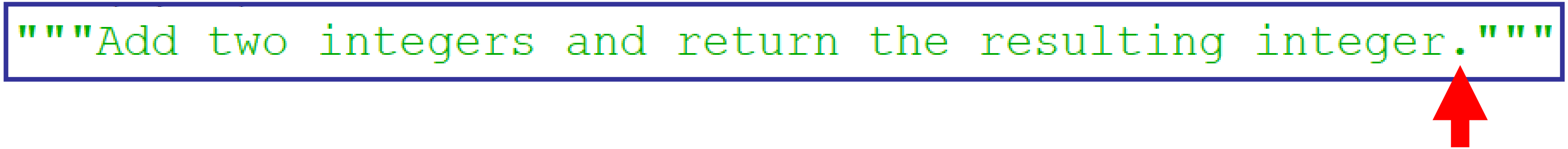
```
"""Add two integers and return the resulting integer."""
```

Description

One–line Docstrings



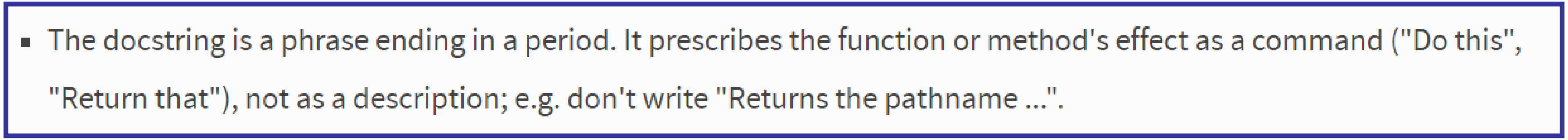
- The docstring is a phrase ending in a period. It prescribes the function or method's effect as a command ("Do this", "Return that"), not as a description; e.g. don't write "Returns the pathname ...".



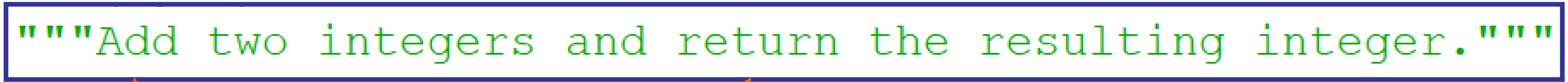
```
"""Add two integers and return the resulting integer."""
```

Docs

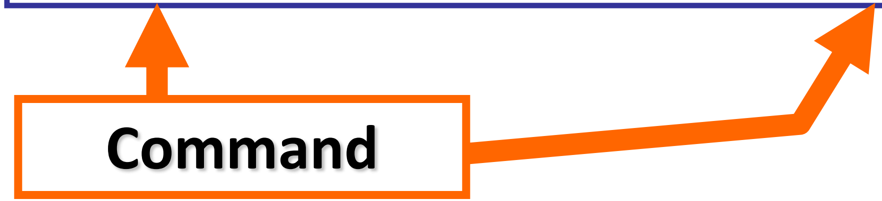
One-line Docstrings



- The docstring is a phrase ending in a period. It prescribes the function or method's effect as a command ("Do this", "Return that"), not as a description; e.g. don't write "Returns the pathname ...".



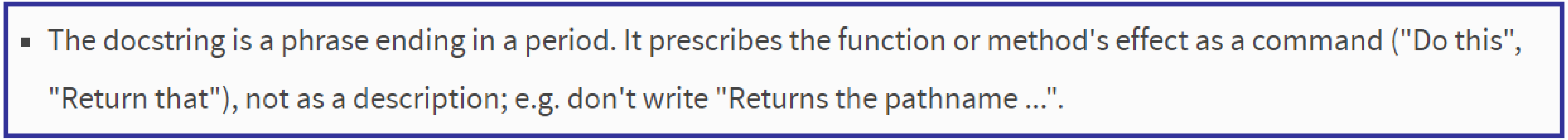
```
"""Add two integers and return the resulting integer."""
```



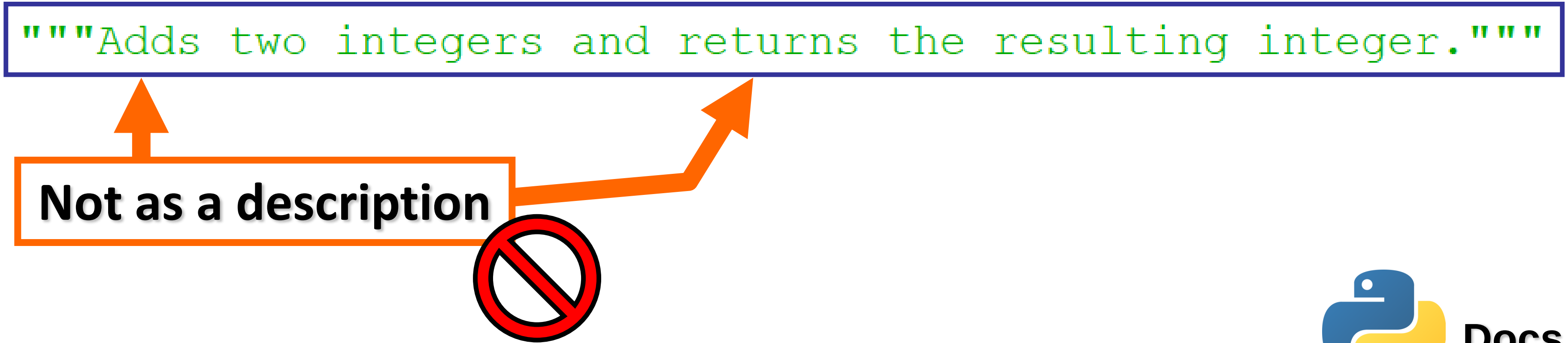
Command

Docs

One-line Docstrings



- The docstring is a phrase ending in a period. It prescribes the function or method's effect as a command ("Do this", "Return that"), not as a description; e.g. don't write "Returns the pathname ...".



```
"""Adds two integers and returns the resulting integer."""
```

Not as a description

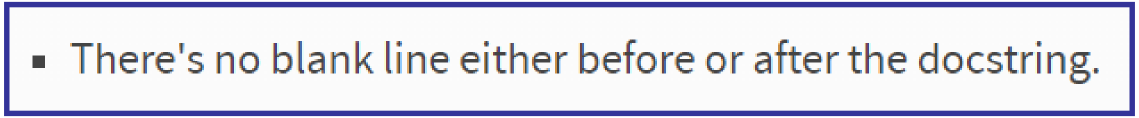
Docs

One–line Docstrings

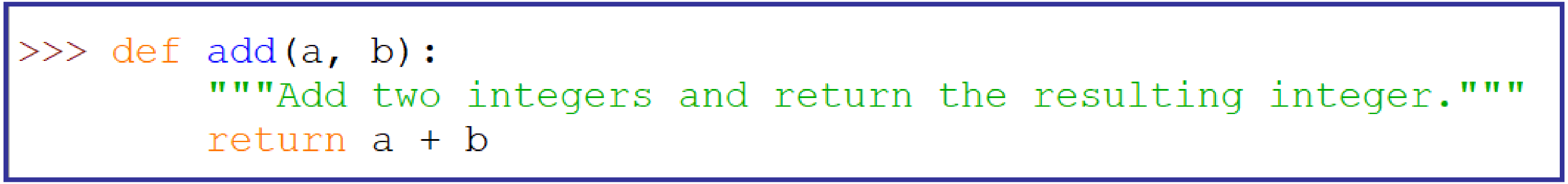
```
"""Adds two integers and returns the resulting integer."""
```

Google Style

One-line Docstrings



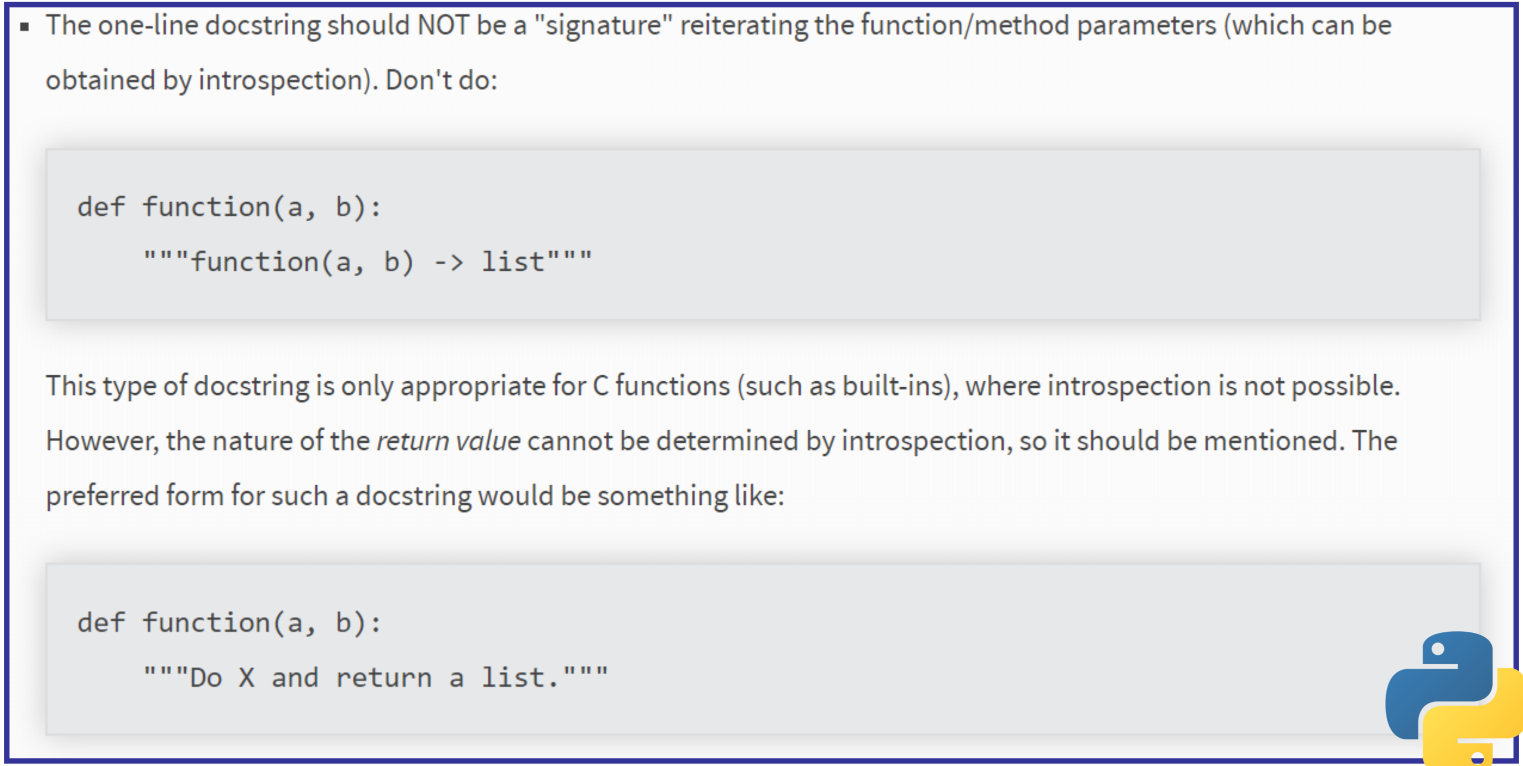
- There's no blank line either before or after the docstring.



```
>>> def add(a, b):
        """Add two integers and return the resulting integer."""
        return a + b
```

Docs

One-line Docstrings



- The one-line docstring should NOT be a "signature" reiterating the function/method parameters (which can be obtained by introspection). Don't do:

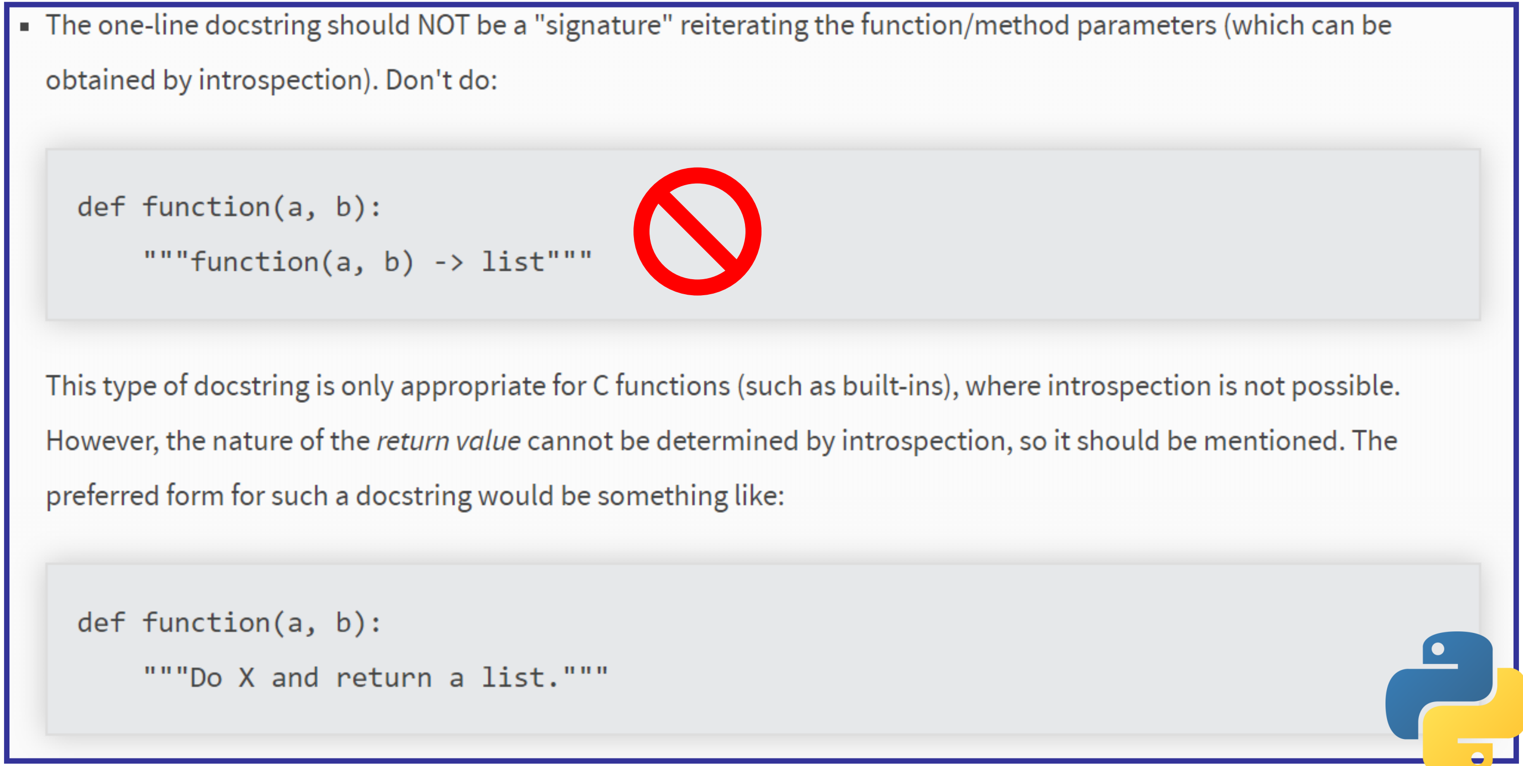
```
def function(a, b):
    """function(a, b) -> list"""
```

This type of docstring is only appropriate for C functions (such as built-ins), where introspection is not possible. However, the nature of the *return value* cannot be determined by introspection, so it should be mentioned. The preferred form for such a docstring would be something like:

```
def function(a, b):
    """Do X and return a list."""
```

Docs

One–line Docstrings



- The one-line docstring should NOT be a "signature" reiterating the function/method parameters (which can be obtained by introspection). Don't do:

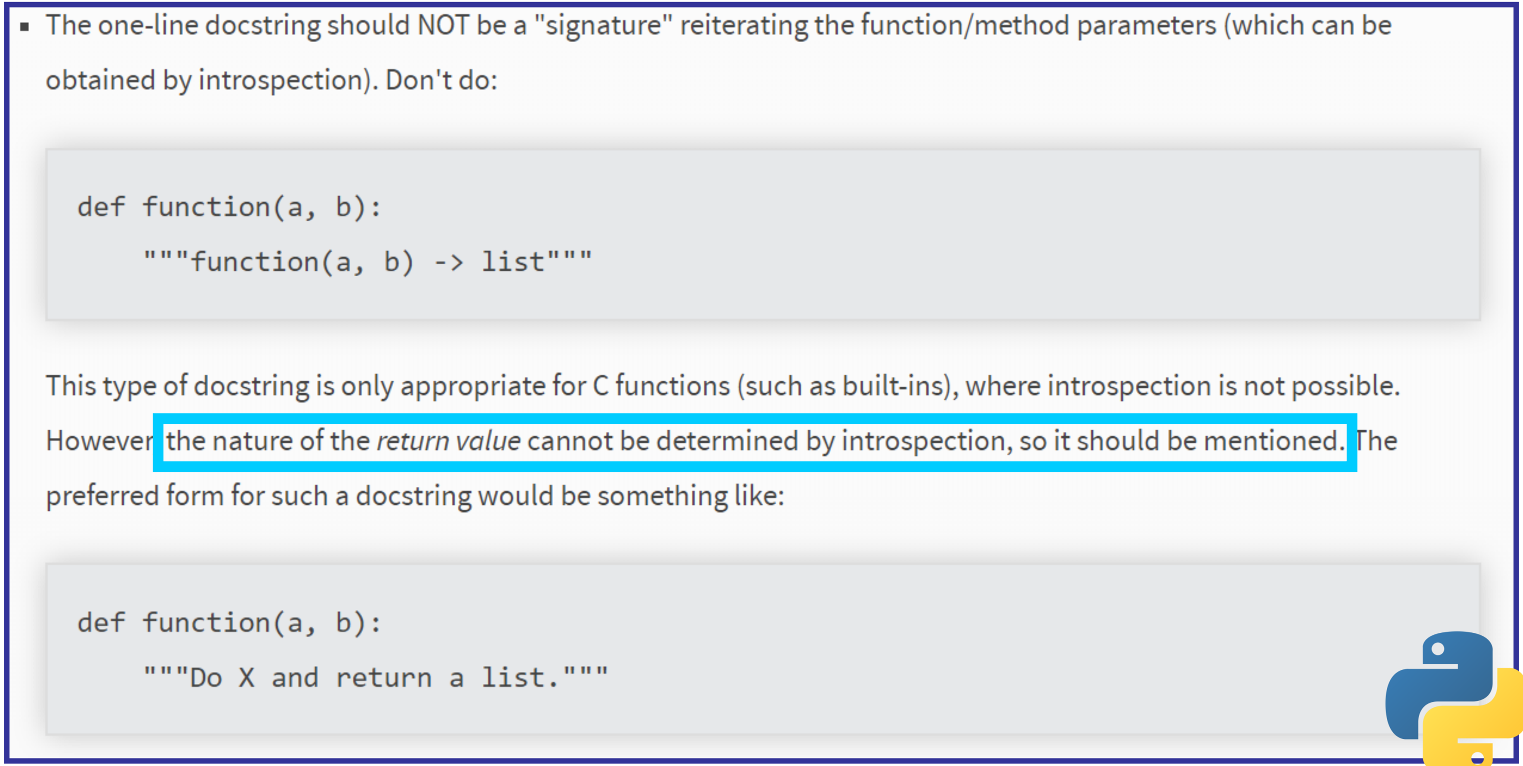
```
def function(a, b):
    """function(a, b) -> list"""
```

This type of docstring is only appropriate for C functions (such as built-ins), where introspection is not possible. However, the nature of the *return value* cannot be determined by introspection, so it should be mentioned. The preferred form for such a docstring would be something like:

```
def function(a, b):
    """Do X and return a list."""
```

Docs

One-line Docstrings



- The one-line docstring should NOT be a "signature" reiterating the function/method parameters (which can be obtained by introspection). Don't do:

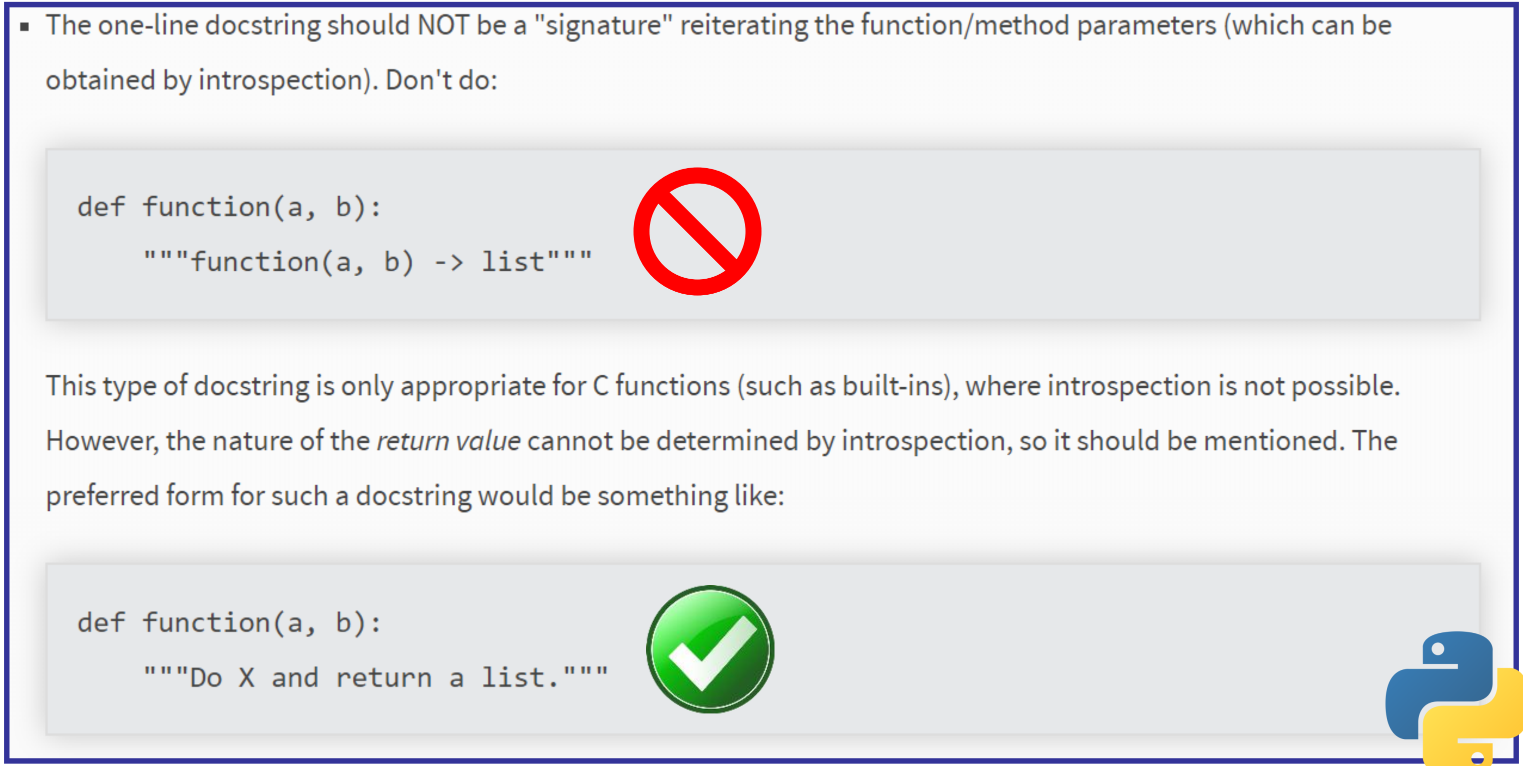
```
def function(a, b):
    """function(a, b) -> list"""
```

This type of docstring is only appropriate for C functions (such as built-ins), where introspection is not possible. However, the nature of the *return value* cannot be determined by introspection, so it should be mentioned. The preferred form for such a docstring would be something like:

```
def function(a, b):
    """Do X and return a list."""
```

Docs

One-line Docstrings



- The one-line docstring should NOT be a "signature" reiterating the function/method parameters (which can be obtained by introspection). Don't do:

```
def function(a, b):
    """function(a, b) -> list"""
```

🚫

This type of docstring is only appropriate for C functions (such as built-ins), where introspection is not possible. However, the nature of the *return value* cannot be determined by introspection, so it should be mentioned. The preferred form for such a docstring would be something like:

```
def function(a, b):
    """Do X and return a list."""
```

✅

Docs

Syntax & Content

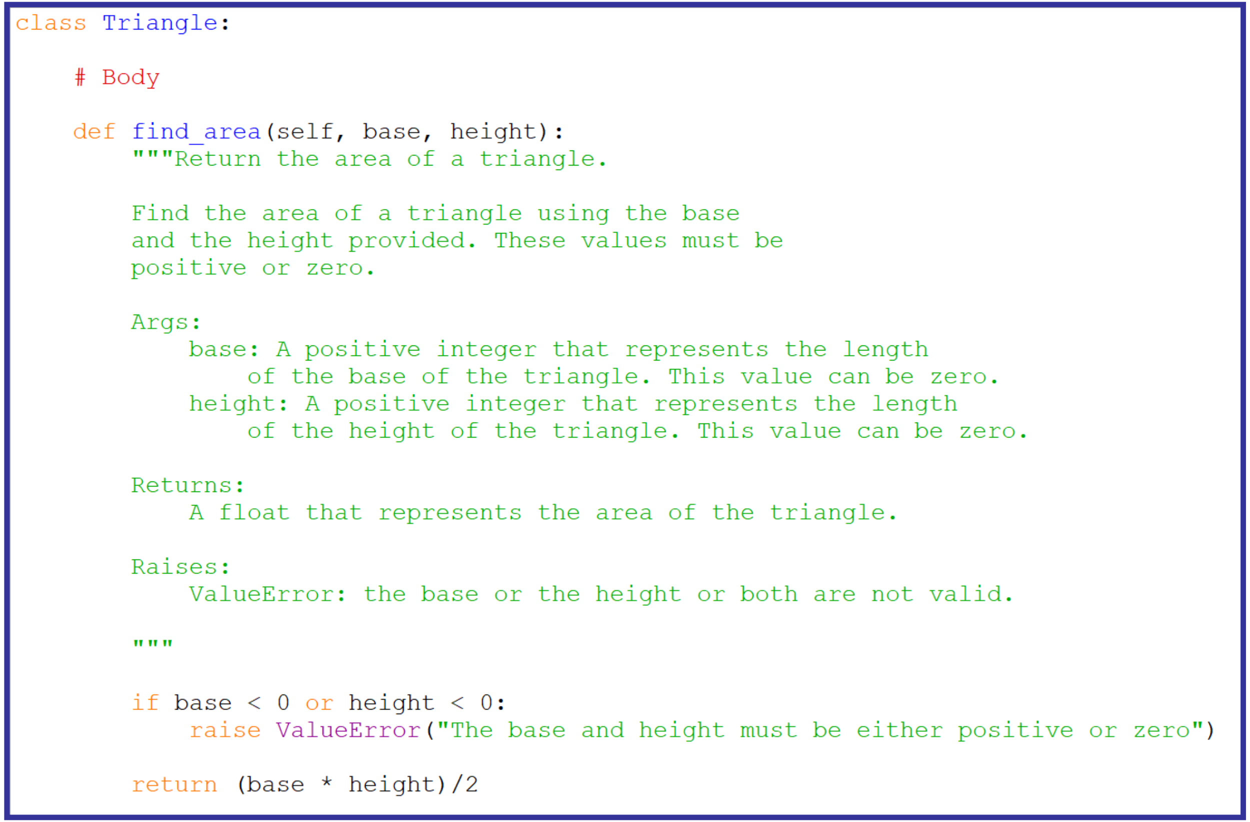
Multi-line Docstrings

Syntax & Content

Multi-line Docstrings

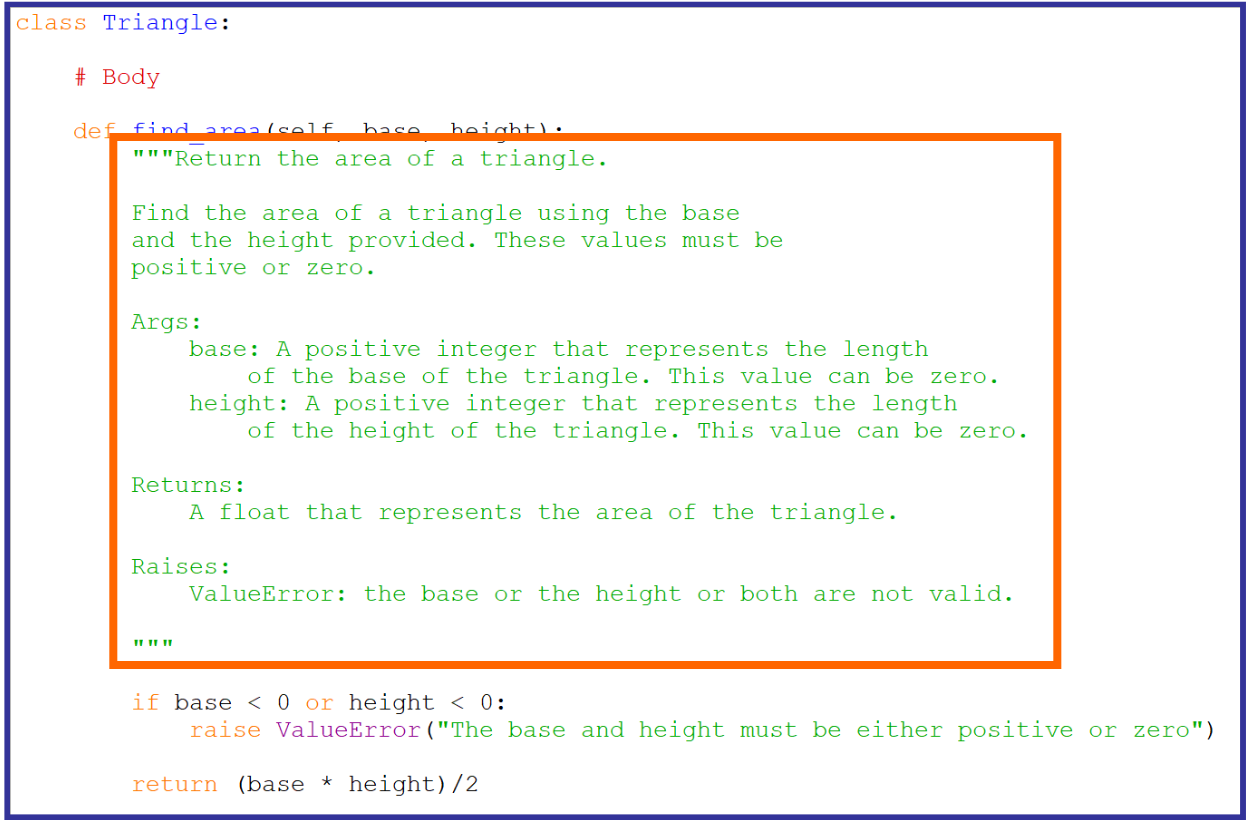
Classes, Methods, and Modules

Syntax & Content



```python
class Triangle:

    # Body

    def find_area(self, base, height):
        """Return the area of a triangle.

        Find the area of a triangle using the base
        and the height provided. These values must be
        positive or zero.

        Args:
            base: A positive integer that represents the length
                of the base of the triangle. This value can be zero.
            height: A positive integer that represents the length
                of the height of the triangle. This value can be zero.

        Returns:
            A float that represents the area of the triangle.

        Raises:
            ValueError: the base or the height or both are not valid.

        """

        if base < 0 or height < 0:
            raise ValueError("The base and height must be either positive or zero")

        return (base * height)/2
```

Syntax & Content



```python
class Triangle:

    # Body

    def find_area(self, base, height):
        """Return the area of a triangle.

        Find the area of a triangle using the base
        and the height provided. These values must be
        positive or zero.

        Args:
            base: A positive integer that represents the length
                of the base of the triangle. This value can be zero.
            height: A positive integer that represents the length
                of the height of the triangle. This value can be zero.

        Returns:
            A float that represents the area of the triangle.

        Raises:
            ValueError: the base or the height or both are not valid.

        """

        if base < 0 or height < 0:
            raise ValueError("The base and height must be either positive or zero")

        return (base * height)/2
```

Methods or Functions

- Arguments
- Optional arguments
- Return value
- Side effects (e.g Mutation)
- Exceptions raised
- Restrictions on when it can be called

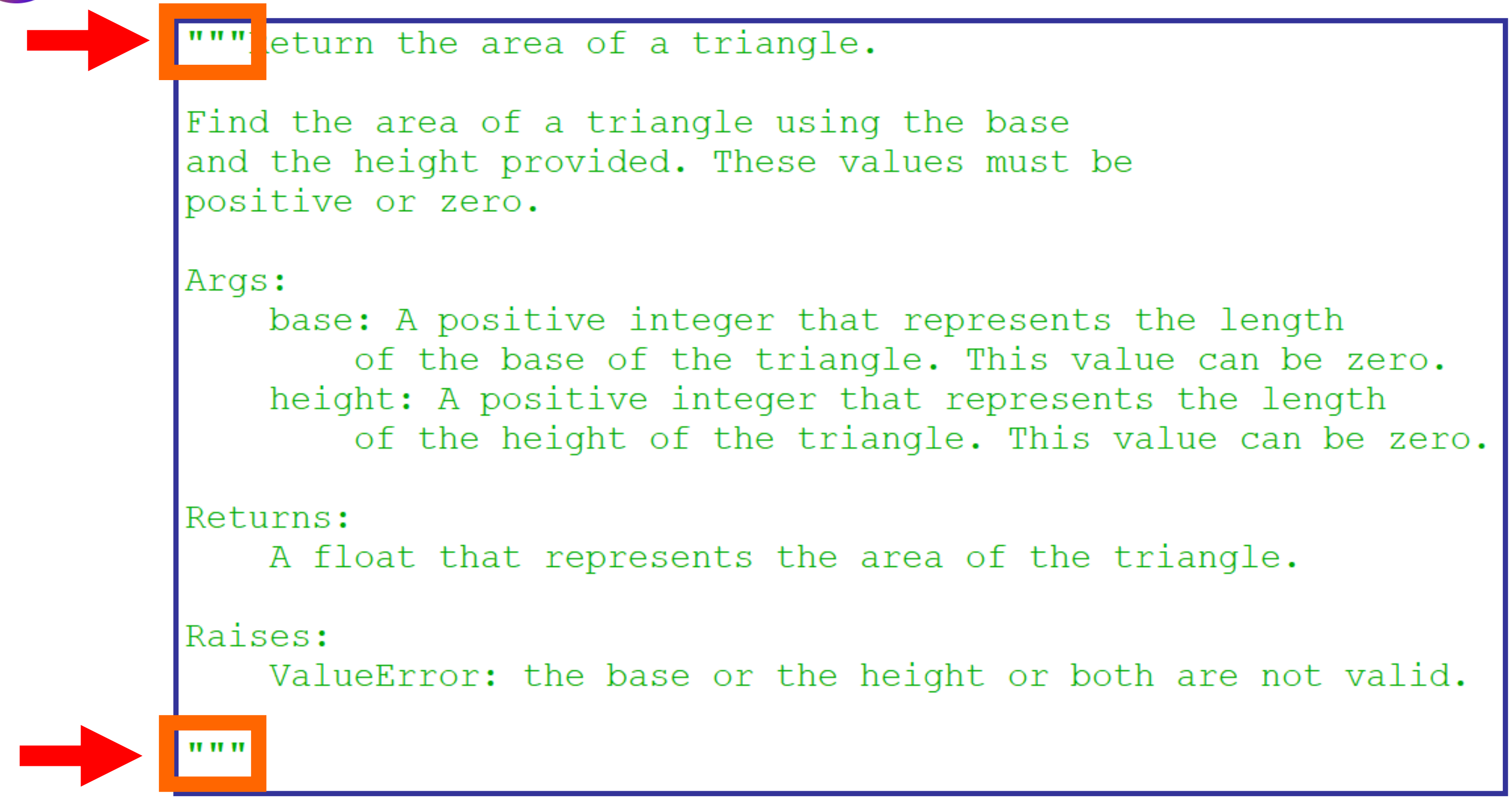
Classes

- Purpose.
- List public methods.
- List public instance variables.
- Effects of inheritance.
- __init__() documented separately.
- Individual methods documented individually.

Syntax & Content

```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content



```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

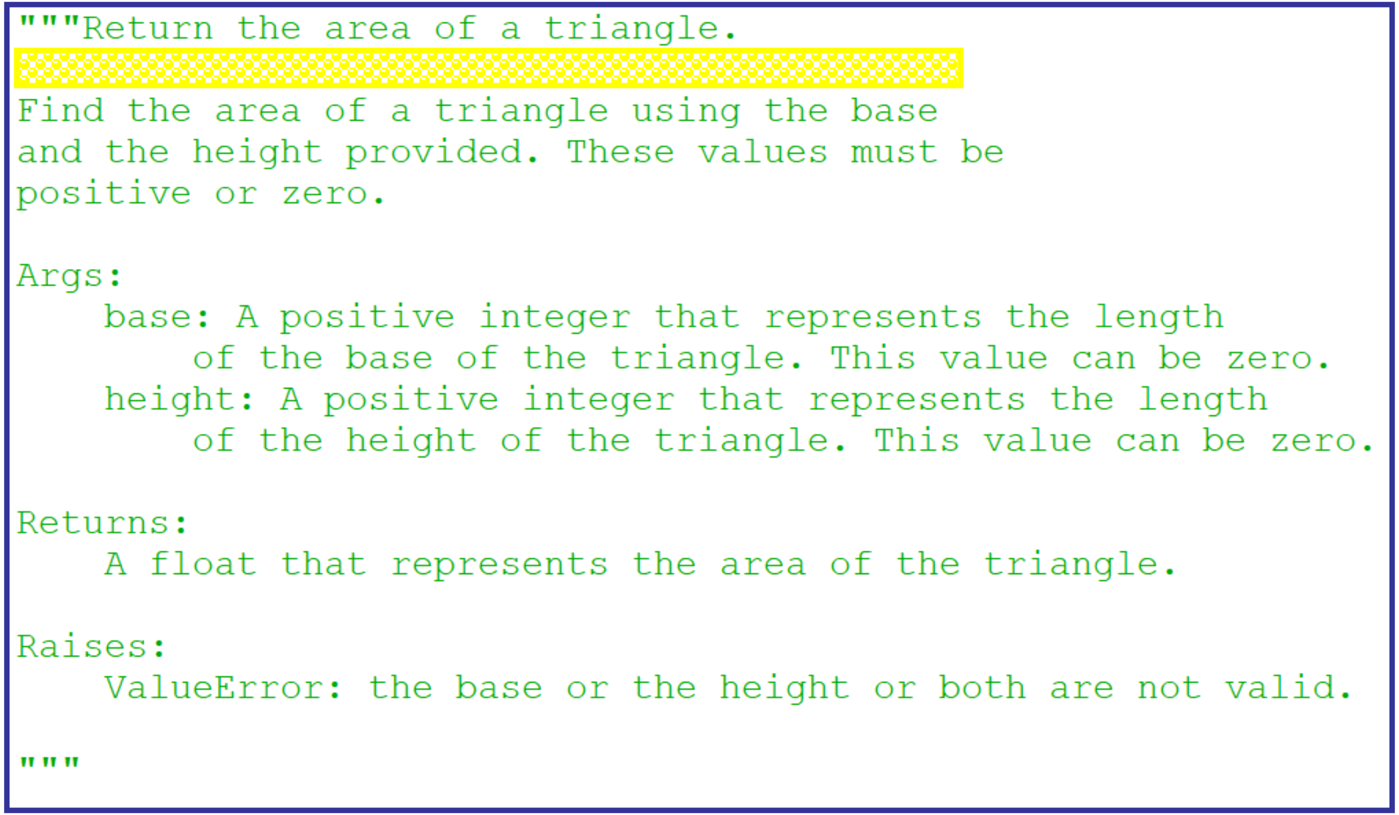
Syntax & Content

```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

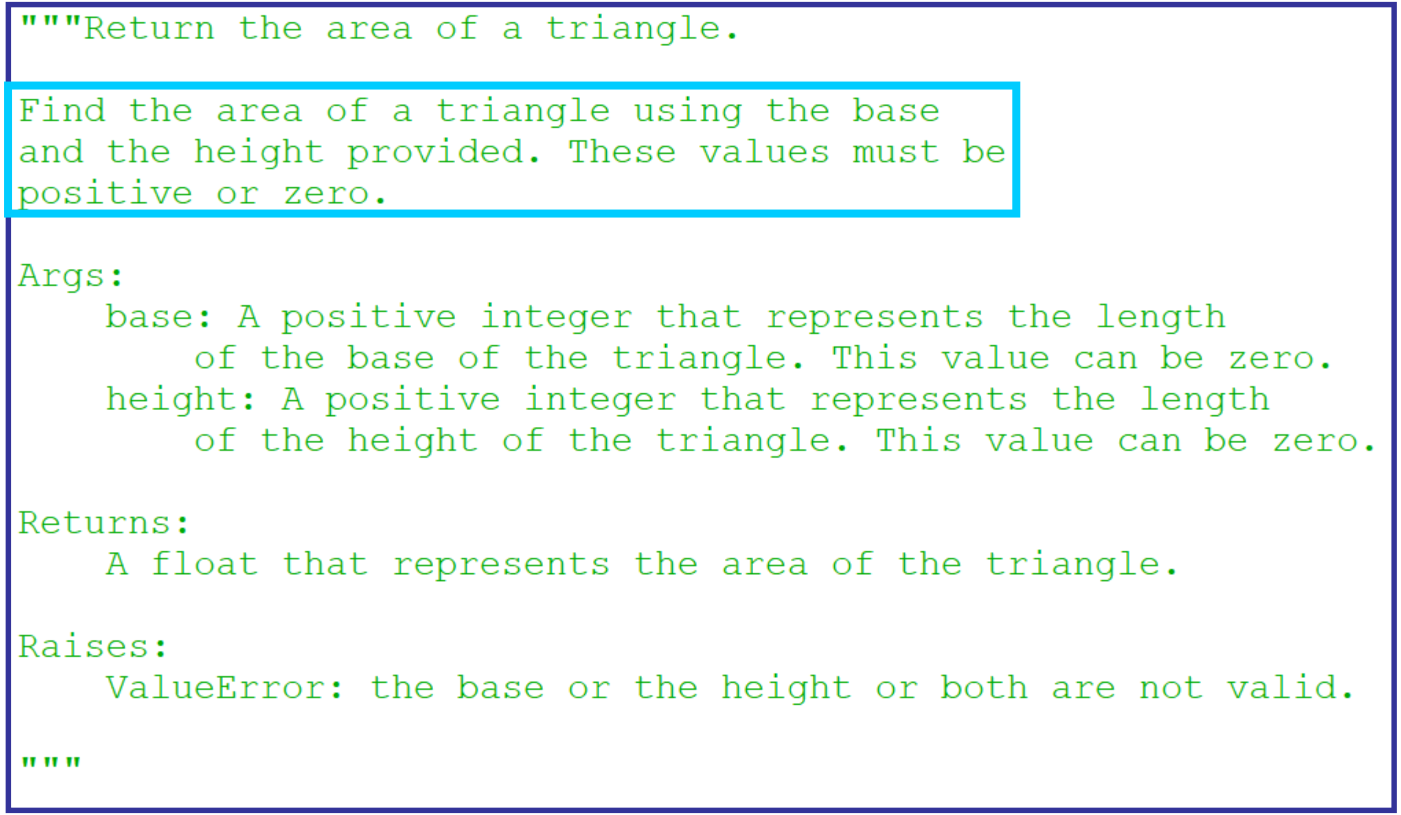
```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content



```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

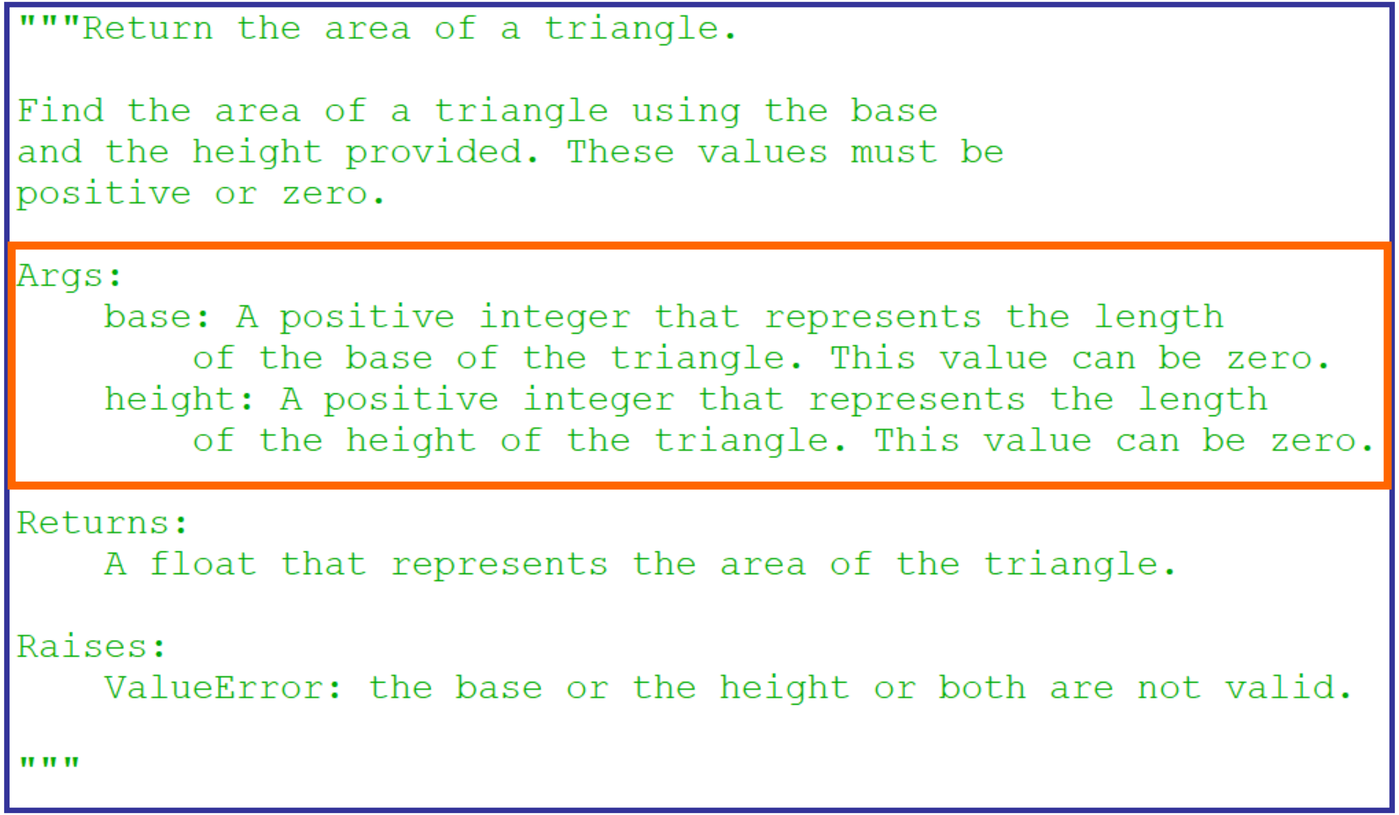


```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

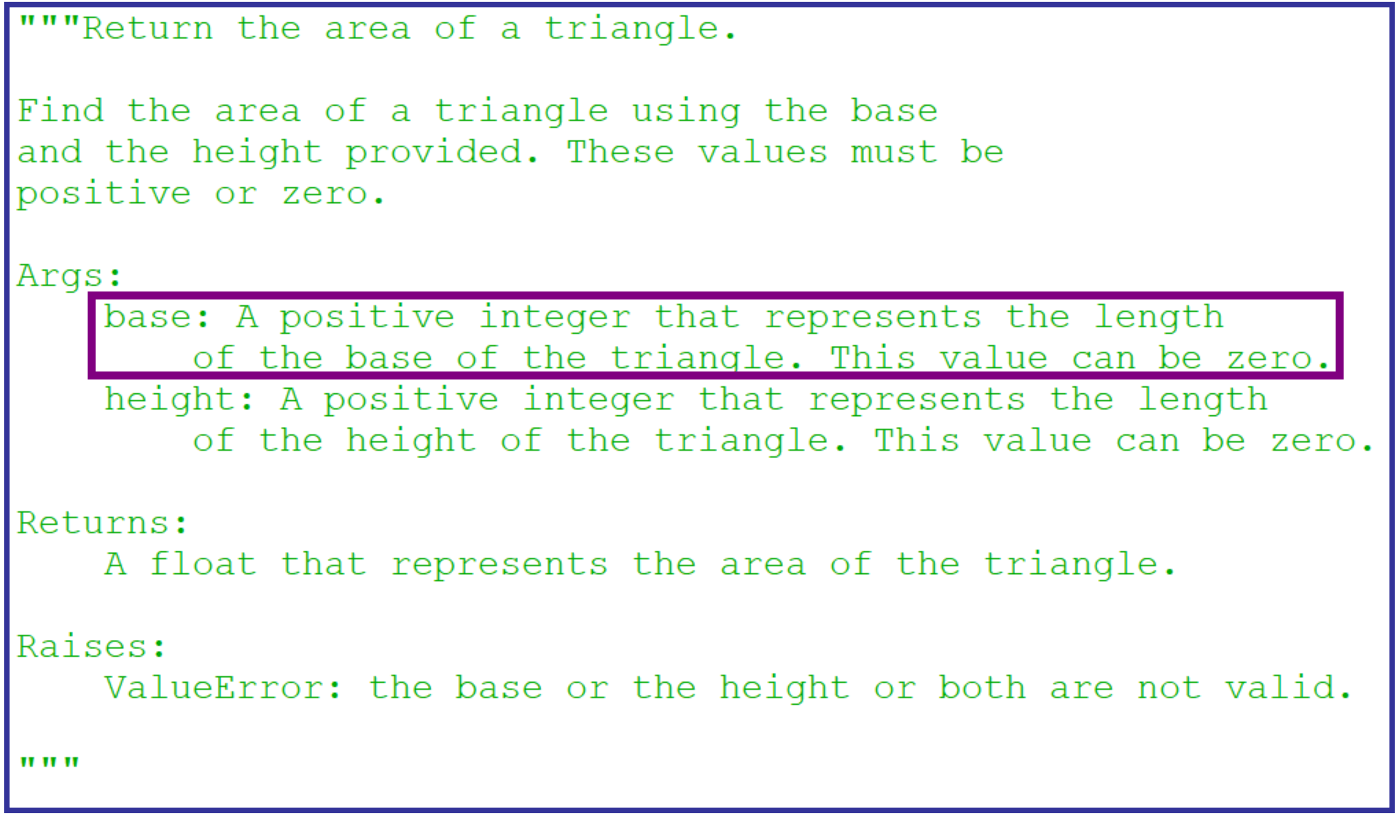
```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content



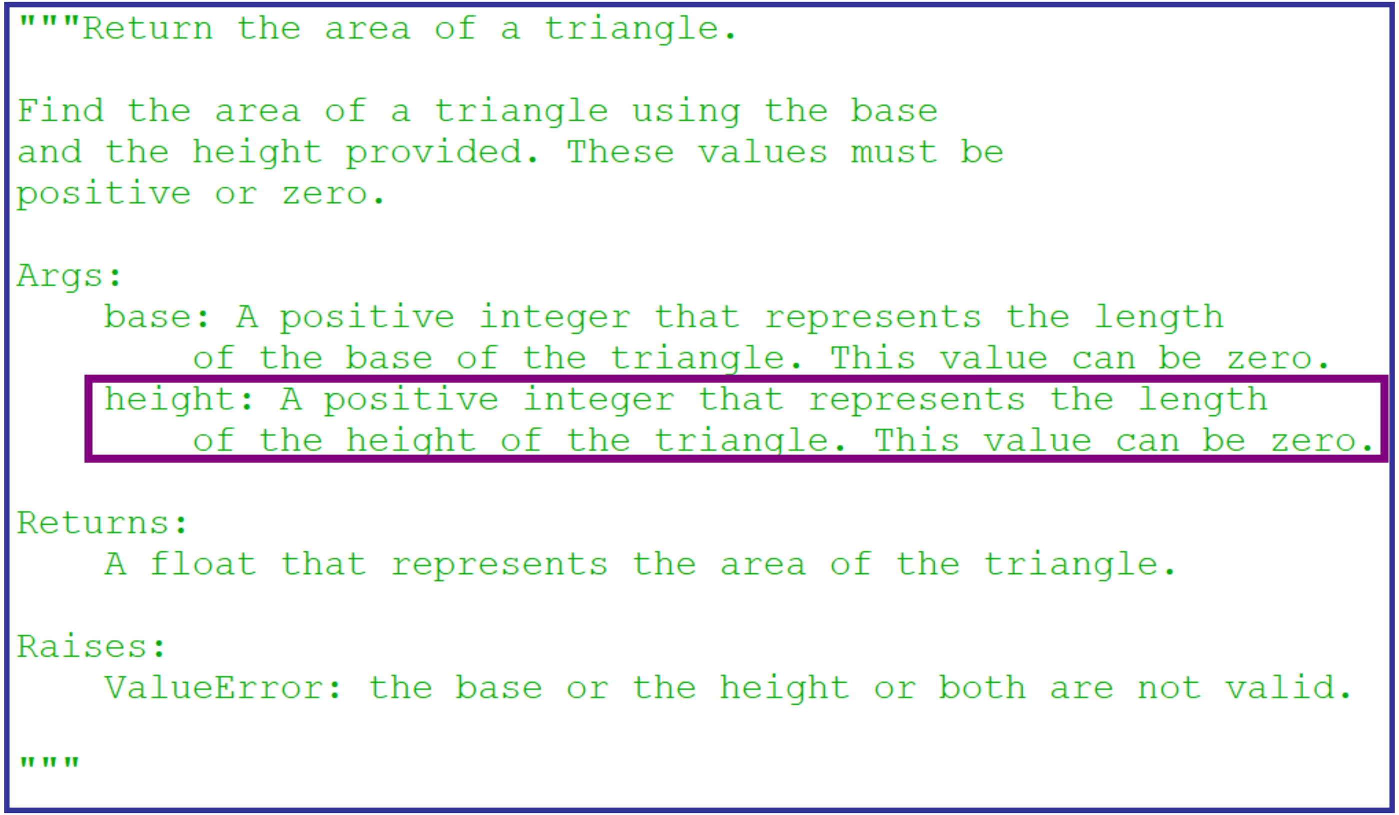
```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content



```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

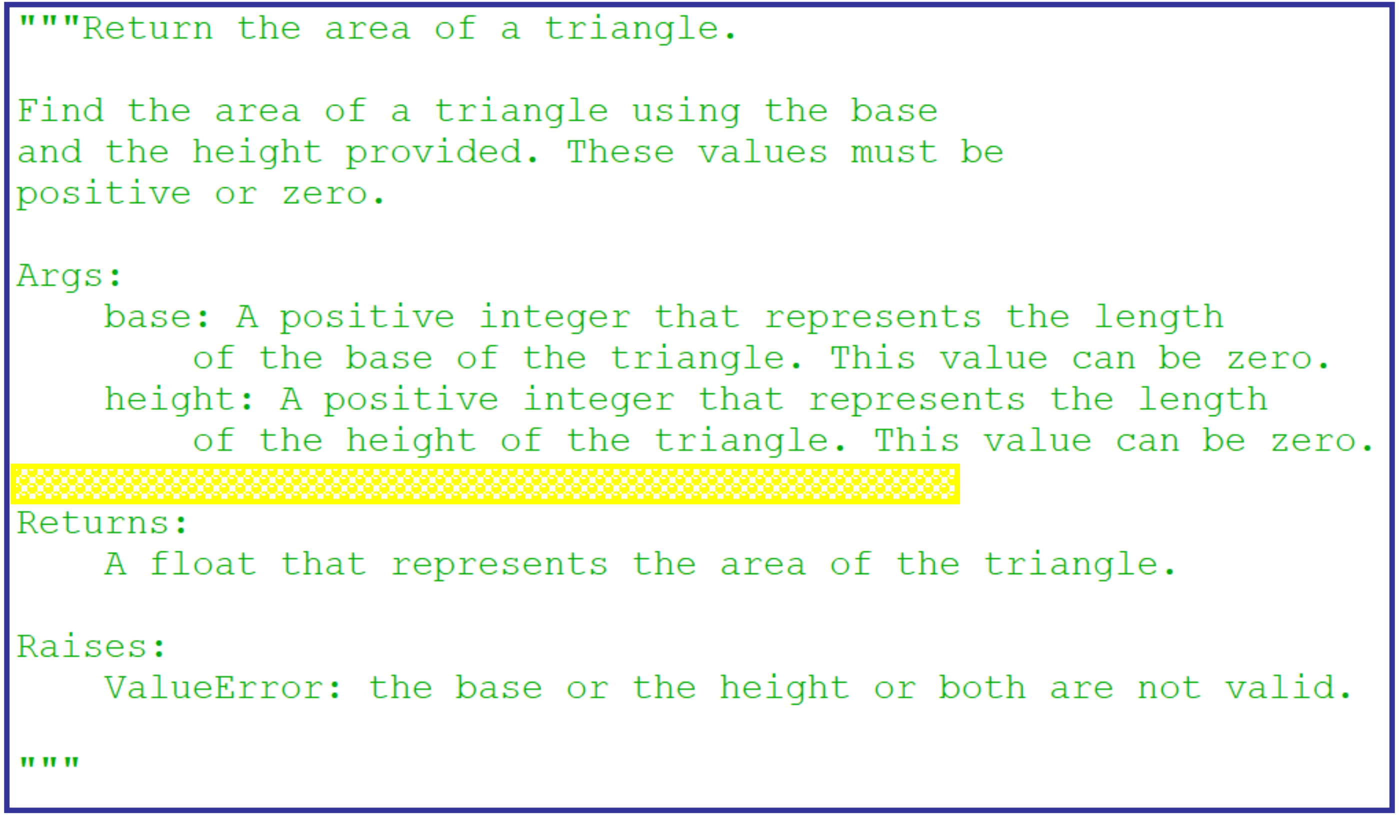


```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

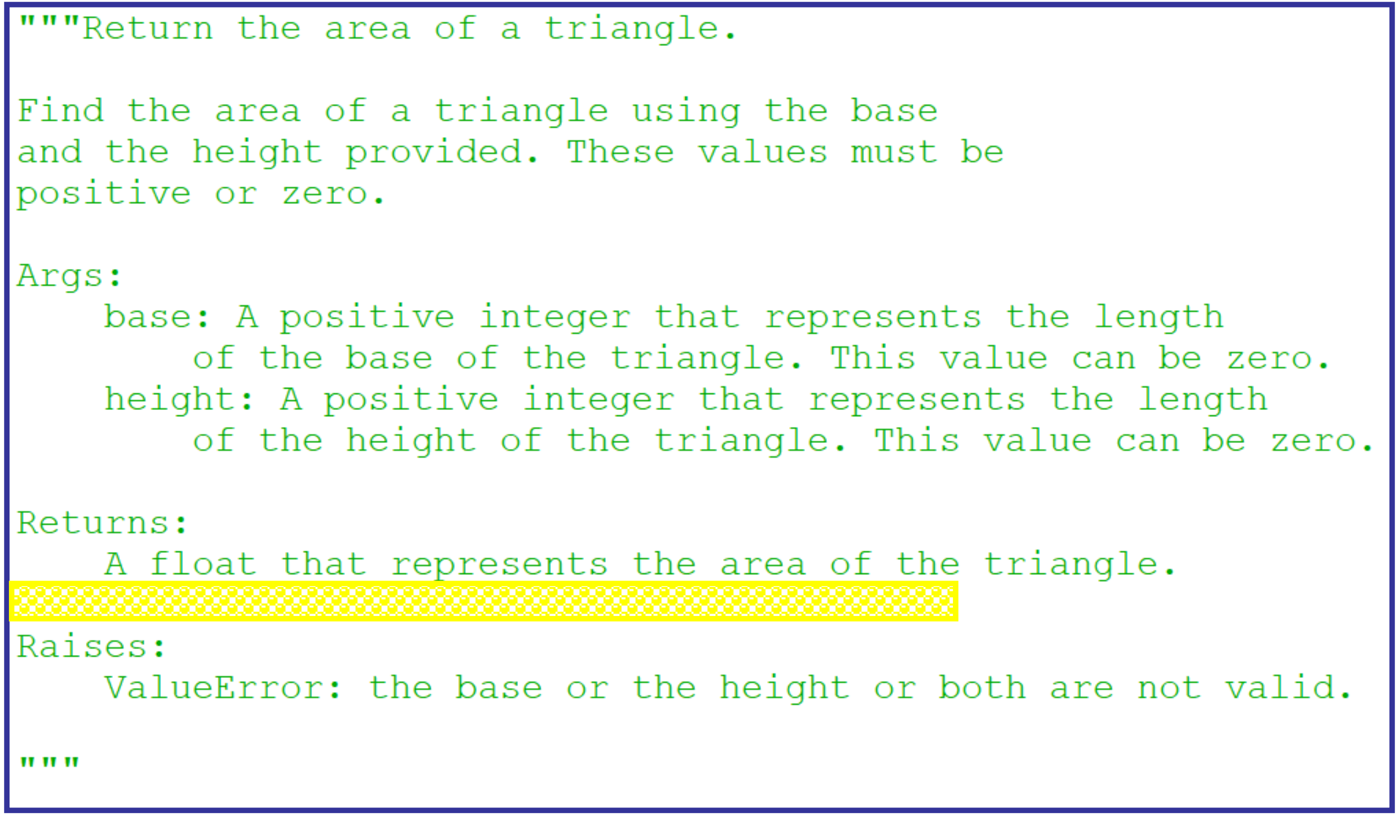


```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content



```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Syntax & Content

```
"""Return the area of a triangle.

Find the area of a triangle using the base
and the height provided. These values must be
positive or zero.

Args:
    base: A positive integer that represents the length
        of the base of the triangle. This value can be zero.
    height: A positive integer that represents the length
        of the height of the triangle. This value can be zero.

Returns:
    A float that represents the area of the triangle.

Raises:
    ValueError: the base or the height or both are not valid.

"""
```

Now... Classes