

## 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:

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.
- Na konci musí být odřádkování a na posledním řádku nesmí být nic (ani mezery).

```
1. from microbit import *
2.
3. while True:
4.     display.scroll("Ahoj světe")
5.     sleep(1000)
```

### 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:**

```
1. from microbit import *
2.
3. for i in range(1, 11):
4.     display.scroll(str(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 říci, že za čárkou musí být mezera. Zkuste studentům říci některé 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.
- `range(3, 1, -1)` – cyklus se provede pro 3 a 2.

Nechte studenty případně vyzkoušet.

Zeptejte se studentů

Proč je uvnitř `display.scroll()` příkaz `str()`?

Protože `display.scroll()` umí vypsat pouze řetězec, použijeme příkaz `str()`, který převede celé číslo na řetězec.

Jaký je rozdíl mezi řetězcem (stringem) a celým číslem (integerem)?

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 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.

```
1. from microbit import *
3.
4. i = 1
5. while (i < 11):
6.     display.scroll(str(i))
7.     i = i + 1
```

Všimněte si práce s proměnnými – definice proměnné na řádce 3 a změny její hodnoty na řádce 6  
Pozor kolem `=` a matematických operátorů **musí být mezery**.

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

## 3. krok – 10 minut

Ukázka dalších funkcí pro objekt `display`:

```
1. from microbit import *
2. display.show("X")
3. sleep(1000)
4. display.clear()
```

Příklad zobrazí znak X pomocí `display.show()` po dobu jedné sekundy a pak smaže displej pomocí `display.clear()`.

`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 toto zůstane svítit tak dlouho než zobrazíte něco jiného nebo smažete obrazovku pomocí `display.clear()`.