

PYTHON

For cyklus

Vypracovala: Ing. Eva Gabonayová

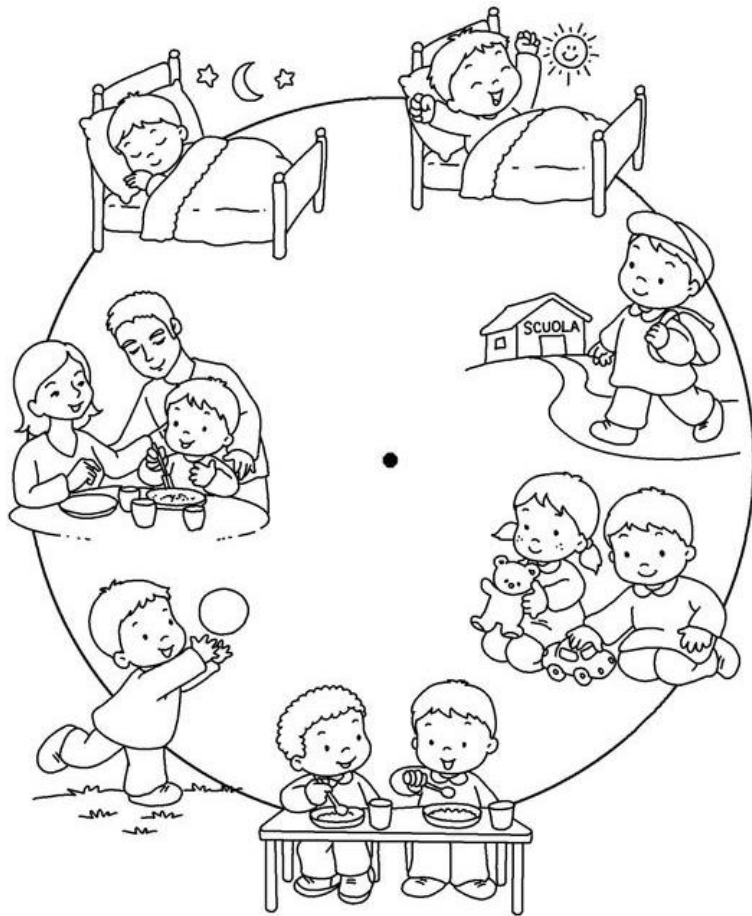
Predmet: Informatika

Vzdelávacia oblasť: Matematika a práca s informáciami

Dátum spracovania: 23.5.2021



Cyklus



Cyklus

- Programová konštrukcia **cyklus** nám poskytuje prostriedok umožňujúci opakovať činnosť alebo činnosti.
- Pri jeho použití je potrebné vedieť, čo sa má opakovať a **dokedy** sa to má opakovať.
- Činnosť, ktorá sa opakuje, označujeme ako **telo cyklu**, podmienku, ktorá určuje dokedy sa bude telo cyklu opakovať, nazývame **podmienka cyklu**.

Cyklus FOR



- cyklus so známym počtom opakovaní
- vieme presne, koľkokrát telo cyklu prebehne

Príklad - náhrdelník



```
pero . penup ()
```

```
koralky ()
```



```
koralky ()
```



```
koralky ()
```



```
koralky ()
```



Príklad



```
pero.penup()
```

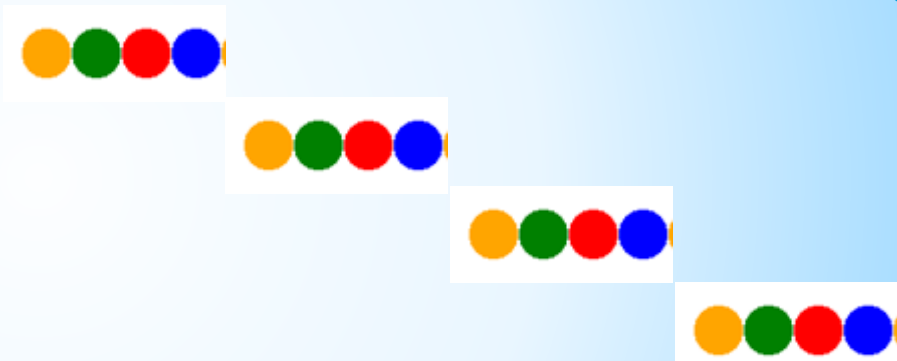
```
koralky()
```

```
koralky()
```

```
koralky()
```

```
koralky()
```

```
for i in range(4):  
    koralky()
```



Príklad



```
pero.penup()  
koralky()  
koralky()  
koralky()  
koralky()
```

```
for i in range(4):  
    koralky()
```

Cyklus FOR aj vlastná funkcia umožňujú skrátit' a sprehľadnit' programový kód.

... ako to funguje

```
for i in range(4) :      Hlavička cyklu  
    koralky()           Telo cyklu
```

- do **premennej i** sa bude postupne prirad'ovat' nasledovná hodnota zo zoznamu hodnôt (1,2,3,4); začíname s prvou hodnotou, teda v tomto prípade 1
- pre každú hodnotu so zoznamu sa vykonajú príkazy, ktoré sú v **tele cyklu**, t.j. tie príkazy, ktoré sú odsadené
- v našom príklade koralky()
- všimnite si znak **,:'** na konci riadka s for - ten je tu povinne, bez neho by to nefungovalo

... ako to funguje

```
for i in range(4):  
    koralky()
```

Telo cyklu

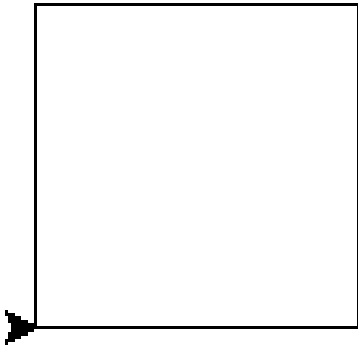
- tvoria príkazy, ktoré sa majú opakovať; definujú sa **odsadením** príslušných riadkov
- odsadenie je povinné a musí byť minimálne 1 medzera, odporúča sa odsadzovať vždy o **4 medzery**
- ak telo cyklu obsahuje viac príkazov, všetky musia byť odsadené o rovnaký počet medzier
- telo cyklu nesmie byť prázdne, musí obsahovať aspoň jeden príkaz

Príklad - štvorec

```
def stvorec() :  
    pero.forward(100)  
    pero.left(90)  
    pero.forward(100)  
    pero.left(90)  
    pero.forward(100)  
    pero.left(90)  
    pero.forward(100)  
    pero.left(90)
```

Príklad - štvorec

```
def stvorec():  
    for i in range(4):  
        pero.forward(100)  
        pero.left(90)
```



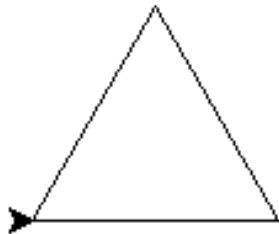
```
def stvorec():  
pero.forward(100)  
pero.left(90)  
pero.forward(100)  
pero.left(90)  
pero.forward(100)  
pero.left(90)  
pero.forward(100)  
pero.left(90)
```

Príklad - trojuholník

```
def trojuholnik():  
    pero.forward(100)  
    pero.left(120)  
    pero.forward(100)  
    pero.left(120)  
    pero.forward(100)  
    pero.left(120)
```

Príklad - trojuholník

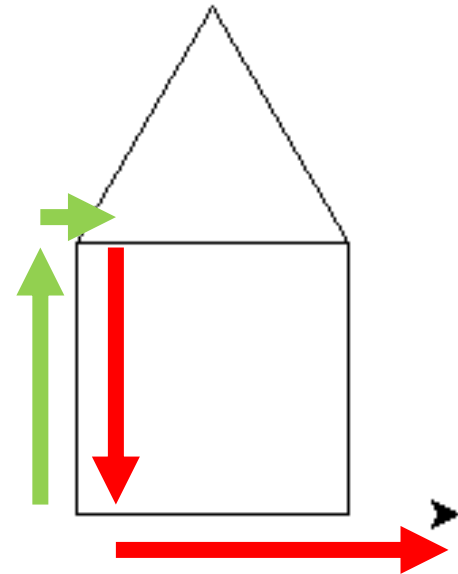
```
def trojuholnik():  
    for i in range(3):  
        pero.forward(100)  
        pero.left(120)
```



```
def trojuholnik():  
    pero.forward(100)  
    pero.left(120)  
    pero.forward(100)  
    pero.left(120)  
    pero.forward(100)  
    pero.left(120)
```

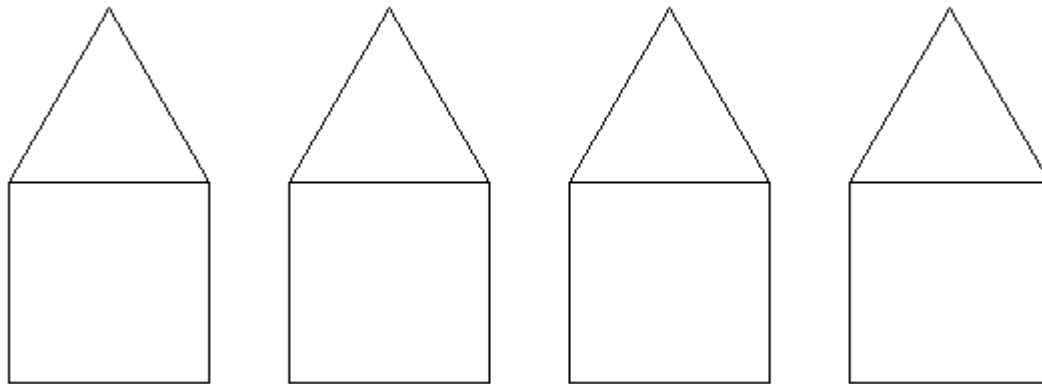
Príklad - dom

```
def dom():  
    stvorec()  
    posun1()  
    trojuholnik()  
    posun2()
```



Príklad - ulica

```
def ulica():  
    for i in range(4):  
        dom()
```



Ďakujem za pozornosť!

A decorative graphic element consisting of several horizontal lines of varying lengths and colors (light blue and white) extending from the right side of the text area towards the right edge of the slide.