Simple scene changes

Let's say we want to tell the story of Winston as an illustrated story book, with the user clicking to read the next part of the story. We'll start off with a main scene that just has a title:

```
background(235, 247, 255);
fill(0, 85, 255);
textSize(39);
text("The Story of Winston", 10, height/2);
```

Now, we want it so the user can click to see the first part of the story, Winston's epic birth. To do that, we can define a `mouseClicked` function that will be called whenever the user clicks the mouse, and we can put in the code to draw our second scene there. Note that we have to call `background()` before drawing the second scene, otherwise we'll see both scenes on top of eachother:

```
mouseClicked = function() {
    // Scene 2
    background(173, 239, 255);
    fill(7, 14, 145);
    text("Lil Winston is born!", 10, 100);
    image(getImage("creatures/BabyWinston"), width/2, height/2);
};
```

Try it out below- edit the code for the second scene and notice that it's annoying to edit, because you have to click every time you want to see what your second scene looks like.

```
 mouseClicked = function() {

   // Scene 2

   background(173, 239, 255);

   fill(7, 14, 145);

   text("Lil Winston is born!", 10, 100);

   image(getImage("creatures/BabyWinston"), width/2, height/2);

};



// Scene 1

background(235, 247, 255);

fill(0, 85, 255);

textSize(39);

text("The Story of Winston", 10, height/2);
```

Let's solve that problem first. Yes, you and I can survive with annoying-ness, but we want to all be more productive programmers, and won't we be more productive if we can edit scene 2 and see the results in real-time? In this case, an easy thing to do is to wrap all of the scene 2 code inside a function, call that function from `mouseClicked`, and then call that function when we're debugging.

```
var drawScene2 = function() {
    background(173, 239, 255);
    fill(7, 14, 145);
    text("Lil Winston is born!", 10, 100);
    image(getImage("creatures/BabyWinston"), width/2, height/2);
};

mouseClicked = function() {
    drawScene2();
};

// Scene 1
background(235, 247, 255);
fill(0, 85, 255);
textSize(39);
text("The Story of Winston", 10, height/2);

drawScene2();
```

As long as we've made scene 2 into a function, we may as well make scene 1 into a function as well, just wrapping all that code up and calling it. Now, try this one out below, and see how easy it is to comment out the `drawScene2()` call when you want to edit that code and see it immediately.

```
 var drawScene1 = function() {

   // Scene 1

  background(235, 247, 255);

     fill(0, 85, 255);

    textSize(39);

    text("The Story of Winston", 10, height/2);

};



var drawScene2 = function() {

    background(173, 239, 255);

    fill(7, 14, 145);

    text("Lil Winston is born!", 10, 100);

    image(getImage("creatures/BabyWinston"), width/2, height/2);

};



mouseClicked = function() {

    drawScene2();

 };



drawScene1();
```

```
//drawScene2();
```

Great, so we have a main scene and a second scene. What if we want to display a third scene? Or go back to the first scene once we click on the third? We need to change the logic of inside `mouseClicked` so that it *conditionally chooses* which of the scenes to show, instead of always calling scene 2. That means we need an if statement that will check some sort of condition. Let's think about this in terms of pseudo-code first:

```
// When the user clicks the mouse:
    // if the current scene is the first one, go to the second
    // if the current scene is the second one, go to the third
    // if the current scene is the third one, go back to the first
```

It looks like we need to keep track of the "current scene", and it'd make the most sense to store it as a number. Let's declare a global variable `currentScene` and check it inside `mouseClicked`.

```
var currentScene;

mouseClicked = function() {
    if (currentScene === 1) {
        drawScene2();
    } else if (currentScene === 2) {
        drawScene3();
    } else if (currentScene === 3) {
        drawScene1();
    }
};
```

The conditions look like our pseudo-code, but there's one problem: we never actually set `currentScene` to a value, so those conditions will never be true. We *could* set it inside the if conditions, but it's probably better to set it inside the scene drawing functions themself, so that the variable is set correctly no matter where we call scene drawing functions from.

```
var drawScene1 = function() {
    currentScene = 1;
    ...
};

var drawScene2 = function() {
    currentScene = 2;
    ...
};

var drawScene3 = function() {
    currentScene = 3;
    ...
};
```

We've put it all together in the program below. Click through it and see how it cycles around to the beginning of the story. Try adding a scene four (Winston meeting Oh Noes? Winston meeting Winstonia and moving to Winstonsin?), and spin it off:

```
var currentScene;



var drawScene1 = function() {

  currentScene = 1;

  background(235, 247, 255);
```

```
    fill(0, 85, 255);
    textSize(39);
    text("The Story of Winston", 10, height/2);
};

var drawScene2 = function() {
    currentScene = 2;
    background(173, 239, 255);
    fill(7, 14, 145);
    textSize(39);
    text("Lil Winston is born!", 10, 100);
    image(getImage("creatures/BabyWinston"), width/2, height/2);
};
var drawScene3 = function() {
    currentScene = 3;
    background(173, 239, 255);
    fill(7, 14, 145);
    textSize(39);
    text("Winston grows up!", 10, 100);
    image(getImage("creatures/Winston"), width/2, height/2);
};

mouseClicked = function() {
    if (currentScene === 1) {
        drawScene2();
    } else if (currentScene === 2) {
        drawScene3();
    } else if (currentScene === 3) {
        drawScene1();
    }
```

```
};


drawScene1();

 //drawScene2();
```