

..

Python jako programovací jazyk

Python je interpretovaný, vysokoúrovňový a objektově orientovaný programovací jazyk, který v roce 1991 vytvořil nizozemský programátor Guido van Rossum. Je navržen s důrazem na **čitelnost kódu**, jednoduchou syntaxi a maximální produktivitu vývojářů. Díky tomu se stal jedním z nejoblíbenějších a nejvšestrannějších jazyků na světě.

Klíčové vlastnosti Pythonu

Filozofie jazyka staví na přehlednosti a odmítá zbytečnou složitost (tzv. **The Zen of Python**). Mezi hlavní charakteristiky patří:

Vlastnost	Popis
Syntaxe a čitelnost	Využívá odsazení (indentaci) pomocí mezer namísto závorek `{}` nebo klíčových slov pro bloky kódu.
Dynamické typování	Datové typy proměnných se kontrolují a přiřazují až za běhu programu (runtime).
Garbage Collection	Automatická správa paměti – programátor nemusí ručně alokovat ani uvolňovat paměť.
Multiplatformnost	Běží na systémech Windows, Linux, macOS i na vestavěných zařízeních.

Jak Python funguje pod pokličkou?

Ačkoliv je Python primárně **interpretovaný jazyk**, jeho proces spouštění má dvě fáze, což jej přibližuje kompilovaným jazykům:

1. **Kompilace do bytekódu:** Při spuštění skriptu (soubor `.py``) interpret nejprve přeloží zdrojový kód do optimalizovaného mezikódu (tzv. *\*bytekódu\**, soubory `.pyc``). 2. **Vykonávání v PVM:** Tento bytekód je následně spuštěn v prostředí **PVM** (Python Virtual Machine), který jej interpretuje na instrukce pro konkrétní hardware.

Klíčové oblasti využití Pythonu

Díky obrovskému ekosystému knihoven a balíčků (dostupných přes správce **pip**) se Python uplatňuje v celé řadě moderních oborů:

*** Umělá inteligence a Data Science:** Absolutní lídr v oblasti strojového učení (ML) a hlubokého učení (DL) díky knihovnám jako **TensorFlow**, **PyTorch**, **Scikit-learn**, **Pandas** a **NumPy**. *** Webový vývoj:** Tvorba robustních backendových aplikací a API pomocí frameworků jako **Django**, **FastAPI**

nebo **Flask**. * **Automatizace a skriptování**: Psaní systémových skriptů, automatizace rutinních úloh, web scraping (nástroje jako **BeautifulSoup** či **Scrapy**). * **DevOps a cloud**: Integrace s nástroji pro správu infrastruktury, CI/CD pipelines a cloudové služby.

Ukázka syntaxe

Díky čistému designu je kód v Pythonu velmi srozumitelný. Zde je ukázka definice třídy a asynchronního zpracování:

```
import asyncio

class Pozdravovac:
    def __init__(self, jmeno):
        self.jmeno = jmeno

    async def pozdrav(self):
        # Simulace asynchronního čekání (např. dotaz do DB)
        await asyncio.sleep(1)
        return f"Ahoj, {self.jmeno}!"

# Spuštění
async def main():
    objekt = Pozdravovac("Svět")
    zprava = await objekt.pozdrav()
    print(zprava)

asyncio.run(main())
```

Výhody a nevýhody Pythonu

* **Výhody**: * Extrémně nízká bariéra pro začátečníky, snadno se učí. * Obrovská komunita a obrovské množství hotových knihoven (PyPI). * Rychlý vývoj a vysoká produktivita (méně řádků kódu pro dosažení cíle). * **Nevýhody**: * Nižší výpočetní rychlost ve srovnání s nízkoúrovňovými jazyky (C, C++, Rust), což je dáno dynamickou povahou interpretru. * Global Interpreter Lock (**GIL**) – mechanismus v tradiční implementaci CPython, který omezuje paralelní běh vícero vláken na jednom procesoru (řeší se procesy nebo asynchronním kódem).

Tip: Chcete-li překonat rychlostní limity standardního Pythonu při náročných numerických výpočtech, využívají se knihovny jako **NumPy**, které mají kritické části napsané v optimalizovaném jazyce C.

From:

<https://www.serviceit.cz/> - IT ENCYKLOPEDIE

Permanent link:

https://www.serviceit.cz/doku.php?id=it_encyklopedie:python

Last update: **2026/08/11 14:29**

