

Python

Daniel Kubíček

September 2023

Contents

1	Co je to Python a jeho vlastnosti	3
2	Historie Pythonu	4
3	Instalace Pythonu na Windows	6
4	Python Enhancement Proposal	7
5	Klíčová slova Pythonu	8
6	Tabulka základních datových typů v Pythonu	10
7	Komentáře v Pythonu	11
8	Použité Zdroje	12

1 Co je to Python a jeho vlastnosti

- Python je vysokoúrovňový programovací jazyk, který byl vyvinut Guido van Rossumem a poprvé uveden na trh v roce 1991. Jedná se o interpretovaný jazyk, což znamená, že kód může být spouštěn přímo bez předchozí kompilace. Python je známý svou jednoduchou a čitelnou syntaxí, což z něj činí oblíbený jazyk pro začínající programátory i profesionály.
- Některé klíčové vlastnosti Pythonu zahrnují:
- Čitelná syntaxe: Python používá odsazení (mezer nebo tabulátorů) k určení bloků kódu, což vede k čitelnému a konzistentnímu kódu.
- Vysokoúrovňový jazyk: Python má zabudované datové typy a funkce, což usnadňuje práci s různými úkoly bez potřeby složitých manipulací s pamětí.
- Multiplatformní: Python je dostupný na různých operačních systémech, což umožňuje psát kód, který běží na různých platformách.
- Rozšiřitelnost: Python umožňuje integraci kódu napsaného v jiných jazycích, což umožňuje využít existující knihovny a rozšíření.
- Velká komunita: Python má aktivní a rozsáhlou komunitu uživatelů a vývojářů, což znamená, že existuje mnoho knihoven a nástrojů pro různé účely.
- Python se používá pro širokou škálu aplikací, včetně webových aplikací, automatizace úkolů, datové analýzy, strojového učení, umělé inteligence, vědeckého výzkumu a mnoho dalších oblastí. Je to jeden z nejpopulárnějších programovacích jazyků na světě a má silnou pozici v oblasti vývoje softwaru.



Figure 1: Logo Pythonu

2 Historie Pythonu

- Python byl vytvořen v roce 1990-1991 Guido van Rossumem v Matematickém centru Stitching v Nizozemsku jako následník jazyka nazývaného ABC. Jméno Python pochází z Monty Python's Flying Circus. Tvůrci tohoto zábavného pořadu tvrdí, že nemá žádný význam a jim akorát připadalo zábavné. Nemá oficiální logo, přestože na mnoha webech se objevuje had (přesněji krajta), což je jeden z anglických významů slova Python. Přesto oba tvůrci souvislosti s hady odmítají. Python jazyk se vyvíjí a postupem času vznikly tři nekompatibilní major verze. Verze Python 1, která byla vydána v roce 1991 a jejíž poslední aktualizace proběhla v roce 2000. Nyní se tedy již nepoužívá. Další verze Python 2, která byla vydána v roce 2000 a jejíž poslední verze byla vydána v roce 2020. Python 3, verze 3.0 byla vydána v roce 2008 a je nyní aktuální řadou.



Figure 2: Guido van Rossumem

Table 1: Tabulka vývojových milníků

Rok	Událost
1989	Guido van Rossum začal vyvíjet Python v prosinci 1989.
1991	První verze Pythonu 0.9.0. Tato verze obsahovala několik základních prvků, jako jsou blokové struktury a dynamické datové typy.
1994	Python 1.0 byl vydán. Tato verze přinesla několik vylepšení a rozšíření, včetně mapování slovníku a modulu pro manipulaci s textovými řetězci.
2000	Python 2.0, přinesl důležité změny a vylepšení, včetně podpory pro Unicode, list comprehensions a garbage collection.
2008	Python 3.0, tato verze přinesla řadu zásadních změn, včetně sjednocení typů textu a bajtů, odstranění některých zastaralých funkcí a zlepšení syntaxe.
2010	Python 2.7 byl vydán jako poslední verze série 2.x. Měl velkou bázi uživatelů a byl používán v mnoha projektech a aplikacích.
2015	Python 3.5 přidal podporu pro await/async, nové operátory a typové kontroly.
2020	Python 2.7 dosáhl konce životnosti a byl oficiálně ukončen.
2020	Python 3.9 přinesl nové nástroje a zvýšil efektivitu vývoje v Pythonu. Umožnily programátorům psát čitelnější a efektivnější kód.
2023	Aktuální datum - Python je stále populární a aktivně vyvíjen.

3 Instalace Pythonu na Windows

- Běž na oficiální stránku pythonu python.org
- klikni na v menu na Download
- Stáhni si vhodný instalační balíček pro tvoji verzi windows (32bit/64bit).
- Stažený balíček spustí. Na začátku instalace zaškrtni: **Use admin privileges when installing py.exe** a také **add python.exe to PATH**. Tyto volby ti zjednoduší vytvoření virtuálního prostředí.
- Pak zmáčkni **Install now** a dále se drž instrukcí.

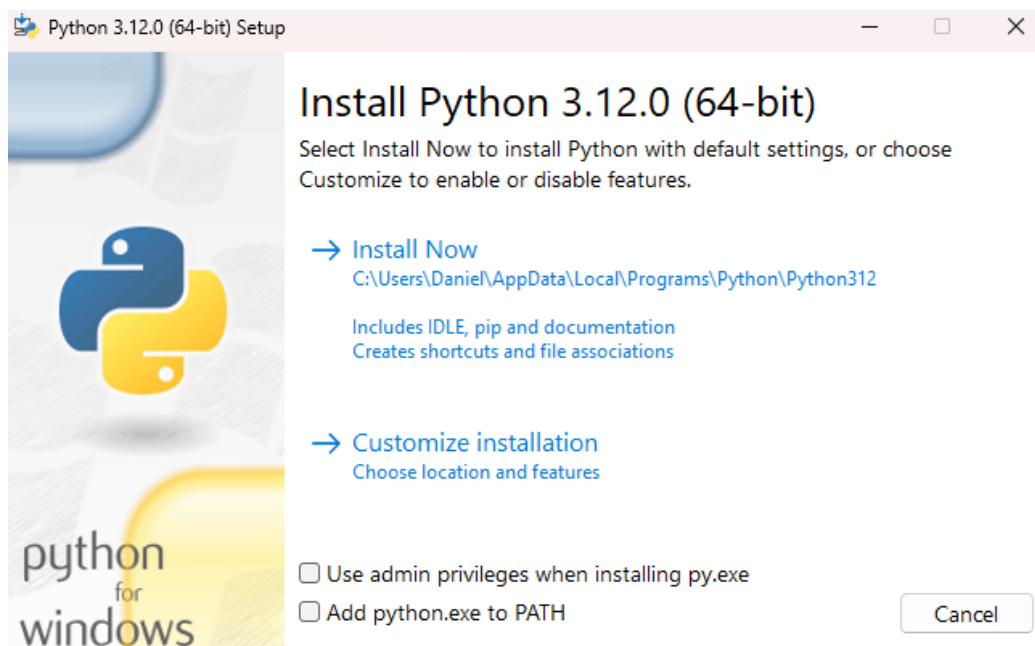


Figure 3: Instalační balíček pythonu

4 Python Enhancement Proposal

- Python Enhancement Proposal (PEP) je návrh nebo dokument, který slouží k popisu nových funkcí, rozšíření nebo změn v programovacím jazyku Python. PEPy jsou způsobem, jakým komunita Pythonu diskutuje o nápadech, které by měly být začleněny do jazyka, a jakým způsobem se tyto nápady stávají součástí oficiálního Pythonu.
- PEP 8 je jedním z Python Enhancement Proposal (PEP) dokumentů, který definuje oficiální směrnice pro styl psaní kódu v jazyce Python. Jeho název je "Style Guide for Python Code" a slouží k tomu, aby vývojáři psali konzistentní, čitelný a čistý kód, který je snadný k udržování a sdílení.
- Zde je ukázka správně formátovaného bloku python kódu **listings**:

```
1 def pozdrav(name):
2     if name:
3         print("Ahoj, " + name)
4     else:
5         print("Ahoj, -svete")
```

- Zde je ukázka špatně formátovaného bloku python kódu **listings**:

```
1 def pozdrav(name):
2     if name:
3         print("Ahoj, " + name)
4     else:
5         print("Ahoj, -svete")
```

- První ukázka obsahuje správně odsazený kód, zatímco druhá ukázka má nesprávné odsazení, což by bylo v rozporu s PEP 8.

5 Klíčová slova Pythonu

- **and** - logický operátor: současně platí, a zároveň
- **as** - příkaz využitý při řešení výjimek
- **assert** - příkaz využitý při řešení výjimek
- **break** - ukončení současné činnosti a přechod na další (například vyskočení z cyklu)
- **class** - třída
- **continue** - ukončí současný průběh cyklu a zahájí nový průchod cyklem
- **def** - vytvoření vlastní funkce
- **del** - zrušení proměnné nebo funkce
- **elif** - příkaz při vícenásobném větvení (například: if ... elif ... elif ... else...)
- **else** - ukončení větvení - provede se to, co nesplňuje podmínky předchozích větví
- **except** - provede následující příkazy, jestliže dojde k výjimce (například: try ... except ...)
- **exec** - dynamické spouštění kódu programu
- **finally** - závěrečná část bloku, kde může nastat chyba (například: try ... except ... finally ...)
- **for** - cyklus se známým počtem opakování
- **from** - import funkce z externího modulu (například: from ... import ...)
- **global** - vytvoření globální proměnné, ke které lze přistupovat ze všech míst programu
- **if** - jestliže, zahajovací podmínka pro větvení
- **import** - načtení externího modulu nebo jeho položek
- **in** - logický operátor: v, ve = vrací True, jestliže to, co je vlevo, je obsaženo v tom, co je vpravo

- **is** - logický operátor: zjištění, zda jsou dva objekty totožné, obsahují stejná data
- **lambda** - vytvoření funkce pro jednorázové použití
- **local** - vytvoření lokální proměnné, například uvnitř funkce
- **nonlocal** - vytvoření proměnné s omezeným rozsahem platnosti
- **not** - logický operátor: není, opak, negace
- **or** - logický operátor: nebo
- **pass** - příkaz *pass* nedělá nic, používá se tam, kde nějaký příkaz musí být, ale neprovádí se žádná činnost
- **raise** - příkaz využitý při řešení výjimek
- **return** - vrací výsledek funkce, podobně jako *print*, ale bez jakýchkoliv úprav
- **try** - spouští blok, ve kterém se může očekávat chyba, výjimka (například: `try ... except ... finally ...`)
- **while** - cyklus s předem neznámým počtem opakování, provádí se, dokud platí podmínka
- **with** - používá se v kontextových managers
- **yield** - používá se v generátorech pro generování hodnot
- **True** - pravda, platí, 1, ano
- **False** - nepravda, neplatí, 0, ne
- **None** - žádný, nic

6 Tabulka základních datových typů v Pythonu

Table 2: Základní/primitivní datové typy v Pythonu

Datový typ	Zastupuje	Příklady
int	Celé číslo	42, -5, 2
float	Desetinné číslo	3.14, -0.2, 1.2
bool	Booleovská hodnota	True, False
complex	Komplexní číslo	$1 + 2d$, $-3 + 2n$, $0 + 10k$
str	Řetězec znaků	'Hello', '42'

7 Komentáře v Pythonu

- **Jednořádkový komentář:** Jednořádkový komentář v Pythonu je způsob, jak vložit poznámku nebo vysvětlení přímo do kódu na jediném řádku. Komentáře jsou ignorovány interpretrem Pythonu a slouží pouze pro lidské čtení kódu. Syntaxe jednořádkového komentáře je následující:

Příklad:

```
1      # Toto je jednoradkovy komentar
```

- **Víceřádkový komentář:** V Pythonu lze použít víceřádkové komentáře pomocí tzv. "docstringů" (dokumentačních řetězců), což jsou víceřádkové řetězce, které jsou umístěny na začátku funkce, třídy, modulu nebo metody a slouží k dokumentaci daného kódu. Docstrings jsou běžně používány pro vytváření dokumentace kódu, protože jsou snadno dostupné pomocí některých nástrojů a můžete je snadno extrahovat pro generování dokumentace.

Příklad:

```
""" Toto je víceřádkový komentář,  
který slouží k dokumentaci této funkce. """
```

8 Použité Zdroje

- 1.Instalace Pythonu
- 2.Klíčová slova Pythonu
- 3.Komentáře v Pythonu
- 4.Historie Pythonu