Beginner's Guide

How programs work in mBlock

Start with the blocks to learn about how mBlock works!



```
move 10 steps
turn ↻ 15 degrees
turn ↺ 15 degrees
go to mouse-pointer ▼
go to x: -177 y: 8
glide 1 secs to mouse-pointer ▼
glide 1 secs to x: -177 y: 8
point in direction 90
point towards mouse-pointer ▼
change x by 10
set x to -177
change y by 10
set y to 8
```

```
when ⚑ clicked
set size to 100 %
set Score ▼ to 0
show variable Score ▼
set Health Point ▼ to 100
show variable Health Point ▼
go to x: -177 y: 8
repeat until timer < 0
  set size to Health Point %
  if key up arrow ▼ pressed? and y position < 170 then
    change y by 5
  if key down arrow ▼ pressed? and y position > -167 then
    change y by -5
```

Panda | Block Number:24

Start - Connect your first block to the start state. If nothing is attached, nothing happens when you run your program.

Loops - You can put blocks inside a control block to make a loop! In this case, the blocks inside repeat until there is an obstacle in front.

Block Area - select your category and then drag out the blocks you want to use in your code.

Block Shapes

The blocks are designed in different shapes, each indicating one data type. The blocks can be connected to each other vertically like a jigsaw puzzle. Programs (or scripts) in mBlock 5 are compiled by connecting different types of blocks vertically like a jigsaw puzzle. mBlock 5 provides blocks of five shapes, as described in the following:
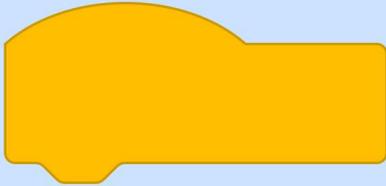
Hat block is used to start a script and is always placed on the top of other blocks. Each hat block uses its own way to activate a script, and therefore scripts can be executed separately at different time points.

Stack block is rectangular block that can fit on or under other blocks, with a notch at the top and a bump at the bottom. Stack blocks are used to execute the main commands, and therefore are the majority among all the blocks.

Reporter block contains a value that can be a numerical value or character string. A reporter block can be used in any script that requires data but can't be used independently. A reporter block can be fit into another block as long as the slot fits.

Boolean block contains a condition, which can be either "true" or "false". A boolean block must be put into the hexagonal slot of another block. It can't be used independently.

Cap block ends a script or program. It can only be put under all other blocks.

Add & Connect your Devices

Devices | Sprites | Background

CyberPi

⇆ Switch to direct connection

Connection

🔗 Connect

Mode Switch ⓘ

Upload | Live

+ Add

Connect your device

How to use device?

Add your Devices - you can click this button to add your devices and run your programs on them.

Connect your Devices - you have two ways to connect the device:

1. Click "Connect" button, but you have to download the mLink app.

2. Click "Switch to direct connection" button, and connect your device via Bluetooth or serial.
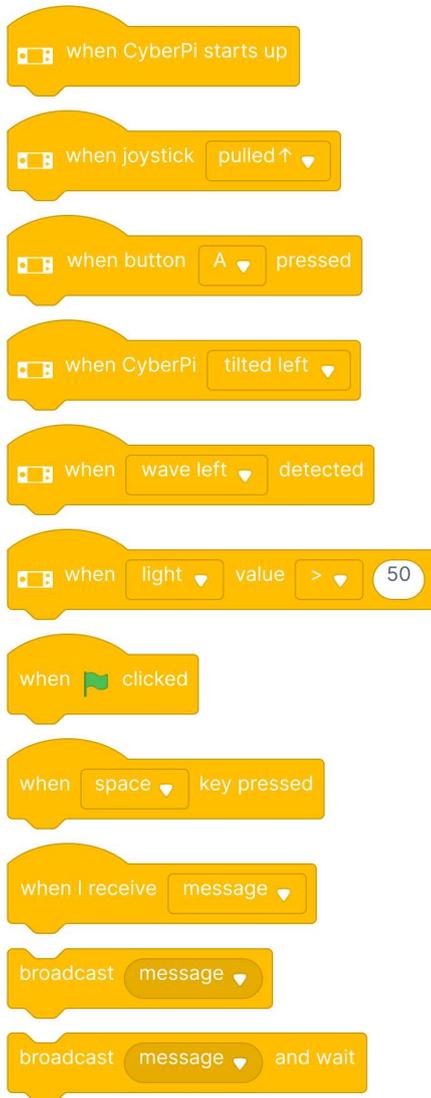
Bluetooth | Serial

Mode Switch

You can switch between two modes.
- **Upload Mode**: Upload code to a device to ensure the device follows instructions as programmed even after the device is disconnected.

- **Live Mode**: This mode is ideal for debugging your code in real time.

Blocks in mBlock

Events

when CyberPi starts up

when joystick pulled↑ ▾

when button A ▾ pressed

when CyberPi tilted left ▾

when wave left ▾ detected

when light ▾ value > ▾ 50

when 🚩 clicked

when space ▾ key pressed

when I receive message ▾

broadcast message ▾

broadcast message ▾ and wait

Use these special trigger blocks that initiate the execution of CyberPi's program or specific parts of it

They act like starting points that wait for something to happen before your code kicks in.

Drag and stack additional blocks below a "When" block to write your program.

Start the execution of your program at various times, such as when CyberPi starts up, when joystick pulled, when button A pressed, when flag clicked, and when space key pressed.

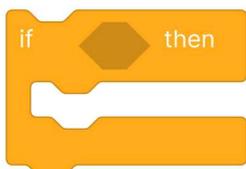when space ▾ key pressed

forever

if tilted forward ▾ ? then

display
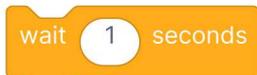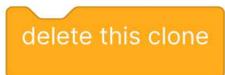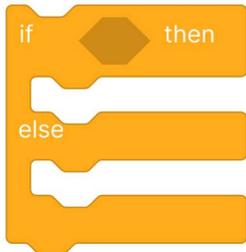
else

turn off LED all ▾
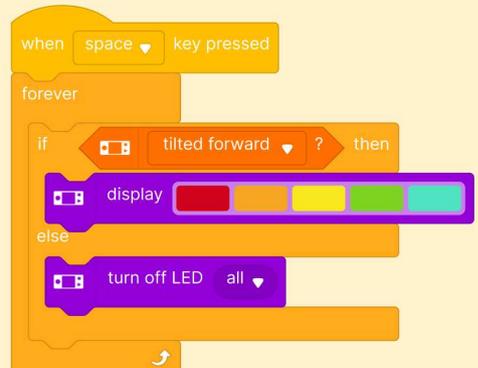
Example

In Live mode, after you press the space key, the LEDs on CyberPi are lit up when you tilt CyberPi forward and turned off when you place it in another posture.
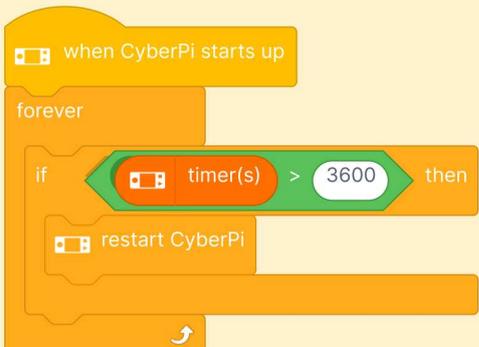
Blocks in mBlock

Control

Control how your program will start

Use a block from the Events section to indicate the event that will start executing your program.

Drag and stack additional blocks below a "When" block to write your program.

Start the execution of your program at various times, such as when CyberPi starts up, when joystick pulled, when button A pressed, when flag clicked, and when space key pressed.

Blocks shown:

- if ⬡ then / else
- wait until ⬡
- repeat until ⬡
- stop all ▾
- when I start as a clone
- create clone of myself ▾
- delete this clone
- wait 1 seconds
- repeat 10
- forever
- if ⬡ then

Example block stack:

```
when CyberPi starts up
forever
    if  timer(s) > 3600  then
        restart CyberPi
```

Example

In Live mode, after you press the space key, the LEDs on CyberPi are lit up when you tilt CyberPi forward and turned off when you place it in another posture.

Blocks in mBlock

Audios
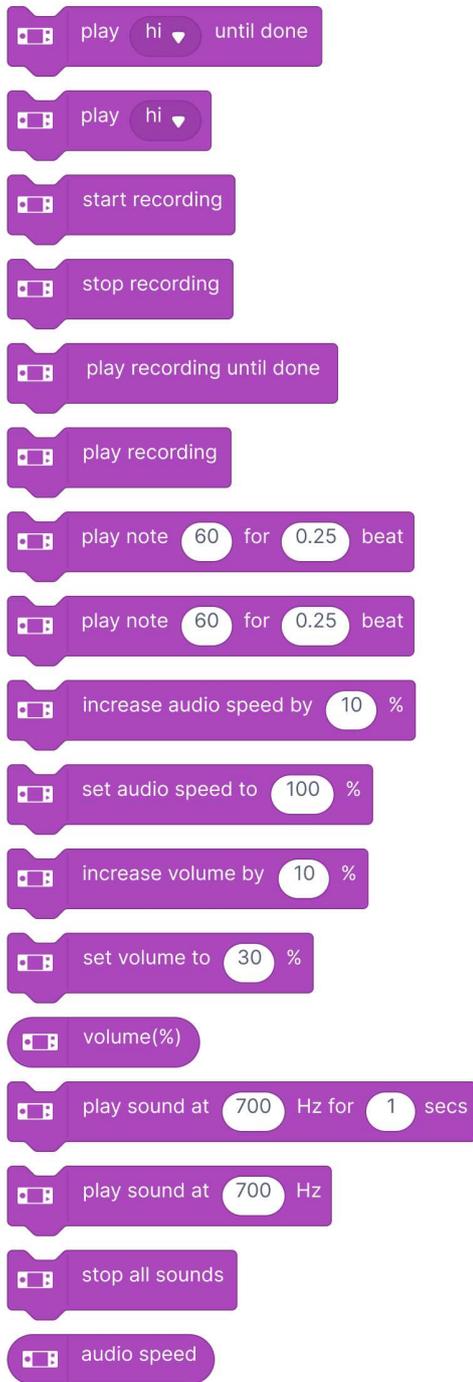
Use CyberPi to play sound and record audio.

Let CyberPi play some built-in sounds, such as hi, bye, yeah, wow.

Let CyberPi start, stop and play recording.

Play specific piano notes in specific beats. Also, play some percussion instruments, such as snare, bass drum, side stick, crash cymbal.

Adjust the audio spend and set the playback speed directly.

You can use Audio Spend as a variable.

play hi until done

play hi

start recording

stop recording

play recording until done

play recording

play note 60 for 0.25 beat

play note 60 for 0.25 beat

increase audio speed by 10 %

set audio speed to 100 %

increase volume by 10 %

set volume to 30 %

volume(%)

play sound at 700 Hz for 1 secs

play sound at 700 Hz

stop all sounds

audio speed

Example

when space key pressed
forever
if button A pressed? then
start recording
if button B pressed? then
stop recording
if joystick pulled↓ ? then
play recording

In Live mode, press the space key, and then press button A on CyberPi to record sounds; Press button B to stop the recording, and move the joystick down to play the sounds recorded.
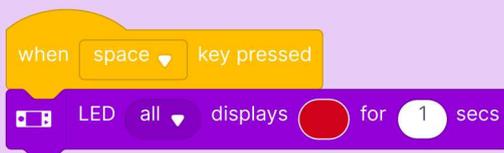
Blocks in mBlock

LED
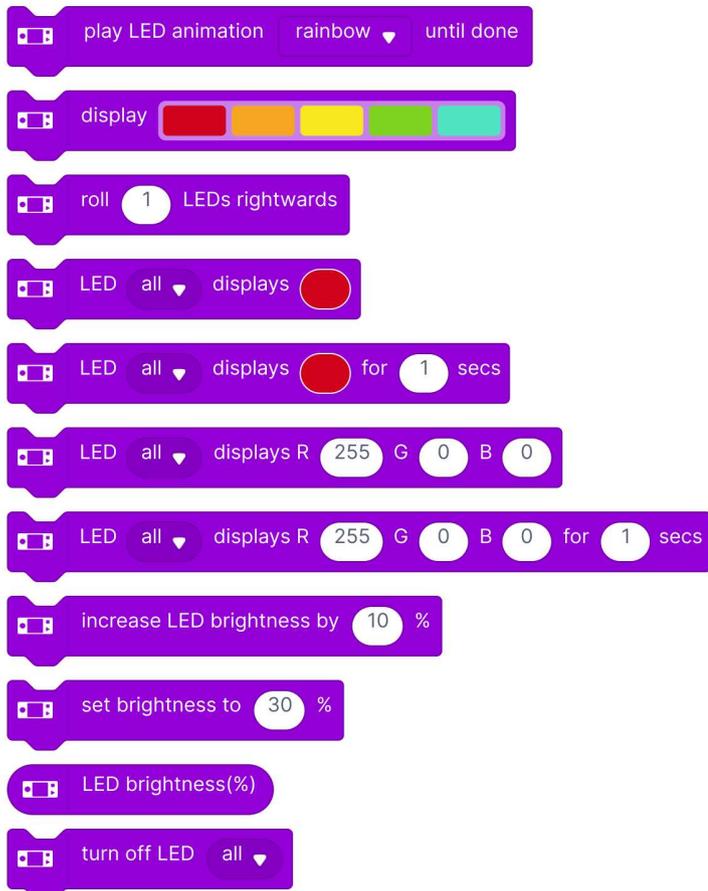
Make CyberPi's LEDs to light up in various colors, blink at different rates, and create light effects

Play with CyberPi's LEDs using pre-designed animations and drawing tools.

play LED animation rainbow ▾ until done

display 🟥🟧🟨🟩🟦

roll 1 LEDs rightwards

LED all ▾ displays ⬤

Turn on a specific LED (1 to 5) or all LEDs in a chosen color for a specified duration.

LED all ▾ displays ⬤ for 1 secs

LED all ▾ displays R 255 G 0 B 0

LED all ▾ displays R 255 G 0 B 0 for 1 secs

Adjust the brightness of a specific LED (0 for off, 100 for brightest).

increase LED brightness by 10 %

set brightness to 30 %

LED brightness(%)

turn off LED all ▾

Turn off a specific LED or all LEDs.

---

when space ▾ key pressed
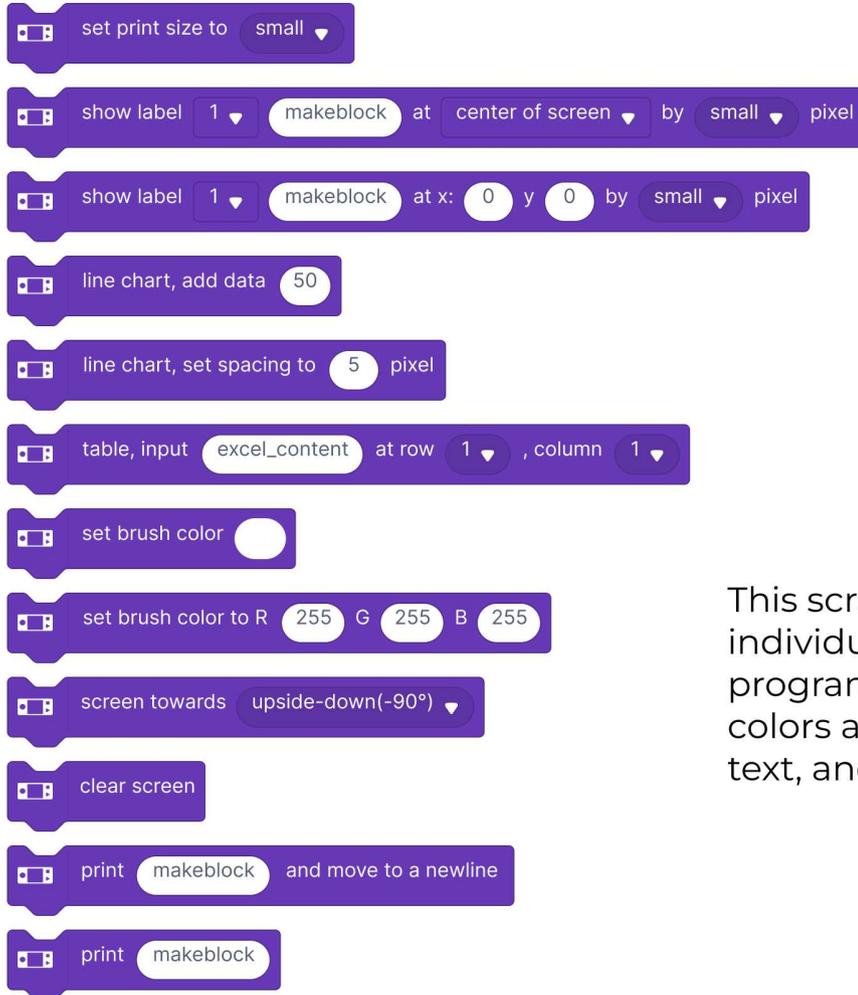
LED all ▾ displays ⬤ for 1 secs

Example

In Live mode, when you press the space key, the first LED on CyberPi is lit up in red for one second.

Blocks in mBlock

Display

Control CyberPi's built-in LED matrix screen to display something.

set print size to small ▼

show label 1 ▼ makeblock at center of screen ▼ by small ▼ pixel

show label 1 ▼ makeblock at x: 0 y 0 by small ▼ pixel

line chart, add data 50

line chart, set spacing to 5 pixel

table, input excel_content at row 1 ▼ , column 1 ▼

set brush color ⬭

set brush color to R 255 G 255 B 255

screen towards upside-down(-90°) ▼

clear screen

print makeblock and move to a newline

print makeblock

This screen consists of a grid of individual LEDs that can be programmed to light up in various colors and patterns to create visuals, text, and animations.

---

when space ▼ key pressed
  set print size to small ▼
  print makeblock and move to a newline
  wait 1 seconds
  set print size to middle ▼
  print makeblock and move to a newline

Example

In Live mode, when you press the space key, the texts are displayed on CyberPi's screen with automatic line breaks, as shown in the following figure.

Blocks in mBlock

Motion Sensing

[tilted forward ▼] ?

[tilted forward ▼] angle (°)

[wave up ▼] detected?

shaking strength

waving direction (°)

waving speed

motion sensor [x ▼] acceleration(m/s²)

angle speed around [x ▼] axis(°/s)

rotated angle around [x ▼] axis (°)

Work with CyberPi's built-in accelerometer, button, joystick, speaker, light sensor and micophohe

Use CyberPi to detect movement, tilt, and shakes, enabling you to program your CyberPi to react to physical actions.

Use CyberPi to gather information about its environment through its built-in sensors. Create fun virtual physical interactions with other blocks!

Sensing

button [A ▼] pressed?

button [A ▼] press counts

button [A ▼] pressed?

button [A ▼] press counts

reset button [A ▼] press counts

reset joystick [middle pressed ▼] counts

loudness

ambient light intensity

timer(s)

reset timer

hostname

```
when [ space ▼ ] key pressed
forever
    if < [ tilted forward ▼ ] ? > then
        display [ ▮▮▮▮▮ ]
    else
        turn off LED [ all ▼ ]
```

Example

In Live mode, after you press the space key, the LEDs on CyberPi are lit up when you tilt CyberPi forward and turned off when you place it in another posture.

Blocks in mBlock

IoT
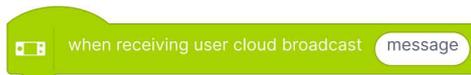
Enable communication and interaction between CyberPi and other devices or the cloud.

```
connect to Wi-Fi  ssid  password  password
```

```
network connected?
```

```
connect to Wi-Fi again
```

```
disconnect from Wi-Fi
```

Connect to Wi-Fi. This series of blocks can help you control the Wi-Fi connection with CyberPi.

```
when receiving user cloud broadcast  message
```

```
user cloud broadcast  message  value received
```

```
send user cloud broadcast  message
```

```
send user cloud broadcast  message  with value  1
```

Send user cloud broadcast. This series of blocks can allow you to send and receive message via cloud broadcast.

```
location  highest temperature (°C) ▼
```

```
air quality  location  AQI ▼
```

```
location  sunrise ▼  time ▼
```

```
(0) UTC ▼  's  year ▼
```

```
(0) UTC ▼  's date and time
```

```
year ▼  of date and time
```

Obtain weather information for a certain area. After connected CyberPi to the Internet, this series of blocks can allow you to obtain weather information for a certain area, including temperature, air quality, age, time zone and other information.

Example

```
when CyberPi starts up
connect to Wi-Fi  ssid  password  password
wait until  network connected?
display
```

Compile this program in Upload mode and upload it to CyberPi. Press button A. CyberPi starts to connect to the specified Wi-Fi. When the LEDs are lit up in green, the connection is successful.
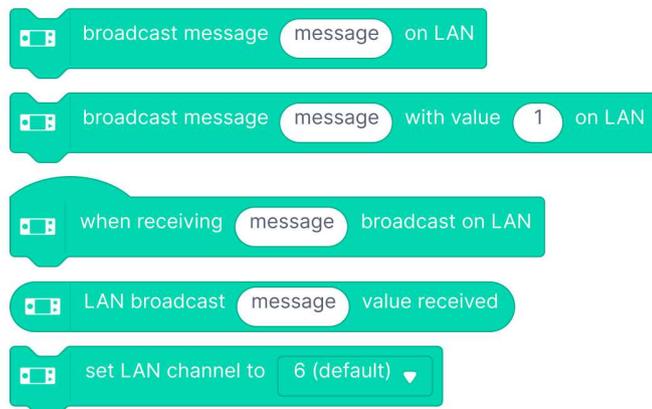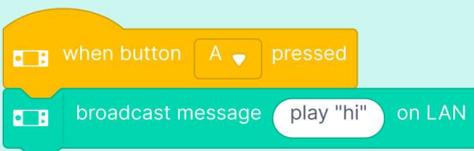
Blocks in mBlock

AI

Program CyberPi to perform some basic Artificial Intelligence tasks such as Speech Recognition and Text Translation

- speak auto ▾ hello world
- recognize (4) English ▾ 3 secs
- speech recognition result
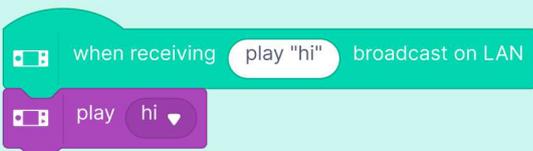- translate hello into Spanish ▾

LAN

Connect and interact with other CyberPi devices within the same local network. This series of blocks enable you to create programs where multiple CyberPis can communicate , react to each other's messages, and trigger actions on each other.

- broadcast message message on LAN
- broadcast message message with value 1 on LAN
- when receiving message broadcast on LAN
- LAN broadcast message value received
- set LAN channel to 6 (default) ▾

CyberPi 1

- when button A ▾ pressed
- broadcast message play "hi" on LAN

CyberPi 2

- when receiving play "hi" broadcast on LAN
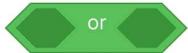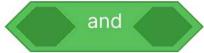- play hi ▾

Example

Compile the two programs in Upload mode and upload them to CyberPi 1 and 2 respectively.
Restart CyberPi 1 and 2.
Press button A on CyberPi 1. CyberPi 2 says "hi."

Blocks in mBlock

Operator

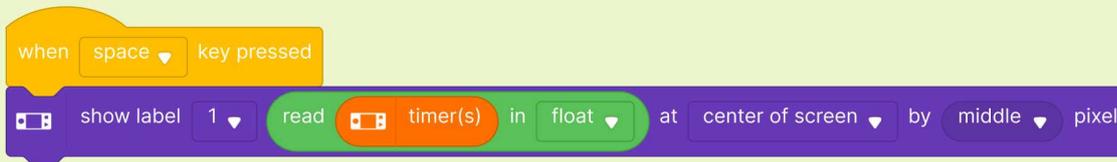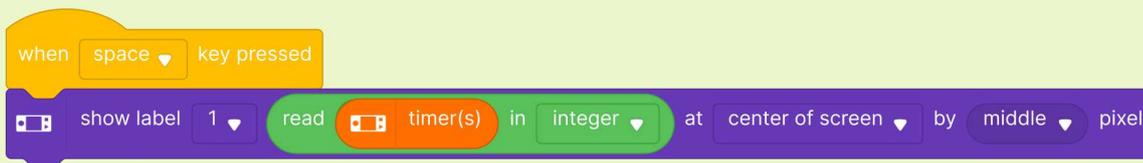Perform various operations on data

+

-

*

/

Perform basic arithmetic operations like addition, subtraction, multiplication, and division using the corresponding operator blocks

pick random **1** to **10**

Pick random number from 1 to 10.

> 50

< 50

= 50

Compare values and make decisions based on the outcome.

and

or

Create conditional statements where your program's behavior changes based on certain conditions.

not

read **in** integer ▾

Read a variable as an integer, floating point number, or string.

---

when space ▾ key pressed

show label **1** ▾ read 🔲 timer(s) in integer ▾ at center of screen ▾ by middle ▾ pixel

when space ▾ key pressed

show label **1** ▾ read 🔲 timer(s) in float ▾ at center of screen ▾ by middle ▾ pixel

Example

Compile this program in Upload mode and upload it to CyberPi. Press button A. CyberPi starts to connect to the specified Wi-Fi. When the LEDs are lit up in green, the connection is successful.
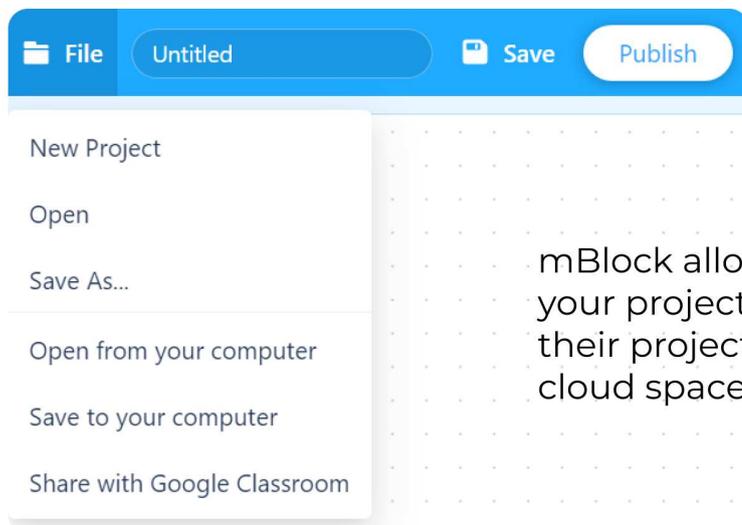
Block Shapes

Variables

Creating variables to meet specific needs

| Make a Variable |

☑ a

set a ▾ to 0

change a ▾ by 1

show variable a ▾

hide variable a ▾

| Make a List |

In programming, variables act like boxes to hold values. They are called "variables" because what is in them can change.

Make a Variable. Click this button to create a new variable for your programming.

Creating and Saving Projects

📁 File   Untitled   💾 Save   Publish

New Project

Open

Save As...

Open from your computer

Save to your computer

Share with Google Classroom

mBlock allows you to save, create, and load your projects. This allows students to name their projects and save them to the local or cloud spaces.