

Python Delete Property

If this Python Tutorial saves you
hours of work, please **whitelist it in**
your ad blocker 🙏 and

Donate Now

(<https://www.pythontutorial.net/donation/>)

to help us ❤️ pay for the web
hosting fee and CDN to keep the

website running.

Summary: in this tutorial, you'll learn how to use the `property()` class to delete the property of an object.

To create a `property` (<https://www.pythontutorial.net/python-oop/python-properties/>) of a class, you use the `@property` decorator (<https://www.pythontutorial.net/python-oop/python-property-decorator/>). Underhood, the `@property` decorator uses the `property` class that has three methods: setter, getter, and deleter.

By using the deleter, you can delete a property from an object. Notice that the `deleter()` method deletes a property of an object, not a `class` (<https://www.pythontutorial.net/python-oop/python-class/>).

The following defines the `Person` class with the `name` property:

```
from pprint import pprint
```

```
class Person:  
    def __init__(self, name):  
        self._name = name
```

```
@property
```

```

def name(self):
    return self._name

@name.setter
def name(self, value):
    if value.strip() == '':
        raise ValueError('name cannot be empty')
    self._name = value

@name.deleter
def name(self):
    del self._name

```

In the `Person` class, we use the `@name.deleter` decorator. Inside the deleter, we use the `del` keyword to delete the `_name` attribute of the `Person` instance.

The following shows the `__dict__` of the `Person` class:

```
pprint(Person.__dict__)
```

Output:

```

mappingproxy({'__dict__': <attribute '__dict__' of 'Person' objects>,
              '__doc__': None,
              '__init__': <function Person.__init__ at 0x000001DC41D62670>,
              '__module__': '__main__',
              '__weakref__': <attribute '__weakref__' of 'Person' objects>,
              'name': <property object at 0x000001DC41C89130>})

```

The `Person.__dict__` class has the `name` variable. The following creates a new instance of the `Person` class:

```
person = Person('John')
```

The `person.__dict__` has the `_name` attribute:

```
pprint(person.__dict__)
```

Output:

```
{'_name': 'John'}
```

The following uses the `del` keyword to delete the `name` property:

```
del person.name
```

Internally, Python will execute the `deleter` method that deletes the `_name` attribute from the `person` object. The `person.__dict__` will be empty like this:

```
{}
```

And if you attempt to access `name` property again, you'll get an error:

```
print(person.name)
```

Error:

```
AttributeError: 'Person' object has no attribute '_name'
```

Summary

- Use the deleter decorator to delete a property of an instance of a class.