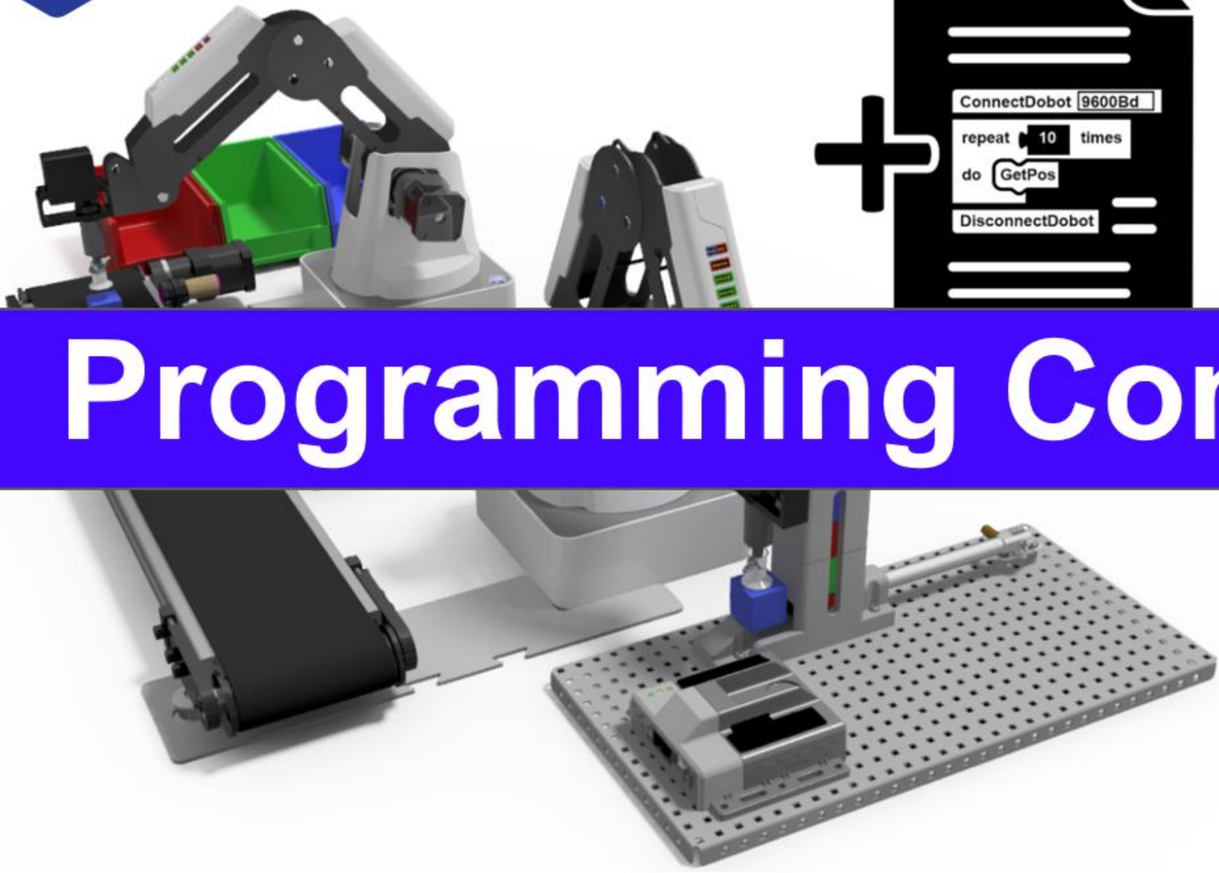
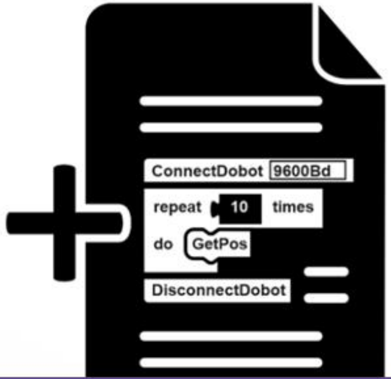
DOBOT





Blockly & Dobot

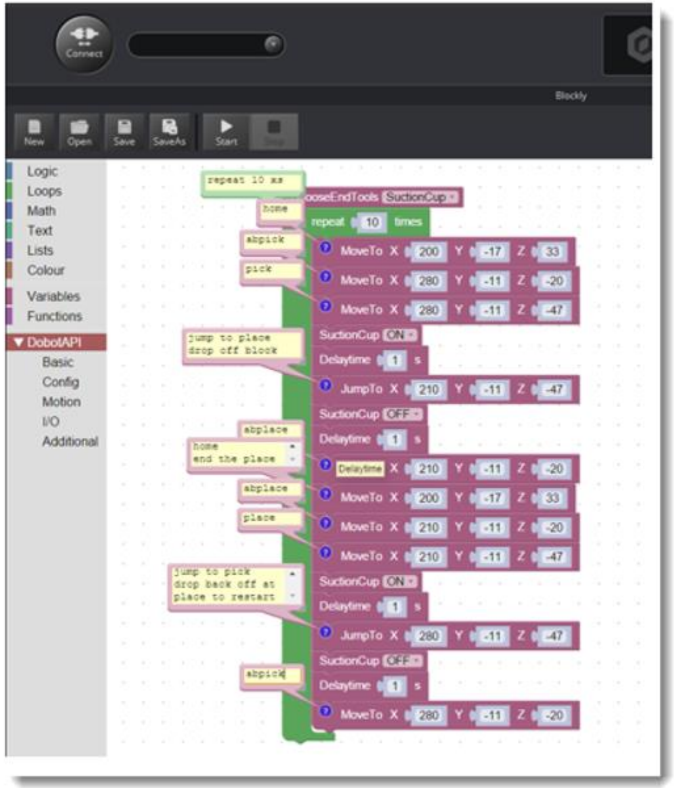
Programming Commands

www.ChrisandJimCIM.com

Blockly Commands for the Dobot Magician

Blockly Definition:

A programming language used to program the Dobot Magician. Lines of complex code are represented by simple “blocks” that fit together to form a program. **Blockly** is a graphical programming method rather than text based.



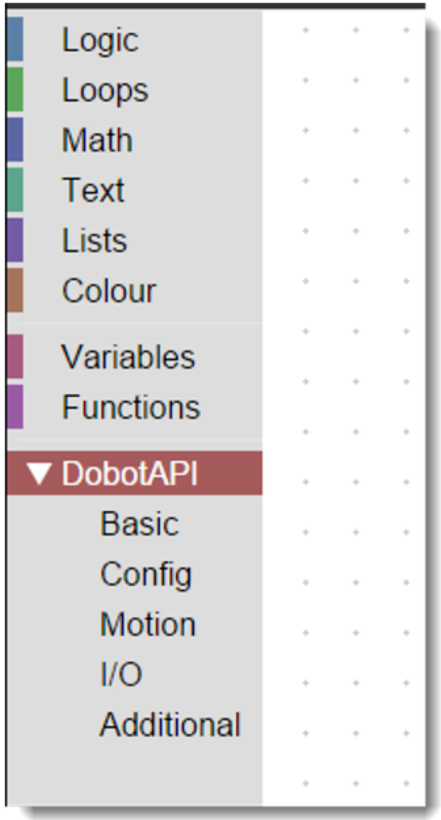
```
ooseEndTools SuctionCup
repeat 10 times
    MoveTo X 200 Y -17 Z 33
    MoveTo X 280 Y -11 Z -20
    MoveTo X 280 Y -11 Z -47
    SuctionCup ON
    Delaytime 1 s
    JumpTo X 210 Y -11 Z -47
    SuctionCup OFF
    Delaytime 1 s
    Delaytime X 210 Y -11 Z -20
    MoveTo X 200 Y -17 Z 33
    MoveTo X 210 Y -11 Z -20
    MoveTo X 210 Y -11 Z -47
    SuctionCup ON
    Delaytime 1 s
    JumpTo X 280 Y -11 Z -47
    SuctionCup OFF
    Delaytime 1 s
    MoveTo X 280 Y -11 Z -20
```

Blockly Commands for the Dobot Magician

Types of Commands in Blockly

In DobotStudio, Blockly commands are broken up into nine different categories with one category, **DobotAPI**, broken up into 5 subcategories.

Each of these have similar commands grouped together, and this presentation will describe and define some of the most common blocks that are used in programming the Dobot Magician.



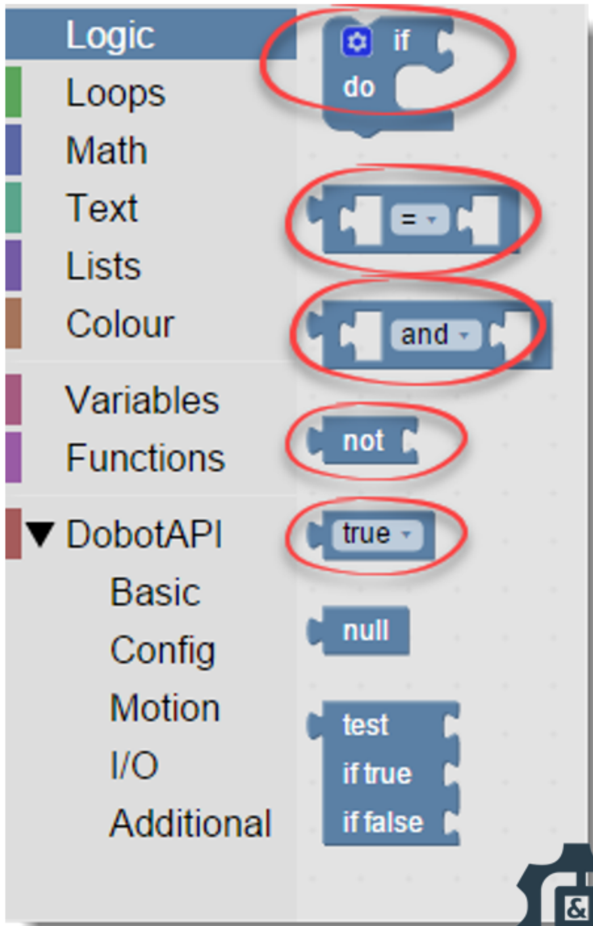
Blockly Commands: Logic

Logic

Logic commands allow you to use Boolean operands to make your robot complete complex operations.

Some of the important tasks that these blocks will allow you to do are:

- ***If Else** statements*
- *Set two programming elements to <,>,=*
- *Use **AND**, **OR** and **NOT***
- *Set something to **TRUE** or **FALSE***

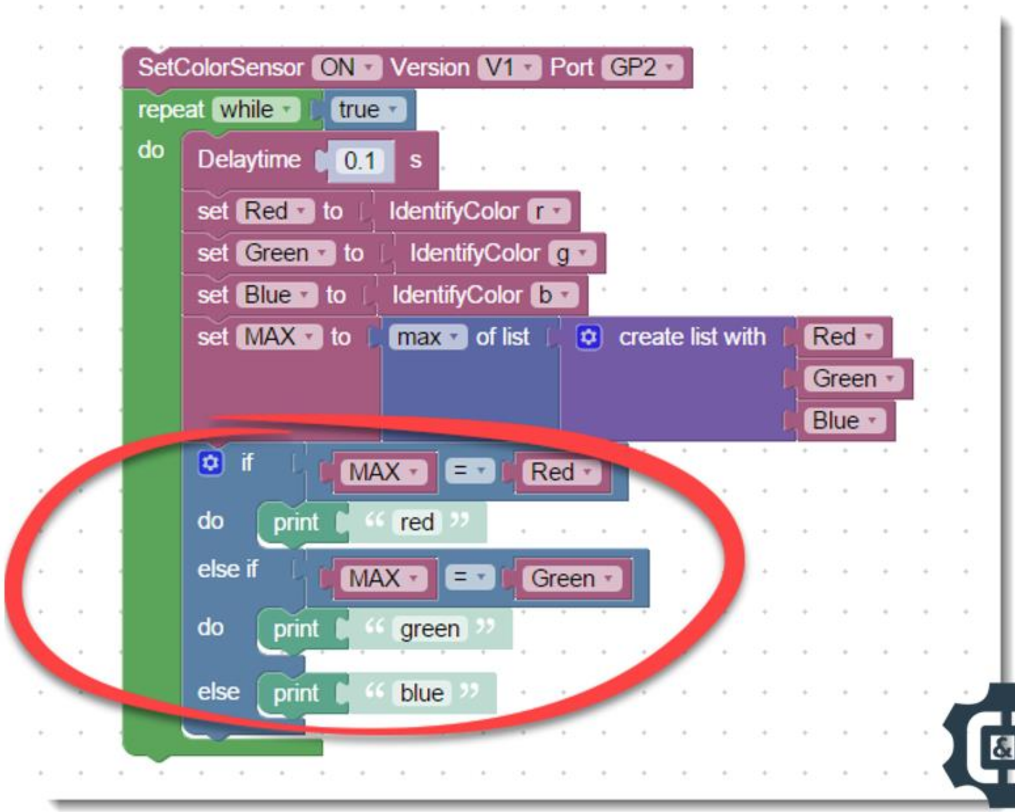


Blockly Commands: Logic

Logic

*In this example an **If Else** statement is used to make a color sensor print the color of the part being sensed.*

Notice that there are actually three different conditions.



```
SetColorSensor ON Version V1 Port GP2
repeat while true
do  Delaytime 0.1 s
    set Red to IdentifyColor r
    set Green to IdentifyColor g
    set Blue to IdentifyColor b
    set MAX to max of list create list with Red
                                            Green
                                            Blue
    if MAX = Red
    do   print " red "
    else if MAX = Green
    do   print " green "
    else print " blue "
```

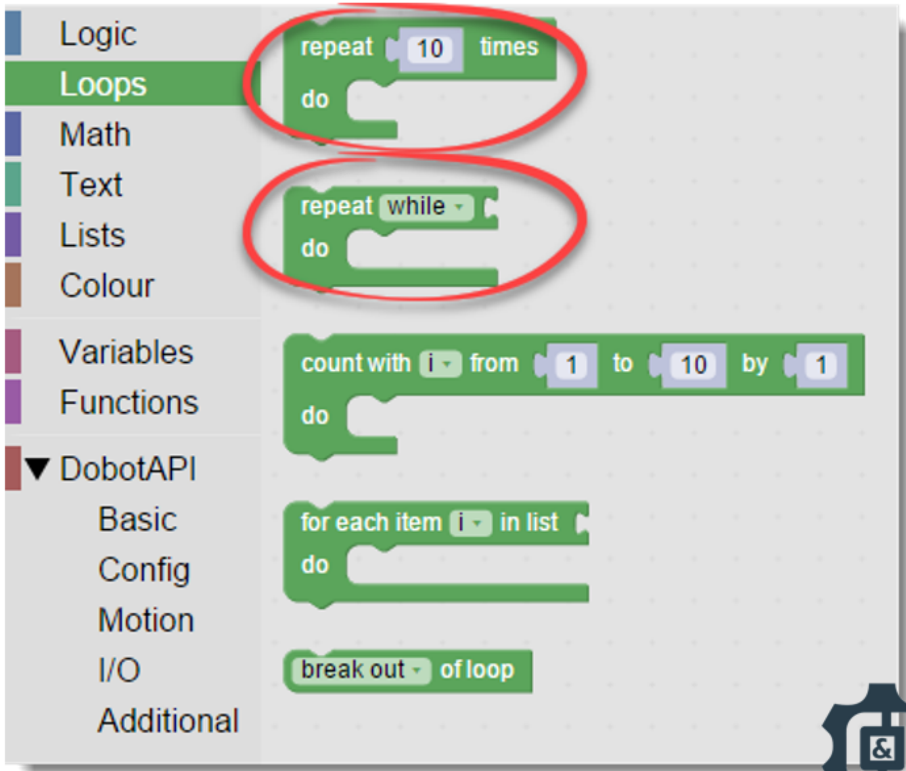
Blockly Commands: Loops

Loops

Logic commands that will allow you to repeat actions within a program.

Some of the important tasks that these blocks will allow you to do are:

- ***Repeat** a number of times*
- ***Repeat while** something is happening*
- ***Repeat forever***

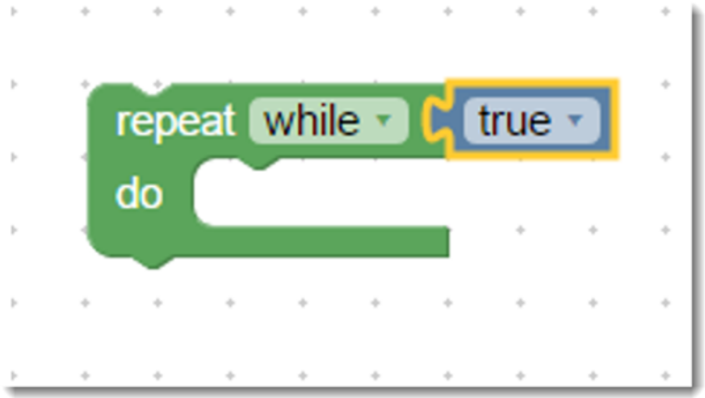


Blockly Commands: Loops

Repeat forever

*Blockly logic commands that allows you to use **TRUE** with the repeat command to continually complete an action or set of actions*

*In this example you can use this block with a **TRUE** block and make a block, or group of blocks run continuously. This could be used when you want to continuously look for an input.*



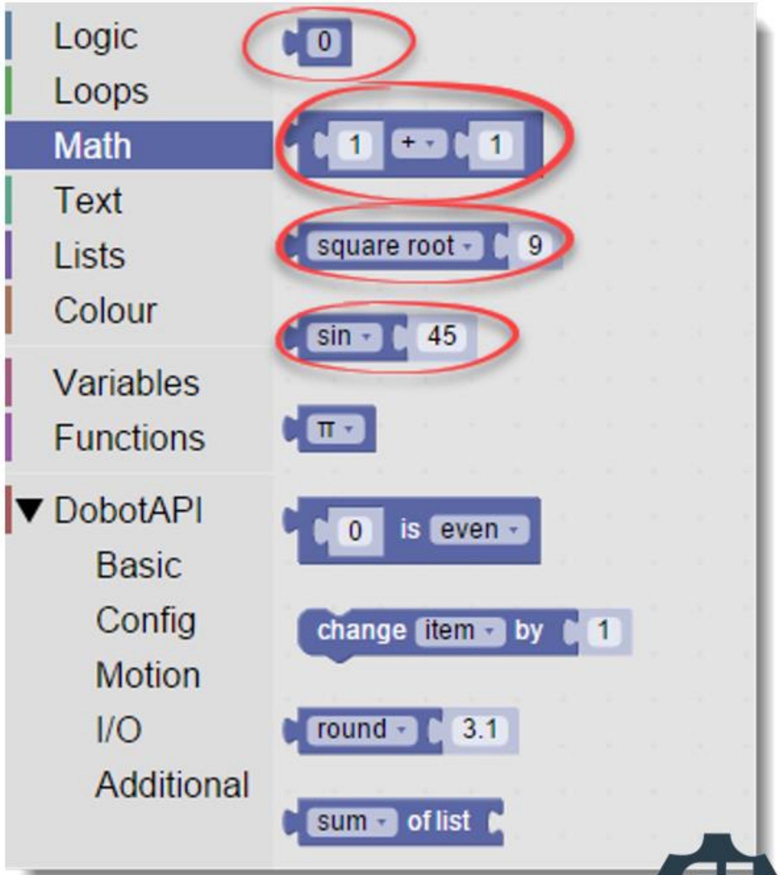
Blockly Commands: Math

Math

Logic commands that allow you to use mathematics on numbers in your program. These are all very self explanatory.

Some of the important tasks that these blocks will allow you to do are:

- *Return a number of your choice*
- *Return a SUM of two numbers*
- *Return the sine/cosine/tangent of a number*
- *Return the square root of a number*

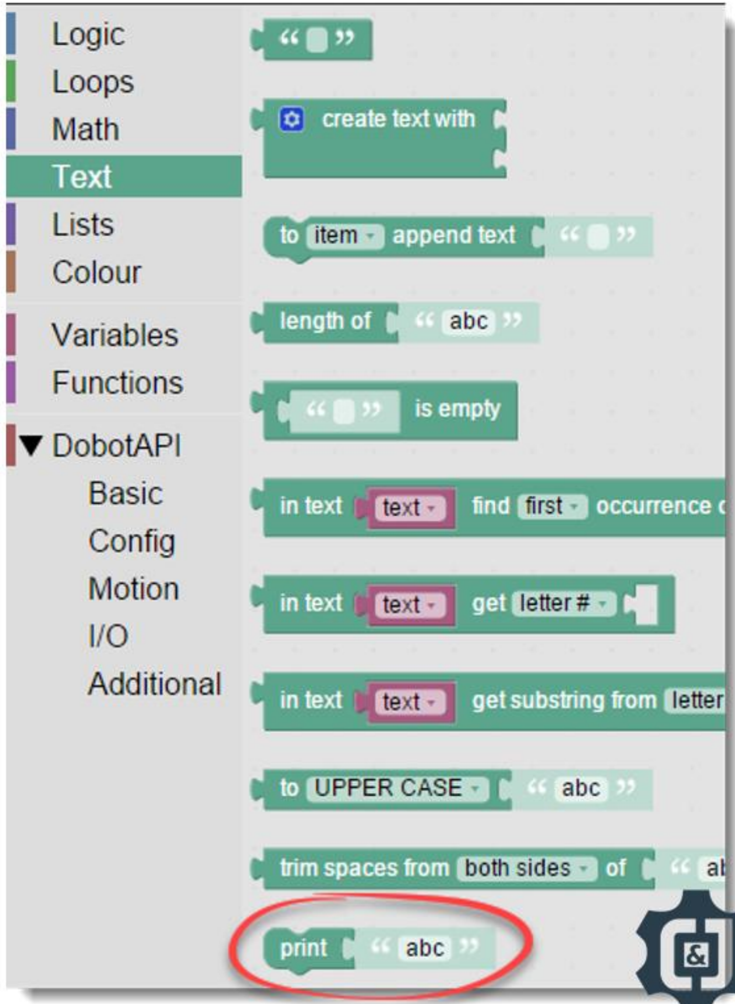


Blockly Commands: Text

Text

*Logic commands that will allow you to "Print" **text** and other programming elements to the running log so that you can see what is happening in your program in real time.*

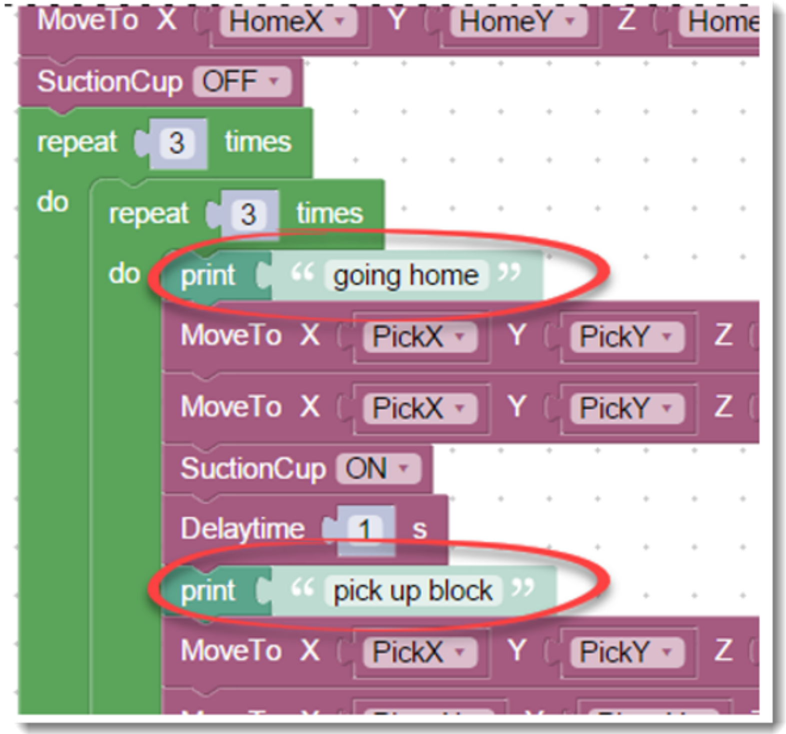
This is a great way to troubleshoot a complex program.



Blockly Commands: Text

Text

*In this example, it will print "**going home**" in the running log while the robot is moving to a home position and then "**pick up block**" when going to the pick position.*

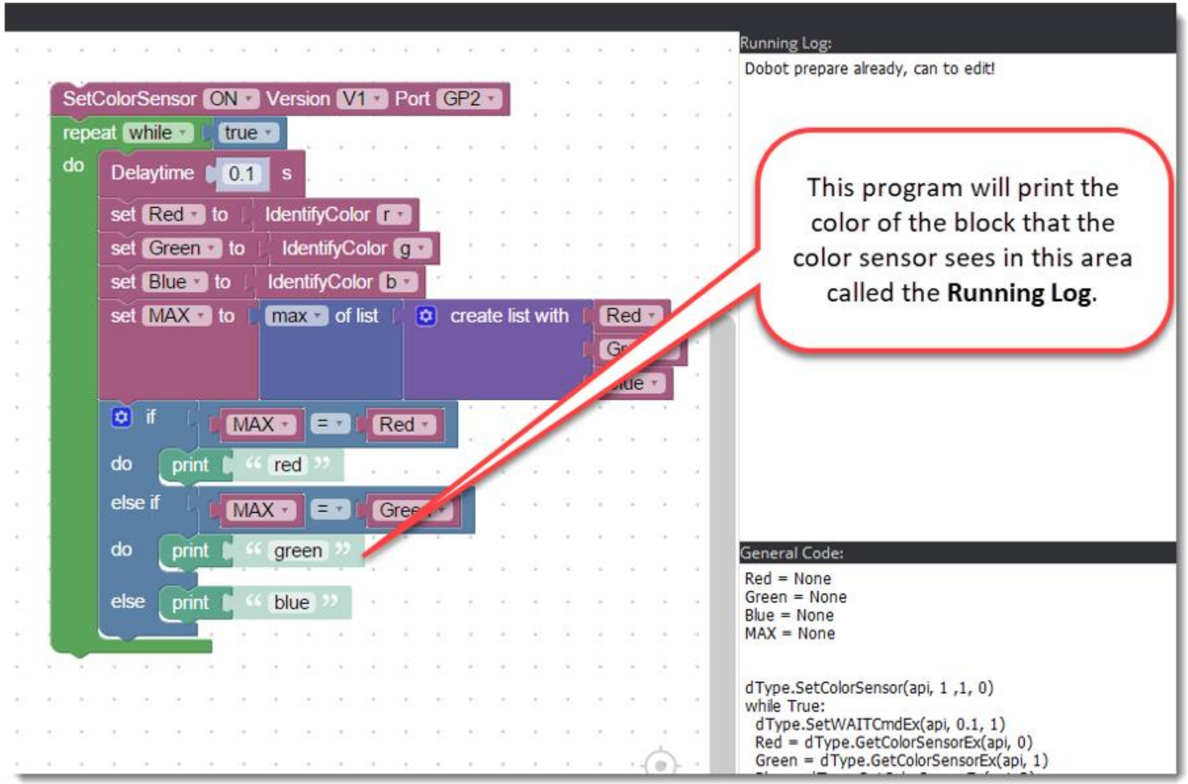


```
MoveTo X HomeX Y HomeY Z Home
SuctionCup OFF
repeat 3 times
do  repeat 3 times
    do  print " going home "
        MoveTo X PickX Y PickY Z
        MoveTo X PickX Y PickY Z
        SuctionCup ON
        Delaytime 1 s
        print " pick up block "
        MoveTo X PickX Y PickY Z
```

Blockly Commands: Text

Text

*The **Running Log** appears to the right of the programming window in Dobot Studio and runs constantly.*



```
SetColorSensor ON Version V1 Port GP2
repeat while true
do  Delaytime 0.1 s
    set Red to IdentifyColor r
    set Green to IdentifyColor g
    set Blue to IdentifyColor b
    set MAX to max of list create list with Red
                                            Gr
                                            ue
    if MAX = Red
    do print "red"
    else if MAX = Gree
    do print "green"
    else print "blue"
```

Running Log:

```
Dobot prepare already, can to edit!
```

This program will print the color of the block that the color sensor sees in this area called the **Running Log**.

General Code:

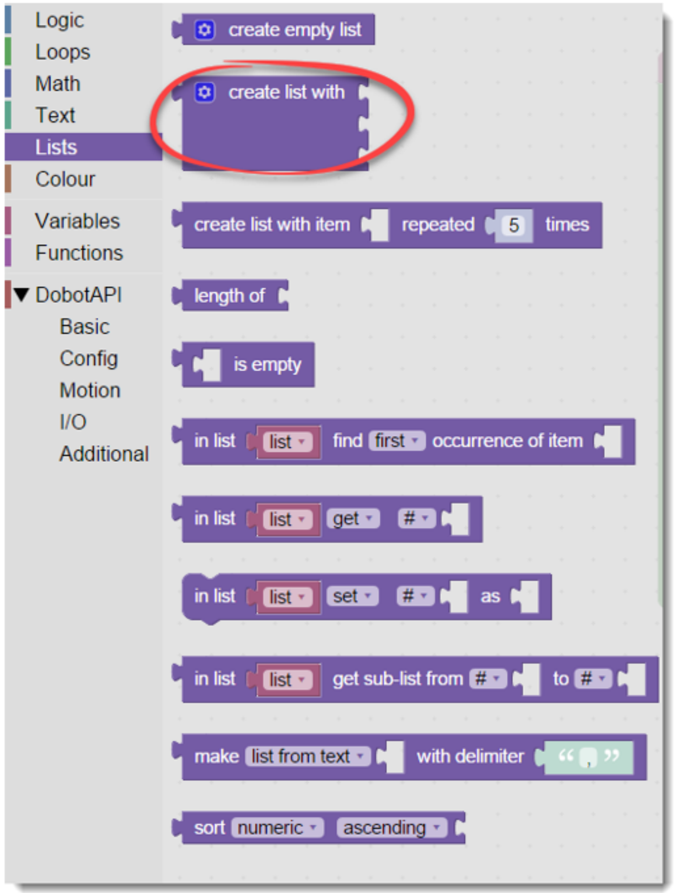
```
Red = None
Green = None
Blue = None
MAX = None


dType.SetColorSensor(api, 1 ,1, 0)
while True:
  dType.SetWAITCmdEx(api, 0.1, 1)
  Red = dType.GetColorSensorEx(api, 0)
  Green = dType.GetColorSensorEx(api, 1)
```

Blockly Commands: Lists

Lists

*Logic commands that will allow you to build and deal with lists. A **list** is an ordered set of items that can be used by the rest of your program.*

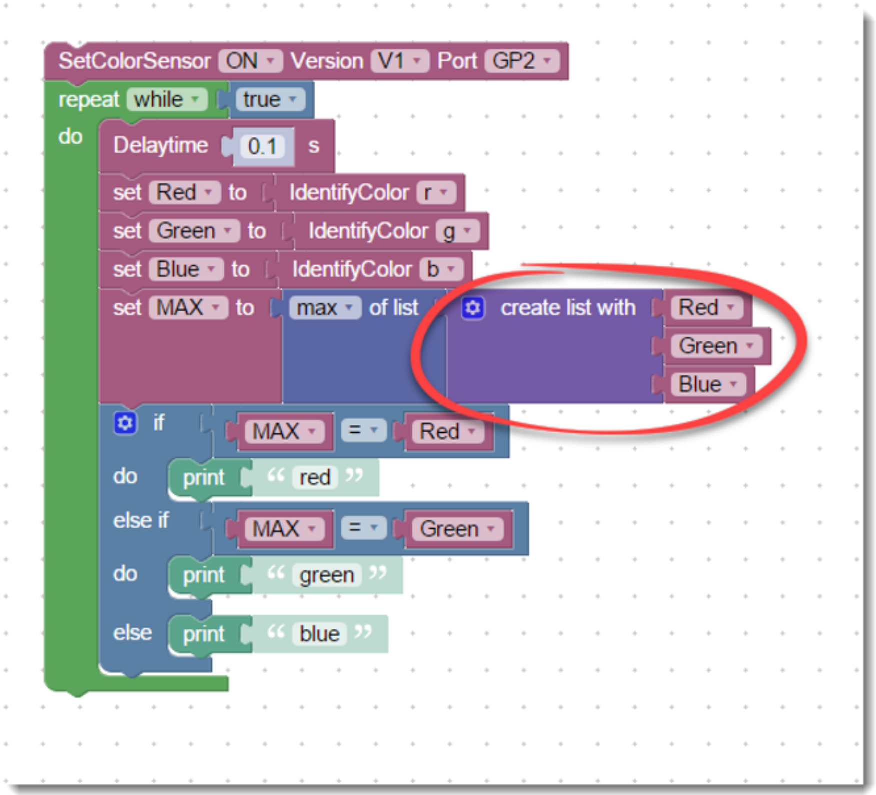
One of the important tasks that these blocks will allow you to do is to build a list when sorting colors with a color sensor.



Blockly Commands: Lists

Lists

*In this example program a **list** is used when a color sensor checks to see what color a block is. It then **prints** to to the running log the value: Red, Green, or Blue.*



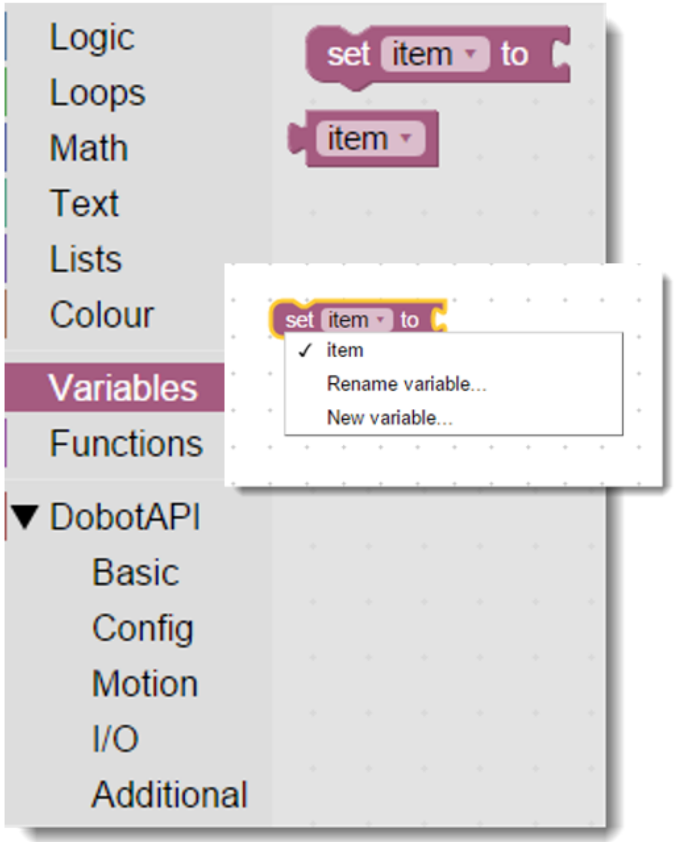
```
SetColorSensor ON Version V1 Port GP2
repeat while true
do
    Delaytime 0.1 s
    set Red to IdentifyColor r
    set Green to IdentifyColor g
    set Blue to IdentifyColor b
    set MAX to max of list create list with Red
                                            Green
                                            Blue
    if MAX = Red
    do print " red "
    else if MAX = Green
    do print " green "
    else print " blue "
```

Blockly Commands: Variables

Variables

Logic commands that will allow you to set variables in a program and call them out for use when needed.

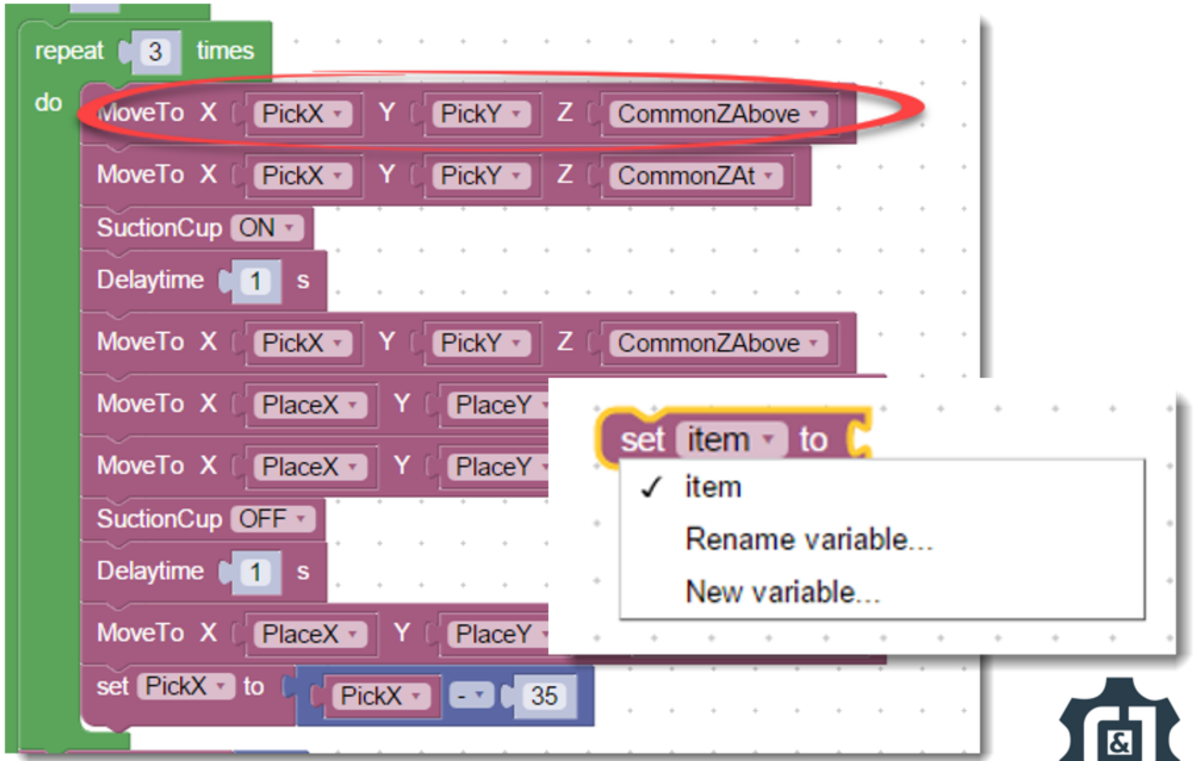
*In this example, you can click on "item", make a new **variable**, then use that **variable** in multiple places in the program. This makes it easy to make a change in a program. Change the **variable** once, and it changes it everywhere.*



Blockly Commands: Variables

Variables

*In this example program you can can see that **variables** were used to set the Pick X, Y, and Z values as well as the Place X, Y, and Z values.*



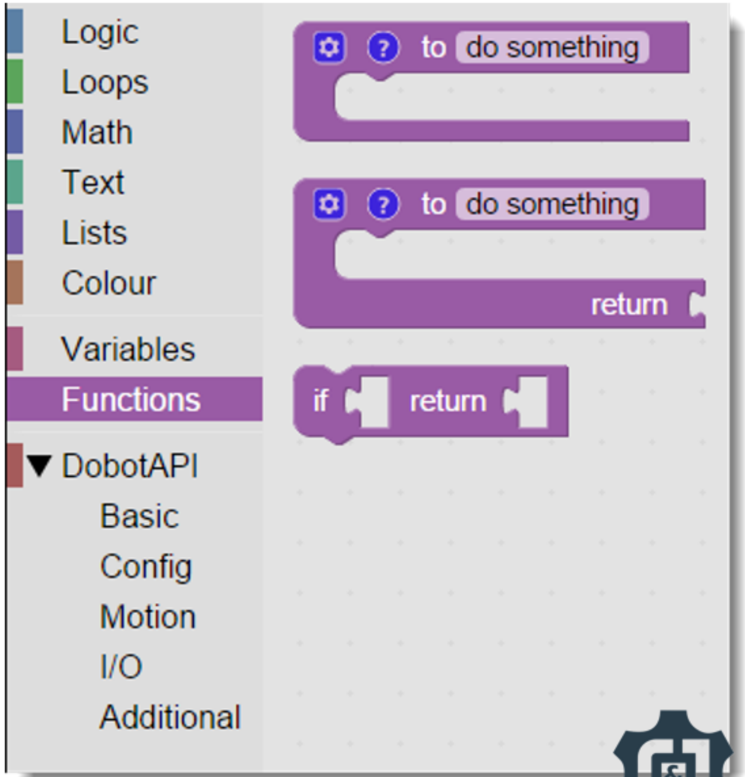
```
repeat 3 times
do
  MoveTo X PickX Y PickY Z CommonZAbove
  MoveTo X PickX Y PickY Z CommonZAt
  SuctionCup ON
  Delaytime 1 s
  MoveTo X PickX Y PickY Z CommonZAbove
  MoveTo X PlaceX Y PlaceY
  MoveTo X PlaceX Y PlaceY
  SuctionCup OFF
  Delaytime 1 s
  MoveTo X PlaceX Y PlaceY
  set PickX to PickX - 35
```

Blockly Commands: Functions

Functions

Logic commands that allow you to name a section of a program and then use it repeatedly, simplifying it for the programmer and end user. ***Functions*** *can also be called voids.*

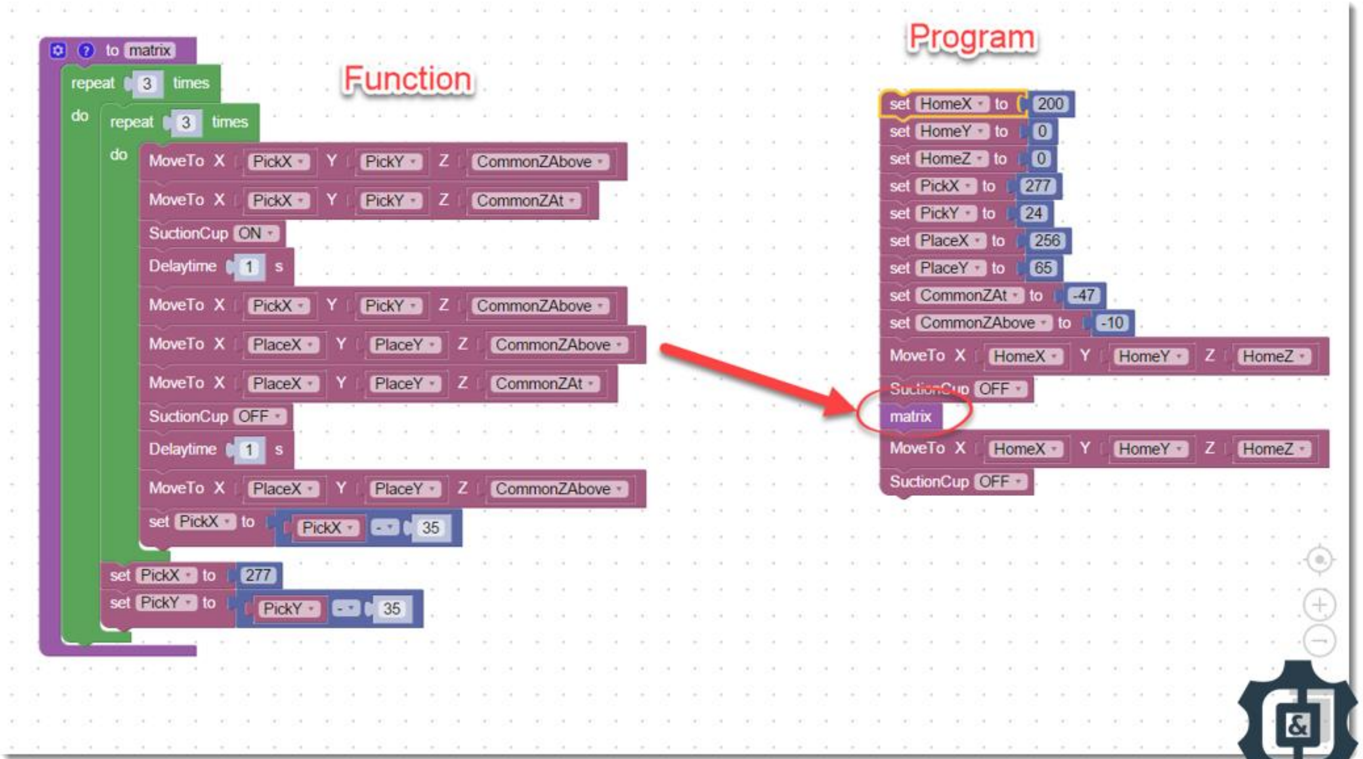
Click on "do something", name your function, then drag what you want it to do into the block.



Blockly Commands: Functions

Functions

*In this example program, the **Function** is on the left, and the program where it is called out is on the right.*



```
to matrix
  repeat 3 times
  do
    repeat 3 times
    do
      MoveTo X PickX Y PickY Z CommonZAbove
      MoveTo X PickX Y PickY Z CommonZAt
      SuctionCup ON
      Delaytime 1 s
      MoveTo X PickX Y PickY Z CommonZAbove
      MoveTo X PlaceX Y PlaceY Z CommonZAbove
      MoveTo X PlaceX Y PlaceY Z CommonZAt
      SuctionCup OFF
      Delaytime 1 s
      MoveTo X PlaceX Y PlaceY Z CommonZAbove
      set PickX to PickX - 35
    set PickX to 277
    set PickY to PickY - 35
```

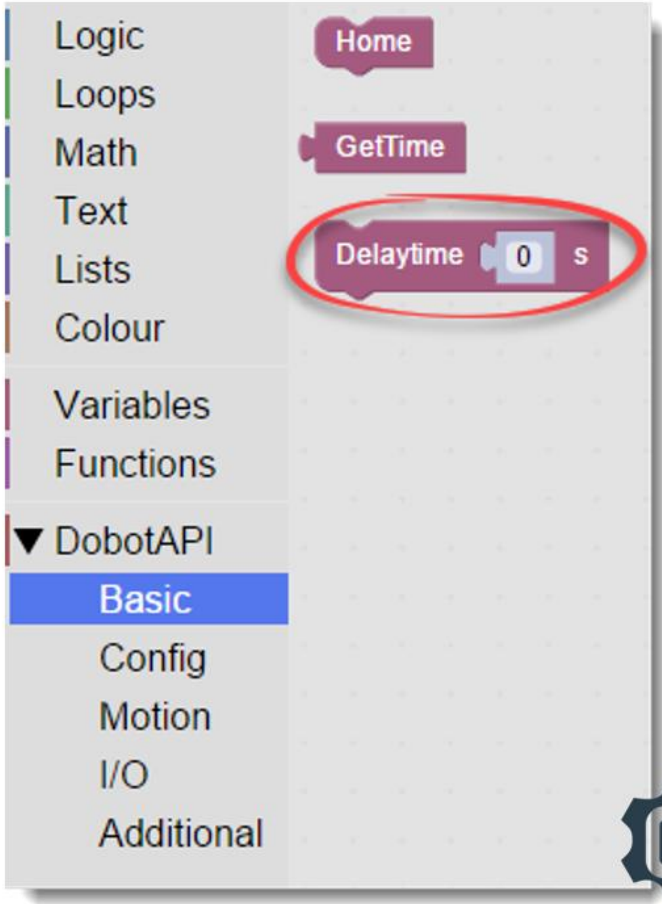
```
set HomeX to 200
set HomeY to 0
set HomeZ to 0
set PickX to 277
set PickY to 24
set PlaceX to 256
set PlaceY to 65
set CommonZAt to -47
set CommonZAbove to -10
MoveTo X HomeX Y HomeY Z HomeZ
SuctionCup OFF
matrix
MoveTo X HomeX Y HomeY Z HomeZ
SuctionCup OFF
```

Blockly Commands: Basic

Dobot API - Basic

*The most important command in this section is the **Delaytime** command.*

This allows you to set a delay time within a program between steps when timing is critical. It is measured in seconds and decimal seconds can be used.

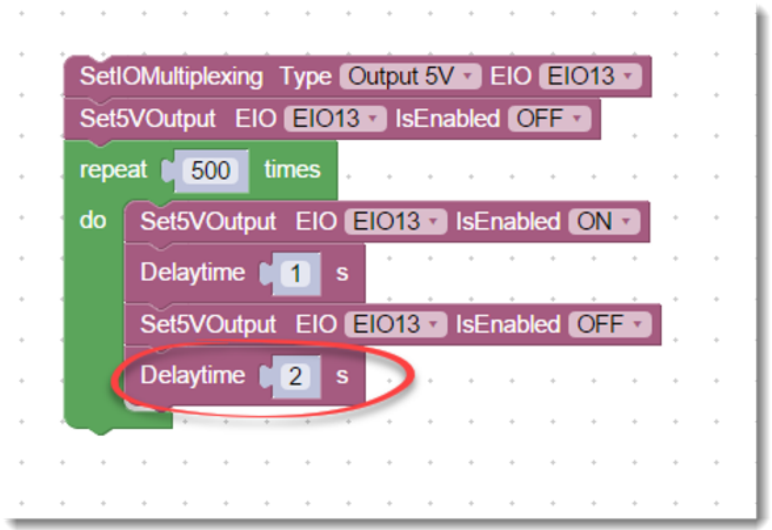


Blockly Commands: Basic

Dobot API - Basic

In this example program a robot is sending a signal to another device to test the connection. It is set to do this 500 times.

The ***Delaytime*** *after the output is turned off must be greater than 1 second otherwise the other machine does not have time to complete its process.*



```
SetIOMultiplexing Type Output 5V EIO EIO13
Set5VOutput EIO EIO13 IsEnabled OFF
repeat 500 times
do  Set5VOutput EIO EIO13 IsEnabled ON
    Delaytime 1 s
    Set5VOutput EIO EIO13 IsEnabled OFF
    Delaytime 2 s
```

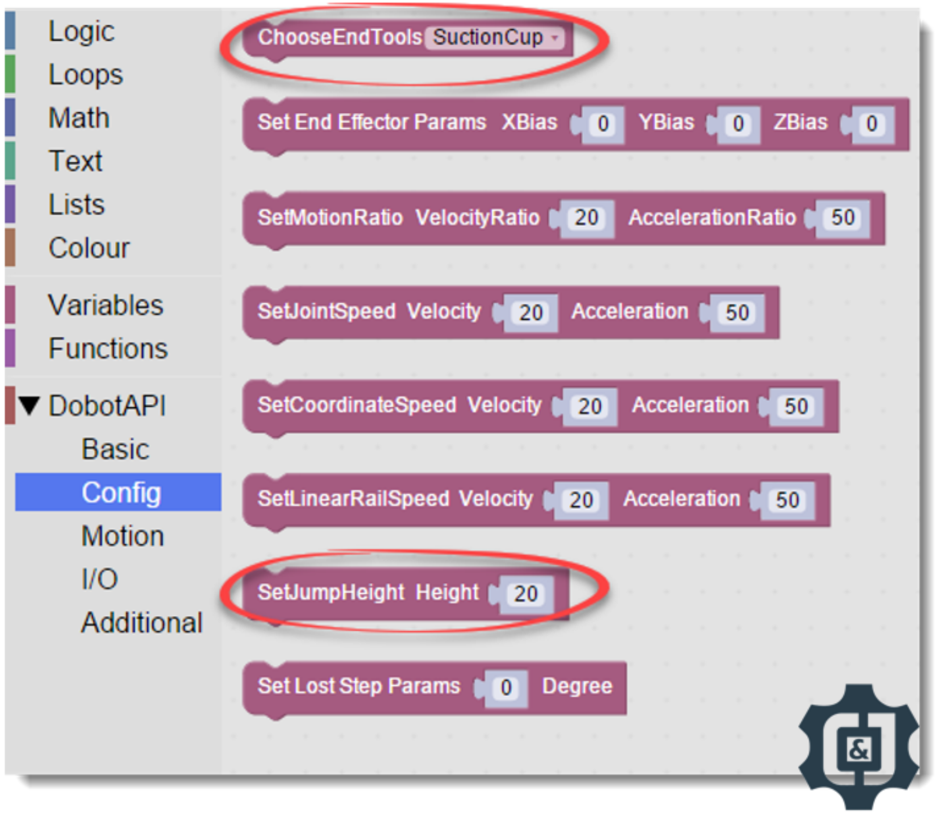
Blockly Commands: Config

Dobot API - Config

Logic commands that allow to configure certain items in the program.

The two most important tasks are the

- ***ChooseEndTools** – allows you to choose your end effector*
- ***SetJumpHeight**-allows you to choose the height of a jump move.*



ChooseEndTools SuctionCup
Set End Effector Params XBias 0 YBias 0 ZBias 0
SetMotionRatio VelocityRatio 20 AccelerationRatio 50
SetJointSpeed Velocity 20 Acceleration 50
SetCoordinateSpeed Velocity 20 Acceleration 50
SetLinearRailSpeed Velocity 20 Acceleration 50
SetJumpHeight Height 20
Set Lost Step Params 0 Degree

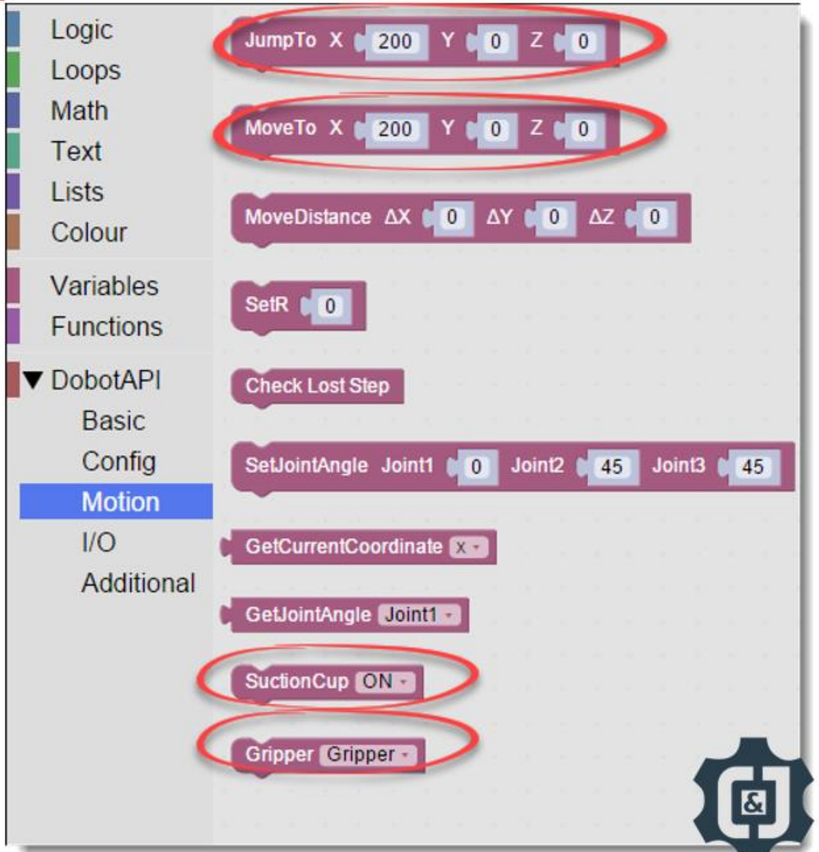
Blockly Commands: Motion

Dobot API - Motion

Logic commands that control the motion of the robot arm.

Some of the important tasks that these will allow you to do are:

- ***JumpTo** a cartesian coordinate*
- ***MoveTo** a cartesian coordinate*
- *Turn the Suction Cup on or off*
- *Open and close the gripper*

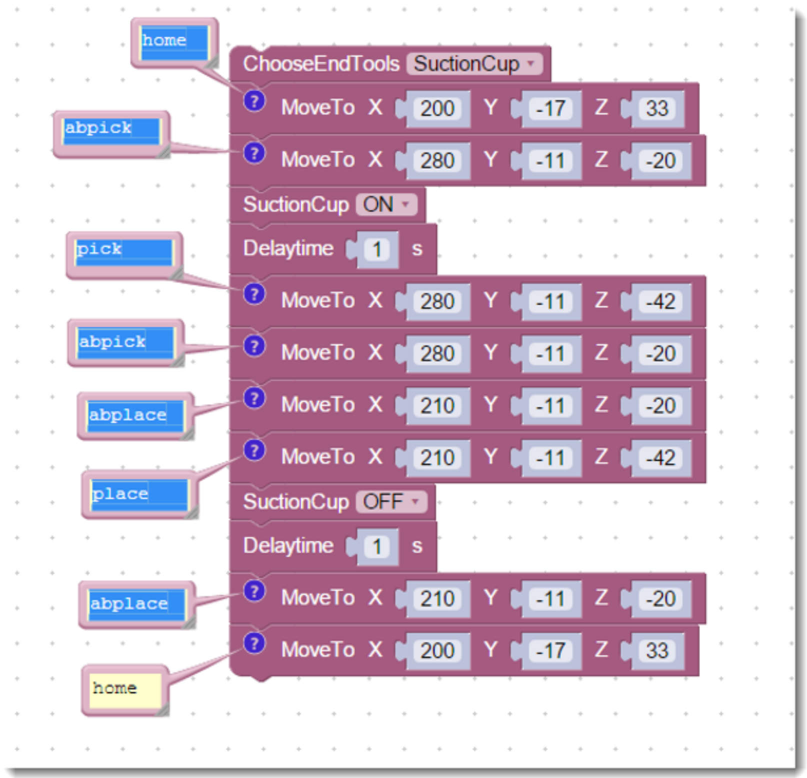


Blockly Commands: Motion

Dobot API - Motion

This example program completes a pick and place of an object in a workcell.

*Notice how **MoveTo** was used to move the robot between points.*



```
ChooseEndTools SuctionCup
MoveTo X 200 Y -17 Z 33        home
MoveTo X 280 Y -11 Z -20       abpick
SuctionCup ON
Delaytime 1 s
MoveTo X 280 Y -11 Z -42       pick
MoveTo X 280 Y -11 Z -20       abpick
MoveTo X 210 Y -11 Z -20       abplace
MoveTo X 210 Y -11 Z -42       place
SuctionCup OFF
Delaytime 1 s
MoveTo X 210 Y -11 Z -20       abplace
MoveTo X 200 Y -17 Z 33        home
```

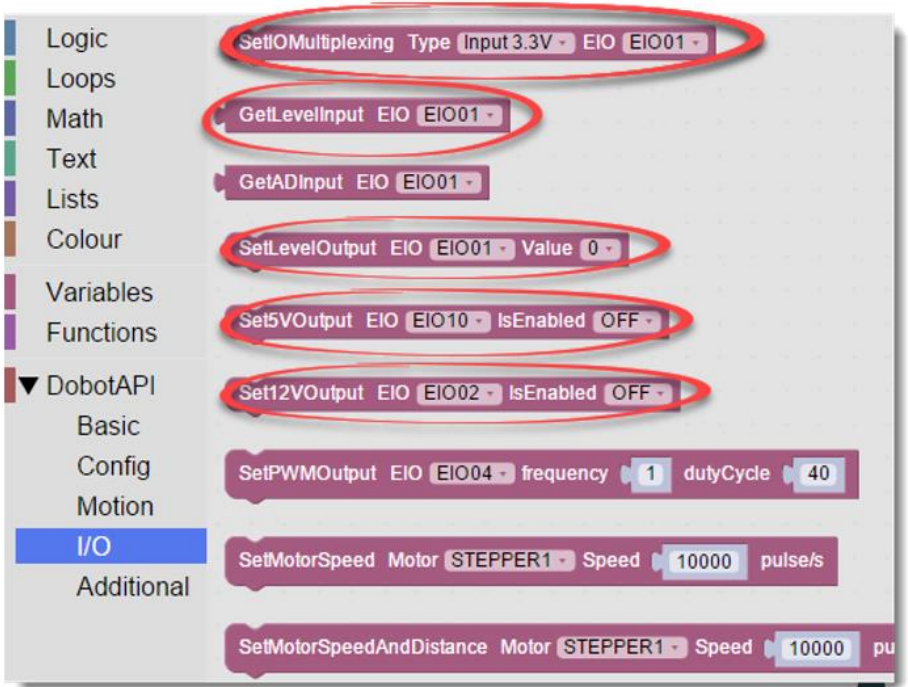
Blockly Commands: I/O

Dobot API – I/O

Logic commands that deal with Inputs and outputs

Some of the important ones are:

- *Set an input type and choose the port*
- *Check the level of an input*
- *Set a 3.3, 5, or 12v output*



SetIOMultiplexing Type Input 3.3V EIO EIO01

GetLevelInput EIO EIO01

GetADInput EIO EIO01

SetLevelOutput EIO EIO01 Value 0

Set5VOutput EIO EIO10 IsEnabled OFF

Set12VOutput EIO EIO02 IsEnabled OFF

SetPWMOutput EIO EIO04 frequency 1 dutyCycle 40

SetMotorSpeed Motor STEPPER1 Speed 10000 pulse/s

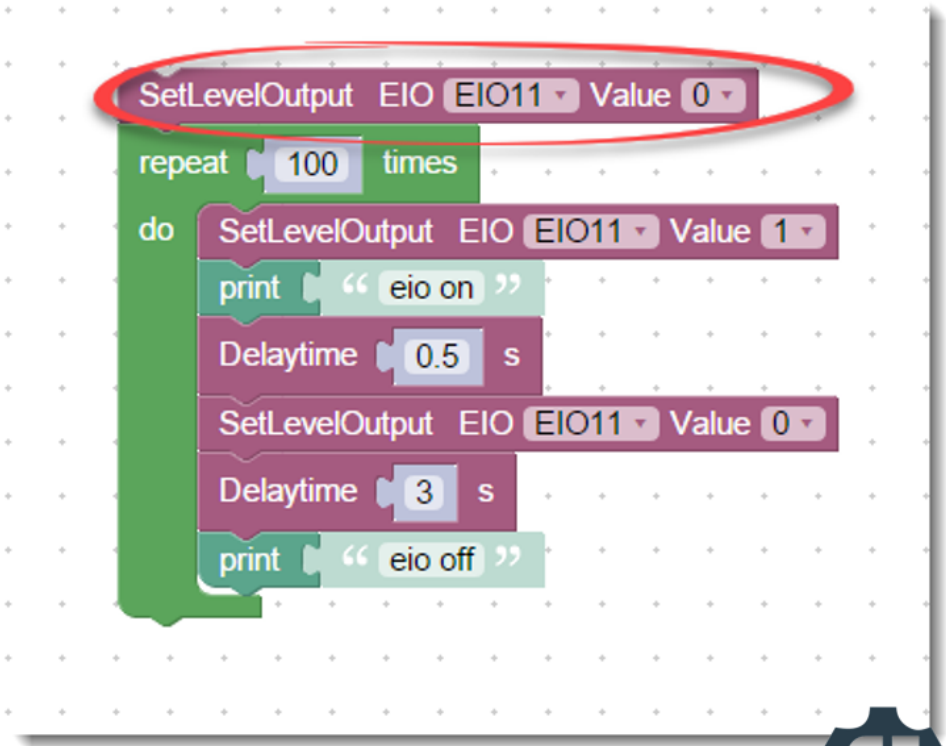
SetMotorSpeedAndDistance Motor STEPPER1 Speed 10000 pu

Blockly Commands: I/O

Dobot API – SetLevelOutput

This example program was written to test an output by turning it on and off 100 times

Notice how the value of the output was set at 0 to start, then in the loop it uses a "1" to turn it on and "0" to turn it off.



```
SetLevelOutput EIO EIO11 Value 0
repeat 100 times
do  SetLevelOutput EIO EIO11 Value 1
    print " eio on "
    Delaytime 0.5 s
    SetLevelOutput EIO EIO11 Value 0
    Delaytime 3 s
    print " eio off "
```

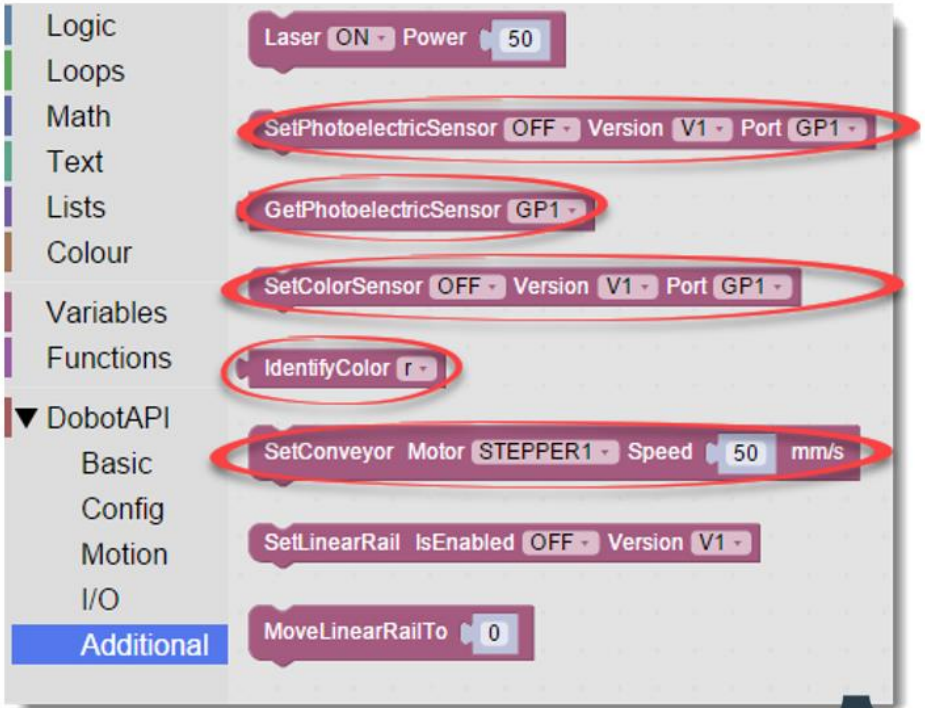
Blockly Commands: Additional

Dobot API – Additional

Logic commands that deal mainly with sensors and outputs built specifically for the Magician.

Some of the important ones are:

- ***SetPhotoSensor**, **GetPhotoSensor***
- ***SetColorSensor***
- ***IdentifyColor***
- ***SetConveyor** Motor port & Speed*



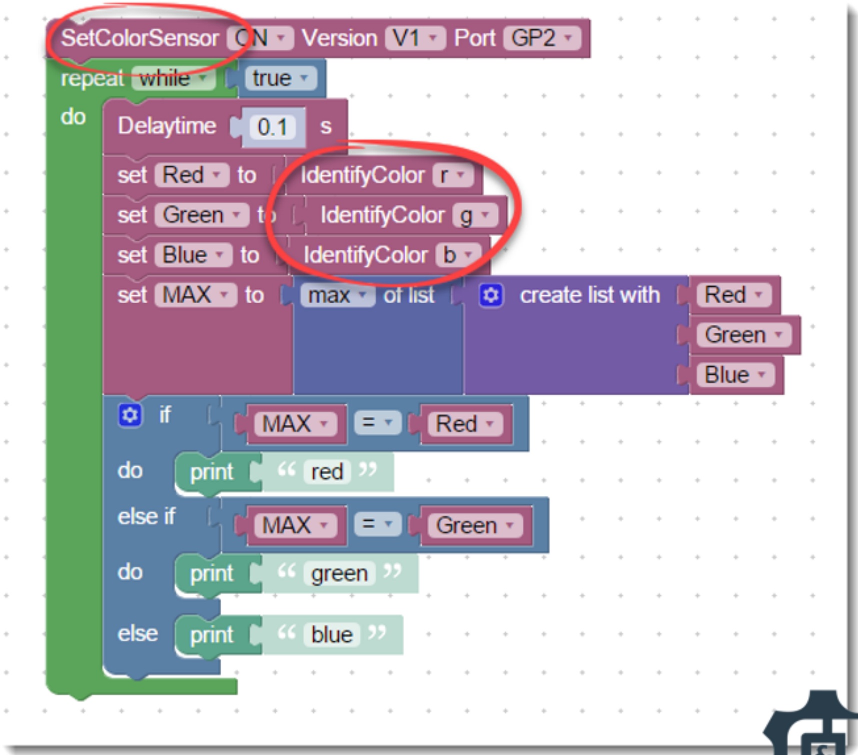
Laser ON Power 50
SetPhotoelectricSensor OFF Version V1 Port GP1
GetPhotoelectricSensor GP1
SetColorSensor OFF Version V1 Port GP1
IdentifyColor r
SetConveyor Motor STEPPER1 Speed 50 mm/s
SetLinearRail IsEnabled OFF Version V1
MoveLinearRailTo 0

Blockly Commands: Additional

Dobot API – Additional

*In this example program **SetColorSensor** tells the program to turn it on and that you are using a V1 color sensor on port GP2.*

*The **IdentifyColor** block is then used to identify the color of an object in front of the sensor where variables are used to name the values "Red", "Blue", and "Green", then the names are printed to the Running Log.*



```
SetColorSensor ON Version V1 Port GP2
repeat while true
do
    Delaytime 0.1 s
    set Red to IdentifyColor r
    set Green to IdentifyColor g
    set Blue to IdentifyColor b
    set MAX to max of list create list with Red
                                            Green
                                            Blue
    if MAX = Red
    do    print "red"
    else if MAX = Green
    do    print "green"
    else  print "blue"
```

Resources