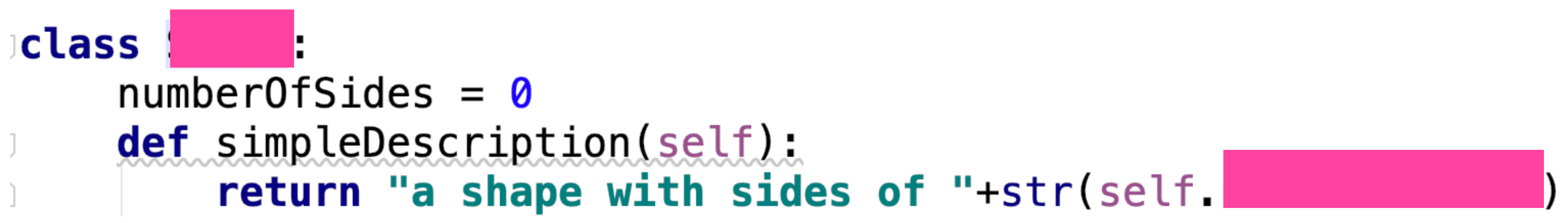
Class

Shape
NamedShape
Square
EquilateralTriangular

宣告稱為Shape的類別

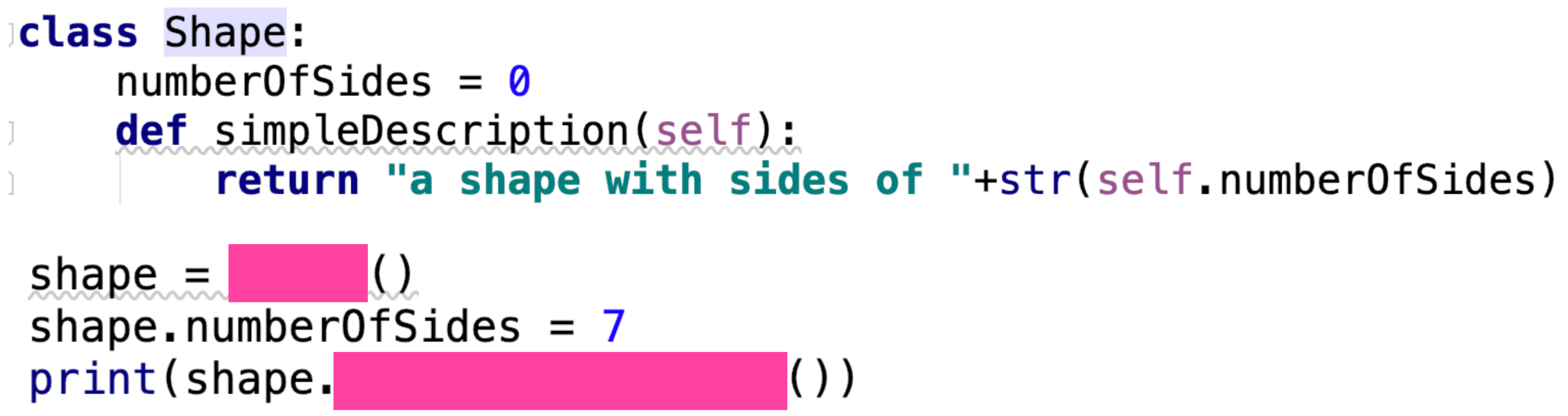
- 形狀類別-   class Shape
  - 變數 numberOfSides
  - 函數 simpleDescription

宣告稱為Shape的類別



```
class [redacted]:
    numberOfSides = 0
    def simpleDescription(self):
        return "a shape with sides of "+str(self.[redacted])
```

宣告類別為Shape的7邊形變數

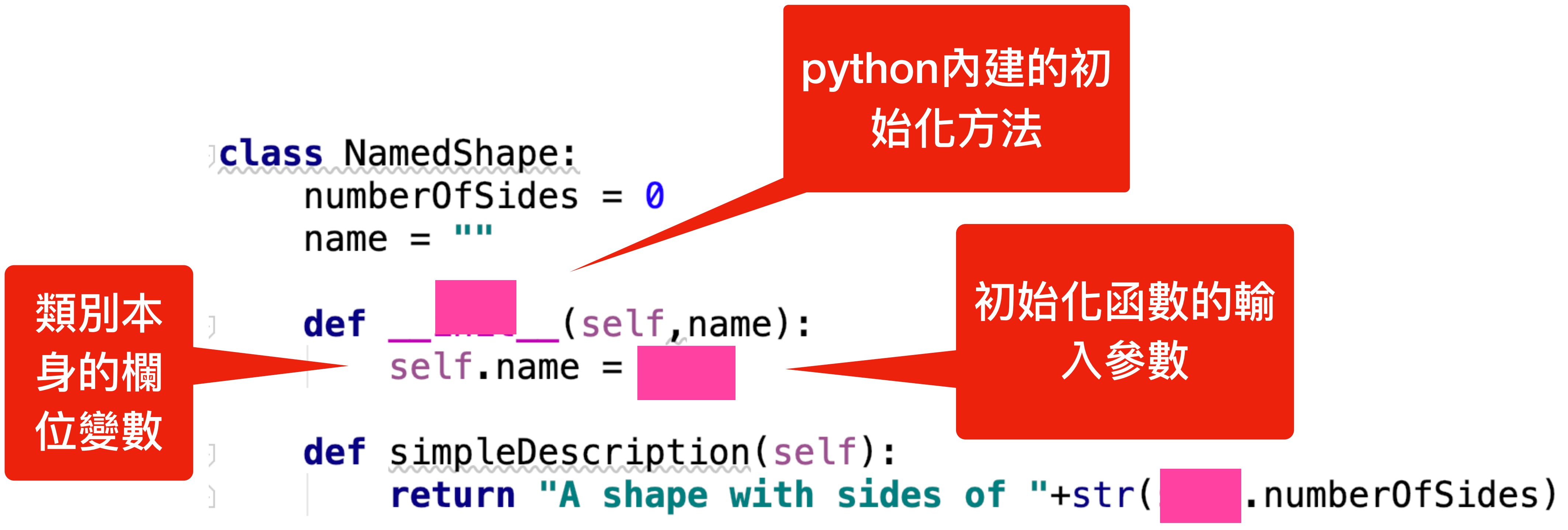


```python
class Shape:
    numberOfSides = 0
    def simpleDescription(self):
        return "a shape with sides of "+str(self.numberOfSides)

shape = [redacted]()
shape.numberOfSides = 7
print(shape.[redacted]())
```

宣告稱為NamedShape的類別

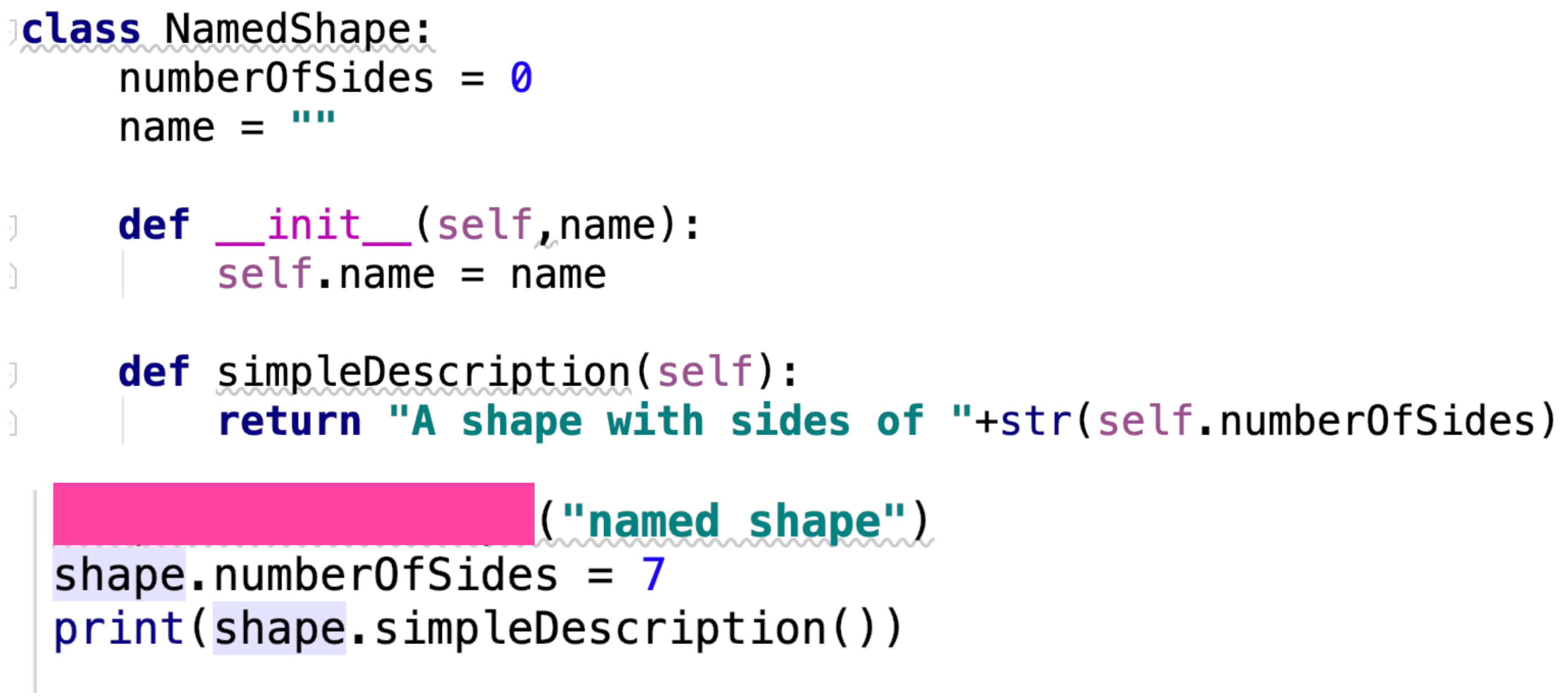
- 有名稱的形狀-　class NamedShape
    - 變數 numberOfSides, name　Properties
    - 函數 init　Methods
    - 函數 simpleDescription

宣告稱為NamedShape的類別



```
class NamedShape:
    numberOfSides = 0
    name = ""

    def __[redacted]__(self,name):
        self.name = [redacted]

    def simpleDescription(self):
        return "A shape with sides of "+str([redacted].numberOfSides)
```

宣告型態為NamedShape的變數 shape



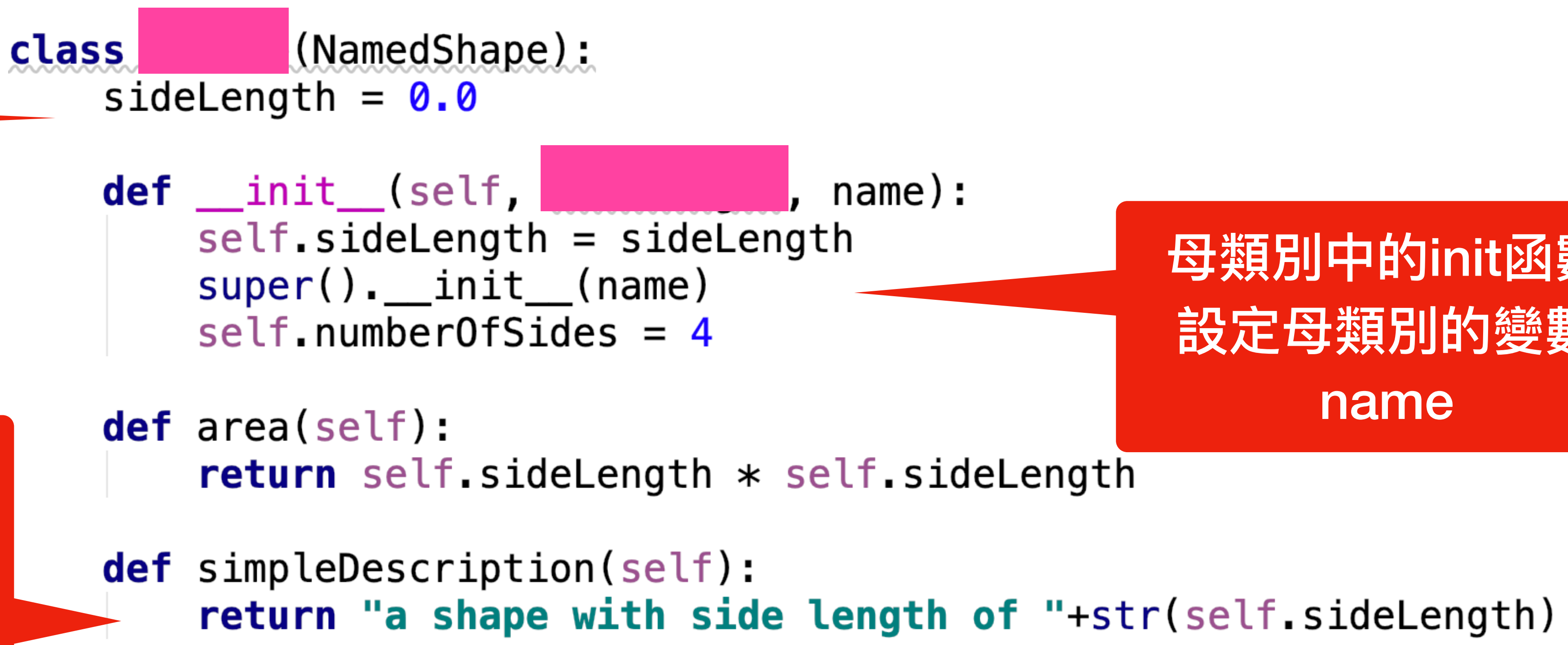
```python
class NamedShape:
    numberOfSides = 0
    name = ""

    def __init__(self,name):
        self.name = name

    def simpleDescription(self):
        return "A shape with sides of "+str(self.numberOfSides)

[redacted]("named shape")
shape.numberOfSides = 7
print(shape.simpleDescription())
```

宣告類別為NamedShape的類別Square

- 正方形-　class Square(NamedShape)
    - 變數 sideLength
    - init
    - 函數 面積
    - 覆寫函數 simpleDescription

母類別為NamedShape

宣告類別為NamedShape的類別 Square



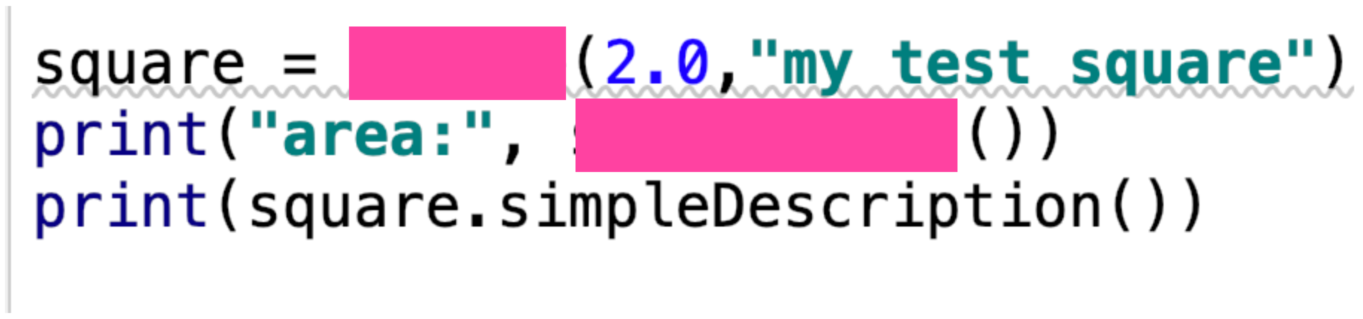
```
class [redacted](NamedShape):
    sideLength = 0.0

    def __init__(self, [redacted], name):
        self.sideLength = sideLength
        super().__init__(name)
        self.numberOfSides = 4

    def area(self):
        return self.sideLength * self.sideLength

    def simpleDescription(self):
        return "a shape with side length of "+str(self.sideLength)
```

Public variable

母類別中的init函數設定母類別的變數name

覆寫母類別中的simpleDescription函數

宣告類別為Square的正邊形變數 square，

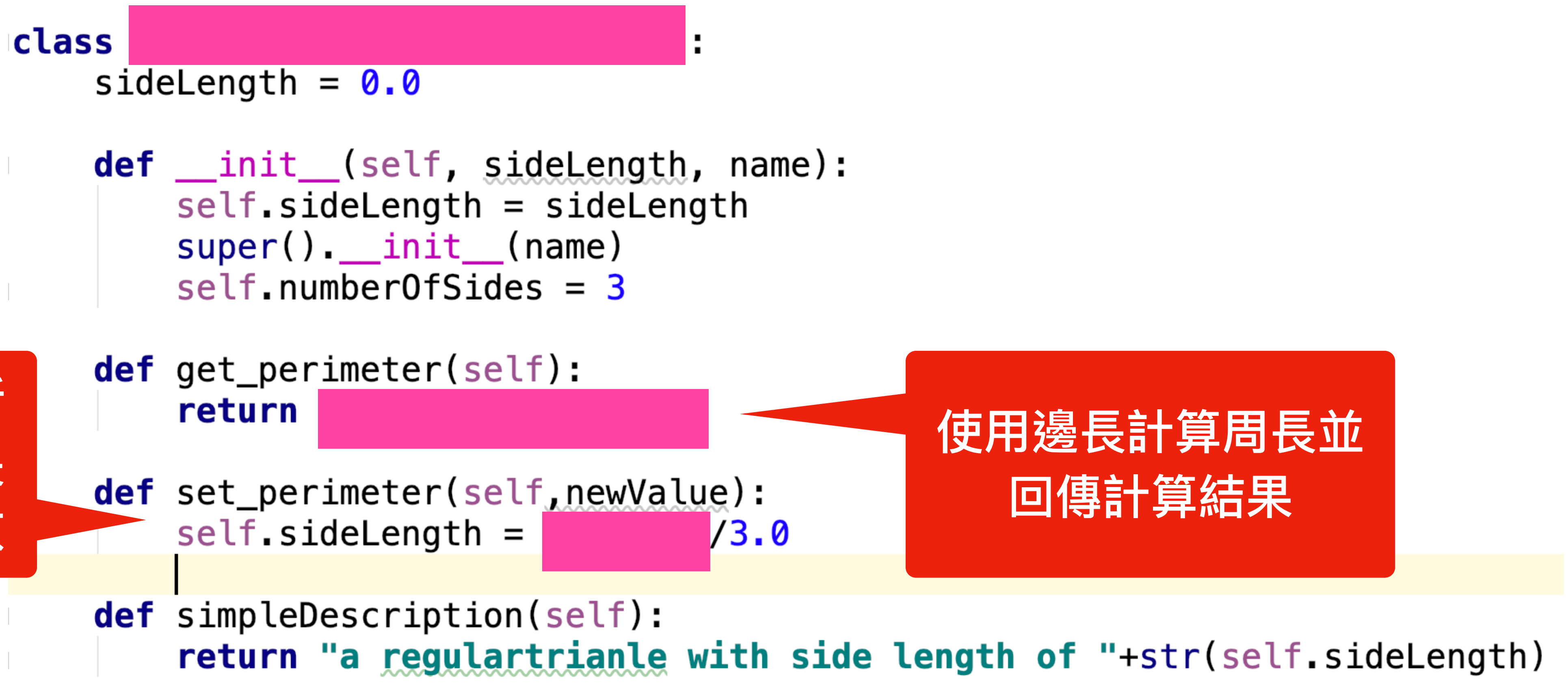


```
square = [redacted](2.0,"my test square")
print("area:", [redacted]())
print(square.simpleDescription())
```

宣告類別為NamedShape的類別RegularTriangle

- 等邊三角形-　class RegularTriangle(NamedShape)
  - 變數 sideLength
  - init
  - 函數 周長
  - 覆寫函數 simpleDescription

宣告類別為NamedShape的類別 RegularTriangle



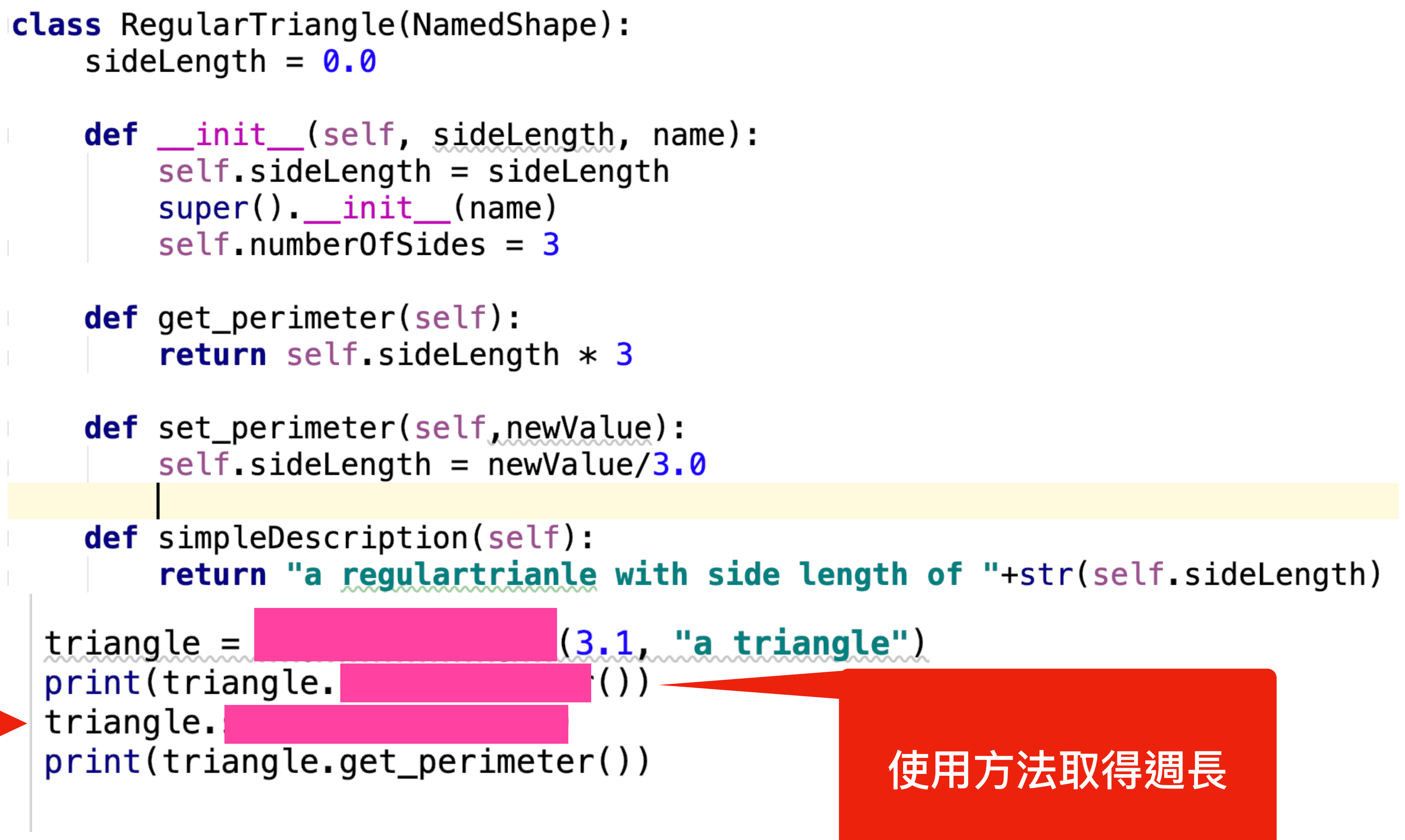
```
class                          :
    sideLength = 0.0

    def __init__(self, sideLength, name):
        self.sideLength = sideLength
        super().__init__(name)
        self.numberOfSides = 3

    def get_perimeter(self):
        return

    def set_perimeter(self,newValue):
        self.sideLength =         /3.0

    def simpleDescription(self):
        return "a regulartrianle with side length of "+str(self.sideLength)
```

以周長的新值計算邊長並設定變數

使用邊長計算周長並回傳計算結果

宣告類別為RegularTriangle的變數 triangular，並呼叫其方法

et_perimeter set_perimeter(9.9)

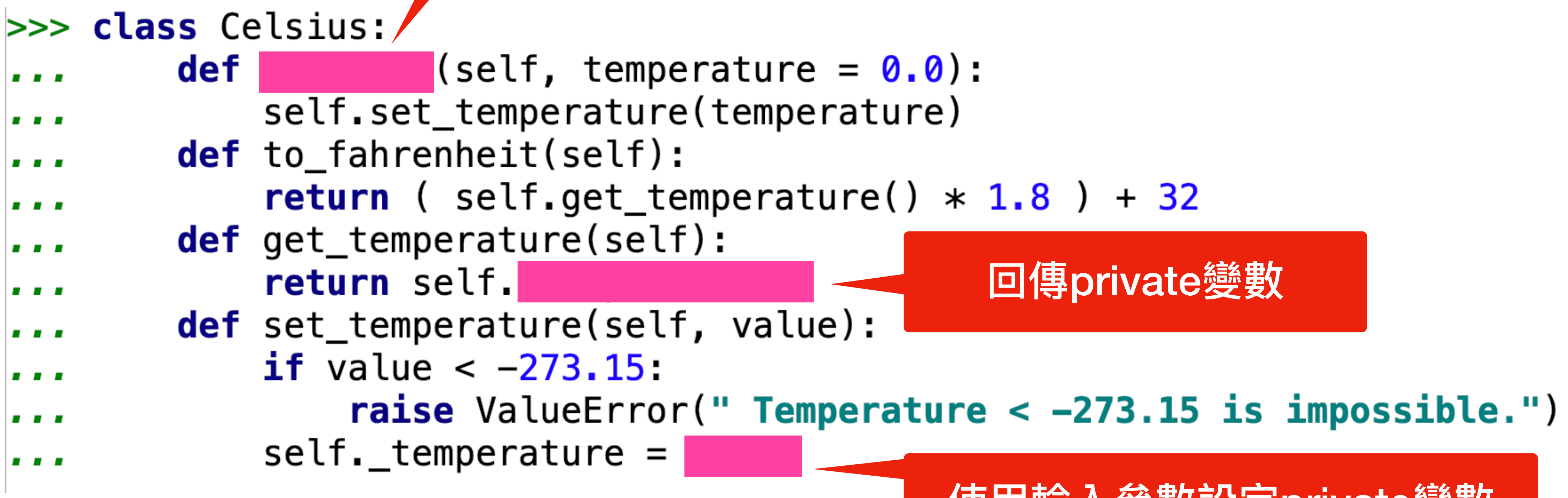


```
class RegularTriangle(NamedShape):
    sideLength = 0.0

    def __init__(self, sideLength, name):
        self.sideLength = sideLength
        super().__init__(name)
        self.numberOfSides = 3

    def get_perimeter(self):
        return self.sideLength * 3

    def set_perimeter(self,newValue):
        self.sideLength = newValue/3.0

    def simpleDescription(self):
        return "a regulartrianle with side length of "+str(self.sideLength)
triangle = [redacted](3.1, "a triangle")
print(triangle.[redacted]())
triangle.[redacted]
print(triangle.get_perimeter())
```

使用方法將週長設定為9.9

使用方法取得週長

使用python內建初始化方法



```
>>> class Celsius:
...     def           (self, temperature = 0.0):
...         self.set_temperature(temperature)
...     def to_fahrenheit(self):
...         return ( self.get_temperature() * 1.8 ) + 32
...     def get_temperature(self):
...         return self.
...     def set_temperature(self, value):
...         if value < -273.15:
...             raise ValueError(" Temperature < -273.15 is impossible.")
...         self._temperature =
```

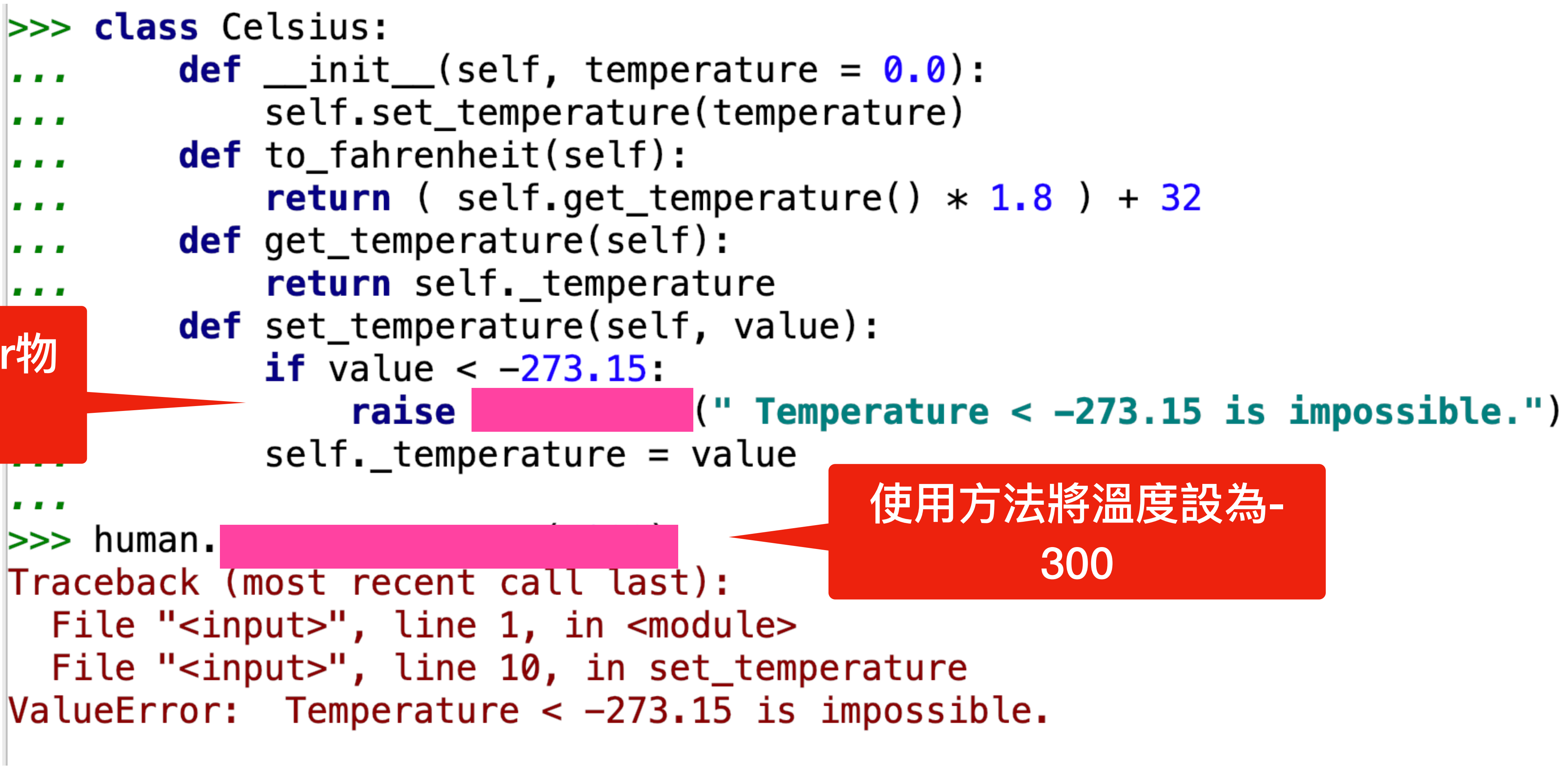
回傳private變數

使用輸入參數設定private變數

```
>>> class Celsius:
...     def __init__(self, temperature = 0.0):
...         self.set_temperature(temperature)
...     def to_fahrenheit(self):
...         return ( self.get_temperature() * 1.8 ) + 32
...     def get_temperature(self):
            return self._temperature
        def set_temperature(self, value):
            if value < -273.15:
                raise ValueError(" Temperature < -273.15 is impossible.")
            self._temperature = value
...
... [redacted](37)
... print(human.[redacted]())
... print(human.to_fahrenheit())
37
98.60000000000001
```

宣告型態為
Celsius的變數
human

使用方法取得溫數數值



```
>>> class Celsius:
...     def __init__(self, temperature = 0.0):
...         self.set_temperature(temperature)
...     def to_fahrenheit(self):
...         return ( self.get_temperature() * 1.8 ) + 32
...     def get_temperature(self):
...         return self._temperature
        def set_temperature(self, value):
            if value < -273.15:
                raise [redacted](" Temperature < -273.15 is impossible.")
...         self._temperature = value
...
>>> human.[redacted]
Traceback (most recent call last):
  File "<input>", line 1, in <module>
  File "<input>", line 10, in set_temperature
ValueError:  Temperature < -273.15 is impossible.
```

引發ValueError物件

使用方法將溫度設為-300