[**HTML5 Responsive Image Slider With Captions**](http://demosthenes.info/blog/831/HTML5-Responsive-Image-Slider-With-Captions)

[css](http://demosthenes.info/blog/css) / [galleries](http://demosthenes.info/blog/css/galleries)

[Updated](http://demosthenes.info/blog/831/HTML5-Responsive-Image-Slider-With-Captions) [Comment 38](http://demosthenes.info/blog/831/HTML5-Responsive-Image-Slider-With-Captions#disqus_thread)

Estimated reading time: 2 minutes, 54 seconds

After students and site visitors see the [responsive CSS3 Image Slider](http://demosthenes.info/blog/627/Make-A-Responsive-CSS3-Image-Slider), one of their first questions is usually “how can I add a caption for each image?”

The solution is fairly straightforward, especially if you use HTML5 semantic markup. As this already uses the basic code, I won’t repeat an explanation of the animation itself, which you can read in the original article. Instead, I’ll focus on how to style and use the new elements.

**The HTML**

By comparison to the original example, the markup reflects the fact that each picture is now surrounded by a [<figure>](http://demosthenes.info/blog/413/Captioning-images-in-HTML5) element:

<div id="captioned-gallery">

<figure class="slider">

<figure>

<img src="hobbiton-lake.jpg" alt>

<figcaption>Hobbiton, New Zealand</figcaption>

</figure>

<figure>

<img src="wanaka-drowned-tree.jpg" alt>

<figcaption>Wanaka, New Zealand</figcaption>

</figure>

<figure>

<img src="utah-peak.jpg" alt>

<figcaption>Utah, United States</figcaption>

</figure>

<figure>

<img src="bryce-canyon-utah.jpg" alt>

<figcaption>Bryce Canyon, Utah, United States</figcaption>

</figure>

<figure>

<img src="hobbiton-lake.jpg" alt>

<figcaption>Hobbiton, New Zealand</figcaption>

</figure>

</figure>

</div>

(I’ve deliberately left the alt attribute blank in this case to save on space, but you should enter an appropriate description for [accessibility](http://demosthenes.info/blog/accessibility)).

**The CSS**

The <figure> elements effectively take the place of the images in the original slider, as shown in the CSS (*sans* [vendor prefixes](http://demosthenes.info/blog/217/CSS-Vendor-Prefixes-and-Flags) to save space):

div#captioned-gallery { width: 100%; overflow: hidden; }

figure { margin: 0; }

figure.slider {

position: relative; width: 500%;

font-size: 0; animation: 40s slidy infinite;

}

figure.slider figure {

width: 20%; height: auto;

display: inline-block;

position: inherit;

}

font-size and margin are set to 0 to suck all the air out of the parent <figure> element and its children, which are placed side-by-side by applying display: inline-block. Those same children inherit relative positioning from their parent, which will come in handy in just a moment.

figure.slider img { width: 100%; height: auto; }

figure.slider figure figcaption {

position: absolute; bottom: 0;

background: rgba(0,0,0,0.3);

color: #fff; width: 100%;

font-size: 2rem; padding: .6rem;

}

I’ve used the old trick of positioning an [absolute element inside an relative container](http://demosthenes.info/blog/134/CSS-Positioning-relative-the-underappreciated) to ensure that each caption is always at the bottom of its associated image. font-size is reset, and the caption given a color and [rgba](http://demosthenes.info/blog/44/Color-in-CSS#rgb) background so that the text always remains readable, no matter what photograph is behind it.

**Making A Responsive Caption**

While the images are responsive, the caption text is not. There are three ways of dealing with that:

1. Size the text in [vw](http://demosthenes.info/blog/660/Using-vw-and-vh-Measurements-In-Modern-Site-Design) units rather than [rem](http://demosthenes.info/blog/673/REM-Not-the-Band) or em.
2. Resize the text at appropriate breakpoints using @media queries.
3. Use an [@media](http://demosthenes.info/blog/154/CSS3-media-queries) query to hide the captions entirely when the viewport is too small.

In this case, I’ve chosen the second option:

@media screen and (max-width: 500px) {

figure.slider figure figcaption { font-size: 1.2rem; }

}

**Variations**

Alternatively, you could set the captions to be invisible by default, at least until the user hovered over an image:

figure.slider figure figcaption {

position: absolute; bottom: -3.5rem;

background: rgba(0,0,0,0.3);

color: #fff; width: 100%;

font-size: 2rem; padding: .6rem;

transition: .5s bottom;

}

figure.slider figure:hover figcaption { bottom: 0; }

This would bring up the caption from the bottom of each image. However, when the slider moved to the next image, the <figcaption> for *it* would not yet be activated. To solve that, we could make a similar declaration using an adjacent selector:

figure.slider figure:hover + figure figcaption { bottom: 0; }

**Conclusion**

While this works, there’s quite a bit of code repetition and redundancy. In the [CSSslidy script](http://demosthenes.info/blog/871/CSS-Slidy-20-Captions--more) I show how to eliminate that using progressive JavaScript.