

# Anotace

- Unit-testy
- klíčové slovo `with`, načítání souborů
- context managery,
- iterable a klíčové slovo `yield`

# Unit testy

- Testujeme jednotky (typicky funkce),
- lze využít knihovny UnitTest, PyTest,... (ty první dvě podporuje Studio).
- Otevřeme adresář, do JSON souboru přidáme informace o testech, otevřeme projekt, přidáme soubor testů.
- Knihovna UnitTest definuje třídu TestCase. Testovací metody jsou ty, které začínají test...

# Technická stránka

- Open local folder, kliknout na ikonu show all files v Solution exploreru.
- Tam napastit:  
"TestFramework": "unittest",  
"UnitTestRootDirectory": "testing",  
"UnitTestPattern": "test\_\*.py".
- Jsou-li zdrojaky jinde (v src, pak i:  
"SearchPaths": [".\\src"]

- Otevřít solutionu (projekt), pravým čudlíkem nad projektem vybrat Properties,
- *Add new, Unit test...*
- Tam nainportovat moduly k testování.
- Do tříd dědicích od TestCase dělat metody začínající test\_

# with

je klíčové slovo, nad jehož fungováním strávíme delší čas

- Zahajuje blok, pro který si otevřeme tzv. kontext-manažer.
- Kontext-manažer může být reprezentován objektem osazeným metodami `__enter__` a `__exit__`, při vstupu se zavolá ta první, při výstupu (z with-bloku) ta druhá.

- Typické použití:

```
with open("soubor.txt") as f:  
    for i in f: print(i)
```

- Objekt reprezentující soubor se může chovat jako iterátor (bude za chvíli).
- Výhodou je, že v případě výjimky kontext-manažer může pustit systémové prostředky (zde deskriptor souboru).
- Takto lze zpracovávat soubory po řádcích (bude cvičeno na Linuxu).

# Iteratory

zachraňují prostorovou složitost for-cyklu

- Například funkce `range` nevrací nám dosud známé iterovatelné struktury (seznam/tuple/...), ale prvky se postupně generují.
- Jednotlivé prvky generujeme klíčovým slovem `yield`, syntakticky postupujeme jako u `return`.
- Výsledek volání takové funkce přiřadíme...
- ... a můžeme přes něj iterovat.

## Příklad

```
def fib(kolik:))
    yield 1
    mensi,vetsi=1,2
    while(kolik>=vetsi):
        yield vetsi
        vetsi,mensi=mensi+vetsi,vetsi
p10=fib(10)
print("Fibonacciho cisla do deseti jsou:")
for i in p10: print(i,end=' ')
print()
```