

The vars () function in Python

The vars () function in Python returns the `__dict__` attribute of the passed object if the object has the `__dict__` attribute.

The vars () function in Python returns the `__dict__` attribute of the passed object if the object has the `__dict__` attribute .

The `__dict__` attribute is a dictionary containing the object's changeable attributes.

Note: calling `vars ()` without parameters returns a dictionary containing the local symbol table.

The syntax of vars () function in Python

```
vars(object)
```

Parameters of vars () function:

The function `vars ()` has a maximum of 1 parameter:

1. *object*: can be a module, class, instance, or any object with the `__dict__` attribute .

Value returned from vars ()

1. `Vars ()` returns the `__dict__` property of the original passed object. If the object passed to `vars ()` does not have the `__dict__` attribute, then `TypeError` will result .
2. If no arguments are passed, `vars ()` works like the `locals ()` function .

For example: How does the vars () function work?

```
class Foo:  
def __init__(self, a = 5, b = 10):  
self.a = a  
self.b = b
```

```
InstanceOfFoo = Foo()  
print(vars(InstanceOfFoo))
```

Run the program, the result is:

```
{'a': 5, 'b': 10}
```

See also: Built-in Python functions

You finished reading the article "**The vars () function in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.
