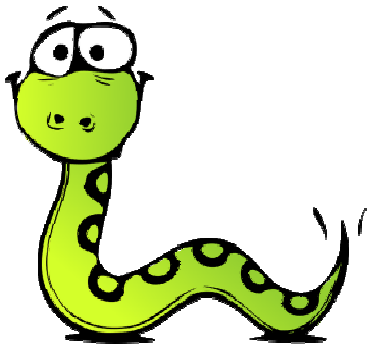
Programování v jazyce Python pro střední školy

Metodický list pro učitele

Lekce 1 – Výrazy



Andrej Blaho

Ľubomír Salanci

Václav Šimandl

Cíle lekce

- Seznámit se s číselnými typy (celá i desetinná čísla) a s aritmetickými operacemi
- Naučit se správně vyhodnocovat výrazy i s různou prioritou operací
- Seznámit se s chybovými hlášeními od počítače (např. `SyntaxError`, `ZeroDivisionError`)

Dovednosti

- Spuštění prostředí **IDLE**

Osvojená syntaktická pravidla

- Správný zápis čísel, operací a závorek

Průběh výuky

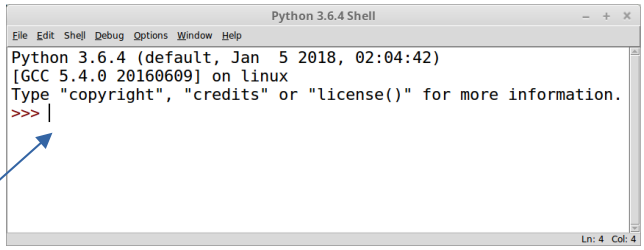
Před výukou je potřeba mít v počítačových učebnách nainstalovaný a otestovaný Python verze 3.6.0 nebo novější. Je nutné jej nainstalovat tak, aby byla ikona programu lehko přístupná (například na Ploše).

Stačí základní instalace prostředí IDLE (nebudou potřeba žádná jiná vývojová prostředí – PyCharm, Visual Studio, Eclipse a podobně). Toto prostředí se nám zdá jednoduché a pro potřeby vyučování postačuje. Výhodou je, že žáci si toto prostředí mohou sami nainstalovat a používat i doma. Naše úlohy by samozřejmě fungovaly i v jiných vývojových prostředích.

Doporučujeme **nedělat** teoretické úvody do programování (například nevysvětlovat, co je program, procesor, vlastnosti algoritmů a podobně). Je vhodné, aby žáci ihned pracovali v prostředí Python a prakticky se seznamovali se zápisy programů a reakcemi počítače.

1. Najdi na počítači ikonu programu Python a spusť jej. 

Když se program spustí, uvidíš:



sem budeš zapisovat příkazy

Nechme žáky nalézt a spustit programovací prostředí. Dále je nechme experimentovat s následujícími úlohami, čímž se budou seznamovat s jazykem Python jako s kalkulačkou v interaktivním režimu. Cílem je, aby si žáci zvykali na komunikaci s počítačem – to znamená, že po zápisu dostávají odpověď nebo chybovou zprávu a potom zapisují další příkazy.

2. Zkus za `>>>` napsat matematický výraz `1 + 2 + 3` a potvrď klávesou *Enter*. Co Python odpoví?

3. Python dokáže fungovat jako kalkulačka. Jaké budou výsledky následujících výrazů?

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

V předchozí úloze jsou důležité experimenty s operátory, závorkami a prioritami operací. Zápisy jsme sestavili jako gradované – je dobré, aby je žáci sami postupně vyzkoušeli. Mezery okolo operátorů není potřeba psát, ale když si žáci na jejich psaní zvyknou, program se zpřehlední („provzdušní“) a navíc takto programy zapisují i profesionálové.

Zde už mohou žáci tvořit chybné zápisy, případně nemusí rychle porozumět, jak počítač výraz vyhodnotil. Pokud to bude potřeba, můžeme jim individuálně vysvětlit, jak se výraz vyhodnotil. Cennější však je, když žáci budou tyto poznatky sami objevovat (nedoporučujeme řešit tyto úlohy dopředu na tabuli).

Žáci by se měli naučit rozpoznávat i situace, jak počítač zareaguje na nesprávné výrazy. Proto následují úlohy, v nichž úmyslně zadáváme nesprávné zápisy:

4. Pozor, zápisy musí být napsané zcela správně. Jinak uvidíš různá chybová hlášení. Co se stane, pokud zadáš následující příkazy?

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

5. Někdy se však i po správném zápise může objevit chybové hlášení. Co se stane, pokud zadáš `10 / (6 - 2 * 3)` ?

Python se ti chybovými hlášeními snaží pomoci, abys chybu snadněji našel. Například:

`SyntaxError: invalid syntax` označuje, že jsi něco napsal nesprávně
`ZeroDivisionError: division by zero` oznamuje, že chceš dělit nulou

Při chybných výrazech počítač někdy vypíše několik řádků s chybou. Proto je nutné se učit rozpoznávat, co je v chybovém hlášení důležité (většinou je to poslední řádek takové zprávy). Zároveň by si měli žáci zvyknout, že v interaktivním režimu zapsaný příkaz, který potvrdili klávesou *Enter*, nelze dodatečně upravit nebo smazat. Pokud jej tedy chtějí upravit, musí jej napsat znovu.

Následuje několik slovních úloh. Cílem je, aby žáci sami sestavili správný výraz, resp. zrealizovali kroky výpočtu pomocí počítače.

6. Petrovi bylo přesně před dvěma měsíci 16 let. Využij Python jako kalkulačku a spočítej, kolik je mu nyní přibližně dní. Předpokládej, že rok má 365 dní a měsíc má 30 dní.

Očekáváme, že žáci sestaví výraz `16 * 365 + 2 * 30` a nechají jej vyhodnotit počítačem.

V této úloze je vhodné žáky vést, aby ještě před sestavením výrazu v Pythonu přibližně odhadli výsledek výpočtu. Díky tomu budou schopni posoudit, zda může být Pythonem zobrazený výsledek správný či nikoliv. Pokud by zobrazený výsledek posoudili jako nesprávný, měli by sami začít hledat chybu v zápisu příkazu. Podobný přístup je vhodný taktéž u dalších úloh a úloh v dalších lekcích.

7. Pokračuj v předchozí úloze a pomocí Pythonu vypočítej:

   a) kolik je to hodin,
   b) kolik je to sekund.

Možné řešení:

a) `5900 * 24`
b) `141600 * 60 * 60`

8. Použij znovu Python jako kalkulačku a vytvoř pro něj zápis, pomocí kterého vypočítá součet všech lichých čísel od `1` do `19`. Jaký bude výsledek?

Možné řešení: `1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 + 17 + 19`

9. Zjisti, která číslice se vyskytuje nejčastěji ve výsledku výrazu:

   ```
   123456789 * 11111111111111111111
   ```

   Nejčastější číslici snadno poznáš pohledem na výsledek spočítaného součinu.

Cílem této úlohy je, aby žáci viděli, že celočíselná aritmetika umožňuje pracovat s velkými čísly. Žáci si mohou vyzkoušet práci i s jinými velkými čísly.

10. Lenka sbírala květiny. První den jich natrhala 15, druhý den jich natrhala o 4 více než předcházející den a třetí den jich natrhala ještě o 1 více než v oba předcházející dny dohromady. Použij Python jako kalkulačku a vypočítej, kolik květin natrhala za všechny 3 dny dohromady.

Možné řešení: `15 + 15 + 4 + 15 + 15 + 4 + 1`

Zde je vhodné upozornit žáky na to, že po použití závorek bude výraz čitelnější – budeme lépe vidět počty květin pro jednotlivé dny:

```
15 + (15 + 4) + (15 + 15 + 4 + 1)
```

Cílem další úlohy je, aby žáci použili závorky. Třetí podúloha je označená * což znamená, že je určená pro žáky, kteří mají chuť nad úlohou déle přemýšlet:

11. Jirka si koupil hru za 79 korun. Později si koupil hru za dvojnásobek této ceny a ještě k tomu připlatil 5 korun. Nakonec si koupil hru za trojnásobek ceny druhé hry a ještě k tomu připlatil 17 korun. Použij Python jako kalkulačku a vypočítej:

    a) kolik Jirka zaplatil za třetí hru
    b) kolik Jirka zaplatil za všechny tři hry dohromady
    c*) vymysli co nejkratší zápis, kterým lze úkoly z a) a b) vypočítat

Postup při promýšlení řešení:

první hra: `79`
druhá hra: `79 * 2 + 5`
třetí hra: `(79 * 2 + 5) * 3 + 17`

Možné řešení:

a) `(79 * 2 + 5) * 3 + 17`
b) `79 + 79 * 2 + 5 + (79 * 2 + 5) * 3 + 17`
c) `79 + (79 * 2 + 5) * 4 + 17` nebo `79 * 9 + 5 * 4 + 17`

12. Použij Python jako kalkulačku a vypočítej součet následujících čísel: jedna, jedna polovina, jedna třetina, jedna čtvrtina, …, až jedna desetina

Možné řešení:

```
1 + 1/2 + 1/3 + 1/4 + 1/5 + 1/6 + 1/7 + 1/8 + 1/9 + 1/10
```

Následující úlohu je dobré řešit společně:

13. Do sešitu si vytvoř tabulku, do níž zapiš všechny aritmetické operace, se kterými jsme se zatím v Pythonu seznámili.

Je možné sestavit takovouto tabulku, kterou si žáci mohou zaznamenat do sešitu:

+	Sčítání
−	Odečítání
*	Násobení
/	Dělení

Následující úlohy nejsou povinné – jsou určené žákům, kteří by o ně měli zájem. V úlohách se žáci seznamují se zápisem pro operaci umocnění.

14* Výpočet `2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2 * 2` je umocnění `2` na `10`. V Pythonu se toto zapisuje jako: `2 ** 10`. Tipni si, kolik číslic bude ve výsledku umocnění `2` na `30`. Poté použij Python jako kalkulačku, vypočítej pomocí něho `2` umocněno na `30` a ručně spočítej počet číslic ve výsledku. Byl tvůj odhad správný?

Řešení: `2 ** 30`

Počet číslic lze zjistit i programově – to však žáci ještě nevědí. Proto stačí číslice v této úloze spočítat manuálně. Ačkoliv je exponent `30`, počet číslic čísla 2^{30} bude překvapivě malý – jen `10`.

Následující úloha má výzkumný charakter a je zaměřená na prioritu operací:

15* Zjisti, jak se počítá hodnota `2 ** 8 - 1`. Tedy zda se nejdříve vypočítá mocnina `2 ** 8`, od které se odečte `1`, nebo se nejdříve vypočítá rozdíl `8 - 1` a touto hodnotou se potom umocní číslo `2`. Zjisti, jak je to s operacemi násobení a umocňování – tedy jak se počítají výrazy `3 * 2 ** 5` a `2 ** 5 * 3`.

Očekávané zjištění: umocňování má nejvyšší prioritu z doposud známých operací (unární mínus jsme zatím neučili), násobení a dělení má menší prioritu a nejnižší má sčítání a odečítání. Při zkoumání můžeme žákům pomoci radou, aby si všimli výsledků, když použijí závorky: `(3 * 2) ** 5` oproti `3 * (2 ** 5)`. Při vysvětlování, pokud to bude potřebné, žákům můžeme prozradit, že počítač si výrazy s různými operacemi také závorkuje.

16* Matematici vědí, že když sečtou několik za sebou jdoucích mocnin čísla `2` počínaje `2` na `0`, dostanou jinou mocninu čísla `2` zmenšenou o `1`. Zkontroluj, zda součet čísel `2 ** 0, 2 ** 1, 2 ** 2, … 2 ** 9` dává hodnotu `2 ** 10 - 1`.

Je třeba napsat a nechat vyhodnotit zápisy:

```
2 ** 10 - 1
2**0 + 2**1 + 2**2 + 2**3 + 2**4 + 2**5 + 2**6 + 2**7 + 2**8
+ 2**9
```