

Metatřídý v Pythonu a jejich využití

Petr Zemek

Prezentace článku do předmětu Dynamické jazyky (DJA)

21. dubna 2009



Úvod

- Co je to metatřída?
 - třída, jejíž instance jsou třídy

- K čemu metatřída může sloužit?
 - vytvoření třídy (new/init)
 - informace potřebné při vytvoření třídy, instancí této třídy a používání těchto instancí (mro)
 - změny ve třídě před jejím vytvoření
 - zajištění provedení určitých akcí při jiné akci (konstruktor)



Jak si vytvořím v Pythonu metatřídou?

```
class MyMetaclass(type):  
    def __new__(meta, name, bases, dict):  
        # Zde lze provést modifikace třídy  
        # před jejím vytvořením  
        return type.__new__(meta, name, bases, dict)
```

```
class MyClass(object):  
    __metaclass__ = MyMetaclass
```

```
>>> type(MyClass)  
<class '__main__.MyMetaclass'>
```



Jaké je možné využití metatříd v Pythonu?

- Aspektově orientované programování
 - kontrola práv, logování, profilování, ladění
- Modifikace atributů tříd
 - přidání/změna/odebrání atributů (i parametrů metod)
- Provádění akcí při vytváření tříd a jejich instancí
 - registrace tříd (pluginy)
- Mnohá další použití
 - kontrola dokumentace, databázové schéma, frameworky, ...



Závěr

Tim Peters, Python guru (`comp.lang.python`):

Metatřídý jsou mnohem hlubší magie než to, o co se bude muset 99% uživatelů kdy starat. Jestli přemýšlíte o tom, zda je potřebujete, tak odpověď je, že nepotřebujete (lidé, kteří je opravdu potřebují, s naprostou jistotou ví, že je potřebují, a nemusí tedy vysvětlovat, proč je potřebují).



Děkuji za pozornost!