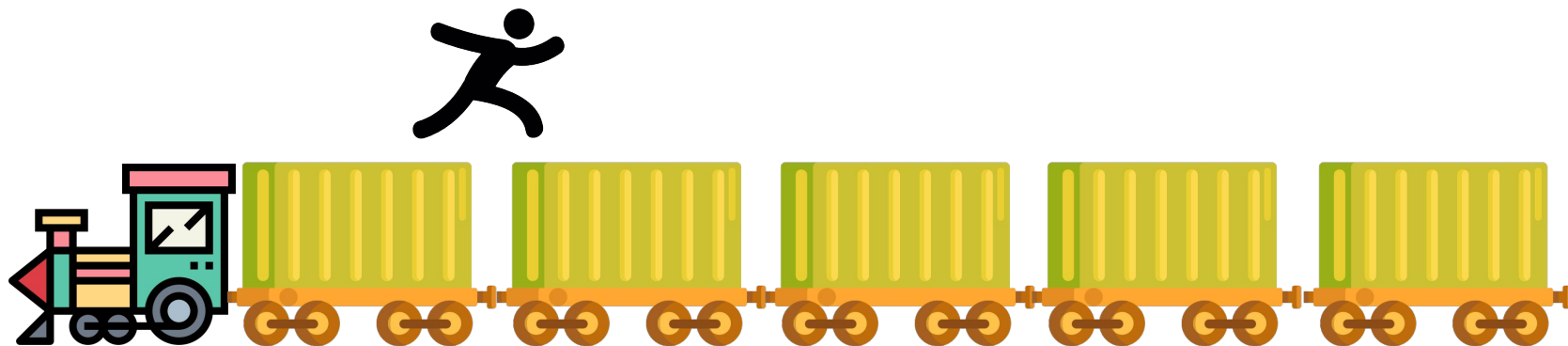


ZÁKLADY PROGRAMOVÁNÍ V JAZYCE PYTHON

LEKCE 4 - CYKLUS FOR, PODMÍNKY

CYKLUS FOR

for lze volně přeložit jako **pro** (každý prvek něco udělej)



CYKLUS FOR - SYNTAX

`in` tady není operátor `in`



for prvek **in** sekvence:

Indentace → **vykonej_neco**

Mimo cyklu bez indentace → **kod_po_cyklu**

"indentace" by správně byla "odsazení". Cizí výraz používáme proto, abyste si rozuměli s počítačem, až Vám nahlásí chybu.

CYKLUS FOR - JENOM OPAKOVÁNÍ KÓDU

```
a = [ '1', '2', '3' ]  
k = 0  
print(a[k])  
k = k + 1  
print(a[k])  
k = k + 1  
print(a[k])
```



```
a = [ '1', '2', '3' ]  
for i in a:  
    print(i)
```

MOŽNOSTI ITERACE

```
les = ["lípa", "topol", "bříza", "metasekvoje", "borovice"]
```

```
for stromek in les:  
    print(stromek)
```

```
cislo = 1
```

```
for stromek in les:  
    print("Strom číslo " + str(cislo) + " je " + stromek)  
    cislo += 1
```

MOŽNOSTI ITERACE - FUNKCE RANGE

```
cislo = 1
for stromek in les:
    print("Strom číslo " + str(cislo) + " je " + stromek)
    cislo = cislo + 1
```

Trochu kostrbaté



```
for i in range(len(les)):
    print("Strom číslo " + str(i+1) + " je " + les[i])
```

MĚSÍCE - ZADÁNÍ VE STUDIJNÍCH MATERIÁLECH

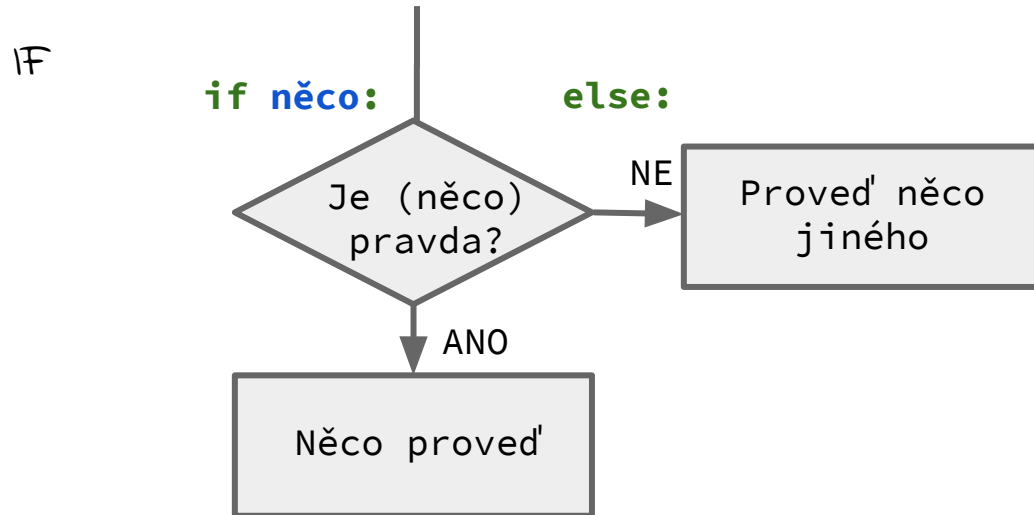


I GET STUCK IN THIS LOOP EVERY MONTH.

Zdroj [xkcd](#)

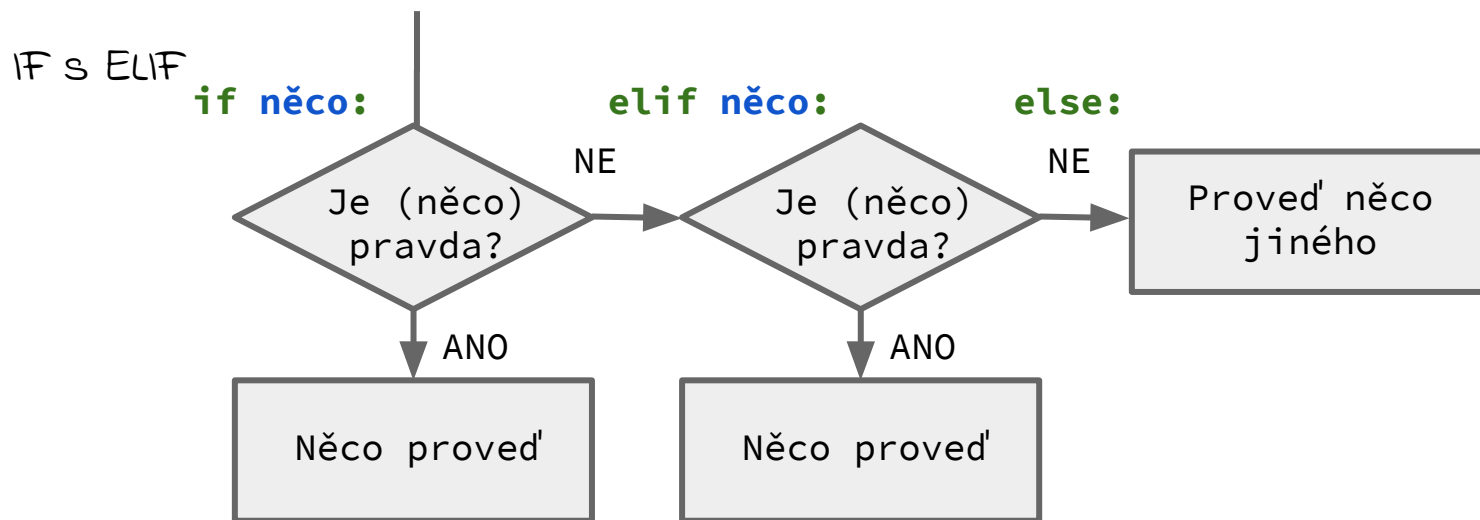
PODMÍNKY

Podmínky jsou dalším ze stavebních kamenů programování.
Pracují na principu jestli něco, tak něco, nebo něco jiného.



PODMÍNKY

Podmínky jsou dalším ze stavebních kamenů programování.
Pracují na principu jestli něco, tak něco, nebo něco jiného.



SYNTAX IF

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

```
elif <podmínka> :  
    <kód>  
    <kód>
```

```
else :  
    <kód>  
    <kód>
```

```
<jiný kód>
```

Podobně jako u cyklu `for` a `while` i zde používáme odsazení pro odlišení vnitřního a vnějšího kódu.

PŘÍKLAD

```
a = 5
```

```
b = 10
```

```
if a > b:
```

```
    print('Číslo ',a,' je větší než ',b)
```

```
elif a == b:
```

```
    print('Číslo ',a,' je rovno ',b)
```

```
else:
```

```
    print('Číslo ',b,' je větší než ',a)
```

CVIČENÍ

Pomocí kombinace **for** cyklu a **if** podmínky vypište všechna čísla z listu `l`:

```
l = ['ponožka', 154, 'Dobby', print, 82, True, 0, -43]
```

```
for i in l:
```

```
....
```

Nápoveda: vzpomeňte si na funkci `type`

CVIČENÍ

Pomocí kombinace **for** cyklu a **if** podmínky vypište všechna čísla (i čísla s desetinnou tečkou) z listu `l`.

Využijte logický operátor (**and**, **or**).

```
l = ['ATOM', 1, 'N', 'PRO', 'A', 1, 8.316, 21.206,  
     21.530, 1.00, 17.44, 'N']
```

POUŽITÍ IF V CYKLECH - CONTINUE, BREAK

continue - přeskočí zbytek kódu v iteraci cyklu

break - vyskočí z cyklu

```
for i in range(100):
    if i > 50:
        break
    if i > 10:
        continue
    if i%2 == 0:
        print('číslo ',i, ' je sudé.')
    if i%2 == 1:
        print('číslo ',i, ' je liché.')
print('Poslední číslo bylo ', i)
```

POUŽITÍ IF V CYKLECH - CONTINUE, BREAK

continue - přeskočí zbytek kódu v iteraci cyklu

break - vyskočí z cyklu

```
i = 0
while True:
    i += 1
    if i > 50:
        break
    if i > 10:
        continue
    if i%2 == 0:
        print('číslo ',i, ' je sudé.')
    if i%2 == 1:
        print('číslo ',i, ' je liché.')

print('Poslední číslo bylo ', i)
```

POUŽITÍ IF V CYKLECH - CONTINUE, BREAK

continue - přeskočí zbytek kódu v iteraci cyklu

break - vyskočí z cyklu

```
for i in range(100):
    if i > 50:
        break
    if i > 10:
        continue
    if i%2 == 0:
        print('číslo ',i, ' je sudé.')
    if i%2 == 1:
        print('číslo ',i, ' je liché.')
print('Poslední číslo bylo ', i)
```

```
i = 0
while True:
    i += 1
    if i > 50:
        break
    if i > 10:
        continue
    if i%2 == 0:
        print('číslo ',i, ' je sudé.')
    if i%2 == 1:
        print('číslo ',i, ' je liché.')
print('Poslední číslo bylo ', i)
```


NÁKUPY 1.0 A 2.0 - ZADÁNÍ VE STUDIJNÍCH MATERIÁLECH

