

RESPONSIVE VIDEO

“Fluid” means that a video should be able to increase or decrease in size as needed **based on the width of a parent element or the viewport size.**

In theory, making fluid videos should be easy. Unfortunately, this is **not the always the case.**

Many movies are embedded within HTML pages using **Youtube, Vimeo or Viddler** players.

These players use the **iframe, object and embed elements** - which do not respond to min and max size as easily as the video element.

Luckily, there is a **CSS based solution**, created by Thierry Koblentz.

<http://alistapart.com/article/creating-intrinsic-ratios-for-video>

Understanding
aspect ratio

The aspect ratio of an image or a movie describes the **proportional relationship** between its width and its height.

Aspect ratio is commonly expressed with two numbers separated by a colon.

width:height

Most movies have aspect ratios of
either **16:9** or **4:3**.

The solution

Step 1:

Wrap the video (iframe, object embed) inside a parent box. In our case, the parent box will be called “video-container”.

.video-container



```
<div class="video-container">  
  <iframe width="560"  
    height="315"  
    src="Youtube-URL"  
    frameborder="0"  
    allowfullscreen></iframe>  
</div>
```

Step 2:

Style the “video-container” box so that it is the correct aspect ratio. (We will look at how to create this aspect ratio very soon).

16



9

Step 3:

Make the video inside that box stretch to fit the dimensions of the box. We can do this by setting the width and height to 100%.

16



9

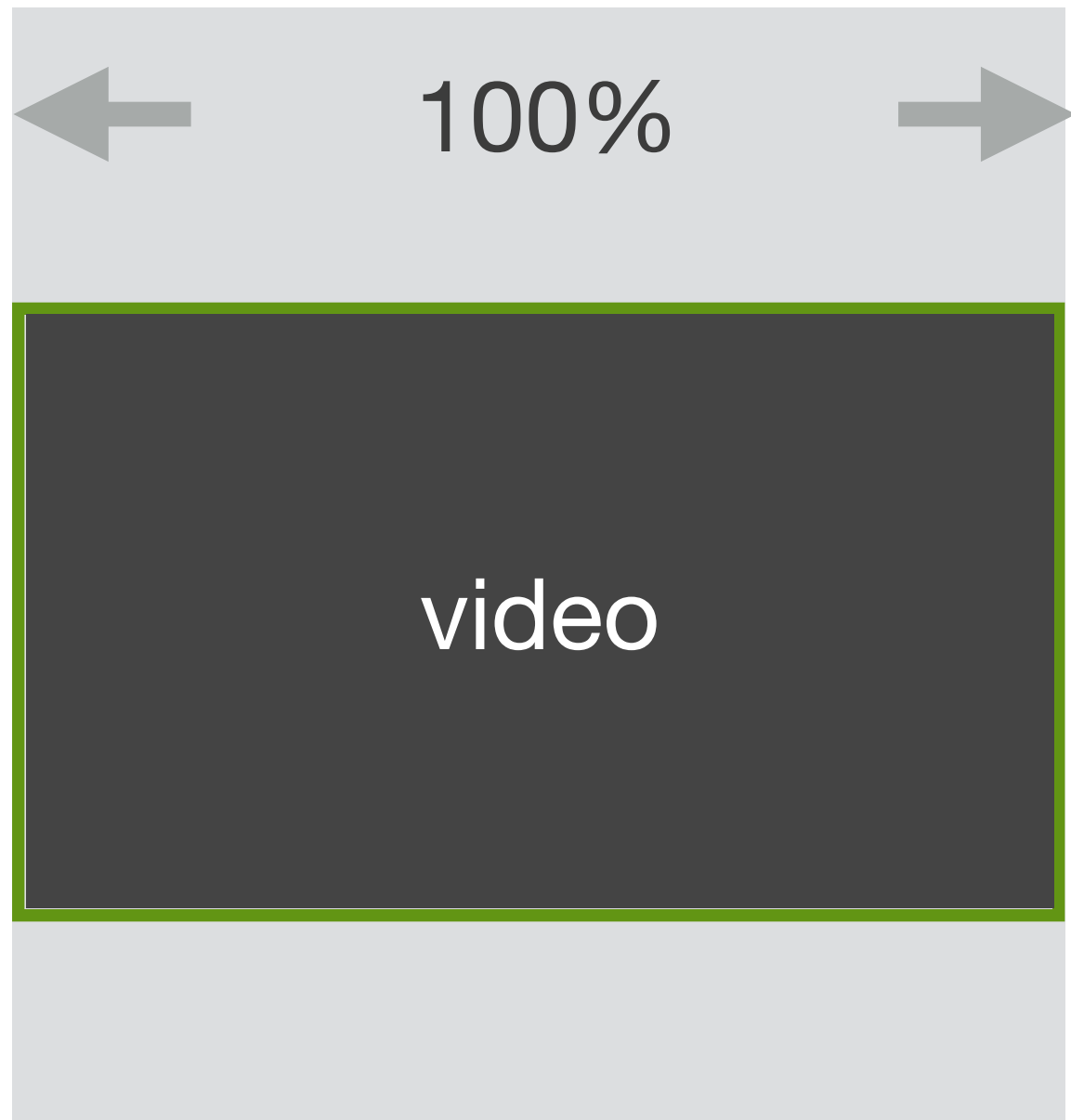
```
.video-container iframe,  
.video-container object,  
.video-container embed  
{  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
}
```

Creating the
width aspect
ratio

So, how would we make our
“video-container” box have **an
aspect ratio of 16:9?**

We are going to allow our “video-container” box to **spread to the full width of the parent container.** This will allow it to be “fluid”.

To do this, we simply **do not define a width** for the “video-container” box. It will be automatically spread to fit the parent (ie. 100% wide).



Creating the
height aspect
ratio

If the width is now 100%, then the height of the “video-container” box should be:

$$9/16 = 0.5625 \text{ (or } 56.25\%)$$

100%

video

56.25%

The problem is that setting a percentage-based height on the “video-container” box **will not work.**

If we use a percentage of 56.25% to define the height of the “video-container”, this percentage **will not be a percentage of the “video-container” box’s width.**

The “video-container” box’s percentage-based height would be defined **in relation to the height of any parent container.**

This is not what we want. So, how
do we **get around the problem?**

Step 1:

We set the height of the “video-container” box to “0”.

100%

0%

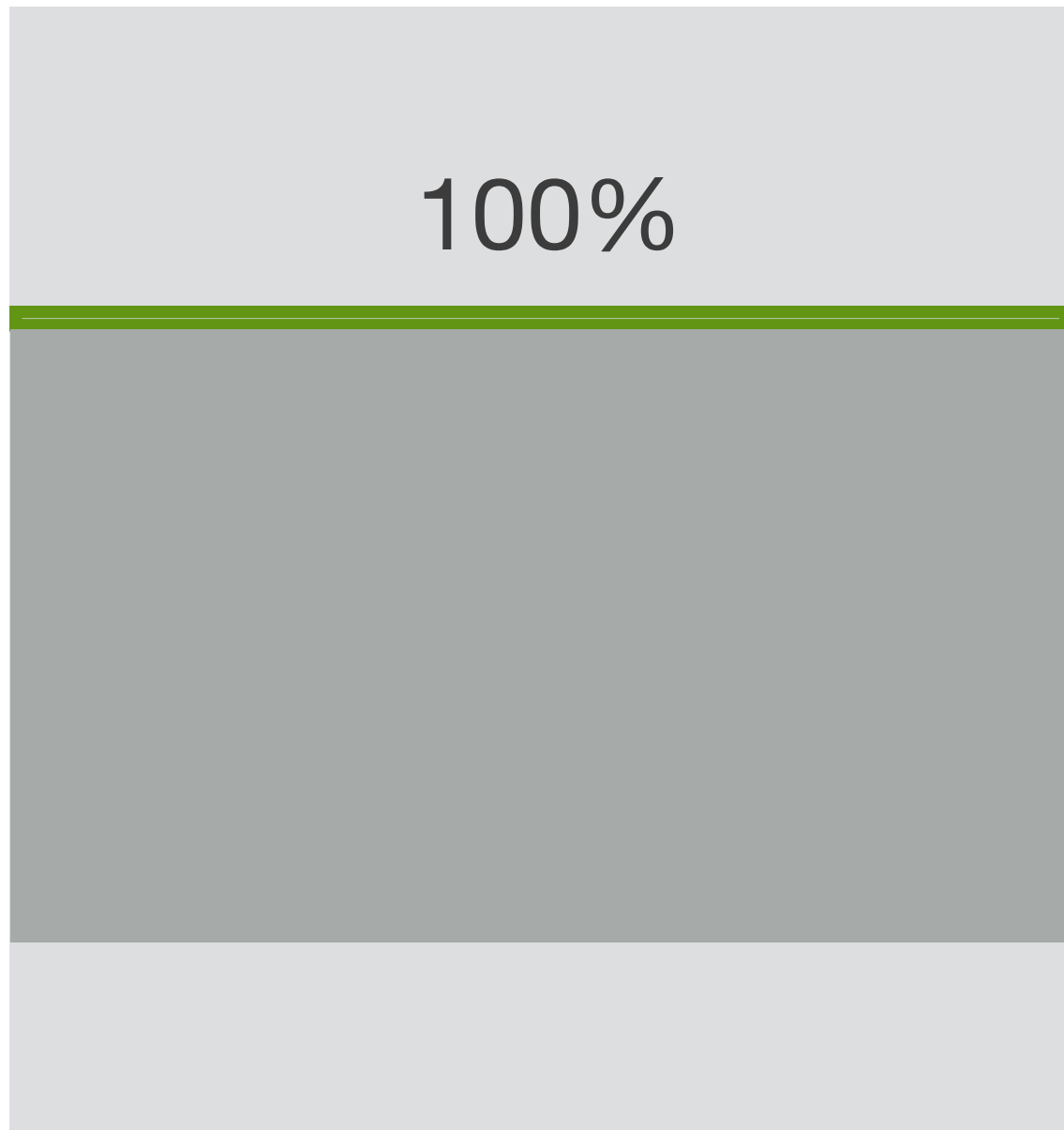
Step 2:

We then add padding-bottom of 56.25% to create the height.

100%

0%

56.25%



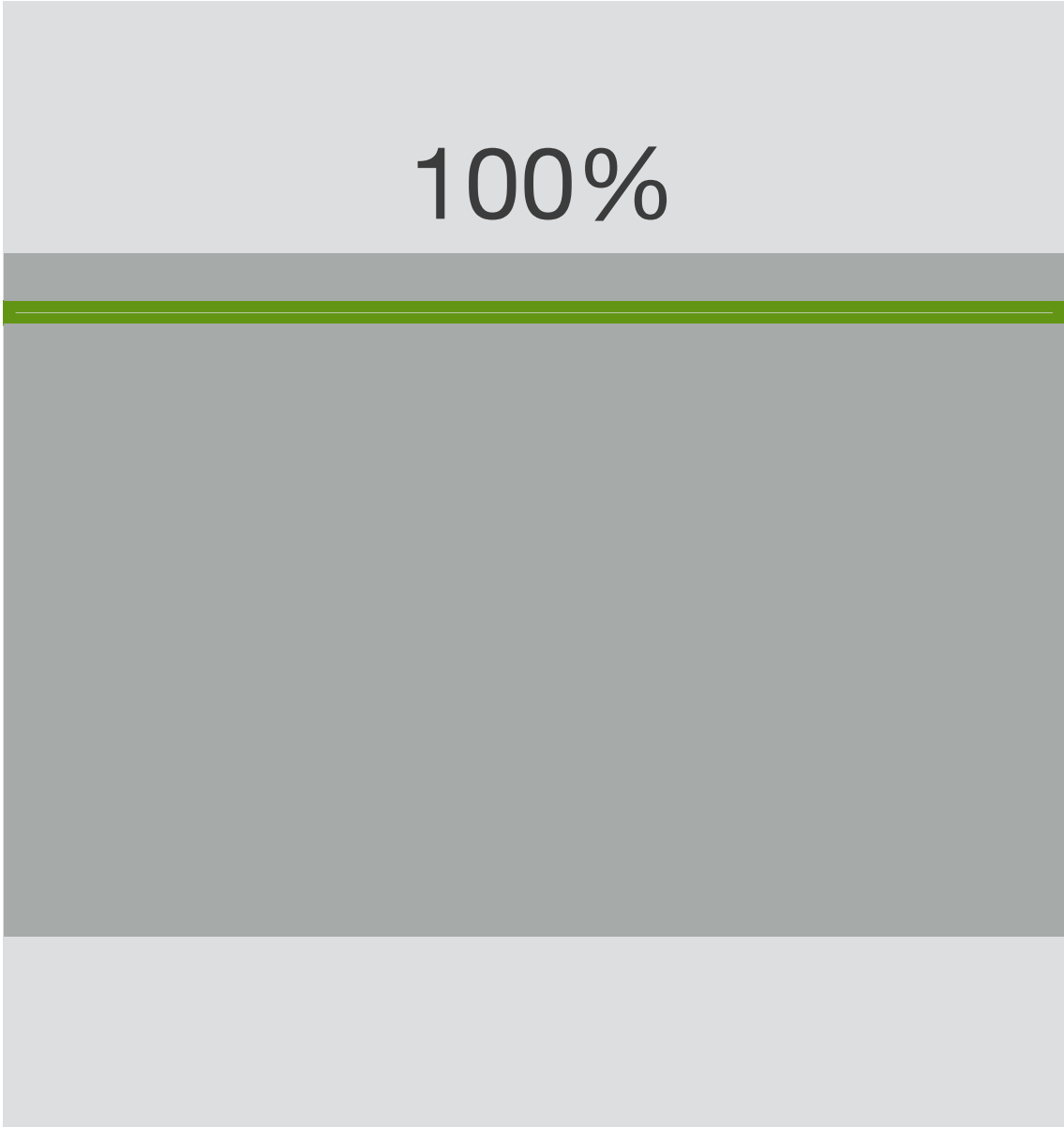
```
.video-container
{
    position: relative;
    height: 0;
    overflow: hidden;
    padding-bottom: 56.25%;
}
```

The height of the box is now:

0 (height) + 56.25% (padding-bottom)

Step 3:

We also need to add 30px of padding-top for youtube/vimeo control panel.



30px

56.25%

```
.video-container
{
    position: relative;
    height: 0;
    overflow: hidden;
    padding-bottom: 56.25%;
    padding-top: 30px;
}
```


The height of the box is now:

0 (height) + 56.25% (padding-bottom) + 30px (padding-top)



Russ Weakley

Max Design

Site: maxdesign.com.au

Twitter: twitter.com/russmaxdesign

Slideshare: slideshare.net/maxdesign

Linkedin: linkedin.com/in/russweakley