
Intenzivní kurz Pythonu

Vydání 1.0

Petr Viktorin

20.03.2016

1	Informace pro kouče	3
1.1	Čas	3
1.2	Formát	3
1.3	Co s sebou	3
1.4	Jako koučovat	3
1.4.1	Usmívej se :)	3
1.4.2	Nesahej na klávesnici	4
1.4.3	Chceš říct něco víc?	4
1.4.4	Buď dobrý kouč	4
2	Den 1	5
2.1	[30min] Seznámení	5
2.1.1	Kdo jsme a proč tu jsme	5
2.1.2	Proč Python?	5
2.2	[30min] Kontrola instalace	5
2.2.1	Python	5
2.2.2	Editor	5
2.2.3	Git	5
2.2.4	Účet na GitHubu	5
2.3	[2h 30m] Základy programování v Pythoňu 1	6
2.3.1	Řetězce a <code>print()</code>	6
2.3.2	Typy objektů	6
2.3.3	Proměnné	6
2.3.4	Řetězce	6
2.3.5	Slovníky	6
2.4	[1h] Oběd	6
2.5	[2h 30m] Základy programování v Pythoňu 2	6
2.5.1	Program v souboru	6
2.5.2	Podmínky: <code>if</code>	6
2.5.3	Cykly: <code>for</code> a <code>while</code>	7
2.5.4	Ramena obrů: <code>import</code>	8
2.5.5	Funkce: <code>def</code>	8
2.5.6	Čtení souborů: <code>with</code>	9
2.5.7	Procvičení	9
2.6	[30min] Zadání projektu	9
2.6.1	Zadání domácího úkolu	9
2.6.2	Varianta 1: Webová aplikace	10
2.6.3	Varianta 2: Hra	10

2.6.4	Varianta 3: Nástroj pro příkazovou řádku	10
2.7	[30min] Poděkování a rozloučení	11
3	Den 2	13
3.1	[1h] Třídy	13
3.1.1	Chování vs. stav	13
3.1.2	Vlastní třídy	13
3.1.3	Atributy	13
3.1.4	<code>self</code>	13
3.1.5	<code>__init__</code>	13
3.1.6	Jak to využít v praxi	13
3.2	[2h] Git	14
3.2.1	Co to je Git	14
3.2.2	Začínáme: <code>git clone</code>	14
3.2.3	Zjištění stavu: <code>git status</code>	14
3.2.4	Přidávání souborů: <code>git add</code>	14
3.2.5	Vytvoření revize: <code>git commit</code>	14
3.2.6	Přehled revizí: <code>git log a gitk</code>	14
3.2.7	Nahrání změn: <code>git remote a git push</code>	14
3.2.8	Stažení změn: <code>git pull</code>	14
3.2.9	Konflikty a jak je řešit	14
3.2.10	Shrnutí	14
3.3	[30min] Oběd	15
3.4	[1h] Jak na webovou aplikaci	15
3.4.1	Instalace frameworku <code>flask</code>	15
3.4.2	Jak funguje Internet	15
3.4.3	Dotaz a odpověď ve Flasku	15
3.4.4	Základy HTML	15
3.4.5	Základy CSS	15
3.4.6	Formuláře a HTTP metody: GET, POST a PUT	15
3.4.7	Jak si uložit stav	15
3.4.8	Něco o bezpečnosti	15
3.5	[1h] Jak na hru	15
3.5.1	Instalace knihovny <code>pygame</code>	15
3.5.2	Smyčka událostí	16
3.5.3	Kreslení obrázků	16
3.5.4	Rotace	16
3.5.5	Vstup a reakce na něj	16
3.5.6	Základy herní fyziky	16
3.5.7	Výkon programu	16
3.6	[30min] Jak na program pro příkazovou řádku	16
3.6.1	Instalace knihovny <code>click</code>	16
3.6.2	Druhy programů pro příkazovou řádku	16
3.6.3	Konvence příkazové řádky	16
3.6.4	Jak na to s Clickem	16
3.7	[1h 30m] Další témata – podle vybraných projektů	17
3.7.1	(?) Dědičnost tříd	17
3.7.2	(?) Stahování informací z Webu: <code>requests</code>	17
3.7.3	(?) Webové API: <code>json</code>	17
3.7.4	(?) Práce s textem: <code>regex</code>	17
3.7.5	(?) Práce s tabulkovými daty: <code>csv</code>	17
3.7.6	(?) Základy numerických výpočtů: <code>numpy, scipy, matplotlib</code>	17
3.7.7	(?)	17
3.8	[30min] Zadání domácího úkolu; práce na projektu	17

4	Den 3	19
4.1	Práce na projektech	19
4.2	Doplňující témata	19
4.2.1	Nasazení	19

V dubnu rozjíždí Czechitas a Pyvec intenzivní kurz Pythonu. Tři soboty, od 10:00 do 18:00 (se) budeme učit programování – od úplných začátků až po hotový projekt. A potom se sejdeme na [Pyvu](#), a projekty představíme veřejnosti!

Na těchto stránkách vzniká rozvrh a obsah kurzu. Detaily ještě nejsou všude vyplněné, to budu dělat postupně :)

Jestli něco není jasné z osnovy, [dej vědět](#), abych tomu mohl nastavit prioritu :)

Informace pro kouče

Jdeš koučovat? Děkujeme! Tady je pár rad, jak na to.

1.1 Čas

Začínáme v deset, ale předtím ještě musíme rozdělit účastnice do skupinek, a případně dořešit organizační problémy, tak prosím buď na místě už v **9:30**.

Skončíme v šest. Přetahovat neplánujeme, ale radši počítej s tím, že se na cestu domů vydáš v **18:30**.

1.2 Formát

První a druhou sobotu bude „hlavní kouč“ vysvětlovat a zadávat úkoly. Ostatní koučové se budou věnovat svým skupinkám účastnic, odpovídat na otázky, nebo napovídat když nebudou vědět jak dál.

Třetí sobotu už bude program individuální: každá skupinka bude pracovat na svém projektu, pod vedením „svého“ kouče. Hlavní kouč bude k dispozici, kdyby si ostatní koučové nevěděli rady.

1.3 Co s sebou

Vezmi si s sebou laptop. Nejspíš nebude potřeba, ale je dobré ho mít.

1.4 Jako koučovat

Poznámka: Většina rad je převzata z [OpenTechSchool Coaching Guide](#).

1.4.1 Usmívej se :)

Když se účastnice zeptá na něco, co už mělo být jasné, znamená to, že my, koučové, jsme to špatně vysvětlili. Zkus to napravit a vysvětlí to znovu.

Když účastnice udělá popáté tu stejnou chybu, zhluboka se nadechni a znovu vysvětlí problém – nejlépe podrobněji než předtím.

1.4.2 Nesahej na klávesnici

Klávesnice účastnic jsou z lávy. Jakmile se dotkneš klávesnice, ztrácíš jeden život.

Je jasné, že umíš problém vyřešit rychleji – ale když ho vyřešíš sám, ztrácí to smysl. Jakmile se dotkneš klávesnice, účastnice přestane dávat pozor. Nech ji aby aspoň napsala, co jí nadiktuješ.

Když *opravdu* musíš něco napsat na její klávesnici, napřed se zeptej jestli je to v pohodě. A nezapomeň vysvětlit co děláš.

1.4.3 Chceš říct něco víc?

Dostaneš-li doplňující otázku, zkontroluj si, jestli máš čas na ni odpovědět. Intenzivní kurz některé detaily schválně vynechává, aby se všechno stihlo.

Když má účastnice vyřešenou aktuální úlohu a nic nového se nepřednáší, klidně vysvětluj, ale kdyby hrozilo že přestanete stíhat, shrň odpověď do co nejmenšího času.

A neříkej jim o věcech, které nejsou v materiálech, dokud několikrát za sebou neuvidíš, že by jim ulehčily život. Příliš detailů, zkratk a zlepšováků může začátečníky zahltnout.

1.4.4 Bud' dobrý kouč

Základní předpoklad: každá účastnice má nulové znalosti ale nekonečnou inteligenci.

Občas se zeptej, jestli všechno funguje a jestli je všechno jasné.

Když se něco povede, nešetři chválou!

Na chytré otázky je radost odpovídat; dej tuto radost najevo. Hloupé otázky neexistují.

Když je něco špatně v materiálech, ohlaš chybu. Nebo pošli pull request. Jestli na to nemáš čas hned, zapiš si to.

2.1 [30min] Seznámení

2.1.1 Kdo jsme a proč tu jsme

XXX

2.1.2 Proč Python?

XXX

2.2 [30min] Kontrola instalace

2.2.1 Python

`python --version`, případně `python3 --version`, musí ukázat Python 3.4 nebo Python 3.5.

2.2.2 Editor

Textový editor musí být nastavený tak, aby odsazoval o 4 mezery a ukazoval čísla řádků.

2.2.3 Git

`git --version` musí fungovat.

2.2.4 Účet na GitHubu

Účet musí být založený.

2.3 [2h 30m] Základy programování v Pytohnu 1

2.3.1 Řetězce a `print()`

XXX: Vede Nast'a, viz její notebook

2.3.2 Typy objektů

XXX: Čísla, řetězce, `bool`, seznamy, slovníky; převádění typů; operace

2.3.3 Proměnné

XXX: Vede Nast'a, viz její notebook

2.3.4 Řetězce

XXX: Vede Nast'a, viz její notebook

2.3.5 Slovníky

XXX: Vede Nast'a

2.4 [1h] Oběd

Ňam Ňam!

2.5 [2h 30m] Základy programování v Pytohnu 2

2.5.1 Program v souboru

```
print('Ahoj, světe!')
```

```
print('Ahoj, světe!')  
print(tady je chyba)
```

2.5.2 Podmínky: `if`

```
strana = float(input("Zadej stranu čtverce: "))  
if strana < 0:  
    print("Záporné čtverce neexistují")  
elif strana == 0:  
    print("Čtverec je prázdný")  
else:  
    print("Takový čtverec má obsah:", strana * strana)
```

```

cislo = int(input("Zadej číslo: "))
if cislo % 2 == 0:
    print('Číslo je sudé.')
else:
    print('Číslo je liché.')
if cislo % 5 == 0:
    print('Číslo je dělitelné pěti.')
else:
    print('Číslo není dělitelné pěti.')

```

```

cislo = float(input("Zadej číslo: "))
if cislo == int(cislo):
    if cislo % 2 == 0:
        print('Číslo je sudé.')
    else:
        print('Číslo je liché.')
else:
    print('Číslo není celé!')

```

Zkus napsat program, který se zeptá na dvě otázky:

- Jsi šťastná?
- Jsi bohatá?

a podle odpovědí odvěti:

- *šťastná i bohatá* – Gratuluji!
- *jen bohatá* – Zkus se víc usmívat!
- *jen šťastná* – Zkus míň utrácet!
- *ani jedno* – To je mi líto...

2.5.3 Cykly: for a while

```

for jmeno in 'Jana', 'Anna', 'Petra':
    print('Přichází', jmeno + '!')
    print(jmeno, 'zase odchází.')

```

```

for cislo in range(10):
    if cislo % 2 == 0:
        print('Číslo', cislo, 'je sudé.')
    else:
        print('Číslo', cislo, 'je liché.')

```

```

odpoved = input('Řekni Ááá! ')
while odpoved != 'Ááá':
    print('Špatně, zkus to znovu!')
    odpoved = input('Řekni Ááá! ')

print('Hotovo, ani to nebolelo!')

```

```

while True:
    print('počkej, než se počítač unaví...')

```

```
soucet = 0

for cislo in 2, 945, 24, 3:
    soucet = soucet + cislo

print(soucet)
```

Zkus napsat program, který sčítá čísla, která zadá uživatel. Když uživatel zadá nulu, program skončí.

2.5.4 Ramena obrů: import

```
from random import randrange

print('Házím kostkou...')

vysledek = randrange(6) + 1

print('Výsledek je', vysledek)
```

Zkus napsat program, který náhodně vybere a vypíše “kámen”, “nůžky”, nebo “papír”.

2.5.5 Funkce: def

```
def pozdrav():
    print('Ahoj!')
```

```
pozdrav()
```

```
def pozdrav(jmeno):
    print('Ahoj, ', jmeno + '!')
```

```
pozdrav('Lucko')
pozdrav('Aničko')
pozdrav('Terko')
```

```
def obsah_ctverce(strana):
    return strana * strana
```

```
vysledek = obsah_ctverce(4)
print(vysledek)
```

```
def ano_nebo_ne(otazka):
    while True:
        odpoved = input(otazka + ' (ano/ne) ')
        if odpoved == 'ano':
            return True
        elif odpoved == 'ne':
            return False
        else:
            print('Nerozumím, zkus to znovu.')
```

```
stastna = ano_nebo_ne('Jsi šťastná?')
bohata = ano_nebo_ne('Jsi bohatá?')
```

Zkus napsat funkci def napis_hlasku(nazev, skore), která např. po zavolání napis_hlasku('Tvoje', 256) vypíše:

```
Tvoje skóre je 256
Skvělé!
```

ale po zavolání `napis_hlasku('Protivníkovo', 5)` vypíše:

```
Protivníkovo skóre je 5
Aspoň něco...
```

Hlášky můžou být třeba:

- 1000 a víc: Světový rekord!
- 100 a víc: Skvělé!
- 10 a víc: Ujde to.
- jinak: Aspoň něco...

2.5.6 Čtení souborů: `with`

```
with open('basnicka.txt') as soubor:
    obsah = soubor.read()

print(obsah)
```

```
with open('basnicka.txt') as soubor:
    for radek in soubor:
        print(radek.rstrip())
```

Zkus napsat program, který přečte nějaký soubor, všechna písmenka převede na velká, a výsledek vypíše.

2.5.7 Procvičení

Napiš hru Oko bere:

- Začínáš s 0 body.
- Počítač v každém kole vypíše kolik máš bodů, a zeptá se, jestli chceš pokračovat.
 - Pokud ne, hra končí.
 - Pokud ano, počítač „otočí kartu“ (náhodně vybere číslo od 2 do 10), a přičte její hodnotu k bodům.
 - Pokud máš víc než 21 bodů, prohráváš.
- Cílem hry je získat co nejvíc bodů, ideálně 21.

2.6 [30min] Zadání projektu

2.6.1 Zadání domácího úkolu

Vymyslete téma projektu! Při vymýšlení komunikujte s koučem. Řekne vám co jde jednoduše, a na co by byla potřeba víc času.

2.6.2 Varianta 1: Webová aplikace

Naučíme se *vytvářet webové stránky*, a ptát se jejich návštěvníků na různé informace, které pak na stránkách můžeme zobrazit.

Příklady:

- Plánování akce (kdy má nejvíc lidí zároveň čas?)
- Blog nebo portfolio
- Jednoduchá webová hra (pexeso, textovka)

Nedoporučujeme web, který by nutně od začátku potřeboval registraci uživatelů (ta se případně dá dopsat po kurzu).

Taky nedoporučujeme stránky založené na animacích: aby se cokoliv změnilo, uživatel bude muset zmáčknout odkaz nebo tlačítko, a načte se nová stránka.

2.6.3 Varianta 2: Hra

Naučíme se *kreslit animované obrázky a reagovat na klávesnici a myš*, což je ideální na interaktivní hry.

Příklady:

- Had
- Pong
- Šachy, dáma, atd.
- Asteroids
- Skákačka (Mario)

Nedoporučujeme hru založenou na 3D grafice, efektech, nebo zvuku.

Bude-li tvá hra potřebovat scénář nebo mapu, do příště je vymysli. Budeš-li do hry potřebovat obrázky, do příště si nějaké sežeň. (Pokud stahuješ z Internetu, zkontroluj jestli máš právo obrázky použít ve své hře. Vhodné obrázky se dají najít na [OpenGameArt](#), příklady: 1, 2, 3.)

2.6.4 Varianta 3: Nástroj pro příkazovou řádku

Taky se naučíme jak psát programy pro příkazovou řádku – nejjednodušší způsob, jak nechat počítač za nás něco udělat nebo vypočítat.

Tenhle typ projektu by se měl zaměřit buď na automatizaci nějaké nudné činnosti, kterou na počítači často děláš, nebo na analýzu informací.

Příklady:

- Hromadné přejmenování fotek, automatické zařazení do adresářů
- Analýza textu – počet písmen/slov, seznam nejčastějších slov, srovnání počtu teček/čárek/vykičníků/otazníků, vyhledávání...
- Obdobná analýza tweetů z Twitteru
- Zpracování informací z tabulky exportované z Excelu nebo Google Docs
- Analýza obrázku – Je obrázek spíš modrý, nebo spíš červený? Který ze dvou obrázků je kontrastnější?

Tenhle druh projektu je hodně flexibilní, doporučujeme každý nápad konzultovat s koučem :)

2.7 [30min] Poděkování a rozloučení

XXX

Na začátku druhého dne doplníme znalosti Python u třídy

3.1 [1h] Třídy

XXX: Dopsat (viz kurz pro PyLadies)

3.1.1 Chování vs. stav

XXX

3.1.2 Vlastní třídy

XXX

3.1.3 Atributy

XXX

3.1.4 `self`

XXX

3.1.5 `__init__`

XXX

3.1.6 Jak to využít v praxi

XXX

3.2 [2h] Git

3.2.1 Co to je Git

XXX

3.2.2 Začínáme: `git clone`

XXX

3.2.3 Zjištění stavu: `git status`

XXX

3.2.4 Přidávání souborů: `git add`

XXX

3.2.5 Vytvoření revize: `git commit`

XXX

3.2.6 Přehled revizí: `git log` a `gitk`

XXX

3.2.7 Nahrání změn: `git remote` a `git push`

3.2.8 Stažení změn: `git pull`

XXX

3.2.9 Konflikty a jak je řešit

XXX

3.2.10 Shrnutí

XXX

3.3 [30min] Oběd

3.4 [1h] Jak na webovou aplikaci

3.4.1 Instalace frameworku `flask`

XXX

3.4.2 Jak funguje Internet

XXX

3.4.3 Dotaz a odpověď ve Flasku

XXX

3.4.4 Základy HTML

XXX

3.4.5 Základy CSS

XXX

3.4.6 Formuláře a HTTP metody: GET, POST a PUT

XXX

3.4.7 Jak si uložit stav

XXX

3.4.8 Něco o bezpečnosti

XXX

3.5 [1h] Jak na hru

3.5.1 Instalace knihovny `pygame`

XXX

3.5.2 Smyčka událostí

XXX

3.5.3 Kreslení obrázků

XXX

3.5.4 Rotace

XXX

3.5.5 Vstup a reakce na něj

XXX

3.5.6 Základy herní fyziky

XXX

3.5.7 Výkon programu

XXX

3.6 [30min] Jak na program pro příkazovou řádku

3.6.1 Instalace knihovny `click`

XXX

3.6.2 Druhy programů pro příkazovou řádku

XXX

3.6.3 Konvence příkazové řádky

XXX

3.6.4 Jak na to s Clickem

XXX

3.7 [1h 30m] Další témata – podle vybraných projektů

3.7.1 (?) Dědičnost tříd

XXX

3.7.2 (?) Stahování informací z Webu: `requests`

XXX

3.7.3 (?) Webové API: `json`

XXX

3.7.4 (?) Práce s textem: `regex`

XXX

3.7.5 (?) Práce s tabulkovými daty: `csv`

XXX

3.7.6 (?) Základy numerických výpočtů: `numpy`, `scipy`, `matplotlib`

XXX

3.7.7 (?) ...

3.8 [30min] Zadání domácího úkolu; práce na projektu

Úkol:

- Dokončit projekt
- Připravit prezentaci

Následuje individuální nebo skupinová konzultace s koučem.

4.1 Práce na projektech

Dodělejte projekt. Nebudete-li vědět jak dál, koučové vám rádi poradí!

4.2 Doplnující témata

Dle zájmu a zaměření projektů můžeme vysvětlit pokročilejší techniky, které pro dokončení jednotlivých projektů jdou použít.

4.2.1 Nasazení

Webové stránky bude potřeba pustit do světa.