

Úvod

Guido van Rossum je holandský počítačový programátor, známy hlavne ako autor programovacieho jazyka Python.

O pôvode Pythonu napísal Van Rossum v roku 1996 toto:

„Pred šiestimi rokmi, v decembri 1989, som hľadal zábavný programátorský projekt, ktorý by ma zamestnal cez týždeň počas Vianoc. Moja kancelária ... bola zavretá, ale mal som domáci počítač a nič iného na práci. Rozhodol som sa, že napíšem interpreter pre nový skriptovací jazyk, o ktorom som už skôr premýšľal: nasledovníka jazyka ABC, ktorý by zaujal programátorov v Unixe/C. Ako pracovný názov som zvolil Python, lebo som bol v pohodovej nálade (a tiež som veľkým fanúšikom Monty Pythonovho Lietajúceho cirkusu).“ (Úvod knihy *Programming Python*, Mark Lutz, vydalo nakladateľstvo O'Reilly)



V roku 1999 poslal Van Rossum agentúre DARPA finančný návrh, nazvaný *Computer Programming for Everybody* (Počítačové programovanie pre každého), v ktorom hlbšie definoval vytýčené ciele Pythonu:

- ľahký a intuitívny jazyk, ktorý je zároveň dostatočne mocný, aby obstál medzi hlavnými konkurentmi
- otvorený kód, takže sa môže každý zapojiť do jeho vývoja
- kód, ktorý je zrozumiteľný ako bežná angličtina
- vhodný pre bežné každodenné úlohy, umožňujúci vývoj v krátkom čase

Tieto ambície sú preukázateľne naplnené. Python sa stal populárnym programovacím jazykom, čiastočne v prostredí Internetu. Medzi komunitou Pythonu je Van Rossum označovaný ako Benevolent Dictator for Life (BDFL, čiže benevolentný doživotný diktátor) v tom zmysle, že pokračuje v dohľade nad procesom vývoja Pythonu a ak je treba, robí významné rozhodnutia.

(Voľne prebraté: https://sk.wikipedia.org/wiki/Guido_van_Rossum)

Vykonávanie programu, kompilátor, interpreter

Bolo by dobre vedieť, čo sa udeje po spustení napísaného programu. Nasledujúcu kapitolu z práce kolegyne Evy Hanulovej by bola veľká škoda neuviesť.

Na začiatku počítačovej éry programátori písali programy v strojovom kóde. Potom niekoho múdreho napadlo, že programátor sa nemusí trápiť so strojovým kódom a vymyslel a napísal program, ktorý program v inom programovacom jazyku dokázal preložiť do strojového kódu. Takémuto programu sa hovorí prekladač (translator). Najprv to bol taký menší pokrok, lebo prvý jazyk, ktorý bol poľudštením strojového jazyka bol jazyk symbolických adries (assembly language). V počiatkoch sa líšil od strojového kódu len tým, že sa v ňom dali používať mnemonické názvy príkazov a buniek pamäte, v ktorých boli zapamätané hodnoty dôležité pre spracovanie problému. Programátori už viac nemuseli počítaču zadávať program v dvojkovej sústave. Program napísaný v jazyku symbolických adries bol pre školených ľudí čitateľný. Pred spustením museli text programu zadať kompilátoru, ten ho preložil do strojového kódu. Človek si rád uľahčí svoju prácu a tak nasledovala postupnosť vyšších programovacích jazykov. Ich príkazy sú podobnejšie prirodzenej reči, preto sa v nich ľahšie formulujú riešenia problémov.

Preklad programu

Bežne sa používajú 2 prístupy pri preklade programu z vyššieho programovacieho jazyka do nižšieho (napr. strojového kódu). Tieto spôsoby si priblížime príkladom.

1. spôsob pripomína preklad knihy z angličtiny do slovenčiny

Prekladateľ sa dohodne s autorom, že preloží knihu z angličtiny do slovenčiny. Zoberie si výtlačok knihy v angličtine, preloží text do slovenčiny, preložený text odovzdá vydavateľstvu, vydavateľstvo knihu vydá, my si ju kúpime v kníhkupectve a prečítame v slovenčine. Ak rozumieme po slovensky, tak na pochopenie knihy nepotrebujeme anglický originál ani žiadnu inú pomôcku, často si ani neprečítame meno prekladateľa.

Spustíme program prekladač z vyššieho programovacieho jazyka, napr. Pascalu do strojového kódu, podhodíme mu ako vstup súbor s našim programom vo vyššom programovacom jazyku (zdrojový súbor), prekladač spracuje náš zdrojový súbor a vyprodukuje ekvivalentný program v strojovom kóde (cieľový súbor) zvyčajne v tvare súboru s príponou exe, ktorý potom môžeme veľakrát spustiť nezávisle na našom zdrojovom programe a prekladači.

Takýto druh prekladača sa nazýva **kompilátor**.

2. spôsob pripomína simultánny preklad z angličtiny do slovenčiny

Na medzinárodnej konferencii pán Black prednáša svoj príspevok v angličtine. Väčšina účastníkov nerozumie anglickému príspevku. Zavolali na pomoc tlmočníka. Pán Black povie vetu po anglicky a počká kým ju tlmočník povie po slovensky. Takto to ide až do konca prednášky. Tlmočník musí byť počas celej prednášky prítomný na konferencii. Po skončení príspevku pána Blacka si nebudú môcť účastníci prečítať príspevok po slovensky, lebo tlmočník im poskytol len slovný preklad, interpretoval príspevok pána Blacka postupne. Účastníci mali možnosť prednáške porozumieť.

Spustíme program prekladač a povieme mu, ktorý program vo vyššom programovacom jazyku má dať postupne príkaz za príkazom vykonať. Prekladač preloží do strojového kódu prvý príkaz, nechá ho počítač vykonať, potom preloží ďalší príkaz, zase ho nechá vykonať a takto to ide príkaz po príkaze až do konca programu. Počítač program pomocou prekladača vykonal, ale preklad sa neuchoval. Ak budeme chcieť, aby sa program znovu vykonal, musíme znovu pustiť prekladač a povedať mu, aký program má vykonať. Takémuto prekladaču sa hovorí

interpreter.

Preklad a výpočet programu v Java

Programy písané v Jave sa najprv prekladajú do takzvaného bajtkódu (bytecode) a potom je bajtkód interpretovaný pomocou programu, ktorému sa hovorí Java Virtual Machine alebo JVM. Pre všetky platformy je bajtkód rovnaký, JVM je naopak závislá na platforme (platformy sa líšia operačným systémom alebo strojovým kódom). To znamená, že programátor môže vyvíjať program na počítači s jednou platformou a spúšťať na počítači s inou platformou. Zdrojový text programu v jave má príponu java (napr. mojprvy.java). Preložený bude mať príponu class (mojprvy.class).

(koniec prevzatého učiva)

Inštalácia, príkazový a programovací režim

Z <https://www.python.org/downloads/> si stiahnete najnovšiu verziu Pythonu 3.5.x (obrázok vpravo) a nainštalujete.

Pri inštalácii sa riadte obrázkami vpravo, v ktorých vidieť, aké nastavenie sme použili.

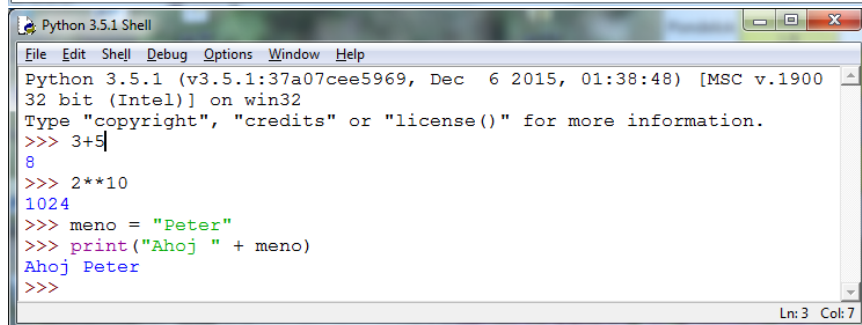
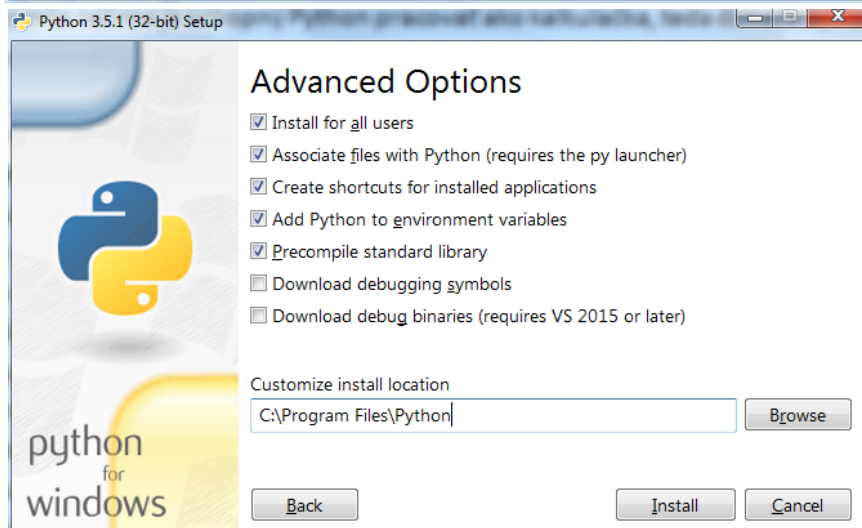
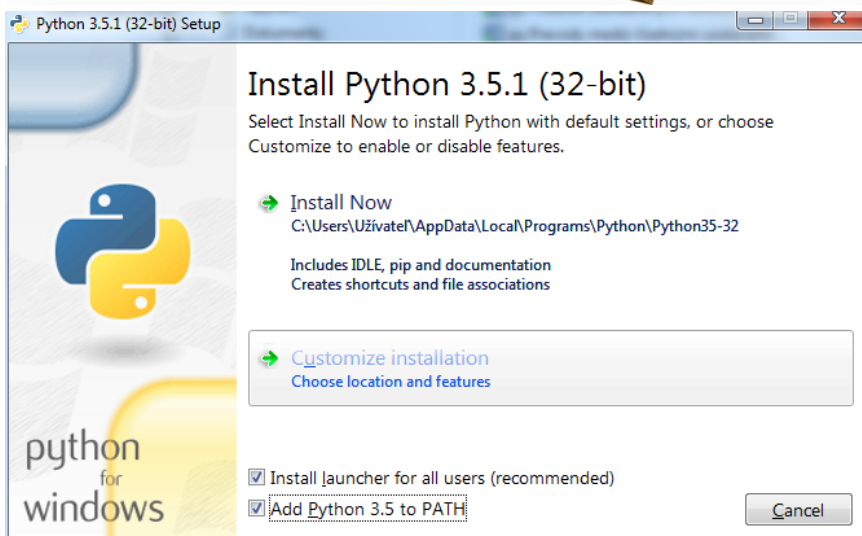
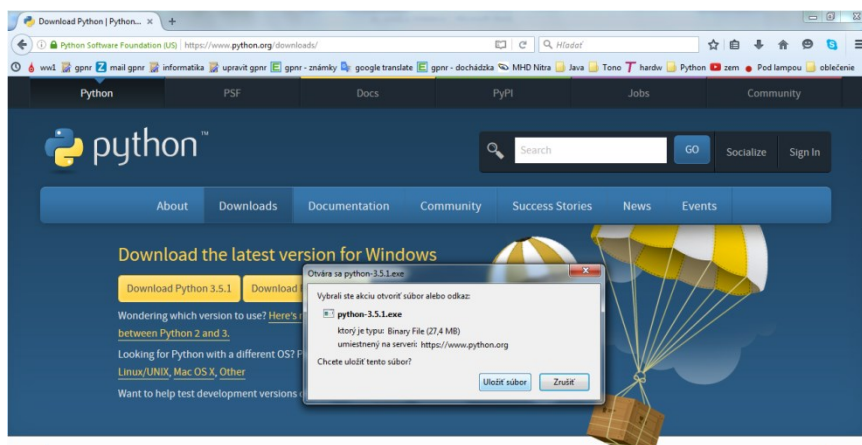
Do priečinka C:\Program Files\Python (tretí obrázok) budeme neskôr umiestňovať aj naše moduly.

Po inštalácii v Microsoft Windows sa vytvorí v ponuke **Štart - Všetky programy** skupina programov označená **Python 3.5** a v nej IDLE (Python 3.5), Python 3.5, Python 3.5 Manuals a Python 3.5 Module Docs. Nás zaujíma IDLE (odporúčame vytvoriť odkaz na plochu), čo je skratka Integrated Development Environment, teda integrovaného vývojového prostredia na písanie programov v Pythone.

Po spustení uvidíme verziu Pythonu a riadok s prompt `>>>` a kurzorom. Python je **v príkazovom režime** a očakáva napísanie príkazu a stlačenie klávesu Enter. Po odoslaní príkazu dôjde k jeho vykonaniu, keďže Python je interpreter. Ak sa príkaz nedá vykonať, vypíše sa chybové hlásenie. Takémuto interaktívnemu oknu hovoríme shell. V tomto režime je schopný Python pracovať ako kalkulačka, teda dokáže vyhodnocovať aj výrazy. Tento režim je dobrý na hranie sa s príkazmi a výrazmi v Pythone (obrázok vpravo).

Na písanie programov slúži **programovací režim**. Zrejme každý program sa ukladá do súboru, preto v shelli použijeme ponuku File - New File. Zobrazí sa ďalšie textové okno (bez promptu), do ktorého píšeme program.

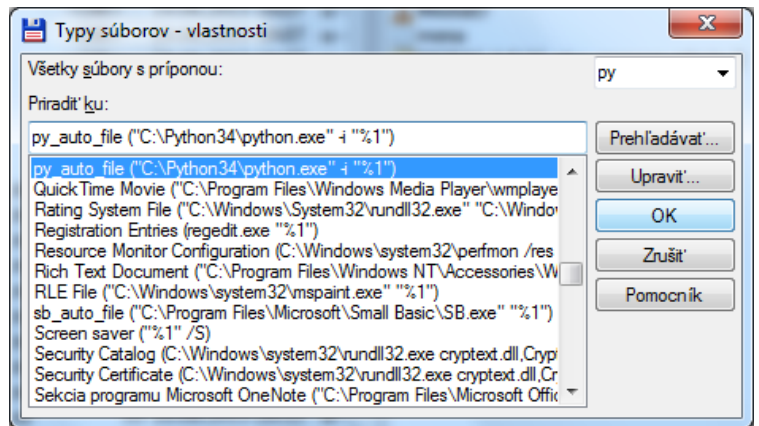
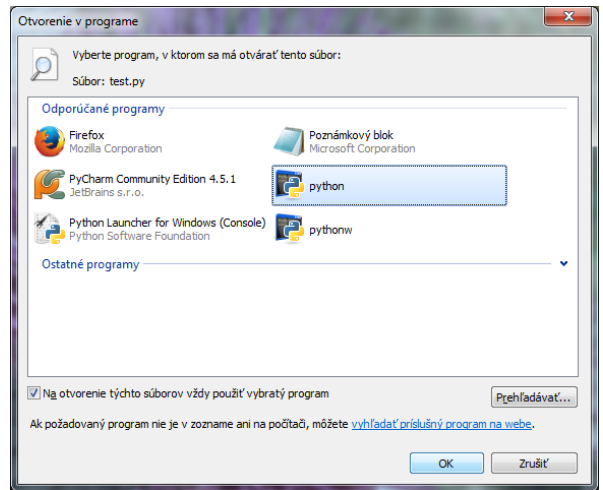
K vykonaniu príkazov dôjde po spustení programu cez ponuku Run - Run Module alebo stlačením klávesu `<F5>`. Štandardne sa požaduje najprv súhlas na uloženie súboru, po stlačení OK sa zobrazí „windowsové okno“ Uložiteľ



ako... a k zadanému názvu sa pripojí prípona py a spustí program (skript). V pôvodnom okne shell sa celý Python reštartne (vypíše ...= RESTART =...) a vykonajú sa príkazy programu. Ukončenie vykonávania programu shell oznámi zobrazením promptu (>>>). K úprave programu musíme prekliknúť do textového okna so zadaným názvom programu v hlavičke okna.

Spúšťanie súborov s príponou py priamo zo systému Microsoft Windows (informatívna časť)

- Po bezchybnej inštalácii by sa v poli Hodnota premennej Path mala nachádzať aj cesta ...;C:\Python35\ (Štart - Ovládací panel - Systém - Rozšírené systémové nastavenia - Spresnenie - Premenné prostredia... - Systémové premenné - Premenná: Path - kliknúť na Upraviť).
- Po kliknutí pravým tlačidlom myši na súbor s príponou py treba vybrať Otvoriť v programe - Vybrať predvolený program... a potvrdiť ponuku python (pozri obrázok) so zaškrtnutým prepínačom Na otvorenie týchto súborov vždy použiť vybraný program. (Program pythonw otvára súbor v programovacom režime!)
- Po dvojkliku na súbor s príponou py sa otvorí nové konzolové okno, vykonajú sa príkazy a okno sa hneď aj zatvorí (pokiaľ neobsahoval input - vstup z klávesnice). Potlačiť zavretie konzolového okna možno dvoma spôsobmi. Buď do každého programu na koniec vložíme input() alebo pridáme parameter -i do vlastností spúšťania súborov s príponou py, najjednoduchšie cez Total Commander ponuka Súbor - Priradiť k... - Všetky súbory s príponou: dopíšeme py, upravíme podľa obrázka (pred aj za -i je medzera) a klikneme na OK. Po tejto úprave budeme musieť zatvárať konzolové okno po ukončení programu cez tlačidlo Zavrieť.



Študijné zdroje

Či sú texty písané pre Python 3.x alebo nižšiu verziu rozlíšite napríklad podľa toho, že až od verzie 3.0 sa v príkaze print začali používať zátvorky. Tiež formátovanie používajúce % je zo staršej verzie, novšie formátovanie využíva funkciu str.format().

Internet (august 2015)

Python Tutorial, Release 3.2.3, Guido van Rossum, Fred L. Drake, Jr., editor

(<http://marvin.cs.uidaho.edu/Teaching/CS515/pythonTutorial.pdf>)

prednášky, Andrej Blaho, FMFI UK Bratislava (<http://python2013.input.sk/>)

prednášky, UPJŠ Košice (http://ics.upjs.sk/~rkb/web/s.ics.upjs.sk/_slivko/python/)

rýchlokurz, UPJŠ Košice, Bc. Maroš Dzuriš (http://ics.upjs.sk/~rkb/web/s.ics.upjs.sk/_dzurisMaros/sps/)

seriál Python (<http://programujte.com/clanky/54-serial-python/>)

zoznam dostupnej literatúry (<http://www.py.cz/TutorialsLiteratura>)

<http://www.thomas-cokelaer.info/tutorials/python/contents.html>

<http://www.geon.wz.cz/buwralug/index.html>

príklady (<http://www.dotnetperls.com/>)

...