# Cykly

Cykly jsou dalším základním stavebním kamenem každého programu. Pokud se má něco např. stokrát opakovat, je jasné, že nebudeme psát stokrát stejný kód.

V Pythonu existují dva základní typy cyklů:

1. FOR cyklus Cyklus s pevným počtem opakování
2. WHILE cyklus Cyklus s neznámým počtem opakování a podmínkou na začátku (cyklus se opakuje, dokud podmínka platí)

# FOR cyklus

Cyklus for použijeme v případě, kdy předem známe počet průchodů cyklem. Cyklus obsahuje tzv. řídící proměnnou (celočíselnou), ve které se postupně během každého průchodu cyklem mění její hodnota.

Cyklus for se zapisuje následovně:

**for *proměnná* in range(rozpětí):**

**příkazy**

**proměnná** je řídící proměnná cyklu, která začíná automaticky od nuly, ale máme možnost počáteční hodnotu nastavit rozšířením range(**počáteční hodnota**, rozpětí)

**range** nastavuje rozpětí. Jakmile řídící proměnná dosáhne hodnoty range, cyklus se ukončí.

např.:

range(3) => hodnota počítadla je 0, 1, 2 (cyklus se vykoná 3x)

range(1, 5) => hodnota počítadla je 1, 2, 3, 4 (cyklus se vykoná 4x)

range (5, 10) => hodnota počítadla je 5, 6, 7, 8, 9 (cyklus se vykoná 5x)

**příkazy** nám říkají, co se má v každém kroku cyklu stát.

Jednoduchý příklad na cyklus for, ve kterém uvidíme, že se nám proměnná inkrementuje (zvyšuje):

for i in range(10):

print("Hodnota řídící proměnné je",i)

Příklad na dekrementaci řídící proměnné:

for i in range(3,0,-1):

print(i)

print("bum!")

Program: Vybuchne to nebo ne??

Vnořené větvení + cyklus s dekrementací řídící proměnné.

a = str(input("Přestřihni modrý (m) nebo červený drát (c)..."))

if (a=="m"): #špatný drat

for i in range(3,0,-1):

print("odpocet",i)

print("bum!")

elif (a=="c"):

print("Přestřihnul jsi ten spravný, hurá")

else:

print("Jsi nervózni? Je to jenom bomba:-) Takový drát tu není");

Program na výpis malé násobilky pomocí opakovaných cyklů for:

Poznámka: všimněte si nového parametru u příkazu print. Příkazem **end=" "** zajistíme, že nedojde k automatickému zařádkování, což se jinak v Pythonu vždy při printu děje. Zařádkovat chceme, až po vypsání všech násobků daného čísla, proto poté vkládáme prázdné **print()**.

print("Mala nasobilka: ")

for i in range(1,11):

print(i, end=" ")

print()

for i in range(1,11):

print(i\*2, end=" ")

print()

for i in range(1,11):

print(i\*3, end=" ")

print()

for i in range(1,11):

print(i\*4, end=" ")

print()

for i in range(1,11):

print(i\*5, end=" ")

print()

for i in range(1,11):

print(i\*6, end=" ")

print()

for i in range(1,11):

print(i\*7, end=" ")

print()

for i in range(1,11):

print(i\*8, end=" ")

print()

for i in range(1,11):

print(i\*9, end=" ")

print()

for i in range(1,11):

print(i\*10, end=" ")

print()

Napsaný program je funkční, nicméně zbytečně zdlouhavý. V podstatě 10x dokola děláme to samé, pouze násobíme proměnnou i vždy vyšším číslem, než u předchozího cyklu. Pokud bychom zavedli ještě jednu proměnnou a použili dva cykly, situace se výrazně zjednoduší:

print("Mala nasobilka: ")

for i in range(1,11):

for j in range(1,11):

print(i\*j, end=" ")

print()

Do bloku for můžeme přidat i příkaz break. Příkaz break z cyklu vyskočí.

print("Program zjistí zda dané slovo obsahuje samohlásky.")

slovo = input("Zadejte slovo: ")

samohlasky = False

for znak in slovo:

if znak in "aáeéěiíoóuúůyý":

samohlasky = True

**break**

if samohlasky:

print(slovo, "obsahuje samohlásky.")

else:

print(slovo, "neobsahuje samohlásky.")

Příklad: Program zjistí, z kolika souhlásek, samohlásek, číslic a případných dalších jiných znaků se skládá zadané slovo.

print("Program zjistí z čeho se skládá slovo.")

slovo = input("Zadejte slovo: ")

samohlasky = 0

souhlasky = 0

ostatni = 0

cisel = 0

for znak in slovo:

if znak in "aáeéěiíoóuúůyý":

samohlasky = samohlasky + 1

elif znak in "bcčdďfghjklmnňpqrřsštťvwxzž":

souhlasky = souhlasky + 1

elif znak in "0123456789":

cisel = cisel + 1

else:

ostatni = ostatni + 1

print(slovo, "má: ")

print("samohlásek", samohlasky)

print("souhlásek", souhlasky)

print("čísel", cisel)

print("ostatních znaků...", ostatni)

## WHILE cyklus

Tento cyklus testuje podmínku **před** průchodem cyklem. Cyklus tedy nemusí proběhnout ani jednou. Příkaz while používáme, závisí-li ukončovací podmínka na nějakém příkazu v těle cyklu. Dokud je podmínka splněna, provádí se příkazy

Cyklus while se zapisuje následovně:

while (podminka):

{

příkazy

}

Jednoduchý příklad – vypiš čísla 1 až 10:

i=1

while (i<=10):

print(i)

i=i+1

Jednoduchý příklad – načítej čísla, dokud nebude zadána nula:

a = 1

while (a != 0):

a = int(input("Zadejte číslo: "))

Příklad na sčítání dvou čísel zadaných z klávesnice s dotazem k uživateli, zda chce proces opakovat s další dvojicí čísel:

pokracovat = 1

while (pokracovat == 1):

a = int(input("Zadejte prvni číslo: "))

b = int(input("Zadejte druhé číslo: "))

print ("Součet zadaných čísel je:",a+b)

pokracovat = int(input("Přeješ si pokračovat? 1 = Ano, 0 = Ne: "))

Příklad: Program vypíše všechna sudá čísla, a to až do maximálního čísla, které uživatel zadá. Např.: bude zadáno 8, vypíše se 0, 2, 4, 6, 8

print("Program vypíše všechna sudá čísla.")

maximum = int(input("Zadejte maximální číslo: "))

cislo = 0

while (cislo <= maximum):

if (cislo % 2 == 0): #číslo je sudé

print(cislo)

cislo = cislo + 1

input("\nProgram ukončíte stisknutím libovolné klávesy...")

Příklad: program bude od uživatele načítat čísla, dokud nezadá 999. Poté se zobrazí průměr ze zadaných čísel.

print("Program bude načítat čísla, dokud nezadáte 999")

cislo = int(input("Zadejte číslo: "))

suma=0

pocet=0

while (cislo != 999):

suma = suma + cislo

pocet = pocet + 1

cislo = int(input("Zadejte číslo: "))

prumer = suma/pocet

print("Průměr zadaných čísel je:", prumer)

**Generování (pseudo)náhodných čísel**

Program vygeneruje náhodné číslo v rozsahu 10 až 15.

import random

random.randint(10,15)

Výpis náhodného čísla v rozsahu 1 až 100:

import random

print(random.randint(1,100))

Program vygeneruje dvě náhodná čísla v rozsahu 1 až 20, s těmito čísly pak provede základní matematické operace (sčítání, odčítání, násobení, dělení) a vše přehledně zobrazí na monitoru.

import random

a=random.randint(1,20)

b=random.randint(1,20)

print("První číslo:", a)

print("Druhé číslo:", b)

print ("a + b = ",a+b)

print ("a - b = ",a-b)

print ("a \* b = ",a\*b)

print ("a / b = ",a/b)

Příklad:

Program vygeneruje náhodné číslo z intervalu 0 až 9 a uživatel se ho pokusí uhádnout. Program ho bude informovat, zda hádané číslo je vyšší, nebo nižší než jeho tip. Až uživatel číslo uhádne, bude informován o tom, kolik pokusů k tomu potřeboval.

import random

nahodne=random.randint(0,9)

pocet =1

tip=int(input("Tipni si číslo v rozsahu 0 až 9:"))

while (tip != nahodne):

if (tip > nahodne):

print("Hledané číslo je menší")

elif (tip < nahodne):

print("Hledané číslo je větší")

pocet=pocet+1

tip=int(input("Tipni si číslo v rozsahu 0 až 9:"))

print("Číslo bylo uhádnuto na", pocet,"pokus")

Vytvořte program, který bude tisknout: Kolik je 7 x 2 ?, kde ty 2 čísla (7 i 2) jsou náhodně generovány v rozsahu 1 a 10 (včetně). Poté, co si uživatel tipne výsledek, zobrazí se slovní hodnocení jeho odpovědi (tj. „správně“, případně „špatně“) a uživateli bude nabídnuta možnost pokračovat. Pokud pokračovat chce, budou mu vygenerována dvě nová čísla a proces se opakuje. Až bude chtít uživatel skončit, dozví se kolik a z kolika zadaných příkladů měl správně.

import random

a=random.randint(1,10)

b=random.randint(1,10)

spravne=0

pocet=0

pokracovat=1

while (pokracovat==1):

print("Kolik je:",a,"\*",b)

tip=int(input("Tip: "))

vysledek = a\*b

if(tip == vysledek):

print("Správně!")

spravne=spravne+1

else:

print("Špatně")

a=random.randint(1,10)

b=random.randint(1,10)

pocet=pocet+1

pokracovat=int(input("Chcete pokračovat? 1 = ano: "))

print("Z",pocet,"příkladů jste měli", spravne,"výsledky správně")