

4. Programování - proměnné, datové typy, operace

používání proměnných

Proměnné

- slouží k uchovávání hodnot, výsledků a veškerých dat, které se mohou v průběhu měnit
- názvy mohou obsahovat:
 - malá a velká písmena (bez diakritiky)
 - číslice (nesmí začínat číslicí)
 - podtržítko
- měly by se pojmenovávat podle toho, co obsahují!
- Python rozlišuje malá a velká písmena! (ab, AB, aB, Ab jsou 4 různé proměnné)
- nelze použít klíčová (vyhrazená) slova, v takovém případě je v Pythonu konvence za název připojit podtržítko _
- nemůže nastat přetečení proměnných (nejsme omezeni velikostí)

operátor přiřazení

- umožňuje vkládat do proměnné hodnotu
- proměnná je vždy vlevo a výraz vpravo, výraz se vyhodnotí první
- původní hodnota proměnné se ztratí!

Vícenásobné přiřazení

`a=b=c=1`

Zkrácená přiřazení +=, -=, *=, /=, //=

`a+=1` stejné jako `a=a+1`

datové typy

Python používá automatickou deklaraci typů (s výjimkou vstupu). Proměnné jednoho typu lze přepsat jiným typem.

Čísla

- celá (př. 15)
- desetinná (př. 3.141) - pouze desetinná tečka
- komplexní (př. 5+4j) - bez mezer

+ vědecký zápis 2.3E210 / 1e-57

Celá čísla lze zadávat i pomocí prefixu v různých číselných soustavách

prefix základ soustavy

0b 2

0o 8

0x 16

Boolean

pravdivostní hodnota True/False - píše se s velkým písmenem na začátku

Další datové typy: řetězce, seznamy, n-tice, slovníky, množiny a další speciální objekty

převodní funkce

Umožňují převést hodnotu z jednoho datového typu na jiný.

int() - převede hodnotu na celé číslo, desetinná část je odstraněna (ne zaokrouhlena!)

- 10.478 -> 10

- "12" -> 12
- "3.141" -> error!

float() - převede hodnotu na desetinné (reálné) číslo

- 10 -> 10.0
- "10.154" -> 10.154

complex() - převede číslo na komplexní

bool() - převede cokoli na True/False - vše kromě 0, None a prázdných kolekcí je True

str() - převede cokoli na řetězec

- 400 -> "400"
- 10.478 -> "10.478"
- (10+5j) -> "(10+5j)"

ord(), chr() - na ordinální hodnotu a zpět (1 znak)

list(), tuple(), dict(), set()

vstup a výstup

Vstup dat

Pro vstup dat ze systémového zařízení užíváme příkazu **input**, který vždy přečte řetězec znaků. V konzoli je ukončen klávesou ENTER.

```
s = input("zadej hodnotu: ")
```

Pro vstup numerických dat (int, float...) musíme použít převodní funkce pro přetypování:

```
y = float(input("Zadej desetinné číslo: "))
```

Výstup na obrazovku

Používá se příkaz **print**. Pokud chceme na jeden řádek zobrazit více proměnných, pak je oddělujeme čárkami

- print("text") - vypíše do výstupu zadaný řetězec (text) a přejde na nový řádek
- print() - prázdný řádek
- print("text", end="") - neproběhne zalomení po printu
- print("text1", 10, "text2") - tisk více hodnot, defaultně oddělené mezerou
- print("text1", 10, "text2", sep=", ") - tisk více hodnot oddělené čárkou a mezerou

aritmetické operace a operátory, priorita

- + sčítání
- odčítání
- * násobení
- / dělení
- // celočíselné dělení, zaokrouhluje vždy DOLŮ! → 1//2 = 0; -1//2 = -1
- % zbytek po celočíselném dělení
- ** umocňování

Priorita operátorů je stejná jako v běžně používaných zápisech v matematice:

- 1) Kulaté závorky
- 2) Umocňování
- 3) Násobení a dělení (i celočíselné // a zbytek %)
- 4) Sčítání a odčítání

příkaz import, použití knihoven

Každou knihovnu, kterou chceme použít, je nutné importovat:

- 1) import knihovna → knihovna.funkce()
- 2) import knihovna as m → m.funkce()
- 3) from knihovna import * → funkce()

POZOR! - může dojít k přepsání importovaných proměnných nebo fcí, pokud se v importovaných knihovnách jmenují stejně

použití a funkce knihovny math

V jazyku Python je možné využít matematickou knihovnu math, která je určena pro výpočet matematických funkcí a obsahuje standardní funkce a konstanty:

- pi, e – konstanty – Ludolfovo a Eulerovo číslo
- sin, cos, tan – goniometrické funkce (argument musí být v radiánech)
- asin, acos, atan – invertní funkce k předchozím
- radians – převádí stupně na radiány
- sqrt – druhá odmocnina
- pow(a, b) – mocnina čísla a na b
- log(a, b) – logaritmus čísla a o základu b (default Eulerovo číslo)
- trunc – odseknutí desetinné části reálného čísla
- ceil – zaokrouhlení nahoru
- floor – zaokrouhlení dolů

Funkce, které jsou v základu Pythonu:

- round - zaokrouhlení
- abs - absolutní hodnota
- Druhou odmocninu můžeme získat jako umocnění na 1/2. Podobným způsobem můžeme získat i odmocniny vyšších řádů

Kromě knihovny math lze použít knihovnu pro práci s komplexními čísly **cmath**.

komentáře

Komentáře slouží k popisu kódu a k jeho vysvětlení.

jednořádkový komentář

Víceřádkové komentáře jako takové v Pythonu nejsou, ale můžeme použít víceřádkové řetězce, které nepřirazujeme do žádné proměnné:

"""

víceřádkový

komentář

"""