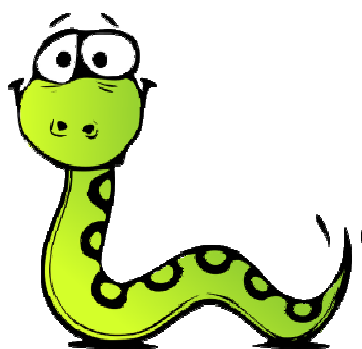




# Programování v jazyce Python pro střední školy

## Lekce 17

### Větvení a konstrukce



Andrej Blaho

Ľubomír Salanci

Václav Šimandl

1. Kamarádku pozdravíš neformálně „Ahoj“, ale starší lidi pozdravíš formálněji, například „Dobrý den“. Napiš program `pozdravy_podle_veku.py`, ve kterém do proměnné `vek` přiřadíš věk člověka. Potom použij příkaz větvení na to, aby se program podle věku rozhodl, který z uvedených dvou pozdravů vypíše. Otestuj, jaké pozdravy se vypisují pro různé hodnoty proměnné `vek`.

2. Na brigádě ve stánku se zmrzlinou dostaneš mzdu podle následujícího pravidla:

- když budeš pracovat méně než 10 hodin, vyděláš si 80 korun za hodinu,
- jinak si vyděláš 100 korun za hodinu.

Vytvoř nový program, ve kterém do proměnné `hodin` přiřadíš počet hodin, které jsi odpracoval. Potom pomocí příkazu větvení vypiš, kolik si vyděláš. Program by měl vypsát:

Vyděláš si 560 korun.  
pro `hodin = 7`

Vyděláš si 1200 korun.  
pro `hodin = 12`

3. Předchozí úloha se dá řešit i takto:

```
hodin = 20
if .....:
    mzda = .....
else:
    mzda = .....
print('Vyděláš si', mzda, 'korun.')
```

Doplň namísto vytečkovaných částí správné výrazy. Ověř, že program správně počítá mzdu pro různé hodnoty proměnné `hodin`.

4. Mobilní operátor *Vegafon* počítá platby za přenesená data podle následujících pravidel:

- když za den přeseš méně než 10 megabajtů dat, zaplatíš za každý megabajt 2 koruny,
- jinak zaplatíš za celý den 20 korun.

Napiš program `mobilni_data.py`, ve kterém do proměnné `megabajty` přiřadíš počet přenesených megabajtů dat za jeden den. Použij příkaz větvení na to, abys do proměnné `cena` přiřadil vyúčtovanou cenu. Nakonec tuto cenu vypiš. Výpis může vypadat například takto:

Zaplatíš 12 korun.  
pro `megabajty = 6`

Zaplatíš 20 korun.  
pro `megabajty = 20`

5. Mobilní operátor *Zodrafon* počítá platby za přenesená data podle odlišných pravidel:

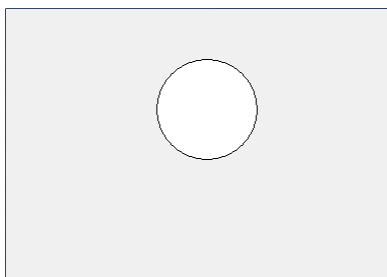
- když za den přenešes méně než 10 megabajtů, zaplatíš za každý megabajt 1 korunu,
- jinak zaplatíš 10 korun a k tomu za každý megabajt nad limit 10 megabajtů 3 koruny.

Napiš program `mobilni_data2.py`, který počítá a vypisuje cenu podle těchto pravidel, a ověř, zda funguje správně. Program by měl například vypsát:

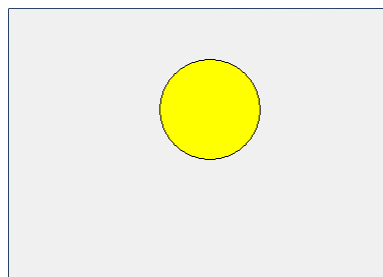
```
Zaplatíš 6 korun.  
pro megabajty = 6
```

```
Zaplatíš 40 korun.  
pro megabajty = 20
```

6. Vytvoř program `den_noc.py`, který podle zadaného času nakreslí do grafické plochy slunce nebo měsíc. Do proměnné `cas` přiřaď počet hodin. Použij příkaz větvení na to, aby se pro `cas < 8` kreslil měsíc jako bílý kruh, jinak se kreslilo slunce jako žlutý kruh. Poloměr kruhu nechť je v obou případech 50 a střed kruhu má souřadnice [200, 100]. Program by měl například nakreslit:

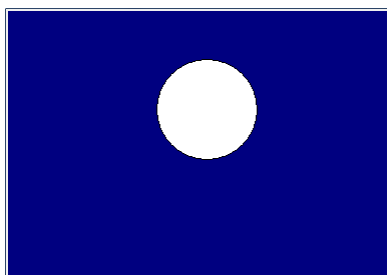


pro `cas = 4`

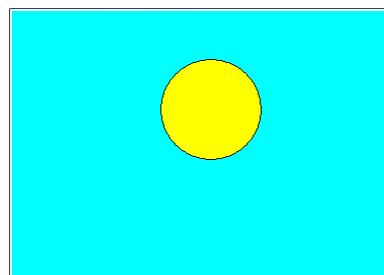


pro `cas = 14`

7. Do předchozího řešení doplň kreslení pozadí – měsíc se nakreslí na tmavomodré pozadí, slunce na světlemodré pozadí:



pro `cas = 4`



pro `cas = 14`

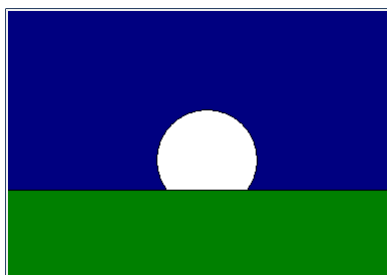
Návod, jak to udělat, najdeš na další straně.

Stačí, když do každé větve přidáš příkaz na kreslení velkého obdélníku, který překryje celou grafickou plochu:

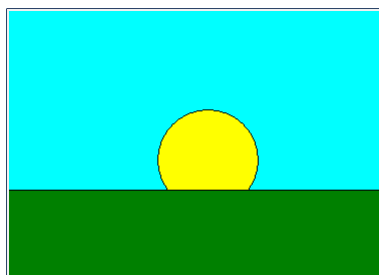
```
if .....:
    canvas.create_rectangle(....., fill='navy')
    canvas.create_oval(....., fill='white')
else:
    canvas.create_rectangle(....., fill='cyan')
    canvas.create_oval(....., fill='yellow')
```

Příkazy, které tvoří tělo větví `if` i `else`, nech odsazené od kraje (Python tam automaticky vložil 4 mezery)

8. Uprav předchozí program tak, aby se nejdříve do proměnných `x`, `y` přiřadily souřadnice středu kruhu a ty se potom použily v příkazech `create_oval`. Kromě toho přidej na úplný konec programu i kreslení zeleného obdélníku, který bude představovat krajinu. Potom program pro `x = 200` a `y = 150` bude kreslit scény jako na následujících obrázcích:



pro cas = 4



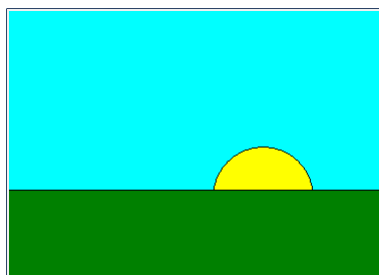
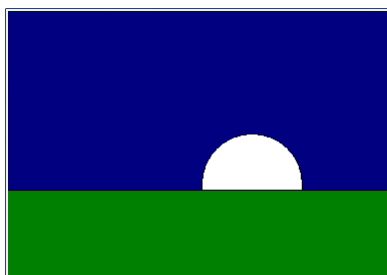
pro cas = 14

Horní okraj zeleného obdélníku umísti na y-ovou souřadnici 180.

9. Vylepši předchozí program tak, aby fungoval jako náhodný generátor krajinek – přiřaď na začátku do proměnných `x`, `y`, `cas` náhodná čísla:

- `x` z rozsahu 100, 300
- `y` z rozsahu 100, 200
- `cas` z rozsahu 0, 16

Program několikrát spust' a sleduj, zda se krajinky vytváří dle tvého očekávání:



10. Následující program vypisuje denní harmonogram:

```
for i in range(8):
    print(i, 'ještě spím')
for i in range(6):
    print(8 + i, 'jsem ve škole')
```

Diskutuj se spolužákem, co konkrétně program vypíše. Poté program spust' v Pythonu a zkontroluj, zda byla Tvá domněnka správná.

Předchozí program se dá zapsat i takto, jen pomocí jediného cyklu:

```
for i in range(14):
    if i < 8:
        print(i, 'ještě spím')
    else:
        print(i, 'jsem ve škole')
```

} tělo cyklu – je potřeba jej odsadit od kraje

tyto příkazy je potřeba odsadit od kraje ještě více

Tělo cyklu `for` obsahuje jeden **vnořený** příkaz větvení `if else`. Vyzkoušej, jak funguje tato verze.

11. Vytvoř nový program `chudy_bohaty.py`. V něm podobně jako v předchozí úloze použij cyklus s vnořeným větvením a vypiš:

```
Mám 0 korun, jsem chudý
Mám 10 korun, jsem chudý
Mám 20 korun, jsem chudý
Mám 30 korun, jsem chudý
Mám 40 korun, jsem chudý
Mám 50 korun, jsem bohatý
Mám 60 korun, jsem bohatý
Mám 70 korun, jsem bohatý
Mám 80 korun, jsem bohatý
Mám 90 korun, jsem bohatý
```

Proměnná cyklu se mění od 0 do 9 a vypisuje se desetinásobek hodnoty této proměnné.

12. Víš, pro která čísla  $n$  platí, že  $n^2$  je menší než  $5 * n$ ? Napiš program `nasobky_peti.py`, který pro všechna čísla od 0 do 10 otestuje tento vztah a vypíše o tom patřičnou informaci, například:

```
...
16 je menší než 20
25 je větší nebo rovno 25
36 je větší nebo rovno 30
...
```

Co tvůj program zjistil? Pro která  $n$  onen vztah platí?

13. Vytvoř nový program `koralky_na_niti.py`, ve kterém pomocí jediného cyklu s vnořeným větvením nakresli 15 korálků jako na obrázku níže. Prvních 8 korálků bude červených a zbylých 7 modrých.



- 14\* Uprav předchozí program tak, aby se prvních 8 korálků barvilo náhodně pomocí

```
random.choice(['red', 'yellow'])
```

a zbylých 7 pomocí

```
random.choice(['blue', 'green']).
```

Výsledek může vypadat například jako na následujícím obrázku:



- 15\* Vytvoř nový program `rozsypane_koralky.py` a v něm definuj podprogram `koralek`. V tomto podprogramu vygeneruj náhodné pozice `x`, `y` pro střed korálku. Program potom podle x-ové souřadnice nakreslí:

- červený nebo žlutý korálek, pokud je  $x < 200$ ;
- modrý nebo zelený korálek pro ostatní čísla.

Využij kreslení korálků z předchozí úlohy.

Když zavoláš podprogram `koralek` v cyklu stokrát, můžeš dostat například takovýto obrázek:

