

# HTML



# CSS



## HTML & CSS: FUNDAMENTALS OF DEVELOPMENT

Instructor: Beck Johnson

Week 2 Extras



# BACKGROUND IMAGES

# BACKGROUND COLOR REVIEW

```
p {  
  background-color: gray;  
  color: white;  
}
```

This is a paragraph  
with the background  
color set to gray.

# BACKGROUND IMAGES

Can set background of an element as an **image** (instead of a color) with the property **background-image**

The **value** is `url("path")`, where **path** is the **relative** or **absolute** path to where the image lives, like this:

```
p {  
    background-image: url("images/kitten.jpg");  
    color: white;  
}
```



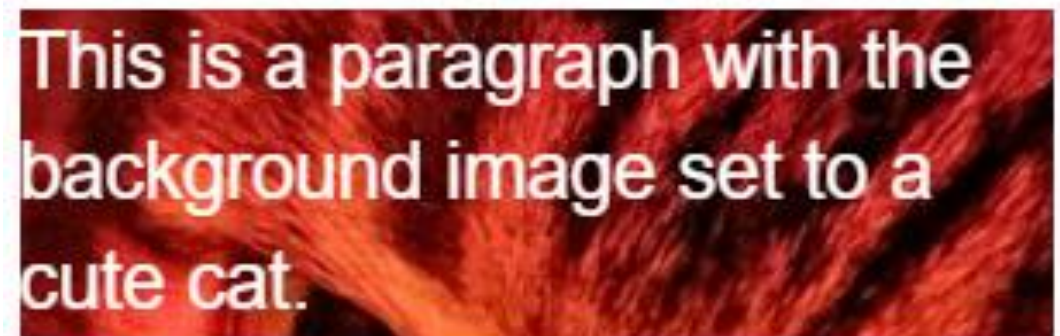
# BACKGROUND IMAGES

```
p {  
    background-image: url("images/kitten.jpg");  
    color: white;  
}
```



The amount of image that displays in the background is calculated based on image size and container size.

- Make sure to resize images so that the part you want visible is within the “view window”
- Or...





# BACKGROUND POSITION EXAMPLES

**background-position:** allows you to move a background image around within its container

- By default, an image is positioned at the top left side of the container

```
section {  
    background-image: url("octopus.jpg");  
    background-position: top left;  
}
```



Image width: 600px by 800px

# BACKGROUND POSITION EXAMPLES

Container width: 600px by 200px



`background-position: top left;`



`background-position: center center;`



`background-position: bottom right;`

# BACKGROUND REPEAT

**background-repeat:** defines if (and how) the background image will repeat

- By default, background images are repeated until they fill the entire container

```
p {  
    background-image: url("codepen.gif");  
    background-repeat: repeat;  
}
```

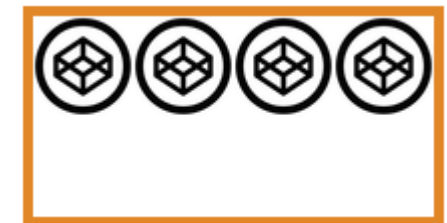


# BACKGROUND REPEAT

**repeat**: tile the image in **both** directions



**repeat-x**: tile the image **horizontally**



**repeat-y**: tile the image **vertically**



**no-repeat**: don't repeat, just show the image **once**



# BACKGROUND ATTACHMENT

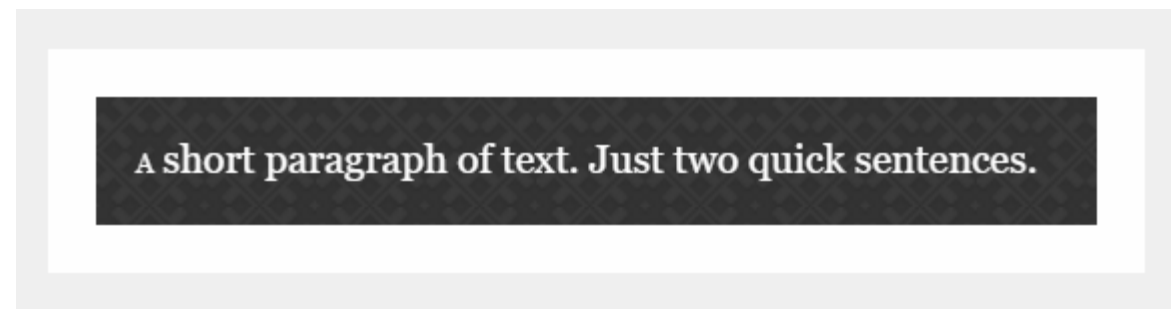
**background-attachment:** images usually scroll with the main view, but setting to **fixed** means the image stays in place when the user scrolls the page

- Difficult to describe, so check out [this demo](#) or [this demo](#)

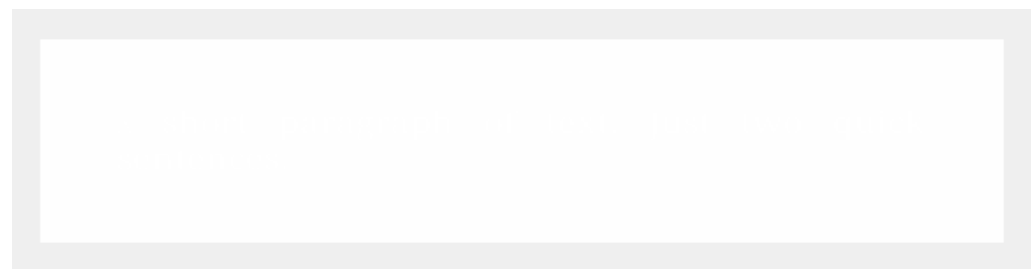
```
section {  
    background-image: url("pattern.png");  
    background-attachment: fixed;  
}
```

# FALLBACK BACKGROUND COLOR

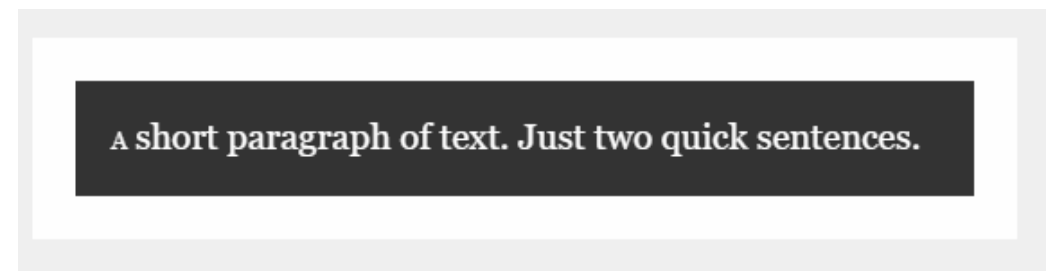
If your background image is dark and your text is light



You may want to specify a **background-color** in **addition** to a **background-image** so that content is visible while the image is loading



So instead of a “blank” area...



...the user can see content while the image downloads

# BACKGROUND GRADIENTS

You can set `background-image` to `linear-gradient`, which is a gradient that the browser draws for you:

```
section { background: linear-gradient(black, white); }
```



As many colors as you want can be blended, separated by commas:

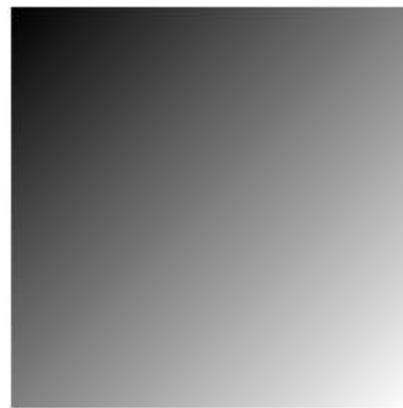
```
section {  
    background: linear-gradient(#ea992e, red, #9e5308);  
}
```



# BACKGROUND GRADIENTS

By default `linear-gradient` draws from top to bottom, but you can set the gradient to draw at an angle instead by starting with `to`

```
section { background: linear-gradient(to bottom right, black, white); }
```



```
section {  
    background: linear-gradient(to right, red, #f06d06, yellow, green);  
}
```

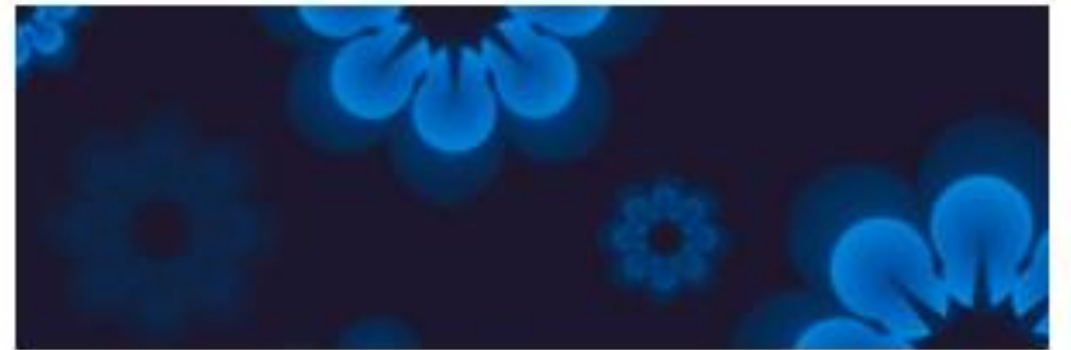




# BACKGROUND GRADIENTS

Background gradients can use rgba colors, meaning you can create a gradient that fades to transparent:

```
body {  
    background-image: url("flowers.png");  
}
```



```
header {  
    background-image: linear-gradient(to  
right, rgba(255,255,255,0),  
rgba(255,255,255,1));  
}
```

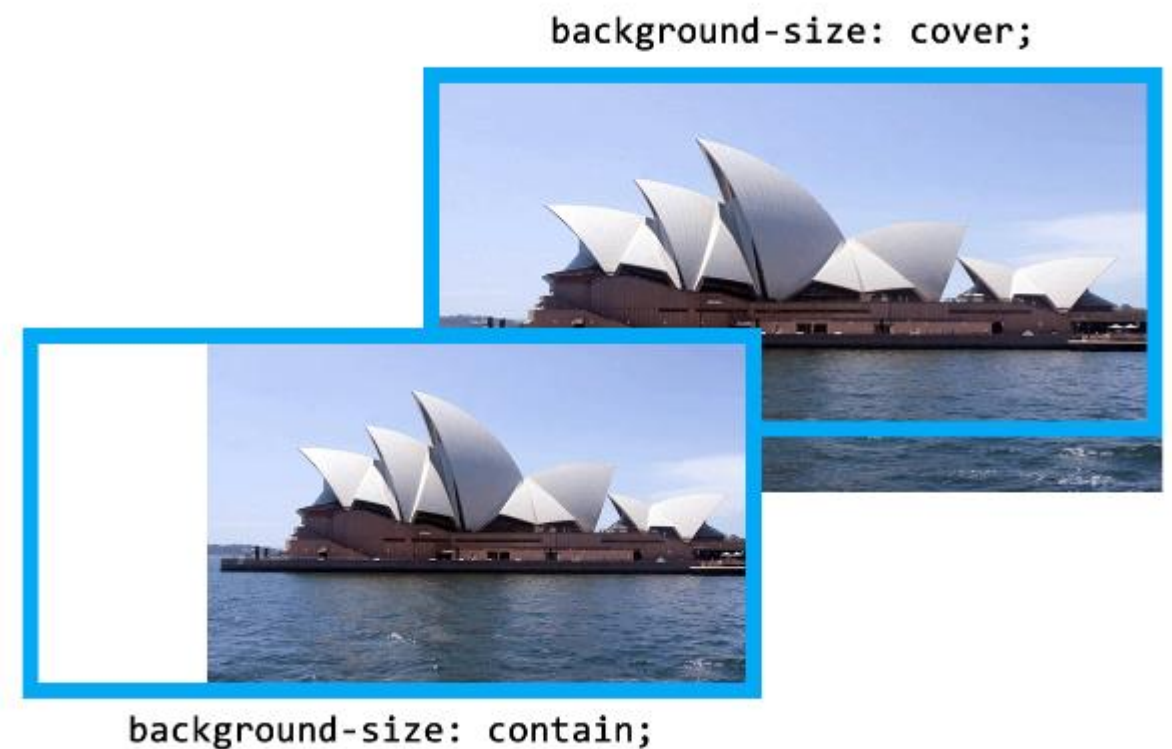


# BACKGROUND SIZE

**background-size:** specifies how much of the container that the image covers

**cover:** always cover the entire container (even if that means cropping an edge, or stretching the image)

**contain:** always show the whole image (even if that means there is space on the sides or bottom)



# HEIGHT AND WIDTH

To ensure that a background image fully displays, you can set the **height** (and/or **width**) attribute on the element using CSS:

```
header {  
    background-image: url("images/hero.png");  
    height: 600px;  
}
```

# HEIGHT AND WIDTH

`height` and `width` can be set on (most) elements to change how much room they take up on the page.

- We'll discuss later why elements like `<a>` and `<em>` don't change when you set their `height` or `width`

The `value` of this property must be a positive number.

- Units are either `px` or `em`
- Or you can specify a percentage

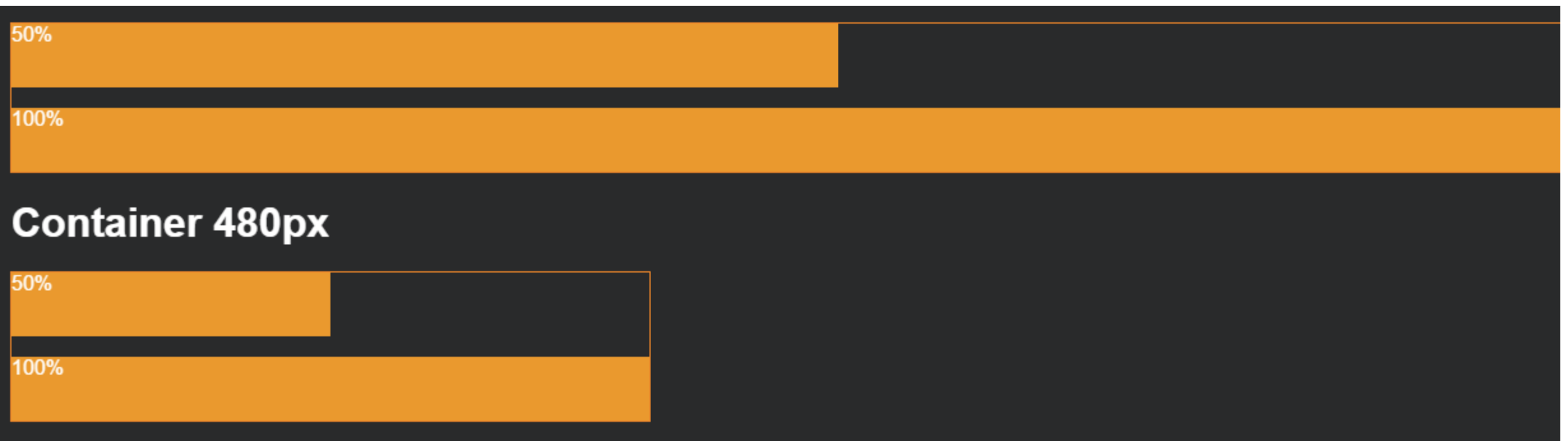
```
header { height: 6em; }
```

# HEIGHT AND WIDTH %

Percentage is based on the element's **parent's** width or height

```
section { width: 50%; }
```

If that **section** were inside a 480 px wide container, it would end up being 240 px wide.

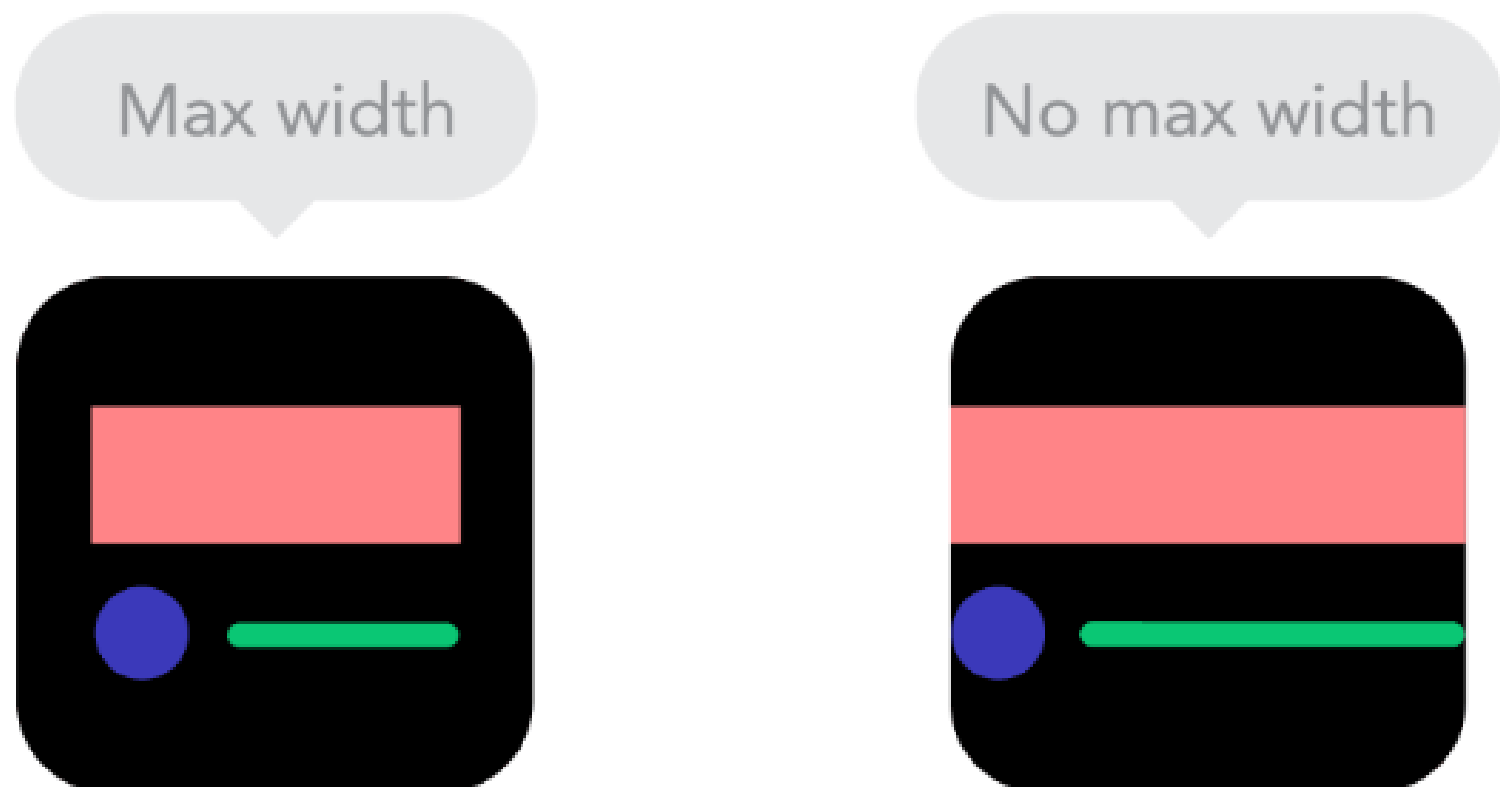




# MAX-HEIGHT AND MAX-WIDTH

To ensure an element is **never larger** than a certain value, use **max-height** or **max-width**

- Typically used to make sure content (particularly text) doesn't spread too far out on large monitors



# MIN-HEIGHT AND MIN-WIDTH

Specify **min-height** or **min-width** if you want to ensure an element is **never smaller** than a certain value.

- This is especially helpful if your size is “dynamic” (based on percentage) and will vary depending on device

```
img {  
    width: 50%;  
    min-width: 350px;  
}
```



# MIN-MAXING

`height` and `width` fix an element to a specific size regardless of display size

- If `width` is wider than the display – scroll bars
- If `width` is smaller than the display – content may wrap even if there is room

`min-height`, `min-width`, `max-height`, and `max-width` allow elements to change when the display size changes, but still allow some control over presentation.

# MIN-MAXING

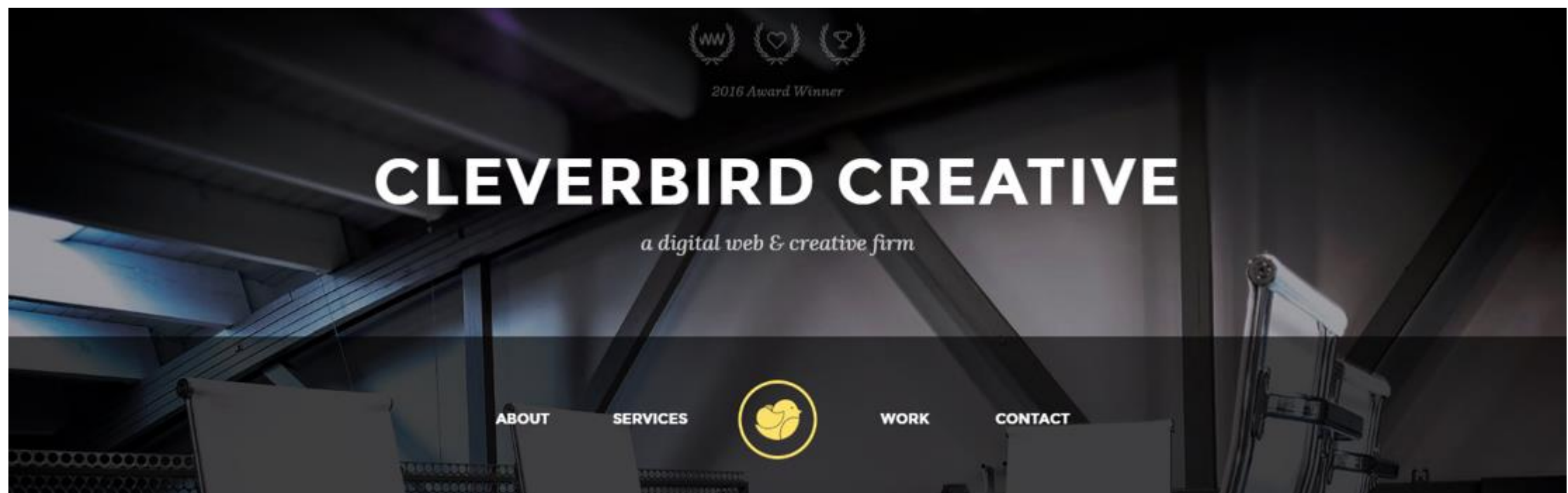
You can choose to set only `width` and/or `height`, only `min-width/min-height`, and only `max-width/max-height` – or any or all of them, depending on your design.

For example, this `section` will expand up to 500 px wide, and then get no bigger. If you shrink your browser, it will shrink until its 100 px wide, and then get no smaller.

```
section {  
    min-width: 100px;  
    max-width: 500px;  
}
```

# NOT ALL HEROES WEAR CAPES

A common use of **background-image** is to create a “hero” image with text overlaying it







**PRACTICE TIME!**

# MAKE A HERO

Add a “hero image” to your site.

- Play around with a bunch of the background properties we learned to make your hero look pretty
- Try setting a `width` and `height`. What happens when you resize your browser window? Change to `min-width` – what changes?