

ĎALŠIE NASTAVENIE PROSTREDIA

⚙️ Pridaj PIP do PATH

- 🔍 **Čo to znamená?** Pridáš adresu Python package manažera do default adries pre Python compiler.
- 💡 **Prečo to potrebujem?** Musíš povedať tvojmu OS, kde má hľadať PIP (package manager).
- 🔧 **Ako to urobím?** Windows key + R → napíš 'sysdm.cpl' → stlač 'OK' → 'Advanced' → 'Environment Variables' → 'System Variables' → 'Path' → 'Edit' → Pridaj adresu tvojich Python Scripts (typicky vyzerá asi takto: C:\Users\[meno]\AppData\Local\Programs\Python\Python313\Scripts)

KNIŽNICE

💡 Užitočné knižnice

NATIVE LIBRARIES (a.k.a. Built-In alebo Standard Libraries)

NÁZOV	POUŽITIE
math	základné manipulácie s číslami
random	randomizovanie dát a generovanie náhodných hodnôt
re	rýchle manipulácie s textom a pattern matching

Zisti viac o Python native library [tu](#) 🔗.

EXTERNAL LIBRARIES (a.k.a. Third-Party Libraries)

NÁZOV	POUŽITIE
numpy	pohodlnejšie operácie s číslami (vrátane pokročilých vecí)
pandas	práca s tabuľkami a databázami
matplotlib	kreslenie grafov z dát

Zisti viac o Python external libraries [tu](#) 🔗.

🔧 Ako sfunkčniť Native Libraries

- 1 Vo svojom .py súbore, napíš `import [library_name] as [alias]` a importuj knižnicu pod aliasom.
- 2 Začni používať funkcie pridanej knižnice: `[alias].[function]`.

🔧 Ako sfunkčniť External Libraries

- 1 Vo svojom VSCode terminále (ctrl + `), run `pip install [library_name]`.
- 2 Vo svojom .py súbore importuj knižnicu pod aliasom: `import [library_name] as [alias]`.
- 3 Začni používať funkcie pridanej knižnice: `[alias].[function]`.

ERRORS & DEBUGGING

💡 Časté errors a exceptions

Typ	KEDY NASTÁVA
SyntaxError	Python nedokáže skompilovať script
NameError	referuješ premennú, ktorú si ešte nepomenovala
ZeroDivisionError	niekde v kóde je delenie nulou
IndexError	iteruješ mimo hraníc
TypeError	operácia alebo funkcia dostala argument zlého typu

Zisti viac o Python built-in exceptions (a exception hierarchies!) [tu](#) 🔗.

💡 Zachytávanie chýb

🔗 print

```
...
print([výraz ktorý si chceš prekontrolovať])
...
```

🔗 try-except-else-finally

```
try:
    [tento blok bude vyskúšaný ako prvý]
except ##[opt_exception_name]:
    [tento blok bude vykonaný ak nastane vyššie uvedená výnimka]
...
##else:
    ##[tento blok bude vykonaný ak nenastanú žiadne chyby]
##finally:
    ##[tento blok bude vykonaný bez ohľadu na to, či nastanú chyby]
```

Zisti viac o try-except statements [tu](#) 🔗.

🔗 unittest

```
import unittest

[all_functions]

class TestAddFunction(unittest.TestCase):
    def [test_name](self):
        self.[assert_statement]([assert_statement_args])

if __name__ == '__main__':
    unittest.main()
```

Vráti . (t.j. bodku) ak test prešiel a F ak test neprešiel. Zisti viac o Python unittest [tu](#) 🔗. Zisti viac o assert statements [tu](#) 🔗.

**Všeobecné tipy****1**

Niekedy budeš chcieť prepísať viac takých istých inštancií naraz. Podrž kláves 'alt' a klikni na všetky inštancie, ktoré by si chcela prepísať súčasne, čím pridáš viacero kurzorov.