# [What is the difference between \_\_init\_\_ and \_\_call\_\_?](https://stackoverflow.com/questions/9663562/what-is-the-difference-between-init-and-call)

I want to know the difference between \_\_init\_\_ and \_\_call\_\_ methods.

For example:

class test:

def \_\_init\_\_(self):

self.a = 10

def \_\_call\_\_(self):

b = 20

The first is used to initialise newly created object, and receives arguments used to do that:

class Foo:

def \_\_init\_\_(self, a, b, c):

# ...

x = Foo(1, 2, 3) # \_\_init\_\_

The second implements function call operator.

class Foo:

def \_\_call\_\_(self, a, b, c):

# ...

x = Foo()

x(1, 2, 3) # \_\_call\_\_