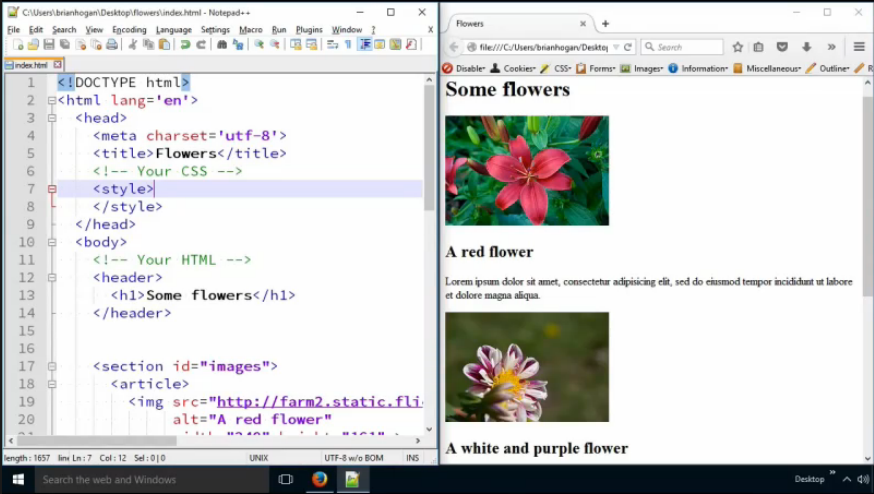
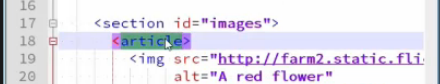
**Tutorial: Floats and Clears**

Hi, today we're going to look at how we use CSS to lay out elements on a page.



I have this basic web page here that contains three images of flowers. Each image and its corresponding description is wrapped in an `article` element.



First, let's talk about block elements and inline elements. The image element is an inline element. However, the article itself is a block element, as are the h2 elements and the paragraph elements below. So, it looks like the image is on its own line, since the elements that follow it are block elements. This gives us that linear "top down" structure that we have on our page.

As you can see we waste a lot of space on this page. It would be great if we can organize this differently; pull the title and description up next to the picture.

Block elements can have a width, height, margin, and padding.

If we give the *article* a width. Type the following CSS in the styles tag in the head:

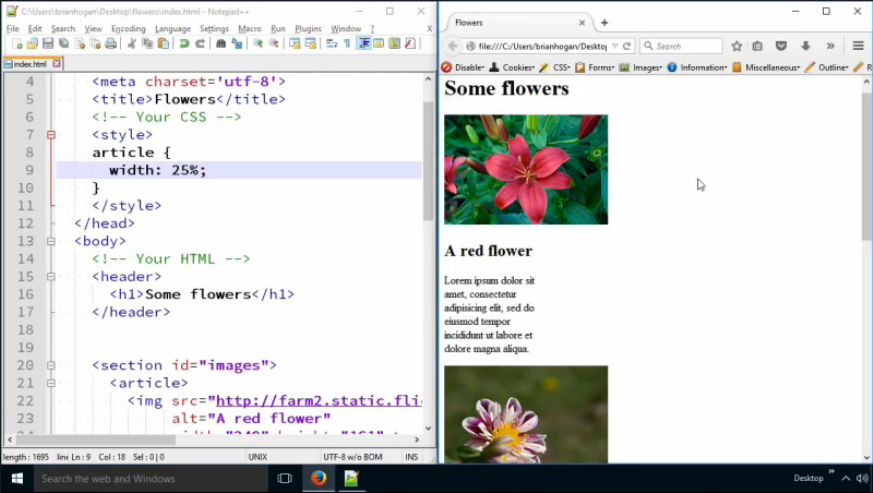
**article {**

**width: 25%;**

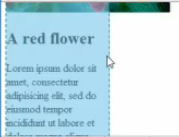
**}**

**Refresh the browser.**

You'll notice that it automatically constrains everything inside 25% width of the page. That is except for the image; images will always overflow a width if the image is larger than the width settings.

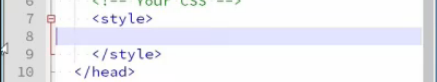


The block elements h2 and p, by default, are always going to expand to the width of their container. They have a width of 100% by default. So, by default, they fill to fit the width of their container. The container is 25% of the width of the page. So, the contents of the paragraph are constrained. But since they take up 100% of the container, they end up also taking up 25% of the width. Block elements will always stretch to fill their container. Before we applied the 25% width on the article, it stretched to fill the width of the page, and so did its children.

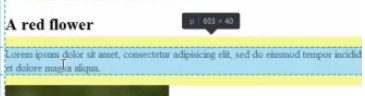


Let's remove that constraint again and put things back.

**Delete the contents in the style tag.**



**Refresh your browser.** The p and h2 should be the full width of the browser again.



We can use the *display* property in CSS to change how elements are displayed.

We can use **display: inline** to make an element an inline element.

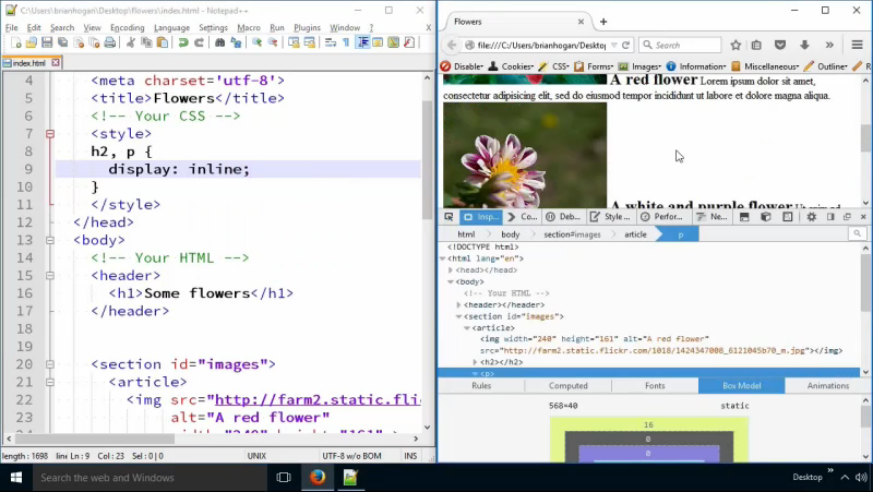
We can make our *h2* and *p* elements inline elements. Type the following CSS in the styles tag in the head:

**h2, p {**

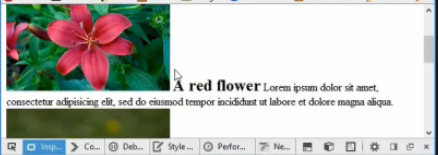
**display: inline;**

**}**

**Refresh the browser.**



And now there's no line break between the heading and the paragraph, and they fall on the same line as the image.



But what if we wanted the image to be on the left and the text to be on the right?

Well we don't use the *display* property for that. Instead, we use "floats" and "clears."

By default, the browser renders everything top-to-bottom, in what we call the "document flow." But we can take things out of the document flow a number of ways. We can position things by X and Y coordinates which is called absolute positioning, or we can simply tell elements to "wrap" around other elements.

Let's see how this works.

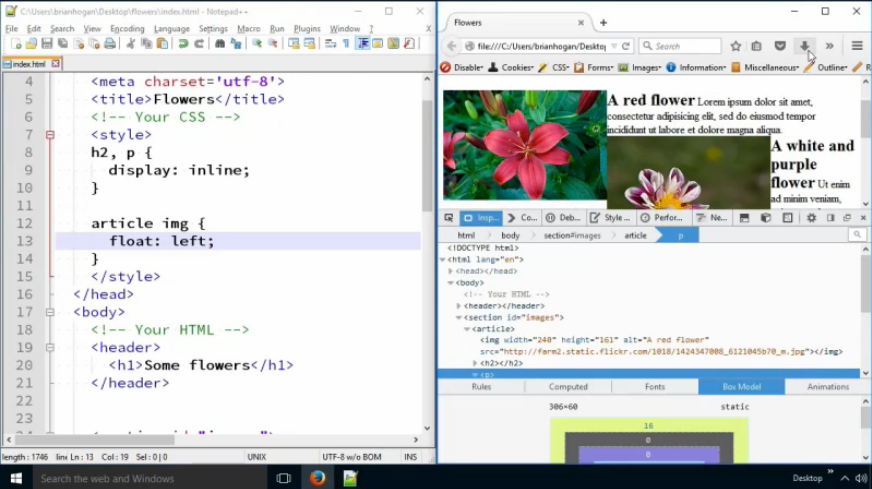
First, we'll take the *img* elements inside of the articles and we'll tell them to "float left." Add the following CSS to the styles tag in the head:

**article img {**

**float: left;**

**}**

**Refresh the browser.**



This will pull them out of the document flow and everything that follows it will sit to the right. The image will be pulled left and everything that follows will fall right.

And now we can see that everything sits to the right of the image, and we get this cascading "staircase" like effect on the page. That's not exactly what we want.



We must "turn off" the floating behavior again, and we do that on the element that we want to "return" to normal behavior. In our case, we want each article to start on its own line, so it makes sense that the article will be the element that we "clear" the floats.

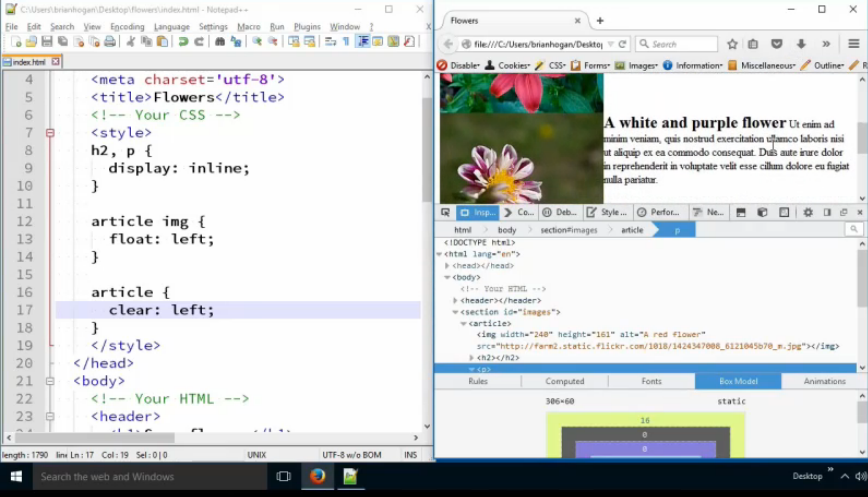
Below is the code we use to clear floats. Type in the following CSS in the styles tag in the head:

**article {**

**clear: left;**

**}**

**Refresh the browser.**



We're clearing the floats on the left with this.

Now, if you look, that footer notice is sitting up next to the text of the last article.



We can fix that by applying a clear to the footer. Our clear only applies to article elements, not the footer, and so the footer gets pulled up. Type in the following CSS into the styles tag in the head:

**footer {**

**clear: both;**

**}**

This time we're using both because for the footer, we might not know how things were floated. We can float things on the right, too. By using both, we're saying "put things back to the defaults."

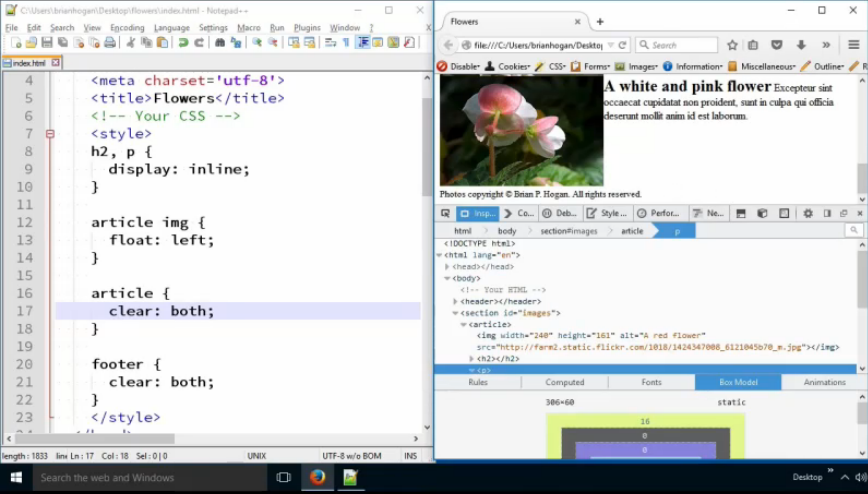
Let's change the previous clear to use **both** as well. Find the article in the styles tag and change the clear to **both**.

**article {**

**clear: both;**

**}**

**Refresh the browser.**



And now you can see that everything is back where we wanted it, although the images are snug up against each other. It would be nice to get some space between each article.

Well, we know that we can space elements by adding a margin. Let's try that - we'll add a bottom margin to the article. Type in **margin-bottom: 10px;** to the article in the Styles tag in the head:

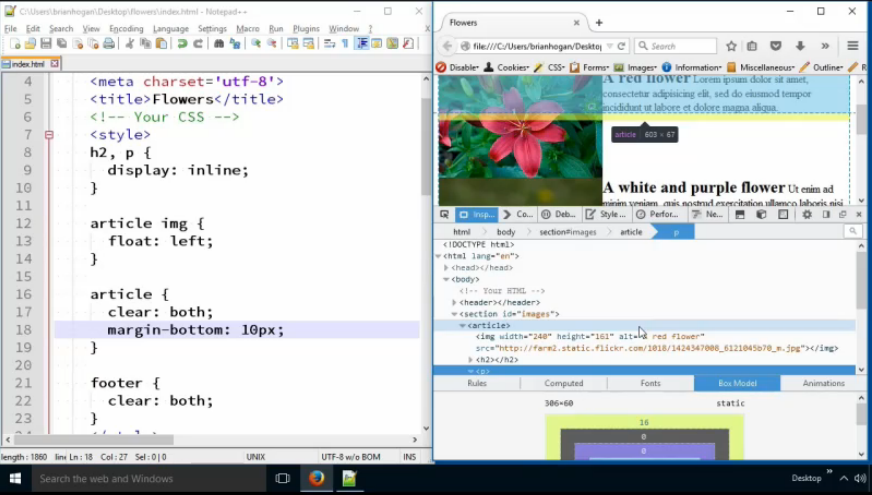
article {

clear: both;

**margin-bottom: 10px;**

}

But if we **refresh**, we'll see that this doesn't do anything at all. Unfortunately, we're dealing with something called "margin collapse." Notice the yellow row on the right side of the image below. The margin was added.



You see, we floated the contents of the article. We've taken them out of the normal document flow. So now the article has collapsed and its margins are being added to the collapsed height.

Kind of annoying, and not what we expected. There are a lot of ways to fix this. Let's look at a simple one: we give the article an explicit height, or at least a minimum height. Type in the following CSS into the article in the styles tag in the head:

**article {**

clear: both;

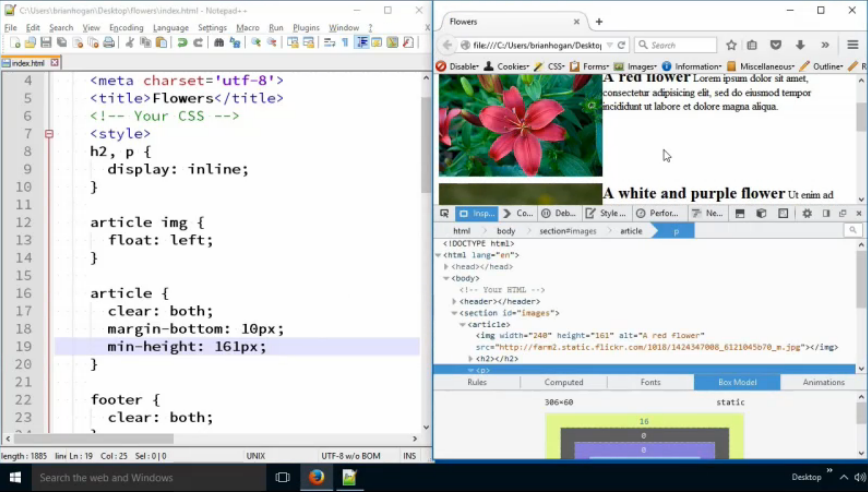
margin-bottom: 10px;

**min-height: 161px;**

**}**

That minimum height is the height of the images in this document.

**Refresh the browser.**



But if we look at the space, that looks LOT more than 10 pixels. Let's inspect.

It turns out that the h2 elements have a margin on them. Remember: the browser has its own default styles.

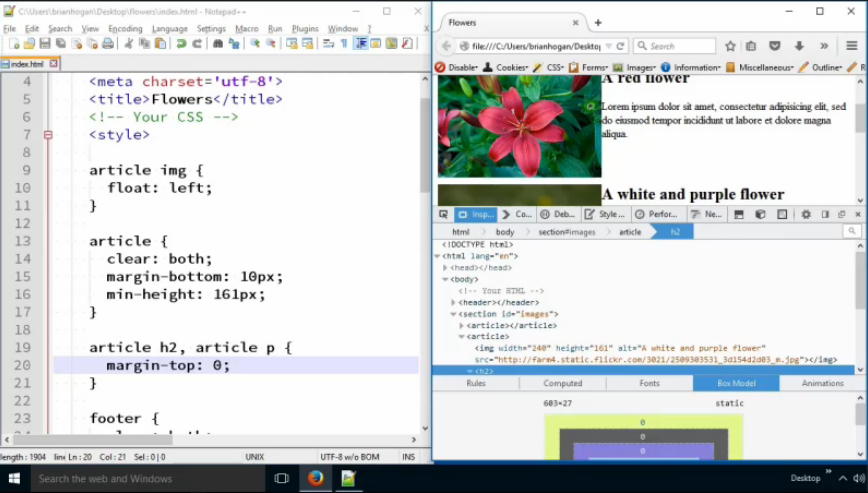
We must override them, so let's set all the *h2* and *p* tags inside of the article so they have no top margin. Type in the following CSS into the styles tag in the head:

**article h2, article p {**

**margin-top: 0;**

**}**

**Refresh the browser.**



Now, the text is a little too close to the images, so can we adjust the h2 and p elements a bit? Add magin-top:0; to the article h2, article p CSS:

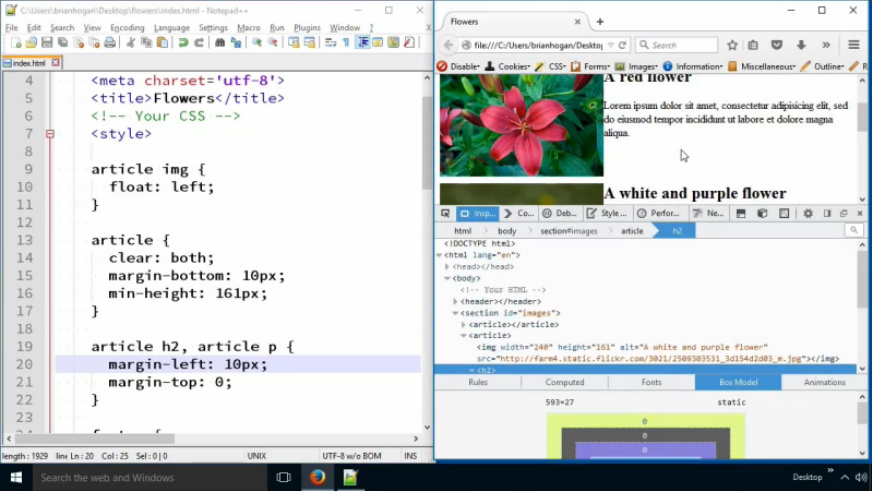
**article h2, article p {**

**margin-left: 10px;**

margin-top: 0;

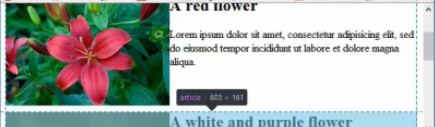
**}**

**Refresh the browser.**



Doesn't look like it changed. Let's look at the inspector...

It looks like the edge of the element moved over, but the TEXT content didn't! Another example of margin collapse.



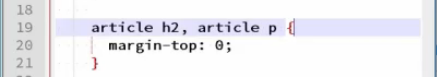
We're going down a rabbit hole here, and it's not looking pretty.

What we really want is space around the image, so the images don't touch and the text doesn't touch. So, here's how we can achieve this:

First, we'll **remove the left margin on the p and the h2**. And we'll **remove the bottom margin between the articles**.

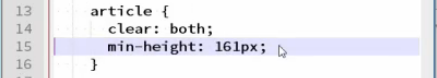
Delete the margin-left in the **article h2, article p {**

**margin-left: 10px;**



Delete the margin-bottom in the **article {**

**margin-bottom: 10px;**



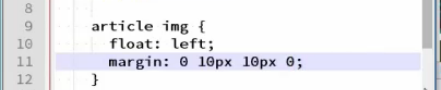
Next, we'll just add margin on the right and bottom of the *img* itself.

Add margin to the **article img**:

**article img {**

float: left;

**margin: 0 10px 10px 0;**



This will eliminate the issues we're having.

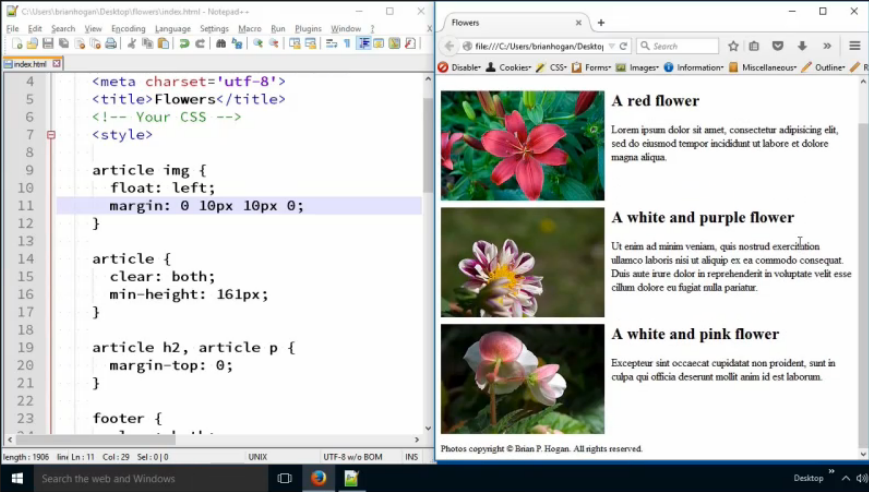
You can now **remove the h2,p styles in the styles tag**, we don’t need that anymore. Delete the following CSS:

**h2, p {**

**display: inline;**

**}**

**Refresh the browser and make it full screen.**



How did I know this would work? Experience through experimentation. And you'll gain that experience by doing exactly what I did - writing some code and using the web inspector.

**Wrapping Up**

In this tutorial, you learned about the *display* property and how it affects the default styling of elements. Some elements are *block* elements, which means they start on their own line and have a default width of 100%. And they can have height, width, margin, and padding applied. Inline elements don't have widths, and they sit on the same line as other elements.

You also learned about how to use *float* to pull an element out of the normal document flow, and then how to use *clear* to put things back the way they were. This is a technique you'll use from time to time to bring images into text and make text wrap around them.

Study this for a bit before moving on to the next lesson.

Here is the finalized HTML & CSS for this tutorial:

<!DOCTYPE html>

<html lang='en'>

<head>

<meta charset='utf-8'>

<title>Flowers</title>

<style>

article img {

float: left;

margin: 0 10px 10px 0;

}

article{

clear: both;

min-height: 161px;

}

footer{

clear: both;

}

article h2, article p{

margin-top: 0;

}

</style>

</head>

<body>

<header>

<h1>Some flowers</h1>

</header>

<section id="images">

<article>

<img src="http://farm2.static.flickr.com/1018/1424347008\_6121045b70\_m.jpg"

alt="A red flower"

width="240" height="161" >

<h2>A red flower</h2>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do

eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</article>

<article>

<img src="http://farm4.static.flickr.com/3021/2509303531\_3d154d2d03\_m.jpg"

alt="A white and purple flower"

width="240" height="161" >

<h2>A white and purple flower</h2>

<p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris

nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in

reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla

pariatur.</p>

</article>

<article>

<img src="http://farm2.static.flickr.com/1108/1423464589\_e42c6182ac\_m.jpg"

alt="A white and pink flower"

width="240" height="161">

<h2>A white and pink flower</h2>

<p>Excepteur sint occaecat cupidatat non proident, sunt in culpa qui

officia deserunt mollit anim id est laborum.</p>

</article>

</section>

<footer>

<small>Photos copyright &copy; Brian P. Hogan. All rights reserved.</small>

</footer>

</body>

</html>