Coding Basics 1

Code a Portrait

For Step One: Use Functions to Create a Self-Portrait

What are functions?

Functions are a common type of instruction in programming that tell a computer to perform a certain task. For example, look at this function:

```
drawNose();
```

This function would tell a computer to draw a nose — perhaps on a web site page.

In coding, using a function is calling a function, or invoking a function.

In order for a function to be called, it must be previously defined, or declared. That means that once you have created a function, or defined, a function, you can use it, or call it, again and again in your code. The rules for writing code are called *syntax*.

For example, to call a function with Javascript, you would write it like this:

```
theNameOfTheFunction();
```

Here are the rules for calling (or writing) a function in JavaScript:

- A function starts with a name, which can't have spaces in it, or use other special characters (except underscores '_').
- The name can include numbers — as long as they're not at the beginning.
- The name is directly followed by parentheses '()'. The parentheses tell JavaScript to run the function.
- A semicolon shows that the function has ended, just as a period ends a sentence in English. A semicolon looks like this: ';'

Here's an example of a function call that is correctly written:

```
say_promise();
```

But none of these would work:

```
1st_task();
make cake;
do-good ();
makeCake()again;
```

List of simple functions: Use these functions to....	

Choose the color of the drawing	`useBlue()`
	`useYellow()`
	`useBlack()`
	`useRed()`
	`useGreen()`
	`useBrown()`
	`usePink()`

Draw the shape of the face	`drawRoundFace()`
	`drawOvalFace()`
	`drawSquareFace()`

Draw the eyes	`drawEye()`
	`drawWideEye()`
	`drawWink()`

Draw facial features	`drawNose()` `drawEar()`

Draw the mouth	`drawMouthClosed()`
	`drawMouthOpen()`
	`drawSmile()`

Draw the hair	`drawHair()`
	`drawShortHair()`
	`drawLongHair()`

For Step Two: Write Code to Create a Portrait

Arguments make functions more specific.

Let's say that we want to make a chocolate cake. We could write a makeCake(); function. That would be an instruction to make a cake — but it doesn't say what flavor of cake we want.

We only have to add one argument to specify which type of cake we want. We could code it this way:

```
makeCake("chocolate");
```

The makeCake function reads the argument and creates a chocolate cake.

Sometimes a function needs several arguments to work.

For example, let's look at this function:

```
drawEye();
```

To draw a green eye that appears in a certain position on a computer screen, we would need to add three arguments:

1. a color
2. an x position
3. a y position.

So we would add those three arguments to the drawEye(); function and it would look like this:

```
drawEye("green", 5, 10);
```

This would draw a green eye, and place it at the (5, 10) coordinates.

List of functions with arguments: Use these functions to...	
Choose the color of the drawing	`useColor(nameOfColor);` Replace nameOfColor with whatever color you wish to use in quotes, e.g. “blue”, “pink”, “lilac”.
Draw the shape of the face	`drawFace(shape, x, y);` Replace shape with the face shape of your choice, e.g. “heart”, “oval”, “round”. Replace x and y with the locations of the center of the face shape.
Draw the eyes	`drawEye(color, x, y);` Replace x and y with the locations of the center of one eye. To draw the other eye, write a second line with the same function, but use the coordinates corresponding to the other eye.

Draw facial features	`drawEar(x, y);`
	Replace x and y with the locations of one ear. Repeat the function with coordinates for the other ear.
	`drawNose(x, y);`
	Replace x and y with the locations of the center of the nose.

Draw the mouth	`drawMouthClosed(x, y);`
	`drawMouthOpen(x, y);`
	`drawSmile(x, y);`

Draw the hair	`drawHair(color, style, x, y);` Replace color and style with colors and a hairstyle. Replace x and y with the location of the hair.

You can use these functions to show a symbolic attribute or add text:
`graphic();` This function takes three arguments: the name of the graphic image in quotation marks followed by the x and y coordinates where it should go. For Example, graphic(“star”, 4, 12); is a function that calls for a star to appear at location (4, 12).
`text(words, x, y );` For the text() function, the first argument inside the parentheses is a word or phrase inside of quotation marks. You can add a location to the word by adding x,y coordinates as the place to start writing. For example, text(“Brilliant”, 5, 10); would place the “B” that starts the word “Brilliant” at (5,10) on the grid.