



PRAHA CODINGSCHOOL

Python další kroky

Obsluha chyb, logování událostí
Rekurze funkcí
Lektor: Martin Rosický

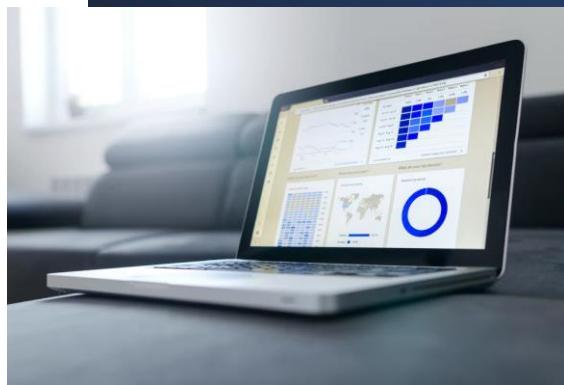


Obsluha chyb v programu

Chybové stavy

Zachycení chyby

Logování událostí





Chyby

Vývojové

- Syntaktické
 - Většinu odhalí IDE již při psaní
- Programátorské
 - Chybná logika
 - Nesprávné vyhodnocení stavu
 - Neošetřené vstupní parametry

Při běhu programu

- Chybové stavy
 - vstupních a výstupních operací
 - při komunikaci po síti
- Neočekávaný vstupní parametr



Chybové stavy

Očekávané

- Vstup / výstup
 - čtení z disku / zápis na disk
 - čtení z databáze / zápis do databáze
 - síťová komunikace
 - interakce s uživatelem
- Výsledek programové logiky
 - reakce na stav programu
 - validace vstupů
 - součást logiky programu

Neočekávané

- Cokoliv, co programátor nepředpokládal
- Způsobí pád programu

Traceback (most recent call last):
File „<input>“, line 1, in <module>
 cts_tup[2] = 'Liberec'
TypeError: 'tuple' object does not support item assignment



Logování událostí

- Zaznamenání informací o běhu programu pro umožnění zpětné analýzy
- Více úrovní
 - CRITICAL
 - ERROR
 - WARNING
 - INFO
 - DEBUG
- Různé typy výstupů
 - obrazovka
 - soubor
 - systém
- Python má propracovaný logovací systém



Zachycení chyby

```
try:  
    # operace, u které očekáváme chybu  
    os.mkdir(pth)  
  
except Exception as err:          # zachycení chyby, čím konkrétnější, tím lépe  
    # reakce na chybu  
    log.error(f"Unable to create folder '{pth}': {err}")  
    log.debug(f"Unable to create folder '{pth}'", exc_info=True)  
    raise err           # odeslání chyby na vyšší úroveň programu  
  
else:  
    # k chybě nedošlo  
    log.info(f"Created folder '{pth}'")  
  
finally:  
    # toto se provede vždy  
    log.debug(f"Folder '{pth}' check done.")
```



Rekurze

Funkce /procedura/ volá
sama sebe

Procházení struktur





Rekurze: funkce volá sama sebe

```
def collect_files(root):
    output = []
    for fname in os.listdir(root):
        fpath = os.path.join(root, fname)
        # zjistíme obsah adresáře
        if os.path.isdir(fpath):
            output.append(collect_files(fpath))
        else:
            finfo = os.stat(fpath)
            data = {
                'path': fpath,
                'name': os.path.basename(fpath),
                'split_name': os.path.splitext(os.path.basename(fpath)),
                'created': finfo.st_ctime,
                'modified': finfo.st_mtime,
                'accessed': finfo.st_atime,
                'size': finfo.st_size
            }
            output.append(data)
    return output
# informace přidáme na seznam
```