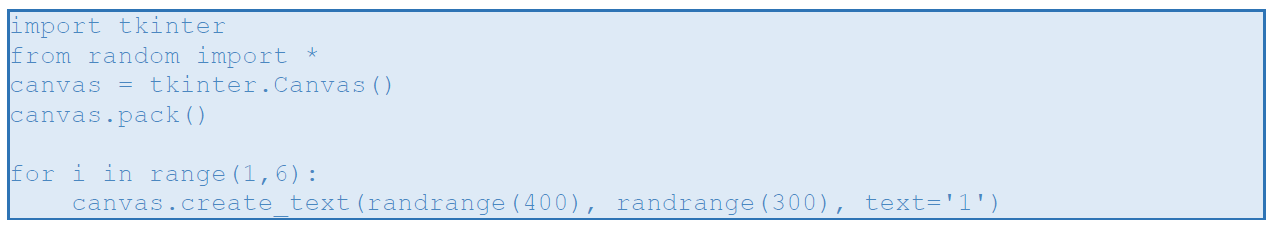
3. hodina – PRÍKAZ cyklu FOR

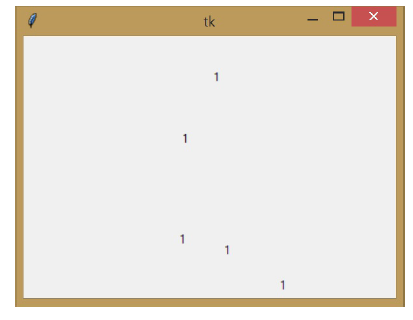
Cyklus FOR má tvar: for i in range(od,do,krok):

Napríklad:



Program bude päťkrát opakovať príkaz create\_text a zatiaľ zakaždým vypíše 1. Zápisom

for i in range(1, 6): sme zapísali, že nasledujúca časť programu sa bude opakovať a postupne si budeme v premennej i pamätať čísla uvedené v zátvorke, čiže 1 až 5. Aj keď je tam zapísané 1 až 6, pre hodnotu 6 sa už telo cyklu nezopakuje. Takto bude vyzerať naše okno:

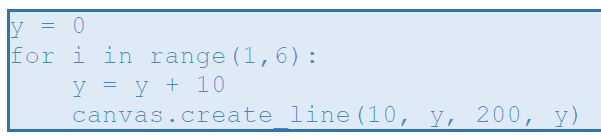


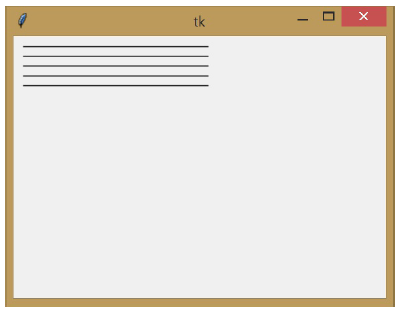
Keď nahradíme zápis text='1' zápisom text=i, bude sa nám namiesto čísla 1 vypisovať hodnota v

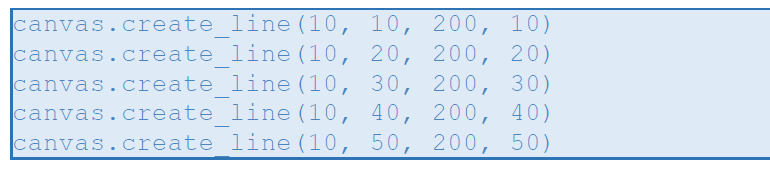
premennej i, čiže postupne čísla 1 až 5.

Pri FOR cykle si musíme dávať pozor na **správne odsadenie príkazov**.

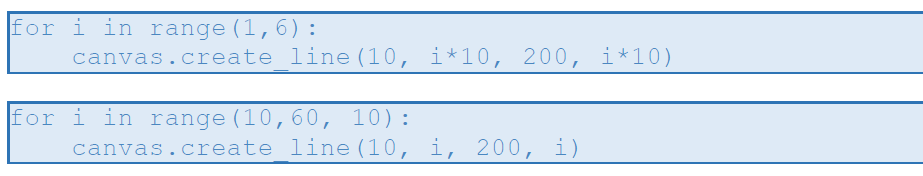
Zistite, čo nakreslí tento program:



Premennej y ešte pred cyklom nastavíme hodnotu na nulu. Potom sa bude opakovať cyklus postupne pre i s hodnotami 1, 2, 3, 4 a 5. Zakaždým sa v programe zvýši hodnota premennej y o desať, čiže v y budú postupne hodnoty 10, 20, 30, 40, 50 a pre tieto hodnoty sa zakaždým nakreslí vodorovná čiara. Táto čiara má rovnakú y-novú súradnicu, ako je práve hodnota premennej y, a čiara začína na x-ovej súradnici 10 a končí na x-vej súradnici 200. Ak si za premenné postupne dosadzujeme hodnoty, ktoré budú mať, v skutočnosti sa nakreslia čiary s týmito hodnotami súradníc.

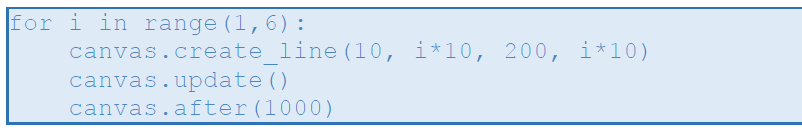


Rovnaký efekt – vykreslenie rovnakej notovej osnovy – získame aj nasledujúcimi dvomi zápismi. Porozmýšľajte a zdôvodnite, prečo aj tieto programy nakreslia to isté.

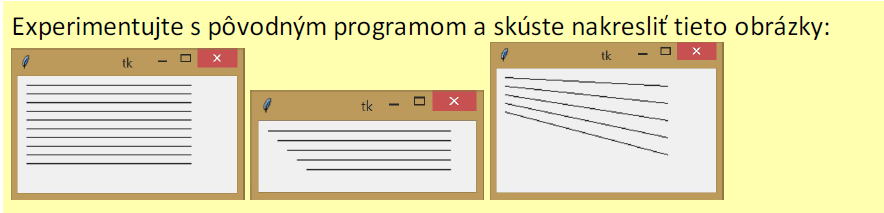


V tomto programe sme v zápise range použili tri čísla. Tretím číslom určíme, akým krokom sa má hodnota premennej i meniť. Ak ju neuvedieme ako doposiaľ, automaticky je krok nastavený na 1.

For cyklus opakuje príkazy veľmi rýchlo, my teda vidíme hneď výsledok. Ak by sme chceli vykresľovanie spomaliť, môžeme v cykle použiť príkaz canvas.after(1000). Tento príkaz pozdrží vykonávanie programu na čas uvedený v zátvorke. Čas uvádzame v milisekundách, čiže canvas.after(1000) čaká 1 sekundu. K tomuto príkazu musíme ešte použiť príkaz canvas.update(), ktorý zabezpečí, aby sa aj medzi čakaniami aktualizovalo grafické plátno. Ak by sme ho nepoužili, program by pracoval pomalšie (s čakaním) a grafické plátno by sa nám prekreslilo až po skončení for cyklu. Hotový program zapíšeme takto:

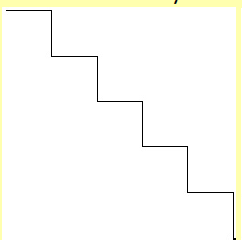


Úlohy na precvičenie:



Úloha na odovzdanie:

Pomocou FOR cyklu nakreslite schodisko:



Kto chce, môže si skúsiť aj schodisko nahor:

