

Рефлексия в Python

Рефлексия – это способность компьютерной программы изучать и модифицировать свою структуру и поведение (значения, мета-данные, свойства и функции) во время выполнения.

В языке Python всё является объектами: переменные, функции, классы, экземпляры классов, модули. Внутреннее устройство любого объекта всегда доступно во время выполнения программы. Интерпретатор никогда не скрывает внутреннее устройство объектов, поэтому рефлексия в этом языке наиболее естественный процесс.

Функции для работы с рефлексией

– `type(object)` – возвращает тип объекта

```
o1 = type('X', (object,), dict(a='Foo', b=12))

print(type(o1))
print(vars(o1))

x = 5
s = "geeksforgeeks"
y = [1,2,3]
print(type(x))
print(type(s))
print(type(y))

class test:
    a = 'Foo'
    b = 12

o2 = type('Y', (test,), dict(a='Foo', b=12))
print(type(o2))
print(vars(o2))
```

Output:

```
class 'int'
class 'str'
class 'list'
```

Output:

```
{'b': 12, 'a': 'Foo', '__dict__': , '__doc__': None, '__weakref__': }
{'b': 12, 'a': 'Foo', '__doc__': None}
```

- `isinstance(object, classinfo)` – проверяет, является ли объект классом `classinfo` или его подклассом. `Classinfo` может быть как классом, так и кортежем классов. Если `Classinfo` ни то, ни другое, то возбуждается исключение `TypeError`.

```
class Test:
    a = 5

TestInstance = Test()

print(isinstance(TestInstance, Test))
print(isinstance(TestInstance, (list, tuple)))
print(isinstance(TestInstance, (list, tuple, Test)))
```

Output:

```
True
False
True
```

- `callable(x)` – проверяет, является ли `x` вызываемым объектом (функцией или классом с методом `__call__`)

```
class Fool:
    def __call__(self):
        print('Print Something')

print(callable(Fool))
```

Output:

```
True
```

- `dir(x)` – возвращает список всех атрибутов. Если объект `x` имеет метод `__dir__`, то `dir(x)` вызовет его, иначе информация будет получена из свойства `__dict__` или из типа объекта

```
number = [1,2,3]
print(dir(number))

characters = ["a", "b"]
print(dir(number))
```

Output:

```
>>> number = [1,2,3]
>>> print(dir(number))
['_add', '_class', '_contains', '_delattr', '_delitem', '_dir', '_doc', '_eq', '_format', '_ge', '_getattribute', '_getitem', '_gt', '_hash', '_iadd', '_imul', '_init', '_iter', '_le', '_len', '_lt', '_mul', '_ne', '_new', '_reduce', '_reduce_ex', '_repr', '_reversed', '_rmul', '_setattr', '_setitem', '_sizeof', '_str', '_subclasshook', '_append', '_clear', '_copy', '_count', '_extend', '_index', '_insert', '_pop', '_remove', '_reverse', '_sort']
>>> number = ["a", "b"]
>>> print(dir(number))
['_add', '_class', '_contains', '_delattr', '_delitem', '_dir', '_doc', '_eq', '_format', '_ge', '_getattribute', '_getitem', '_gt', '_hash', '_iadd', '_imul', '_init', '_iter', '_le', '_len', '_lt', '_mul', '_ne', '_new', '_reduce', '_reduce_ex', '_repr', '_reversed', '_rmul', '_setattr', '_setitem', '_sizeof', '_str', '_subclasshook', '_append', '_clear', '_copy', '_count', '_extend', '_index', '_insert', '_pop', '_remove', '_reverse', '_sort']
```

– `getattr(obj, attr, default)` – возвращает значение атрибута `attr` объекта `obj`. Если такого атрибута нет, то вернёт значение по умолчанию (`default`).

```
class Employee:
    salary = 25000
    company_name= "geeksforgeeks"

employee = Employee()
print('The salary is:', getattr(employee, "salary"))
print('The salary is:', employee.salary)
```

Output:

```
The salary is: 25000
The salary is: 25000
```

– `setattr(obj, attr, value)` – устанавливает новое значение атрибута `attr` объекта `obj`. Если указанного атрибута нет, то он добавляется.