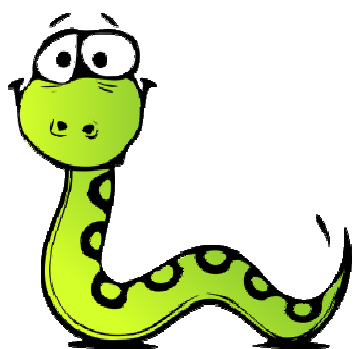




Programování v jazyce Python pro střední školy

Metodický list pro učitele
Lekce 3 – Program



Andrej Blaho
Lubomír Salanci
Václav Šimandl

Cíle lekce

- Naučit se používat příkaz `print`, který je určený na výpis textů a číselných hodnot
- Pochopit, co je to program, jak se zapisuje a vykonává
- Pochopit, že texty jsou jiné než čísla
- Seznámit se s různými chybovými situacemi v programech

Dovednosti

- Vytvoření a uložení programu
- Vícenásobné spouštění programu
- Přepínání se mezi oknem editoru a interaktivním oknem (konzolí)

Osvojená syntaktická pravidla

- Zápis příkazu `print` a jeho parametrů
- Zápis textů pomocí apostrofů nebo uvozovek

Průběh výuky

První úloha slouží k zopakování poznatků z minulé lekce.

1. Do proměnné `suma` přiřaď počet korun, které budeš měnit na eura. Do proměnné `kurz` přiřaď kurz eura (například 1 euro za 25.23 korun). Do proměnné `dostanes` přiřaď hodnotu výrazu, kterým se vypočítá, kolik eur dostaneš za měněnou sumu. Začni takto:

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


Očekávané řešení:

```
>>> suma = 100
>>> kurz = 25.230
>>> dostanes = suma / kurz
>>> dostanes
3.963535473642489
```

Doposud se zadávaly příkazy samostatně do příkazového řádku v interaktivním režimu. To bylo výhodné, pokud žáci Python používali jako inteligentní kalkulačku, případně se seznamovali s konceptem proměnné a dělali drobné experimenty. V této lekci chceme naučit žáky sestavovat programy. Aby program dokázal zobrazovat výsledky, je potřeba žáky seznámit s příkazem `print`. Začínáme jednoduchými výpisy, stále v interaktivním režimu.

2. Vyzkoušej nový příkaz. Co vykoná?

```
>>> print('Ahoj, já jsem počítač')
```


toto jsou apostrofy – najdi je na klávesnici

Příkaz `print` slouží na **vypisování** textů. Text, který se má vypsát, napíšeš mezi apostrofy.

Zatím se ti může zdát vypisování textů pomocí příkazu `print` zbytečné – vždyť jen vypsals na nový řádek text, který jsi napsal mezi apostrofy. Jak ale uvidíš v dalších úlohách, bude tento příkaz velmi užitečný při vytváření programů.

V těchto materiálech budeme texty uvádět v apostrofech, ačkoliv stejně tak bychom mohli používat uvozovky – tj. `print("Ahoj, já jsem počítač")`. Nechceme žáky zatěžovat oběma variantami. Rozhodli jsme se používat apostrofy, protože ty se v jazyce Python používají obvykle. U žáků nemusíme trvat na používání apostrofů, pokud by chtěli používat uvozovky. Při psaní těchto příkazů mohou mít žáci zpočátku problémy s použitím správných symbolů, neboť na české klávesnici existují různé typy znaků, které vypadají jako apostrofy. Žáci nemusí vypisovat texty s diakritikou – budeme spokojeni například i s výpisem `print('Ahoj, já jsem pocitac')`.

Se žáky můžeme diskutovat, co znamená anglické slovo `print`, i o tom, že v jazyce Python má tento příkaz význam *vypsání* do interaktivního okna.

Následuje úloha na trénování:

3. Použij příkaz `print` a vypiš pomocí něho jména dvou svých kamarádů, například:

Moji kamarádi jsou Vašek a Jana.

Řešení:

```
>>> print('Moji kamarádi jsou Vašek a Jana.')  
Moji kamarádi jsou Vašek a Jana.
```

Cílem následujících experimentů je, aby se žáci lépe seznámili s příkazem `print`:

4. Zjisti, co Python vypíše v případě následujících příkazů:

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

Vidíš, že `print` vypíše i hodnotu výrazu, která není mezi apostrofy.

Žáci by měli přijít na následující poznatky:

- Obyčejné výrazy v příkazu `print` se vyhodnotí a zobrazí se jejich hodnota:

```
>>> print(1 + 2 * 3)
7
```

- To, co je v apostrofech, počítač nevyhodnocuje, i kdyby to byl standardní výraz:

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

- V příkazu `print` nemusíme uvést, co chceme vypsat – pak se zobrazí prázdný řádek:

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

Upozornění pro učitele: slovo `print` lze napsat i bez závorek, jen jako slovo:

```
>>> print
<built-in function print>
```

Potom Python zobrazí informaci, že `print` je standardní příkaz (funkce) jazyka Python.

Terminologická poznámka: Ačkoliv `print` označujeme v naší učebnici jako příkaz, je to ve skutečnosti funkce. Podobně označujeme v dalších kapitolách jako příkazy i další funkce či metody. K tomuto zjednodušení jsme se uchýlili záměrně, neboť nechceme začátečníky touto problematikou (např. rozdíly mezi příkazy, funkcemi a metodami) zatěžovat.

Cílem následujících úloh je, aby žáci zjistili, že v příkazu `print` mohou uvést více textů a hodnot, které se postupně za sebou vypíší (přitom platí, že jeden příkaz `print` se rovná jednomu řádku výpisu). Tento způsob výpisu chceme použít proto, aby s námi Python srozumitelně komunikoval:

5. Příkaz `print` umí vypsat víc věcí – vyzkoušej následující příkazy. Co způsobí čárka v jednotlivých příkazech?

```
>>> print('Mám rád', 'kapustu')
>>> print('Moje oblíbené číslo je', 42)
>>> print('Do školy jsem šel', 2 * 10, 'minut')
```

Žáci by si měli všimnout, že:

- Výsledný text bude na jednom řádku, mezi vypisované texty se vloží mezera:

```
>>> print('Mám rád', 'kapustu')
Mám rád kapustu
```

- Můžeme kombinovat texty i hodnoty (a mezi vypisované části se též vloží mezera):

```
>>> print('Moje oblíbené číslo je', 42)
Moje oblíbené číslo je 42
```

- Když uvedeme výraz, ten se vyhodnotí a na obrazovku se vypíše výsledek:

```
>>> print('Do školy jsem šel', 2 * 10, 'minut')
Do školy jsem šel 20 minut
```

Upozornění pro učitele: je potřeba dávat pozor na čárky – pokud chybí, zobrazí se chybové hlášení:

```
>>> print('Moje oblíbené číslo je' 42)
SyntaxError: invalid syntax
```

Následuje úloha na trénování:

6. V Anglii se délka neměří v centimetrech, ale v palcích. Jeden palec je přibližně 2.5 centimetru. Vypiš pomocí příkazu `print` svoji výšku v palcích podobně jako v následující ukázce:

Měřím 72.0 palců

Výšku v palcích nepočítej na kalkulačce, ale vytvoř v příkazu `print` výraz, ve kterém uvedeš svou výšku v centimetrech a podle kterého Python výpočet provede.

Řešení:

```
>>> print('Měřím', 180 / 2.5, 'palců')
Měřím 72.0 palců
```

Další úlohy jsou zaměřené na chybové zprávy, aby se žáci postupně naučili, co znamenají a kdy se mohou vyskytnout:

7. Prozkoumej, co se stane, když zapomeneš napsat v příkazu:

```
>>> print('Ahoj)           ... apostrof?
>>> print(Ahoj)           ... oba apostrofy?
>>> print 'Ahoj'          ... závorky?
>>> print('Ahoj' 10)       ... čárku?
```

Žáci by měli vidět následující chybová hlášení:

- Chybějící apostrof – neukončený text:

```
>>> print('Ahoj)
SyntaxError: EOL while scanning string literal
```

- Chybějící oba apostrofy – `Ahoj` je chápáno jako proměnná, do které jsme ale nepřiradili hodnotu:

```
>>> print(Ahoj)
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    print(Ahoj)
NameError: name 'Ahoj' is not defined
```

- Chybějící závorky:

```
>>> print 'Ahoj'
SyntaxError: Missing parentheses in call to 'print'
```

- Chybějící čárka:

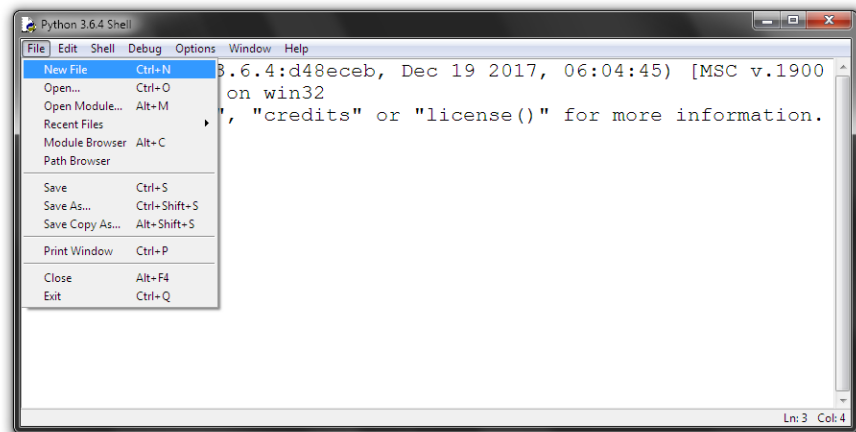
```
>>> print('Ahoj' 10)
SyntaxError: invalid syntax
```

Se žáky můžeme o chybách a jejich významu diskutovat (význam v češtině). Ze znalosti chybových zpráv však žáky nezkoušíme.

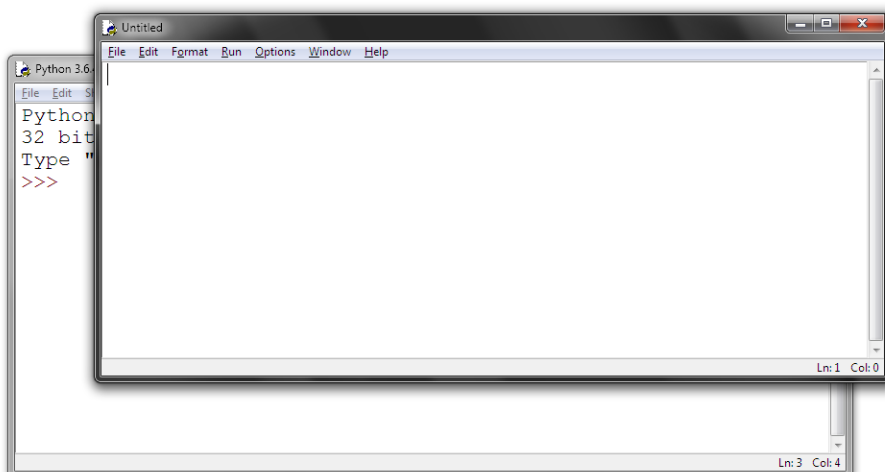
Nyní se žáci seznámí s postupem, jak se vytváří nový program. V budoucnu budou tento postup často opakovat.

8. Zatím jsi s Pythonem pracoval v interaktivním režimu. Za symboly `>>>` jsi zapisoval jednotlivé příkazy, které se ihned vykonaly. Dále budeš vytvářet programy – nejdříve napíšeš všechny příkazy, až potom tento program spustíš.

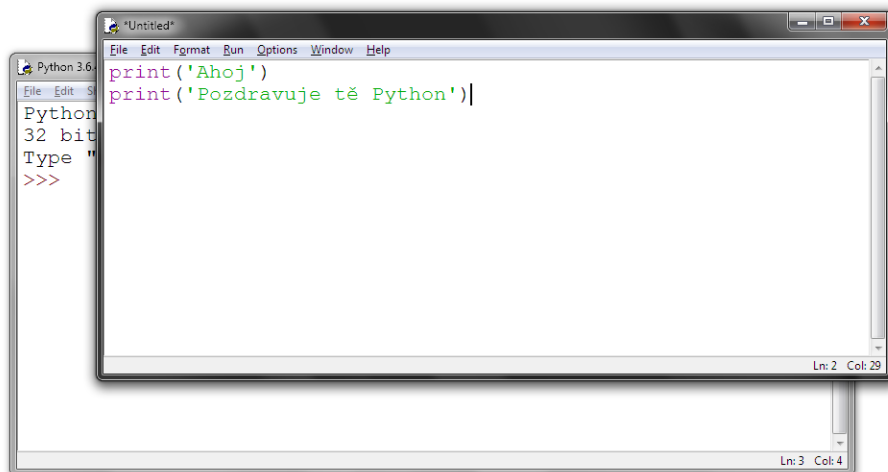
A. Z hlavní nabídky zvol *File* ► *New File*:



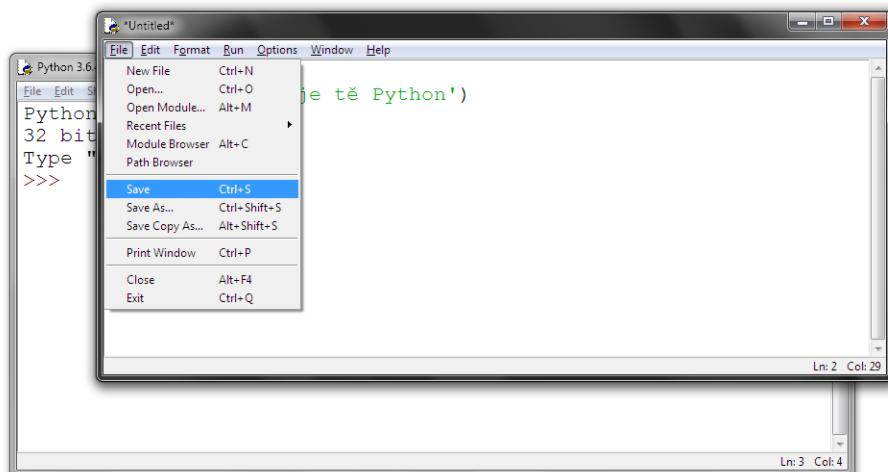
B. Otevře se nové okno, ve kterém budeš zapisovat program:



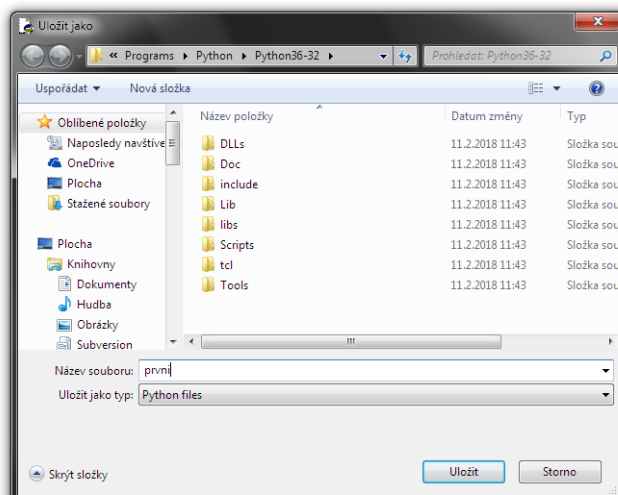
C. Do **okna s programem** napiš tyto příkazy:



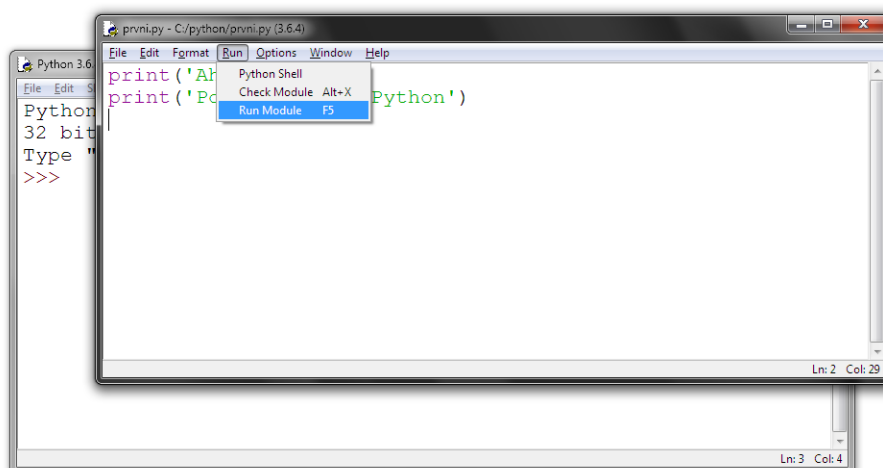
D. Ulož program do souboru volbou z nabídky *File* ► *Save*:



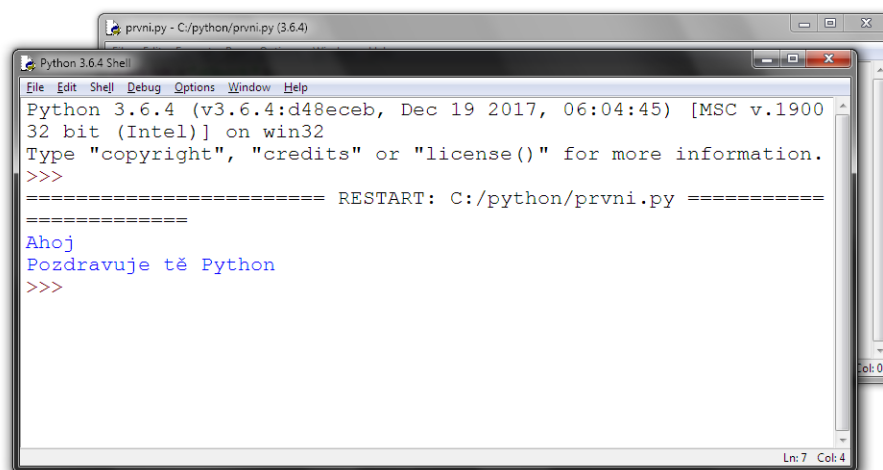
E. Najdi složku, do které chceš soubor uložit. Potom do políčka *File name* napiš název souboru, například `první.py` (příponu psát nemusíš), a klikni na tlačítko *Save*:



F. Nyní program **spust'** volbou *Run ► Run Module*:



G. V interaktivním okně uvidíš výsledek:



Je potřeba si uvědomit, že žáci mohou mít následující problémy:

- Od kroku B uvidí žáci více oken. Problém může být v orientaci, ve kterém okně se právě nachází, ve které hlavní nabídce mají hledat potřebné položky (např. kde spustit program).
- Najít místo, kam soubor uloží (to záleží na nastaveních v počítačové učebně).
- Do kterého okna se od této chvíle píše program a do kterého se zobrazují výpisy pomocí příkazu `print`.
- Kam a jak se budou zobrazovat chyby v programu.

Zamyšlení:

- Žáci budou rádi, když se seznámí s klávesovými zkratkami `F5` pro spuštění programu, `Ctrl+N` pro nový program, `Ctrl+S` pro uložení souboru. Z našeho pohledu jsou toto pro některé žáky užitečné pomůcky, ale určitě žáky ze znalosti klávesových zkratk nezkoušíme.
- Když změníme program a neuložíme jej (`Ctrl+S`), Python se před spuštěním dotáže na uložení souboru. Toto může být časem obtěžující. Pokud uznáte za vhodné, nastavte

prostředí Python v počítačové učebně tak, aby byl program před spuštěním automaticky uložen do souboru:

- v hlavní nabídce zvolte *Options ► Configure IDLE*
- v dialogovém okně, které se zobrazí, zvolte záložku *General*
- v části *Editor Preferences* zvolte volbu s názvem *No Prompt*

V rámci učebnice budou žáci vytvářet řadu programů a časem se složka, do níž ukládají soubory se svými programy, stane nepřehlednou. Je proto vhodné jim už nyní doporučit systematické ukládání souborů s programy do složek. To může vypadat například následovně:

- ve vhodném umístění vytvořit složku *Python*
- v této složce *Python* vytvořit podsložku *lekce03* a do ní ukládat všechny programy vytvořené v této lekci
- na začátku 4. lekce vytvořit ve složce *Python* podsložku *lekce04* a do ní ukládat všechny programy vytvářené ve 4. lekci
- obdobně pokračovat ve všech dalších lekcích

Při plnění dalších úloh, v nichž je vytvářen program, mohou někteří žáci psát kód programu omylem přímo do interaktivního okna. V tom případě je vhodné je na chybu individuálně upozornit a navést je, aby si pro program vytvořili nový soubor pomocí volby *File ► New File*.

Cílem následující úlohy je, aby si žáci natrénovali postup při editování a spouštění programu.

9. Přepni se zpět do svého programu a přidej další příkaz `print`, kterým vypíšeš text „Dnes je středa“ (místo středy doplň aktuální den v týdnu). Program ulož a spust' jej.

Řešení:

```
print('Ahoj')
print('Pozdravuje tě Python')
print('Dnes je středa')
```

Žáci by si měli uvědomit, že vytvořený program lze (na rozdíl od příkazů zapisovaných do interaktivního okna) dodatečně upravovat a opakovaně spouštět. Zároveň sbírají zkušenosti, jak se vykonává program, který obsahuje více příkazů za sebou. Zjistí, že počítač příkazy vykoná postupně tak, jako když my čteme text.

Cílem následujících úloh je trénink vytváření nových programů, používání příkazu `print` a spouštění, případně opravování programů.

10. Vytvoř program `basnicka.py`, který vypíše úryvek tvé oblíbené básničky nebo písničky. Jestli tě žádná nenapadá, můžeš vypsát tuto básničku:

```
Na topole nad jezerem  
seděl vodník podvečerem:  
Sviť, měsíčku, sviť,  
ať mi šije niť.
```

Řešení:

```
print('Na topole nad jezerem')  
print('seděl vodník podvečerem:')  
print('Sviť, měsíčku, sviť,')  
print('ať mi šije niť.')
```

Úlohu lze řešit i za použití jediného příkazu `print`, jak uvádíme u obdobné úlohy v metodickém listu 12. lekce. Taková řešení jsou založena na technických fintách a nejsou příliš přehledná, a proto je žákům neprozrazujeme.

Vytvoření nového programu lze pojímat dvěma způsoby:

- Žák v hlavní nabídce zvolí *File ► New File* a bude vytvářet zcela nový program.
- Žák v okně s existujícím programem zvolí *File ► Save As...* a bude upravovat tento již existující program.

Je na učitelově uvážení, který přístup žákům doporučí. Při úpravách existujícího programu hrozí, že se žáci budou snažit co nejvíce kódu z předchozí úlohy použít v té aktuální. Pokud bude aktuálně řešená úloha značně odlišná od předchozí, může být tato snaha kontraproduktivní a žáky zdržovat. Jako vhodné se tak může jevit vytvoření zcela nového programu, do něž si žáci v případě potřeby zkopírují kratší kusy kódu z již existujícího programu. Tento postup bude užitečný zejména v dalších lekcích při práci s grafickou plochou, kde se v každém programu používá několik stále se opakujících příkazů.

11. Pomocí příkazu `print` se dají vypisovat veselé věci. Vytvoř nový program `vizitka.py`, který pomocí příkazu `print` vypíše tvoji vizitku, například takovouto:

```
+-----+  
|               www               |  
| Petr      ( o o ) |  
| LEV       ( ~ )  |  
|           "       |  
| Počítačový král  |  
+-----+
```

Možné řešení:

```
print('+-----+')
print('|          www          |')
print('| Petr      ( o o )      |')
print('| LEV       ( ~ )        |')
print('|              "         |')
print('| Počítačový král       |')
print('+-----+')
```

V této úloze:

- Předpokládáme, že si žáci vytvoří svůj obrázek
- Žáci musí vytvořit nový program, vhodně jej nazvat a uložit,
- Když žáci použijí ve své vizitce nějaké speciální znaky, ty mohou dělat problémy, například:
 - Pokud by ve výpisu chtěli použít apostrof (nebo v příkazu `print` používají na označení textu uvozovky a i ve vypisovaném textu by chtěli použít uvozovky),
 - Zpětné lomítko `\` se v textu používá na zadávání speciálních symbolů. Pokud je za znakem mezera, je to v pořádku, ale například kombinace `print('\x0e')` vypíše `␣`.

12. Vytvoř program `python.py`, který ze znaků `%` (procenta) vypíše zvětšený text PYTHON:

```
% % % % % % % % % %
% % % % % % % % % %
% % % % % % % % % %
% % % % % % % % % %
% % % % % % % % % %
```

Řešení:

```
print('% % % % % % % % % %')
print('% % % % % % % % % %')
print('% % % % % % % % % %')
print('% % % % % % % % % %')
print('% % % % % % % % % %')
```

13* Vytvoř program `prezdivka.py`, který ze znaků `#`, `$`, `€` nebo jiných vypíše tvoje jméno nebo přezdívkou. Výška písmen bude nejméně pět řádků.

Možné řešení:

```
print('€ € €€ € € €€€')
print('€€ €€ € € € €')
print('€ € € € €€ €')
print('€ € € € € €')
print('€ € €€ € € €€€')
```

Při řešení úkolu mohou mít žáci potíže se zápisem uvedených speciálních znaků. Žákům můžeme individuálně poradit, že na Internetu existují weby obsahující seznamy těchto speciálních znaků včetně informací, jak je zapsat na české klávesnici. Nám se osvědčil například web <http://znakynaklavesnici.cz/>

Ze znalosti způsobů zápisu speciálních znaků žáky v žádném případě nezkoušíme.

Zatím jsme úmyslně vypisovali jen texty a konstantní výrazy. Výrazům s proměnnými se budeme věnovat v další lekci.