

14 Main

PVA2 Programování a vývoj aplikací

 Adam Fišer



Obsah

1. Obsah
2. Co je `__name__`
3. Ukázka: hodnota `name`
4. Problém: kód běží při importu
5. Řešení: `main guard`
6. Proč funkce `main()`
7. Doporučená struktura
8. Praktický příklad: pozdrav (I/O vs logika)
9. Aplikace
10. Použití z jiného souboru
11. Časté chyby

Co je `__name__`

- Python si pro každý soubor (modul) nastaví proměnnou `__name__` .

Platí:

- přímé spuštění souboru → `__name__ = "__main__"`
- import souboru → `__name__ = "nazev_modulu"`

Ukázka: hodnota name

```
print("Hodnota __name__ je:", __name__)
```

Vyzkoušej:

- `python demo_name.py`
- `import` z jiného souboru (uvidíš jinou hodnotu)

Problém: kód běží při importu

app.py

```
1 def pozdrav(jmeno: str) → str:
2     return f"Ahoj, {jmeno}!"
3
4 print(pozdrav("Pepa")) # spustí se i při importu
```

runner.py

```
1 import app
2 print("Import hotový.")
```

➡ Import vyvolá i výpis z app.py (nechceme).

Řešení: main guard

Opravené app.py:

```
1 def pozdrav(jmeno: str) → str:  
2     return f"Ahoj, {jmeno}!"  
3  
4 if __name__ == "__main__":  
5     print(pozdrav("Pepa"))
```

Ted':

- python app.py → vypíše pozdrav
- import app → nic se samo nespustí

Proč funkce `main()`

`main()` je „dirigent“ programu:

- načte vstupy
- zavolá další funkce
- vypíše výstup
- ošetří chyby / ukončení

Výhody:

- přehlednější program
- snadnější testování
- žádné vedlejší efekty při importu

Doporučená struktura

- pomocné funkce nahoře
- main() řídí tok
- guard úplně dole

```
1  def funkce_a():
2      ...
3
4  def funkce_b():
5      ...
6
7  def main():
8      ...
9      funkce_a()
10     funkce_b()
11
12  if __name__ == "__main__":
13     main()
```

Praktický příklad: pozdrav (I/O vs logika)

```
1  def nacti_jmeno() → str:
2      return input("Zadej jméno: ").strip()
3
4  def pozdrav(jmeno: str) → str:
5      return f"Ahoj, {jmeno}!"
6
7  def main():
8      jmeno = nacti_jmeno()
9      if not jmeno:
10         print("Nezadal jsi jméno.")
11         return
12         print(pozdrav(jmeno))
13
14  if __name__ == "__main__":
15      main()
```

Aplikace

```
1 def vypocet_dph(cena: float, sazba: float = 0.21) → float:
2     return cena * (1 + sazba)
3
4 def main():
5     cena = float(input("Zadej cenu bez DPH: "))
6     print("Cena s DPH:", vypocet_dph(cena))
7
8 if __name__ == "__main__":
9     main()
```

Použití z jiného souboru

runner.py

```
1  from app import vypocet_dph
2
3  print(vypocet_dph(100))
4  print(vypocet_dph(100, 0.12))
```

➡ Importuješ funkci bez spuštění `main()` z `app.py` .

Časté chyby

Kód v globálu - tzn. mimo funkce

- spustí se při importu
- `print("Start")` # běží i při importu

Příliš mnoho logiky v `main()`

- `main()` má řídit tok, ne obsahovat vše.

Míchání I/O a logiky

- ideál: výpočty vrací hodnotu
- `main()` řeší čtení a výpis

Děkuji za pozornost

Otázky?

Repository / Prezentace