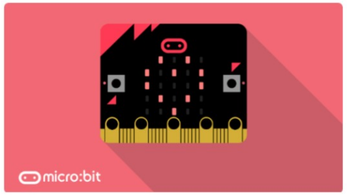
Robotics with Lego EV3 + MS Makecode

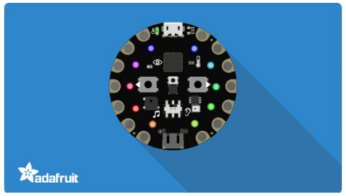
Andrea Sterbini – sterbini@di.uniroma1.it

Microsoft Makecode.com

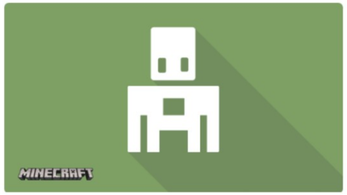
Many development systems supported (embedded/robotics/game)



micro:bit



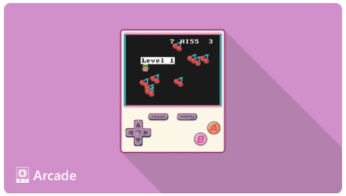
Adafruit



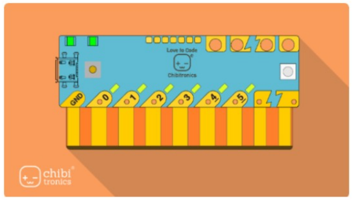
Minecraft

Lego EV3

Cue



Arcade



Chibi chip

Blockly-based visual programming

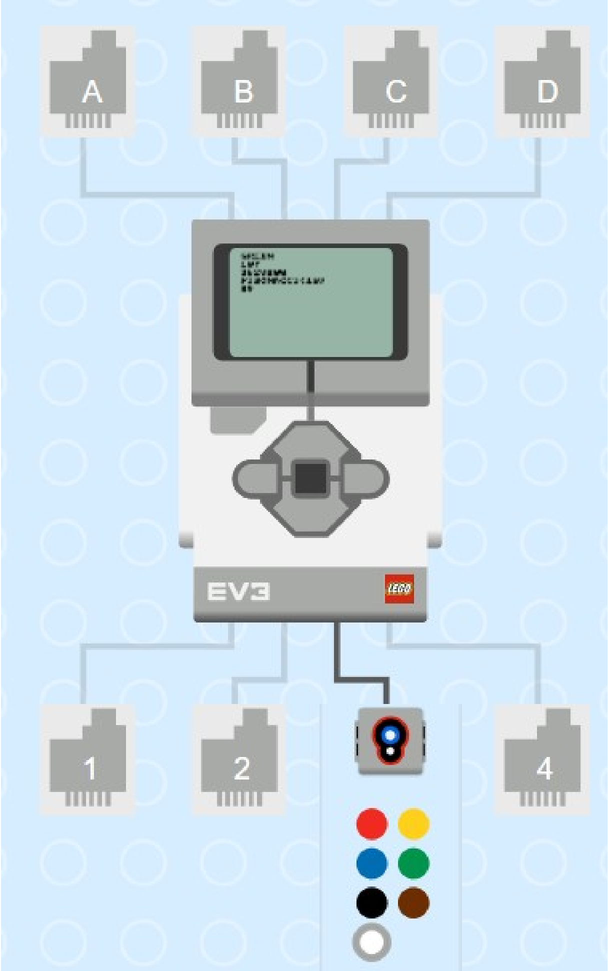
More systems in https://makecode.com/labs

MS Makecode: EV3 robotics
https://makecode.mindstorms.com

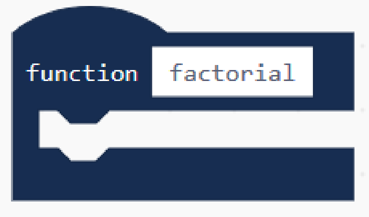
INSTALLATION
- just upgrade the EV3 firmware to 1.10E
- the IDE runs in the browser

EXECUTION
- EV3 is seen as a disk when connected by USB
- Just download the generated code to the EV3
    - (in Linux there is a mount error)

DEBUG
- browser-based simulator (AUTO-CONFIGURED)



Makecode standard blocks



Types!!!	integers, strings, floats, lists	
Lists of any?	YES	
Functions?	YES	
Function args?	NO	(YES in TypeScript mode)
Return?	NO	(YES in TypeScript mode)
Variables?	GLOBAL	(LOCAL in TypeScript mode)
Messages?	YES	
Message params.?	YES	
Static TypeScript?	YES	(NEW!!!!)

Makecode EV3-specific blocks

Brick buttons:

- on button XXX pressed event
- pause until ...
- is button … ?

Brick LCD screen:

- clear, show image, show text
  show number, show port

Touch sensor:

- on touch XXX event
- pause until ...
- is touch … ?

Color sensor:

- on color XXX detected event
- on color sensor X dark/light
- pause until …
- color

Ultrasonic sensor:

- on US X object detected
- pause until …
- distance

Gyroscope sensor:

- rate, angle, reset

EV3 Sensor Calibration blocks

Calibrate color sensor XXX for reflected/ambient light

Set color sensor XXX dark/bright to THRESHOLD

Set ultrasonic sensor XXX object detected/near to THRESHOLD

Set infrared sensor XXX object detected/near to THRESHOLD

EV3 Motors

Run motor X/XY at V speed for N rotations/degrees/seconds/msec

Drive motors XY at V1,V2 speeds for N rot/deg/sec/msec

Steer motors XY at Y ratio V speed for N rot/deg/sec/msec

Pause until motor X/XY ready

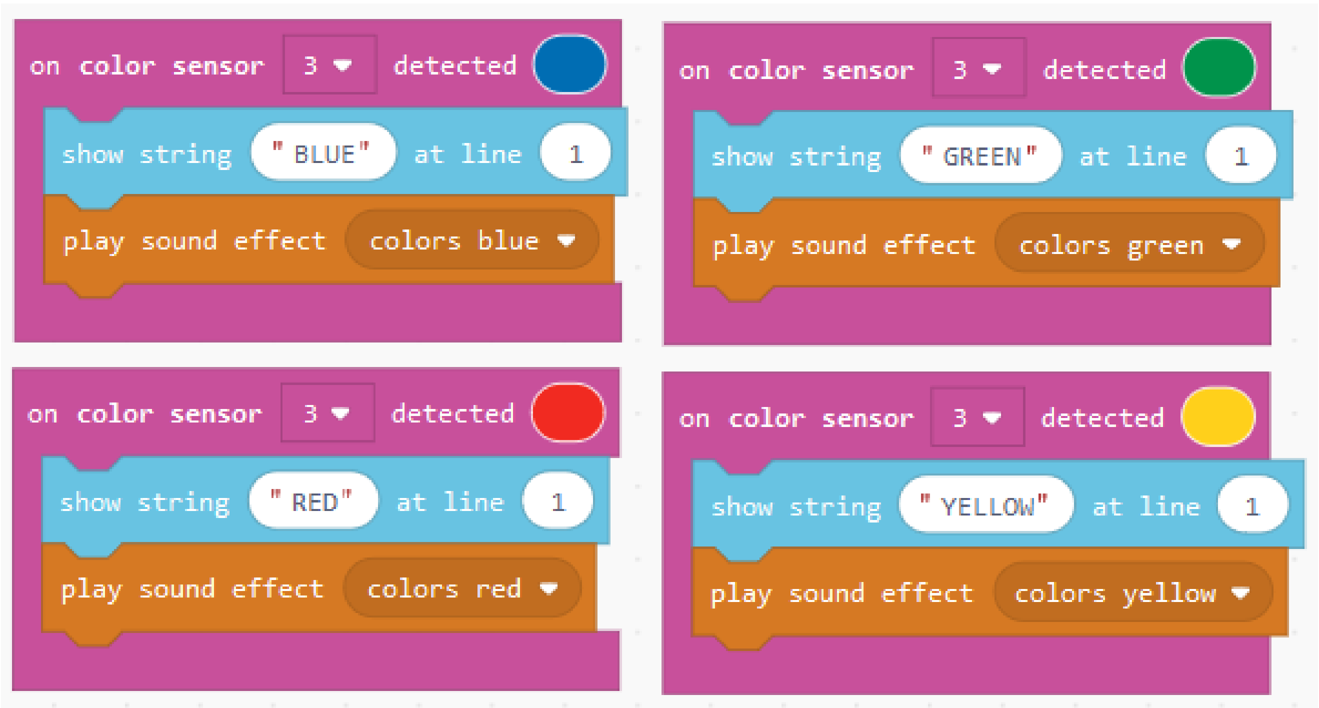
Motor X speed/angle

Set motor X brake/pause/inverted/regulated ON/OFF

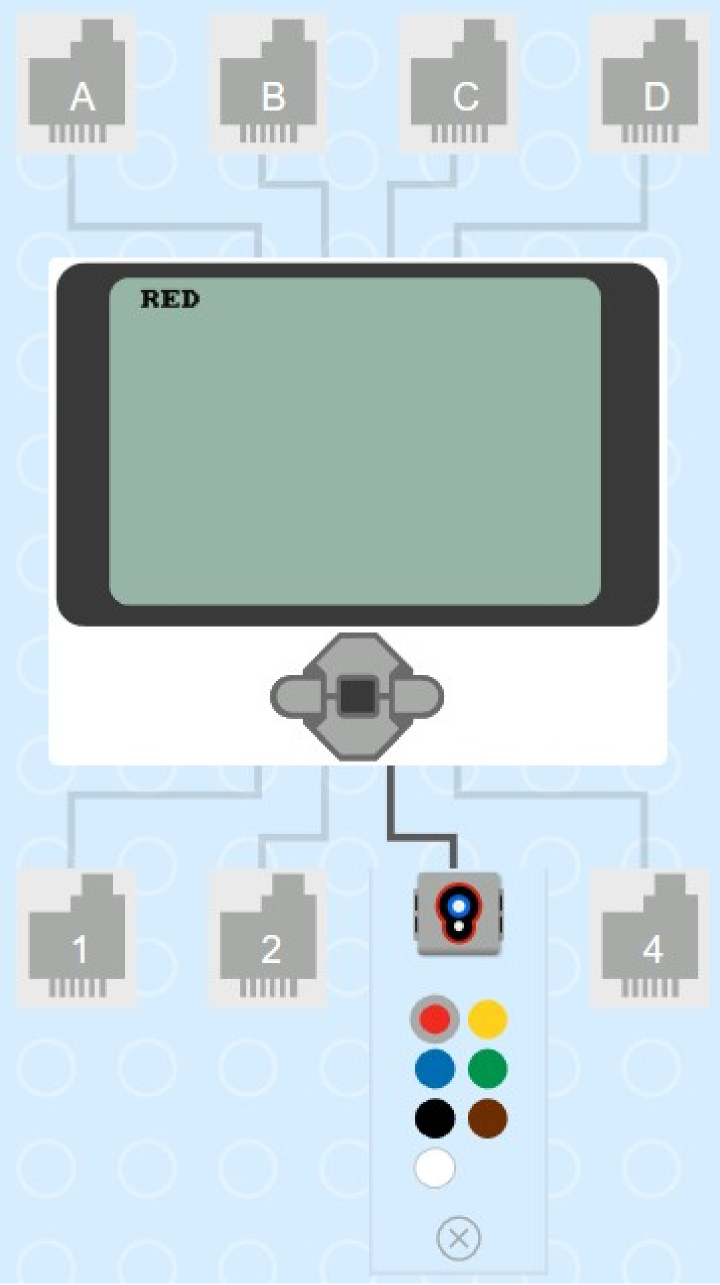
Control flow (blocks)

One main thread	NO MULTI	Parallel threads?	EXPLICIT
One forever loop	NO MULTI	(“run in parallel”)	
		Wait for all threads?	YES
Sensor events	ONE EACH		
		New events?	YES
		Parametric events?	YES
Counted loops?	YES	Wait for event?	YES
Foreach?	YES		
Do-while?	NO		
While-do?	YES	Timers?	YES

Color recognizer example



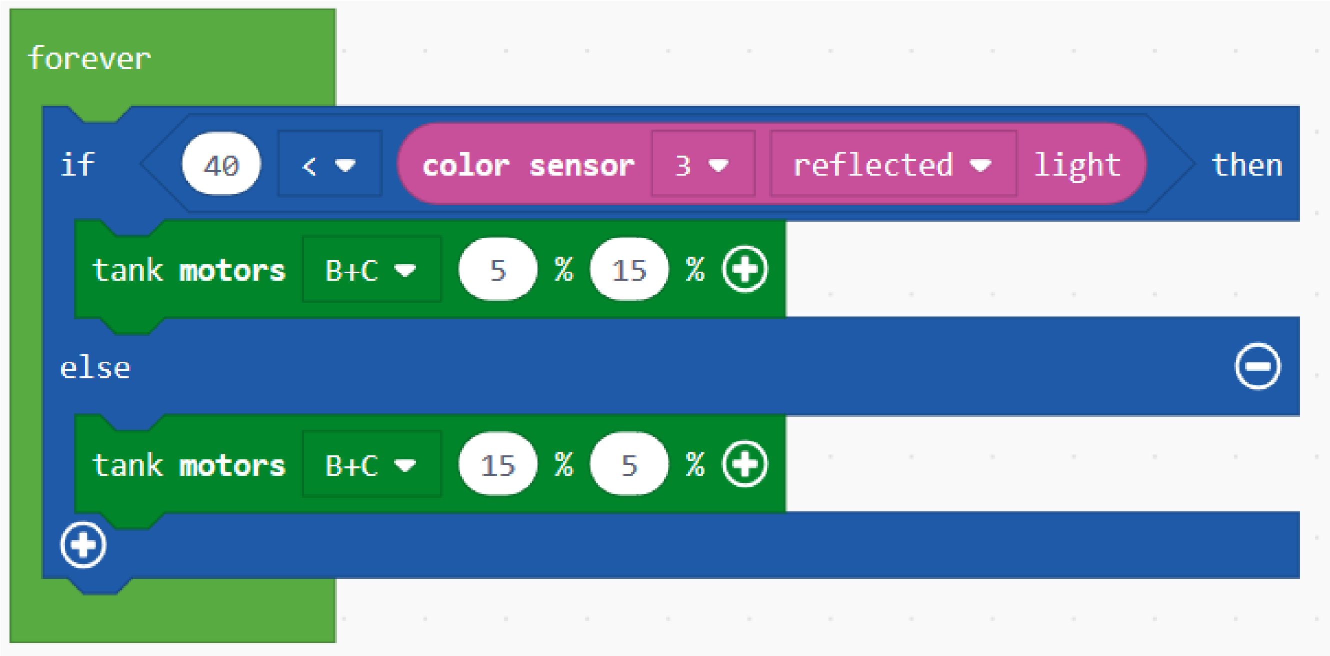
```
on color sensor 3 ▾ detected 🟦
    show string " BLUE " at line 1
    play sound effect colors blue ▾

on color sensor 3 ▾ detected 🟩
    show string " GREEN " at line 1
    play sound effect colors green ▾

on color sensor 3 ▾ detected 🟥
    show string " RED " at line 1
    play sound effect colors red ▾

on color sensor 3 ▾ detected 🟨
    show string " YELLOW " at line 1
    play sound effect colors yellow ▾
```



```
RED
```

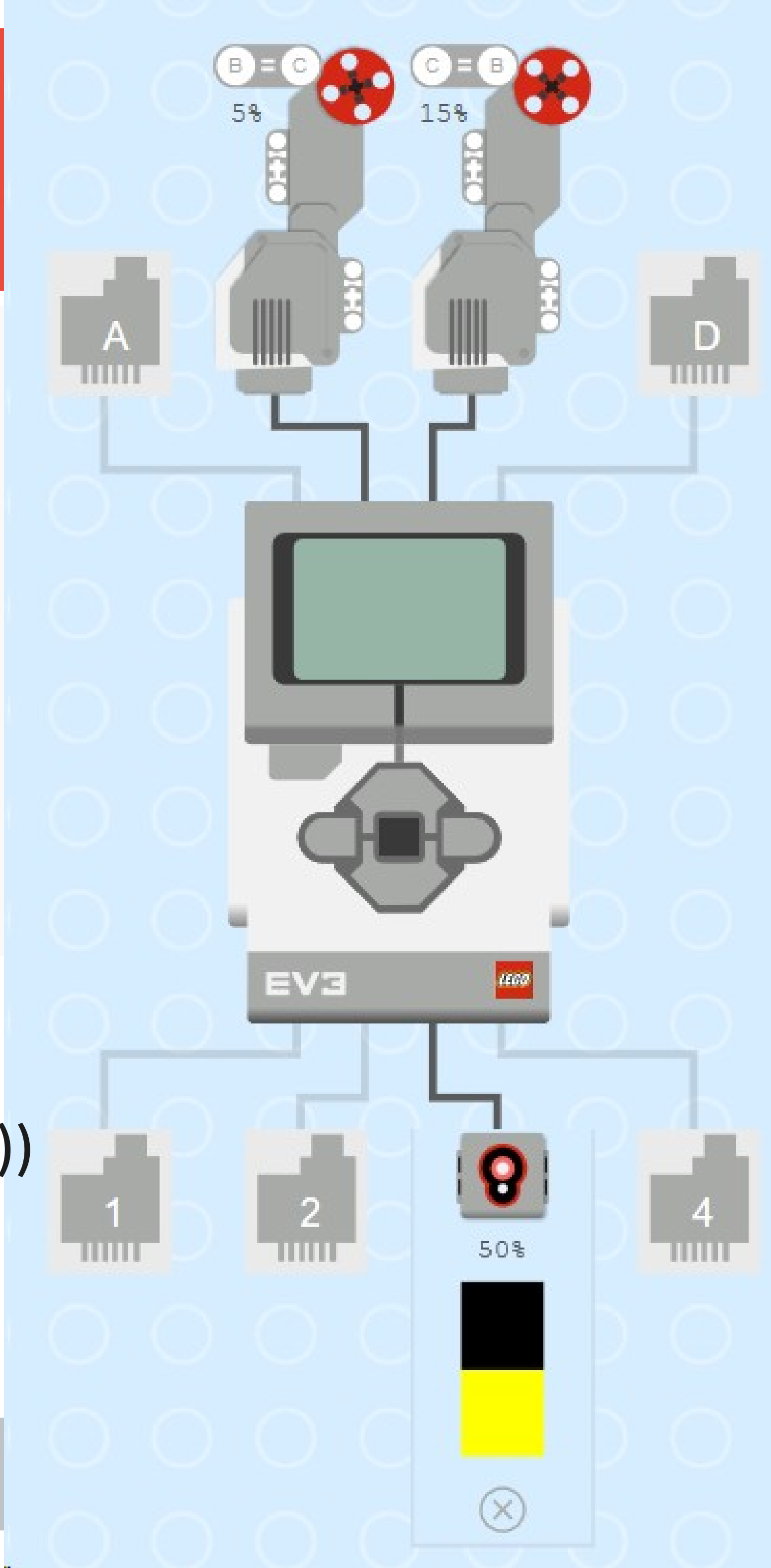
```
sensors.color3.onColorDetected(ColorSensorColor.Blue,
function (){
    brick.showString("RED", 1)
    music.playSoundEffect(sounds.colorsBlue)
})
```

Line follower example

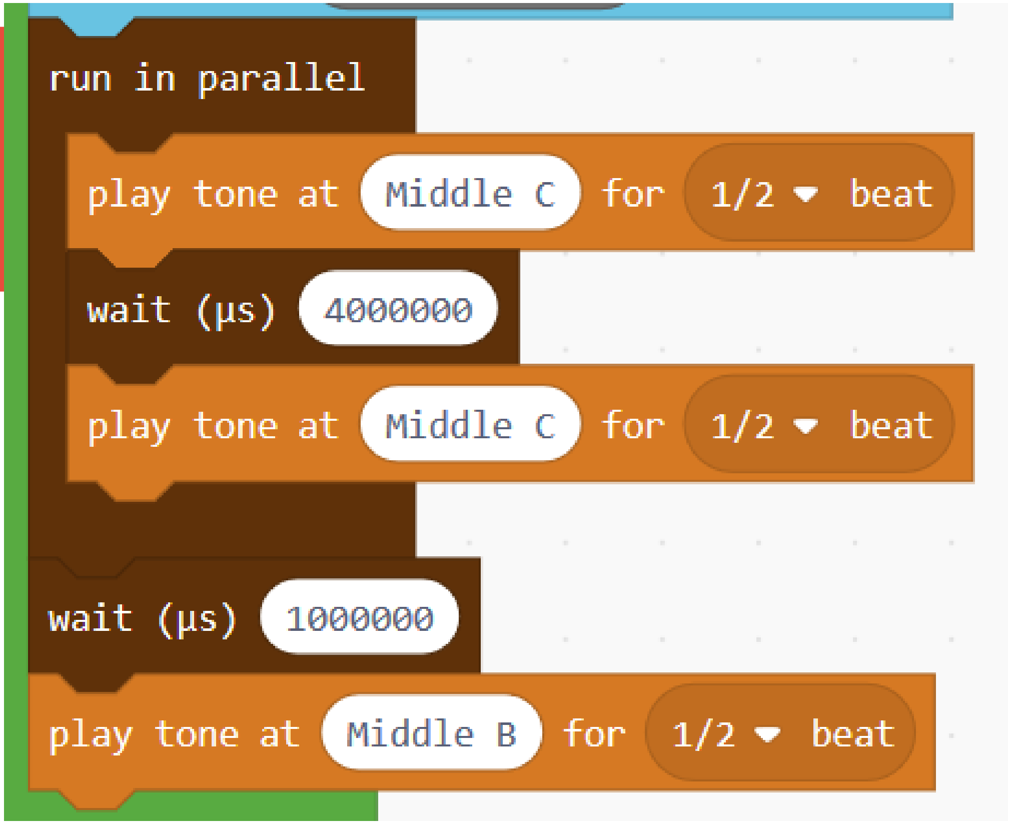


```
forever
  if  40  <  color sensor  3  reflected  light  then
    tank motors  B+C  5 %  15 %
  else
    tank motors  B+C  15 %  5 %
```

```
forever(function () {
  if (40 <
sensors.color3.light(LightIntensityMode.Reflected))
        {    motors.largeBC.tank(5, 15)    }
else    {    motors.largeBC.tank(15, 5)    }})
```



Parallel thread example



```
run in parallel
  play tone at Middle C for 1/2 beat
  wait (µs) 4000000
  play tone at Middle C for 1/2 beat
wait (µs) 1000000
play tone at Middle B for 1/2 beat
```

In parallel:

- wait then beep
- beep, wait then beep

```
control.runInParallel(function () {
  music.playTone(262, music.beat(BeatFraction.Half))
  control.waitMicros(4000000)
  music.playTone(262, music.beat(BeatFraction.Half))

})
control.waitMicros(1000000)
music.playTone(494, music.beat(BeatFraction.Half))
```

TypeScript mode

Editor with colour highlight, autocompletion and documentation

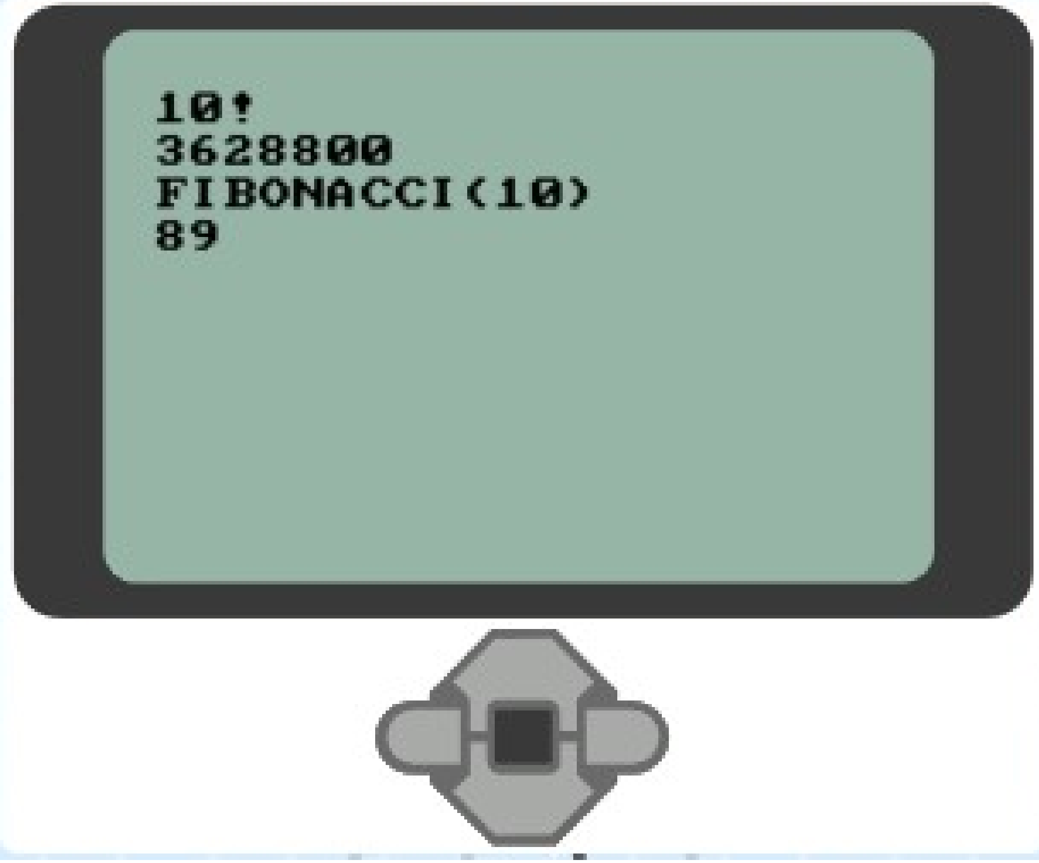
Static TypeScript (Typed JavaScript)

Object-oriented! (to be investigated)

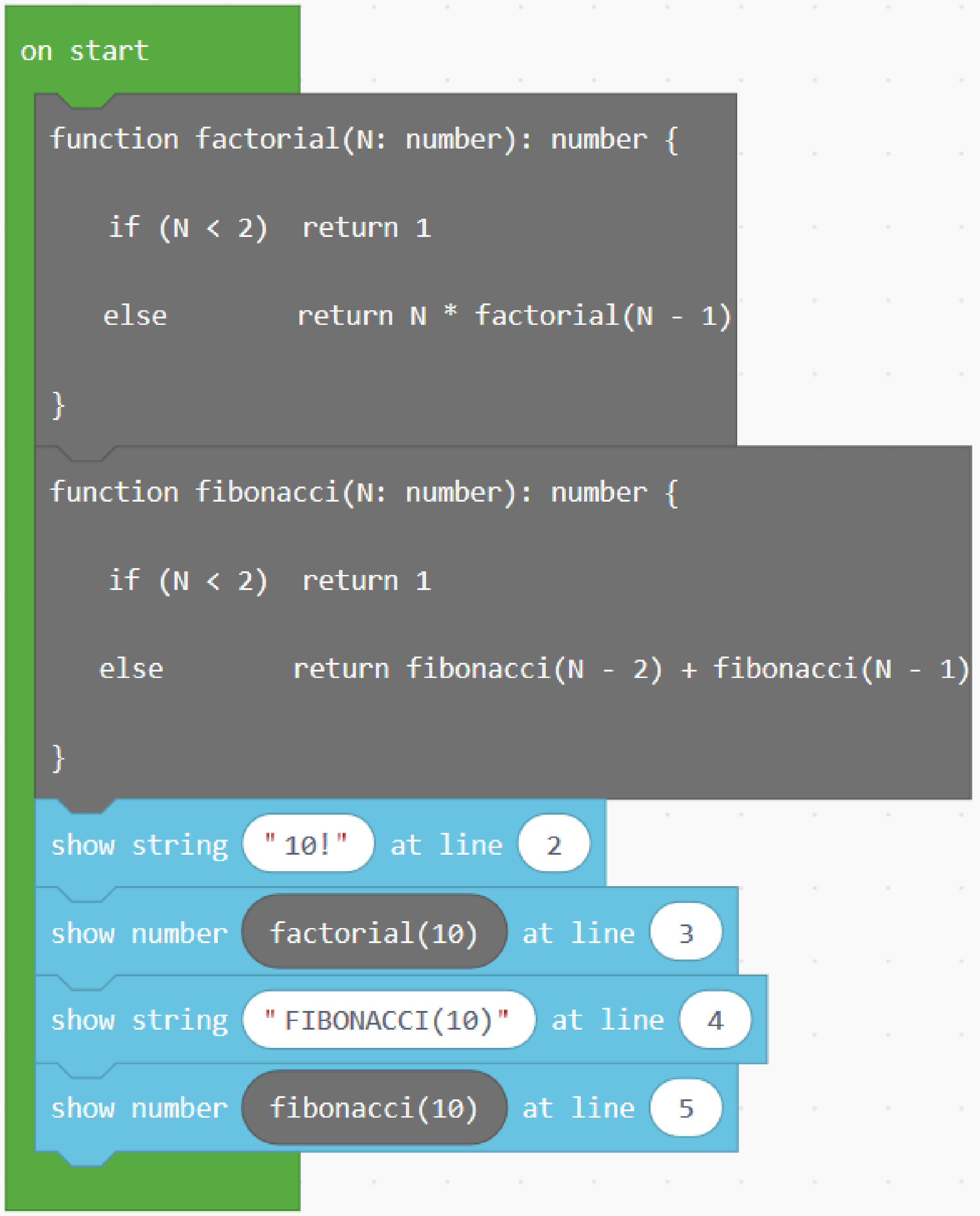
A sequence of statements and declarations
FOLLOWED by an infinite loop

Static Python in a near future?

Recursion example



```
10!
3628800
FIBONACCI(10)
89
```



```
on start
    function factorial(N: number): number {

        if (N < 2)  return 1

        else        return N * factorial(N - 1)

    }
    function fibonacci(N: number): number {

        if (N < 2)  return 1

        else        return fibonacci(N - 2) + fibonacci(N - 1)

    }
    show string "10!" at line 2
    show number factorial(10) at line 3
    show string "FIBONACCI(10)" at line 4
    show number fibonacci(10) at line 5
```

Demo

https://makecode.mindstorms.com

DEMO