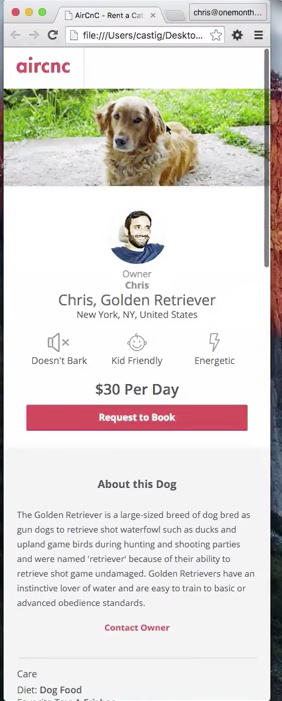
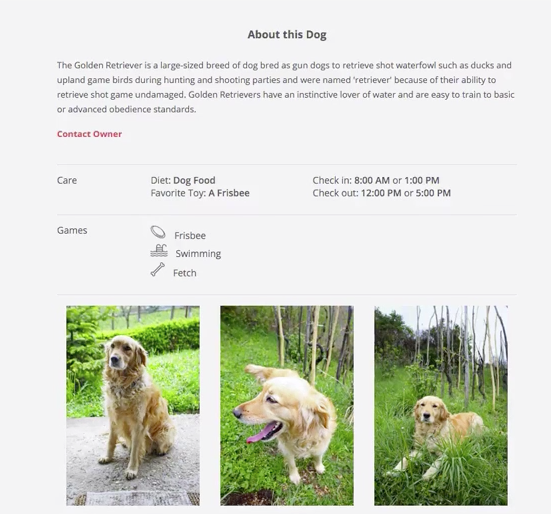
**Is It Time for Dogs to Have a Social Network of Their Own?**

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 1 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/is-it-time-for-dogs-to-have-a-social-network-of-their-own?autoplay=1#step-notes)
* [Discussion (3)](https://onemonth.com/courses/html/steps/is-it-time-for-dogs-to-have-a-social-network-of-their-own?autoplay=1#discussion)



In this lesson, we get started on our final project—AirCnC—a social network for cats and canines that looks very similar to AirBnB. In this lesson I’ll provide you with a design mockup (exactly like how it’s done in the real-world if you were working with a paying client). You’ll receive a design for both the mobile and desktop views. What’s great about this lesson is that if you’re up for a challenge? Then you can jump right on your own. If you get stuck? We’re here to help.

If you would like to jump right into coding, here's a couple mock-ups of how AirCnC should look on desktop and mobile.

The mobile view:

:The desktop view

# All Designs Go To "Section"

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 3 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/all-designs-go-to-section?autoplay=1#step-notes)
* [Discussion (2)](https://onemonth.com/courses/html/steps/all-designs-go-to-section?autoplay=1#discussion)

In this lesson, we start thinking strategically about our website's layout and how we'll break it up into different elements like sections, header, divs, etc.

Here's the HTML Code:

<!DOCTYPE html>

<html>

<head>

<title>AirCnC - Rent a Cat or Dog</title>

<link rel="stylesheet" href="css/reset.css">

<link rel="stylesheet" href="css/style.css">

<meta name="description" content="AirC&amp;C is the premiere site renting cute cats and canines">

<link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700,600' rel='stylesheet' type='text/css'>

<meta name="viewport" content="width=device-width, initial-scale=1">

</head>

<body>

<header>

<img src="images/logo.png" alt="AirCnC Logo" class="logo">

</header>

<div class="hero">

<img src="images/dog-hero-image.png" alt="Terry Golden Retriever">

</div>

</body>

</html>

And here's the CSS:

.button {

background: rgb(255, 89, 95);

color: white;

text-decoration: none;

display: block;

padding: 0.75rem 5rem;

border-bottom: 1px solid rgb(227, 25, 29);

border-radius: 2px;

width: 95%;

margin: 0 auto;

}

/\*

\* Header and .hero Styles

\*/

header {

background: #FFF;

}

header .logo {

border-right: 1px solid #DCE0E0;

}

.hero img {

width: 100%;

}

Keep in mind how the mobile version of the site relates to the desktop view. That will be important in how you create (and later manipulate) sections, containers, and divs.

A lot of the specifications for class names and other stylistic information is in the comps, so be sure to refer back to them.

Code as much as you can before you look at the answers!

# Get Friendly With The Layout

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 4 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/get-friendly-with-the-layout?autoplay=1#step-notes)
* [Discussion (4)](https://onemonth.com/courses/html/steps/get-friendly-with-the-layout?autoplay=1#discussion)

Now it's time to write the responsive code! We get started on the website by writing the HTML markup and we add the text.

Here's what the structure of the page looks like before we add any images or assets:

<section class="info>

<div class="container">

<div=class="profile">

<div class="owner-details">

</div><!-- .owner-details -->

<div class="pet-details"?

<div class="icon">

</div?<!-- .icon -->

</div><!-- .pet-details -->

</div><!-- .profile -->

<div class="booking">

</div><!-- .booking -->

</div><!-- .container -->

</section><!-- .info -->

And here's what it can look like once you've filled in the information:

<body>

<header>

<img src="images/logo.png" alt="AirCnC Logo" class="logo">

</header>

<div class="hero">

<img src="images/dog-hero-image.png" alt="Terry Golden Retriever">

</div>

<section class="info">

<div class="container">

<div class="profile">

<div class="owner-details">

<img src="images/chris-castiglione.png" alt="Owner image">

<h2 class="relationship">Owner</h2>

<h2 class="name">Chris</h2>

</div><!-- .owner-details -->

<h1>Chris, Golden Retriever</h1>

<p class="location">New York, NY, United States</p>

<div class="icons column">

<div class="icon">

<img src="images/icons/mute.svg" alt="Doesn't Bark">

<h5>Doesn't Bark</h5>

<img src="images/icons/baby.svg" alt="Kid Friendly">

<h5>Kid Friendly</h5>

<img src="images/icons/lightning.svg" alt="Energetic">

<h5>Energetic</h5>

</div><!-- .icon -->

</div><!-- .pet-details -->

</div><!-- .profile -->

<div class="booking">

<h2>$30 <span class="time">Per Day</span></h2>

<a href="#" class="button">Request to Book</a>

As you code, make sure you're being descriptive! Classify your assets not just in a way that makes sense, but in the way that most fully and succinctly conveys what you want them to do.

Also, make sure your writing with an eye towards CSS. Adding classes now will make it easier for you to customize and make your code responsive later.

**SVG -**Scalable Vector Graphics. They look good at any size!

You can check out other icons at the [Noun Project](https://thenounproject.com/).

**Icons**  
[Lightning Bolt](https://thenounproject.com/search/?q=lightning&i=11769); [Baby](https://thenounproject.com/search/?q=child&i=108606); [Mute](https://thenounproject.com/search/?q=quiet&i=133990); [Bone](https://thenounproject.com/search/?q=bone&i=165338); [Swimming Pool](https://thenounproject.com/search/?q=swimming&i=173084).   
  
**Profile Pic**  
The owner profile pic is from [UI Faces](https://uifaces.co/).   
  
**Images**  
Hero image: [One](http://www.istockphoto.com/photo/close-up-of-golden-retriever-looking-at-camera-53897762).   
Small images: [One](http://www.istockphoto.com/photo/close-up-of-golden-retriever-in-nature-53897892), [Two](http://www.istockphoto.com/photo/golden-retriever-in-grass-53898250), [Three](http://www.istockphoto.com/photo/golden-retriever-sitting-53897992)

# Responsive Design Layout by Example

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 5 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/responsive-design-layout-by-example?autoplay=1#step-notes)
* [Discussion (7)](https://onemonth.com/courses/html/steps/responsive-design-layout-by-example?autoplay=1#discussion)

We now turn to CSS and start styling the first section of the webpage with a lot of trial and error, tweaking, and tasty beverages.

If you need a guideline for how the CSS can look, we've included all of it below:

/\*

\* Info

\*/

.info {

background: #FFF;

text-align: center;

padding: 25px;

}

.info h1 {

margin-bottom: 5px;

}

.info .profile {

margin-bottom: 24px;

}

.info .owner-details img {

border-radius: 50%;

margin-bottom: 6px;

}

.info h2.relationship, .info h2.name {

color: #909597;

margin-bottom: 3px;

}

.info h2.name {

font-weight: bold;

}

.info p.location {

margin-bottom: 20px;

}

.info .icon {

float: left;

width: 33.3%;

}

.info .icon img {

height: 30px;

margin-bottom: 3px;

}

.info .booking h2 {

font-weight: 600;

font-size: 22px;

margin-bottom: 13px;

}

.info h2.price time {

font-size: 16px;

}

# CSS Media Queries: How To Reach Your Breakpoint(s)

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 6 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/css-media-queries?autoplay=1#step-notes)
* [Discussion](https://onemonth.com/courses/html/steps/css-media-queries?autoplay=1#discussion)

Our website looks pretty good on small screens. Now it's time to expand it and make it look just as good on the desktop. In this lesson, you’ll learn how to set up CSS media queries so that we can begin coding the desktop version of our AirCnC site.

Here's an example of what a media query looks like inside your CSS .style sheet.

@media only screen and (min-width: 1060px) {

.info{

background: red;

}

}

Media queries are conditional. The code you write within them kicks in when, here, a certain screen size is achieved. If you want to learn more, you can also check out Mozilla articles on media queries, @[Media](https://developer.mozilla.org/en-US/docs/Web/CSS/@media) and [Using Media Queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries).

# Layout Continued: Everything You Code In A Box To The Left

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 7 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/layout-continued-everything-you-code-in-a-box-to-the-left?autoplay=1#step-notes)
* [Discussion (2)](https://onemonth.com/courses/html/steps/layout-continued-everything-you-code-in-a-box-to-the-left?autoplay=1#discussion)

We continue with the task of making our website desktop-friendly (using media queries). We enclose the page's content in a container and style it so that it is centered on the page. You'll learn some tips to consider when styling divs and containers.

**Tip**:  Whenever you create a container, set a width and your margin to zero auto. Like so:

.container {

width: 1040px;

margin: 0 auto;

}

There's no one right answer for how to set up your code, as long as your able to get your page looking great. But if you'd like to see the way I have set up my media query? Check it out:

@media only screen and (min-width: 1060px) {

.container {

width: 1040px;

margin: 0 auto;

}

.column {

float: left;

}

.button {

width: 275px;

margin: auto;

}

.info h1 {

margin-bottom: 11px;

}

.info h2 {

margin-bottom: 3px;

}

.info .profile {

width: 660px;

border-right: 1px solid #DCE0E0;

}

.info .booking {

width: 380px;

}

.info .owner-details {

width: 20%;

}

.info .pet-details {

width: 80%;

text-align: left;

}

.info .pet-details .icon {

width: inherit;

margin-right: 50px;

}

.info .pet-details .icon img {

display: block;

margin: 3px auto 3px;

}

}

# Rems vs. Pixels: When To Be Firm

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 8 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/rems-vs-pixels-when-to-be-firm?autoplay=1#step-notes)
* [Discussion (4)](https://onemonth.com/courses/html/steps/rems-vs-pixels-when-to-be-firm?autoplay=1#discussion)

We're ready to convert our pixels to rems and make our code as responsive as possible. But before we do, let's talk about when and why to convert pixels to rems. You can certainly use rems as the only unit of measure in you're CSS, but this would entail performing a lot of math. By using media queries and thus being able to specifically target different sized screens, you can reduce some of you're cognitive load by using pixels instead of rems.

**Why do we use pixels for media queries and rems for the values?**Rems are a unit of measurement relative to the root element (aka the HTML defaults). That way we can adjust sizing up or down just by changing the font-size on the root element. Most of the time we'll want to do this on text sizes, and occasionally padding or margin.

Rems are great for adjusting the aesthetics of our page. But the downside to this is that it makes us do a lot of math in our heads. With media queries we don't need to worry about the size relative to the root element, so we can save ourselves the mental math and just use pixels when all we're looking to do is kick in elements at a specific size value.

# Your Responsive Homework

[Week 4: Learn How to Build AirBNB Using Responsive HTML & CSS](https://onemonth.com/courses/html#how-to-build-airbnb-html-css) · Lesson 9 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/fetch-the-rest-of-the-site?autoplay=1#step-notes)
* [Discussion (12)](https://onemonth.com/courses/html/steps/fetch-the-rest-of-the-site?autoplay=1#discussion)

Keep coding, and finish up the site so cats and canines alike finally get the responsive platform they deserve. We've left some tasks that you need to implement on your website and a few extra-credit ones that you can also tackle. If you get stuck, the community is a great place to reach out to and brainstorm ideas with.

## Your next challenge is to finish AirCnC. You should accomplish the following:

1. Finish the project for the mobile view.
2. Add the CSS to style it for the desktop view.

## Extra credit features you could build:

1. Add a [lightbox](http://lokeshdhakar.com/projects/lightbox2/) effect on the images.
2. Hook up the red "Request to Book" button to [Stripe](https://stripe.com/docs/api#intro) or an email address using the [mailto](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Email_links" \t "_blank) attribute.
3. Personalize it! Put in your own personal info, or a friend of your's dog, and make them a site for they're dog. Or cat! The internet always needs more cats.

# Get Your Site Live with Github Pages!

[Bonus](https://onemonth.com/courses/html#bonus) · Lesson 4 ·

* [Lesson Notes](https://onemonth.com/courses/html/steps/get-your-site-live-with-github-pages?autoplay=1#step-notes)
* [Discussion (11)](https://onemonth.com/courses/html/steps/get-your-site-live-with-github-pages?autoplay=1#discussion)

Let's show the whole internet what you've done!