**Syntax**

* Je jednoduchá, b**lok príkazov je určený odsadením. N**a odsadzovanie kód sa používa tabulátor alebo medzery, netreba zátvorky (napr. C).
* V Pythone je dôležité konzistentné odsadzovanie - oficiálne smernice odporúčajú veľkosť zhruba štyroch medzier.
* **Za výrazy určujúce tok programu sa v Pythone píše dvojbodka.**
* Písanie kódu je vhodné robiť v dobrom prostredí (programe), napr. IDLE či PyCharm

**Premenné, základné dátové typy a funkcie**

* V Pythone premenné nemusíme deklarovať s ich dátovým typom. Vďaka tomu je vývoj v jazyku rýchlejší a jazyk sa zo začiatku lepšie osvojuje.
* V Pythone sú premenné brané ako odkazy na objekty, vďaka čomu ich môžeme používať vo funkciách a volať na nich metódy. Premenná nám väčšinou uchováva nejakú hodnotu určitého (dátového) typu.

**Základné dátové typy**

Celé číslo

Anglicky integer, skratka int - uchováva v pamäti celé číslo, ktorého veľkosť je limitovaná len pamäťou počítača.

Desatinné číslo

Anglicky floating point number, skratka float - uchováva v pamäti desatinné číslo, nie je však neobmedzene presné. Pri desatinných čísla sa nepíše čiarka, ale bodka. Čísla začínajúce nulou sa dajú písať bez nuly. Napr. 0.25 sa dajú písať ako .25.

Reťazec

Anglicky string, skratka str - uchováva v pamäti reťazec (sled znakov - napr. Slovo). Definuje sa tak, že sa výraz, ktorý uchováva, zapíše do úvodzoviek. Buď jednoduchých ' ', alebo do dvojitých " ".

Dátové typy deklarovať nemusíme, ak nechceme nejaké špeciálne.

Kategórie dátových typov:

|  |  |
| --- | --- |
| Text Type: | str |
| Numeric Types: | int, float, complex |
| Sequence Types: | list, tuple, range |
| Mapping Type: | dict |
| Set Types: | set, frozenset |
| Boolean Type: | bool |
| Binary Types: | bytes, bytearray, memoryview |
| None Type: | NoneType |

**Vypísanie dátového typu:**

x = 5
print(type(x))

**Hodnoty uložené v premenných**

Vytvorenie premennej a :

>>> a = 1

Máme a, čo je odkaz na objekt s hodnotou 1. Zvýšime hodnotu premennej o 1:

>>> a = a + 1

A teraz vypíšeme hodnotu a:

>>> a

2

Premenných môžeme i viac. Pridáme si premennú b.

>>> b = 3

A teraz vydelíme b premennú a a to celé uložíme do c:

>>> c = b / a

Koľko je c ?

>>> c

1.5

Python si určí, akého typu bude premenná. Zistiť typ premennej môžeme pomocou funkcie type():

>>> type(c)

<class 'float'>

Premenná c je desatinné číslo.

**Veľkosť písmen**

Python na rozdiel od niektorých iných jazykov, **rozlišuje veľkosť písmen pri premenných,** takže a a A sú úplne iné premenné!

>>> a = 1

>>> A = 2

>>> a

1

>>> A

2

**Hello World!**

V takmer každom jazyku je prvý program vypísanie slov Hello World! My si ju tiež vytvoríme. Otvoríme si IDLE -> ďalej potom file -> New File (alebo môžete stlačiť Ctrl + N). Vyskočí vám nové okno. Tam zapíšeme nasledujúci kód, ktorý si hneď vysvetlíme:

#!/usr/bin/env python3

print("Hello world!")

Prvý riadok je komentár, ktorý zároveň slúži na spustenie interpretu (Pythone) na unixových systémoch. Komentár začína # a Python ďalej ignoruje čo je za ním (na riadku). Potom nasleduje medzera pre prehľadnosť. Na ďalšej riadku je funkcia print(). Táto funkcia zobrazí reťazec alebo číslo v konzole.

Ak program (skript) spustíte, dostanete toto:

Hello world!

Teraz skúste zapísať do vášho nového programu tento kód a ten potom uložte a spustite:

#!/usr/bin/env python3

print("Ja som prvý riadok.")

print("A ja druhý.")

výstup:

Ja som prvý riadok.

A ja druhý.





Funkcia print() ale dokáže vypísať viac textu naraz, stačí oddeliť text čiarkou:

>>> print("Ja som prvá časť textu", "a ja druhá!")

Výsledok:

Ja som prvá časť textu a ja druhá!

Reťazce sa dajú aj "sčítať":

>>> print("Ja som prvá časť textu" + " " + "a ja druhá!")

Výstup do konzoly je rovnaký ako predtým a vyzerá takto:

Ja som prvá časť textu a ja druhá!

**Papagáj**

Skúsme nejako reagovať na vstup od užívateľa. Napíšeme program papagáj, ktorý bude dvakrát opakovať to, čo používateľ napísal. Ešte sme neskúšali z konzoly nič načítavať, ale je to veľmi jednoduché. Skúsme si napísať nasledujúci kód:

print("Ahoj, som virtuálny papagáj Lora, rád opakujem!")

slovo = input("Napíš niečo: ") #získa od užívateľa vstup a uloží ho do premennej

vysledok = slovo + " " + slovo # vytvorí novú premennú

print(vysledok)

Tu je výsledok:

Ahoj, som virtuálny papagáj Lora, rád opakujem!

Napíš niečo: Nazdar!

Nazdar! Nazdar!

**Parsovanie**

Funkcia input() vracia reťazec. Ale čo keď potrebujeme dostať od užívateľa napríklad číslo?

Musíme vykonať tzv. Parsovanie. Číslo vytvoríme z reťazca (alebo z niečoho iného) pomocou vstavanej funkcie int(). Naopak môžeme previesť číslo (alebo niečo iné - napr. Desatinné číslo na reťazec pomocou funkcie str().

**Program zdvojnásobovač**

print("Zdvojnásobovač!")

cislo = int(input("Zadajte číslo: ")) # reťazec z input () sa prevedie na celé číslo

print("Jeho dvojnásobok je:", 2 \* cislo)

Tu je príklad:

Zdvojnásobovač!

Zadajte číslo:

3

Jeho dvojnásobok je: 6

**Kalkulačka**

Teraz si vytvoríme konzolovú aplikáciu, ktorá dokáže sčítať, odčítať, násobiť a deliť dve čísla.

print("Vitajte v kalkulačke\n")

prve\_cislo = float(input("Zadajte prvé číslo: "))

druhe\_cislo = float(input("Zadajte druhé číslo: "))

print("Súčet:", prve\_cislo + druhe\_cislo)

print("Rozdiel:", prve\_cislo - druhe\_cislo)

print("Súčin:", prve\_cislo \* druhe\_cislo)

print("Podiel:", prve\_cislo / druhe\_cislo)

input("\nĎakujem za použitie kalkulačky, aplikáciu ukončíte ľubovoľnou klávesou.")

Znak \n v texte spôsobí, že sa text presunie na novy riadok. To znamená, že text za týmto znakom sa zobrazí na ďalšom riadku (v tomto prípade sa urobí prázdny riadok, keďže za týmto znakom nič nenasleduje).

*Ak budete mať v Pythone viacslovné názvy premenných, odporúča sa oddeľovať názvy podčiarknikom.*

Výsledok spustenia kalkulačky:

Vitajte v kalkulačke

Zadajte prvé číslo:

3

Zadajte druhé číslo:

2

Súčet: 5

Rozdiel: 1

Súčin: 6

Podiel: 1.5

Ďakujem za použitie kalkulačky, aplikáciu ukončíte ľubovoľnou klávesou.