

Short quiz on the basics of classes

Joakim Sundnes

Oct 21, 2022

Do this short quiz on your own. First answer the questions, then turn to the next page and check your answers. If you struggle with any of the questions you should consider re-reading Chapter 8 and/or the slides from the lectures. It is important to have a good understanding of classes before we move on to object oriented programming.

1 Quiz: fundamentals of classes

You should now have a good idea of what this class does:

```
from math import sin

class Fnc:
    def __init__(self, v0, v1):
        self.v0 = v0
        self.v1 = v1

    def __call__(self, x):
        return self.v0*sin(self.v1*x)

    def __str__(self):
        return f'f(x) = {self.v0} * sin({self.v1}*x)'

    def df(self,x):
        return self.v0*self.v1*cos(self.v1*x)
```

Questions:

1. How many attributes and how many methods does the class have?
2. Why do some methods have names on the form `__xxx__`?
3. What is the variable `self`?
4. Which methods are called by the code lines below?

```
f = Fnc(7,2)
print(f)
print(f(3.5))
print(f.df(3.5))
```

2 Answers:

```
from math import sin

class Fnc:
    def __init__(self, v0, v1):
        self.v0 = v0
        self.v1 = v1

    def __call__(self, x):
        return self.v0*sin(self.v1*x)

    def __str__(self):
        return f'f(x) = {self.v0} * sin({self.v1}*x)'

    def df(self,x):
        return self.v0*self.v1*cos(self.v1*x)
```

1. Two attributes and four methods
2. These are called special methods, and are automatically called by certain operations on the class instances
3. The variable `self` refers to the object/instance itself. When calling a method from outside the class, the object is automatically passed as the first argument to the method.
4. The methods called are

```
f = Fnc(7,2)           #calls __init__
print(f)              #calls __str__
print(f(3.5))         #calls __call__
print(f.df(3.5))     #calls df
```