

PRŮVODCE TEORIÍ

První program – Hello world

Otevřete si editor *mu* a stiskněte tlačítko New. Měli byste vidět následující text:

```
1. from microbit import *
2.
3. # Write your code here :-)
```

Pozor čísla řádků nejsou součástí kódu, ale odpovídají číslům řádku v programu *mu*. V této publikaci uvádím čísla řádků pro snazší odkazování na konkrétní místa ve zdrojovém kódu.

Program by měl končit odřádkováním a na posledním řádku nesmí být žádné znaky (ani mezery).

Na řádku 1 se zavádí systémová knihovna, která zavádí potřebné funkce a metody pro práci s micro:bitem. Tímto řádkem **musí** začínat všechny vaše programy.

Znak # na začátku třetího řádku znamená, že se jedná o komentář. Tento řádek můžete klidně smazat a kód programu psát místo něj. Je možné i psát na řádek 2 anebo pod komentář na řádek 4.

Zkuste pro začátek následující kód:

```
1. from microbit import *
2.
3. display.scroll("Ahoj svete")
```

Popis: řádek 3 znamená, že po displeji micro:bitu má běžet kód uvedený v uvozovkách. Stiskněte tlačítko Save a kód uložte. Programu můžete dát příponu .py (např. ahoj_svete.py). Pokud jí neuvedete, bude mu přiřazena automaticky.

Nyní připojte svůj micro:bit pomocí USB kabelu k počítači. Počkejte asi pět vteřin a pak stiskněte tlačítko Flash. Vyčkejte až přestane blikat žlutá LED dioda na micro:bitu a pak byste měli vidět, jak váš text přeběhne přes displej micro:bitu.

Nyní můžete vyzkoušet následující modifikaci kódu:

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

Popis: Na řádku 3 je nyní zaveden tzv. nekonečný cyklus. Jeho příkazy jsou odsazené o čtyři mezeríky od začátku řádků. Pozor – je třeba dodržet stejný počet mezer (může být i vyšší, ale násobek čtyř) a nelze použít tabulátor. Na řádku 5 je pak příkaz sleep – čekej 1000 milisekund – 1 sekundu.

Program v nekonečné smyčce vypisuje text a pak čeká jednu sekundu.

Další příklady

Zadání: Napište program, který vypíše čísla od jedné do deseti a pak skončí.

Řešení:

```
1. from microbit import *
2.
3. for i in range(1, 11):
4.     display.scroll(str(i))
```

Popis: Na řádce 3 je zaveden cyklus s pevným počtem opakování. Hodnota proměnné `i` se mění dle rozsahu intervalu `range(a, b)` od `a` do `b-1`. Chcete-li tedy od 1 do 10 musíme psát takto. Za čárkou v intervalu musí být mezera. Pozor na konci řádku je dvojtečka, tady se také často dělá chyba. Na řádce čtyři je pak výpis čísla. Protože funkce `display.scroll` pracuje pouze s řetězci musíte příkazem `str(i)` převést číslo na řetězec.

Zadání: Řešte předchozí příklad pomocí funkce `while`

Řešení:

```
1. from microbit import *
2.
3. i = 1
4. while (i < 11):
5.     display.scroll(str(i))
6.     i = i + 1
```

Popis: Na řádce 3 do proměnné `i` přiřadíte hodnotu 1. Pozor okolo = jsou vyžadovány mezery. Na řádce 4 je cyklus, který se opakuje dokud je `i` menší než 11. Pozor kolem nerovnosti musí být mezery a na konci řádku je dvojtečka. Na řádce 6 zvyšujeme hodnotu proměnné `i` o jedničku. Pozor opět na chybějící mezery.

Zadání: Po dobu jedné vteřiny zobraz na displeji písmeno X.

Řešení:

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

Popis: Na řádce 3 zobrazíte písmeno X (opět nutno jako řetězec). Na řádce 4 čeká program jednu sekundu a příkaz na řádce 5 smaže displej.

Přednastavené obrázky

MicroPython obsahuje asi padesát připravených obrázků. Ukázka jejich použití je v následujícím kódu:

```
1. from microbit import *
2.
3. display.show(Image.SAD)
4. sleep(1000)
5. display.show(Image.SMILE)
6. sleep(1000)
7. display.show(Image.HAPPY)
8. sleep(1000)
9. display.clear()
```

Zobrazení obrázků je na řádcích 3, 5 a 7. Jak je vidět, jedná se o konstanty začínající slovem Image.

Seznam všech obrázků naleznete v příloze A anebo v dokumentaci MicroPythonu pro micro:bit.

Příklad: Pomocí konstant obrázků `Image.HEART` a `Image.HEART_SMALL`, simulujte úder srdce.

Řešení:

```
1. from microbit import *
2.
3. for i in range(1, 100):
4.     display.show(Image.HEART)
5.     sleep(400)
6.     display.show(Image.HEART_SMALL)
7.     sleep(400)
8. display.clear()
```

Vlastní obrázky

Příklad: Zobrazte na displeji obrázek rakety

Řešení:

```
9. from microbit import *
10.
11. raketa = Image("00900:"
12.                "05550:"
13.                "05550:"
14.                "09990:"
15.                "90909:")
16.
17. display.show(raketa)
```

Popis: Struktura na řádcích 3 až 7 popisuje obrázek. Pětice čísel ukončených dvojtečkou uzavřená do apostrofů popisuje vždy jeden řádek displeje shora dolů. Číslo pak znamená intenzitu světla od 0 (dioda nesvítí) po 9 (dioda svítí naplno). Na řádku 9 je pak obrázek zobrazen.

Nyní si na základě tohoto příkladu sestojíme pohyblivý obrázek startující rakety. Zdrojový kód je následující:

```
1. from microbit import *
2.
3. raketa1 = Image("00900:"
4.                "05550:"
5.                "05550:"
6.                "09990:"
7.                "90909:")
8.
9. raketa2 = Image("00900:"
10.               "05550:"
11.               "05550:"
12.               "09990:"
13.               "99999:")
14.
15. raketa3 = Image("05550:"
16.                "05550:"
17.                "09990:"
18.                "99999:"
19.                "00000:")
20.
21. raketa4 = Image("09990:"
22.                "99999:"
23.                "00000:"
24.                "00000:"
25.                "00000:")
26.
27. raketa5 = Image("99999:"
28.                "00000:"
29.                "00000:"
30.                "00000:"
31.                "00000:")
32.
```

```

33.raketa6 = Image("00000:"
34.           "00000:"
35.           "00000:"
36.           "00000:"
37.           "00000:")
38.
39.raketa = [raketa1, raketa2, raketa3, raketa4, raketa5,
           raketa6]
40.display.show(raketa, delay=500)

```

Na řádcích 3 až 37 je postupně šest obrázků, označených raketa1 až raketa6. Na řádce 39 je z těchto obrázků sestavena struktura zvaná list (seznam). Ta je pak na řádce 40 postupně zobrazována, kdy mezi jednotlivými snímky je pauza půl sekundy.

Práce s konkrétní diodou

Příklad: Sestrojte program, který bude náhodně rozsvěcet jednotlivé diody s různou intenzitou světla.

Řešení:

```

1.from microbit import *
2.import random
3.
4.while True:
5.    x = random.randint(0, 4)
6.    y = random.randint(0, 4)
7.    intenzita = random.randint(0, 9)
8.    display.set_pixel(x, y, intenzita)
9.    sleep(10)

```

Program používá generátor náhodných čísel. Pro jeho použití je nutné načíst knihovnu random na řádce 2. Na řádcích 5 až 7 je pak tento generátor volán funkcí random.randint, která má dva parametry a, b a vrací náhodné celé číslo z uzavřeného intervalu <a,b>.

Funkce na řádce 8 display.set_pixel má tři parametry x, y, intenzita a nastavuje na souřadnicích x (sloupec) a y (řádek) diodu na intenzitu (0 až 9). Bod 0,0 je vlevo nahoře, vpravo dole pak 4,4.

Použití funkce sleep je nutné jinak dochází k příliš rychlému „blikání“.

Příklad: Upravte předchozí zadání tak, že budete nastavovat pouze dvě úrovně intenzity (0 a 9) a to tak, že budete náhodně vybírat souřadnice a pokud dioda na dané souřadnici nebude svítit, tak jí rozsvítíte a naopak.

Řešení:

```
1. from microbit import *
2. import random
3.
4. while True:
5.     x = random.randint(0, 4)
6.     y = random.randint(0, 4)
7.     if (display.get_pixel(x, y)):
8.         display.set_pixel(x, y, 0)
9.     else:
10.        display.set_pixel(x, y, 9)
11.    sleep(10)
```

Popis: Na řádce sedm je použita funkce `display.get_pixel(x, y)`. Ta vrací hodnotu svícení dané diody. V našem příkladě využíváme toho, že pokud dioda svítí, vrátí hodnotu větší než nula a tudíž je podmínka splněna.

Pozor druhá úroveň odsazení musí být opět násobek čtyř a je tedy osm mezer.

PŘÍLOHA – SEZNAM PŘIPRAVENÝCH OBRÁZKŮ

- `Image.HEART`
- `Image.HEART_SMALL`
- `Image.HAPPY`
- `Image.SMILE`
- `Image.SAD`
- `Image.CONFUSED`
- `Image.ANGRY`
- `Image.ASLEEP`
- `Image.SURPRISED`
- `Image.SILLY`
- `Image.FABULOUS`
- `Image.MEH`
- `Image.YES`
- `Image.NO`
- `Image.CLOCK12`, `Image.CLOCK11`, `Image.CLOCK10`, `Image.CLOCK9`, `Image.CLOCK8`, `Image.CLOCK7`, `Image.CLOCK6`, `Image.CLOCK5`, `Image.CLOCK4`, `Image.CLOCK3`, `Image.CLOCK2`, `Image.CLOCK1`
- `Image.ARROW_N`, `Image.ARROW_NE`, `Image.ARROW_E`, `Image.ARROW_SE`, `Image.ARROW_S`, `Image.ARROW_SW`, `Image.ARROW_W`, `Image.ARROW_NW`
- `Image.TRIANGLE`
- `Image.TRIANGLE_LEFT`
- `Image.CHESSBOARD`
- `Image.DIAMOND`
- `Image.DIAMOND_SMALL`
- `Image.SQUARE`
- `Image.SQUARE_SMALL`
- `Image.RABBIT`
- `Image.COW`
- `Image.MUSIC_CROTCHET`

- `Image.MUSIC_QUAVER`
- `Image.MUSIC_QUAVERS`
- `Image.PITCHFORK`
- `Image.XMAS`
- `Image.PACMAN`
- `Image.TARGET`
- `Image.TSHIRT`
- `Image.ROLLERSKATE`
- `Image.DUCK`
- `Image.HOUSE`
- `Image.TORTOISE`
- `Image.BUTTERFLY`
- `Image.STICKFIGURE`
- `Image.GHOST`
- `Image.SWORD`
- `Image.GIRAFFE`
- `Image.SKULL`
- `Image.UMBRELLA`
- `Image.SNAKE`