

PRŮVODCE HODINOU II-1

Co bude v této hodině potřeba:

- PC se editorem mu.
- Micro:bit s USB kabelem zakončeným micro USB. Pozor nefungují všechny kabely. Pokud budete používat jiné než koupené spolu s micro:bitem, je nutné je předem vyzkoušet.
- Pokud je k dispozici, tak dataprojektor
- Prezentaci k této lekce
- Pracovní listy pro studenty

1. krok 5 minut

Vysvětlíte, že micro:bit obsahuje celkem tři tlačítka. Dvě na přední straně a jedno na zadní straně.

Tlačítko umístěné na zadní straně nás nebude v této lekci zajímat. Nelze jej programovat a slouží k resetu micro:bitu.

Na přední straně pak má dvě programovatelná tlačítka označená A a B. V MicroPythonu pro ně existují dvě proměnné `button_a` a `button_b`. Pokud studenti ví co je to objektové programování, použijte správný pojem objekty `button_a` a `button_b`. Můžete rovněž využít tuto látku k prvnímu seznámení s objektovým programováním.

2. krok 10 minut

Začněte tímto jednoduchým příkladem:

```
1. from microbit import *
2. while True:
3.     if button_a.is_pressed():
4.         display.show(Image.HAPPY)
5.     if button_b.is_pressed():
6.         display.show(Image.SAD)
7.     sleep(100)
8.     display.clear()
```

Příkazy pro dotaz zda je tlačítko stisknuto jsou na řádcích 4 a 6. Nechte studenty přijít na jejich význam – test stisku tlačítka. Funkce (metoda) vrací 1 (stisknuto) nebo 0 (nestisknuto). 1 znamená splněná podmínka (True), 0 nesplněná (False). Existuje i funkce `button_a.was_pressed()`, která testuje, zda tlačítko bylo stisklé od minulého testování nebo od začátku programu.

Pozor na správné odsazení bloku ve druhé úrovni (po `if`) – musí být 8 mezer.

3. krok 15 minut

Nyní si vysvětlíme význam logických spojek **and** (a) a **or** (nebo). Začněte tímto příkladem:

```
1. from microbit import *
2.
3. while True:
4.     if (button_a.is_pressed()) and (button_b.is_pressed()):
5.         display.show(Image.HEART)
6.         sleep(100)
7.         display.clear()
```

Spojka **and** mezi dvěma testy na řádce 4 má význam a – obě podmínky musí být splněny současně.

Program nepatrně změňte (pouze na řádce 4):

```
1. from microbit import *
2.
3. while True:
4.     if (button_a.is_pressed()) or (button_b.is_pressed()):
5.         display.show(Image.HEART)
6.         sleep(100)
7.         display.clear()
```

Spojka **or** má význam nebo. Stačí když je splněna alespoň jedna z podmínek.

4. krok 15 minut

Funkce `get_presses()` vrací počet stisknutí daného tlačítka od startu nebo poslední kontroly.

Napište a spusťte následující kód:

```
1. from microbit import *
2.
3. sleep(10000)
4. display.show(str(button_a.get_presses()))
```

Pozor na složitou konstrukci na řádce 4 a správný počet závorek. Jedná se vlastně o složenou funkci. Pokud již měli složenou funkci studenti v matematice – můžete porovnat. I zde se funkce vyhodnocují od vnitřní (dotaz na počet stisků) k vnější (zobrazení na displej).

Pokud zbyde čas, nechte studenty příklad přepsat bez složené funkce. Měli by mít něco jako:

```
1. from microbit import *
2.
3.
4. sleep(10000)
5. a = button_a.get_presses()
6. b = str(a)
7. display.show(b)
```

První zápis je kratší, druhý pro začátečníka srozumitelnější.

Pokud postupujete přímo podle curricula, budete na příští hodinu potřebovat ke každému micro:bitu dva vodiče s krokodýly a nějaký hardware pro zvukový výstup.

Řekněte studentům ať si přinesou na příští hodinu sluchátka nebo repráček s jackem. Případně si připravte piezzobuzzer.

Doporučujeme, aby studenti měli sluchátka ať se vzájemně nepřehlušují rámušem. Vy si naopak připravte repráčky ať vše můžete dobře demonstrovat.