

PRŮVODCE HODINOU I-2

V této hodině nejprve rozšíříme příklad „Ahoj_sвете“ z minulé hodiny. Později přidáme dva další příklady (jeden z nich ve dvou modifikacích). Na těchto příkladech si současně připomeneme nebo se naučíme psaní cyklů v Pythonu.

Co bude v této hodině potřeba:

- PC s editorem Mu
- Micro:bit s USB kabelem
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekci
- Pracovní listy pro studenty

1. krok 10 minut

Řešte úlohu s nekonečným výpisem textu „Ahoj světe“. Vysvětlíte studentům význam cyklu while True:

```
from microbit import *
while True:
    display.scroll("Ahoj světe")
    sleep(1000)
```

Obraťte pozornost na správnou syntaxi.

- True musí být s velkým T.
- Na konci řádku se zápisem cyklu musí být znak dvojtečky. Na ten se často zapomíná.
- Odsazení těla cyklu musí být o právě čtyři mezery. Editor Mu vám pomůže automatickým odsazením dalších řádků. Studenti mohou být z jiných verzí Pythonu například zvyklí na používání tabulátoru – to zde není možné.

2. krok – 25 minut

Napište dvěma různými způsoby program, který vypíše čísla od 1 do 10 a pak skončí.

Použijete postupně cykly for a while.

Zápis s cyklem for:

```
from microbit import *
for i in range(1, 11):
    display.scroll(i)
```

Vysvětlíte syntaxi programu. Jedná se o *cyklus s pevným počtem opakování*. Je třeba vysvětlit, že zápis range (1, 11) znamená v Pythonu rozsah od 1 do 10. Je to vždy o jednu méně, než je

mez vpravo – **častý zdroj chyb**. Dále je třeba říct, že za čárkou v závorce musí být mezera. Zkuste studentům říct některé další příklady zápisu range:

- `range(3, 1)` – cyklus se neprovede (3 je větší než 1)
- `range(1, 4, 2)` – cyklus se provede pro 1 a 3. Iterace je po dvou. Význam dosazuj do proměnné od jedné čísla zvětšená o 2 dokud je číslo menší než 4.
- `range(3, 1, -1)` – cyklus se provede pro 3 a 2. Sestupná iterace.

Nechte studenty případně vyzkoušet. Pokud to potřebujete, popis syntaxe cyklu `for` a rozsahu `range` je v příloze učebnice.

Zeptejte se studentů

Jaký je rozdíl mezi řetězcem (`string`) a celým číslem (`integer`)?

Řetězec je posloupnost znaků anglické abecedy, čísel a dalších znaků. Přesněji definováno, jedná se o znaky s ASCII kódem 32 až 127.

Definici celého čísla známe z matematiky. Zcela přesně v našem případě je to číslo bez desetinné čárky z intervalu `<-2147483648, 2147483647>`.

Zápis s cyklem `while`:

Jedná se o cyklus s neurčitým počtem opakování.

Pohovořte o dané syntaxi. Za příkazem `while` následuje podmínka a cyklus (odsazené řádky za dvojtečkou) se provádí tak dlouho dokud podmínka platí.

```
from microbit import *
i = 1
while (i < 11):
    display.scroll(i)
    i = i + 1
```

Všimněte si práce s proměnnými – definice proměnné na řádku 3 a změny její hodnoty na řádku 6. Pozor kolem znaku `=` a matematických operátorů **musí být mezery** – Mu je tam vyžaduje, ačkoliv v jiných verzích Pythonu nejsou nutné. Ukažte to studentům na kontrole syntaxe.

Pozor na mezery. Např. V zápisu: `i = i + 1` musí být mezera jak kolem znaku `=` tak okolo znaku `+`.

Zkuste se studentů zeptat, zda je jim bližší zápis s `for` nebo s `while`?

Vysvětlíte pojem negace podmínky – `not(podmínka)`

Je totéž jako v našem příkladu `while not(i > 11):` ?

Není správně by bylo: `while not(i >= 11):`

3. krok – 10 minut

Ukázka dalších funkcí pro objekt `display`:

```
from microbit import *  
display.show("X")  
sleep(1000)  
display.clear()
```

Příklad zobrazí znak X pomocí příkazu `display.show()` po dobu jedné sekundy a pak smaže displej pomocí `display.clear()`. Funkce `display.show()` zobrazí řetězec nebo číslo znak po znaku. Prodlevu mezi změnami lze nastavit (v milisekundách pomocí parametru. Např. `display.show("Ahoj svete", 1000)`). Poslední znak zůstane svítit na displeji. Pokud je parametrem jen jeden znak nebo jednociferné číslo, pak zůstane svítit tak dlouho, než zobrazíte něco jiného nebo smažete obrazovku pomocí `display.clear()`.