

## 2.6.6 Adjust for Mobile Devices

The hover animation looks great, but this only applies to desktop users. Hover effects don't really work that well on touchscreen devices. How would mobile users be able to see the trainers' names, then? The best user experience might be to always display the trainer information on small screens, which we can do by using media queries.

### HIDE PRO TIP

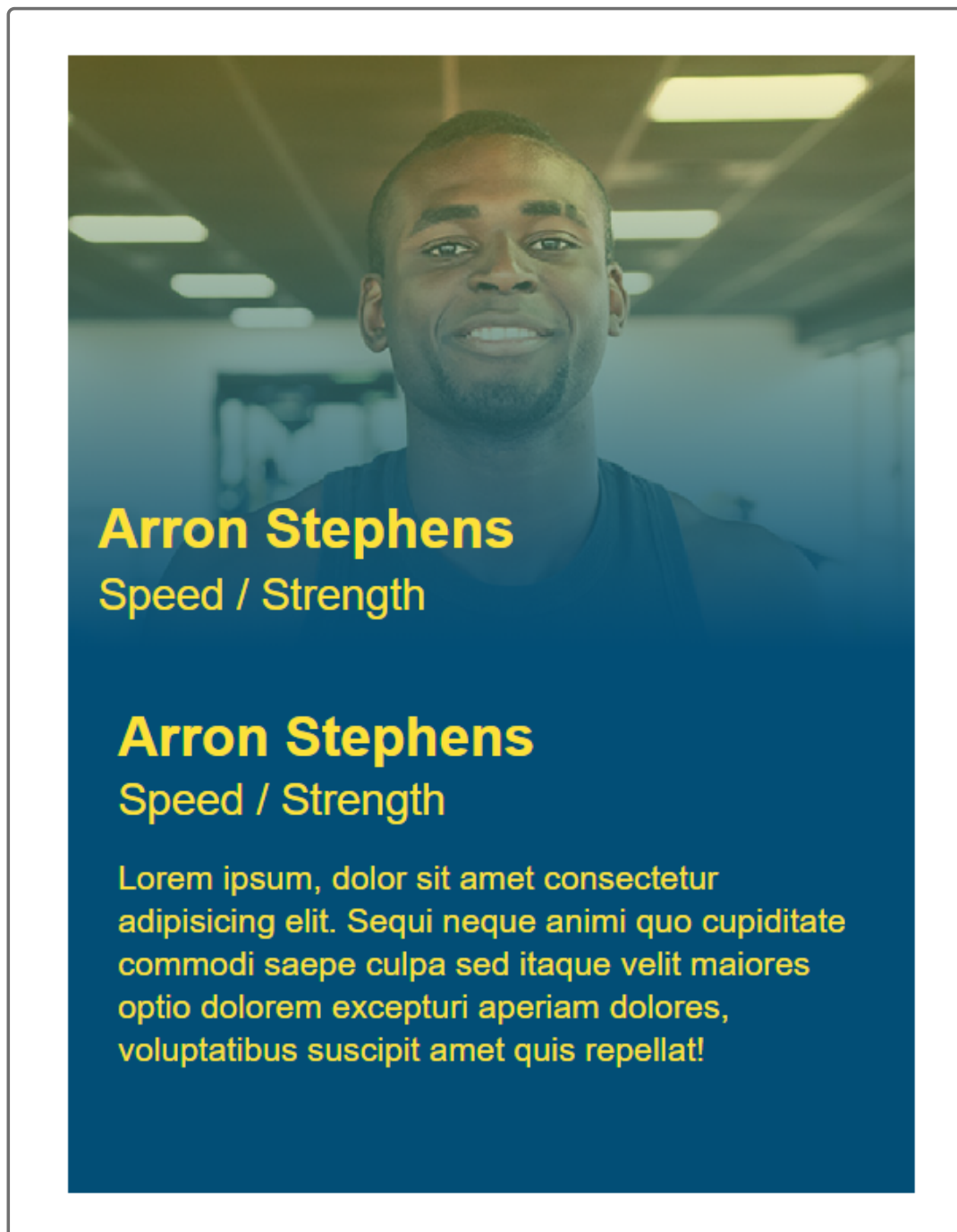
To easily test your webpage on multiple screen sizes, use the "Toggle device toolbar" button in Chrome's DevTools to simulate common mobile devices.

First, let's re-add the trainer name and role to the `trainer-bio` container:

```
<div class="trainer-bio">
  <!-- re-add name and role here -->
  <h3>Arron Stephens</h3>
```

```
<h4>Speed / Strength</h4>
<p>
  Lorem ipsum, dolor sit amet consectetur adipisicing elit. Sequi ne
</p>
</div>
```

This means we now have duplicate content—the `<h3>` in `trainer-img` and the `<h3>` in `trainer-bio`—as you can see in the following screenshot:



We'll hide one or the other based on the screen size. In the normal desktop view, we want the name and role to be part of the hover animation, so the `trainer-bio` info can be hidden by default. The style sheet already has rules specified for these elements:

```
.trainer-bio h3 {  
  font-size: 28px;  
}  
  
.trainer-bio h4 {  
  font-weight: lighter;  
  font-size: 22px;  
  margin-bottom: 15px;  
}
```

We just need to add one more declaration to hide them. Try each one separately to see which one produces the best results:

```
opacity: 0;  
visibility: hidden;  
display: none;
```

Once you've seen them in action, can you tell what the difference is between `visibility: hidden` and `display: none`?

The `visibility: hidden` declaration turns the element invisible but doesn't remove it from the flow. The `display: none` declaration makes it seem like the element doesn't exist at all. That's the one we want in this case.

Next, we'll need to add the same `display: none` declaration to our gradient pseudo-element and `trainer-img` name and role elements. Of course, we only want to hide these elements on smaller touchscreen devices.

In an earlier lesson, we declared a set of media query rules for tablets:

```
@media screen and (max-width: 768px) {  
  
}
```

Inside this media query, add additional rules that hide/remove the gradient, name, and role.

### HIDE HINT

Add `display: none;` to a CSS rule that encompasses `.trainer-img h3`, `.trainer-img h4`, and `.trainer-img::after`.

To verify that we did everything correctly, the hover animation should only work in full view. Smaller screen sizes shouldn't show any name/role information (yet). Remember to use the "Toggle device toolbar" option in the DevTools to test it out.

Next, in the same media query where we hid the hover elements, let's reveal `.trainer-bio h3` and `.trainer-bio h4`. If `display: none` removes an element entirely from the flow of the webpage, what display value would bring it back?

### HIDE HINT

Think about what type of element an `<h3>` is—block or inline—and how it normally flows. Would `display: block` or `display:`

`inline` be more appropriate?

When you have the mobile version working, take a moment to celebrate! This was a challenging lesson. Building animations can be tricky, but they can also make clients very happy and that is our ultimate goal.

Now that we're done with this task, let's go through our Git routine:

1. Type `git status` to verify that the correct files were modified.
2. Type `git add -A` or `git add .` to stage any changed files.
3. Commit the changes using `git commit -m "trainer hover animation"`.
4. Push the branch to GitHub by using `git push origin feature/animation`.
5. Use `git checkout develop` to switch branches.
6. Type `git merge feature/animation` to merge the new feature/changes into the develop branch.
7. Push the updated develop branch to GitHub by using `git push origin develop`.
8. Close the GitHub issue!