

 Python akademie - lekce 4 - 07.11.2024

▼

# 04\_01: FOR smyčka (loop)

## Zajímavé odkazy z celé lekce:

- [Coolgenerator](#)
- [For loop na Real Python](#)
- [Příručka k používání syntaxe for/else](#)
- [Dokumentace k iterování pomocí for loopu](#)
- [Příklad ohlášení continue rozebraný na stackoverflow](#)
- [Oficiální dokumentace k funkci sorted](#)
- [Python Tutor](#)

## ▼ Úvod do cyklů

Máme nachystané 4 slovníky.

```
film_1 = {
    "JMENO": "Shawshank Redemption",
    "HODNOCENI": "93/100",
    "REZISER": "Frank Darabont"
}

film_2 = {
    "JMENO": "The Godfather",
    "HODNOCENI": "92/100",
    "REZISER": "Francis Ford Coppola"
}

film_3 = {
    "JMENO": "The Dark Knight",
    "HODNOCENI": "90/100",
    "REZISER": "Christopher Nolan"
}

film_4 = {
    "JMENO": "The Prestige",
    "HODNOCENI": "85/100",
    "REZISER": "Christopher Nolan"
}
```

Ze slovníku `filmovy_slovník` chceme získat všechny klíče "REZISER" a uložit hodnoty do pomocné proměnné. Jak budeme postupovat:

```
filmovy_slovník = {
    film_1["JMENO"]: film_1,
    film_2["JMENO"]: film_2,
    film_3["JMENO"]: film_3,
    film_4["JMENO"]: film_4
}

from pprint import pprint
pprint(filmovy_slovník)
```

```

{
  'Shawshank Redemption': {
    'HODNOCENI': '93/100',
    'JMENO': 'Shawshank Redemption',
    'REZISER': 'Frank Darabont'
  },
  'The Dark Knight': {
    'HODNOCENI': '90/100',
    'JMENO': 'The Dark Knight',
    'REZISER': 'Christopher Nolan'
  },
  'The Godfather': {
    'HODNOCENI': '92/100',
    'JMENO': 'The Godfather',
    'REZISER': 'Francis Ford Coppola'
  },
  'The Prestige': {
    'HODNOCENI': '85/100',
    'JMENO': 'The Prestige',
    'REZISER': 'Christopher Nolan'
  }
}

```

1. Vytvoříme list `reziseri`,

```
reziseri = []
```

2. Namapujeme hodnoty s režiséry a vložíme do listu,

```

reziser1 = filmy_slovník["The Dark Knight"]["REZISER"]
reziser2 = filmy_slovník["The Godfather"]["REZISER"]
reziser3 = filmy_slovník["Shawshank Redemption"]["REZISER"]
reziser4 = filmy_slovník["The Prestige"]["REZISER"]

reziseri.append([reziser1, reziser2, reziser3, reziser4])

```

3. Vypíšeme obsah nového listu

```

print(reziseri)

[['Christopher Nolan', 'Francis Ford Coppola', 'Frank Darabont', 'Christopher Nolan']]

```

V takovém případě je s pomocí znalostí, které již máme, snadné najít řešení.

Jak byste si ale vedli, pokud bychom měli filmů ve slovníku 10, nebo 20 a co třeba 100?

Řešením takovéto komplikace jsou **cykly** (jinak také smyčky).

Ty nám umožní opakovaně provádět specifický kus zápisu (kódu).

## ✓ For loop (~cyklus for)

```
for pismeno in "Python Akademie":
    print(pismeno)
```

```
promenna = 'x'
```

```

P
y
t
h
o
n

A
k
a
d
e
m
i
e

```

```
promenna
```

```
'x'
```

- `for` je klíčový výraz v záhlaví (zápisu) cyklu
- `pismeno` je dočasná proměnná vytvořená v rámci cyklu a jejíž obsah se v každém kroku cyklu přepíše
- `in` klíčový výraz ukazující zdroj dočasné proměnné
- Python Akademie je tzv. *iterovatelný* údaj, tedy proměnná, kterou můžeme procházet (`str`, `list`, `set`, `dict`)
- řádek s předpisem musí být zakončený dvojtečkou
- následují odsazené instrukce, které se budou opakovat v každém kroku

## ▼ For loop s podmínkou

```
pismena = ["a", "b", "c", "d", "e", "f", "g", "h", "i"]
print(pismena)
```

```
↩ [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i' ]
```

Ze seznamu `pismena` chceme najít písmeno "c" a upozornit na jeho výskyt. Takto to vypadá jednoduše, ale představte si, že by seznam obsahovat tisíce záznamů.

```
for pismeno in pismena:
    if pismeno == "c":
        print(f"Mam hodnotu -> '{pismeno}'<-----")
    else:
        print(f"Nemam 'c', ale '{pismeno}'")
else:
    print("-" * 33, "Dokoncil jsem hledani pismena 'c'", "-" * 33, sep="\n")
```

```
↩ Nemam 'c', ale 'a'
Nemam 'c', ale 'b'
Mam hodnotu -> 'c'<-----
Nemam 'c', ale 'd'
Nemam 'c', ale 'e'
Nemam 'c', ale 'f'
Nemam 'c', ale 'g'
Nemam 'c', ale 'h'
Nemam 'c', ale 'i'
-----
Dokoncil jsem hledani pismena 'c'
-----
```

```
for pismeno in pismena:
    if pismeno == "c":
        print(f"Mam hodnotu -> '{pismeno}'<-----")
    else:
        print(f"Nemam 'c', ale '{pismeno}'")

print("-" * 33, "Dokoncil jsem hledani pismena 'c'", "-" * 33, sep="\n")
```

```
↩ Nemam 'c', ale 'a'
-----
Dokoncil jsem hledani pismena 'c'
-----
Nemam 'c', ale 'b'
-----
Dokoncil jsem hledani pismena 'c'
-----
Mam hodnotu -> 'c'<-----
-----
Dokoncil jsem hledani pismena 'c'
-----
Nemam 'c', ale 'd'
-----
Dokoncil jsem hledani pismena 'c'
-----
Nemam 'c', ale 'e'
-----
Dokoncil jsem hledani pismena 'c'
-----
Nemam 'c', ale 'f'
-----
Dokoncil jsem hledani pismena 'c'
-----
Nemam 'c', ale 'g'
-----
Dokoncil jsem hledani pismena 'c'
-----
```

```

Nemam 'c', ale 'h'
-----
Dokončil jsem hledání písmena 'c'
-----
Nemam 'c', ale 'i'
-----
Dokončil jsem hledání písmena 'c'
-----

```

*poznámka.* Pomocný `else` se přidává k cyklu `for` pokud chceme nějaký proces provést po skončení cyklu. Opatrně lze jej přeskočit.

## ▼ For loop s podmínkou

Cykly v Pythonu můžeme doplnit ohlášením, pokud potřebujeme ovlivnit průběh smyčky/cyklu.

Ohlášení	Použití
<code>break</code>	přeskočí zbytek smyčky (vč. <code>else</code> větve) a pokračuje kódem pod ní
<code>continue</code>	vrací se k definici smyčky
<code>pass</code>	tzv. <i>placeholder</i> , zabrání potenciální výjimce

## ▼ Ohlášení break

```
for pismeno in "Python Akademie":
    print(pismeno)
```



```

P
y
t
h
o
n

A
k
a
d
e
m
i
e

```

```
for pismeno in "Python Akademie":
    print(pismeno)
    break
```

```
print('ahoj')
```



```

P
ahoj

```

```
for pismeno in "Python Akademie":
    print(pismeno)
else:
    print("-" * 29, "Konec smyčky!", "-" * 29, sep="\n")
```



```

P
y
t
h
o
n

A
k
a
d
e
m
i
e
-----
Konec smyčky!
-----


```

```
for pismeno in "Python Akademie":
    print(pismeno)
    break
else:
    print("-" * 29, "Konec smycky!", "-" * 29, sep="\n")
```

 P

```
for pismeno in "Python Akademie":
    print(pismeno)
    break


print("-" * 29, "Konec smycky!", "-" * 29, sep="\n")
```

 P

```
-----
Konec smycky!
-----
```

```
for pismeno in "Python Akademie":
    if pismeno == " ":
        print(f"Nasel jsem mezeru a koncim")
        break
    else:
        print(f"Nasel jsem '{pismeno}', coz neni mezera")


print("Pokracuji s interpretaci naseho zapisu ^.^")
```



```
Nasel jsem 'P', coz neni mezera
Nasel jsem 'y', coz neni mezera
Nasel jsem 't', coz neni mezera
Nasel jsem 'h', coz neni mezera
Nasel jsem 'o', coz neni mezera
Nasel jsem 'n', coz neni mezera
Nasel jsem mezeru a koncim
Pokracuji s interpretaci naseho zapisu ^.^
```

```
for pismeno in "Python_Akademie":
    if pismeno == " ":
        print(f"Nasel jsem mezeru a koncim")
        break
    else:
        print(f"Nasel jsem '{pismeno}', coz neni mezera")


print("Pokracuji s interpretaci naseho zapisu ^.^")
```



```
Nasel jsem 'P', coz neni mezera
Nasel jsem 'y', coz neni mezera
Nasel jsem 't', coz neni mezera
Nasel jsem 'h', coz neni mezera
Nasel jsem 'o', coz neni mezera
Nasel jsem 'n', coz neni mezera
Nasel jsem '_', coz neni mezera
Nasel jsem 'A', coz neni mezera
Nasel jsem 'k', coz neni mezera
Nasel jsem 'a', coz neni mezera
Nasel jsem 'd', coz neni mezera
Nasel jsem 'e', coz neni mezera
Nasel jsem 'm', coz neni mezera
Nasel jsem 'i', coz neni mezera
Nasel jsem 'e', coz neni mezera
Pokracuji s interpretaci naseho zapisu ^.^
```

## ▼ Ohlášení continue

```
for pismeno in "Python Akademie":
    print(f"{pismeno}")
```




```
P
y
t
h
o
n


A
k
a
d
```

e  
m  
i  
e


```
for pismeno in "Python Akademie":
    print(f"{pismeno}")
    continue
```

 P  
y  
t  
h  
o  
n  
  
A  
k  
a  
d  
e  
m  
i  
e


```
for pismeno in "Python Akademie":
    print(f"{pismeno}")
    continue
else:
    print("-" * 29, "Konec smycky!", "-" * 29, sep="\n")
```

 P  
y  
t  
h  
o  
n  
  
A  
k  
a  
d  
e  
m  
i  
e  
-----  
Konec smycky!  
-----

```
for pismeno in "Python Akademie":
    if pismeno in "Python":
        continue
    print(f"{pismeno}")
```

 A  
k  
a  
d  
e  
m  
i  
e

```
for pismeno in "Python Akademie":
    if pismeno not in "Python":
        print(f"{pismeno}")
```

 A  
k  
a  
d  
e  
m  
i  
e

✓ Ohlášení pass

```
for pismeno in "Python Akademie":
    print(f"{pismeno}")
    pass
```

```
↩ P
y
t
h
o
n

A
k
a
d
e
m
i
e
```

Klíčový pojem `pass` funguje spíše jako tzv *placeholder*. Jeho účelem je zabránit případnému pádu našeho souboru a možnosti doplnit potenciální zápis později.

```
for pismeno in "Matous":
    # budu iterovat skrze promennou: str
    pass
```

## ▼ Nestovaný for

```
jmena = [
    ["Matous", "Marek", "Lukas", "Jan"],
    ["Lucie", "Aneta", "Michaela", "Lenka"],
    ["Helmut", "Hammet", "Hetfield", "Harold"]
]
```

```
for rada in jmena:
    print(f"{rada = }")
```

```
↩ rada = ['Matous', 'Marek', 'Lukas', 'Jan']
rada = ['Lucie', 'Aneta', 'Michaela', 'Lenka']
rada = ['Helmut', 'Hammet', 'Hetfield', 'Harold']
```

```
rada
```

```
↩ ['Helmut', 'Hammet', 'Hetfield', 'Harold']
```

```
for rada in jmena:
    print(f"{rada = }")
    for jmeno in rada:
        print(f"{jmeno = }")
```

```
for rada in jmena:
    print(f"{rada = }")
    for jmeno in rada:
        print(f"{jmeno = }")
        for pismenko in jmeno:
            print(pismenko)
```

```
↩ rada = ['Matous', 'Marek', 'Lukas', 'Jan']
jmeno = 'Matous'
M
a
t
o
u
s
jmeno = 'Marek'
M
a
r
e
k
jmeno = 'Lukas'
```

```

L
u
k
a
s
jmeno = 'Jan'
J
a
n
rada = ['Lucie', 'Aneta', 'Michaela', 'Lenka']
jmeno = 'Lucie'
L
u
c
i
e
jmeno = 'Aneta'
A
n
e
t
a
jmeno = 'Michaela'
M
i
c
h
a
e
l
a
jmeno = 'Lenka'
L
e
n
k
a
rada = ['Helmut', 'Hammet', 'Hetfield', 'Harold']
jmeno = 'Helmut'
H
e
l
m

```

```

for rada in jmena:
    pass
    for jmeno in rada:
        print(f"{jmeno = }")

for rada in jmena:
    for jmeno in rada:
        print(f"{jmeno = }")

for rada in jmena:
    print("-" * 40, f"{rada = }", "-" * 40, sep="\n")
    for jmeno in rada:
        print(f"{jmeno = }")

for rada in jmena:
    for jmeno in rada:
        if jmeno == "Helmut":
            print(f"Tady je, '{jmeno}'!")
            break

```

## Špatné smyčky s for

```
sekvence_tupl = ("Austin", "New York", "San Francisco")
```

```
for udaj in sekvence_tupl:
```

```
    # něco udělej..
```

```
    # udělej ještě něco..
```

```
    # ...tak počkat, co je to proměnná "udaj" ?!
```

```
    print(udaj)
```

## Jak napsat pěknou smyčku s for



```
# oprav řešení ze špatné ukázky
mesta = ("Austin", "New York", "San Francisco")

for mesto in mesta:
    # .. něco udělej..
    # .. udělej dalšího
```