

 Python akademie - lekce 2 - DD.MM.YYYY



02_02: Podmínky

✓ Zajímavé odkazy z této lekce:

- [Oficiální dokumentace](#)
- [If/Else of W3Schools](#)
- [Real Python](#)

Rozhodování je proces, který budeš potřebovat pokud budeš chtít, aby **tvůj zápis** uměl sám vyvodit závěr.

V Pythonu můžeš rozhodování zapsat pomocí tzv. **podmínkového zápisu**.

✓ Předpis `if`

syntaxe

```
if <podmínka>:  
    <kód>  
    <kód>
```

Jednoduchý podmínkový zápis (předpis podmínkové větve s `if`) zapisujeme následovně:

1. Klíčové slovo `if`,
2. podmínka (představme si `bool(...)`),
3. předpisový řádek ukončený dvojtečkou,
4. odsazený následující řádek.

```
if 1 == 1:  
    print('1 je stejná jako 1')
```

```
if 'jablko' == 'jablko':  
    print(1 + 1)  
    print('Dam si radši pomeranc')
```

Python se podívá na **podmínku** (příp. výraz) a pokud ji vyhodnotí **jako pravdivou** (`True`), **provede odsazený řádek** pod předpisem.

Pokud ji vyhodnotí **jako nepravdivou** (`False`), **přeskočí odsazený řádek** pod předpisem.

```
if True:  
    print("Ahoj, studenti!")  
print("Pokracuji v kodu pod podminkou...")
```

►  Klikni zde pro vysvětlení

```
if False:  
    print("Ahoj, Matousi!")  
print("Pokracuji v kodu pod podminkou...")
```

✓ Předpis `if / else`

Velmi často se dostaneš do situace, kdy potřebuješ napsat dvojí průběh (*~control flow*) tvého programu.

Například:

1. *Bud' je uživatel registrovaný, může přidat obrázek,*
2. *nebo uživatel není registrovaný a nemůže.*

nebo:

1. *Bud' je uživatel starší 18 let, může pokračovat,*
2. *nebo není starší 18 let a nemůže.*

Tento mluvený předpis *bud'/nebo* můžeme v Pythonu zapsat jako `if / else`.

```
registrovani = True

if registrovani:
    print("Přidávám obrázek")
else:
    print("Neregistrovaní uživatelé nemohou přidat obrázky")

registrovani = False

if registrovani:
    print("Přidávám obrázek")
else:
    print("Neregistrovaní uživatelé nemohou přidat obrázky")
```

Pro předpis s `if` (nebo také *podmínkovou větev*) vždycky platí kontrola podmínky.

Pokud bude podmínka vyhodnocená jako nepravdivá (`False`), Python provede odsazený zápis v `else` (tady se žádná podmínka nepíše!)

✓ Předpis `if / elif / else`

Pokud je potřeba zapsat podmínku **rozvětvenější** (nestačí dvě možnosti), můžeme podmínkový zápis rozšířit další větví s klíčovým slovem `elif` a výrazem:

```
JMENO = "Radim"

if JMENO == "Radim":
    print("Ahoj, Radime!")
elif JMENO == "Lukas":
    print("Ahoj, Lukasi!")
else:
    print("Ahoj, vsem ostatnim!")
```

```
JMENO = "Lukas"

if JMENO == "Radim":
    print("Ahoj, Radime!")
elif JMENO == "Lukas":
    print("Ahoj, Lukasi!")
else:
    print("Ahoj, vsem ostatnim!")
```

```
JMENO = "Petr"

if JMENO == "Radim":
    print("Ahoj, Radime!")
elif JMENO == "Lukas":
    print("Ahoj, Lukasi!")
else:
    print("Ahoj, vsem ostatnim!")
```

Pro předpis s `if` (nebo také *podmínkovou větev*) vždycky platí kontrola podmínky.

Pokud bude podmínka v předpisu s `if` vyhodnocená jako nepravdivá, pokračuj k podmínce v předpisu `elif`. Pokud je pravdivá, proved' odsazený zápis u `elif`.

Pokud bude podmínka vyhodnocená jako nepravdivá (`False`), Python provede odsazený zápis v `else` (tady se žádná podmínka nepíše!)

✓ Více výrazů v jednom záhlaví podmínky

Pomocí logických operátorů můžete tvořit ještě striktnější podmínkové větve, které vám umožní specifičtější průběh:

```
JMENO = "Lukas"
JE_ZDRAVY = True

if JMENO == "Matous" and JE_ZDRAVY:
    ZPRAVA = "Ahoj, Matousi! Tak at te zdravi neopousti!"

elif JMENO == "Lukas" and JE_ZDRAVY:
    ZPRAVA = "Ahoj, Lukasi! Tak at te zdravi neopousti!"

elif JMENO == "Matous" and not JE_ZDRAVY:
    ZPRAVA = "Ahoj, Matousi! Hlavne se brzy uzdrav!"

elif JMENO == "Lukas" and not JE_ZDRAVY:
    ZPRAVA = "Ahoj, Lukasi! Hlavne se brzy uzdrav!"

else:
    ZPRAVA = "Ahoj, vsem ostatnim!"

print(ZPRAVA)
```

✓ Ternární operátor

V Pythonu 2.5 byl přidán jednořádkový zápis pro jednoduchou podmínku `if/else`. Jde o pokročilejší variantu zápisu takové podmínky:

```
X if Y else Z
```

- **X**, proved' pokud `True`
- **Y**, podmínka
- **Z**, proved' pokud `False`

```
ridicak = True

if ridicak:
    print('Muzu ridit')
else:
    print('Nemuzu ridit')

ridicak = True

print("Muzu ridit" if ridicak else "Nemuzu ridit")
```

```
rid_opravneni = False

print("Muzu ridit" if rid_opravneni else "Nemuzu ridit")
```

Pokud je výraz za klíčovým slovem `if` vyhodnocený jako **pravdivý** (`True`), proved' `x` (resp. část zápisu před podmínkou).

V opačném případě proved' `y`, tedy část zápisu za `else`.

✓ Netradiční podoba

Dokonce je možné na vyhodnocení výrazu použít celočíselnou hodnotu booleanu.

```
rid_opravneni = True

status = ["Nemuzu ridit", "Muzu ridit"][rid_opravneni] # True -> 1, False -> 0

print(status)

rid_opravneni = False

status = ["Nemuzu ridit", "Muzu ridit"][rid_opravneni] # True -> 1, False -> 0

print(status)
```

Druhá hranatá závorka představuje indexovou část, díky které vybereme hodnotu v první závorce.

Nezapomeňme, že `True == 1` a `False == 0`.

Tato varianta je ale velice **nečitelná**, a proto se s ní ve skutečném zdrojovém kódu setkáte jen zřídka.