

7. Programování - cyklus while

Cyklus - obecná charakteristika

- používají se k opakování bloku příkazů (v Pythonu oddělený tabulátorem / 2/4 mezerami) několikrát po sobě
- buď na základě pevně daného počtu opakování nebo podmínky → 2 hlavní typy: **for** a **while**

cyklus while - syntaxe a popis

- umožňuje opakované provádění příkazů, dokud platí podmínka
- podmínka má stejnou podobu jako u IFu (logický výraz)
- může nastat nekonečný cyklus!

while podmínka:

Příkazy

Příkazy v cyklu se opakují tak dlouho, dokud je podmínka pravdivá; když platit přestane, cyklus skončí. Příklad:

```
cislo = 1
```

```
while cislo < 1000:
```

```
    print (cislo)
```

```
    cislo *= 2
```

```
print(„Tak to je konec“)
```

Jakmile počítač uvidí `while cislo < 1000:`, udělá toto:

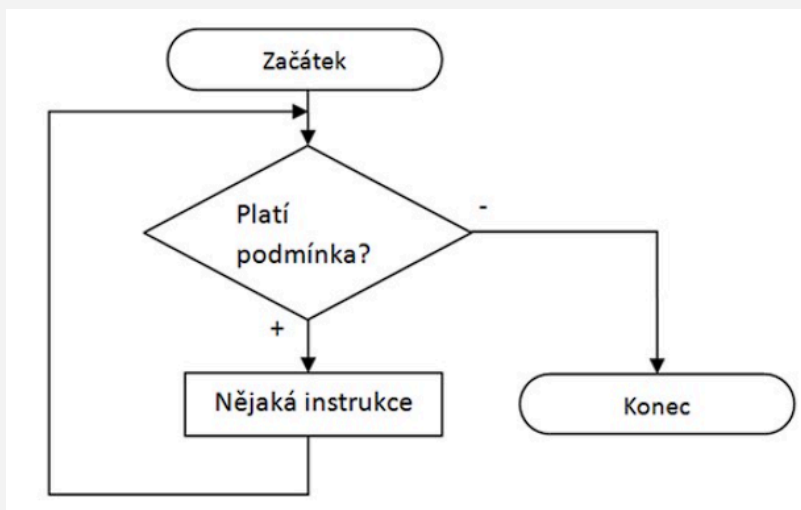
- Podívá se jestli (`cislo < 1000`) je pravda nebo ne.
- Pokud ne, opustí cyklus: bude pokračovat čímkoliv, co následuje za koncem cyklu.
- Pokud je, půjde zpracovat obsah cyklu...
- ...a vrátí se zpět na začátek zjistit, jestli podmínka stále platí.

Jinými slovy, vykonává obsah cyklu pořád dokola, ale jen dokud (`while`) je podmínka (`number < 1000`) pravdivá. Po skončení se přesune na příkazy za cyklem.

využití

Používáme, pokud nelze říci nebo nevíme, kolikrát se má cyklus opakovat (jsme řízeni podmínkou/vstupem od uživatele).

vývojový diagram



nekonečný cyklus - popis a vlastní příklad

Nekonečný cyklus nastává, pokud podmínka nikdy nedosáhne nepravdivé hodnoty a v cyklu nikdy nedojdeme k příkazu `break`. Klasický případ:

```
i = 0
```

```
while i < 10:
```

```
    print(i)
```

```
    # i += 1 (zapomeneme incrementnout i)
```

Nekonečný cyklus můžeme přerušit stisknutím kombinace Ctrl+C nebo zavřením terminálu.

Někdy je vhodné pracovat přímo s nekonečným cyklem (`while True: ...`) jako hlavním cyklem hry/programu, který skončí jeho zavřením.

příkazy `break` a `continue`

`break`

- okamžitě ukončí vykonávání cyklu
- příkazy uvedené v tělu za `break` se už neprovádí
- program pokračuje za cyklem

```
while True:
```

```
    heslo=input("Zadej vánoční kód:")
```

```
    if heslo=="jedle":
```

```
        break
```

```
    print("Špatný kód, zadej znovu.")
```

```
print("Správný kód! Veselé Vánoce!")
```

`continue`

- způsobí ukončení aktuální iterace (opakování)
- zbytek bloku cyklu za `continue` se neprovede
- program pokračuje následující iterací (`while` kontroluje podmínku, `for` přechází na další prvek)

Cykly `for` i `while` mají nepovinnou sekci **`else`**. Kód v sekci `else` se provádí po korektním ukončení cyklu. Pokud byl cyklus ukončen pomocí `break`, nebo `return`, nebo pokud nastala výjimka, kód v `else` se neprovede.

náhodná čísla

V programování se používají pro náhodné rozhodování nebo náhodný výběr prvků ze seznamu. Chceme-li pracovat s náhodným výběrem nebo čísly, potřebujeme knihovnu **`random`** (musíme nejdříve importovat).

- **`randint(od, do)`** - vygeneruje náhodné číslo z intervalu `<od, do>` (`od` a `do` jsou celá čísla)
- **`randrange(0, 100, 2)`** - generuje náhodné celé číslo od 0 do 99, sudé
- **`choice(data)`** - vybere z `dat` (obvykle řetězec nebo seznam) náhodný prvek
- **`sample(data, k)`** - vybere z `dat` (obvykle řetězec nebo seznam) `k` náhodných prvků
- **`random()`** - generuje náhodné reálné číslo od 0 do 1
- **`uniform(x, y)`** - generuje náhodné reálné číslo `x <= c < y`