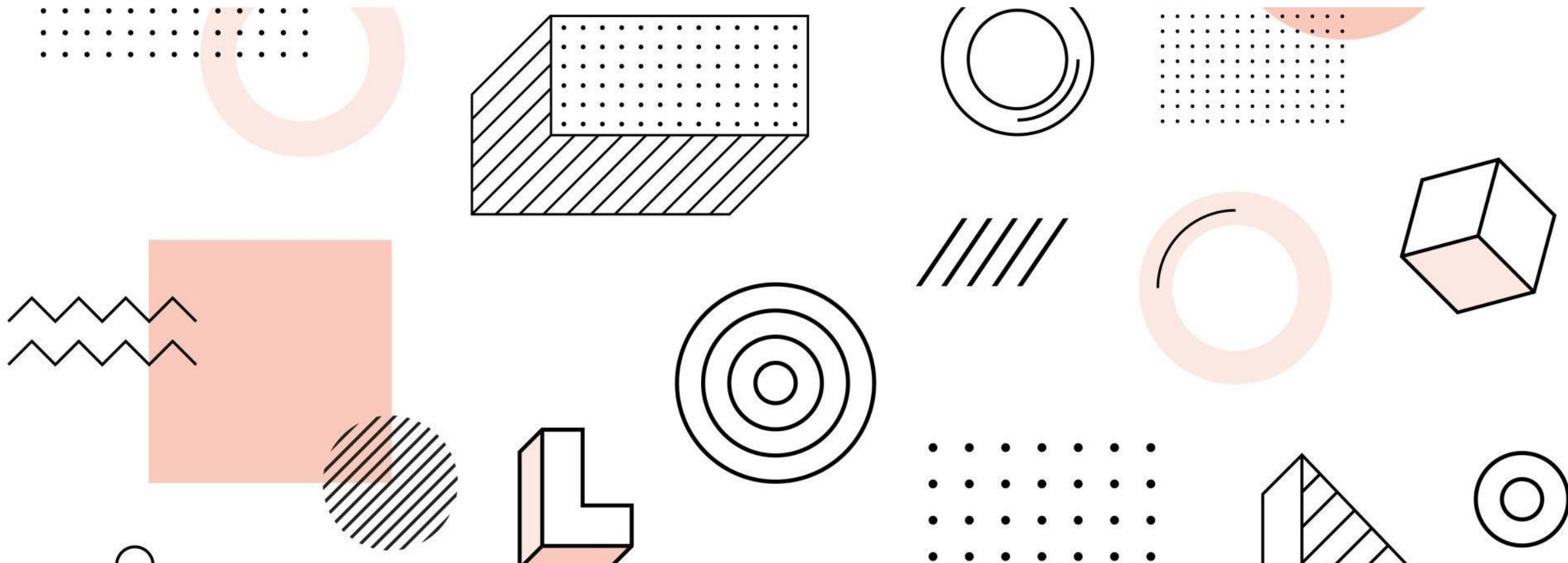


PYTHON PRO INŽENÝRSKÉ VÝPOČTY

Týden 9. Testování a ladění



Tak začneme!

Testování a ladění jsou základními kroky v procesu vývoje softwaru. Dneska se seznámíme s několika běžnými přístupy k testování:

- **Manuální testování:** V tomto případě se kód ručně spouští s různými vstupy a ověřují se očekávané výstupy. Ke kontrole mezivýsledných hodnot a identifikaci případných problémů můžete použít tiskové příkazy (print).
- **Testování jednotek:** Testování jednotek je praxe psaní automatizovaných testů pro ověření správnosti jednotlivých jednotek kódu, jako jsou funkce nebo třídy. Modul **doctest** poskytuje rámec pro vytváření a spouštění jednotkových testů.
- **Zpracování výjimek:** Pomocí bloků try-except můžete zachytávat a zpracovávat výjimky. Protokolování nebo vypisování podrobností o výjimkách může pomoci při diagnostice chyb.
- **Ladění pomocí modulu pdb:** Python má vestavěný modul pdb (Python Debugger), který umožňuje interaktivní ladění. Vložením bodů přerušení do kódu můžete pozastavit provádění v určitých bodech a interaktivně kontrolovat proměnné, procházet řádky kódu a analyzovat stav programu.
- **Integrovaná vývojová prostředí (IDE):** IDE pro Python poskytují výkonné ladicí funkce. Nabízejí funkce, jako jsou body přerušení, postupné provádění, kontrola proměnných a další.

DOCTEST

Modul **doctest** je testovací framework, který umožňuje psát testy přímo v dokumentačních řetězcích kódu (docstrings). Poskytuje způsob, jak vytvářet testy, které jsou vloženy do dokumentace kódu a mohou být automaticky prováděny a ověřovány.

```
def sum_digits(string):  
    """ Return the sum of all digits in a  
    string.  
    >>> sum_digits('Hello World! 1, 2, 3, 4')  
    10  
    """  
    sum = 0  
    for char in string:  
        if char in '0123456789':  
            sum += int(char)  
    return sum  
  
if __name__ == '__main__':  
    import doctest  
    doctest.testmod(verbose=True)
```

EXCEPTIONS

Výjimky jsou mechanismem pro zpracování a hlášení chyb, které se vyskytnou během provádění programu. Když nastane výjimečný stav, je vyvolán objekt výjimky, který lze následně zachytit a zpracovat příslušným kódem. Výjimky jsou reprezentovány třídami a každý typ chyby má svou vlastní odpovídající třídu výjimek.

```
def divide(a, b):  
    try:  
        result = a / b  
        print("Result:", result)  
    except ZeroDivisionError:  
        print("Error: Division by zero is not allowed.")  
  
divide(10, 2)      # Result: 5.0  
divide(10, 0)      # Error: Division by zero is not allowed.
```

LADĚNÍ VE VS CODE

