Python akademie - lekce 7 - 28.11.2024

07_02: Další přiřazování

Zajímavé odkazy z této lekce:

- Zajímavé příklady vícenásobného přiřazování
- Augmented assignment operators
- Oficiální dokumentace zkráceného přiřazování

Vícenásobné přiřazení hodnot

Doposud přiřazuješ hodnoty k odkazům tímto způsobem:

```
jmeno = "Matous"
```

```
print(jmeno)
```

```
Matous
```

```
jmeno_1 = ["Matous", "Lukas"][0]
```

```
jmeno_2 = ["Matous", "Lukas"][1]
```

```
print(jmeno_1, jmeno_2, sep="\n")
```

```
Matous
Lukas
```

Současně ale existují i **další varianty přiřazení** hodnoty/hodnot.

Vícenásobné přiřazení (LS = PS)

Pokud máš na pravé straně (PS) více hodnot, můžeš je rozdělit.

Rozdělení probíhá následovně:

```
jmeno_1, jmeno_2 = ["Matous", "Jan"]  # LS: 2 nazvy promennych = PS: 2 udaje
```

```
print(jmeno_1, jmeno_2, sep="\n")
```

```
Matous
Jan
```

```
jmeno_3, jmeno_4 = "Matous", "Jan"
```

```
jmeno_3
```

```
'Matous'
```

Přiřazování při vrácených hodnotách:

```
def pomocna_f():
    return (11, "20")
```

```
cislo, string = pomocna_f()
```

```
print(cislo, string, sep="\n")
```

```
11
20
```

Hodnot a proměnných může být samozřejmě více.

Zásádní je dodržet s tímto zápisem pravidlo, kolik hodnot, tolik proměnných.

```
jmeno_1, jmeno_2 = ["Matous", "Lukas", "Petr"]
```

```
jmeno_1, jmeno_2, jmeno_3 = ["Matous", "Lukas", '']
```

```
jmeno_1, jmeno_2, jmeno_3 = ["Matous", "Lukas"]
```

Vícenásobné přiřazení s hvězdičkou (*)

Syntaxe je velice podobná té předchozí.

Nicméně doplněná hvězdička má za účel sbalit všechny zbývající hodnoty do jedinné proměnné.

```
jmeno_1, jmeno_2, *zbytek_jmen = ["Matous", "Marek", "Lukas", "Jan"]
```

```
print(jmeno_1, jmeno_2, zbytek_jmen, sep="\n")
```

```
Matous
Marek
['Lukas', 'Jan']
```

Všimněte si jak se hodnoty v proměnných změní, pokud změníme pořadí, kdy hvězdičku zapíšeme:

```
jmeno_1, *zbytek_jmen, jmeno_2 = ["Matous", "Marek", "Lukas", "Jan"]
```

```
print(jmeno_1, zbytek_jmen, jmeno_2, sep="\n")
```

```
Matous
['Marek', 'Lukas']
Jan
```

```
jmeno_1, *zbytek_jmen, jmeno_2, jmeno_3 = ["Matous", "Marek", "Lukas", "Jan", "Petr", "Kr
```

```
print(jmeno_1, jmeno_2, jmeno_3, zbytek_jmen, sep="\n")
```

```
Matous
Petr
Krystof
['Marek', 'Lukas', 'Jan']
```

Zkrácené přiřazování (~augmented assignment)

```
x = 10
```

```
x = x + 5
```

```
x
```

V klasickém zápisu musí python udělat několik kroků:

1. zavolat proměnnou `x`
2. přičíst k ní pětku

```
a = 10
```

```
a += 5
```

```
a
```

Ve zkráceném přiřazení python udělá oba kroky v jeden moment, což je výpočetně výhodnější, ikdyž počet řádků kódu to nesníží.