

What is Coding?

- **Code**
 - *The instructions in a computer program.*
 - *Instructions written by a programmer in a programming language are often called source code.*

What is Coding?

- **Program**
 - *The process of developing and implementing various sets of instructions to enable a computer to do a certain task.*
 - *These instructions are considered computer programs and help the computer to operate smoothly.*
 - *The language used to program computers is not understood by an untrained eye.*

What is Coding?

- **Algorithm**
 - *a process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer.*

Coding is...



Telling the
computer exactly
what to do... by
giving it step by
step instructions
to follow

WHAT IS CODING?

- Coding is about logical reasoning skills, creative problem solving and discovery of new solutions, as well as creating new solutions – innovations.
- It is about breaking bigger problems into smaller ones to be able to solve the bigger problems and create bigger solutions.
- Coding is commanding computer what to do; programming and instructing, so that it knows what, how and when to do the things needed.

Types of Coding

- *Draw Programming (beginner)*
- *Block Programming (intermediate)*
- *Text Based Coding (Advanced)*

Draw Programing (beginner)

- *The simpler method of programming.*
- *Users draw lines to program their robot and can modify speed and colour.*

Ozobot (Drawing based coding)



Ozobot (Drawing based coding)



Color code reference chart

OzoCodes

For best results, use OzoCodes on black lines and sections of straight lines.

ozobot

SPEED



SNAKE DANCE



FAST



SLOW



TURN90



CRUISE



NITRO BOOST

DIRECTION



GO LEFT



GO STRAIGHT



GO RIGHT



JUMP LEFT



JUMP STRAIGHT



JUMP RIGHT



U TURN



U TURN (LINE END)

TIMERS



TIMER ON (30 SEC. TO STOP)



TIMER OFF



PAUSE (2 SEC.)

COOL MOVES



TORNADO



ZIGZAG



SPIN



BACKWALK

WIN/EXITS



WIN/EXIT (PLAY AGAIN)



WIN/EXIT (GAME OVER)

COUNTERS



ENABLE X-ING COUNTER



ENABLE TURN COUNTER



ENABLE PATH COLOR COUNTER



ENABLE OZOPULL COUNTER



OZOPULL +1



OZOPULL -1

Ozobot (Drawing based coding)

Tips: Drawing Codes



X
Codes on
colored lines



X
Different
sizes



X
White
spaces



X
Overlapping
colors



X
Too dark



✓
Codes on
black lines

Ozobot (Drawing based coding)

Tips: Drawing Lines



Too Thin!



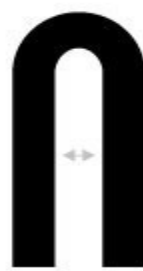
Too Thick!



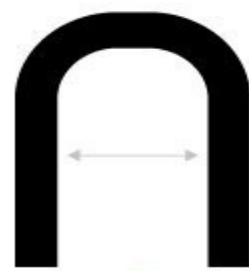
Inconsistent!



Just Right



Too Close!



Just Right



Too Sharp!



Just Right



Just Right

Block Programming (intermediate)

- *Coders can utilise the familiar block-based drag and drop interface to code their robot.*
- *Pre-programmed blocks allow for a wide variety of actions and variables.*

Scratch (block programming/coding)



Scratch (block programming/coding)

Scratch interface showing a project titled "Full 16 Frame Scratch Cat Walk Cycle" by griffpatch. The stage displays a Scratch Cat sprite walking on a grassy field with a blue sky and green trees in the background.

The interface includes a top menu bar (File, Edit, Tips), a toolbar with icons for saving, undo, redo, and help, and a top right area with "Sign in to save" and "Sign in" buttons.

The left sidebar shows the "Sprites" panel with a "New sprite" button and a list of available sprites: Scratch Cat, Tree_3, Glass-Tal..., Glass-Tal..., grass3, grass2, and Tree_2.

The center panel displays the "Scripts" and "Events" block categories. The "Events" category is selected, showing various event triggers like "when green flag clicked", "when space key pressed", "when this sprite clicked", "when backdrop switches to backdrop", "when loudness > 10", "when I receive message1", "broadcast message1", and "broadcast message1 and wait".

The right panel shows the "Scripts" block palette with the following code blocks:

- when green flag clicked
- forever loop containing:
 - wait 0.02 secs
 - next costume
- when this sprite clicked
- set size to 50 %
- go to x: -195 y: -123
- clear
- switch costume to Walk1
- repeat 8 loop containing:
 - stamp
 - change x by 55
 - next costume
- set size to 120 %
- go to x: 0 y: 55

The bottom right corner shows the Scratch Cat sprite's coordinates: x: 5, y: 1.

Sphero EDU app (block programming/coding)

The screenshot displays the Sphero EDU app's block programming interface. At the top, there is a menu icon (three horizontal lines) and a blue 'Run' button with a play icon. On the right side of the top bar, there are three circular icons: a steering wheel, a headset, and a code editor icon. The main workspace is a light blue area where code blocks are assembled. On the left, a vertical sidebar lists various block types: Strobe, Delay, Spin, Strobe, Back LED, Roll, and Back LED. Each block has input fields for values like time (ms) or degrees. A callout bubble with a hand icon and the text 'Drag code here to delete it' is positioned in the upper right of the workspace. At the bottom, a teal bar contains ten icons representing different actions: Roll, Set Speed, Set Heading, Stop, Raw Motor, Spin, Set Color, Back LED, Fade, and Strobe. Below this bar, a row of category labels is visible: Actions, Events, Sensors, Controls, and Operators. A faint watermark 'www.thanksmailcarrier.com' is visible in the bottom right corner of the workspace.

Run

Strobe 1ms 0

Delay 500ms

Spin 0° 0ms

Strobe 0ms 0

Back LED 0

Roll 1000ms 190 181°

Back LED 0

Drag code here to delete it

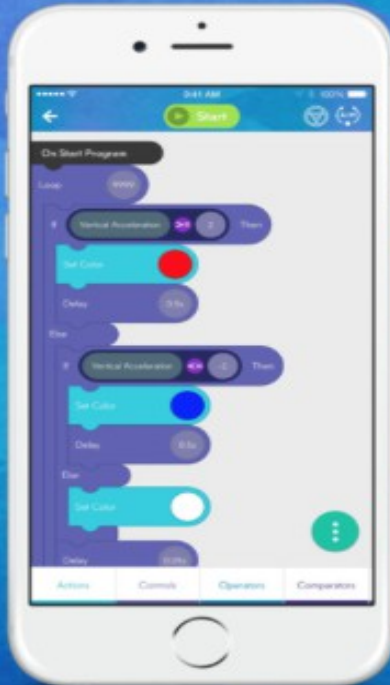
Roll Set Speed Set Heading Stop Raw Motor Spin Set Color Back LED Fade Strobe

Actions Events Sensors Controls Operators

www.thanksmailcarrier.com

Sphero EDU app (block programming/coding)

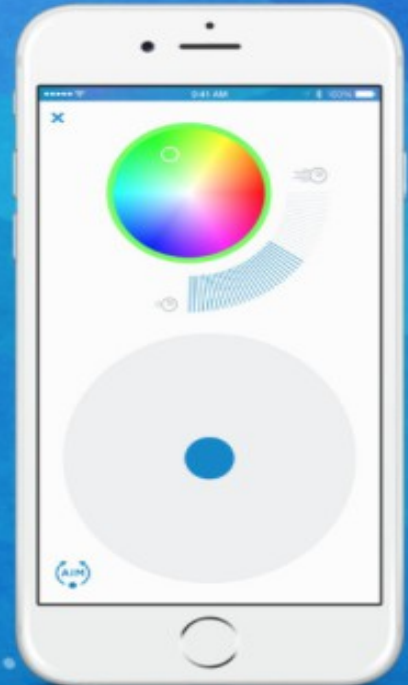
Block-Based Programming



Text-Based Code Viewer



Take A Drive



Text Based Coding (advanced)

- *Users can program with a text editor to write custom syntax.*
- *An integrated "wiki" will support users graduating from blocks to text; the same type of tool real developers use.*

HTML (Text based code)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>*</title>

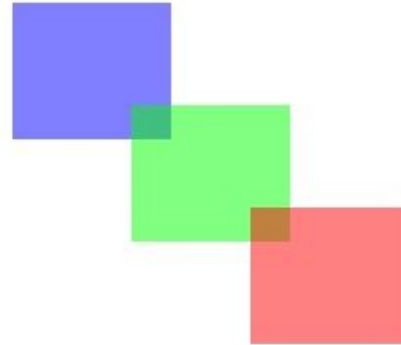
<style type="text/css">
body{
    font-family:"Lucida Grande", "Lucida Sans Unicode", Verdana, Arial, Helvetica, sans-serif;
    font-size:12px;
    background-color:#585858;
}
p, h1, form, button{border:0; margin:0; padding:0;}
.spacer{clear:both; height:1px;}
/* ----- Form ----- */
.myform{
    margin:0 auto;
    width:400px;
    height:190px;
    padding:12px;
    margin-left: auto ;
    margin-right: auto ;
    margin-top: auto;
}
/* ----- basic ----- */
#basic{
    border:solid 2px #DEDEDE;
}
#basic h1 {
    font-size:14px;
    font-weight:bold;
    margin-bottom:8px;
}
#basic p{
    font-size:11px;
    color:#666666;
    margin-bottom:20px;
    border-bottom:solid 1px #dedede;
    padding-bottom:10px;
}
#basic label{
    display:block;
    font-weight:bold;
    text-align:right;
    width:140px;
    float:left;
}
```

Code Monster (Text based code)



A big `clearRect()` erases everything. Uncomment the big `clearRect()` (remove the `//`) and see what happens.

```
1 var size = 80;
2 var offset = 20;
3 var w = c.canvas.width;
4 var h = c.canvas.height;
5
6 c.fillStyle = "rgba(0, 0, 255, 0.5)";
7 c.fillRect(offset, offset, size, size);
8 offset = offset + 60;
9 c.fillStyle = "rgba(0, 255, 0, 0.5)";
10 c.fillRect(offset, offset, size, size);
11 // c.clearRect(0, 0, w, h);
12 offset = offset + 60;
13 c.fillStyle = "rgba(255, 0, 0, 0.5)";
14 c.fillRect(offset, offset, size, size);
15
```



RESET


Erasing

BACK

Python (Text based code)

hoc.codehs.com/hoc_editor.html#course=16#module=103#item=162

251Tech Animoto AR Atomic BigHugeLabs BrainPOP Diigo EasyTech Ed Tech Glogster Google Apps Primary Google Apps Secon... Google

 HOUR OF CODE with CodeHS

STUDENT SIGN UP

TEACHER SIGN UP

★ Challenge: Draw Something

DIRECTIONS

Overview

Write a program to draw something fun or creative on the screen.

Requirements:

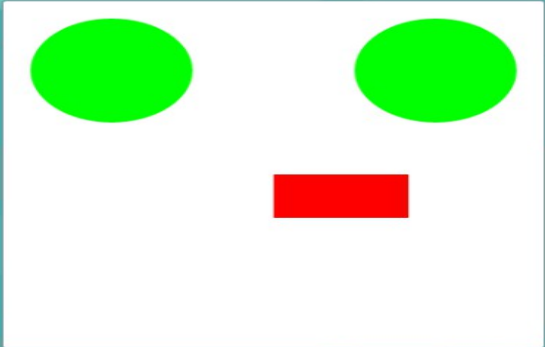
- You must use at least one line, at least one rectangle, and at least one circle.
- You must use shapes of at least two different colors.
- You must use shapes of at least two different sizes.
- You must have at least five shapes total.
- You must leave a comment at the top of your program explaining what you are trying to do.
- Otherwise, there are no other requirements!

RUN

RESET

CHECK + CONTINUE

```
1 function start(){
2   var rect = new Rectangle(100, 50);
3   rect.setPosition(getWidth() / 2, getHei
4   rect.setColor(Color.red);
5   add(rect);
6
7   var circ = new Circle(60);
8   circ.setPosition(80 , 80);
9   circ.setColor(Color.green);
10  add(circ);
11
12  var circ = new Circle(60);
13  circ.setPosition(getHeight() - 80 , 80);
14  circ.setColor(Color.green);
15  add(circ);
16 }
```



90% Complete

Python (Text based code)

IF YOU WANT TO:	USE THIS CODE:
make text bold	<code>text</code>
make text italic	<code><i>text</i></code> OR <code>text</code>
underline text	<code><u>text</u></code>
strikethrough text	<code>text</code>
force an extra space	<code>&nbsp;</code>
add a line break	<code>
</code>
make a new paragraph	<code><p></code>
align text within a paragraph	<code><p align="center"></code> (or "left", "right")
insert an image	<code></code>
adjust image size	<code></code>
insert a link	<code>link</code>
direct link to a new tab or window	<code>link</code>
insert a button	<code></code>
insert a table (for archive pages)	<code><table><tbody>all table content</tbody></table></code>
insert a new row within your table	<code><tr>row content</tr></code>
insert a new box within a row of your table	<code><td>box content</td></code>



Everybody in this country should
learn how to program a
computer...

*...because it teaches you how to
think.*

Steve Jobs

It would be wonderful if every kid wrote computer programs and understood how computers work. It would certainly make you a better thinker.

Bill Gates





In fifteen years we'll be teaching programming just like reading and writing. We'll be looking back and wondering why we didn't do it sooner.

Mark Zuckerberg