

PEP 8 - Průvodce stylem pro Python

Stručný výtah pro středoškoláky

Proč se starat o styl kódu?

Čitelný kód = snadnější pochopení, méně chyb, lepší spolupráce. PEP 8 je doporučený styl pro Python.

1. Pojmenování proměnných a funkcí

✅ **SPRÁVNĚ:** snake_case (malá písmena s podtržítky)

```
python

jmeno_studenta = "Jan"
pocet_bodu = 100
maximalni_delka = 50

def vypocti_prumer(znamky):
    pass

def nacti_soubor(cesta):
    pass
```

❌ **ŠPATNĚ**

```
python

JmenoStudenta = "Jan"    # CamelCase - používá se jen pro třídy
pocetBodu = 100           # camelCase - nepoužívá se v Pythonu
PočetBodů = 100           # diakritika - vyhni se jí
p = 100                   # příliš krátké, nejasný význam
```

2. Pojmenování konstant

✅ **SPRÁVNĚ:** VELKÁ_PÍSMENA_S_PODTRŽÍTKY

```
python
```

```
PI = 3.14159
```

```
MAX_POKUS = 3
```

```
NAZEV_SKOLY = "Gymnázium"
```

3. Mezery kolem operátorů

✓ SPRÁVNĚ

```
python
```

```
x = 5
```

```
y = x + 10
```

```
vysledek = (a + b) * c
```

```
seznam = [1, 2, 3, 4]
```

✗ ŠPATNĚ

```
python
```

```
x=5          # chybí mezery kolem =
```

```
y=x+10       # chybí mezery kolem operátorů
```

```
vysledek=(a+b)*c # příliš stísněné
```

```
seznam=[1,2,3,4] # chybí mezera za čárkou
```

4. Mezery po čárkách

✓ SPRÁVNĚ

```
python
```

```
seznam = [1, 2, 3, 4, 5]
```

```
funkce(a, b, c)
```

```
slovník = {"jmeno": "Jan", "vek": 16}
```

✗ ŠPATNĚ

```
python
```

```
seznam = [1,2,3,4,5]    # chybí mezery
```

```
funkce(a,b,c)           # chybí mezery
```

```
slovník = {"jmeno":"Jan"} # chybí mezery
```

5. Délka řádku

Maximálně 79 znaků na řádek (aby se kód vešel na obrazovku)

✅ SPRÁVNĚ - dlouhý řádek rozdělený

```
python

# Varianta 1: rozdělení do závorek
vysledek = (dlouha_promenna + jina_dlouha_promenna
            + dalsi_promenna)

# Varianta 2: rozdělení seznamu
seznam_jmen = [
    "Anna", "Barbora", "Cecílie",
    "Dana", "Eva", "Františka"
]
```

6. Prázdné řádky

```
python

# Dva prázdné řádky mezi funkcemi
def prvni_funkce():
    return 1

def druha_funkce():
    return 2

# Jeden prázdný řádek uvnitř funkce pro oddělení logických bloků
def slozitejsi_funkce():
    # Inicializace
    x = 0
    y = 0

    # Výpočet
    vysledek = x + y

    # Návrátová hodnota
    return vysledek
```

7. Importy

✓ SPRÁVNĚ

```
python
```

```
# Každý import na samostatném řádku
```

```
import math
```

```
import random
```

```
import sys
```

```
# Nebo skupinově z jednoho modulu
```

```
from math import sqrt, pi, cos
```

✗ ŠPATNĚ

```
python
```

```
import math, random, sys # vše na jednom řádku
```

8. Komentáře

```
python
```

```
# Komentář začíná # a mezerou
```

```
x = 5 # Inline komentář má dvě mezery před #
```

```
# Víceřádkový komentář
```

```
# vysvětlující složitější
```

```
# kód nebo logiku
```

9. Odsazení

Vždy 4 mezery (ne tabulátor!)

```
python
```

```
def funkce():
    if podminka:
        print("Vždy 4 mezery")
    for i in range(10):
        print(i)
```

10. Porovnávání s None, True, False

✓ SPRÁVNĚ

```
python

if x is None:
    pass

if hodnota is True:
    pass

if seznam: # kontrola, zda není prázdný
    pass
```

✗ ŠPATNĚ

```
python

if x == None: # používej "is", ne "=="
    pass

if hodnota == True: # zbytečné
    pass
```

🔗 Nejčastější chyby začátečníků

- ✗ pocetBodu → ✓ pocet_bodu
- ✗ x=5+3 → ✓ x = 5 + 3
- ✗ funkce(a,b,c) → ✓ funkce(a, b, c)
- ✗ Diakritika v názvech → ✓ Pouze ASCII znaky
- ✗ Příliš krátké názvy (x), (p), (tmp) → ✓ Popisné názvy

Zdroje

- **Originální PEP 8:** <https://peps.python.org/pep-0008/>
 - **pep8.org:** <https://pep8.org/> (přehlednější verze)
 - **Automatická kontrola:** nástroj `flake8` nebo `pylint`
-

Tip na závěr

"Kód se čte mnohem častěji, než se píše."

Proto je důležité psát čitelný kód! Tvoji spolužáci (a ty sám za týden) ti poděkují. 😊