



## Example Program – JavaScript Blocks Editor

```

function runbump
  call function forward
  show icon
  while not @ button B is pressed
  do
    pause (ms) 100
  call function stop

function forward
  digital write pin PB to 0
  digital write pin P12 to 1
  digital write pin PB to 0
  digital write pin P16 to 1
  show arrow North

function stop
  digital write pin PB to 1
  digital write pin P12 to 1
  digital write pin PB to 1
  digital write pin P16 to 1
  show icon

function left
  digital write pin PB to 0
  digital write pin P12 to 1
  digital write pin PB to 1
  digital write pin P16 to 0
  show arrow West

function right
  digital write pin PB to 1
  digital write pin P12 to 0
  digital write pin PB to 0
  digital write pin P16 to 1
  show arrow East

forever
  call function runbump
  call function back
  pause (ms) 500
  call function left
  call function stop
  pause (ms) 1000
  
```

- Go to [www.microbit.org/code](http://www.microbit.org/code) and open the **JavaScript Blocks Editor**.
- Drag the file **microbit-robot-jsb.hex** onto the work area.
- This program acts as a simple maze solver by driving the robot **forward** until it hits an **obstacle**. It then **backs off** and **turns left** a quarter turn before repeating.
- Test it, download it and **experiment** with how it works!





## Example Program – Python Editor

```
1 from microbit import *
2
3 def motors(la, lb, ra, rb):
4     pin8.write_digital(la)
5     pin12.write_digital(lb)
6     pin0.write_digital(ra)
7     pin16.write_digital(rb)
8
9 def forward():
10    motors(0,1,0,1)
11    display.show(Image.ARROW_N)
12
13 def back():
14    motors(1,0,1,0)
15    display.show(Image.ARROW_S)
16
17 def left():
18    motors(0,1,1,0)
19    display.show(Image.ARROW_W)
20
21 def right():
22    motors(1,0,0,1)
23    display.show(Image.ARROW_E)
24
25 def stop():
26    motors(1,1,1,1)
27    display.show(Image.NO)
28
29 def run_bump():
30    forward()
31    display.show(Image.TRIANGLE)
32    while not button_b.was_pressed():
33        sleep(100)
34    stop()
35
36 while True:
```

```
37     run_bump()
38     stop()
39     sleep(250)
40     back()
41     sleep(500)
42     stop()
43     sleep(250)
44     left()
45     sleep(400)
46     stop()
47     sleep(1000)
48
```

- Go to [www.microbit.org/code](http://www.microbit.org/code) and open the **Python Editor**.
- Drag the file **robot.py** onto the work area.
- This program acts as a simple maze solver by driving the robot **forward** until it hits an **obstacle**. It then **backs off** and **turns left** a quarter turn before repeating.
- Test it, download it and **experiment** with how it works!

