

Lesson 7: Setting Backgrounds & Gradients

- backgrounds: create site's look & feel
 - establish groupings
 - assign priority
- possible backgrounds: solid color
image
gradient] can combine

Adding a Background Color

- background-color property** = to set a solid background color
background property also works
 - accepts: keywords
 - hexadecimals - most common
 - RGB
 - HSL
 - RGBA / HSLA
- set transparent background color: use RGBA / HSLA
provide alternate hexcode
 - format: background-color: #hexcode;
background-color: rgba() / hsla();

Adding a Background Image

- background-image property** = to set a background image
image repeats \updownarrow & \leftrightarrow from top-left by default
 - format: background-image: url("image source");
- background-repeat property** = control appearance of repeated background image
 - values: repeat \updownarrow \leftrightarrow
repeat-x \leftrightarrow
repeat-y \updownarrow
no-repeat
- background-position property** = control placement of background image
offset is relative to top-left corner
 - format: background-image: horizontal offset vertical offset
 - accepts: keyword values
length values (relative & absolute)

- keyword values: top / bottom
left / right
center

→ (eg. left top) = 0 0

→ (eg. right top) = 100% 0

→ (eg. left bottom) = 0 100%

→ (eg. right bottom) = 100% 100%

- shorthand background image properties:

- format: background-color background-image background-position background-repeat

→ (eg. background: #000 url("bg.png") 20px 10px no-repeat;)

Designing Gradient Backgrounds

- gradient backgrounds are treated as background images
- use background/background-image property
- include vendor prefixes for best support
- add solid background color as backup prior to gradient

- **linear-gradient()** = for linear gradient backgrounds
default direction ↓

- format: background: linear-gradient(beginning color, ending color);

- change direction via: keywords
degree values

- keyword values: top
bottom
left
right
to ____

can combine words

- degree values: 0-360deg

- format: background: linear-gradient(direction value, color1, color2);

→ (eg. bottom left) : from bottom left corner to top right corner ↗
≠ "to bottom left"
if background area is a perfect square → = 45deg

- **radial-gradient()** = works from inside to outside element
default direction ↕↔

- format: `background: radial-gradient(starting color, ending color);`

- use css gradient generator: <https://www.colorzilla.com/gradient-editor/>

- **gradient color stops** = to add multiple colors to a gradient
separate colors w/ a comma
by default colors are distanced equally → first @ 0% & last @ 100%.

→ (eg. `background: linear-gradient(to right, #fff, #3ac, #368);`)

- choose color's position: declare length value after color value

→ (eg. `background: linear-gradient(to right, #fff, #3ac 85%, #368);`)

Using Multiple Background Images

- separate multiple background values w/ a comma
- layer order: first declared background image | top / foreground
...
last declared background image | bottom / background

→ (eg. `background: url("fore.png") 0 0 no-repeat,
url("middle.png") 0 0 no-repeat,
url("back.png") 0 0 no-repeat;`)

→ (eg. `background: url("bg.png") 20px 50% no-repeat, linear-gradient(#3ac, #368);`)

Exploring New Background Properties

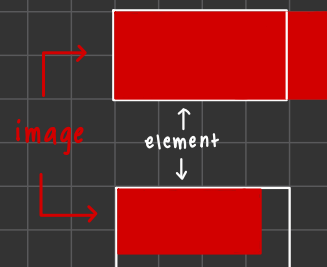
- new background properties: `background-size`
`background-clip`
`background-origin`
- **background-size property** = to change the size of a background image

- accepts: general length values
keyword values

- keyword values: **auto** = preserves aspect ratio

cover = preserves aspect ratio
shrinks/stretches image to cover entire element
image is often cut off to occupy full space

contain = preserves aspect ratio
shrinks/stretches image to remain/reside w/in element
always shows the full image
image often doesn't occupy full space of element



- format: `background-size: width, height`
required

→ (eg. `background-size: auto 75%;`)

• `background-clip property` = to specify the surface area a background image will cover

• `background-origin property` = to control where the background-position should originate

- values: `border-box` = extends background into border of element
background-clip default value

`padding-box` = extends background into padding of element, but contained w/in border
background-origin default

`content-box` = contains background w/in border & padding of element