



# Figure as a universal container

I've been looking at some way to group content in a semantic way. For example, rather than using:

```
<div class="video">
  <!-- iframe for youtube video -->
</div>
```

I would like to find an equivalent unit of content that would allow me to group the children and still be semantically valid. Looking at the code from Dudley Storey's [The New Code](#) that I saw his use of the `figure` element to do this. Knowing that Dudley would never do something that went against spec I thought I'd look at `figure` and see if would do what I want it to do.

The `figure` element is surprisingly flexible in what it can hold. According to the HTML specification:

The `figure` element represents some flow content, optionally with a caption, that is self-contained (like a complete sentence) and is typically referenced as a single unit from the main flow of the document.

Self-contained in this context does not necessarily mean independent. For example, each sentence in a paragraph is self-contained; an image that is part of a sentence would be inappropriate for `figure`, but an entire sentence made of images would be fitting.

The element can thus be used to annotate illustrations, diagrams, photos, code listings, etc.

From the [HTML5 specification](#)

This means that we can go way beyond just doing the standard `figure` and image setup:

```
<figure>
  <img src='path/to/my/image.png' alt=''>
  <figcaption>Short description</figcaption>
</figure>
```

We can also do something like this:

```
<figure class='video'>
  <iframe width="560" height="315" src="https://www.youtube.com/embed/rn7?
</figure>
```

## Captions, generated text and other uses

figcaption is used to describe the element. If we go back to the image example:

```
<figure>
  <img src='path/to/my/image.png' alt=''>
  <figcaption>Short description</figcaption>
</figure>
```

We can use CSS generated content to tailor the way the figcaption look and add generated text to make sure that each image is identifiable on its own with a figure number before the text:

```
figure {
  counter-increment: figure-count;
  max-width: 100%;

  img {
    max-height: auto;
    max-width: inherit;
  }

  figcaption {
    font-size: .75rem;
```

```

    color: rgb(51, 51, 51);
}

figcaption::before {
    content: 'Figure ' counter.figure-count ': ';
}
'Figure '

```

The two key elements to create self-increasing counters are:

The counter-increment rule in the figure element increases the figure-count counter for every image that appears in the page.

The figcaption::before pseudo element that will insert the text, along with the current value of the figure-count counter.

If we insert new images or delete existing ones the values will adjust automatically without us having to do any manual work, the CSS rules will handle the numbering of the figures. We could do the same thing for videos and any other elements that

We can also use figcaption to describe or otherwise enhance the content of the figure. In the video example we could use figcaption to hold buttons for a custom player, transcript enablement and other enhancements we choose to make to the video. Below is how we could write the markup for the custom buttons:

```

<figure class='video'>
  <iframe id="video" width="560" height="315"
  src="https://www.youtube.com/embed/rn7szaphdWk"
  frameborder="0" allowfullscreen></iframe>
  <figcaption>
    
    
    
  </figcaption>
</figure>

```

```
</figcaption>  
</figure>
```

and then script the buttons to actually work with the video.

There are more things you can do with figures. To make sure your code still validates as HTML look at the spec for the [figure](#) and [figcaption](#) elements to make sure that the content you want to put inside the figure or caption is valid.