



# Operate the Santavator

Our next objective is to fix and operate the **Santavator**, an elevator that gives access to additional floors in Kringle Castle.

## Objective

Talk to Pepper Minstix in the entryway to get some hints about the Santavator.

Difficulty: 2/5

## Pepper Mintstix dialog:

There's a Santavator that moves visitors from floor to floor, but it's a bit wonky. You'll need a key and other odd objects. Try talking to Sparkle Redberry about the key. For the odd objects, maybe just wander around the castle and see what you find on the floor. Once you have a few, try using them to split, redirect, and color the Super Santavator Sparkle Stream (S4). You need to power the red, yellow, and green receivers with the right color light!

## Sparkle Redberry's dialog:

Hey hey, Sparkle Redberry here!

The Santavator is on the fritz. Something with the wiring is grinchy, but maybe you can rig something up?

Here's the key! Good luck!

On another note, I heard Santa say that he was thinking of canceling KringleCon this year!

At first, I thought it was a joke, but he seemed serious. I'm glad he changed his mind.

Have you had a chance to look at the Santavator yet?

With that key, you can look under the panel and see the Super Santavator Sparkle Stream (S4).

To get to different floors, you'll need to power the various colored receivers.

... There MAY be a way to bypass the S4 stream.

## Hints

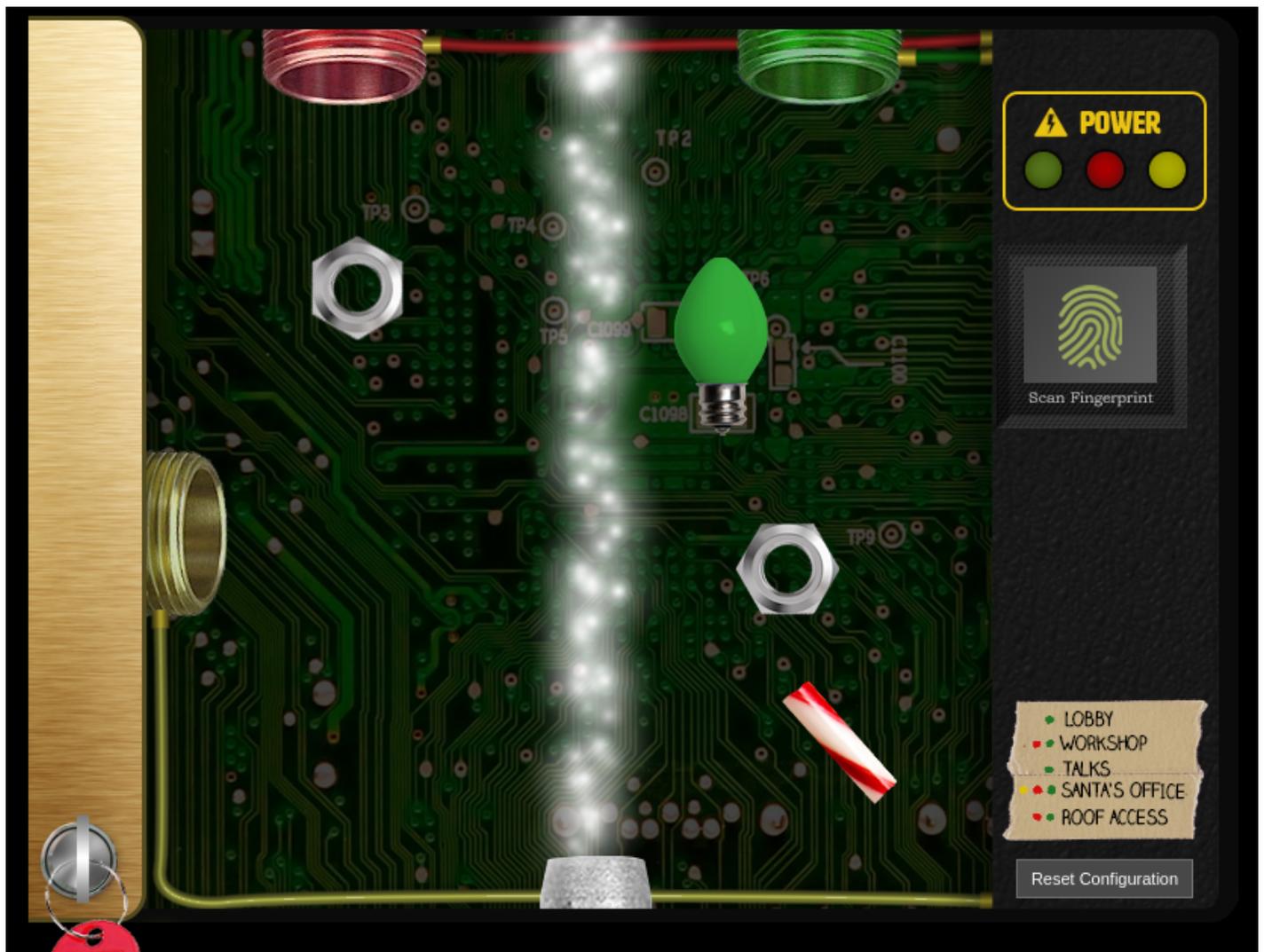
It's really more art than science. The goal is to put the right colored light into the receivers on the left and top of the panel.

## Solution

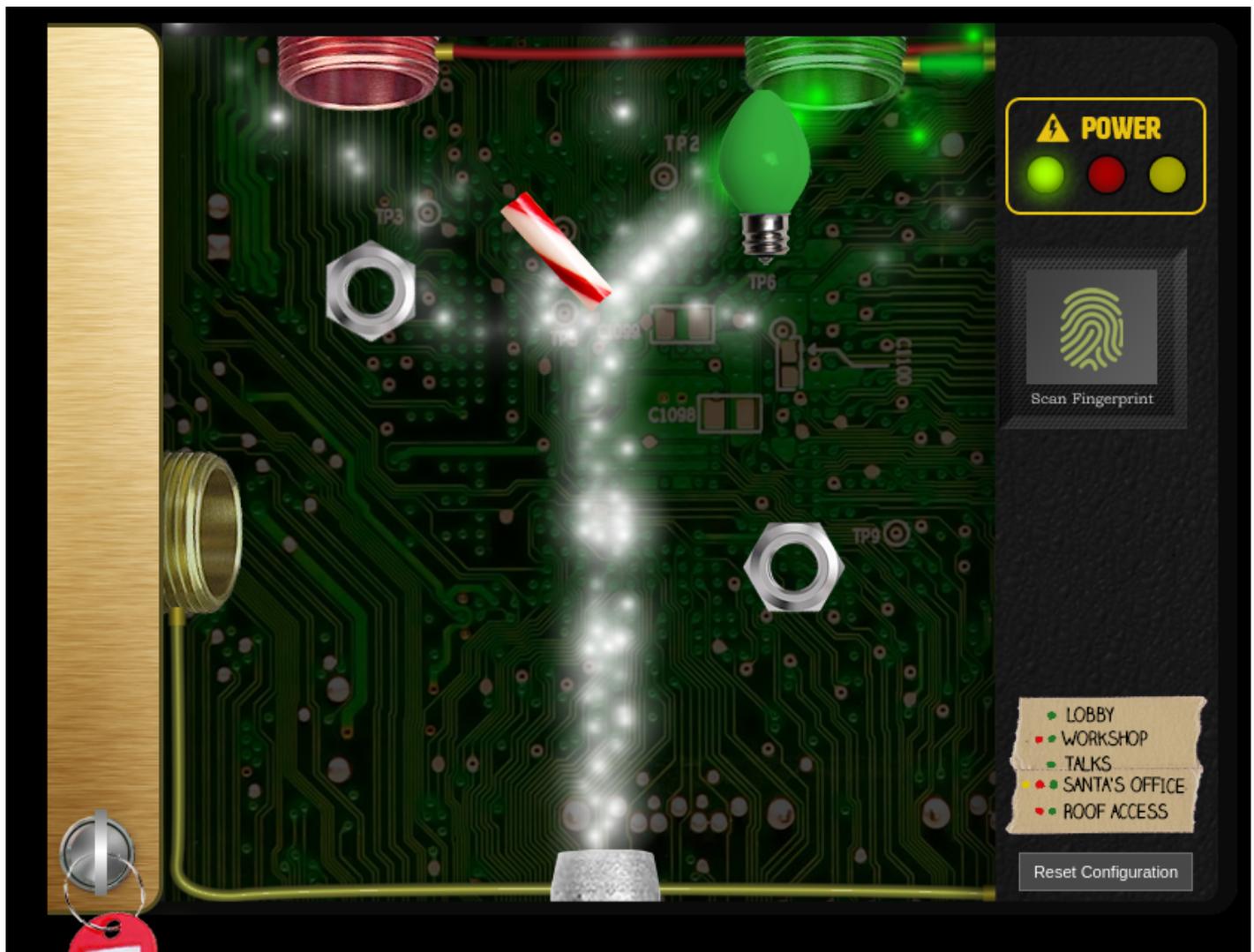
In the top-right corner of the courtyard you'll find the elf Sparkle Redberry standing next to the **Santavator**, a magic-powered elevator. He'll give you a key to open the operator panel in the Santavator:



Also, in your wanderings around the castle, you've probably seen and picked up some random items laying on the floor (a candycane in the entryway of the castle, for example). Those items will be behind the operator panel, which opens by clicking on the key:



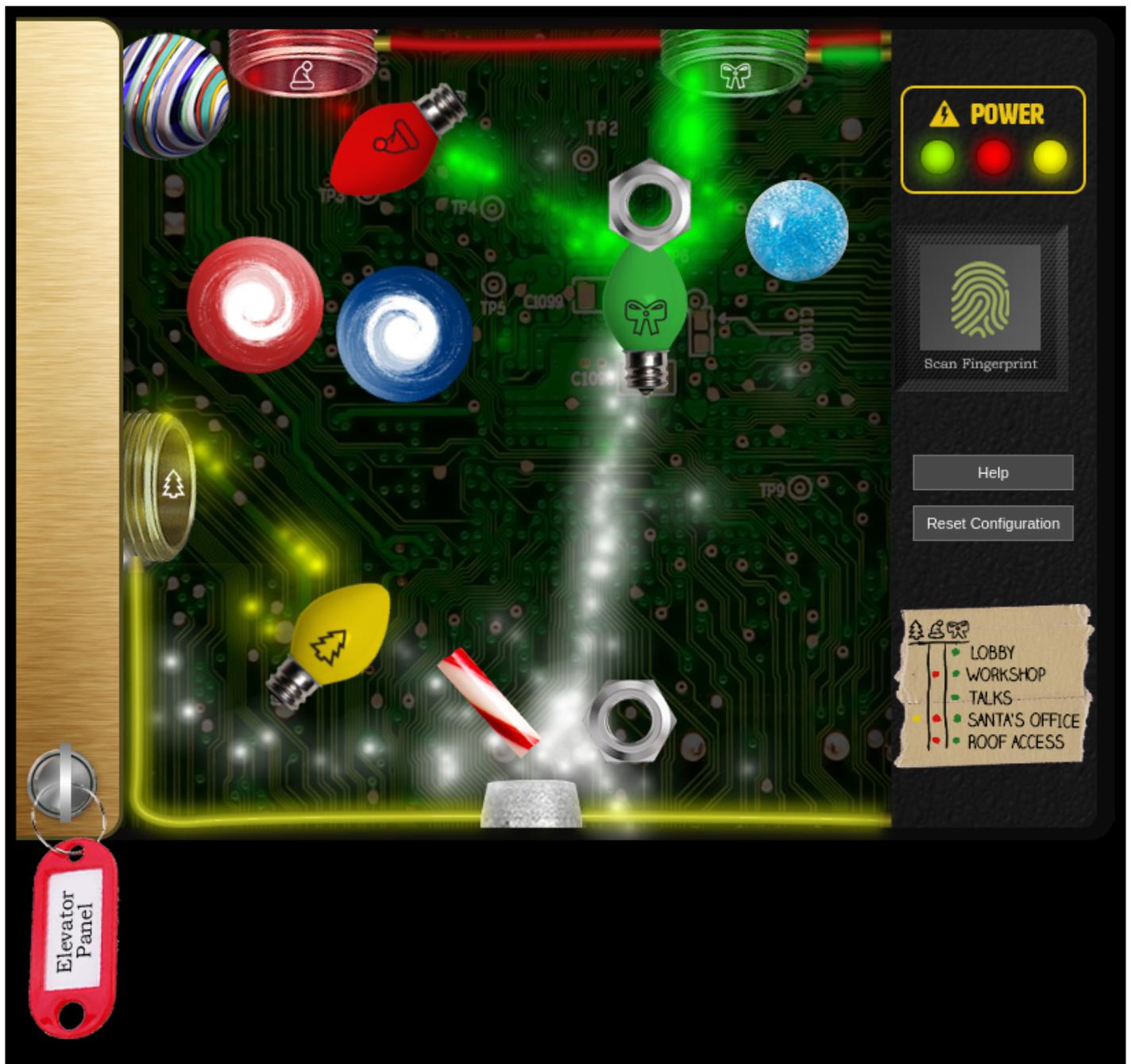
The legend in the bottom-right corner shows what color receiver needs to be "powered" for the button for that floor to be active. For example, powering the green receiver will allow access to the **Lobby** and the **KringleCon Talks** floors. The collected items can be dragged around to split, direct, and change the colors of the Sparkle Stream. For example, this configuration will power the green receiver:



Closing the panel with the Sparkle Stream in that configuration now allows access to floor 2 (KringleCon talks):



Visiting any floor other than the Lobby will fulfill the objective. Eventually more objects can be found around the castle, and with much creative moving of objects, all the receiver can be powered:



There is a simpler way, that doesn't involve any objects, splitting Sparkle streams, and powered receiver. The answer lies in the source code to the Santavator application, and why client-side security checks can be a bad idea.

The application that controls the Santavator is an embedded iframe in the browser window. The application source is at <https://elevator.kringlecastle.com/app.js>. Of interest are the following blocks of JavaScript:

```

326 const handleBtn = event => {
327   const targetFloor = event.currentTarget.attributes['data-
328 floor'].value;
329   $.ajax({
330     type: 'POST',
331     url: POST_URL,
332     dataType: 'json',
333     contentType: 'application/json',
334     data: JSON.stringify({
335       targetFloor,
336       id: getParams.id,
337     }),
338     success: (res, status) => {
339       if (res.hash) {
340         __POST_RESULTS__({
341           resourceId: getParams.id || '1111',
342           hash: res.hash,
343           action: `goToFloor-${targetFloor}`,
344         });
345       }
346     }
347   });
}

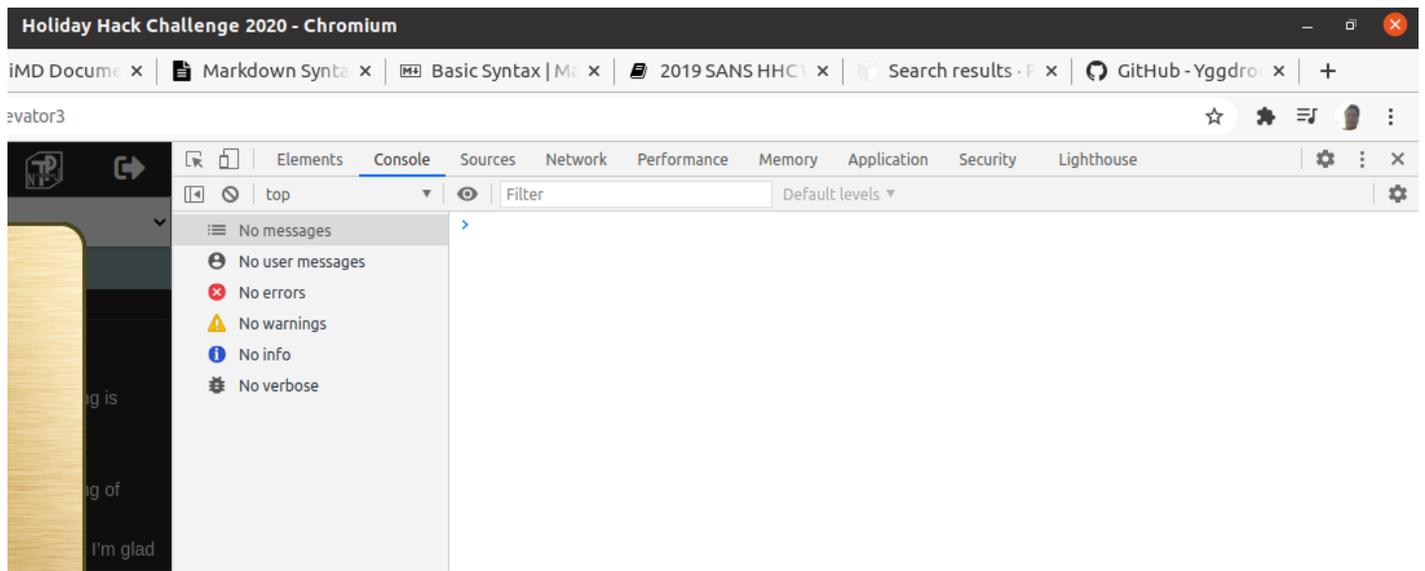
```

```

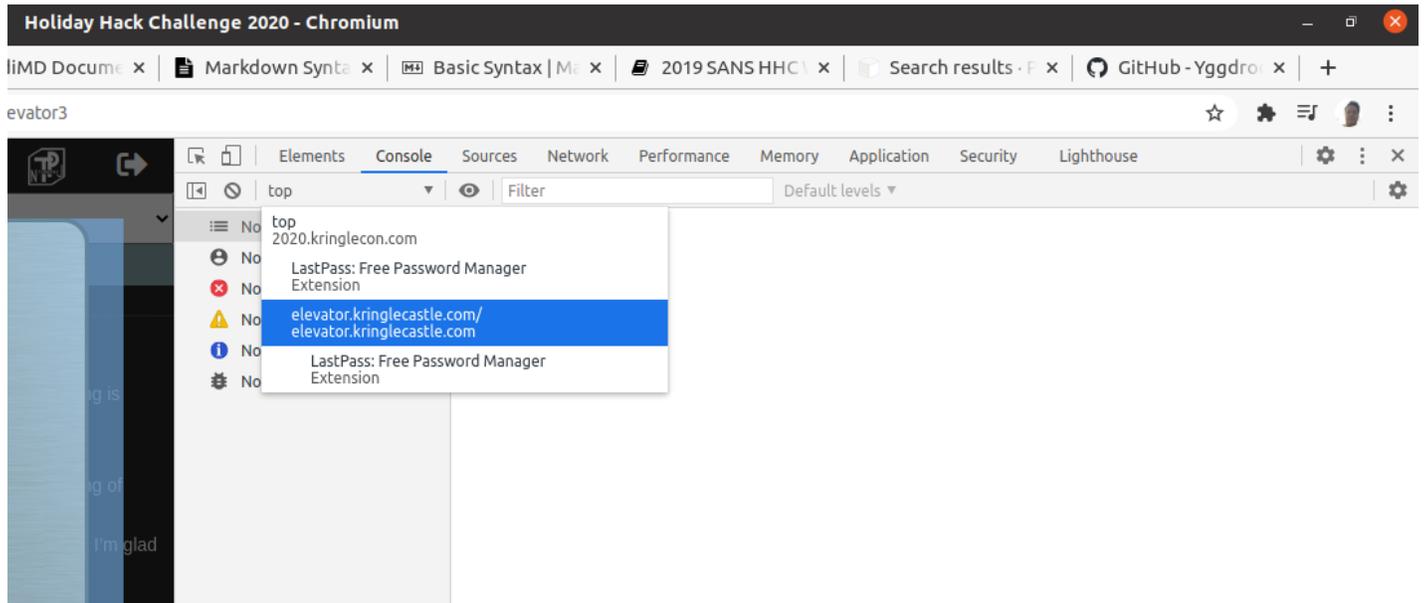
383 const btn1 = document.querySelector('button[data-
384 floor="1"]');
385 const btn2 = document.querySelector('button[data-
386 floor="1.5"]');
387 const btn3 = document.querySelector('button[data-
388 floor="2"]');
389 const btn4 = document.querySelector('button[data-
390 floor="3"]');
391 const btnr = document.querySelector('button[data-
392 floor="r"]');
393
    btn1.addEventListener('click', handleBtn);
    btn2.addEventListener('click', handleBtn);
    btn3.addEventListener('click', handleBtn);
    btn4.addEventListener('click', handleBtn4);
    btnr.addEventListener('click', handleBtn);

```

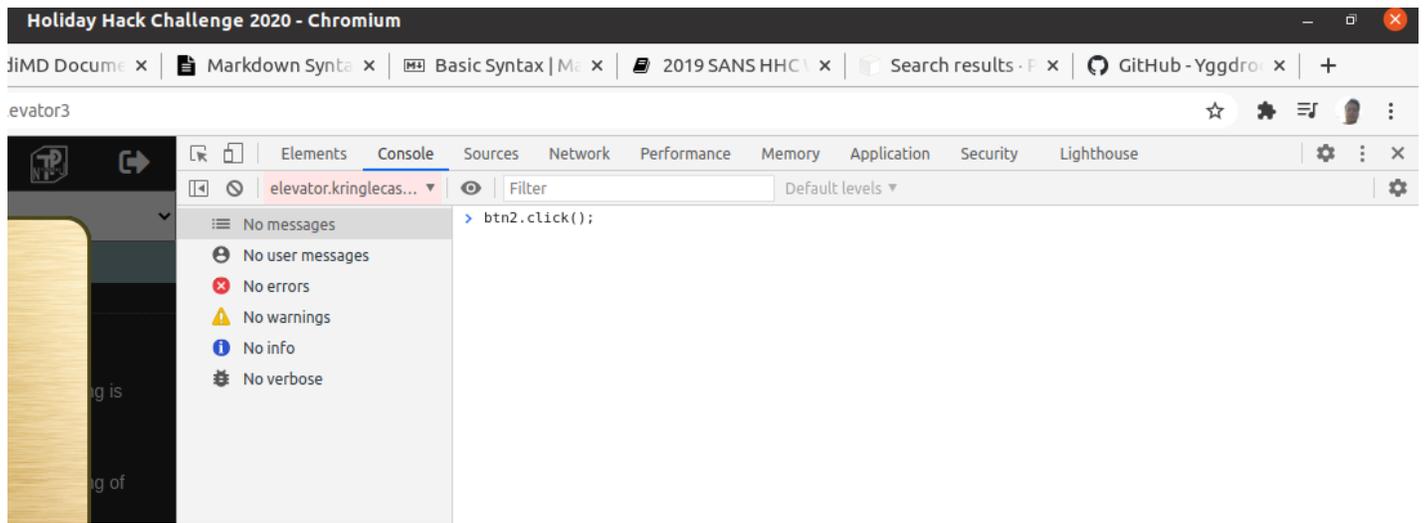
The `handleBtn` function is called when any of the buttons are clicked on, with the appropriate floor data (ignoring `handleBtn4` for now). We can simulate a click on the button in the browser's JavaScript console using `btnX.click()`. Open the browser's Developer Tools menu and go to the Console tab (Google Chrome shown):



Because the Santavator code is an iFrame, it runs in a separate JavaScript context from the main page. In Chrome, that context can be selected via the dropdown menu at the top-left of the tools:



Switch the context to `elevator.kringlecastle.com`, and enter `btnX.click();` at the `>` prompt. You'll be taken to the floor associated with the button.



## Answer

Visit any floor other than the Lobby to fulfill this objective.