

Domácí projekty 4

Tohle je čtvrtá sada projektů. Udělej si Gitový repozitář, do kterého si ukládej řešení. Jakmile ti bude (u programovacích úloh) nějaké řešení fungovat, ulož si ho jako revizi v Gitu. (Občas je to připomenuto i u jednotlivých projektů.)

Na začátek trochu zdánlivě nudného opakování, ať si trochu osvěžíme programování. To ale neznamená, že tu nemůžeš najít něco zvláštního a překvapivého!

0. Co dělá funkce `print`?
1. Co *vrací* funkce `print`?
2. Co dělá pojmenovaný argument `end` funkce `print`?
3. Co dělá pojmenovaný argument `sep` funkce `print`?
4. Čím se liší chyby, které dostaneš, když zadáš tyhle příkazy?
`int('blabla')`
`float('blabla')`
`int('8.9')`
`int(8.9)`

Následující sada projektů není jednoduchá, ale uděláš-li ji, pochopíš různá použití cyklu `for`.

5. Pomocí cyklu `for` a funkce `range()` napiš program, který vypíše:

```
a
a
a
a
```

Až to bude fungovat, dej to do Gitu!

6. Pomocí cyklu `for` napiš program, který vypíše toto:

```
Řádek 0
Řádek 1
Řádek 2
Řádek 3
Řádek 4
```

Funguje? Dej to do Gitu!

7. Jak jsi pojmenovala proměnnou, kterou jsi v minulém příkladu použila? Vymysli pro ni název, který nejlépe vystihuje, co proměnná obsahuje.

Je důležité proměnnou pojmenovat výstižně, jinak se v dalších projektech můžeš ztratit. Možné řešení je uvedeno na konci zadání.

8. Pomocí cyklu `for` napiš program, který vypíše:

```
0 na druhou je 0
1 na druhou je 1
2 na druhou je 4
3 na druhou je 9
4 na druhou je 16
```

Jak pojmenuješ proměnnou cyklu?

9. Pomocí cyklů `for` a parametru `end` pro `print` napiš program, který postupně z jednotlivých 'X' vypíše:

```
X X X X X
X X X X X
X X X X X
X X X X X
X X X X X
```

„Z jednotlivých 'X'“ znamená, že nepoužiješ např. `print('X X X X X')`.

Jak pojmenuješ proměnnou cyklu? A tu druhou?

10. Napiš program, který vypíše „tabulku“ s násobilkou:

```
0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12
0 4 8 12 16
```

Funguje? Dej to do Gitu!

11. Napiš program, který postupně z jednotlivých 'X' vypíše:

```
X
X X
X X X
X X X X
```

Funguje? Do Gitu s tím!

12. Pomocí cyklu `for` a příkazu `if` napiš program, který vypíše následující řádky. Funkci `print` volej pouze uvnitř v cyklu:

```
první řádek
není první
není první
není první
```

13. Pomocí cyklů `for` a příkazu `if` napiš program, který z jednotlivých 'X' a mezer vypíše:

```
X X X X X X
X           X
X           X
X           X
X X X X X X
```

14. Programy s cyklem `for` uprav tak, aby počet řádků (či velikost čtverce/trojúhelníku/tabulky) mohl zadat uživatel.

Funguje? Do Gitu s tím!

Následující sada projektů může vyžadovat delší zamyšlení. A to zamyšlení je důležitější než samotná odpověď.

15. Co dělá tenhle kód?

```
for c in 'Ahoj světe!':
    print(c)
```

16. Vymyslíš lepší jméno pro proměnnou `c` z minulé úlohy?

17. Co dělá tenhle kód?

```
for c in 38:
    print(c)
```

18. Už víš, co dělá `for` s `range()`, výčtem hodnot a řetězcem. Dokážeš to zobecnit, popsat `for` jednodušeji než jak je popsán v materiálech?

Ted'několik programovacích oříšků pro dlouhé chvíle. Nemáš-li čas, přeskoč je.

19. Napiš program, který se zeptá na 3 čísla a zjistí, jestli je jejich součet větší než 10.

Funguje? Do Gitu s tím!

20. Napiš program, který načte číslo a zjistí, jestli je sudé. *Sudá čísla jsou beze zbytku dělitelná dvěma.*

21. Napiš program, který vypíše čísla od jedné do 100, ale:

- Pokud je číslo dělitelné třemi, napíše místo něj „bum“.
- Pokud je číslo dělitelné pěti, napíše místo něj „bác“.
- Pokud je číslo dělitelné pěti i třemi zároveň, napíše místo toho „bum-bác“.

Funguje? Do Gitu s tím!

22. Máš-li ráda matematiku* a nebojíš-li se výzvy, načti od uživatele číslo n a:

- Vypočti faktoriál $n!$ (součin všech celých čísel od 1 do n).
- Zjisti, jestli je n prvočíslo.
- Vypiš prvních n členů Fibonacciho posloupnosti (1, 1, 2, 3, 5, 8, 13, 21, ...).

* t.j. nemáš-li ráda matematiku, neďalej tenhle projekt :)

A nakonec něco na oddech (snad)...

23. Tohle je poslední projekt ze čtvrté sady. Kolik je v této sadě projektů?

Možné řešení projektu 7: číslo_radku