

Multiple dispatch in Python

Mantas Zimnickas
VilniusPy #19

PyConLT 2019

20% discount for PyConLT tickets:

VILNIUSPY@PYCONLT

Discount code is valid until today 23:00.

What is multiple dispatch?

Multiple dispatch is a
runtime subtype-based polymorphism.

What is polymorphism?

A function that can handle arguments of different types.

Python is polymorphic

```
>>> len('a')  
1
```

```
>>> len([1])  
1
```

Function overloading

```
def len(x: str):
```

```
    . . .
```

```
def len(x: list):
```

```
    . . .
```

Polymorphism in Python

```
def len(x: Any):  
    if isinstance(x, str):  
        ...  
  
    if isinstance(x, list):  
        ...
```

Encapsulation

```
def len(x: Any):  
    return x.__len__()
```


Single dispatch

`x.__len__()`

`__len__(x)`

functools.singledispatch

```
from functools import singledispatch
```

```
@singledispatch
```

```
def len(x):
```

```
    ...
```

```
@len.register
```

```
def len(x: str):
```

```
    ...
```

multiple dispatch

$a + b$

$+(a, b)$

Addition in Python

`a + b`

`a.__add__(b)`

`b.__radd__(a)`

Without multiple dispatch

```
str_add(a, b)
int_add(a, b)
int_and_float_add(a, b)
float_and_int_add(a, b)
...
```

Multiple dispatch in language

```
go(who: Person, where: Location, how: Commute)
```

Real example

```
from datastore import prepare
```

```
@prepare.register()  
def prepare(  
    store: Store,  
    backend: Backend,  
    dtype: DataType,  
    value: str  
):
```

Multiple dispatch in 5 minutes

In 2005 Guido van Rossum wrote:

```
@multimethod(int, int)
def foo(a, b):
    ...code for two ints...

@multimethod(float, float):
def foo(a, b):
    ...code for two floats...

@multimethod(str, str):
def foo(a, b):
    ...code for two strings...
```


pip install multipledispatch

```
from multipledispatch import dispatch
```

```
@dispatch()  
def add(x: int, y: int):  
    return x + y
```

```
@dispatch()  
def add(x, y):  
    return f'{x} + {y}'
```

```
>>> add(1, 2)  
3
```

```
>>> add(1, 'hello')  
'1 + hello'
```

Thank you!