Your website probably looks very good now! But it also probably could use a little bit extra visual feedback to give the user...

Some pointers about JavaScript

This time, JavaScript is indeed a programming language! It can be used for many things, but in the context in which it runs in a browser (therefore, on a webpage), it typically changes the webpage and CSS rules in any way you like. JavaScript in the browser is event-driven, which means code typically executes when an event happens (like: the user clicks on something, or scrolls down the page, or resizes the window, etc.)

But let's experiment right away!

Your project: a smart thumbnail

The point of the this part of your project is to display a large image as a smaller thumbnail, allowing the user to click on it to make it bigger.

During this part of the project, we're going to be writing code in two locations: in a .js file so that the webpages run it each time they load; and we'll also run code in the browser's JavaScript console, just to test stuff as the webpage is live in our browser. First, let's set both of those up.

Setting up the .js file

Create an empty file `behavior.js` in your `/var/www/html` folder; this is the file in which your JavaScript code loaded by your webpage will go. Then, so that the webpages know where to find it, add this in the `<head>` tag of all your webpages: `<script src="behavior.js"></script>`.

Before anything, start by copy-pasting this into your `behavior.js` file:

```
document.addEventListener("DOMContentLoaded", function(event) {

});
```

For this part of the project, all of your JavaScript code must be written between those two lines. Any code outside of these two lines would execute before the HTML page finished loading, so the result could be very random!

To test that your file is properly taken into account by your webpage, write `alert('Hello!');` between the two lines in `behavior.js`. Refresh the webpage; a dialog box should greet you! Think about removing this line before carrying on, so that your users don't get greeted at each refresh!

CUENTA

Setting up your live JavaScript console in the browser

Every modern browser has a JavaScript console, which allows you to execute pieces of JavaScript code live as your webpage is up. Search online where yours is in your browser (for some browsers, it may require to change your settings first).

Once you've found it, type `alert('Hello!');` in it. The dialogue box should show up. Make sure the console remains open for the rest of the project, as the JavaScript console will also tell you if there are code errors in your .js file.

Inserting a large image in the document

Well, you know how to do that! Find a large image online (at least 800 pixels wide), and use its URL to insert it inside your document, wherever you want. You will notice that even if it is wider than your website, it won't overflow, and will resize nicely, because we set a CSS rule for you, to constrain its size.

Then, we want this image to be small at first; and for this too, we already set a CSS rule for you. Simply add this attribute to your `<img>` tag: `class="small"` in your HTML code. Refresh, and you see it's much smaller.

Ayuda

The whole point of this part of the project, is that you will use JavaScript to remove this `"small"` class (and therefore the image will grow) or putting it back (therefore it will get back to being small) when the user clicks.

Isolating the HTML element inside a variable

This challenge is all about "listening" to the event "user clicked on this image", and then changing the image's size when it happens; therefore, the first thing to do is to is to capture this image HTML element in a JavaScript variable that we'll be able to manipulate.

The smartest way to capture one particular HTML element with JavaScript is to give it a unique ID (it's like giving it a first name!), and catch it by giving JavaScript this ID. It's easy to do, you can simply add the attribute `id="smart_thumbnail"` to your `<img>` tag, and there it is, now identifiable!

Then, after refreshing your page, you can run this in your browser's JavaScript console, and you'll see that it indeed returns an element, which is your `<img>` element:

```
document.getElementById("smart_thumbnail");
```

You can copy-paste this line into your `behavior.js` file; but you also have to capture the result in a variable called `thumbnailElement`, so that we can manipulate it more easily later in our JavaScript code. All together, it will look like this:

```
var thumbnailElement = document.getElementById("smart_thumbnail");
```

Catching the click event

Add these lines next in your JavaScript file:

```
thumbnailElement.addEventListener("click", function() {
  // write here
});
```

We did mention before that you could get code to execute on user events; well, code that you write instead of `// write here` will execute whenever the user clicks on your image.

As code that happens when the user clicks on the image, write `alert("I saw you click!");`. Refresh the webpage, then scroll down in your content, and see the dialog box pop up each time you click!

That's great because it means you're catching the click event properly! But a dialog box is not what we want to do, so you should remove your alert statement before carrying on. But note that the rest of your code in this part of the project will be executed when the click is captured, and therefore you will all have to write it there.

Make the image big!

CUENTA

Make the image big!

To make the image bigger or smaller, you have to change its class. In your JavaScript console, run this code:

```
var thumbnailElement = document.getElementById("smart_thumbnail");
thumbnailElement.className;
```

This displays what the class currently is; it should be `"small"`, unless you changed it since.

To make it big, you should remove the class, by running this:

```
thumbnailElement.className = "";
```

To make it small again, you can put it back:

```
thumbnailElement.className = "small";
```

See how it changes from small to big? You should put the line that makes it big in your JavaScript file so that it executes when the user clicks.

One more thing...

One more thing...

After clicking on the image, now that your image is big, your user can... well, do nothing more! Wouldn't it be cool if the image went back to being small when she clicks again?

This will be an extra challenge for you, if you have some time left after the project!

To make it work, you will need to execute some code ***if*** the image is big or ***else*** some other code if not. To that end, you can use what are called an `if` statement and an `else` statement, about which you can learn more here. The `if` statement to check if the image is big should look like this (pay attention to the double-equal sign):

```
if (thumbnailElement.className == "") {
        // write here the code that will execute if the image is big
}
```

Once you like what you have, feel free to move on to the next part of the project.