

# JAK TESTOVAT 2

# modul unittest

- součást Standardní knihovny
- klon JUnit od Kenta Becka a Ericha Gammy  
<https://web.archive.org/web/20150315073817/http://www.xprogramming.com/testfram.htm>
- dříve nazývaný PyUnit  
<http://pyunit.sourceforge.net/pyunit.html>
- existují další podobné moduly

# Testovací případ

- měl by být nezávislý na okolí (všechno si připravit sám)
- podtřída třídy `unittest.TestCase`
- testy jsou funkce, jejichž jméno začíná `test...`
- volají funkce jako
  - `assert`
  - `self.assertEqual`
  - `with self.assertRaises`
- `unittest.main()` spustí všechny testy
- `setUp(self)` a `tearDown(self)`  
předtím připraví a potom uklidí

# Test Driven Development (TDD)

1. nejdříve napsat testy
2. přesvědčit se, že všechny testy **selžou**
3. pak psát kód tak dlouho, až všechny testy **projdou**

<https://code.visualstudio.com/docs/python/testing>

# Pozitivní testy x Negativní testy

- program a) dělá, co má dělat a b) nedělá, co nemá dělat
- když má program selhat, měl by selhat co nejdříve a mělo by to být slyšet
- Selhání (Failure) vs. Chyba (Error)
- výjimky vs. chybové kódy

`isinstance(s, str)`

Např.: `assert isinstance("1234", int)`

`self.assertRaises(výjimka, funkce, parametry)`

Např.: `self.assertRaises(ValueError, int, "abc" )`

# Druhy testů

jednotkové testy (unit-testy)

testy jednotlivých částí

integrační testy

jestli části programu budou spolupracovat

regresní testy

jestli program ještě funguje tak, jak fungoval včera

Funkční a nefunkční požadavky

CI: Continuous integration (Průběžná integrace)

# Poznámka

## Jak importovat modul z určité cesty

```
import sys
sys.path.append("D:/Tom") # pozor na lomítka!
import mujmodul
```