

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

Cyklus

Nitra 2024

Obsah

1	Príkaz "opakovať 4 krát"	3
2	Príkaz "zatiaľ čo pravda"	4
3	Príkaz "pre index od 0 do 9"	5
4	Príkaz "pre prvok hodnota z list"	6
5	Príkaz "každých 500 ms"	7
6	Príkaz "preruš"	8
7	Príkaz "pokračuj"	9

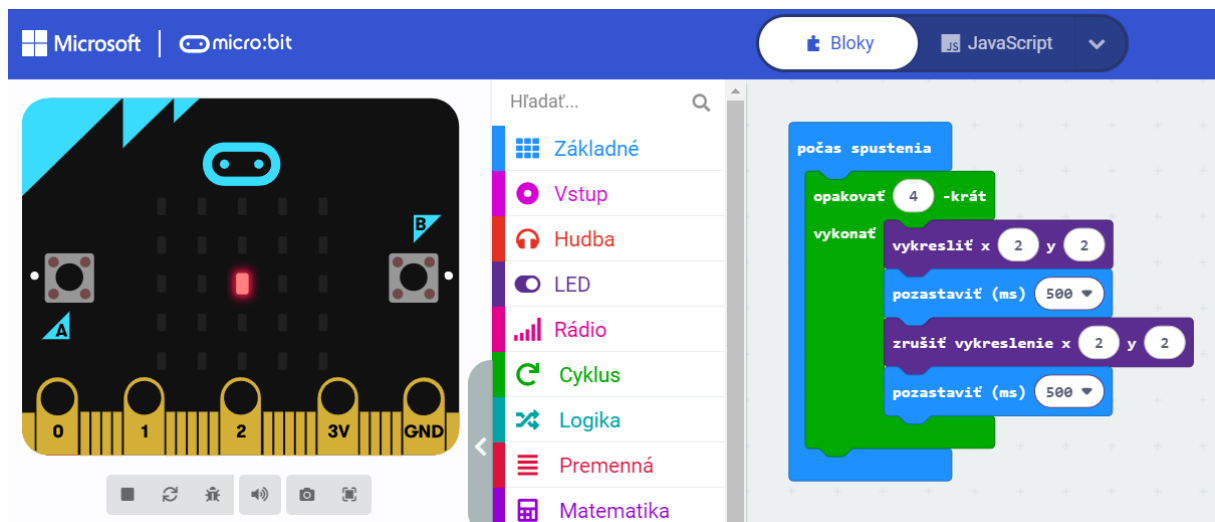
1 Príkaz "opakovať 4 krát"

Príkaz "opakovať 4 krát" slúži na vykonanie určitej akcie alebo súboru akcií štyrikrát za sebou. Tento príkaz umožňuje opakovať blok kódu stanovený počet krát, čo je užitočné na automatizáciu opakujúcich sa úloh.

Ako to funguje?

1. Nastavenie opakovania: Tento príkaz určuje, že nasledujúci blok kódu sa má vykonať štyrikrát.
2. Vykonanie kódu: Počet opakovaní je pevne stanovený na štyri, takže nasledujúci kód sa vykoná presne štyrikrát.

Príklad: Ak chceme rozsvietiť a zhasnúť LED na pozícii (2, 2) štyrikrát za sebou, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude LED na pozícii (2, 2) štyrikrát bliknúť s pauzou medzi zapnutím a vypnutím.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "opakovať 4 krát" vo svojom programe.
3. Vložte akcie, ktoré chcete vykonať opakovane, napríklad rozsvietenie a zhasnutie LED.
4. Spustíte svoj program a sledujte, ako micro:bit vykoná akcie presne štyrikrát!

Tento príkaz je veľmi užitočný na automatizáciu opakujúcich sa úloh a vytváranie interaktívnych projektov na micro:bite.

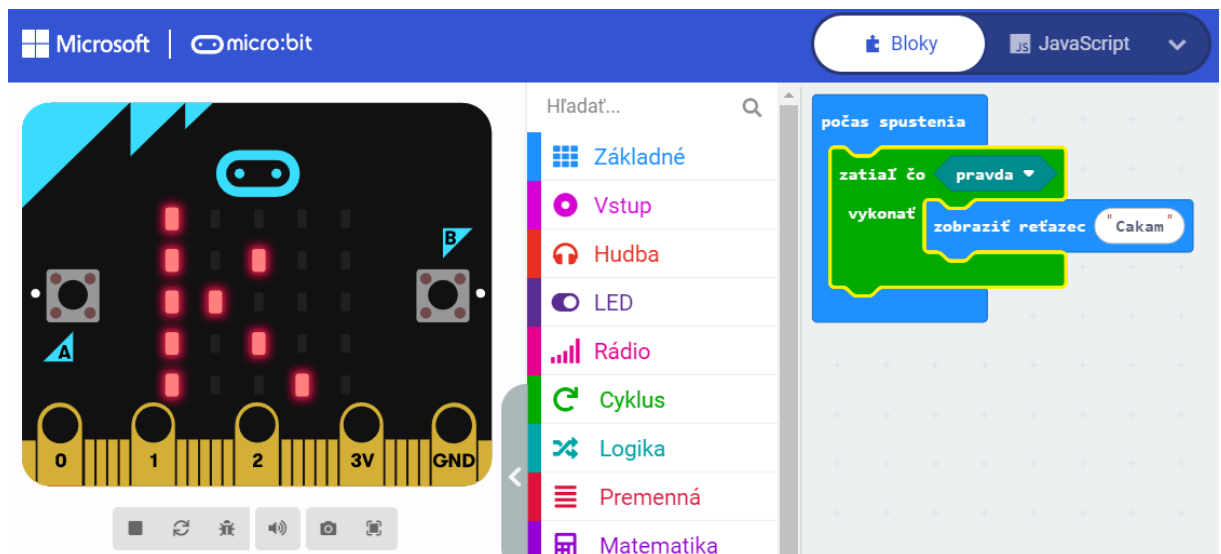
2 Príkaz "zatiaľ čo pravda"

Príkaz "zatiaľ čo pravda" slúži na vytvorenie cyklu, ktorý sa opakuje, pokiaľ je podmienka pravdivá. Tento príkaz umožňuje opakovane vykonávať blok kódu, kým sa nezmení stav podmienky na nepravdivý.

Ako to funguje?

1. Nastavenie podmienky: Tento príkaz obsahuje podmienku, ktorá je spočiatku pravdivá (true).
2. Opakovanie kódu: Blok kódu sa bude opakovane vykonávať, pokiaľ je podmienka pravdivá. Cyklus sa zastaví, keď sa podmienka zmení na nepravdivú (false).

Príklad: Ak chceme neustále zobrazíť text "Cakam", pokiaľ je podmienka pravdivá, použijeme nasledujúci blok kódu:



Po spustení tohto kódu sa blok kódu nebude vykonávať, pretože podmienka je vždy nepravdivá.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "zatiaľ čo pravda" vo svojom programe.
3. Do príkazu vložte akcie, ktoré chcete vykonať opakovane, kým je podmienka pravdivá.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie podľa podmienky!

Tento príkaz je veľmi užitočný na vytváranie opakujúcich sa úloh, ktoré sa vykonávajú, kým sa nesplní určitá podmienka.

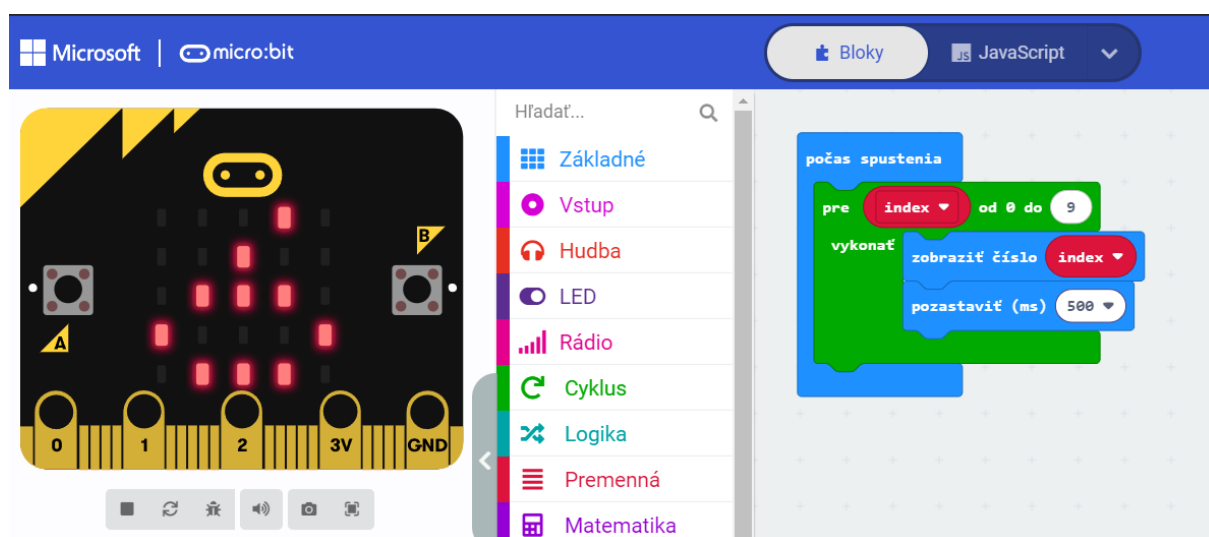
3 Príkaz "pre index od 0 do 9"

Príkaz "pre index od 0 do 9" slúži na vytvorenie cyklu, ktorý postupne vykonáva určitý blok kódu s indexom premennej od hodnoty 0 do hodnoty 9. Tento príkaz umožňuje opakovanie akcie konkrétny počet krát, pričom hodnota premennej sa postupne mení v každom opakovaní.

Ako to funguje?

1. Nastavenie rozsahu: Tento príkaz definuje rozsah indexu premennej od 0 do 9.
2. Vykonanie kódu: Blok kódu sa vykoná 10 krát, pričom hodnota premennej index sa v každom opakovaní postupne zvyšuje od 0 do 9.

Príklad: Ak chceme zobraziť hodnotu premennej index na LED matici, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 9, pričom každá hodnota bude zobrazená na 500 milisekúnd.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pre index od 0 do 9" vo svojom programe.
3. Vložte akcie, ktoré chcete vykonať opakovane, napríklad zobrazenie hodnoty premennej index.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie pre každý index od 0 do 9!

Tento príkaz je veľmi užitočný na automatizáciu opakujúcich sa úloh a na prácu s postupne sa meniacimi hodnotami premennej.

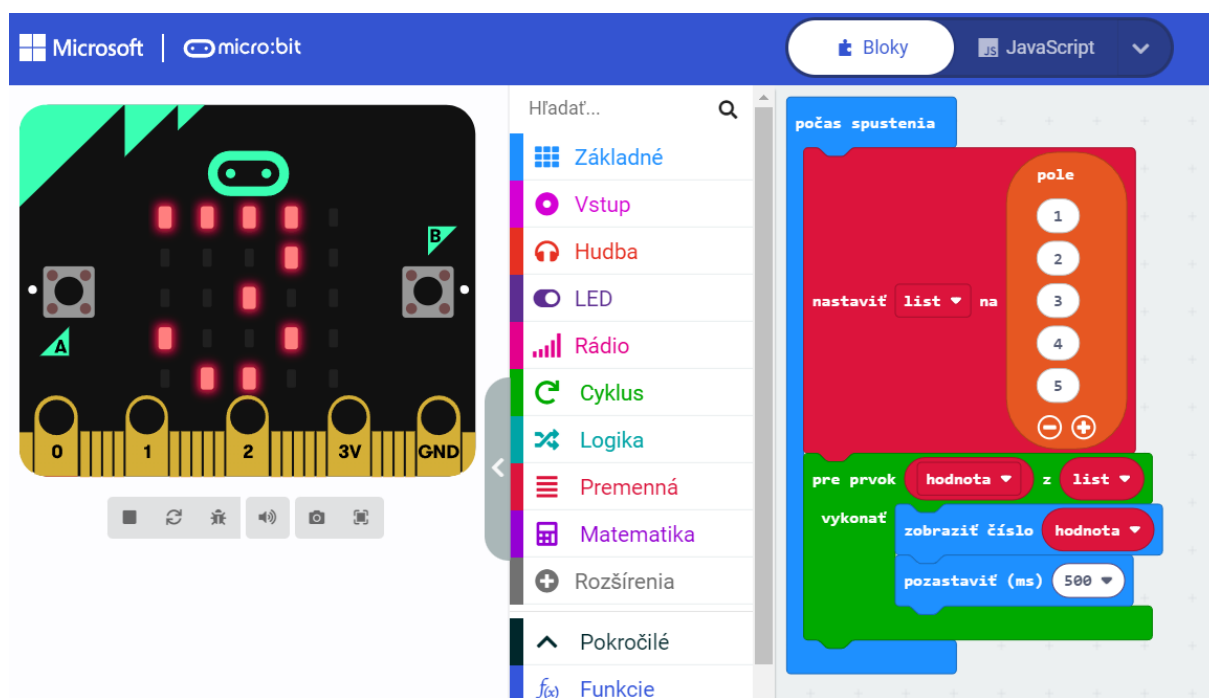
4 Príkaz "pre prvok hodnota z list"

Príkaz "pre prvok hodnota z list" slúži na iterovanie cez každý prvok v zozname a vykonanie určitej akcie pre každý prvok. Tento príkaz je užitočný na prácu s kolekciami údajov, kde chcete spracovať každý prvok v zozname postupne.

Ako to funguje?

1. Nastavenie iterácie: Tento príkaz definuje iteráciu cez každý prvok v zozname.
2. Vykonanie kódu: Blok kódu sa vykoná pre každý prvok v zozname, pričom premenná hodnota predstavuje aktuálny prvok.

Príklad: Ak máme zoznam čísel a chceme zobrazit' každé číslo na LED matici, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude micro:bit zobrazovať každé číslo zo zoznamu s pauzou medzi zobrazeniami.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pre prvok hodnota z list" vo svojom programe.
3. Definujte zoznam a vložte akcie, ktoré chcete vykonať pre každý prvok.
4. Spustíte svoj program a sledujte, ako micro:bit vykonáva akcie pre každý prvok zoznamu!

Tento príkaz je veľmi užitočný na prácu s kolekciami údajov a na automatizáciu opakujúcich sa úloh pre každý prvok zoznamu.

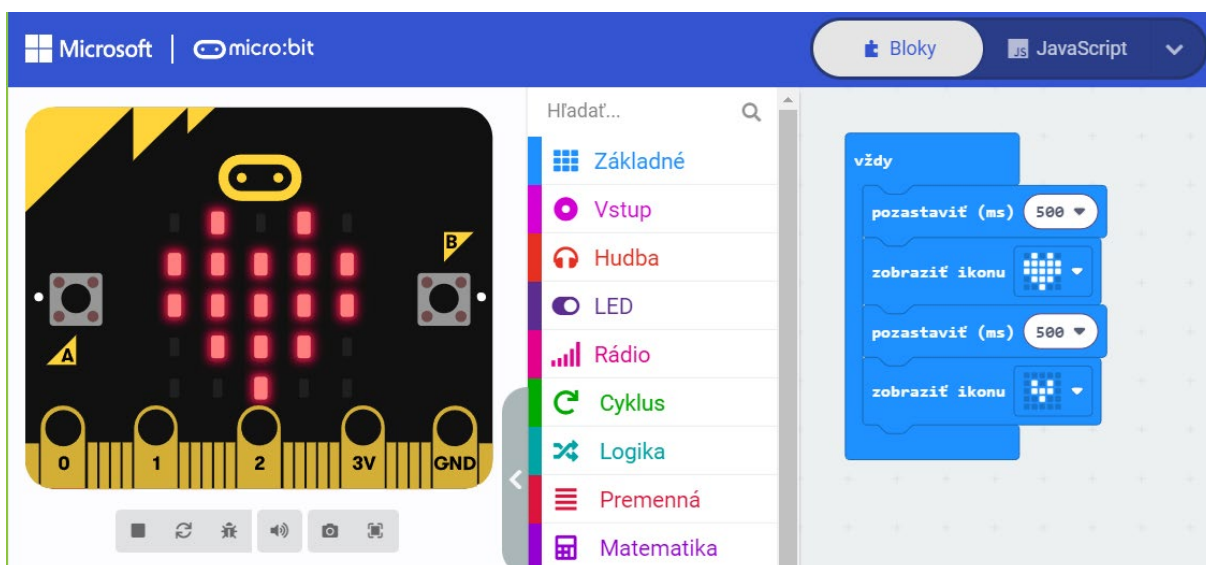
5 Príkaz "každých 500 ms"

Príkaz "každých 500 ms" slúži na opakované vykonanie určitej akcie každých 500 milisekúnd (ms). Tento príkaz umožňuje vytvoriť časovú slučku, ktorá vykonáva blok kódu v pravidelných intervaloch.

Ako to funguje?

1. Nastavenie intervalu: Tento príkaz nastaví interval na 500 ms, teda pol sekundy.
2. Vykonanie kódu: Blok kódu sa bude opakovane vykonávať každých 500 ms, kým sa program nezastaví alebo nezmení podmienky.

Príklad: Ak chceme každých 500 ms zmeniť zobrazenie ikony na LED matici, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude micro:bit každých 500 ms prepínať medzi ikonami srdca a malého srdca.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "každých 500 ms" vo svojom programe.
3. Do príkazu vložte akcie, ktoré chcete vykonať v pravidelných intervaloch.
4. Spustite svoj program a sledujte, ako micro:bit vykonáva akcie každých 500 ms!

Tento príkaz je veľmi užitočný na vytváranie opakujúcich sa úloh a interaktívnych projektov s pravidelnými časovými intervalmi.

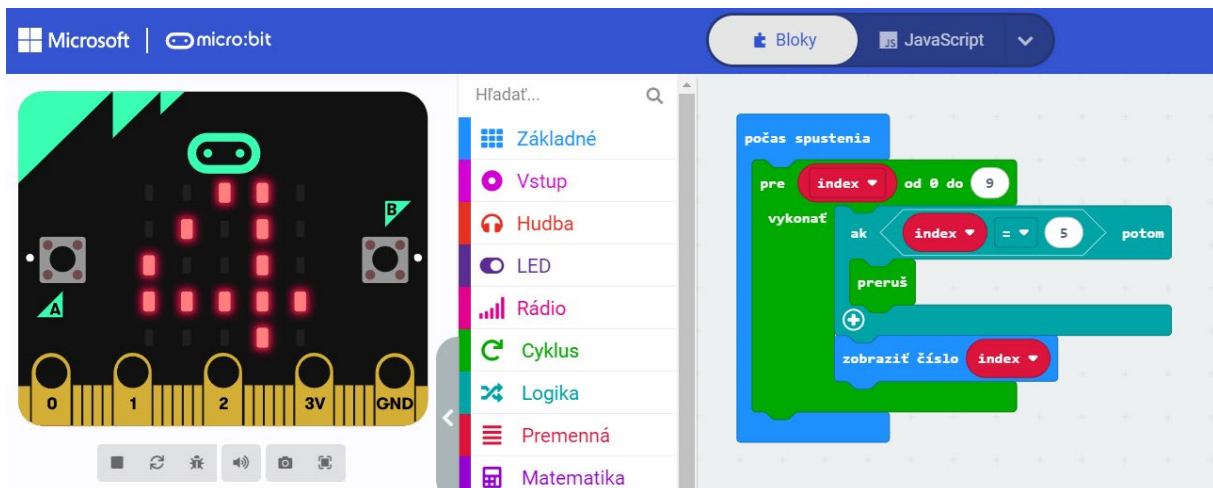
6 Príkaz "preruš"

Príkaz "preruš" slúži na okamžité ukončenie vykonávania aktuálneho cyklu alebo opakujúceho sa bloku kódu. Tento príkaz je užitočný, ak chcete predčasne ukončiť slučku na základe určitej podmienky.

Ako to funguje?

1. Detekcia príkazu: Keď príkaz "preruš" narazí vo vnútri cyklu, okamžite ukončí vykonávanie tohto cyklu.
2. Ukončenie slučky: Po ukončení cyklu sa vykonávanie programu presunie za tento cyklus.

Príklad: Ak chceme ukončiť slučku, keď sa hodnota index rovná 5, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 4 a ukončí slučku, keď sa hodnota index stane 5.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "preruš" vo svojom programe.
3. Nastavte podmienku, kedy chcete ukončiť vykonávanie slučky.
4. Spustíte svoj program a sledujte, ako micro:bit ukončí slučku na základe zadaných podmienok!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na predčasné ukončenie opakujúcich sa úloh.

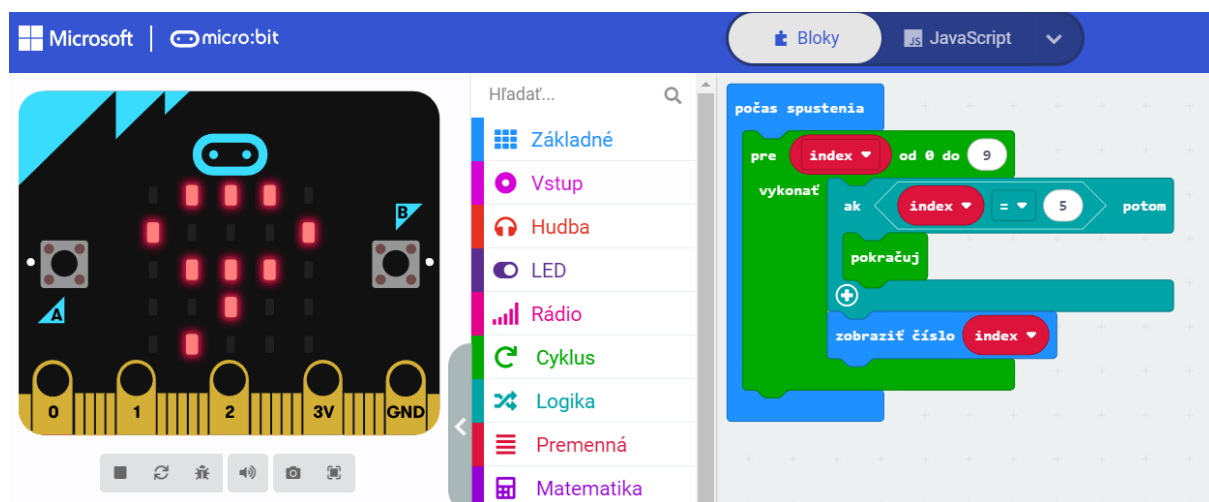
7 Príkaz "pokračuj"

Príkaz "pokračuj" slúži na preskočenie zvyšných príkazov v aktuálnom opakovaní cyklu a pokračovanie s ďalším opakovaním. Tento príkaz je užitočný, ak chcete niektoré iterácie cyklu vynechať na základe určitej podmienky.

Ako to funguje?

1. Detekcia príkazu: Keď príkaz "pokračuj" narazí vo vnútri cyklu, okamžite preskočí zvyšné príkazy v aktuálnom opakovaní.
2. Prechod na ďalšie opakovanie: Po preskočení sa vykonávanie programu presunie na ďalšie opakovanie cyklu.

Príklad: Ak chceme preskočiť zobrazenie čísla 5 a pokračovať s ďalšími číslami, použijeme nasledujúci blok kódu:



Po spustení tohto kódu bude micro:bit zobrazovať hodnoty od 0 do 9, okrem čísla 5, ktoré bude preskočené.

Vyskúšajte si to!

1. Otvorte MakeCode editor na stránke micro:bit.
2. Použite príkaz "pokračuj" vo svojom programe.
3. Nastavte podmienku, kedy chcete preskočiť vykonávanie zvyšných príkazov v cykle.
4. Spustíte svoj program a sledujte, ako micro:bit preskočí zvyšné príkazy a pokračuje s ďalším opakovaním!

Tento príkaz je veľmi užitočný na kontrolu toku vykonávania programu a na vynechanie určitých iterácií cyklu.