


Úvodní popis (docstring)

python

 Copy

 Edit

```
"""
Nejaky pokec o tom co dany skript dela
"""
```

 Sem napiš vlastní popis:

Například:

„Skript slouží k odstranění starých logovacích `.xml` souborů a prázdných podsložek starších než 28 dní. Všechny akce zapisuje do souboru `log_cleanup.txt` ve složce, kde běží.“

Importy – knihovny, které používáme

python

 Copy

 Edit

```
import os
import time
import shutil
from datetime import datetime
```

Knihovna	K čemu slouží
<code>os</code>	práce se složkami a soubory (např. cesty, procházení, mazání)
<code>time</code>	práce s časem v sekundách (timestamp)
<code>shutil</code>	mazání celých složek (např. <code>shutil.rmtree</code>)
<code>datetime</code>	formátování aktuálního data a času pro zápis do logu


Funkce `remove_old_logs_recursive()`

python

 Copy

 Edit

```
def remove_old_logs_recursive(logs_path, days_old=28, print_to_console=True):
```

 Tohle je **vlastní funkce**, která má 3 vstupní parametry:

Parametr	Význam
<code>logs_path</code>	cesta ke složce, kde se budou mazat staré logy
<code>days_old=28</code>	jak staré soubory/složky chceme mazat (ve dnech)
<code>print_to_console=True</code>	jestli chceme logy tisknout i do konzole (nejen do souboru)

Výpočet „času pro mazání“


python

 Copy

 Edit

```
now = time.time()
cutoff = now - (days_old * 86400)
```

- `now` je aktuální čas ve vteřinách od roku 1970 (tzv. unix timestamp)
- `cutoff` je hranice – všechno, co má čas změny **menší než cutoff**, bude považováno za „staré“

 `86400` = počet sekund za 1 den → vynásobením zjistíme „o kolik sekund zpátky“ máme jít

Vytvoření logovacího souboru

python

 Copy


 Edit

```
log_filename = os.path.join(logs_path, "log_cleanup.txt")
log_file = open(log_filename, "w", encoding="utf-8")
```

- Vytváříme textový soubor `log_cleanup.txt` přímo v zadané složce
- `"w"` znamená, že se soubor **přepíše** při každém spuštění
- `utf-8` zajistí, že diakritika a znaky budou fungovat správně

Zahájení mazání

python

 Copy

 Edit

```
log(f"{datetime.now()} - Start cleanup in {logs_path}")
log(f"Deleting files and subdirectories older then {days_old} da
log("-" * 60)
```

- První záznamy do logu – čas spuštění, co se bude mazat a kam
- `"log("-" * 60)"` vytvoří čáru z pomlček pro přehlednost

Procházení všech složek a souborů

python

 Copy

 Edit

```
for root, dirs, files in os.walk(logs_path, topdown=False):
```

- `os.walk()` prochází všechny složky včetně podsložek (rekurzivně)
- `topdown=False` znamená, že začneme „odspodu“ → mazání nejdřív uvnitř podsložek

Mazání `.xml` souborů

python

Copy

Edit

```
for file in files:
    file_path = os.path.join(root, file)
    if file.lower().endswith('.xml'):
        try:
            if os.path.getmtime(file_path) < cutoff:
                os.remove(file_path)
                log(f"Log file is deleted: {file_path}")
        except OSError:
            log(f"Skip unremovable file: {file_path}")
```

Krok	Co dělá
<code>file.lower().endswith('.xml')</code>	kontroluje, jestli má soubor příponu <code>.xml</code> (bez ohledu na velikost písmen)
<code>os.path.getmtime()</code>	zjistí čas poslední změny souboru
<code>os.remove()</code>	smaže soubor
<code>try/except</code>	pokud smazání selže (např. kvůli právům), zachytíme chybu a pokračujeme dál



Mazání starých podsložek

python

Copy

Edit

```
for subdir in dirs:
    subdir_path = os.path.join(root, subdir)
    try:
        if os.path.getmtime(subdir_path) < cutoff:
            shutil.rmtree(subdir_path)
            log(f"Subdirectory is deleted: {subdir_path}")
    except OSError:
        log(f"Skip unremovable subdirectory: {subdir_path}")
```

- Podobná logika jako u souborů
- Místo `os.remove()` používáme `shutil.rmtree()` → smaže celý adresář i s jeho obsahem

✓ Ukončení mazání a zápis cesty k logu

python

Copy

Edit

```
log("-" * 60)
log("Cleanup is done.")
full_log_path = os.path.abspath(log_filename)
log(f"Log file of cleanup is saved: {full_log_path}")
log_file.close()
```

- Logujeme konec
- Získáme absolutní cestu k logu (např. `C:\Users\...\log_cleanup.txt`)
- Zavřeme soubor

Spuštění skriptu

python

Copy

Edit

```
logs_path = "cesta k adresari"
remove_old_logs_recursive(logs_path)
```

- Nastavujeme, kde má skript čistit
- Poté voláme funkci