

# abcPython - Pracovné listy pre Python¶

Pracovné listy pre programovanie v Pythone pre stredné školy. Listy sú určené pre žiakov pričom nepredpokladáme žiadny výklad učiteľa. Tento materiál vznikol vďaka českému projektu [Podpora rozvíjení infromatického myšlení](#).

Zdraví ťa Python

## Pre učiteľov

K dispozícii sú aj metodické materiály ku všetkým vyučovacím hodinám.

© Copyright 2019, Andrej Blaho, Ľubomír Salanci.  
Naposledy aktualizované 10. sep. 2020.

Prispôsobenia pre nevidiacich vykonala Ľudmila Jašková, august 2023.

Vytvorené pomocou [Sphinx](#) 3.2.1.

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

## Prispôsobenia pre nevidiacich

Prispôsobili sme materiál, ktorý sa nachádza na stránke <https://abcpython.input.sk/>

Vynechali sme úlohy, v ktorých sa vyžaduje práca s grafikou. Vynechali sme aj hodiny, na ktorých sa pracuje iba s grafikou. Ide o hodiny 5, 6, 7, 10, 14, 15, 20.

Originálny materiál používa prostredie Idle Python, ktoré nie je prístupné pre nevidiacich. Pri riešení niektorých úloh odporúčame používať **Python konzolu** čítača NVDA a niektorých úlohách odporúčame používať prostredie **Visual Studio Code** alebo online prostredie **Google Colaboratory**.

# Vyučovacie hodiny:

<b>1. hodina - Výrazy</b>	<b>3</b>
<b>2. hodina - Premenné</b>	<b>6</b>
<b>3. hodina - Program</b>	<b>10</b>
<b>4. hodina - Výpisy</b>	<b>13</b>
<b>8. hodina - Podprogramy</b>	<b>15</b>
<b>9. hodina - Náhoda</b>	<b>19</b>
<b>11. hodina - Program s opakovaním</b>	<b>22</b>
<b>12. hodina - Premenná cyklu</b>	<b>25</b>
<b>13. hodina - Výrazy v cykle</b>	<b>27</b>
<b>16. hodina - Vetvenie</b>	<b>29</b>
<b>17. hodina - Vetvenie a konštrukcie</b>	<b>32</b>
<b>18. hodina - Vnorené vetvenie</b>	<b>35</b>
<b>19. hodina - Podprogram s parametrom</b>	<b>39</b>

# 1. hodina - Výrazy

1. V NVDA ponuke sa nastav na podponuku **Nástroje**, v nej najdi **Python konzolu** a spusti ju.

Otvorí sa okno s dvomi editačnými panelmi, medzi ktorými sa môžeš presúvať klávesom **F6**.

Automaticky sa nastavíš na editačný panel, do ktorého môžeš písať. Za `>>>` budeš písať príkazy.

2. Skús za `>>>` napísať matematický výraz  $1 + 2 + 3$  a stlač kláves **Enter**.

Výsledok sa zobrazí v druhom editačnom paneli, ktorý je prístupný iba na čítanie.

3. Python dokáže fungovať ako kalkulačka. Vyskúšaj aj takéto výrazy:

```
>>> 123
>>> 123
>>> 42 - 17
>>> 3 + 4 * 5
>>> (3 + 4) * 5
>>> 25 - 7 - 10
>>> 25 - (7 - 10)
>>> 132 / 11
>>> 1 / 2
>>> 1 + 2 * 3 / (5 - 1)
```

4. Pozor, zápisy musia byť napísané úplne správne. Inak uvidíš rôzne chybové správy. Vyskúšaj napríklad:

```
>>> 22 + 7 *
>>> 19 - (3 4)
```

5. Niekedy aj po dobrom zápise uvidíš chybové hlásenie. Skús napríklad  $10/(6-2*3)$ .

Python sa ti chybovými hláseniami snaží pomôcť, aby si chybu ľahšie našiel.

Napríklad:

- `SyntaxError: invalid syntax` označuje, že niečo si zle napísal
- `ZeroDivisionError: division by zero` oznamuje, že chceš deliť nulou

6. Miro mal pred dvomi mesiacmi 16 rokov. Využi Python ako kalkulačku a spočítaj koľko je to približne dní. Predpokladaj, že rok má 365 dní a mesiac má 30 dní.
  
7. Pokračuj v predchádzajúcej úlohe a pomocou Pythonu vypočítaj:
  - a. koľko je to hodín,
  - b. koľko sekúnd.
  
8. Vytvor pre Python zápis, ktorým vypočíta súčet všetkých nepárnych čísel od 1 do 19.
  
9. Zisti, ktorá cifra sa vyskytuje najčastejšie vo výsledku výrazu:  $123456789 * 11111111111111111111$ .
  
10. Marienka zbierala kvety. V jeden deň natrhala 15, druhý deň natrhala o 4 viac, ako v predchádzajúci a na tretí ešte o 1 viac, ako v oba predchádzajúce dni dokopy. Nechaj Python vypočítať, koľko kvetov natrhala za všetky 3 dni dohromady.
  
11. Janko si kúpil hru za 79 korún. Neskôr si kúpil hru za dvojnásobok tejto sumy a ešte musel priplatiť 5 korún. Nakoniec si kúpil hru za trojnásobok ceny poslednej hry a ešte k tomu priplatil 17 korún. Vymysli zápis pre Python, ktorým vypočíta:
  - a. koľko Janko zaplatil za poslednú hru
  - b. koľko Janko zaplatil za všetky hry dohromady
  - c. (\*) skús vymyslieť najkratší zápis, ktorý to vypočíta
  
12. Vypočítajte súčet takýchto čísel: jedna, jedna polovica, jedna tretina, jedna štvrtina, ..., až jedna desatina.
  
13. Zapiš si do tabuľky všetky aritmetické operácie, s ktorými si sa zatiaľ zoznámil.

14. (\*) Výpočet  $2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2$  je umocnenie 2 na 10. V Pythone: `2 ** 10`. Vypočítaj 2 umocnené na 30 a zisti počet cifier tohto výsledku. Cifry spočítaj ručne.
15. (\*) Zisti ako sa počíta hodnota  $2^{8-1}$ . Teda, či sa najprv vypočíta mocnina  $2^8$ , od ktorej sa odpočíta 1, alebo sa najprv vypočíta rozdiel  $8-1$  a touto hodnotou sa potom počíta mocnina 2. Zisti, ako je to s operáciami násobenia a umocňovania: teda, ako sa počítajú výrazy  $3 \cdot 2^5$  a  $2^5 \cdot 3$ .
16. (\*) Matematici vedia, že keď spočítajú niekoľko za sebou idúcich mocnín 2, dostanú inú druhú mocninu zmenšenú o 1. Skontroluj, či napríklad súčet čísel  $2^0, 2^1, 2^2, \dots, 2^9$ , dáva hodnotu  $2^{10}-1$ .

## 2. hodina - Premenné

1. Cez NVDA ponuku spusti Python konzolu. Nechaj Python vypočítať, čomu sa rovná  $(123 + 456) * 789$ .
2. V matematike si zvykol označovať hodnoty písmenkami, napríklad dĺžka strany štvorca  $a = 100$ . Takto to môžeš urobiť aj v Pythone. Vyskúšaj napísať:
3. 

```
>>> a = 100          # stlač Enter
```

Ak je to bez chyby a nič sa nevypísalo, je to správne. Python si vyrobil **premennú s menom**  $a$ , pritom si **zapamätal**, že má hodnotu 100. Premennú si môžeme predstaviť ako škatuľku označenú písmenom  $a$ , v ktorej je hodnota 100.

3. Skús teraz napísať iba:

```
>>> a                # stlač Enter
```

Uvidíš, akú hodnotu si Python pamätá v premennej  $a$ .

4. Vyskúšaj vyrobiť a nastaviť aj iné premenné:

```
>>> vyska = 167
>>> cena = 22 + 7
```

Teraz máme tri škatuľky. V škatuľke  $a$  je hodnota 100, v škatuľke  $vyska$  je hodnota 167 a v škatuľke  $cena$  je hodnota 29.

5. Skontroluj, že premenné s menami  $vyska$ ,  $cena$  majú správne hodnoty.

### Premenná

Premenná funguje podobne, ako pamäť kalkulačky (tlačidlo **M**) – do nej si vieme uložiť jednu hodnotu a túto neskôr použiť v ďalších výpočtoch. V Pythone si môžeš vytvoriť ľubovoľný počet takýchto „pamätí“.

6. Skús napísať:

```
>>> vek # stlač Enter
```

Keďže premenná vek neexistuje, vypíše sa niekoľko riadkov – dôležitý je posledný riadok:

```
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    vek
NameError: name 'vek' is not defined ... neexistuje premenná vek
```

7. Premenné môžeš použiť aj v matematických zápisoch a Python nám namiesto mena premennej dosadí jej hodnotu. Vyskúšaj zápisy:

```
>>> 190 - vyska
>>> 3 * cena + 10
>>> cena + vyska
```

8. Premennej môžeme **zmeniť** jej obsah – vyskúšaj:

```
>>> cena = 5 * 11
```

Momentálny stav pamäti by sme mohli zakresliť takto – všimni si, že sa zmenila premenná cena:

9. Zmeň hodnotu premennej vyska tak, aby v nej bola tvoja výška. Presvedč sa, že sa tak stalo.

10. Skús aj takéto príkazy – čo vykonajú?

```
>>> obsah = a * a
>>> obsah
>>> a = 1
>>> obsah
```

Popíš pomocou škatuliek obsah premenných.

### Priradovací príkaz

**Priradovací príkaz** je taký zápis, v ktorom sa pred znakom = nachádza nejaké meno premennej a za znakom = je hodnota, ktorú treba do tejto premennej uložiť:

```
meno = hodnota
```

Ak je hodnotou aritmetický výraz, tak sa najprv vyhodnotí a až potom **priradí** (nastaví) do premennej.

11. Priradiť do premennej zmrzlina cenu jednej zmrzliny (napríklad 1.2 euro). Do premennej pocet priradiť počet kamarátov, ktorým chceš kúpiť po jednej zmrzline. Zostav príkaz priradenia bez čísel, ktorým sa do tretej premennej zaplatiť priradí celková suma, ktorú zaplatíš. Presvedč sa, že to počítač dobre vypočítal.
  
12. Priradiť do premenných dlzka, sirka a hlbka rozmery školského bazénu v centimetroch (napríklad s hodnotami dlzka=2500, sirka=1000, hlbka=180). Zostav príkazy priradenia:
  - a. Do premennej litre priradiť výraz na výpočet množstva vody v litroch potrebného na napustenie celého bazéna.
  - b. Do premennej objem priradiť výraz na výpočet, koľko je to kubických metrov.

13. Zapiš takýto výpočet:

- do premennej x priradiť nejakú hodnotu,
- zobraz hodnotu výrazu x a pripočítaj 1, výsledok vynásob 2,
- opäť k výsledku pripočítaj 1 a vynásob 2,
- do tretice opäť k výsledku pripočítaj 1 a vynásob 2.

Napríklad, pre x rovné 5, by si mal dostať výsledok 54.

14. V matematike sa počíta faktoriál nejakého čísla n ako súčin čísel od 1 do n. Napríklad 4 faktoriál je súčin čísel  $1 * 2 * 3 * 4$ . Do premennej factorial10 priradiť hodnotu 10 faktoriál (súčin čísel od 1 do 10).

15. Vyskúšaj a všimni si mená premenných, popíš ich pomocou krabičiek:

```
>>> strana_stvorca = 150
>>> obvod_stvorca = 4 * strana_stvorca
>>> obsah_stvorca = strana_stvorca * strana_stvorca
```

Meno premennej

Premenným môžeš dať takmer ľubovoľné meno zostavené z písmen, číslic a podčiarkovníkov. Nesmie však začínať číslicou, nemôže obsahovať medzeru ani iné špeciálne znaky (bodky, plus, mínus atď).



16. V matematike zapisujeme obsah kruhu  $\pi r^2$  a obvod kruhu ako  $2\pi r$ . Skús podobne, ako v príklade 15, nazvať premenné pre polomer, obsah aj obvod kruhu dlhšími menami a priradiť do nich správne výrazy. Vytvor si aj premennú  $\pi$  s hodnotou 3.14.

17. Povedz, ktoré z týchto mien môžu alebo nemôžu byť premennými a vyskúšaj do nich niečo priradiť:

kuk  
Ahoj!  
1.A  
prva\_trieda  
cierno-biele  
OK  
o0o0o0o  
asdf  
výška  
počet ľudí  
trieda(3)

## 3. hodina - Program¶

1. Cez NVDA ponuku spusti Python konzolu. Do premennej suma prirad' počet korún, ktoré budeš meniť na eurá. Do premennej kurz prirad' kurz eura (napríklad, 1 euro za 25.230 českých korún). Do premennej dostanes prirad' hodnotu výrazu, ktorým sa vypočíta, koľko eur dostaneš za svoju sumu. Začni takto:

```
>>> suma = 100
>>> kurz = ...
```

2. Vyskúšaj nový príkaz:

```
>>> print('Ahoj, ja som počítač')
```

- o toto ' sú **apostrofy** – nájdí ich na klávesnici

print

Príkaz print slúži na **vypisovanie** textov. Text napíšeš medzi apostrofy.

3. Použi print a vypíš pomocou neho mená dvoch svojich kamarátov, napríklad:

```
Moji kamaráti sú Vašek a Mirka.
```

4. Skús aj takéto príkazy:

```
>>> print(1 + 2 * 3)
>>> print('1 + 2 * 3')
>>> print()
```

Vidíš, že print vypíše aj hodnotu výrazu, ak nie je medzi apostrofmi.

5. Príkaz print vie vypísať viac vecí – vyskúšaj:

```
>>> print('Mám rád', 'kapustu')
>>> print('Moje obľúbené číslo je', 42)
>>> print('Do školy som išiel', 2 * 10, 'minút')
```

6. Vypíš pomocou príkazu print svoju výšku v tvare:

V príkaze print nedávaj svoju výšku medzi apostrofy.

7. Preskúmaj, čo sa stane, keď zabudneš:

1. apostrof na konci textu v zátvorke

```
>>> print('Ahoj)
```

2. apostrofy na začiatku a na konci textu v zátvorke

```
>>> print(Ahoj)
```

3. zátvorky

```
>>> print 'Ahoj'
```

4. čiarku medzi rôznymi hodnotami v zátvorke

```
>>> print('Ahoj' 10)
```

8. Zatiaľ si s Pythonom pracoval pomocou Python konzoly v interaktívnom režime. Za symboly >>> si zapisoval jednotlivé príkazy, ktoré sa ihneď vykonali. Odteraz budeš vytvárať programy, čiže najprv napíšeš všetky príkazy, až potom takýto program spustíš.

1. Spusti program **Visual Studio Code**.
2. Z hlavnej ponuky zvol **File ► New File** alebo stlač **Ctrl+N**.
3. Vytvorí sa nové viacriadkové editačné pole, do ktorého budeš zapisovať program.
4. Do **editačného poľa** napíš takéto príkazy:

```
print('Ahoj')  
print('Pozdravuje Ťa Python')
```

5. Ulož program do súboru voľbou z ponuky **File ► Save** alebo stlač **Ctrl+S**.
6. Do políčka **Názov súboru**: napíš názov súboru, napríklad prvý, v políčku **Uložiť vo formáte**: zvol Python a stlač tlačidlo **Uložiť**.
7. Teraz program **spusti** tak, že pomocou **Shift+F10** zobrazíš kontextovú ponuku a v nej zvolíš **Run Python File in Terminal**.
8. Na paneli **Terminal** uvidíš výsledok. Čítač Ti ho prečíta. Pomocou klávesu **F6** sa môžeš presúvať medzi jednotlivými panelmi.

9. Do svojho programu pridaj ďalší príkaz print, ktorým vypíšeš text „Dnes je streda“ (alebo hocikajký iný text). Program ulož a spusti ho.

10. Vytvor program, ktorý vypíše úryvok tvojej obľúbenej básničky alebo pesničky. Ak ti žiadna nenapadá, môžeš vypísať túto básničku:

Stojí, stojí mohyla.  
Na mohyle zlá chvíľa,  
na mohyle tŕnie, chrastie  
a v tom tŕní, chrastí rastie,  
rastie, kvety rozvíja  
jedna žltá ľalija.

## 4. hodina - Výpisy

1. Vytvor program, ktorý ťa po spustení privíta správou s dvoma riadkami:

```
Dobrý deň  
Začína programovanie
```

2. Dopln do predchádzajúceho programu príkazy `print` a vypíš pomocou nich pod sebou hodnoty výrazov: `1*1`, `11*11`, `111*111`, `1111*1111`, ..., `111111111*111111111`.

3. Aj vo svojom programe môžeš používať premenné – vyrob takýto program a spusti ho:

```
vek = 16  
print('Mám', vek, 'rokov')
```

Ak program spustíš, vypíše sa:

```
Mám 16 rokov
```

4. Pridaj na koniec tvojho programu ďalší príkaz, ktorým vypíšeš správu:

```
Na budúci rok budem mať 17 rokov
```

Otestuj svoj program.

5. Zobrazí tvoj program správny výsledok, ak namiesto 16 priradiš do premennej `vek` hodnotu 10? Ak nie, program oprav.

6. Vyrob program peňaženka. Na začiatku prirad' do premennej `peniaze`, koľko máš eur. Do premennej `platba` prirad' cenu nákupu. Použi premenné a vypíš pomocou nich:

```
Mám ... eurá  
Platím ... eurá  
Zostane mi ... eurá
```

7. Školské ihrisko má šírku 50 metrov a dĺžku 80 metrov. V rámci telocviku budeš behať po jeho obvode. Koľko metrov zabehneš po 7 kolách? Na začiatku programu prirad' do premenných `sirka`, `dlzka`, `pocet_kol` hodnoty a pomocou nich vypíš:

Šírka ihriska je 50 metrov, dĺžka je 80 metrov  
Jedno kolo okolo ihriska je 260 metrov  
Po 7 kolách zabehneš 1820 metrov

Zobrazí program správne hodnoty aj v prípade, že do premennej `sirka` priradiš hodnotu 45 a do premennej `dĺzka` priradiš hodnotu 70? Ak nie, program oprav.

8. Internetový obchod s hudbou ponúka 20% zľavu. Chceš kúpiť album, ktorého pôvodná cena bola 7.5 eur. Napíš program, ktorý vypočíta, koľko zaplatíš. V programe použi premenné `povodna_cena`, `zlava` (v percentách), `cena_po_zlave` a pomocou nich zrealizuj výpočty a vypíš:

Cena albumu 7.5 eur  
Zľava 20 percent  
Zaplatíš 6.0 eur

9. Kamaráti Anka, Marienka a Peťo diskutujú na sociálnej sieti. Anka napísala 3 príspevky. Marienka na každý z nich poslal 2 odpovede. Peťo všetko komentuje a na všetky príspevky Anky a Marienky posielal po 5 komentárov. Napíš program, ktorý zhodnotí diskusiu:

Počet príspevkov od Anky: 3  
Počet príspevkov od Marienky: 6  
Počet príspevkov od Peťa: 45

Program vytvor tak, aby sa na začiatku do premenných `pocet1`, `pocet2` a `pocet3` priradili počet príspevkov Anky, počet odpovedí na každý z nich od Marienky a počet komentárov na každý z príspevkov od Peťa.

## 8. hodina - Podprogramy¶

1. Už si zamieňal české koruny na eurá. Teraz vytvor nový program, ktorý zamení eurá na české koruny. Použi premenné `suma` a `kurz`, do ktorých priradiš počiatočné hodnoty – koľko eur ideš zamieňať a aktuálny kurz (napríklad, 25.23 českých korún za 1 euro). Do premennej dostaneš priradiť hodnotu výrazu, ktorým sa vypočíta, koľko eur dostaneš za svoju sumu. Program vypíše výsledok napríklad v tvare:

Za ... eur dostaneš ... českých korún pri kurze ... českých korún za euro.

2. Nasledujúci program pracuje s premennými. Zisti bez použitia počítača, čo program vypíše:

```
kg = 3
c = 20
s = kg * c + 7
print('Za poslanie balíka s hmotnosťou', kg, 'kg zaplatíš', s, 'korún')
```

Svoje zistenie skontroluj na počítači

3. Doteraz si mohol písať iba také príkazy, ktoré poznal počítač. Teraz ho naučíš nové, tvoje vlastné príkazy, takzvané **podprogramy**. Postupuj nasledovne:

1. Vytvor nový program, v ktorom bude napísané iba toto:
  - prázdne zátvorky aj dvojbodka sú veľmi dôležité

```
def vypis_text():
    print('*****')
    print('** Python **')
    print('*****')
```

- príkazy nechaj takto odsunuté (Python tam automaticky vloží 4 medzery)
2. Program spusti – ak, je všetko v poriadku, v paneli Terminál sa zdanlivo nič neudeje.
3. Pridaj na koniec programu nasledujúci riadok a opäť ho spusti.

```
vypis_text()
```

4. Ak si postupoval správne, počítač zobrazí text:

```
*****
** Python **
*****
```

## Definovanie a volanie podprogramu

Čo sa stalo?

- slovom `def` začína **definícia** tvojho nového príkazu – podprogramu
- `vypis_text` je **názov** podprogramu

```
def vypis_text():  
    print('*****')  
    print('** Python **')  
    print('*****')
```

- ďalšie tri riadky tvoria **telo** podprogramu

Po spustení programu sa počítač naučil nový príkaz `vypis_text`. Počítač ho zatiaľ nevykonal, iba sa ho naučil. Skupinu príkazov `print` – teda telo podprogramu `vypis_text`, počítač vykoná až vtedy, keď do programu napíšeš:

```
vypis_text()
```

- prázdne zátvorky sú veľmi dôležité

Takýto zápis sa nazýva **volanie** podprogramu.

### 4. Pridaj do svojho programu ďalšie príkazy – nesmú mať odsadenie:

```
def vypis_text():  
    print('*****')  
    print('** Python **')  
    print('*****')  
  
print('Vitaj!')  
vypis_text()  
print()  
vypis_text()  
print('to je koniec')
```

Keď program spustíš, uvidíš takýto výsledok:

```
Vitaj!  
*****  
** Python **  
*****  
  
*****  
** Python **  
*****  
to je koniec
```

V tomto programe sa najskôr zadefinoval podprogram `vypis_text`. Za ním nasledujú príkazy `print` a príkazy pre volanie podprogramu. Python zobrazil svoju vizitku 2-krát, lebo v programe sú 2 volania podprogramu `vypis_text`.



5. Zmeň predchádzajúci program tak, aby počítač vypísal:

```
Hello!
*****
** I am Python **
*****
How are you?
*****
** I am Python **
*****
I am fine.
*****
** I am Python **
*****
The end
```

6. Vytvor program, ktorý zobrazí, napríklad takúto tvoju vizitku:

```
+-----+
|  Juraj  |
| RUMCAJS |
| IT guru |
+-----+
```

Zadefinuj podprogram `vizitka`, ktorý takúto vizitku zobrazí. Nakoniec tento podprogram zavolaj, aby si overil, že funguje správne.

7. Dopln' do predchádzajúceho programu volania podprogramu `vizitka` tak, aby sa pod seba zobrazilo 10 tvojich vizitiek.

8. Vo svojom programe môžeš definovať aj viac podprogramov. Vytvor nový program a zadefinuj v ňom tri podprogramy. Každý z nich zobrazí jeden z nasledujúcich obrázkov:

- o podprogram `noha` nakreslí takúto nohu (dole sú dva podčiarkovníky vľavo aj vpravo):

```
  |
__|__
```

- o podprogram `obdlznik` nakreslí:

```
####
#  #
####
```

- o podprogram `trojuholnik`:

```
 *
***
*****
```

Na konci programu vlož volanie podprogramov, aby si otestoval každý z nich a skús pomocou nich zobrazit' nasledujúce obrázky.

o stromček

```
 *
***
*****
 *
***
*****
 |
___|___
```

o domček na stračej nôžke

```
 *
***
*****
#####
#   #
#####
 |
___|___
```

## 9. hodina - Náhoda ¶

1. Napíš program, v ktorom zadefinuješ 2 podprogramy s príkazmi print:
  - o podprogram ciara zobrazí v jednom riadku 19 znakov +--+--+ a tak ďalej
  - o podprogram palicky zobrazí na striedačku paličku a dve medzery | | tak, aby znakov bolo 19.

Na konci programu zavolaj na striedačku podprogramy ciara a palicky tak, aby sa zobrazilo:

```
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
```

2. Pridaj do predchádzajúceho programu ešte jeden podprogram stvorcekovy\_papier. Ten využije tvoje podprogramy ciara a palicky tak, že pomocou ich volania zobrazí štvorcovú sieť:

```
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
| | | | | | | | | | | | | | | |
+--+--+--+--+--+--+--+--+--+--+
```

3. V počítači funguje náhoda. Z NVDA ponuky spusti **Python konzolu**. Zadaj do príkazového riadku tieto príkazy:
  4. >>> import random
  5. >>> random.randint(1, 6)

Počítač zobrazí nejaké číslo, napríklad:

5

Nechaj vykonať príkaz random.randint(1, 6) niekoľkokrát. Aké čísla počítač zobrazil?

```
>>> random.randint(1, 6)
6
>>> random.randint(1, 6)
4
>>> random.randint(1, 6)
4
```

Slovo random znamená **náhodný**. Pri vykonaní príkazu `random.randint(1, 6)` si počítač vymyslí nejaké číslo od 1 po 6. Je to podobné, akoby si počítač hodil hracou kockou.

4. Náhodné číslo si môžeš zapamätať. V programe **Visual Studio Code** napíš a spusti (aj viackrát za sebou) takýto program:

```
import random

n = random.randint(1, 6)
print('Na kocke padla', n)
```

5. Vytvor podprogram `hod_kockou` a doplň ho tak, aby hodil kockou 10 krát:

```
import random

def hod_kockou():
    n = random.randint(1, 6)
    print('Na kocke padla', n)

hod_kockou()
```

Mal by si vidieť podobný výpis:

```
Na kocke padla 5
Na kocke padla 3
Na kocke padla 4
Na kocke padla 1
Na kocke padla 3
Na kocke padla 2
Na kocke padla 1
Na kocke padla 1
Na kocke padla 1
Na kocke padla 1
Na kocke padla 3
```

6. Uprav predchádzajúci program tak, aby počítač simuloval hod na 20 stennej kocke.
7. (\*) Máme binárnu hraciu kocku, ktorá má na troch stenách 0 a zvyšných troch 1. Uprav predchádzajúci program, aby simuloval hod takou kockou.
8. (\*) Máme párnú hraciu kocku, ktorá má na stenách tieto čísla: 2, 4, 6, 8, 10, 12. Uprav predchádzajúci program, aby simuloval hod takou kockou.

9. (\*) Máme nepárnu hraciu kocku, ktorá má na stenách tieto čísla: 1, 3, 5, 7, 9, 11.  
Uprav predchádzajúci program, aby simuloval hod takou kockou.

10. (\*) Máme exotickú hraciu kocku, ktorá má na stenách tieto čísla: 1, 4, 9, 16, 25, 36.  
Uprav predchádzajúci program, aby simuloval hod takou kockou.

11. Napíš podprogram predpoved, ktorý vypíše správu:

Dnes bude 15 stupňov.

Číselný údaj podprogram zvolí náhodne z intervalu  $\langle -15, 35 \rangle$ .

12. Vytvor program, ktorý vygeneruje náhodný PIN pre tvoj mobil. Do 4 premenných a, b, c, d priradi náhodné čísla od 0 po 9, a potom ich jedným príkazom print vypíš.  
Napríklad:

Tvoj nový PIN je 1 3 7 3

13. Vytvor generátor náhodných dátumov (nech má každý mesiac 30 dní). Po spustení program vypíše, napríklad:

Koniec sveta bude 30 . 2 . 2019

# 11. hodina - Program s opakovaním

1. Išli sme 1. apríla do troch obchodov, v ktorých mali predajcovia veselú náladu. Každý od nás za nákup vypýtal náhodnú sumu peňazí z intervalu od 4 do 20 eur. Napíš podprogram nakupy, ktorý vygeneruje tri náhodné sumy, vypíše ich a na záver vypíše aj ich súčet, napríklad:

```
Tvoj prvý nákup bol za 8 eur
Tvoj druhý nákup bol za 12 eur
Tvoj tretí nákup bol za 5 eur
Spolu si zaplatil 25 eur
```

2. Vytvor program, ktorý pomocou príkazu print vypíše text 'Teším sa na prázdniny' 5-krát pod seba.
3. V predchádzajúcom programe si mal viackrát nakopírovaný príkaz print(...). Aby si ho nemusel opakovane kopírovať, môžeš to zapísať jednoduchšie:
  - o dvojbodka je tu veľmi dôležitá:

```
for i in range(5):
    print('Teším sa na prázdniny')
```

- o príkaz nechaj odsunutý – Python tam automaticky vloží 4 medzery

Spusti takýto program.

4. Vyskúšaj namiesto 5 dať číslo 10 a program znovu spusti. Experimentuj aj s inými číslami, napríklad 1, 100 a podobne.
5. Uprav program takto a spusti ho:

```
for i in range(5):
    print('Teším sa na prázdniny')
    print('=====')
```

Ak si správne postupoval, po spustení uvidíš:

```
Teším sa na prázdniny
=====
Teším sa na prázdniny
```

```
=====
Teším sa na prázdniny
=====
Teším sa na prázdniny
=====
Teším sa na prázdniny
=====
>>>
```

Je dôležité odsadiť príkazy, ktoré tvoria telo cyklu. Vyskúšaj, čo vypíše takto upravený program:

```
for i in range(5):
    print('Teším sa na prázdniny')
print('=====')
```

Ako program funguje?

- slovom **for** začína príkaz **cyklu**
- číslo **5** tu označuje počet **opakovaní**

```
for i in range(5):
    print('Teším sa na prázdniny')
    print('=====')
```

- odsunuté riadky tvoria **telo** cyklu – tieto príkazy sa vykonajú **5-krát**

6. Je daný nasledujúci program:

```
import random

for i in range(5):
    n = random.randint(1, 100)
    print('bolo vylosované číslo', n)
```

Slovami popíš, čo program vykoná.

7. Napíš program, ktorý simuluje hádzanie dvomi kockami. Zapiš pomocou for cyklu 5 hodov, pričom v tele sa do dvoch premenných priradia dve náhodné čísla, vypíšu sa a vypíše sa aj ich súčet. Výpis môže vyzeráť takto:

```
Na prvej kocke padlo číslo 4
Na druhej kocke padlo číslo 3
Súčet oboch čísel je 7
```

```
Na prvej kocke padlo číslo 2
Na druhej kocke padlo číslo 4
Súčet oboch čísel je 6
```

```
Na prvej kocke padlo číslo 5
```

Na druhej kocke padlo číslo 2  
Súčet oboch čísel je 7

Na prvej kocke padlo číslo 3  
Na druhej kocke padlo číslo 1  
Súčet oboch čísel je 4

Na prvej kocke padlo číslo 1  
Na druhej kocke padlo číslo 4  
Súčet oboch čísel je 5

8. Vyskúšaj, ako funguje `random.choice` – nechaj vykonať každý z príkazov v cykle niekoľko krát:

1. `print(random.choice(['Ahoj', 'Nazdar', 'Servus', 'Čau']))`
2. `print(random.choice('PYTHON'))`
3. `print(random.choice([1 / 2, 1 / 3, 1 / 4, 1 / 5]))`

9. Napíš program, ktorý zobrazuje správy v tvare:

Dnes je ... deň

Namiesto ... sa vypíše jedna z možností 'pekný', 'škaredý', 'upršaný', 'slnečný'.



# 12. hodina - Premenná cyklu

1. Tvoj mladší súrodeneц našiel ľudovú riekanku:

```
kočka leze dírou  
pes oknem  
pes oknem  
nebude-li pršet  
nezmoknem  
nebude-li pršet  
nezmoknem
```

Vyrob program, ktorý ju vypíše pomocou `print`. Použi `for` cykly, aby bolo v programe čo najmenej príkazov `print`.

2. Aha, takto vypíšeš čísla od 0 po 9 – vyskúšaj:

```
for i in range(10):  
    print('číslo', i)
```

Ak si to správne zapísal, program po spustení vypíše:

```
číslo 0  
číslo 1  
číslo 2  
číslo 3  
číslo 4  
číslo 5  
číslo 6  
číslo 7  
číslo 8  
číslo 9
```

Prečo to funguje?

- `i` je premenná, do ktorej príkaz `for` postupne priradzuje čísla od 0 po 9
- rozsah čísel je z intervalu  $<0, 10)$

```
for i in range(10):  
    print('číslo', i)
```

- pre každé číslo sa vykoná telo cyklu a vypíše sa hodnota premennej `i`

3. Čo treba v predchádzajúcom programe zmeniť, aby sa vypísali čísla:

0, 1, ... 10 – teda, aj číslo 10

1, 2, ... 10

2, 4, ... 20

10, 20, ... 100

Vyskúšaj.

for

Príkaz `for` čítame: „pre *i* v rozsahu(...) vykonaj telo cyklu“

4. Vytvor program, ktorý vypíše pomocou `for` cyklu čísla a ich druhé mocniny:

```
0 na druhú je 0
1 na druhú je 1
2 na druhú je 4
3 na druhú je 9
4 na druhú je 16
5 na druhú je 25
6 na druhú je 36
```

5. Máme takúto rozprávku:

```
Na strome bolo 0 vrabcov, jeden priletel a už je na ňom 1 vrabcov
Na strome bolo 1 vrabcov, jeden priletel a už je na ňom 2 vrabcov
Na strome bolo 2 vrabcov, jeden priletel a už je na ňom 3 vrabcov
Na strome bolo 3 vrabcov, jeden priletel a už je na ňom 4 vrabcov
Na strome bolo 4 vrabcov, jeden priletel a už je na ňom 5 vrabcov
Na strome bolo 5 vrabcov, jeden priletel a už je na ňom 6 vrabcov
Na strome bolo 6 vrabcov, jeden priletel a už je na ňom 7 vrabcov
Na strome bolo 7 vrabcov, jeden priletel a už je na ňom 8 vrabcov
Na strome bolo 8 vrabcov, jeden priletel a už je na ňom 9 vrabcov
Na strome bolo 9 vrabcov, jeden priletel a už je na ňom 10 vrabcov
```

Zapíš ju pomocou `for` cyklu.

6. Vrabce z predchádzajúcej rozprávky odlietajú – vymysli program, ktorý to pomocou cyklu `for` porozpráva:

```
Na strome bolo 10 vrabcov, jeden odletel a zostalo na ňom 9 vrabcov
Na strome bolo 9 vrabcov, jeden odletel a zostalo na ňom 8 vrabcov
Na strome bolo 8 vrabcov, jeden odletel a zostalo na ňom 7 vrabcov
Na strome bolo 7 vrabcov, jeden odletel a zostalo na ňom 6 vrabcov
Na strome bolo 6 vrabcov, jeden odletel a zostalo na ňom 5 vrabcov
Na strome bolo 5 vrabcov, jeden odletel a zostalo na ňom 4 vrabcov
Na strome bolo 4 vrabcov, jeden odletel a zostalo na ňom 3 vrabcov
Na strome bolo 3 vrabcov, jeden odletel a zostalo na ňom 2 vrabcov
Na strome bolo 2 vrabcov, jeden odletel a zostalo na ňom 1 vrabcov
Na strome bolo 1 vrabcov, jeden odletel a zostalo na ňom 0 vrabcov
```

## 13. hodina - Výrazy v cykle

1. Minule si vytváral program, ktorý v textovom režime pomocou príkazov for a print vypísal čísla a ich druhé mocniny. Vytvor takýto program, ale namiesto druhých mocnín nech program vypíše tretie mocniny.

```
0 na tretiu je 0
1 na tretiu je 1
2 na tretiu je 8
3 na tretiu je 27
4 na tretiu je 64
5 na tretiu je 125
6 na tretiu je 216
```

2. Je daný nasledujúci program:

```
y = 10
for i in range(11):
    print(y, " ", i)
    y = y + 20
```

Doplň tabuľku, aby bolo vidno, ako sa menia premenné i a y počas vykonávania cyklu:

**i y v príkaze print y po vykonaní  $y = y + 20$**

0	10	30
1	30	
2		
.		
.		
.		
.		
.		
.		
.		
.		

Akú hodnotu bude mať premenná y po skončení cyklu?

3. Hráme počítačovú hru, ktorá má 10 levelov. Po úspešnom prejdení i-teho levelu získame i bodov. Napríklad, po prejdení prvého levelu získame 1 bod. Po prejdení

druhého levelu sa nám ku skóre pripočítajú 2 body, takže spolu už máme 3 body. Po prejdení tretieho levelu sa nám pripočítajú 3 body, takže naše skóre bude 6 bodov atď. Napíš program, ktorý pomocou print a cyklu vypíše, ako rastie skóre po prejdení každého levelu.

Po leveli 1 bude tvoje skóre 1 bodov.  
Po leveli 2 bude tvoje skóre 3 bodov.  
Po leveli 3 bude tvoje skóre 6 bodov.  
Po leveli 4 bude tvoje skóre 10 bodov.  
Po leveli 5 bude tvoje skóre 15 bodov.  
Po leveli 6 bude tvoje skóre 21 bodov.  
Po leveli 7 bude tvoje skóre 28 bodov.  
Po leveli 8 bude tvoje skóre 36 bodov.  
Po leveli 9 bude tvoje skóre 45 bodov.  
Po leveli 10 bude tvoje skóre 55 bodov.

4. Poznáš rozprávku o prefikanom mudrcovi a šachovi, ktorý sľúbil za odmenu toľko zrníek pšenice, koľko sa zmestí na políčka šachovnice? Pomôž šachovi v rozhodovaní, či je taká odmena adekvátne a vytvor preňho jednoduchý program, ktorý vypíše celkový počet zrníek na šachovnici. Na prvé políčko dáme 10 zrníek, na druhé 20, na tretie 30 atď. Políčok na šachovnici je 64.
5. V skutočnosti predchádzajúca rozprávka bola taká, že na prvé políčko dáme 1 zrnko, ale na každé ďalšie dvojnásobok predchádzajúceho (t.j. 2, 4, 8, ...). Vytvor program, ktorý zistí celkový počet zrníek na šachovnici.

## 16. hodina - Vetvenie

1. Napiš program, ktorý pomocou cyklu zistí, čomu sa rovná súčet čísel  $0 + 1 + 2 + \dots + 99$ . Výsledok program vypíše pomocou print.
2. (\*) Napiš program, ktorý pomocou `random.choice(...)` vypíše jeden z nasledujúcich pozdravov: Dobré ráno, Dobrý deň, Dobrý večer, Dobrú noc. Použi cyklus, aby sa vypísalo 10 pozdravov. Napríklad:

```
Dobrá noc  
Dobrý deň  
Dobrá noc  
Dobrý večer  
Dobrá noc  
Dobrá noc  
Dobré ráno  
Dobrá noc  
Dobrý večer  
Dobrá noc
```

3. Počítač dokáže porovnávať čísla. Cez **NVDA ponuku** spusti **Python konzolu**. Napiš do príkazového riadku:

```
>>> 1 < 2          # stlač Enter
```

Počítač čísla porovná a vypíše:

```
True
```

- o počítač nám odpovedal, že je **pravda**, že 1 je menšia ako 2

Vieme, že 3 nie je väčšia ako 4. Vyskúšaj, čo ti počítač vypíše, ak do príkazového riadku zadáš:

```
>>> 3 > 4
```

4. Môžeš porovnávať aj zložitejšie výrazy. Aké výsledky vypíše počítač?

1. `>>> 1 + 2 > 3`
2. `>>> -1 > -2`
3. `>>> (1 + 2) * 3 < 1 + 2 * 3`
4. `>>> a = 100`

```
>>> a < 101
```

5. V programe **Visual Studio Code** chceme vytvoriť program, ktorý by nám povedal, či je vonku teplo alebo zima. Mohlo by to fungovať tak, že do premennej teplota priradíme číslo. Potom by počítač pre teplotu väčšiu ako 20 stupňov vypísal, že je teplo. Inak by nám povedal, že je zima. Takto to naprogramuješ:
- o Nezabudni na dvojbodky na konci riadkov s if a else – sú dôležité

```
teplota = 25
print('Je', teplota, 'stupňov.')
if teplota > 20:
    print('Dnes je teplo.')
else:
    print('Dnes je zima.')
print('Správne sa obleč.')

```

- o odsad' príkazy print – použi tabulátor

Ak si program napísal správne, po jeho spustení uvidíš:

```
Je 25 stupňov.
Dnes je teplo.
Správne sa obleč.

```

6. Zmeň v predchádzajúcom príklade teplotu z 25 na 10. Program spusti. Čo sa vypíše?

Ako program funguje?

- if ... else ... je nový príkaz pre **vetvenie** programu

```
teplota = 25
print('Je', teplota, 'stupňov.')
if teplota > 20:
    print('Dnes je teplo.')
else:
    print('Dnes je zima.')
print('Správne sa obleč.')

```

- **podmienka**, podľa ktorej sa počítač rozhodne, ktorú vetvu ďalej vykoná

Keď počítač uvidí príkaz if ... else ..., najskôr vyhodnotí **podmienku**. Ak je podmienka splnená, vykoná sa príkaz vo vetve if, inak sa vykoná príkaz vo vetve else.

7. Uprav predchádzajúci program tak, aby pre:
- o záporné teploty vypísal Zober si rukavice,
  - o inak vypíše, že Rukavice ti netreba.

Over, že program správne funguje pre kladné aj záporné hodnoty. Čo tvoj program vypíše pre 0?

8. Chceme poslať doporučený list. Vytvor program, ktorý ti poradí s cenou dopisu. Na začiatku programu prirad' do premennej hmotnosť číslo s hmotnosťou tvojho listu. Použi príkaz pre vetvenie programu, aby pre list s hmotnosťou:
- o do 50 g vypísal Zaplatíš 1.25 eura,
  - o inak vypíše, že Zaplatíš 1.4 eura.

Over, že program správne počíta cenu listu pre hmotnosti: 30, 50 a 100 gramov.

Doporučené listy nad 100 g stoja 1.7 eura. Náš program to zatiaľ nevyhodnotí správne.

9. Ideme na zmrzlinu. Cena za jeden za kopček zmrzliny je 1.1 eura. Zmrzlinár však ponúka zľavu: ak vezmeme viac ako 5 kopčekov, cena za každý kopček bude 0.90 eura. Vytvor program, ktorému na začiatku priradiš do premennej počet počet kopčekov a on vypíše výslednú cenu. Napríklad:

Za 5 kopčekov zmrzliny zaplatíš:  
4.5 eurá.

Over, že program správne počíta cenu zmrzliny pre 4, 5 a 6 kopčekov.

10. Vytvor program, ktorý pre pretekárov zobrazuje oznam, či majú stáť, alebo môžu ísť. Do premennej čas prirad' počet sekúnd. Ak je toto číslo menšie ako 30, program vypíše **STOJ!**, inak vypíše **CHOĎ!**.

**STOJ!**      **CHOĎ!**  
pre cas = 5    pre cas = 50

11. Over, že tvoj program vypisuje správny povel pre časy 5, 30 aj 50 sekúnd.

# 17. hodina - Vetvenie a konštrukcie

1. Kamarátku pozdravíš „Ahoj“, ale vieš, že starších ľudí treba pozdraviť slušne, napríklad „Dobrý deň“. Napíš program, v ktorom do premennej vek priradiš vek človeka. Potom použi príkaz vetvenia na to, aby sa program podľa veku rozhodol, ktorý z uvedených dvoch pozdravov vypíše. Otestuj, aké pozdravy sa vypisujú pre rôzne hodnoty premennej vek.
2. Na brigáde v bufete ťa zaplatia podľa takéhoto pravidla:
  - o ak budeš pracovať menej ako 10 hodín, tak zarobíš 3 eurá za hodinu,
  - o inak dostaneš plat 4 eurá za hodinu.

Do premennej hodin prirad' počet tvojich odpracovaných hodín. Potom pomocou príkazu vetvenia vypíš, koľko zarobíš. Program by mal vypísať:

pre hodin = 7:

Zarobíš 21 eur.

pre hodin = 12:

Zarobíš 48 eur.

3. Predchádzajúca úloha sa dá riešiť aj takto:

```
hodin = 20
if .....:
    plat = .....
else:
    plat = .....
print('Zarobíš', plat, 'eur.')
```

Doplň namiesto vybodkovaných častí správne výrazy. Over, že program korektne počíta plat pre rôzne hodnoty premennej hodin.

4. Telefónny operátor *Vegafón* počíta platby za prenesené megabajty podľa takýchto pravidiel:
  - o ak za deň prenesieš menej ako 10 megabajtov, zaplatíš za každý megabajt 0.2 eur,
  - o inak zaplatíš za celý deň 2 eurá.



Napiš program, v ktorom do premennej megabajt priradiš počet prenesených megabajtov za jeden deň. Použi príkaz vetvenia na to, aby si do premennej cena priradil vyúčtovaný počet korún. Nakoniec vypíš vypočítanú cenu. Napríklad:

pre megabajty = 3:

Zaplatíš 0.6 eur.

pre megabajty = 20:

Zaplatíš 2 eur.

5. Telefónny operátor *Zodrafón* počíta platby za prenesené megabajty podľa „výhodnejších“ pravidiel:
- o ak za deň neprenesieš viac ako 10 megabajtov, zaplatíš za každý megabajt 0.1 eur,
  - o inak zaplatíš za každý megabajt navyše (nad 10 megabajtov) 0.2 eur.

Napiš program, ktorý počíta cenu podľa takýchto pravidiel a over, že správne funguje. Napríklad:

pre megabajty = 3:

Zaplatíš 0.3 eur.

pre megabajty = 20:

Zaplatíš 3.0 eur.

6. Vytvor program, ktorý podľa zadaného času vypíše buď **svieti slnko** alebo **svieti mesiac**. Do premennej *cas* priradi počet hodín. Použi príkaz vetvenia na to, aby sa pre *cas < 8* vypísalo **svieti mesiac**, inak sa vypísalo **svieti slnko**.

7. Doplň do predchádzajúceho programu, aby vypísal aj text **vonku je tma**, keď svieti mesiac a **vonku je svetlo**, keď svieti slnko.

Stačí, ak do každej vetvy pridaš príkaz, ktorý vypíše, či je vonku svetlo alebo tma:

```
if .....:
    print(.....)
    print(.....)
else:
    print(.....)
    print(.....)
```

- o Odsať oba príkazy, ktoré tvoria telo vetvy if aj else – použi tabulátor

8. Nasledujúci program vypíše denný harmonogram:

```
for i in range(8):
    print(i, 'ešte spím')
for i in range(6):
    print(8 + i, 'som v škole')
```

Čo program vypíše?

Program sa dá zapísať aj takto, iba pomocou jedného cyklu:

```
for i in range(14):
    if i < 8:
        print(i, 'ešte spím')
    else:
        print(i, 'som v škole')
```

- o príkazy vo vetvách if a else treba odsadiť ešte viac

Telo cyklu for obsahuje jeden **vnorený** príkaz vetvenia if else. Vyskúšaj, ako funguje táto verzia.

9. Podobne, ako v predchádzajúcom príklade, použi cyklus s vnoreným vetvením a vypíš:

```
0 som chudobný
10 som chudobný
20 som chudobný
30 som chudobný
40 som chudobný
50 som bohatý
60 som bohatý
70 som bohatý
80 som bohatý
90 som bohatý
```

Premenná cyklu sa mení od 0 po 9. Vypisuje sa desaťnásobok hodnoty tejto premennej.

10. Vieš, pre ktoré čísla  $n$  platí, že  $n^2$  je menšie ako  $5 * n$ ? Napíš program, ktorý pre všetky čísla od 0 po 10 otestuje tento vzťah a vypíše o tom informáciu. Napríklad:

```
...
16 je menšie ako 20
25 je väčšie alebo rovné ako 25
36 je väčšie alebo rovné ako 30
...
```

Čo tvoj program zistil? Pre ktoré  $n$  to platí?

## 18. hodina - Vnorené vetvenie

1. Napíš program, ktorý zobrazí absolútnu hodnotu čísla. Do premennej `a` prirad' číslo. Použi príkaz vetvenia na to, aby si vypísal absolútnu hodnotu tohto čísla. Napríklad:
  - o ak bude `a = -7`, program vypíše `Absolútna hodnota -7 je 7`,
  - o ak bude `a = 13`, program vypíše `Absolútna hodnota 13 je 13`.

2. Zatiaľ vieš porovnávať dve čísla pomocou `<`, `>`. Vyskúšaj, ako fungujú testy na rovnosť `==` a rôznosť `!=`. Z **NVDA ponuky** spusti **Python konzolu**. Napíš do príkazového riadku nasledujúce výrazy a zisti, čo vypíšu:

```
>>> 1 == 1
>>> 1 == 2
>>> 0 != 2
>>> 0 != 0
>>> x = 100
>>> x == 10 * 10
>>> x != 10 * 10
>>> 11 * 11 - 21 == x
>>> 1000 / 10 - 1 != x
```

3. Vieš, čo sa stane, keď má počítač deliť nulou? Vyskúšaj v príkazovom riadku. V prostredí **Visual Studio Code** napíš program, ktorý počíta prevrätenu hodnotu čísla. Do premennej `n` prirad' číslo. Použi príkaz vetvenia a test na rovnosť na to, aby si:
  - o v prípade, že `n = 0` zobrazil správu `Nulou deliť neviem`,
  - o inak počítač zobrazí výsledok v tvare `1 / 10 = 0.1`.
4. Do premenných `x`, `y` prirad' dve čísla. Napíš program, ktorý vypíše, či sú čísla rovnaké alebo rôzne. Napríklad:

```
pre x = 1, y = 0
```

Čísla sú rôzne.

pre  $x = 2, y = 2$

Čísla sú rovnaké.

5. Do premenných  $a, b$  priradiť dĺžky strán. Napíš program, ktorý vypíše, či je to obdĺžnik alebo štvorec. Napríklad:

pre  $a = 10, b = 10$

Je to štvorec.

pre  $a = 10, b = 20$

Je to obdĺžnik.

6. Hodíme 10 krát hracou kockou a chceme vedieť, koľkokrát padla šestka a koľkokrát iné číslo. Napíš program, ktorý pomocou cyklu a vnoreného príkazu vetvenia spočíta. Napríklad:

```
5
6
5
2
2
3
1
4
5
3
Padlo 1 šestiek a 9 iných čísel
```

7. (\*) Hráme hru, v ktorej hádzeme 10x kockou a získavame prémiiu vždy vtedy, keď za sebou padnú dve rovnaké čísla. Napríklad:

```
4
4
6
5
3
3
3
3
1
5
5
Prémie 4
```

Napíš program, ktorý simuluje 10 hodov kockou a spočíta prémiiu.

8. Futbaloví rozhodcovia sa dohodli, ako budú hráčov hodnotiť za priestupky:
- o ak hráč spraví 0 priestupkov, hrá férovo,
  - o ak spraví 1 alebo 2 priestupky, dostáva žltú kartu,
  - o inak dostane červenú a je vylúčený z hry.

Vidíme, že v tejto úlohe je viac podmienok. Pozri si a vyskúšaj toto riešenie:

```
pocet = 0
if pocet == 0:
    print('Hráš férovo')
else:
    if pocet < 3:
        print('Máš žltú kartu')
    else:
        print('Máš červenú kartu')
```

- o príkazy v oboch vetvách vnoreného if else odsad' ešte viac

Doplň do tabuľky, čo program vypíše, ak do premennej `pocet` priradíme hodnotu:

**program vypíše:**

```
pocet = 0
pocet = 1
pocet = 2
pocet = 3
pocet = 4
pocet = -1
```

9. V televíznej súťaži o pečení *Môj dedo pečie lepšie ako tvoj* sú takéto pravidlá:
- o ak súťažiaci stihne upiecť menej ako 10 koláčikov, je hodnotený ako začiatočník,
  - o ak stihne upiecť aspoň 10 ale menej ako 20 koláčikov, je hodnotený ako pokročilý,
  - o pre aspoň 20 koláčikov, je hodnotený ako super.

Vytvor program, v ktorom najprv do premennej `kolace` priradiš počet upečených koláčikov. Program sa potom rozhodne, akú hlášku vypíše.

Otestuj, že program zobrazí správnu hlášku pre čísla 1, 9, 10, 19, 20, 100.

10. V úlohe 6 vidno, že počítač nevie správne skloňovať slová. Mal by vypísať:
- o Padla 1 šestka...
  - o Padli 2 šestky...
  - o Padlo 5 šestiek...

Napiš program, v ktorom do premennej `n` priradiš číslo a počítač zobrazí gramaticky správnu vetu.

11. Chceš porovnať svoj vek s kamarátkou. Vytvor program, v ktorom do premenných ja a ona priradiš tvoj vek a vek tvojej kamarátky. Program ich porovná a podľa toho vypíše: Sme rovnako starí, Mám menej rokov alebo Ona je mladšia.

# 19. hodina - Podprogram s parametrom

1. Vytvor nový program, v ktorom na začiatku priradiš do premennej číslo číslo od 0 po 999. Použi vetvenie na to, aby program rozhodol a správne vypísal hlásenie o tom, či je číslo jedno-, dvoj- alebo trojciferné. Napríklad, pre číslo = 128 program vypíše:

Číslo 128 je trojciferné.

2. Dokonči nasledujúce podprogramy, ktoré vypíšu správny vek:

```
def mam10():
    vek = 10
    print('Mám', vek, 'rokov')

def mam20():
    vek = .....
    print('Mám', vek, 'rokov')

def mam30():
    vek = 30
    print(.....)

mam10()
mam20()
mam30()
```

Urob to tak, aby sa všetky 3 podprogramy navzájom čo najviac podobali.

3. Predchádzajúce riešenie sa dá zapísať pomocou jediného podprogramu:

```
def mam(vek):
    print('Mám', vek, 'rokov')

mam(10)
mam(20)
mam(30)
```

Vyskúšaj.

Ako program funguje?

- v zátvorke v hlavičke podprogramu je **meno parametra**

```
def mam(vek):
    print('Mám', vek, 'rokov')      # tu sa parameter používa - parameter
funguje ako premenná
```

```
mam(10)
mam(20)
mam(30)
```

- **hodnota** v zátvorkách sa priradí do parametra vek:
  - pri prvom zavolaní mam(10) bude mať parameter vek hodnotu 10
  - pri druhom zavolaní mam(20) bude mať parameter vek hodnotu 20
  - pri treťom zavolaní mam(30) bude mať parameter vek hodnotu 30

4. Dokonči podprogram vypis, ktorý používa parameter x na to, aby vypísal hodnotu parametra x a jeho druhú mocninu:

```
def vypis(x):
    print('Číslo', ...)
    print('Umocnené na druhú sa rovná', .....
```

```
vypis(1)
vypis(2)
vypis(3)
```

Program by mal po spustení vypísať:

```
Číslo 1
Umocnené na druhú sa rovná 1
Číslo 2
Umocnené na druhú sa rovná 4
Číslo 3
Umocnené na druhú sa rovná 9
```

5. Doplní do predchádzajúceho podprogramu príkaz, ktorým sa vypíše aj prevrátená hodnota x:

```
Číslo 1
Umocnené na druhú sa rovná 1
Prevrátená hodnota sa rovná 1.0
Číslo 2
Umocnené na druhú sa rovná 4
Prevrátená hodnota sa rovná 0.5
Číslo 3
Umocnené na druhú sa rovná 9
Prevrátená hodnota sa rovná 0.3333333333333333
```

6. Zadefinuj podprogram oblubujem s parametrom cislo. Podprogram vypíše, či číslo obľubuje:

- ak je číslo menšie ako 7, vypíše Mám rád číslo ...
- inak vypíše Číslo ... sa mi nepáči

Podprogram zavolaj a over, že vypíše:

```
>>> oblubujem(1)
Mám rád číslo 1
```



```
>>> oblubujem(5)
Mám rád číslo 5
>>> oblubujem(10)
Číslo 10 sa mi nepáči
```

7. Použi cyklus a zavolaj podprogram oblubujem z predchádzajúcej úlohy pre čísla od 0 po 10:

```
Mám rád číslo 0
Mám rád číslo 1
...
Mám rád číslo 6
Číslo 7 sa mi nepáči
...
Číslo 10 sa mi nepáči
```

8. Poznáš hru *Myslím si číslo*, v ktorej treba uhádnuť neznáme číslo? Vytvor takú hru na počítači – počítač si vymyslí číslo od 1 po 5 a my ho musíme uhádnuť. Môže to fungovať takto:

- o po spustení programu počítač priradí do premennej číslo náhodne vygenerované číslo,
- o potom vypíše správu Myslím si číslo od 1 do 5. Skús ho uhádnuť...
- o vo svojom programe budeš mať zadaný podprogram skus s parametrom n, ktorý porovná číslo s hodnotou n a vypíše: buď Hurá, uhádol si!, alebo Nie, moje číslo je iné....

Hra môže prebiehať nasledovne – spustíme program a v príkazovom riadku odpovedáme tým, že voláme podprogram skus:

```
===== RESTART =====
Myslím si číslo od 1 do 5. Skús ho uhádnuť...
>>> skus(3)
Nie, moje číslo je iné...
>>> skus(2)
Nie, moje číslo je iné...
>>> skus(5)
Hurá, uhádol si!
>>>
```

9. (\*) Vylepši predchádzajúci program tak, aby nám podprogram skus poradil:

```
===== RESTART =====
Myslím si číslo od 1 do 5. Skús ho uhádnuť...
>>> skus(3)
Nie, moje číslo je menšie...
>>> skus(1)
Nie, moje číslo je väčšie...
>>> skus(2)
Hurá, uhádol si!
>>>
```

10. Vytvor jednoduchý kvíz na sčítanie čísel. Počítač na začiatku vygeneruje dve náhodné čísla z rozsahu od 1 po 10, vypíše ich a my musíme odpovedať tým, že zavoláme podprogram over:

```
===== RESTART =====  
Kolko je 10 + 7 ?  
>>> over(5)  
Nesprávne, malo to byť 17  
>>>  
===== RESTART =====  
Kolko je 4 + 9 ?  
>>> over(13)  
Správne  
>>>
```

11. (\*) Treba preskúmať, ako často padne 6, keď veľa krát hádzeme kockou. Vytvor podprogram stacie s parametrom n, ktorý odsimuluje n hodov s bežnou hracou kockou. Podprogram n-krát vygeneruje náhodné číslo od 1 po 6, a ak padne šestka, zvýši počítadlo o 1. Podprogram na záver vypíše správu v tvare, napríklad:

Pravdepodobnosť výhry pri 10 hodoch: 0.2

Nechaj podprogram vypísať, aké budú pravdepodobnosti výhry pri 10, 100, 1000, 10000, 100000 hodoch.

© Copyright 2019, Andrej Blaho, Ľubomír Salanci.  
Naposledy aktualizované 10. sep. 2020.  
Prispôsobenia pre nevidiacich vykonala Ľudmila Jašková, august 2023.  
Vytvorené pomocou [Sphinx](#) 3.2.1.

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).