

Forum:[Hack Forums](#)

Thread: [\[SOLVED\] Why is my function not defined?](#)

JuiceWater - July 9, 2017, 12:38 pm

A few hours ago I decided that I wanted to learn Python. I only have experience with static languages like C and Java, so I'm still trying to get used to Python's whitespace rules and dynamic variables, but for the most part I've been able to pick it up very quickly.

I've just learned the basics of how classes are defined, and I tried to make one myself:

Code class dog:

```
def print_info(self, dog_name, dog_weight, dog_breed):  
    print(dog_name + " is a " + dog_breed + " who weighs " + dog_weight + "lbs.")  
  
def __init__(self, name, weight, breed):  
    self.name = name  
    self.weight = weight  
    self.breed = breed  
    print_info(name, weight, breed)
```

sparky = dog("Sparky", "50", "dalmatian")

I always get this error:

Code File "c:\Users\Juice\Workspace\python\learning\python.py", line 12, in <module>

```
sparky = dog("Sparky", "50", "dalmatian")
```

File "c:\Users\Juice\Workspace\python\learning\python.py", line 10, in __init__

```
print_info(name, weight, breed)
```

NameError: name 'print_info' is not defined

And in line 10, where I try to call print_info() from the constructor, it is underlined with the tooltip:

Code E0602:Undefined variable 'print_info'

Really, how is it not defined?? print_info() should have been defined before the interpreter even reached line 10, no?

Any explanations would be much appreciated. ?

Vanagander - July 9, 2017, 1:28 pm

`__init__` is a class constructor and the `self` parameter refers to an instance of the object. `__init__` is automatically called when memory is allocated to the object (when an object of the class is created).

This might help to clear it up:

Code class MyClass(object):

```
def test(self, age):  
    print "test called age =", age  
def __init__(self, age):  
    self.age = age
```

```
self.test(age)
MyClass(12)
```

The test() has to be bound to an object upon calling it. So to fix your code put self.print_info() instead of print_info() as such:

Code class dog:

```
def print_info(self, dog_name, dog_weight, dog_breed):
    print(dog_name + " is a " + dog_breed + " who weighs " + dog_weight + "lbs.")

def __init__(self, name, weight, breed):
    self.name = name
    self.weight = weight
    self.breed = breed
    print_info(name, weight, breed)
```

```
sparky = dog("Sparky", "50", "dalmatian")
```

JuiceWater - July 9, 2017, 1:39 pm

(Jul 9, 2017 01:28 PM)Vanagander Wrote: [\[CITE\]](#) __init__ is a class constructor and the self parameter refers to an instance of the object. __init__ is automatically called when memory is allocated to the object (when an object of the class is created).

Now it makes perfect sense, self is used temporarily for each instance of an object, so if I didn't include the self parameter when calling the function I wouldn't have even been able to add more than 1 dog anyway.

Thank you, and I'm sorry for wasting everyone's time. I feel like such an idiot. ???

arkoydas - July 9, 2017, 1:41 pm

Hey mate,

I run your code in Pycharm debugger and I found out that the problem is in line:

PHP Code

```
<?php
```

```
print_info(name, weight, breed)
```

That's because your function has not read from interpreter yet.

Vanagander gave you a good solution.

Malentor - July 10, 2017, 2:40 pm

I made this code , for practice just like you :)

PHP Code

```
<?php
```

```
class Dog():
```

```
    def __init__(self,dog_name,dog_weight,dog_breed):
        self.dog_name = dog_name
        self.dog_weight = dog_weight
        self.dog_breed = dog_breed
```

```
    def show(self):
        self.dog_weight = str(self.dog_weight)
```

```
        print("Dog name is: " + self.dog_name + " Weight is: " + self.dog_weight + " breed is: " + self.dog_breed)
```

```
sparky = Dog("Sparky" , 20 , "dalmatian")  
sparky.show()
```

good luck