

Dohoda

Všetky názvy, ktoré si volí autor programu, sú identifikátory. **Identifikátor** je postupnosť písmen anglickej abecedy alebo číslíc 0 až 9 začínajúca písmenom. Medzi písmená je dodefinovaný aj podčiaričnik. Rozlišujú sa malé a veľké písmená! Niektoré programovacie jazyky povoľujú používať na mieste písmen nie len písmená anglickej abecedy.

Premenné označujeme len malými písmenami; používame názvy, ktoré napovedajú o obsahu premennej, slová v názvoch oddeľujeme podčiaričnikmi, napr. `sucet`, `pocet_prvkov`, `priemerna_teplota` a pod.

Konštanty označujeme len veľkými písmenami; používame názvy, ktoré napovedajú o obsahu konštanty, slová v názvoch oddeľujeme podčiaričnikmi, napr. `MAX_CISLO`.

Názov **funkcie** začína malým písmenom, každé ďalšie slovo v názve začína veľkým písmenom, podčiaričnik sa nepoužíva; názvy funkcií by mali začínať slovesom, napr. `vypocitajAritmetickyPriemer`, `vypisZoznamMien`, `vratPriemernuTeplotuDna`, `vratPocetNenulovychHodnot` a pod.

Viacriadkovú poznámku uzatvárame do trojitých apostrofov `'''` (pravé Alt+P) alebo do trojitých úvodzoviek (`"""`). Poznámku v riadku začíname znakom `#` (pravé Alt+X).

Údajové typy

Program môže spracúvať reťazce znakov (`str`), celé (`int`) alebo reálne (`float`) čísla a iné údajové typy (`tuple`, `list`,...). Pre každý údajový typ sú definované dovolené operácie, napríklad vykonanie príkazu `print("Jano"/2)` zrejme skončí chybou `TypeError: unsupported operand type(s) for /: 'str' and 'int'`; ďalšie príkazy síce nevypíšu chybu, ale pretože pracujú s rôznymi údajovými typmi vypíšu rôzne výsledky, napríklad `print(3*2)` vypíše 6 ale `print(3*"2")` vypíše 222. Preto je dobré sa v údajových typoch trochu orientovať, aj keď Python má tú výhodu, že typ premennej si určuje sám podľa jej aktuálnej hodnoty. Typ výsledku operácie je určený použitou operáciou, napríklad `print(6/2)` vypíše 3.0 (reálne číslo) ale `print(6//2)` vypíše 3 (celé číslo). Typ premennej možno získať funkciou `type(premenná)`.

str

string, reťazec znakov

venujeme mu samostatnú kapitolu Reťazce

int

integer, celé číslo; jeho veľkosť je obmedzená len veľkosťou pamäte, do ktorej sa ukladá

Najpoužívanejšie operácie: `+` (sčítanie), `-` (odčítanie), `*` (násobenie), `//` (celočíselné delenie, `div`), `%` (zvyšok po celočíselnom delení, `mod`), `**` (umocnenie); funkcie napr. `abs(celé číslo)` - absolútna hodnota, `int(reálne číslo)` - vráti celé číslo, odreže desatinnú časť; `round(reálne číslo)` - zaokrúhli na celé číslo; `pow(základ, exponent)` - ako umocnenie; `divmod(a,b)` vráti dvojicu čísel, prvé rovné `a // b`, druhé rovné `a % b`.

Funkcie na prevod medzi číselnými sústavami: `bin(i)` - vráti binárnu hodnotu celého čísla `i`; `hex(i)` - vráti hexadecimálnu (šestnástkovú) hodnotu celého čísla `i`; `oct(i)` - vráti oktánovú (osmičkovú) hodnotu celého čísla `i`; `int(reťazec, základ)` - prevedie reťazec reprezentujúci celé číslo v číselnej sústave so základom `základ` na celé číslo alebo vyvolá výnimku `ValueError`, $2 \leq \text{základ} \leq 36$.

float

označenie pre reálne číslo; názov vznikol pravdepodobne z faktu, že počítač každé reálne číslo uloží v tvare mantisa E exponent, kde mantisa je reálne číslo z intervalu $<0,10$ a E alebo e symbolizuje „10 na“. Napríklad číslo 1234.56789 bude uložené ako `+1.23456789E+03` (desatinná bodka „prepláva“ - float za prvú platnú číslicu); číslo `-0.000000987654321` bude uložené ako `-9.87654321e-08`. Znak „e“ môžete použiť už pri zadávaní hodnoty.

Najpoužívanejšie operácie: `+` (sčítanie), `-` (odčítanie), `*` (násobenie), `/` (delenie), `**` (umocnenie); funkcie napr. `fabs(reálne číslo)` - absolútna hodnota, `float(str alebo int)` - konvertuje na reálne číslo; `round(reálne číslo, presnosť)` - zaokrúhli na reálne číslo so zadanou presnosťou; `pow(základ, exponent)` - ako umocnenie;

Niekoľko desiatok matematických funkcií je k dispozícii v knižnici math (nutný import math) po zápise napr. math.pi vráti hodnotu π , math.sqrt(x) vráti druhú odmocninu z nezáporného x.

bool

boolean, logický typ, obsahuje len dve hodnoty False a True.

Vyhodnotením výrazu typu bool je jedna z hodnôt False alebo True.

Dovolené operácie: not (negácia), and (a zároveň) a or (alebo).

Vo výrazoch typu boolean sa používajú predovšetkým porovnávacie operátory, a to < (menší), <= (menší alebo rovný), == (rovný), != (rôzny), >= (väčší alebo rovný), a > (väčší). Vo výraze sa vyhodnocuje každý pár elementárnych výrazov, to znamená napríklad $1 < x < 2$ sa vyhodnotí ako $1 < x$ a zároveň $x < 2$.

Logickú hodnotu vracia aj použitie operátorov in (obsahuje), not in (neobsahuje), is (je) a is not (nie je). Napríklad "Peter" in "Ján Peter Dušan" vracia True; x = "Peter", výraz x is not "Peter" vráti False.

Číslo nula sa vyhodnocuje ako False, každé iné číslo ako True. Preto napríklad program

```
i = 5
while i:
    print(i, end=" ")
    i -= 1
vypíše hodnoty 5,4,3,2,1,
príkaz
for i in -3,-2,-1,0,1,2,3:
    print(bool(i))
```

vypíše

True

True

True

False

True

True

True

Podobne konštanta None (neexistencia hodnota) sa vyhodnotí ako False.

POZOR! Vyhodnotenie čísla ako hodnoty False alebo True môže byť zradné. Pozri kapitolu Lineárne vyhľadávanie problém nájdenia indexu prvého výskytu hľadanej hodnoty v zozname.

Priorita operátorov

Operátor	Popis
(), [], {}, ""	entica, zoznam, slovník, reťazec
x.attr, x[], x[i:j], f()	atribút, index, rezanie, volanie funkcie
-x	zmena znamienka
**	umocnenie
*, /, //, %	násobenie, delenie, div, mod
+, -	sčítanie, odčítanie
<, <=, >=, >, ==, !=, in, not in, is, is not	operátory porovnávania, príslušnosti (in) a identity (is)
not	negácia
and	a zároveň
or	alebo
lambda	lambda výraz

Výrazy sa vyhodnocujú zľava doprava, od vnútorných zátvoriek k vonkajším.