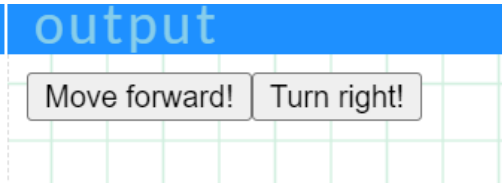


<p>Standard:</p> <ul style="list-style-type: none"> 6.CS.5 Add on-screen buttons and write student-defined functions to encapsulate and name sequences of Pencil Code commands. 	<p>Big Idea:</p> <ul style="list-style-type: none"> A function encapsulates a set of instructions, allowing these instructions to be run with a single line of code.
<p>Essential Questions:</p> <ul style="list-style-type: none"> What is a button in Pencil Code? What is the code for creating a button? How does indentation show what's the button code, and what isn't? How can we move a button to a desired location on the screen? 	
<p>Point:</p> <ul style="list-style-type: none"> Pencil Code's button command creates a button on the screen, which runs code when it's clicked. The button's label is placed in quotes, next to the button command, followed by a comma and an arrow. The arrow is constructed with a hyphen and a greater than sign: <code>-></code> The button's code has to be indented beneath the line of code that creates the button. <pre> 1 button "Move forward!",> 2 fd 100 3 button "Turn right!",> 4 rt 90 </pre>  <ul style="list-style-type: none"> Careful indentation is very important when making buttons in Pencil Code, because it signifies which commands are inside the button code and which are not. Buttons can be resized and repositioned on the Pencil Code screen, just like images. <pre> f = button "Forward", > fd 50 f.width 80 f.jumpto 0, -175 </pre>	
<p>Notes:</p> <p>In the Remote Control lesson, students use Pencil Code's button command to make their projects interactive. The button command creates a labeled button on the screen that runs a certain segment of code whenever it is clicked by the user. The starter code for this lesson includes two buttons: a "Forward" button (which instructs the turtle to move forward 50 steps) and a "Right" button (which instructs the turtle to make a 90-degree right turn). In the first activity, students are instructed to use additional buttons to complete the turtle's remote control. In the second activity, students learn to resize and reposition their buttons, creating a more stylish user interface. The button click is an example of an event—an action or input that a computer program can respond to. As soon as a button is clicked, the computer runs that section of the code.</p>	
<p>Vocab:</p> <ul style="list-style-type: none"> button, function, indentation, event, user interface 	
<p>Materials:</p> <ul style="list-style-type: none"> Chromebooks CS notebooks Do Nows: 3.1a Explore (Remote Control), 3.1b Paper Programming (Image Properties) 	

DAY 1**Do Now (10 mins)**

- **Say:** In today's do now, you have a picture of an Xbox controller. Think about a video game that you like to play! In that game, what is the purpose of each of the buttons and joysticks shown here? What does each one do?
- **Review:** What did you all come up with? What games were you thinking about? How do the controller buttons affect what happens in these games? How do you think the game knows what to do when each of those buttons is pressed? [The game code must somehow know that a button is pressed and then run different code in response.]
- **Frame:** Today, we're actually going to start learning about how to add buttons into our Pencil Code program that the user can click to make something happen on the screen.

Explore (5 mins)

- **Explain:** We're going to start today in the Pencil Code gym, with the [Remote Control chapter](#). When I say go, I want you to run this code and see if you can figure out how it works.
 - What does the button command do?
 - What does the text in quotes do?
 - How does Pencil Code know what code to run when the button is clicked?
 - Try **adjusting the values** for the "Forward" and "Right" buttons - what changes?

Task 1: Add Buttons for Backward and Left (10 mins)

- **Say:** As a first step, I want you to add buttons to make the turtle go backward and turn left.
- **Recap:** Did anyone run into any problems in adding these two buttons? [Kids will likely have had issues with indentation.]

```
10  ###These buttons turn left and move backwards.###
11  button "Backwards", ->
12    bk 50
13  button "Left", ->
14    lt 90
```

- **Demo/Explain:** The **button** command needs to be along the left edge of the code space, but the code that runs when the button is pressed has to be **indented** one level. One way to accomplish the indentation is using the **Tab** key, which is on the left edge of the keyboard; it will either have the word "**Tab**" on it, or an arrow pointing right →|.

Task 2: Finish the Remote (15 mins)

- **Say:** Now **finish** the remote control by including additional buttons that control the turtle's movements. Here are some ideas, but you need to have at least six more buttons:
 - Make buttons that control the **color** and/or **thickness** of the pen.
 - Make a button that **fills** a shape in with a particular color.
 - Make buttons that draw **dots** or **boxes**.

Vocab / Notes (10 mins)

- **Say:** I need everyone to close your computers and take out your computer science **notebooks**. In your notebooks, please turn to the first empty page, put today's date at the top. Copy the following code down:

```
1 button "Move forward!",->
2   fd 100
3 button "Turn right!",->
4   rt 90
```

DAY 2**Do Now (10 mins)**

- **Say:** Today's do-now should be pretty quick. It's asking you what commands we need to make an image smaller and reposition it at the center of the screen. Look back to your notes from Unit 2 if you need to refresh your memory of how we would do this?
- **Review:** What did you all come up with? [We can set the height of the image, and then use jumpto to move it to 0,0 (the origin or center of the screen).]

```
1 c = img 'cactus'
2 c.height 100
3 c.jumpto 0,0
```

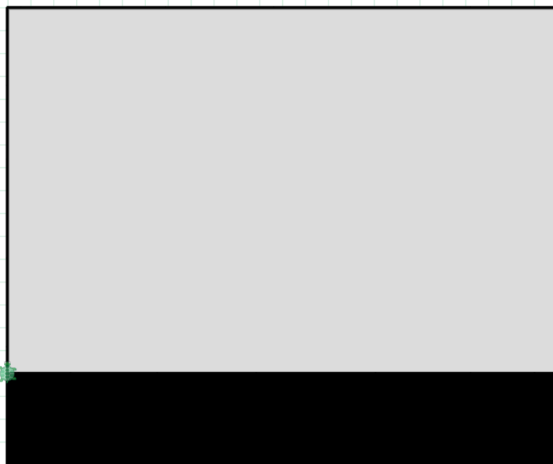
- **Frame:** Today, we're going to use a very similar technique to resize and arrange the buttons on our remote control project, so that we create a more professional looking user interface. If I have a button on the screen, I can set its width and/or height, and then just jumpto to move where I want it.

```
f = button "Forward", ->
  fd 50
f.width 80
f.jumpto 0, -175
```

- This code sets the width of the button to 80 pixels, and then jumps it to 0, -175 on the screen.

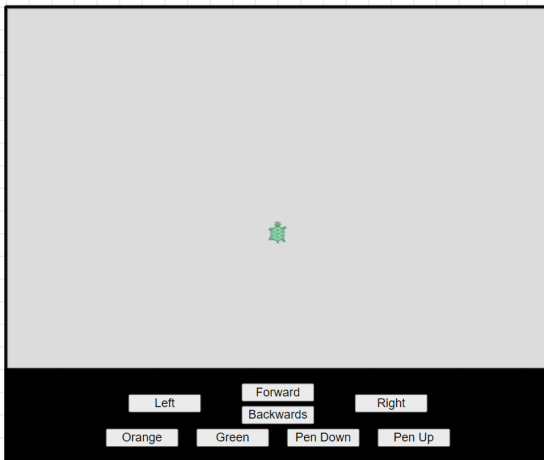
Task 1: Draw a Button Panel and Screen (10 mins)

- **Say:** To begin today, I want you to draw two rectangles in different colors on the screen: one will be the button panel - this is where your buttons will go; the other is the screen where the turtle will run all of your commands.

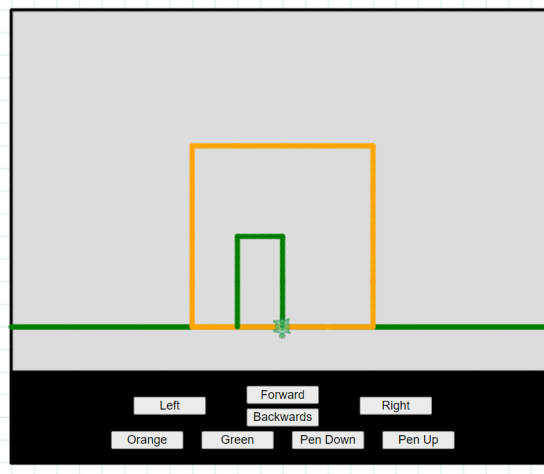


Task 2: Arrange the Buttons on the Button Panel (20 mins)

- Once you've finished with the button panel and screen, **resize** and **arrange** your buttons on the button panel. This will be your **user interface** for the project.

**Extension Challenge: Use Your Remote Control to Draw Something!**

- If you're finished with your Remote Control, use your project to draw something interesting!

**Shareout (10 mins)**

- Leave 10 minutes at the end of class to share out a few of the best remote control designs from your class. Remember that you can find any student's work by going to their **username.pencilcode.net**.