

A leírás a burcsi.hu weboldalról származik az alábbi linkről:

<http://www.burcsi.hu/-info/Microbit>

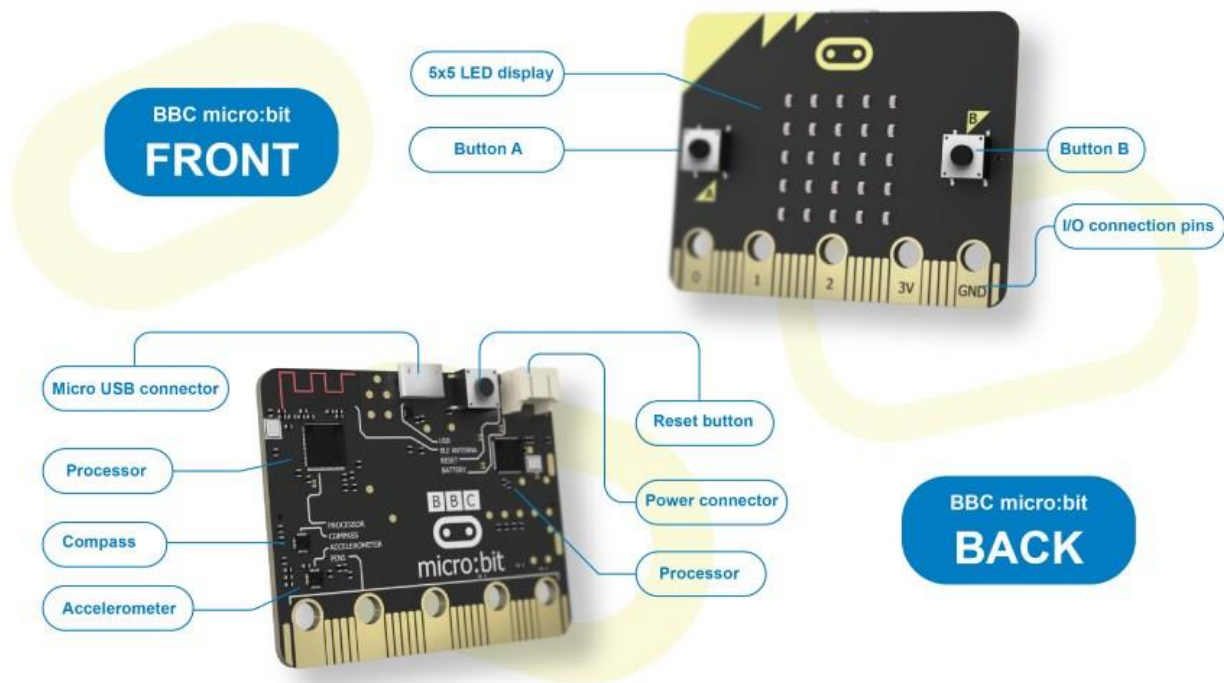
Hardver

Forrás: <https://malnapc.hu/2017/03/bbc-microbit/>

A mini számítógép lelke egy ARM Cortex-M0 processzor 256kB flash és 16kB statikus memóriával valamint egy 2.4GHz-es Bluetooth Low Energy vezeték nélküli hálózati egységgel megtámogatva.

A panel elején kapott helyet:

- egy 25 elemből álló piros LED mátrix,
- 2 darab programozható nyomógomb és a
- 25 pólusú csatlakozósáv. Ez utóbbi 5 nagyméretű csatlakozást kínál, így krokodilcsipesszel vagy 4mm-es banándugóval támadhatunk neki. Föld, +3V és három darab digitális/analog ki vagy bemenet. Ezt a kezdők részére, míg a maradék keskenyebb csatlakozókat, melyek 6 soros I/O-t, SPI és I2C interfészt is tartalmaznak, haladóknak szánták, komolyabb projektekhez.



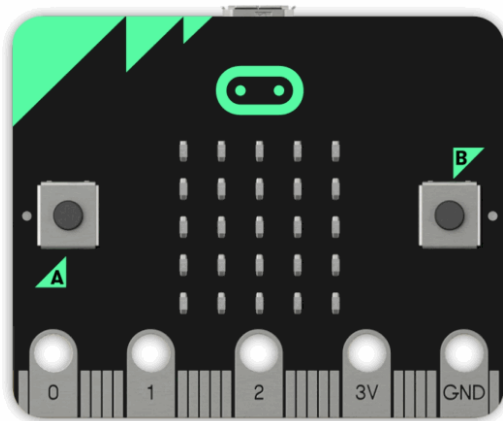
A hátoldalon található

- a már említett CPU és
- bluetooth.
- Mellettük egy gyorsulásmérővel kiegészített iránytű,
- reset gomb,
- táp és USB OTG csatlakozó kapott még helyet.

A tápellátás történhet a programozáshoz is használható USB csatlakozón, vagy külső telep használatakor a "battery connector"-on keresztül. Ezzel a micro:bit máris hordozhatóvá válik.

1. lépés: Csatlakoztatása

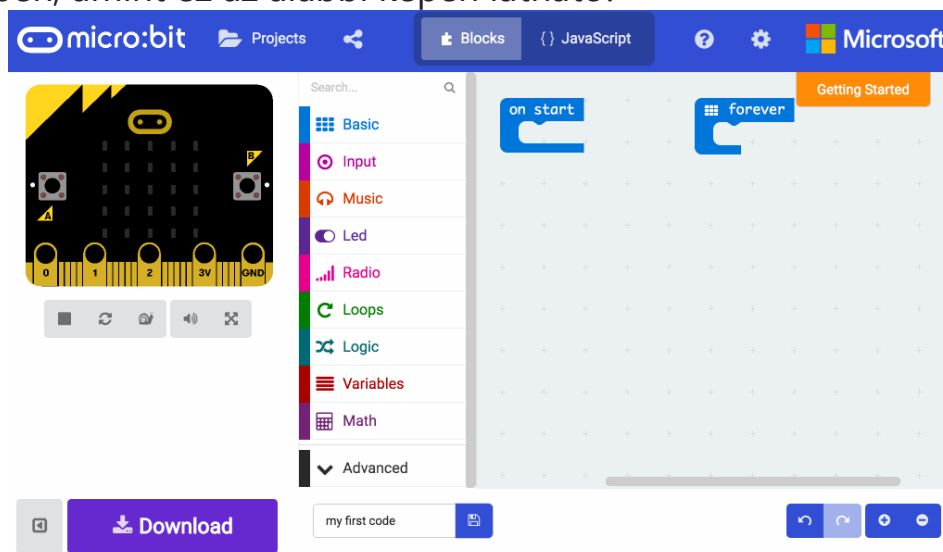
Csatlakoztasd a micro:bit-et a számítógépedhez mikro USB-kábelen keresztül. Mac, PC, Chromebook és Linux rendszerek mind támogatottak. Jön egy szórakoztató alkalmazás, próbáld ki!



A micro:bit megjelenik a számítógépen, mint egy meghajtó, neve: "MICROBIT". Ez azonban nem egy normál USB lemez!

2. lépés: Programozása

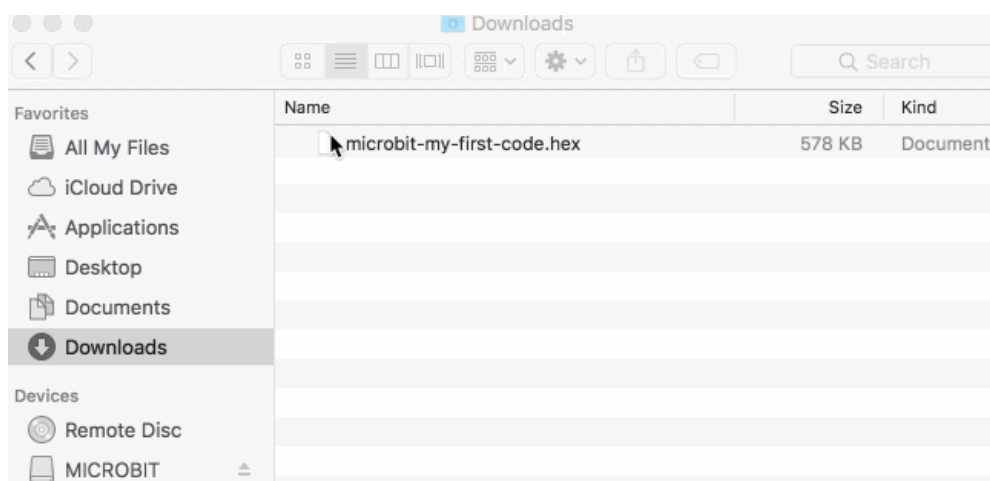
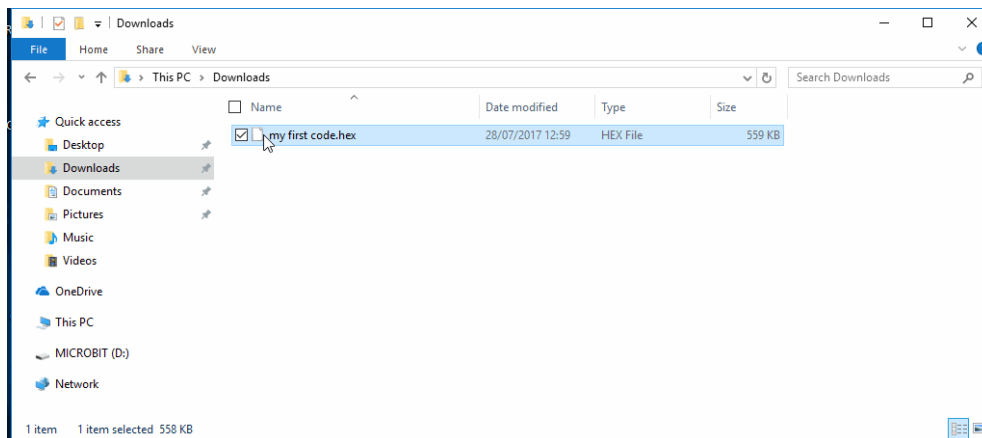
Az egyik [fantasztikus szerkesztőnk](#) segítségével írd be az első mikro-bit kódodat! Húzz és dobj néhány blokkot, és próbáld meg a programot a Szimulátoron a Javascript Blokkszerkesztőben, amint ez az alábbi képen látható!



[Kattints ide, hogy kipróbálhasd magad is ezt a példát!](#)

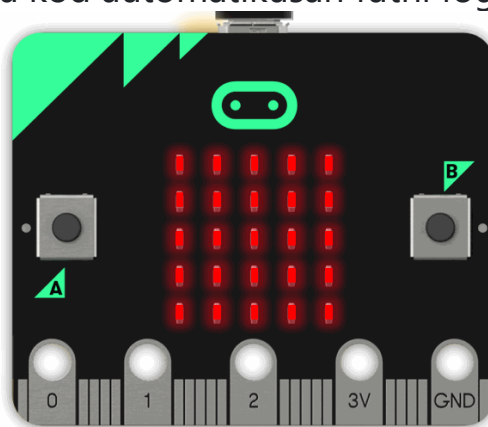
3. lépés: Letöltése

Kattints a szerkesztőben lévő "Letöltés" gombra. Ez letölt egy "hex" fájlt, amely egy kompakt formátumú program, amelyet a micro:bit képes olvasni. Miután a hex fájlt letöltötted, másold át a micro:bit-re! Ez olyan, mint egy fájl USB-meghajtóra történő másolása. (Windows rendszerben jobb egérgombbal kattintva válasszad a "Küldés → MICROBIT" elemet.)



4. lépés: Lejátszása

A mikro:bit működése szünetel, és a sárga LED villog a hátulján, amíg a programkód átmege. Miután befejeződött, a kód automatikusan futni fog!



??? micro: bites sárga LED villog és új kód indítás A MICROBIT meghajtó automatikusan kiadja és visszatér minden egyes programozáskor, de a hex file el fog tűnni.

A mikro:bit csak hex fájlokat fogadhat, és másikat nem tárol!

Milyen jó dolgokat fogsz létrehozni? A mikro:bit válaszolhat a gombokra, a fényre, a mozgásra és a hőmérsékletre. Még a vezeték nélküli üzeneteket is küldhet más mikro:bit-ekre a "Rádió" funkció használatával.

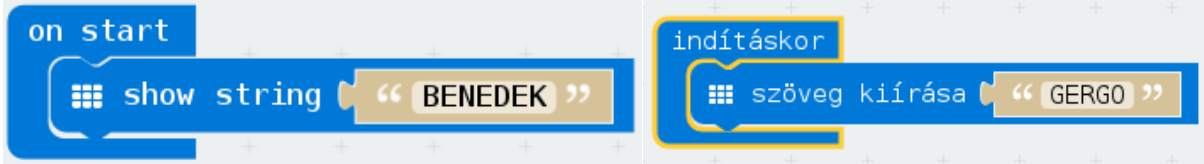
Nézd meg a [hardver](#) oldalát, hogy ötleteket nyerj!

5. lépés: Megtanulása

Ez az oldal megmutatja, hogyan kell használni a micro:bit-et, de a JavaScript Blokkokkal is használhatod a Python-t és a szövegalapú JavaScript-et a mikro:bit programozásához. A [kódhoz](#) lapozz a különböző nyelvek megtekintéséhez, vagy nézzd meg az [ötleteket tartalmazó oldalt](#) néhány olyan dologgal kapcsolatban, amelyeket szeretnél kipróbálni.

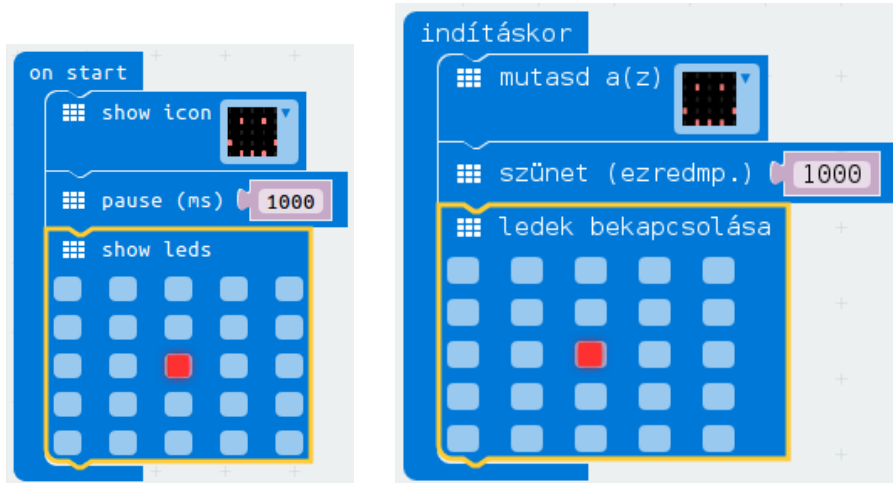
Gyakorló feladatok

A1. Futófény a keresztneveddel



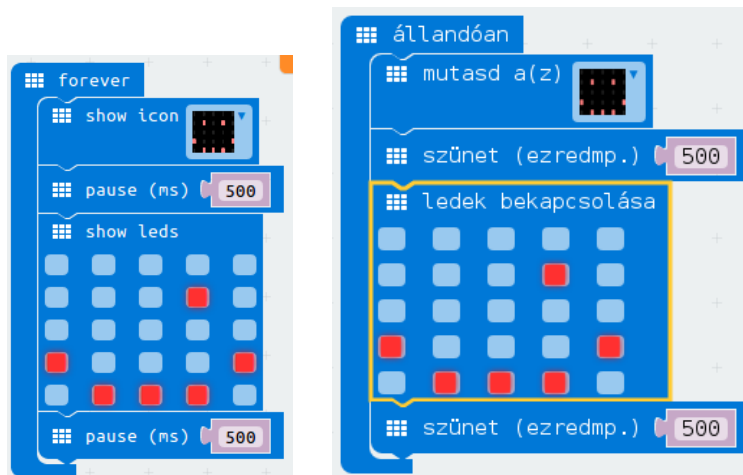
Indításkor írja ki futófényként a keresztnevedet! (Ékezetes betűt NE rakj bele, mert úgysem jeleníti meg!)

A2. Smiley törléssel



Indításkor mutasson boldog szmájlit, várjon 1 másodpercet, majd törölje le a képernyőt és világítson középen egy LED!

A3. Smiley kacsintással



Örökké: mutasson boldog szmájlit, várjon fél másodpercet, majd törölje le a képernyőt és kacsintson egyet, s megint várjon!

A4. Páros számok kiírása

```
forever
  show number 0
  pause (ms) 500
  show number 2
  pause (ms) 500
  show number 4
  pause (ms) 500
  show number 6
  pause (ms) 500
```

```
állandóan
  szám kiírása 0
  szünet (ezredmp.) 500
  szám kiírása 2
  szünet (ezredmp.) 500
  szám kiírása 4
  szünet (ezredmp.) 500
  szám kiírása 6
  szünet (ezredmp.) 500
```

Örökké: számoljon el 0-tól 6-ig párosával!

A5. Villogó led-ek (csak kettő)

```
forever
  plot x 0 y 0
  pause (ms) 200
  unplot x 0 y 0
  pause (ms) 200
  plot x 1 y 0
  pause (ms) 200
  unplot x 1 y 0
  pause (ms) 200
```

```
állandóan
  felkapcsol x 0 y 0
  szünet (ezredmp.) 200
  lekapcsol x 0 y 0
  szünet (ezredmp.) 200
  felkapcsol x 1 y 0
  szünet (ezredmp.) 200
  lekapcsol x 1 y 0
  szünet (ezredmp.) 200
```

Örökké: az első sorban az első két LED-et villogtassa felváltva!

A6. Fő égtájak nyilakkal

```
forever loop
  show arrow North
  pause (ms) 500
  show arrow East
  pause (ms) 500
  show arrow South
  pause (ms) 500
  show arrow West
  pause (ms) 500
```

```
állandóan loop
  mutasd a nyilat Észak
  szünet (ezredmp.) 100
  mutasd a nyilat Kelet
  szünet (ezredmp.) 100
  mutasd a nyilat Dél
  szünet (ezredmp.) 100
  mutasd a nyilat Nyugat
  szünet (ezredmp.) 100
```

Örökké: a fő égtájaknak megfelelő nyilak forogjon körbe É,K,D,Ny sorrendben!

A7. Furcsa dobókocka

```
on start
  show string "KOCKA"

on shake
  show number pick random 0 to 5
```

```
indításakor
  szöveg kiírása " KOCKA "

ha rázás
  szám kiírása véletlen szám 0 és 4
```

Indításkor írja ki: "KOCKA"!

Rázáskor: 0 és 5 közt találjon ki egy számot, s írja ki!

B1. Sorsolás és törlés az A, B gombokkal

```
on button A pressed
  show number pick random 0 to 9

on button B pressed
  show leds
```

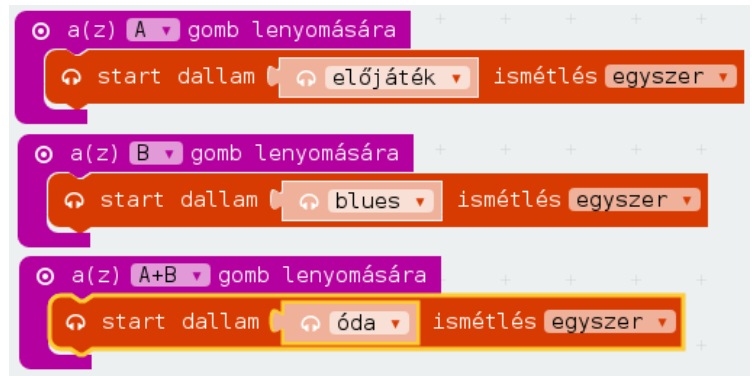
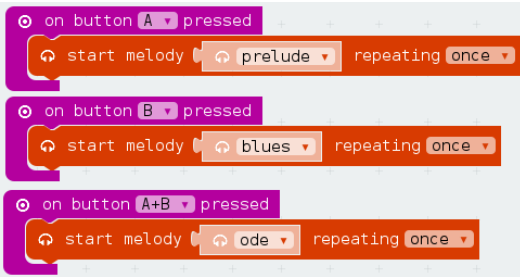
```
a(z) A gomb lenyomására
  szám kiírása véletlen szám 0 és 9

a(z) B gomb lenyomására
  ledek bekapcsolása
```

A gomb: írjon ki egy 0 és 9 közti véletlenszámot!

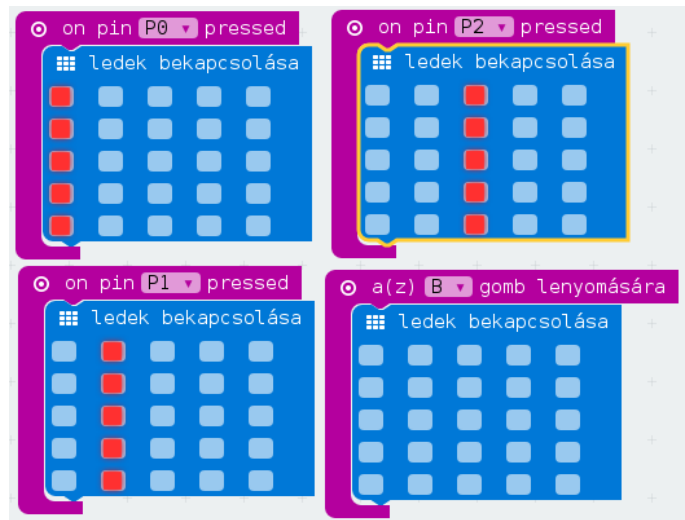
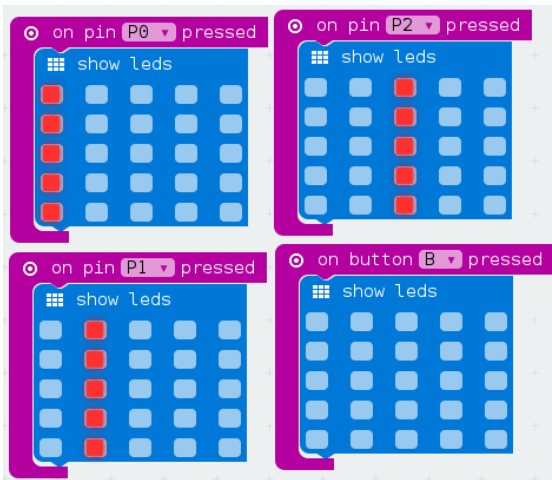
B gomb: törölje a ledet!

B2. Dallamok az A, B, A+B gombokon



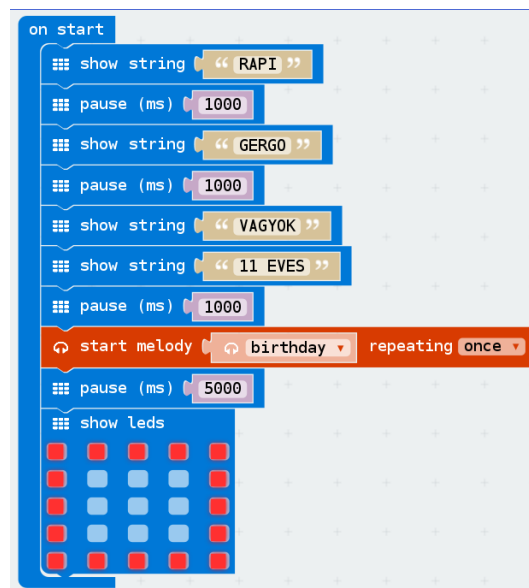
Mindhárom gombra programozzuk egy-egy dalocskát!

B3. Csatlakozólábak (pin0, pin1, pin2)



A pin0, pin1, pin2-re programozzuk azt, hogy gyújtsa ki az első, a 2. ill. a 3. ledoszlopot! A B gomb lenyomásakor töröljük a ledeket!

A1. Rapi Gergő szülinapjára



Írd ki szavanként a következő szöveget: "RAPI GERGO VAGYOK 11 EVES" !

Tégy utánuk egy-egy szünetet! Játszd el a "birthday" nevű dallamot! (A 2. nem tud dallamokat!) Gyűjtsd ki a szélső ledeket!

A2. Hatos lottó

```
show string "HATOS LOTTO"  
repeat 6 times  
do  
  play tone (Hz) C5 for (ms) 1 beat (ms)  
  pause (ms) 1000  
  show number pick random 0 to 44 + 1  
  pause (ms) 1000  
  clear screen
```

```
on start  
  show string "HATOS LOTTO"  
  repeat 6 times  
  do  
    play tone High C for 1 beat  
    pause (ms) 1000  
    show number pick random 0 to 44 + 1  
    pause (ms) 1000  
    clear screen
```

Írd ki a "HATOS LOTTO" szöveget ([szelvény](#))! (Ékezetes betűink nem jelennek meg!)

Játszd le az alábbiakat 6-szor!

- jelezz egy felső C hanggal!
- írd ki egy véletlenszámot 1-45 közt!
- tegyél elegendően hosszú várakozásokat a megfelelő helyekre!

Készítsd el a programot ötös lottóra is ([szelvény](#))! (Milyen sorsolási hiba van a programokban?)

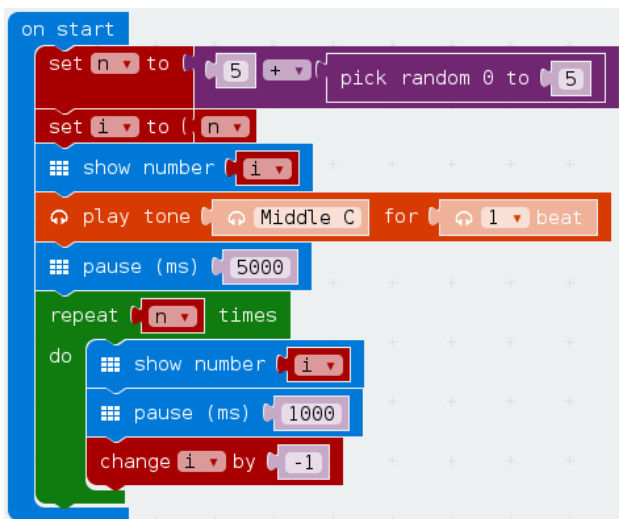
A3. Boci-boci tarka ([video](#))

```
set tempo to (bpm) 120  
play tone (Hz) C for (ms) 1/2 beat (ms)  
play tone (Hz) E for (ms) 1/2 beat (ms)  
play tone (Hz) C for (ms) 1/2 beat (ms)  
play tone (Hz) E for (ms) 1/2 beat (ms)  
play tone (Hz) G for (ms) 1 beat (ms)  
play tone (Hz) G for (ms) 1 beat (ms)
```

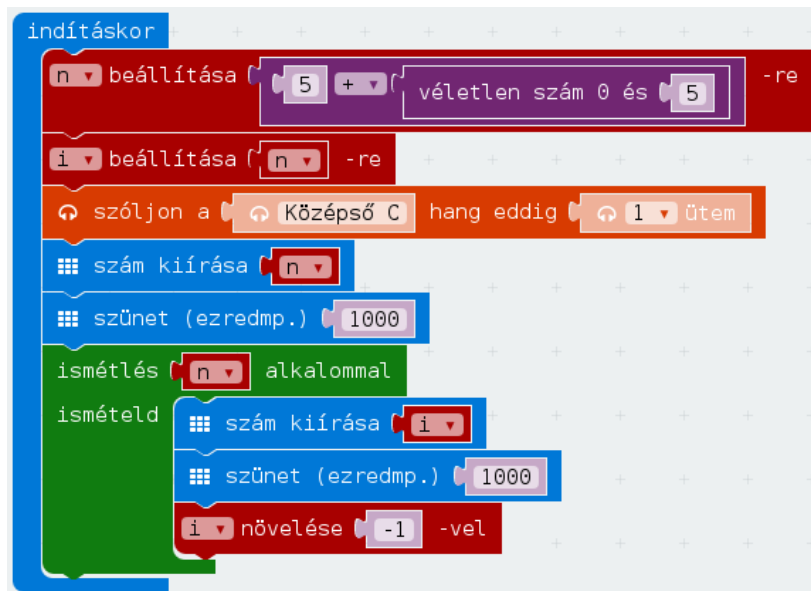
```
on button A pressed  
  play tone Middle C for 1 beat  
  play tone Middle E for 1 beat  
  play tone Middle C for 1 beat  
  play tone Middle E for 1 beat  
  play tone Middle G for 2 beat  
  play tone Middle G for 2 beat
```

"Kottázd le" a "Boci-boci tarka" első néhány hangját! ([Kotta](#)) [Oktató videó](#)

A4. Számsor fordítva kiírva



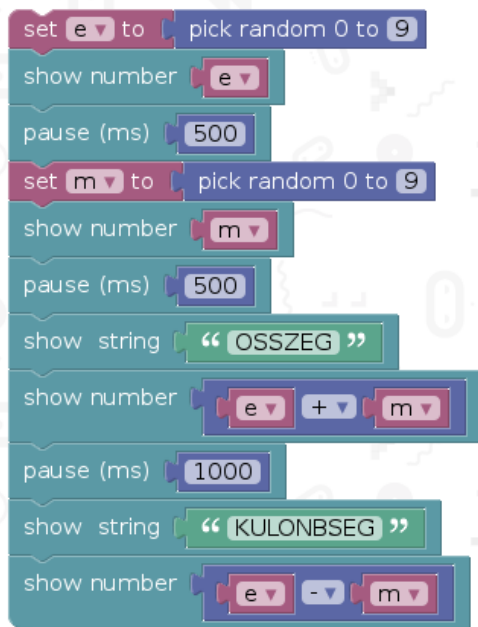
```
on start
  set n to (5 + pick random 0 to 5)
  set i to (n)
  show number i
  play tone Middle C for 1 beat
  pause (ms) 5000
  repeat n times
    do
      show number i
      pause (ms) 1000
      change i by -1
```



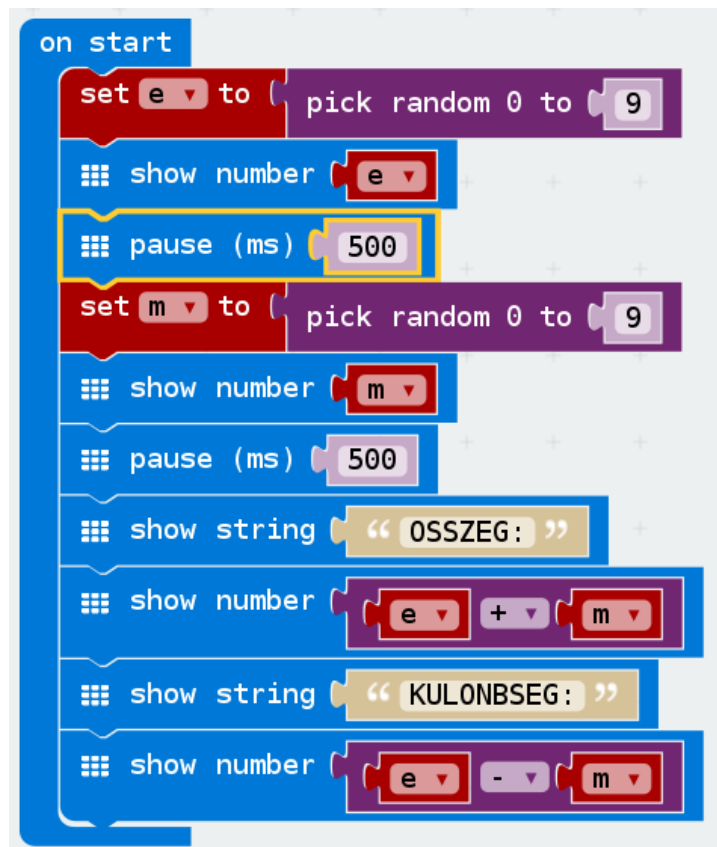
```
indításkor
  n beállítása (5 + véletlen szám 0 és 5) -re
  i beállítása (n) -re
  szóljon a Középső C hang eddig 1 ütem
  szám kiírása n
  szünet (ezredmp.) 1000
  ismétlés n alkalommal
  ismételd
    szám kiírása i
    szünet (ezredmp.) 1000
    i növelése -1 -vel
```

Dobj egy "n" véletlenszámot 5 és 10 közt!
Írd ki, de előtte adj ki valamilyen hangjelzést!
Írd ki az n, ..., 3, 2, 1 számsort!
(Ahol kell, használj szünetet!)

B1. Matematika



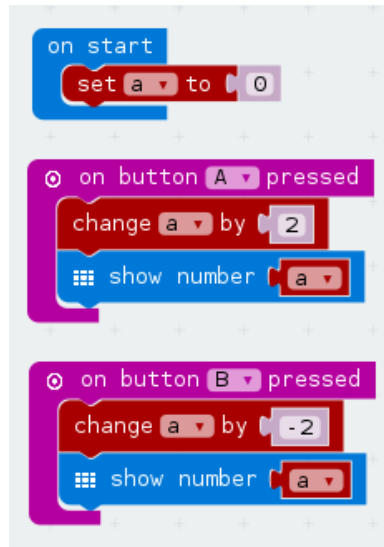
```
set e to pick random 0 to 9
show number e
pause (ms) 500
set m to pick random 0 to 9
show number m
pause (ms) 500
show string "OSSZEG:"
show number (e + m)
pause (ms) 1000
show string "KULONBSEG:"
show number (e - m)
```



```
on start
  set e to pick random 0 to 9
  show number e
  pause (ms) 500
  set m to pick random 0 to 9
  show number m
  pause (ms) 500
  show string "OSSZEG:"
  show number (e + m)
  show string "KULONBSEG:"
  show number (e - m)
```

Dobj egy "e" és egy "m" egyjegyű véletlenszámot (0 is lehet mindegyik)!
Írd ki a két szám összegét és különbségét!
Tegyél elegendően hosszú várakozásokat a megfelelő helyekre!

B2. Páros számok



Hozd létre az "a" változót, és állíts az értékét 0-ra!

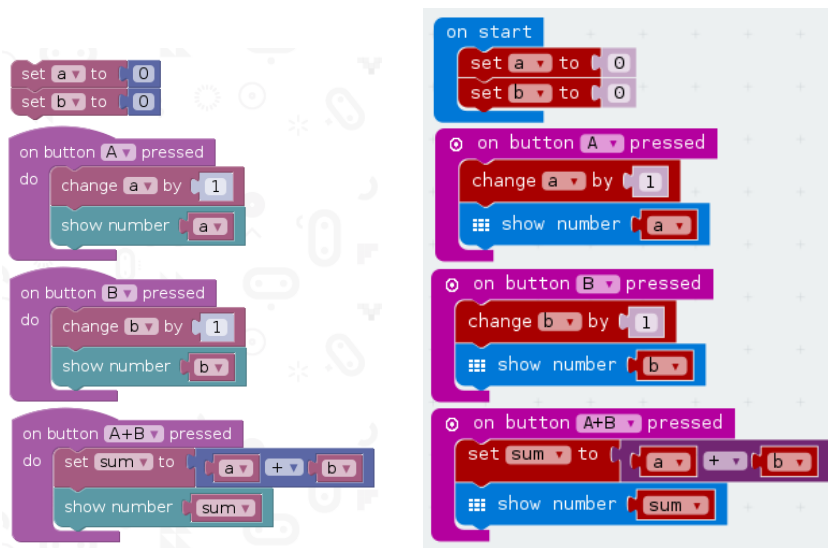
Az "A" gomb lenyomására a következőket programozd:

- növelje 2-vel az "a" változót
- legyen ez látható

A "B" gomb lenyomására pedig a következőket:

- csökkentse 2-vel az "a" változót
- legyen ez látható

B3. Kalkulátor



Állíts be egy "a" és egy "b" változót 0-ra! Hozd létre a "sum" változót is (ezek összegét tárolja majd)!

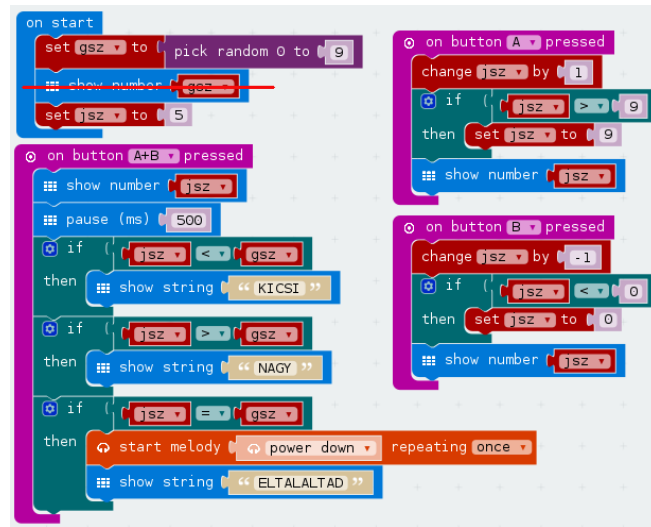
Az "A" gomb illetve "B" gomb lenyomására a következőket programozd:

- növelje 1-gyel az "a" illetve a "b" változót
- legyen ez látható

Az "A+B" gomb lenyomására pedig a következőket:

- a "sum" legyen "a" és "b" összege
- legyen ez látható is [Oktató videó](#)

B4. Szám barchoba



Induláskor dobjon a gép egy véletlenszámot ($0 \leq \text{gsz} \leq 9$)! (Amíg a progit teszteljük, tegyük ezt láthatóvá!)

A játékos számát (tippjét) állítsuk 5-re! ($\text{jsz} = 5$)

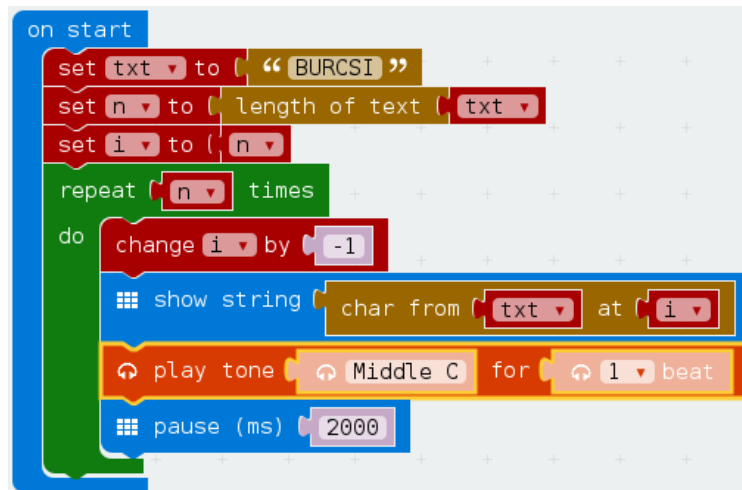
Az "A" gomb lenyomására a "jsz" nőjön 1-gyel, de ha meghaladná a 9-et, állítsuk 9-re!

A "B" gomb lenyomására a "jsz" csökkenjen 1-gyel, de ha 0 alá menne, állítsuk 0-ra!

Az "A+B" gomb lenyomására pedig:

- ha $\text{jsz} < \text{gsz}$, írja ki, hogy "KICSI"
- ha $\text{jsz} > \text{gsz}$, írja ki, hogy "NAGY"
- ha $\text{jsz} = \text{gsz}$, írja ki, hogy "ELTALALTAD", és játsszon le egy dallamot!

B5. Szöveg betűi fordítva kiírva



Tégy a "txt" változóba egy szöveget!

Írd ki a betűit hátulról visszafelé, a kiírások előtt adj ki valamilyen hangjelzést!

(Ahol kell, használj szünetet!)

C1. Előjeles számok összege



Hozd létre három változót: "a", "b", "sum" (a+b összeget tárolja majd)!

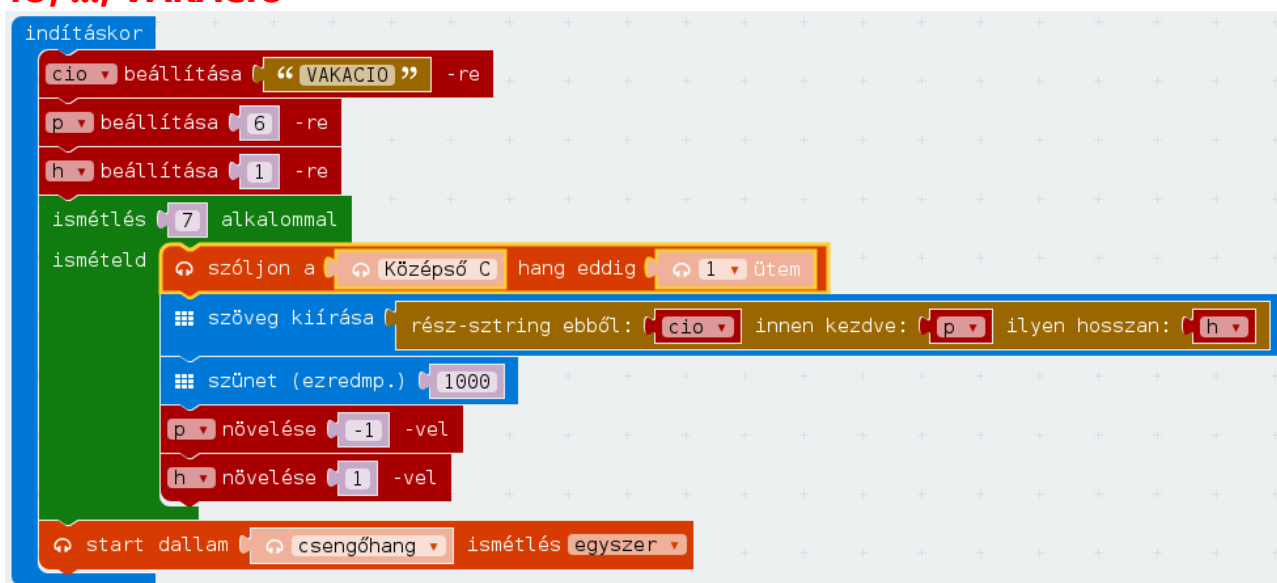
Az "A" gomb illetve "B" gomb lenyomására a következőket programozd:

- legyen az "a" illetve a "b" -20 és +20 közti véletlenszám
- legyen ez látható

Az "A+B" gomb lenyomására pedig a következőket:

- a "sum" legyen "a" és "b" összege
- legyen ez is látható

C2. Ó, IÓ, ..., VAKÁCIÓ



Írasd ki a "VAKACIO" szót a (szünetek előtti) szokott módon, azaz hátulról egyre több betűt kiírva! A kiírások közt tarts szünetet!

C3. Angol ABC

The Scratch code is organized into two main sections. The first section, labeled 'indításkor' (when started), prints the uppercase alphabet 'NAGYBETUK'. It starts with a 'szöveg kiírása' (print text) block containing 'NAGYBETUK', followed by an 'i beállítása' (set i to) block with the value 64. A green loop block 'ismétlés' (repeat) with 26 iterations follows. Inside the loop, 'ismételd' (do) contains an 'i növelése' (increase i by) block with 1, a 'szöveg kiírása' (print text) block with 'szöveg karakterkódból: i', and a 'szünet (ezredmp.)' (wait) block with 1000. After the loop, there is a 'szünet (ezredmp.)' (wait) block with 5000. The second section prints the lowercase alphabet 'kisbetuk'. It starts with a 'start dallam' (start sound) block for 'csengőhang' (chime) with 'ismétlés egyszer' (repeat once), followed by a 'szöveg kiírása' (print text) block with 'kisbetuk', an 'i beállítása' (set i to) block with 96, a green loop block 'ismétlés' (repeat) with 26 iterations. Inside the loop, 'ismételd' (do) contains an 'i növelése' (increase i by) block with 1, a 'szöveg kiírása' (print text) block with 'szöveg karakterkódból: i', and a 'szünet (ezredmp.)' (wait) block with 1000. After the loop, there is a 'szünet (ezredmp.)' (wait) block with 5000, and finally a 'start dallam' (start sound) block for 'csengőhang' (chime) with 'ismétlés egyszer' (repeat once).

Írd ki az angol ABC nagybetűit libasorban!
Írd ki az angol ABC kisbetűit libasorban!
A kiírások közt tarts szünetet!

C4. * Háromjegyű szám fordítva kiírva

The Scratch code is triggered by two button clicks: 'a(z) A gomb lenyomására' and 'a(z) B gomb lenyomására'. The first section, triggered by button A, sets a variable 'hj' to a random number between 100 and 899, plays a 'csengőhang' (chime) sound once, and prints the value of 'hj'. The second section, triggered by button B, performs a digit-by-digit reversal of the number in 'hj'. It uses a series of 'beállítás' (set) blocks: 'e' is set to 'hj' mod 10, 'hj' is set to 'hj' minus 'e', 't' is set to 'hj' divided by 10, 'hj' is set to 'hj' minus 't', 's' is set to 'hj' divided by 10, and finally 'hj' is set to '100 * e + 10 * t + s'. The final value of 'hj' is then printed.

Az "A" lenyomására dobj egy "hj", háromjegyű véletlenszámot 100 és 999 közt, majd írd ki!
A "B" lenyomásával pedig írd ki fordítva, azaz: egyes, tízes, száz! [Algoritmus](#)

C5. * Kétjegyű szám prím-e

The image shows a Scratch script for checking if a two-digit number is prime. The script is organized into several event-driven blocks:

- Indításkor (When the program starts):** Sets the variable 'kj' to 50 and 'sz' to 1.
- a(z) A gomb lenyomására (When key 'A' is pressed):** Increases 'kj' by $10 \times \text{sz}$ and outputs 'kj'.
- a(z) B gomb lenyomására (When key 'B' is pressed):** Increases 'kj' by 'sz' and outputs 'kj'.
- on pin P0 pressed:** Sets 'sz' to $-1 \times \text{sz}$ and outputs 'sz'.
- a(z) A+B gomb lenyomására (When key 'A+B' is pressed):** Sets 'db' to 0 and 'i' to 1. It enters a loop where 'i' is incremented by 1 until 'db' reaches 2. Inside the loop, it checks if 'kj' is divisible by 'i' (remainder = 0). If yes, 'db' is incremented by 1. After the loop, it outputs 'PRIM' if 'db' is 2, otherwise 'NEM PRIM'.

Állítsd be a kétjegyű számot (kj) "50"-re, a szorzót (sz) pedig "+1"-re!
Az "A" lenyomásával "kj"-t növel/csökkentsd " $10 \times \text{sz}$ "-szel, majd írd ki!
A "B" lenyomásával "kj"-t növel/csökkentsd " $1 \times \text{sz}$ "-szel, majd írd ki!
A "pin0"-ra kattintva változzon a szorzó "-1"-szeresére, és írd ki!
Az "A+B" gomb lenyomására írd ki, hogy "PRIM" vagy "NEM PRIM"!

[Algoritmus](#)

VÉGE