Using Hatch to Teach

Teach students basic concepts of coding using the Hatch IDE (Integrated Development Environment).

To access the IDE open using

- app.hatchcoding.com/tryhatch

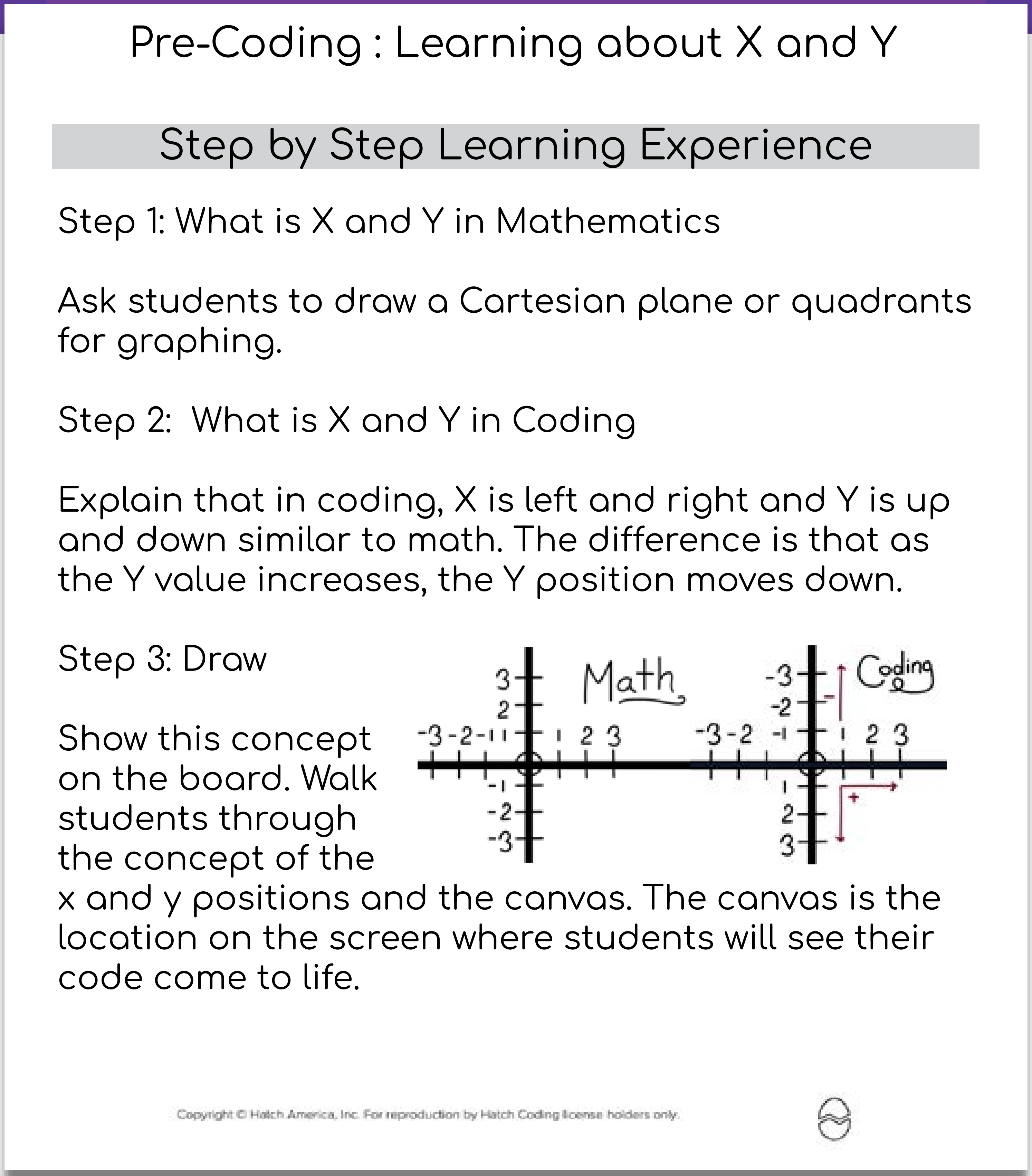
This does not include all functionality and lessons available on the Hatch Platform.

600+ projects are available to students that can be coded in JavaScript and Python.

Challenges are used to extend students and multiple lesson plans are included to cover all aspects of the computer science curriculum.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation



Pre-Coding : Learning about X and Y

Step by Step Learning Experience

Step 1: What is X and Y in Mathematics

Ask students to draw a Cartesian plane or quadrants for graphing.

Step 2: What is X and Y in Coding

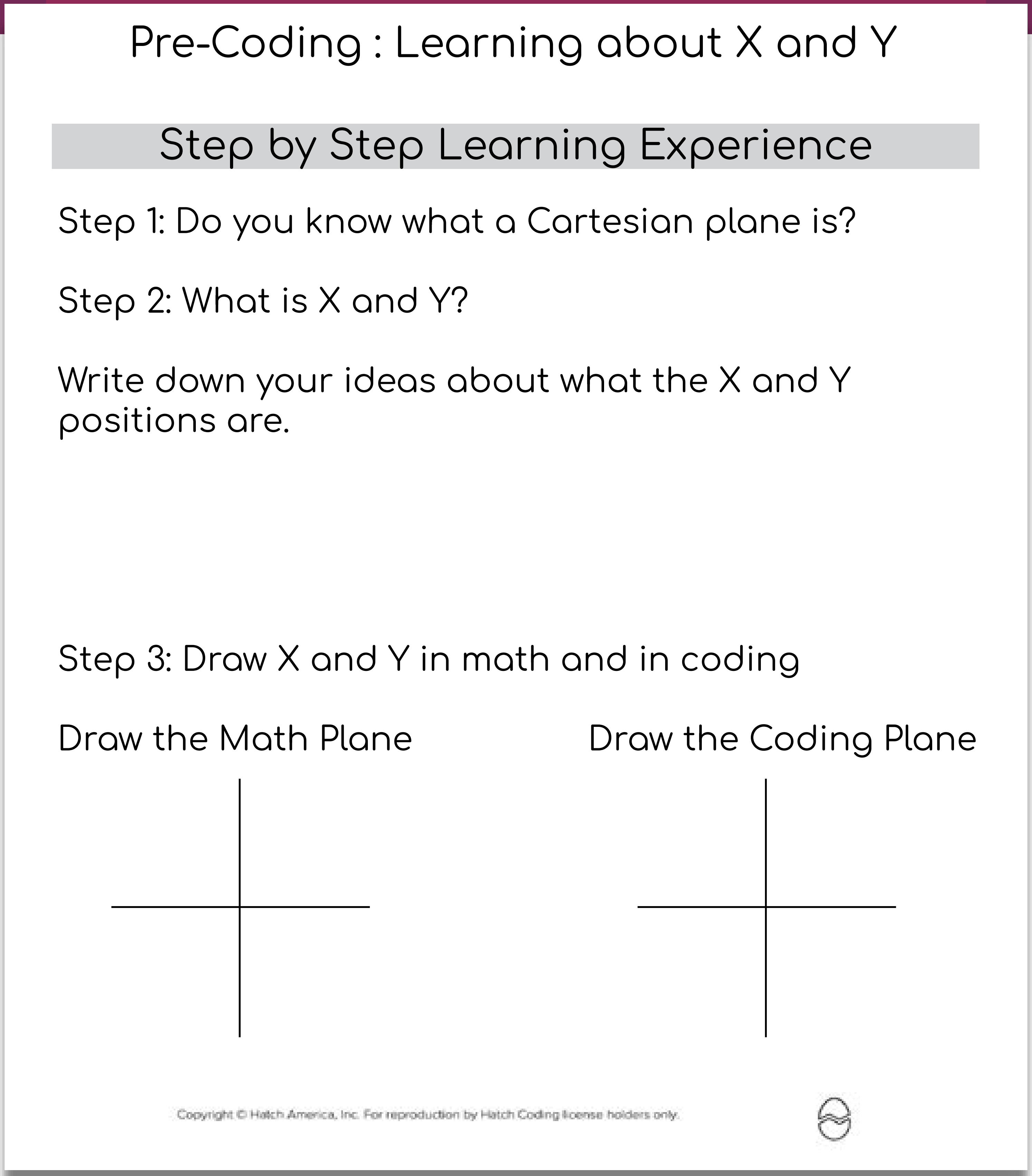
Explain that in coding, X is left and right and Y is up and down similar to math. The difference is that as the Y value increases, the Y position moves down.

Step 3: Draw

Show this concept on the board. Walk students through the concept of the x and y positions and the canvas. The canvas is the location on the screen where students will see their code come to life.

Introduction - Try Hatch - Student Page

Digital Documentation



Pre-Coding : Learning about X and Y

Step by Step Learning Experience

Step 1: Do you know what a Cartesian plane is?

Step 2: What is X and Y?

Write down your ideas about what the X and Y positions are.

Step 3: Draw X and Y in math and in coding

Draw the Math Plane

Draw the Coding Plane

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

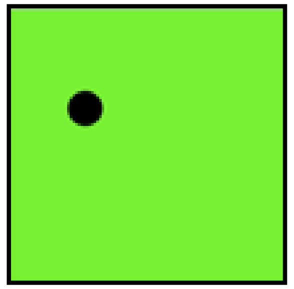
Lesson 1 : Saving code off Platform

Step by Step Learning Experience

Step 1: Google Docs / Cloud documentation

Ask students to open up a cloud based word processor (google docs).

Step 2: Try Hatch



Ask students to open the free IDE using
- app.hatchcoding.com/tryhatch

Step 3: Complete the TWYS

Students will then type into the coding area. Once finished students should be able to interact with a program that looks like the image above.

Step 4: Copy your code into the google doc

Ask students to save their code into a google doc. This google doc should be named “Hold a Ball Code”.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 1 : Saving code off Platform

Step by Step Learning Experience

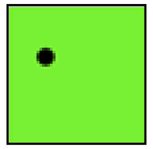
Step 1: Google Docs / Cloud documentation

Open up a cloud based word processor (google docs). You may need to login to your account first.

Step 2: Try Hatch

Open the open access Hatch Coding Area using
- app.hatchcoding.com/tryhatch

Step 3: Complete the TWYS



Type what you see on the left into the middle coding area. When you are done, interact with the program that looks like the image above.

Step 4: Copy your code into the google doc

Save your code into a google doc. This google doc should be named “Hold a Ball Code - Name”.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

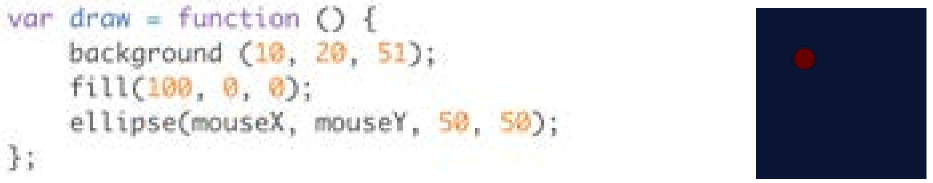
Lesson 2 : Challenge 1

Step by Step Learning Experience

Step 1: Click the challenge tab

Tell students to click the challenge tab. Then read and try to complete the first challenge.

Sample Answer : Numbers in background and fill are changed to complete this challenge.



```
var draw = function () {
    background (10, 20, 51);
    fill(100, 0, 0);
    ellipse(mouseX, mouseY, 50, 50);
};
```

Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc. On the student document their are multiple hints to help students.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 2 : Challenge 1

Step by Step Learning Experience

Step 1: Click the challenge tab

Click the challenge tab. Then read and try to complete the first challenge.

Hints

1. Try to change some of the numbers in your code
2. Go back to the instructions area and click yellow text. This will explain what each of the words mean.
3. Click the explain code button to learn more about how each line of code is working
4. Check out the research manual by pressing “Do more research” in the yellow text area or above the coding area.

Step 2: Save your first challenge in your google doc.

Open up your google doc and save your updated code under challenge 1.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

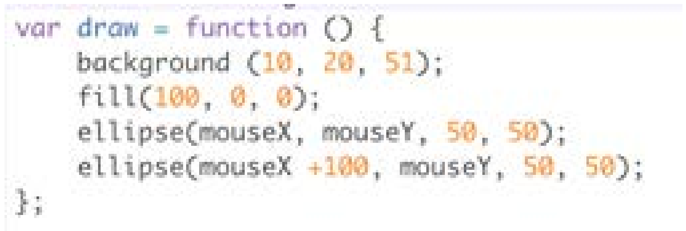
Lesson 3 : Challenge 2

Step by Step Learning Experience

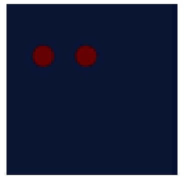
Step 1: Click the challenge tab

Tell students to click the challenge tab. Then read and try to complete the second challenge.

Sample Answer : Students will need to add an additional ellipse and change the location of the first or second mouseX or mouseY; example added 100



```
var draw = function () {
    background (10, 20, 51);
    fill(100, 0, 0);
    ellipse(mouseX, mouseY, 50, 50);
    ellipse(mouseX +100, mouseY, 50, 50);
};
```



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc. On the student document their are multiple hints to help students.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 3 : Challenge 2

Step by Step Learning Experience

Step 1: Click the challenge tab

Click the challenge tab. Then read and try to complete the second challenge.

Hints

1. Use the hints from last challenge to help you as well.
2. What makes the circle appear?
3. What determines the location of the ball?
4. What happens if two objects are in the same location?
5. How do you add or subtract numbers in math?
6. How do you add or subtract numbers in code?

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under challenge 2.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

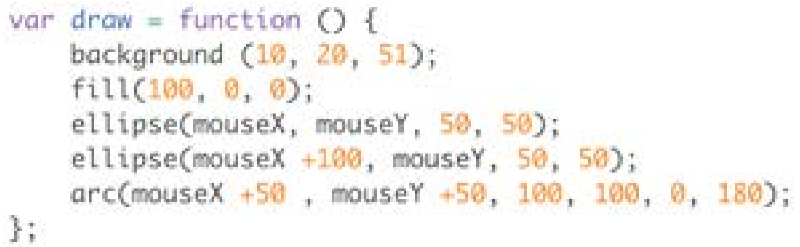
Lesson 4 : Challenge 3

Step by Step Learning Experience

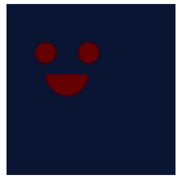
Step 1: Click the challenge tab

Tell students to click the challenge tab. Then read and try to complete the third challenge.

Sample Answer : Students will need to add code that makes a smile. Students may add multiple ellipses, or they may look at the research manual to learn about arcs.



```
var draw = function () {
    background (10, 20, 51);
    fill(100, 0, 0);
    ellipse(mouseX, mouseY, 50, 50);
    ellipse(mouseX +100, mouseY, 50, 50);
    arc(mouseX +50 , mouseY +50, 100, 100, 0, 180);
};
```



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.

Step 3: Compare code

Show students this sample version and ask them to look at other students versions of this code.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 4 : Challenge 3

Step by Step Learning Experience

Step 1: Click the challenge tab

Click the challenge tab. Then read and try to complete the third challenge.

Hints

1. Use the hints from challenge 1 to help you as well.
2. Draw how a smiley face looks.
3. Can you use circles to make a smile?
4. Do you know how to code an arc? How can you find out?

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under challenge 3.

Step 3: Different Solutions

When you are done, your teacher will ask you to look at the answers from other students and the sample solution.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

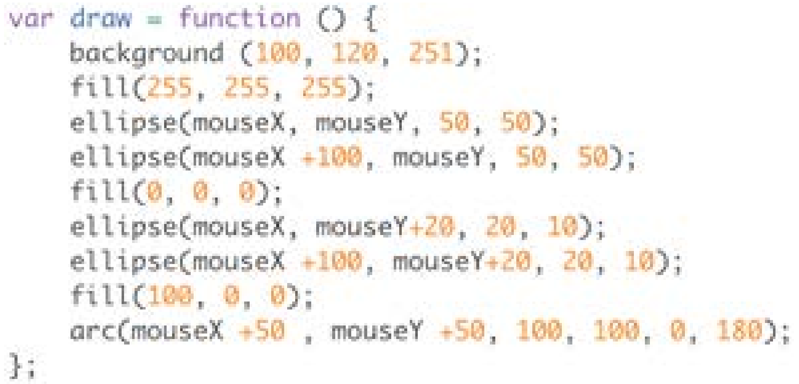
Lesson 5 : Additional Challenge 1

Step by Step Learning Experience

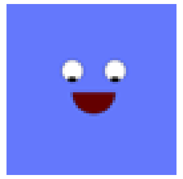
Step 1: Make a Personal Smiley face

Ask students "how could you change your smiley face to look more interesting".

Sample Solution: Added additional ellipses and multiple colors.



```
var draw = function () {
    background (100, 120, 251);
    fill(255, 255, 255);
    ellipse(mouseX, mouseY, 50, 50);
    ellipse(mouseX +100, mouseY, 50, 50);
    fill(0, 0, 0);
    ellipse(mouseX, mouseY+20, 20, 10);
    ellipse(mouseX +100, mouseY+20, 20, 10);
    fill(100, 0, 0);
    arc(mouseX +50 , mouseY +50, 100, 100, 0, 180);
};
```



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.

Step 3: Allow students to compared code and solutions.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 5 : Additional Challenge 1

Step by Step Learning Experience

Step 1: Make a Personal Smiley face

Ask students "how could you change your smiley face to look more interesting".

Hints

1. Add additional circles to make the eyes look more real
2. Add more colors to make a bigger difference.
3. Add additional new shapes you learn from the Reference Manual.

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 1.

Step 3: Different Solutions

When you are done, your teacher will ask you to look at the answers from other students and the sample solution.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation

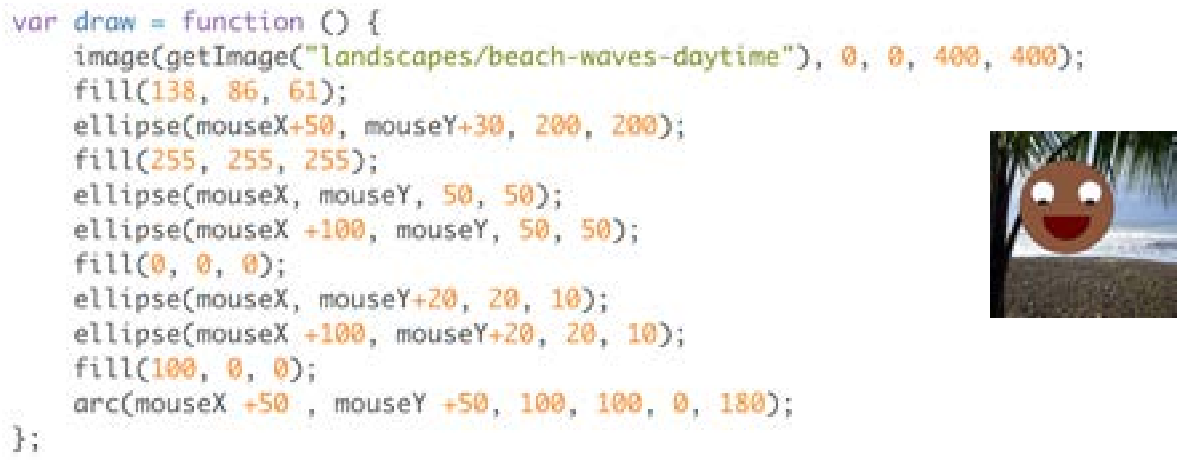
Lesson 6 : Additional Challenge 2

Step by Step Learning Experience

Step 1: Make it into a face in a location

Ask students to make this into more of a person in a location. Use an image to complete this task.

Sample Solution: Added additional ellipses and multiple colors.



```
var draw = function () {
    image(getImage("landscapes/beach-waves-daytime"), 0, 0, 400, 400);
    fill(138, 86, 61);
    ellipse(mouseX+50, mouseY+30, 200, 200);
    fill(255, 255, 255);
    ellipse(mouseX, mouseY, 50, 50);
    ellipse(mouseX +100, mouseY, 50, 50);
    fill(0, 0, 0);
    ellipse(mouseX, mouseY+20, 20, 10);
    ellipse(mouseX +100, mouseY+20, 20, 10);
    fill(100, 0, 0);
    arc(mouseX +50 , mouseY +50, 100, 100, 0, 180);
};
```

Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.

Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 6 : Additional Challenge 2

Step by Step Learning Experience

Step 1: Make a Personal Smiley face

How could you change your smiley face to look more interesting?

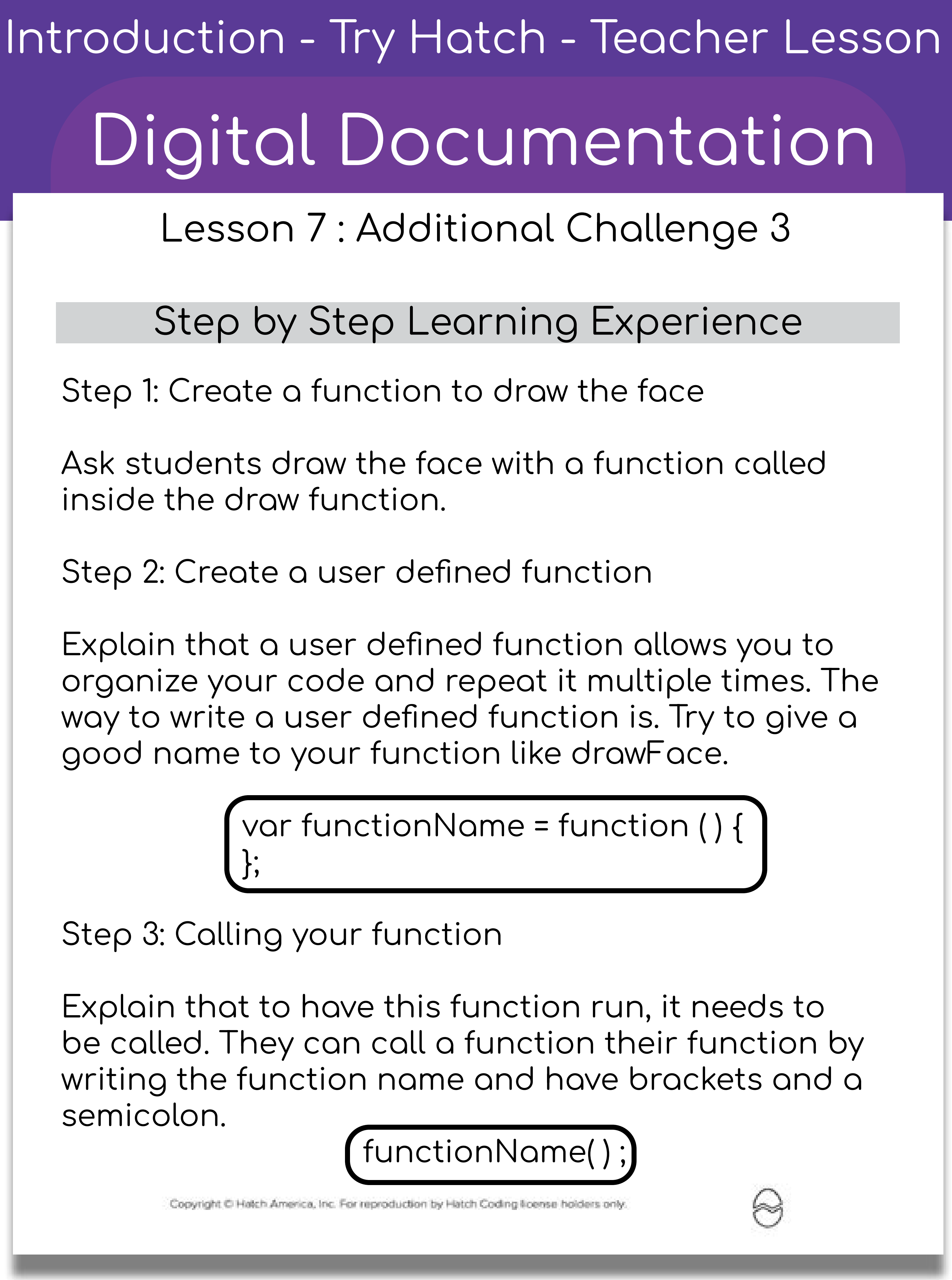
Hints

1. Add a background circle and use the skin color page in the Reference Manual to make it look more like you.
2. Add in an image using image(getImage(...)) learn more about the syntax in the Reference Manual.
3. Find a series of images in the image library
4. Use one of the landscapes as your background.

Images

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 2.



Digital Documentation

Lesson 7 : Additional Challenge 3

Step by Step Learning Experience

Step 1: Create a function to draw the face

Ask students draw the face with a function called inside the draw function.

Step 2: Create a user defined function

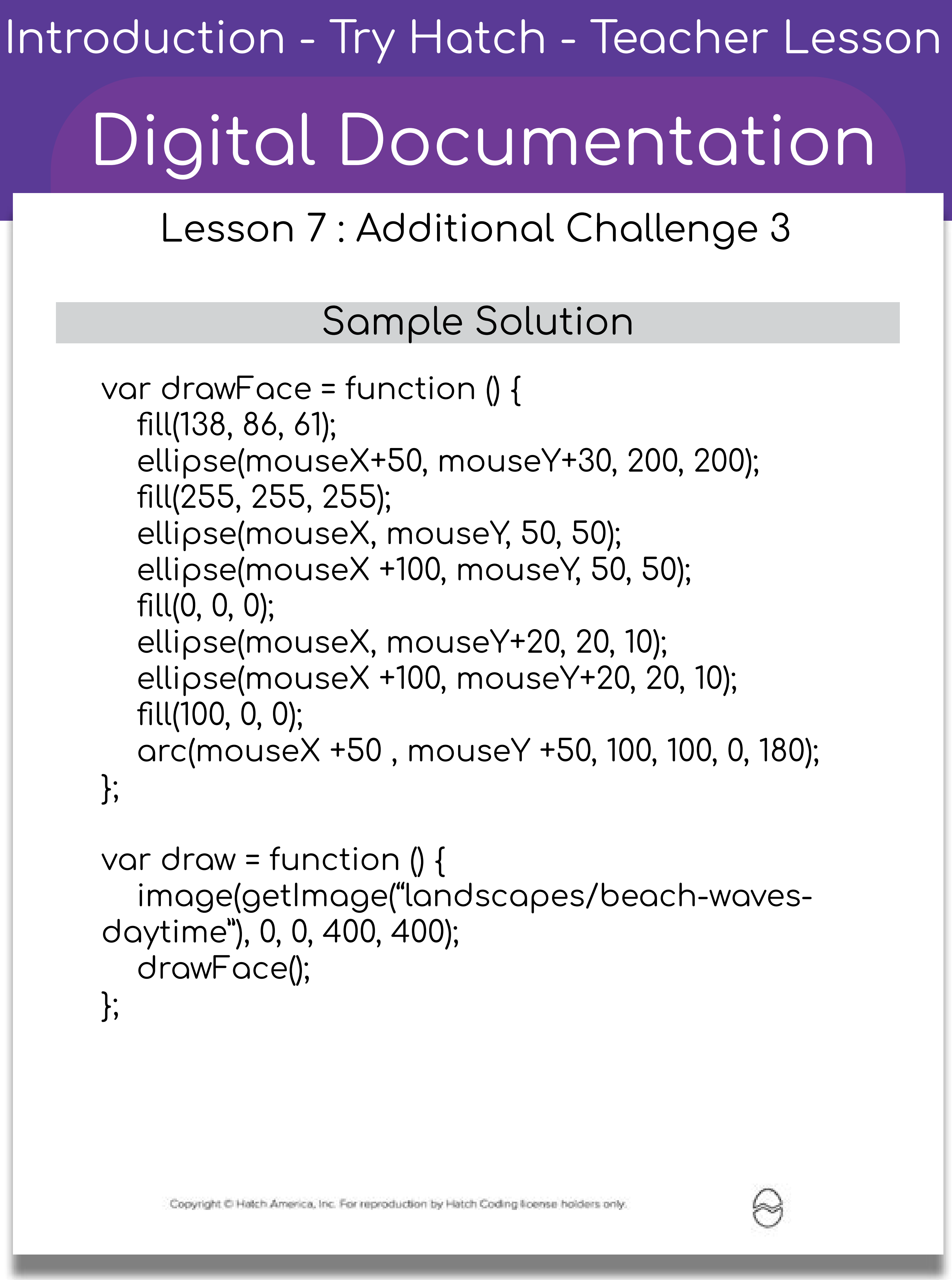
Explain that a user defined function allows you to organize your code and repeat it multiple times. The way to write a user defined function is. Try to give a good name to your function like drawFace.

```
var functionName = function ( ) {
};
```

Step 3: Calling your function

Explain that to have this function run, it needs to be called. They can call a function their function by writing the function name and have brackets and a semicolon.

```
functionName( );
```



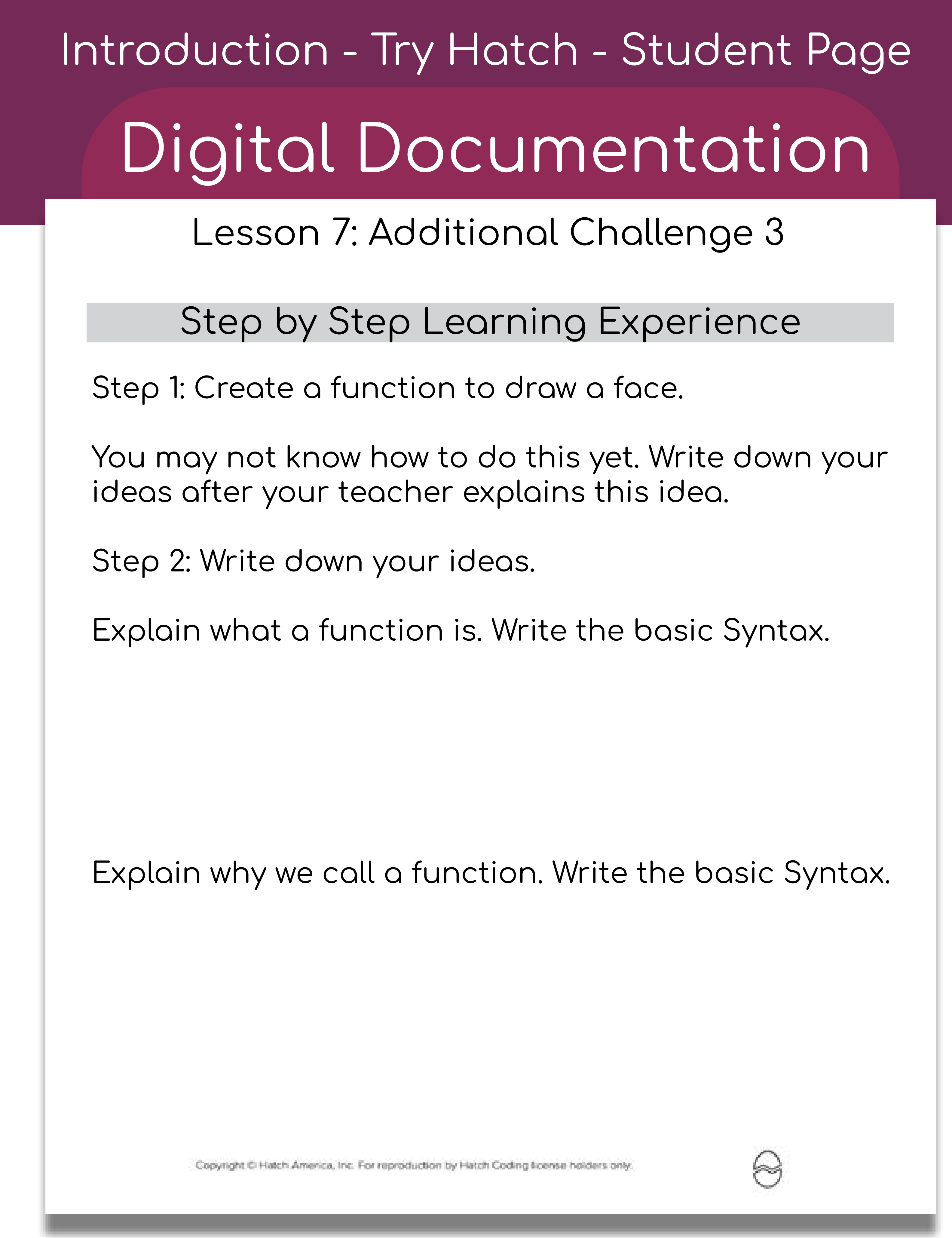
Digital Documentation

Lesson 7 : Additional Challenge 3

Sample Solution

```
var drawFace = function () {
   fill(138, 86, 61);
   ellipse(mouseX+50, mouseY+30, 200, 200);
   fill(255, 255, 255);
   ellipse(mouseX, mouseY, 50, 50);
   ellipse(mouseX +100, mouseY, 50, 50);
   fill(0, 0, 0);
   ellipse(mouseX, mouseY+20, 20, 10);
   ellipse(mouseX +100, mouseY+20, 20, 10);
   fill(100, 0, 0);
   arc(mouseX +50 , mouseY +50, 100, 100, 0, 180);
};

var draw = function () {
   image(getImage(“landscapes/beach-waves-
daytime”), 0, 0, 400, 400);
   drawFace();
};
```



Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 7: Additional Challenge 3

Step by Step Learning Experience

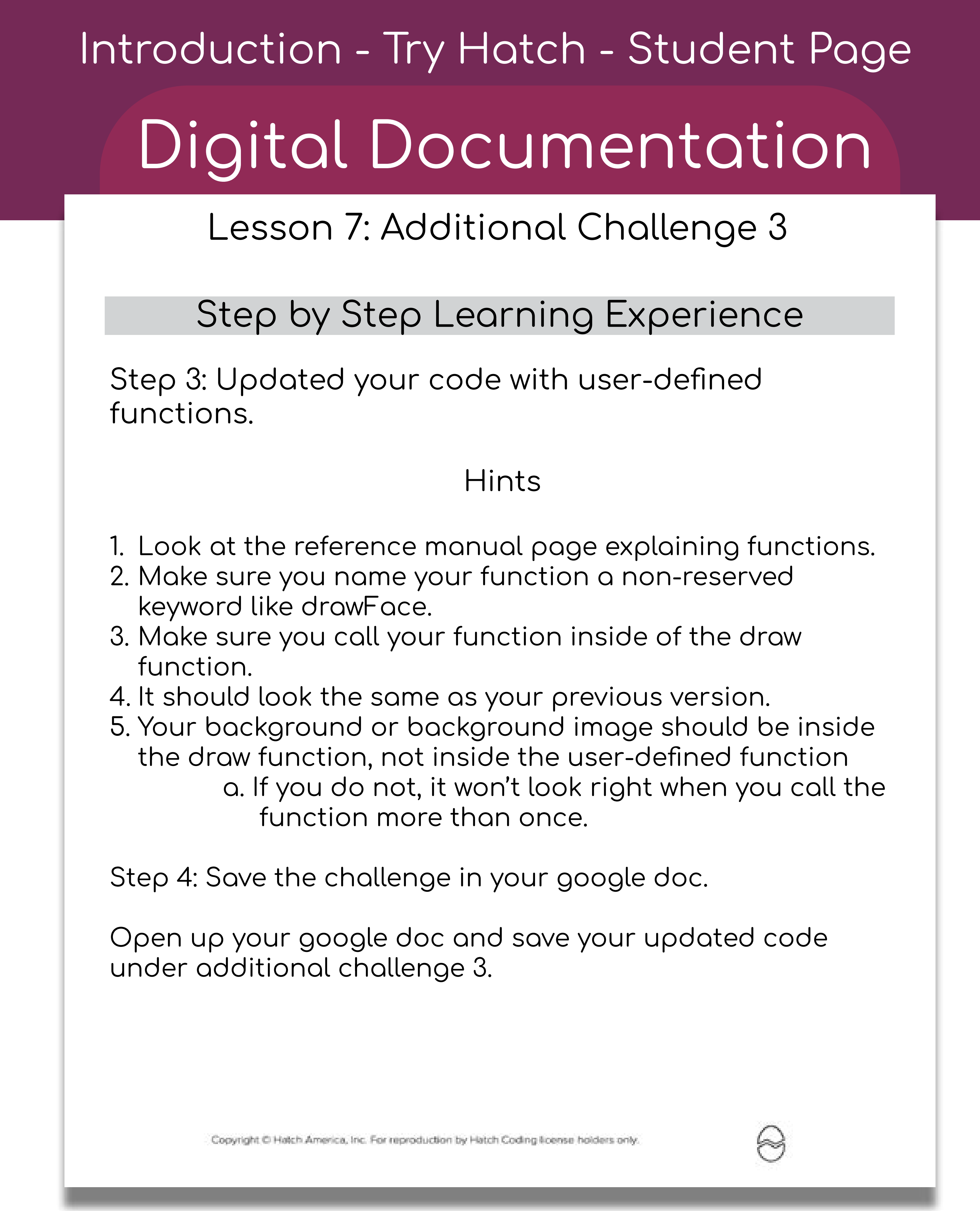
Step 1: Create a function to draw a face.

You may not know how to do this yet. Write down your ideas after your teacher explains this idea.

Step 2: Write down your ideas.

Explain what a function is. Write the basic Syntax.

Explain why we call a function. Write the basic Syntax.



Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 7: Additional Challenge 3

Step by Step Learning Experience

Step 3: Updated your code with user-defined functions.

Hints

1. Look at the reference manual page explaining functions.
2. Make sure you name your function a non-reserved keyword like drawFace.
3. Make sure you call your function inside of the draw function.
4. It should look the same as your previous version.
5. Your background or background image should be inside the draw function, not inside the user-defined function
   a. If you do not, it won't look right when you call the function more than once.

Step 4: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 3.

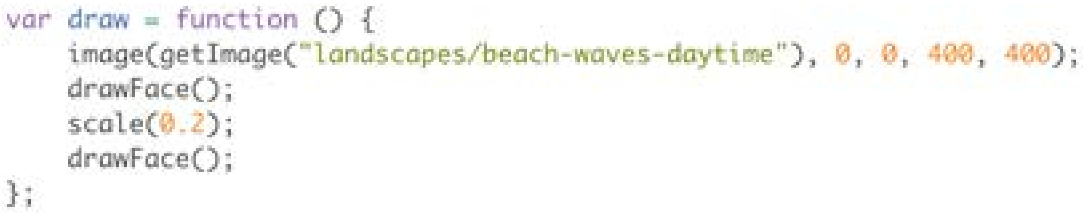
Lesson 8 : Additional Challenge 4

Step by Step Learning Experience

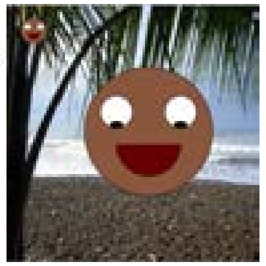
Step 1: Use scale to make a mini face

Ask students to make a mini version of their face appear by using the keyword scale and calling their function a second time.

Sample Solution: Note the drawFace function code can be found in the lesson 3 card. This is needed for this code to function.



```
var draw = function () {
    image(getImage("landscapes/beach-waves-daytime"), 0, 0, 400, 400);
    drawFace();
    scale(0.2);
    drawFace();
};
```



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.

Lesson 8 : Additional Challenge 4

Step by Step Learning Experience

Step 1: Use scale to make a mini face

Make a mini version of your face appear by using the keyword scale and calling their function a second time. You should have 2 faces on your screen

Hints

1. Use the reference Manual to learn about scale
2. Syntax for scale : scale (number to multiply by) ;
3. Call scale after your first user-defined function and before the second.
4. To scale something small the paremeter needs to be less than 1 and example would be 0.2

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 4.

Digital Documentation

Lesson 9 : Additional Challenge 5

Step by Step Learning Experience

Step 1: Use a for loop to make line of faces

Ask students to make a line of faces using a loop. Explain that a loop allows code inside to run multiple times.

Step 2: How to use a loop

Explain there are three parts of a loop. The start, the end and the step.

The start says what number the loop will start.

`var i = 0 ;` The loop start at 0

The end denotes when you want the loop to stop.

`i < 10` The stops at less than 10

The step explains how much we count by.

`i ++` This means count by 1

This is the full syntax for a for loop

`for (var i = 0; i < 10; i ++){`

Digital Documentation

Lesson 9 : Additional Challenge 5

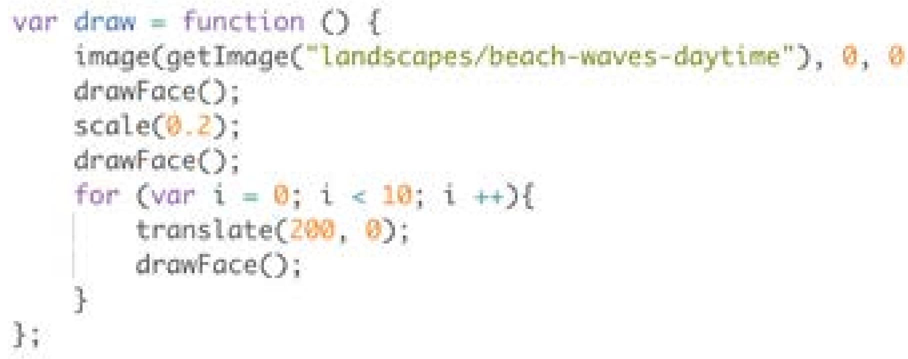
Step by Step Learning Experience

Step 3: Using a loop

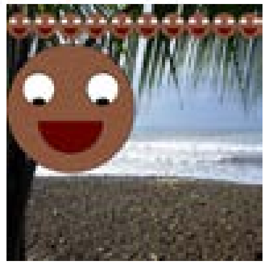
Explain that the loop for this project needs to be within the draw function and they will need to translate their drawFace function each inside the for loop.

Inside the for loop their should be
translate(how far X , how far Y);
drawFace();

Sample Solution: drawFace function code can be found insolution 3.

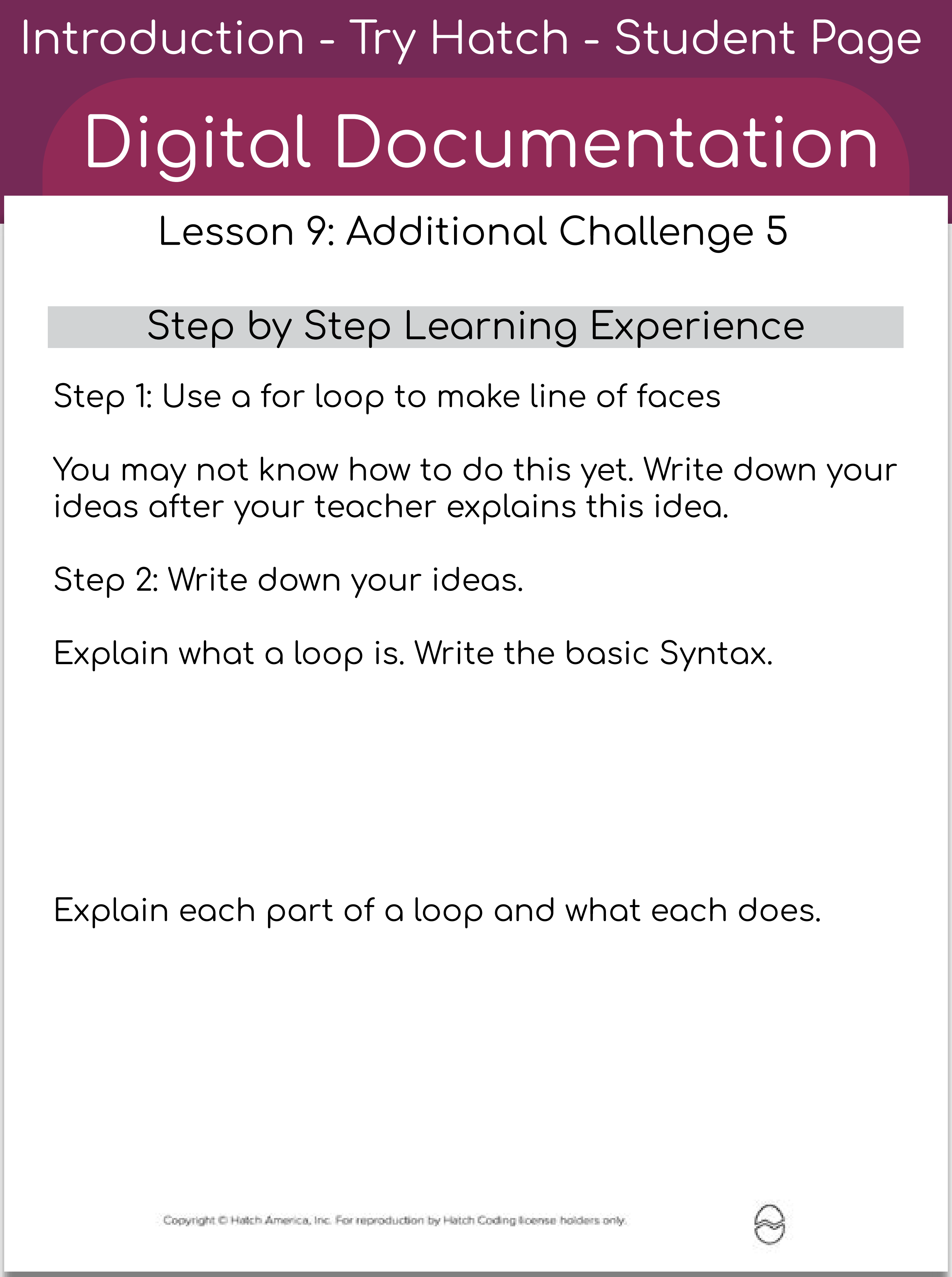


```
var draw = function () {
    image(getImage("landscapes/beach-waves-daytime"), 0, 0, 400, 400);
    drawFace();
    scale(0.2);
    drawFace();
    for (var i = 0; i < 10; i ++){
        translate(200, 0);
        drawFace();
    }
};
```



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.



Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 9: Additional Challenge 5

Step by Step Learning Experience

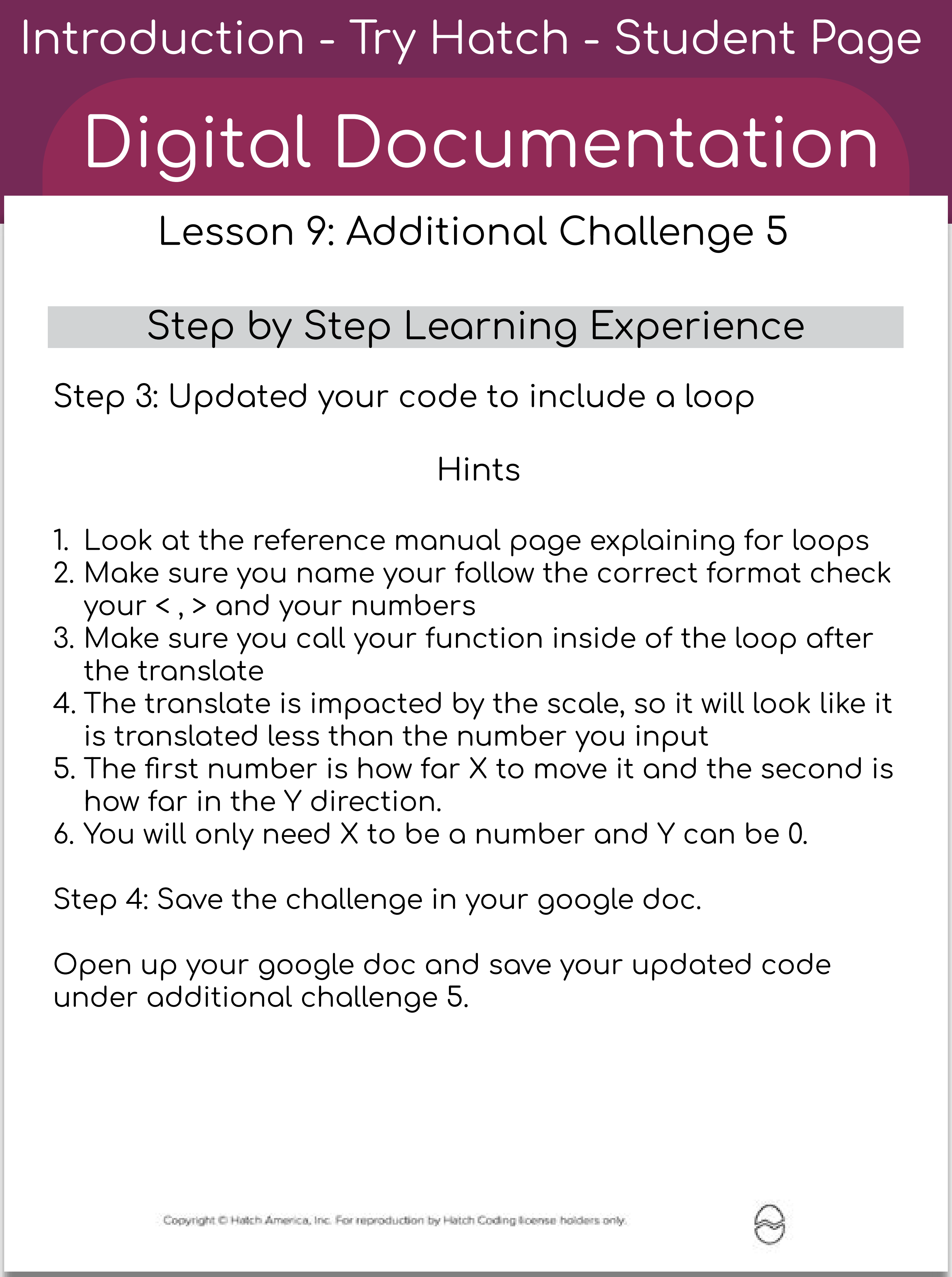
Step 1: Use a for loop to make line of faces

You may not know how to do this yet. Write down your ideas after your teacher explains this idea.

Step 2: Write down your ideas.

Explain what a loop is. Write the basic Syntax.

Explain each part of a loop and what each does.



Introduction - Try Hatch - Student Page

Digital Documentation

Lesson 9: Additional Challenge 5

Step by Step Learning Experience

Step 3: Updated your code to include a loop

Hints

1. Look at the reference manual page explaining for loops
2. Make sure you name your follow the correct format check your < , > and your numbers
3. Make sure you call your function inside of the loop after the translate
4. The translate is impacted by the scale, so it will look like it is translated less than the number you input
5. The first number is how far X to move it and the second is how far in the Y direction.
6. You will only need X to be a number and Y can be 0.

Step 4: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 5.

Lesson 10 : Additional Challenge 6

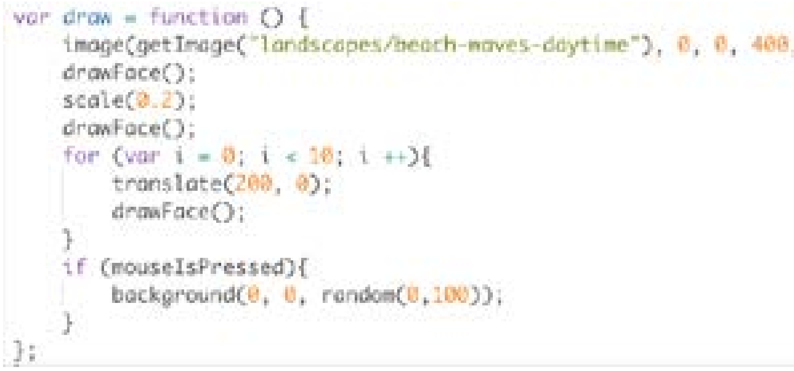
Step by Step Learning Experience

Step 1: Add a mouseIsPressed in an if statement

An if statement checks if what is in the brackets is true. If it is, then the code runs, if not it does not.

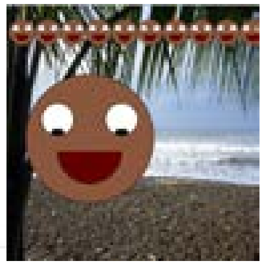
For this lesson you will be adding mouseIsPressed into an if statement and hiding your image with a backround.

Sample Solution: Note the drawFace function code can be found in the lesson 3 card.

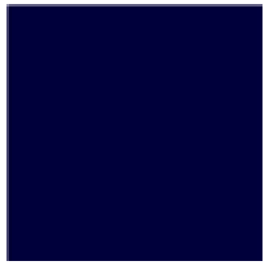


```
var draw = function () {
    image(getImage("landscapes/beach-waves-daytime"), 0, 0, 400, 400);
    drawFace();
    scale(0.2);
    drawFace();
    for (var i = 0; i < 10; i ++){
        translate(200, 0);
        drawFace();
    }
    if (mouseIsPressed){
        background(0, 0, random(0,100));
    }
};
```

Mouse is not Pressed



Mouse is Pressed



Step 2: Save Challenge Code in your google doc

Ask students to save their code into their google doc.

Lesson 10 : Additional Challenge 6

Step by Step Learning Experience

Step 1: Add a mouseIsPressed in an if statement

You will need to read more about if statements prior to starting. Put mouseIsPressed inside of your if statement.

Hints

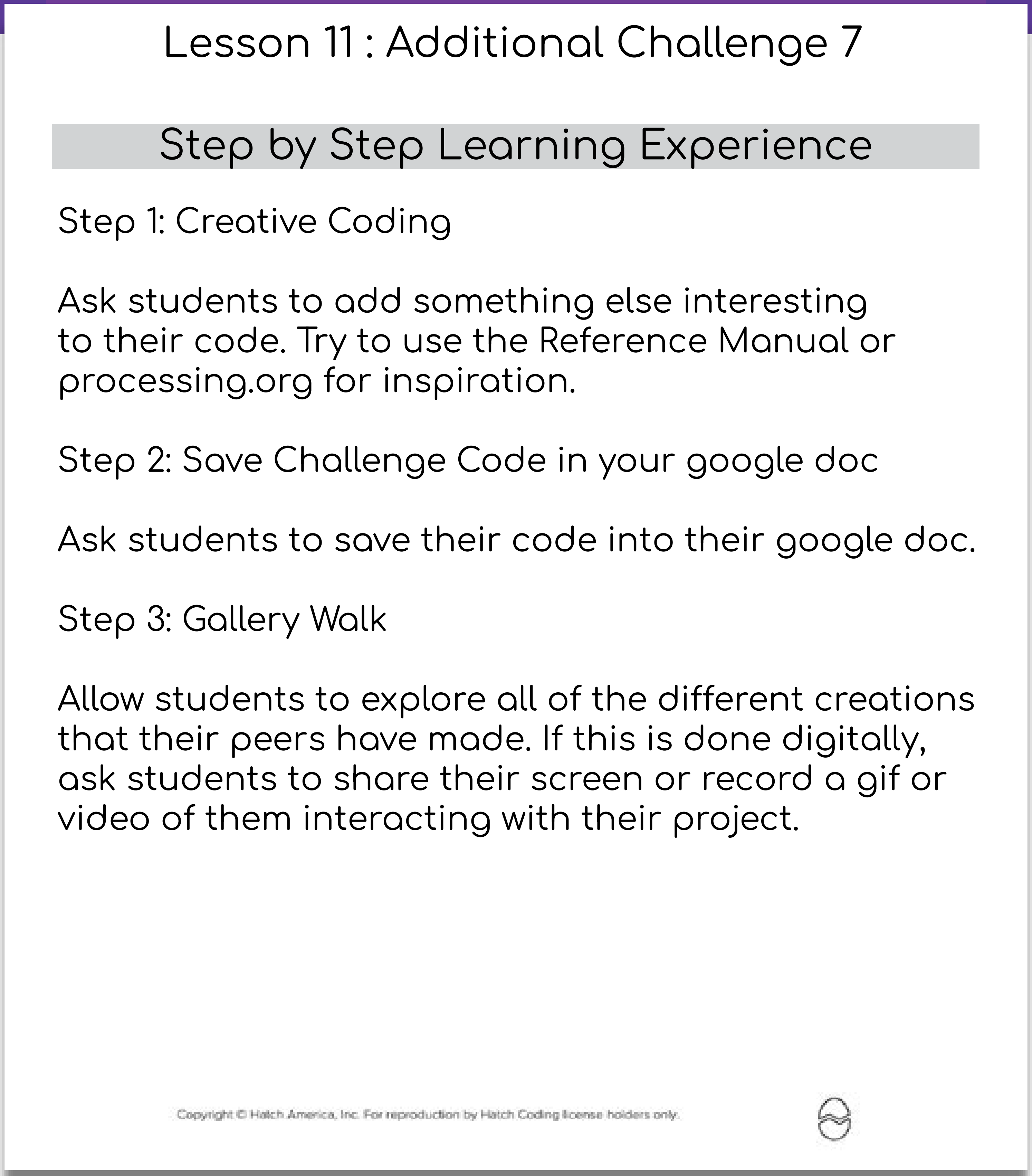
1. Use the reference Manual to learn about if statements
2. If statements check if the parameter is true and if it is, the code inside the { } will run
3. Think about something that you would like to happen when the mouseIsPressed
4. A simple solution is to turn the screen black
   Add a black background into the if statement.

Step 2: Save the challenge in your google doc.

Open up your google doc and save your updated code under additional challenge 4.

Introduction - Try Hatch - Teacher Lesson

Digital Documentation



Lesson 11 : Additional Challenge 7

Step by Step Learning Experience

Step 1: Creative Coding

Ask students to add something else interesting to their code. Try to use the Reference Manual or processing.org for inspiration.

Step 2: Save Challenge Code in your google doc

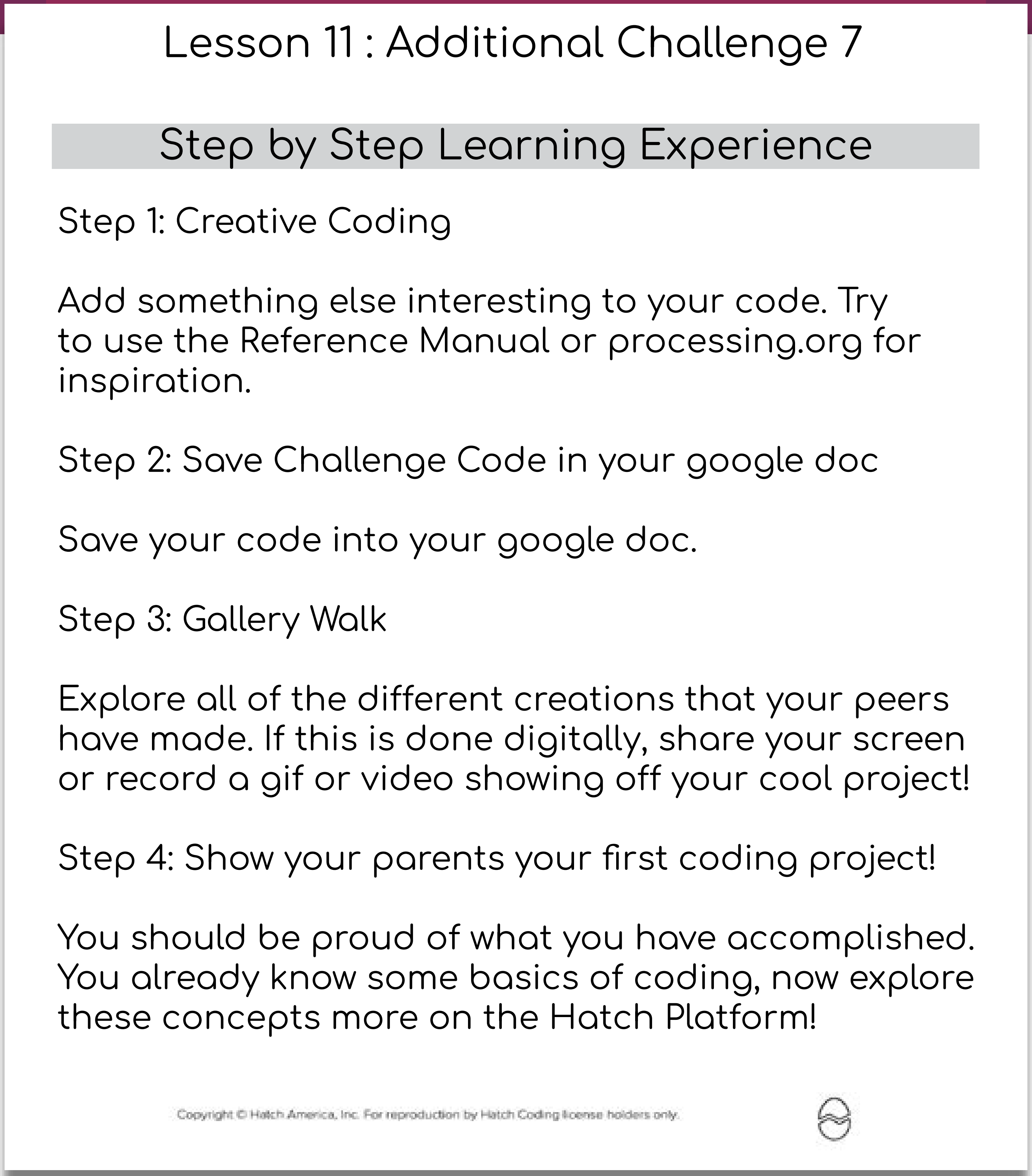
Ask students to save their code into their google doc.

Step 3: Gallery Walk

Allow students to explore all of the different creations that their peers have made. If this is done digitally, ask students to share their screen or record a gif or video of them interacting with their project.

Introduction - Try Hatch - Student Page

Digital Documentation



Lesson 11 : Additional Challenge 7

Step by Step Learning Experience

Step 1: Creative Coding

Add something else interesting to your code. Try to use the Reference Manual or processing.org for inspiration.

Step 2: Save Challenge Code in your google doc

Save your code into your google doc.

Step 3: Gallery Walk

Explore all of the different creations that your peers have made. If this is done digitally, share your screen or record a gif or video showing off your cool project!

Step 4: Show your parents your first coding project!

You should be proud of what you have accomplished. You already know some basics of coding, now explore these concepts more on the Hatch Platform!