Python úvod, čísla, reťazce, priradenie, PEP 8

1 Python program a algoritmus

Algoritmus je konečný postup riešenia úlohy. Program je zápis algoritmu v programovacom jazyku. Zdrojový kód je text programu.
Geometrický algoritmus môže opisovať konštrukciu stredu úsečky. Program môže vypočítať súradnice stredu a vykresliť výsledok.
Python je univerzálny, vysokoúrovňový, multiplatformový a viacparadigmový jazyk. Podporuje procedurálne, objektovo orientované aj funkcionálne prvky.
Python je dynamicky a silno typovaný. Typ patrí objektu; meno premennej nemá vopred deklarovaný typ. Nezlučiteľné typy sa automaticky nekombinujú.
CPython je najpoužívanejšia implementácia Pythonu. IDLE a Visual Studio Code sú vývojové nástroje, nie implementácie jazyka.

2 Interpreter kompilácia a spúšťanie

Kompilátor prekladá zdrojový kód do inej podoby. Interpreter vykonáva inštrukcie programu.
CPython preloží zdrojový kód na bajtkód a vykoná ho vo virtuálnom stroji Pythonu. Bajtkód nie je súbor objektov vytvorených programom.
Interaktívny režim vykonáva príkaz po stlačení Enter a zobrazuje hodnotu samostatného výrazu. Výzva interpretera má tvar `>>>`.
Program v súbore `.py` sa uloží a spustí ako celok. V IDLE sa používa Run Module alebo F5. V skripte sa výsledky zobrazujú funkciou `print()`.

```
2 + 3
print("Ahoj, Python!")
type(3)
```

3 Objekty typy a mená

Hodnota 3 je objekt typu `int`. Literál 3 je zápis, ktorý pri vykonaní programu poskytne príslušný objekt.
Premenná je meno odkazujúce na objekt. Priradenie najprv vyhodnotí pravú stranu a potom spojí meno naľavo s výsledným objektom.
Funkcia `type()` vráti typ objektu. Funkcia `id()` vráti identitu objektu v rámci daného behu programu.
Identifikátor môže obsahovať písmená, číslice a podčiarkovník; nesmie sa začínať číslicou ani byť kľúčovým slovom.
Python rozlišuje veľké a malé písmená.
Identifikátory môžu obsahovať znaky Unicode. V bežnom zdrojovom kóde sa používajú opisné názvy bez diakritiky v štýle `snake_case`.

```
pocet = 10
Pocet = 20
meno_studenta = "Eva"
print(pocet, Pocet, meno_studenta)
print(type(pocet), id(pocet))
```

4 Čísla a operátory

`int` reprezentuje celé čísla s veľkosťou obmedzenou dostupnou pamäťou. Operácie `+`, `-`, `*`, `//`, `%`, `**` s hodnotami `int` zachovávajú presný celočíselný výsledok.
`float` má konečný rozsah a presnosť. Bežný CPython používa binárne čísla s pohyblivou rádovou čiarkou podľa IEEE 754 binary64.
`complex` zapisuje komplexné čísla s imaginárnou jednotkou j, napríklad `2 + 3j`. `bool` má hodnoty `True` a `False` a je podtypom `int`.
Zápisy `1e5` a `1e-2` znamenajú `100000.0` a `0.01`. Desatinným oddeľovačom v zdrojovom kóde je bodka.
Operátory `+`, `-`, `*`, `/`, `//`, `%`, `**` znamenajú sčítanie, odčítanie, násobenie, delenie, delenie nadol, zvyšok a mocninu.
Operátor `/` vracia `float`. Operátor `//` zaokrúhľuje podiel nadol; `-7 // 2` je -4. Zátvorky menia poradie vyhodnocovania.

```
print(10 / 3, 10 // 3, 10 % 3)
print(-7 // 2, 2 ** 10)
print(2 + 3 * 4, (2 + 3) * 4)
print(1e5, 1e-2, 2 + 3j)
print(0.1 + 0.2)
```

Hodnota `0.1` nemá v binárnom type `float` presnú reprezentáciu. Výsledky výpočtov s `float` môžu obsahovať malú zaokrúhľovaciu chybu.

5 Konverzie a zaokrúhľovanie

Aritmetická operácia s `int` a `float` zvyčajne vytvorí `float`. Operácia s reálnym a komplexným číslom vytvorí `complex`.
Funkcie `int()`, `float()`, `complex()`, `str()` a `bool()` vytvárajú hodnoty požadovaného typu, ak je konverzia definovaná.
`int(3.14)` odstráni desatinnú časť smerom k nule. Nevykonáva zaokrúhlenie. Neplatný číselný reťazec vyvolá `ValueError`.
`round(x)` zaokrúhli na celé číslo. `round(x, n)` zaokrúhli na n desatinných miest; záporné n zaokrúhľuje na desiatky, stovky alebo vyššie rády.
Hodnoty presne uprostred sa zaokrúhľujú k najbližšiemu párnemu výsledku: `round(2.5)` je 2 a `round(3.5)` je 4.

```
print(int("37"), int(3.14), int(-3.14))
print(float("3.14"), str(37))
print(round(3.14159, 2))
print(round(-3.6), round(1234, -2))
```

6 Reťazce výstup a vstup

`str` uchováva text ako Unicode. Zdrojové súbory Pythonu používajú predvolene kódovanie UTF-8. Reťazce sú nemenné objekty.

Reťazec sa zapisuje medzi apostrofy alebo úvodzovky. Zápisy `\n`, `\t`, `\\`, `\'` a `\"` predstavujú nový riadok, tabulátor, spätnú lomku, apostrof a úvodzovku.

Operátor `+` spája reťazce, `*` ich opakuje a `in` testuje výskyt. Funkcia `len()` vráti počet znakov. Indexovanie sa začína nulou.

```
text = "Žilina"
print(text[0], len(text))
print("Povedal: 'Ahoj!'")
print("prvý riadok\ndruhý riadok")
print("Ahoj " + "svet", "ha" * 3)
print("lin" in text)
```

`print()` môže vypísať viac hodnôt oddelených čiarkami. Parametre `sep` a `end` menia oddeľovač a ukončenie výstupu.

`input()` vypíše výzvu, načíta riadok a vždy vráti `str`. Číselný vstup treba previesť pomocou `int()` alebo `float()`.

`f-reťazec` vloží hodnoty výrazov do textu. Zápis `:.2f` zobrazí číslo typu `float` na dve desatinné miesta.

```
suma_eur = float(input("Zadaj sumu v eurách: "))
kurz = 26.0  # ilustračná hodnota
suma_czk = suma_eur * kurz
print(f"{suma_eur:.2f} EUR je {suma_czk:.2f} CZK")
```

7 Funkcie ako objekty

`abs` je objekt funkcie. `abs(10)` je volanie funkcie a jeho výsledkom je objekt typu `int`.

Funkciu možno priradiť inému menu a zavolať cez nové meno. `help()` zobrazí dokumentáciu objektu.

```
print(type(abs(10)))
print(type(abs))
help(abs)
absolutna_hodnota = abs
print(absolutna_hodnota(-10))
vypis = print
vypis("Funkcia je objekt.")
```

8 Priradenie

Operátor `=` priraďuje meno k objektu. Operátory `+=`, `-=`, `*=`, `/=`, `//=`, `%=` a `**=` kombinujú operáciu s priradením.

Zápis `a = b = c = 0` priradí tú istú hodnotu viacerým menám. Zápis `x, y = 10, 20` rozbalí hodnoty do viacerých mien.

Pravá strana sa vyhodnotí pred priradením na ľavej strane. Preto zápis `a, b = b, a` vymení hodnoty bez pomocnej premennej.

Viacnásobné priradenie môže spojiť mená s objektmi rôznych typov. Počet mien a počet hodnôt pri rozbaľovaní sa musia zhodovať.

```
pocet = 10
pocet += 1
pocet *= 2
a = b = c = 0
znak, cislo, hodnota = "A", 10, 3.14
a, b = 1, 2
a, b = b, a
print(pocet, a, b, znak, cislo, hodnota)
```

Reťazec a číslo nemožno priamo sčítať. Číslo možno previesť cez `str()` alebo vložiť do f-reťazca.

```
znak, cislo, hodnota = "A", 10, 3.14
text1 = znak + " " + str(cislo) + " " + str(hodnota)
text2 = f"{znak} {cislo} {hodnota}"
print(text1, text2, sep=" | ")
```

9 Moduly balíky a knižnice

Modul je spravidla jeden súbor `.py`. Príkaz `import` sprístupní mená definované v module.

Balík organizuje súvisiace moduly a podbalíky. Bežný balík obsahuje súbor `__init__.py`; tento súbor môže byť prázdny.

Knižnica je všeobecné pomenovanie opakovane použiteľného kódu. Môže obsahovať jeden modul, viac modulov aj viac balíkov; nejde o presnú ďalšiu úroveň štruktúry.

```
import math
print(math.sqrt(2), math.pi)
from math import sin
print(sin(0))
```

10 PEP 8

PEP 8 je odporúčanie pre jednotný a čitateľný štýl zdrojového kódu. Nie je súčasťou syntaxe Pythonu.

Blok kódu sa odsádza štyrmi medzerami. Tabulátory a medzery sa v odsadení nemiešajú.

Medzery sa píšu okolo binárnych operátorov a za čiarkami. Za názvom funkcie sa pred otváracou zátvorkou medzera nepíše.

Premenné a funkcie používajú `snake_case`; triedy používajú `CapWords`. Konštanty sa zvyčajne zapisujú `VEĽKÝMI_PÍSMENAMI`.

Odporúčaná maximálna dĺžka riadka kódu je 79 znakov. Komentár začína znakom # a vysvetľuje význam alebo dôvod.