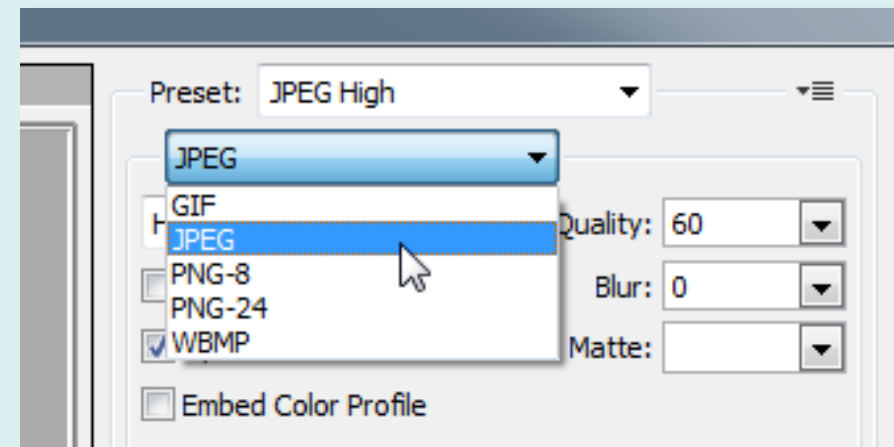


Working with images

Design for web content

4 common raster formats

- JPEG (Joint Photographic Experts Group)
- GIF (Graphics Interchange Format)
- PNG-8 (Portable Network Graphics 8bit)
- PNG-24 (Portable Network Graphics 24bit)



What about WebP?

[WebP](#) is a (relatively) new image file format, promoted by Google. It promises smaller files and good control over compression. However, despite being around since 2010, support for the format is sketchy. Although support in Chrome and Firefox has been good for years, Safari only implemented full support in September 2022. Of course, older browsers like IE are not compatible.

Photoshop does have “save as” support the format but it’s not an option on the “export” modal.

WebP is one to watch for the future, especially as *sustainability* gets traction in web design.

web  p

WebP browser compatibility

WebP image format - OTHER

Usage % of all users
 Global 95.5% + 1.18% = 96.68%

Image format (based on the VP8 video format) that supports lossy and lossless compression, as well as animation and alpha transparency. WebP generally has better compression than JPEG, PNG and GIF and is designed to supersede them. [AVIF](#) and [JPEG XL](#) are designed to supersede WebP.

Current aligned Usage relative Date relative Filtered All

Chrome	Edge*	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS*	Samsung Internet	Opera Mini*	Opera Mobile*	UC Browser for Android	Android Browser*	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-8				10.1												
¹ 9-22		3.1-13.1		¹ 11.5								2.1-3				
² 23-31	12-17	³ 14-15.6	2-64	² 12.1-18			3.2-13.7					¹ 4-4.1				
32-117	18-117	16.0-16.6	65-117	19-102	6-10		14-16.6	4-21		12-12.1		4.2-4.4.4				2.5
118	118	17.0	118	103	11	118	17.0	22	all	73	15.5	118	118	13.1	13.18	3.1
119-121		17.1-TP	119-121				17.1									

Notes Test on a real browser Known issues (1) Resources (7) Feedback

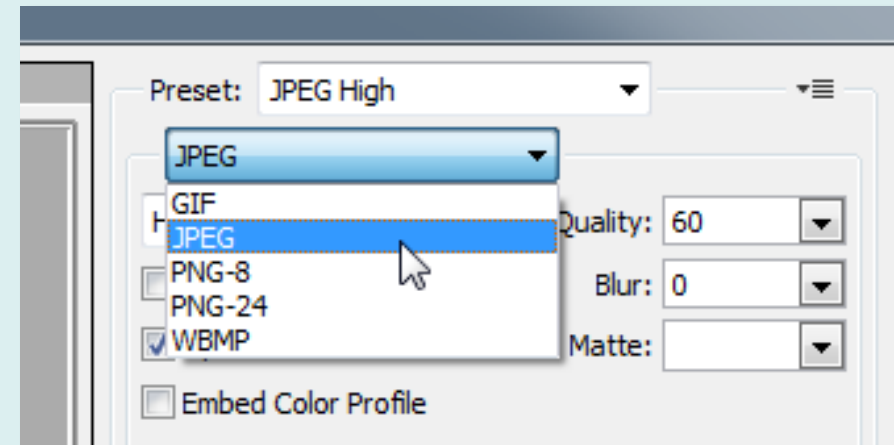
¹ Partial support refers to not supporting lossless, alpha and animated WebP images.
² Partial support refers to not supporting animated WebP images.
³ Safari 14.0 – 15.6 has full support of WebP, but requires macOS 11 Big Sur or later.

Colour Depth

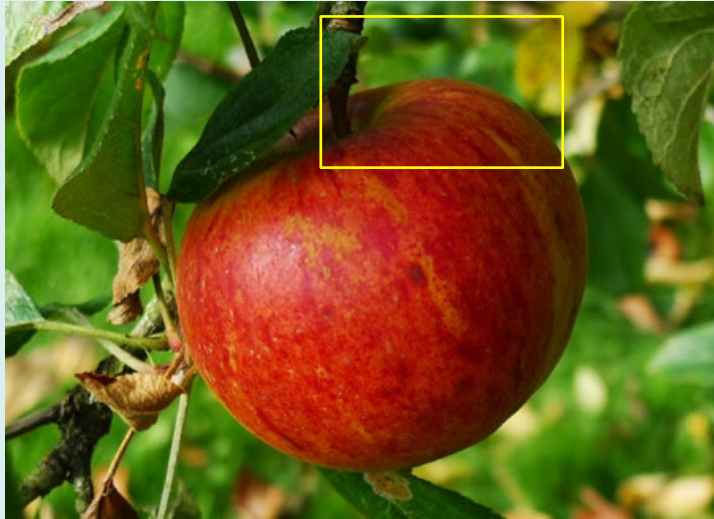
- JPEG True Colour (24 bit)
- GIF 256 colours (8 bit)
- PNG-8 256 colours (8 bit)
- PNG-24 True Colour (24 bit)

“True” colour image formats like JPEG can display around 16 million colours (16,777,216 to be precise, that’s 256 red x 256 green x 256 blue). Since the human eye is capable of discerning fewer colours than this, true colour images are equivalent to what we see in the real world.

256 colour images are sometimes referred to as using an “indexed” colour palette.



256 Colour Dithering



File formats with 256 colour palettes (GIF & PNG-8) are not good for photographic images.

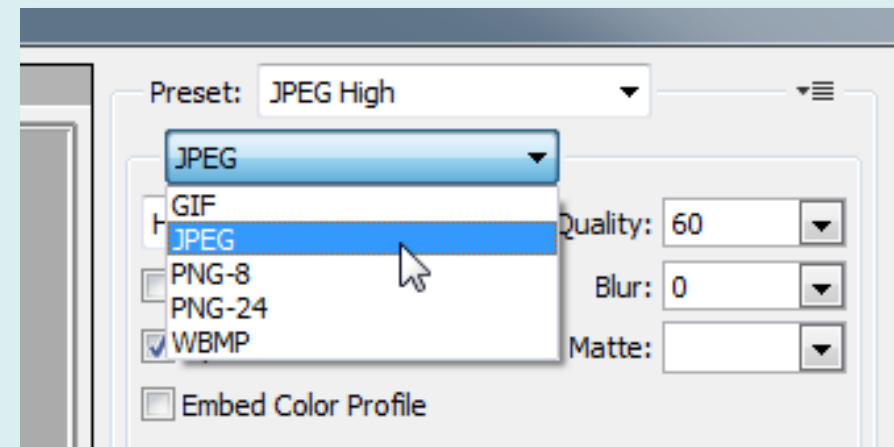
They use a process called “dithering” to simulate additional colours.

<http://en.wikipedia.org/wiki/Dither>

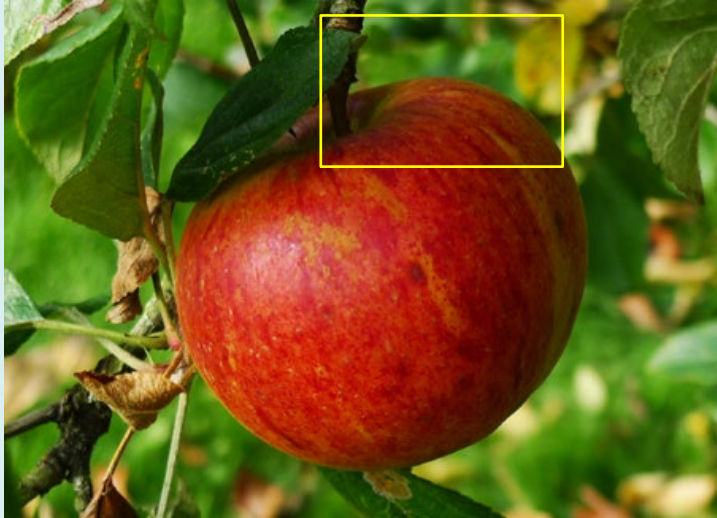
Killer Feature

- JPEG Variable compression
- GIF Frame animation
- PNG-8 Smaller than equivalent GIF
- PNG-24 Alpha transparency

You might wonder why we need 4 different file formats for web design work. Well, they all have different attributes and each one has a “killer” feature that the others don’t and this makes them useful in different situations.



JPEG Compression



The JPEG file format is particularly good for photographs because it has a “true colour” palette and variable compression algorithm that can be used to control file size.

Over-compression can result in “artefacts” appearing in the image.



Image Quality vs JPEG File Size



400px

600px

File size

JPEG 100% = 103KB

JPEG 80% = 59KB

JPEG 60% = 37KB

JPEG 40% = 26KB

PNG-24 = 463KB

Original image size : 2912px x 3264px reduced to 400px x 600px for use on web

Original file size: 6.12MB (612,000KB)

Quality vs Compression



100% = 463kb at 800x600px

low compression = high quality



10% = 18kb at 800x600px

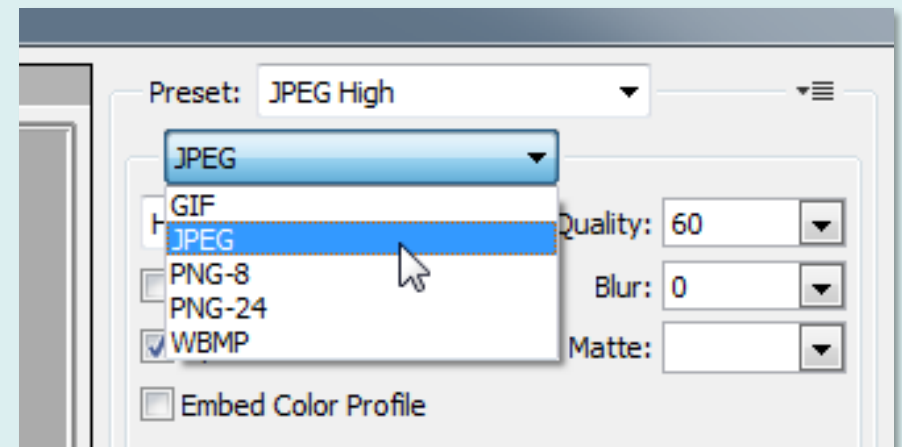
high compression = low quality

Image optimisation is always a balance between image quality and file size.

Transparency

- JPEG None
- GIF 1 colour
- PNG-8 1 colour
- PNG-24 Alpha channel (256 steps)

The key problem with JPEG images is the inability to contain transparency information. Fortunately, the more recent PNG-24 format can display 256 grades of transparency from completely transparent to completely opaque. This information is known as an “alpha channel”. In GIF and PNG-8 files, one of the 256 colours can be designated as transparent.



Best used for...

- JPEG Photographs
- GIF Simple graphics or animation
- PNG-8 Simple graphics (often smaller than GIF)
- PNG-24 Complex graphics (photo & line art)

Each format has a specific use, so make sure you're using the correct format for the type of image you want to present. Using the wrong format will mean using files that are too large or images of poor quality.

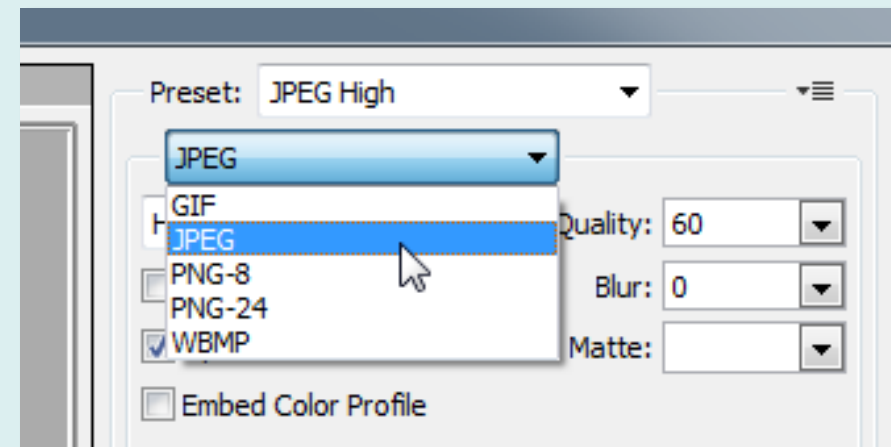


Image type

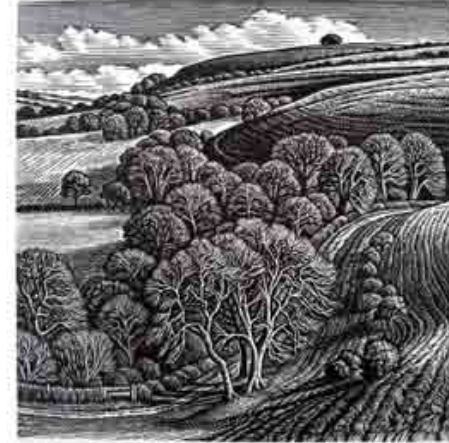
Always use the correct image file format for the type of image you want to display because the fine detail of some graphic images will be lost or become “fuzzy” if the wrong file format is used. Essentially, use JPEG files for photographs and PNG files for graphic images.



512x512px

PNG file used for graphic images

Always use the correct image file format for the type of image you want to display because the fine detail of some graphic images will be lost or become “fuzzy” if the wrong file format is used. Essentially, use JPEG files for photographs and PNG files for graphic images.



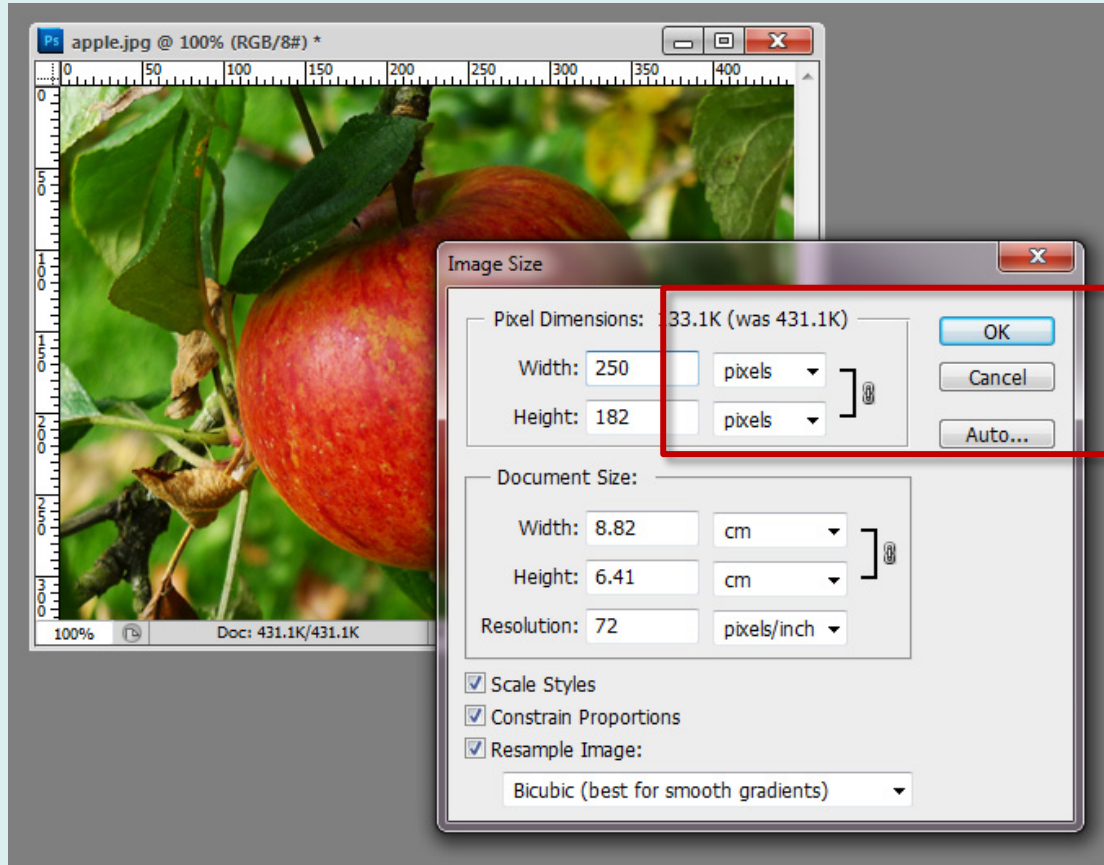
512x512px

JPEG file used inappropriately

The effect of JPEG compression is especially noticeable in the text of the right-hand image.

Note: never use images for text – this is for demonstration only.

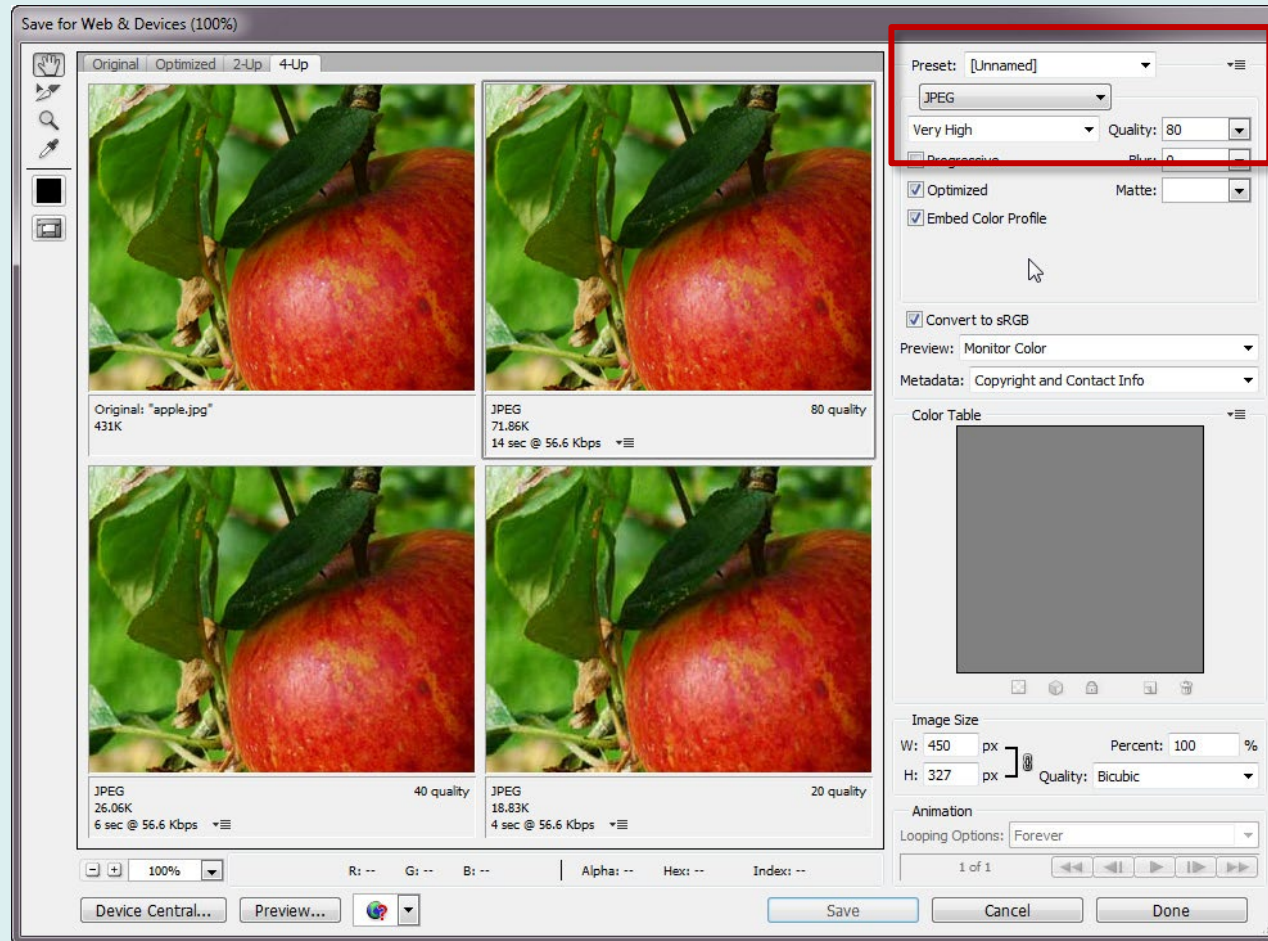
Set your image size...



When working with raster images for web, it's a good idea to set your Photoshop rulers to "pixels" ([Edit > Preferences > Units & Rulers...](#)) because that's the most convenient unit type for our work.

You can then resize your image ([Image > Image Size...](#)) by entering either the required width or height value in the dialogue box. By default, the *aspect ratio* is locked (small chain-link symbol), so changing either the width or the height will result in a proportionally scaled image.

Save for Web...



Photoshop has a special tool for helping us to save our images ready for the web. It offers a choice of all 4 formats of interest to us and it can give us previews of how the image will look once it is saved. It can also tell us how big each file will be and how long it will take to download.

Curiously, Photoshop CC now considers this to be a “legacy” feature, but it remains the best tool for the job. The new [Export As...](#) tool has similar features.

[File > Export > Save for Web \(Legacy\)...](#)

The web standards model

- 1** Content
Text, images, ...
- 2** Structure
HTML
- 3** Presentation
CSS
- 4** Behaviour
JavaScript



Scalable Vector Graphics

Unlike all the other image formats we have talked about so far, SVG images use vectors rather than pixels. In other words, SVGs are created using coordinates rather than a matrix of coloured squares.

SVGs use the XML markup language and can therefore be embedded within HTML if required, unlike raster images that must be linked.

The most obvious advantage of SVG images is that they are infinitely scalable and the quality does not degrade as they get larger. Ideal for responsive web design.

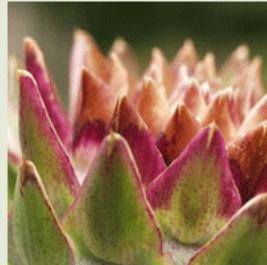
```
<svg viewBox="0 0 2970 1700" xmlns="http://www.w3.org/2000/svg" xml:space="preserve" style="fill-rule:evenodd; clip-rule:evenodd;stroke-linecap:round;stroke-miterlimit:1.5">
  <g transform="scale(1.85625 1.88889)">
    <path style="fill:none" d="M0 0h1600v900H0z"/>
  </g>
  <!-- CONTENT rectangle -->
  <g transform="matrix(0.660362,-0.132369,9.94077e-18,0.613863,603.714,59.398)">
    <rect x="153" y="122" width="461" height="611" style="fill:rgb(66,72,102);stroke:white;stroke-width:9px;"/>
  </g>
  <!-- STRUCTURE rectangle -->
  <g transform="matrix(0.660362,-0.132369,3.5345e-17,0.613863,775.93,184.941)">
    <g>
      <rect x="153" y="122" width="461" height="611" style="fill:rgb(110,77,234);stroke:white;stroke-width:9px;"/>
    </g>
  </g>
  <!-- PRESENTATION rectangle -->
  <g transform="matrix(0.660362,-0.132369,3.5345e-17,0.613863,948.145,310.484)">
    <rect x="153" y="122" width="461" height="611" style="fill:rgb(9,192,16);stroke:white;stroke-width:9px;"/>
  </g>
  <!-- BEHAVIOUR rectangle -->
  <g transform="matrix(0.660362,-0.132369,3.5345e-17,0.613863,1120.36,436.027)">
    <rect x="153" y="122" width="461" height="611" style="fill:rgb(252,166,17);stroke:white;stroke-width:9px;"/>
  </g>
</svg>
```


Image positioning with CSS

CSS Floats

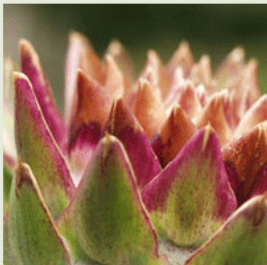
Mauris rhoncus, justo vitae imperdiet sodales, tellus diam ultrices est, non condimentum felis est quis elit. Maecenas dolor. Nulla facilisi. Quisque velit. Nulla facilisi. Sed sed velit. Nam interdum.

Suspendisse congue risus et lectus. Proin pellentesque porttitor nisl. Duis aliquam. In eu felis. Aliquam erat volutpat. [Aliquam scelerisque](#) volutpat leo. Aenean tempor sapien quis lorem. Nulla lorem metus, tincidunt at, porttitor ut, vestibulum ut, nunc. Etiam ac erat non tortor pellentesque bibendum. Ut sodales volutpat ligula. Maecenas vestibulum suscipit sem. Nunc tristique augue laoreet ipsum. Phasellus quis dolor. Fusce pellentesque. Nam mollis ullamcorper nunc. Nam viverra sodales nunc. Praesent vulputate neque vel orci consequat tincidunt.



Sed elit. Nulla sapien risus, pulvinar vel, porttitor at, lobortis non, leo. Vivamus a odio. Sed vulputate quam a nunc. Integer aliquam. Ut ac enim nec ligula venenatis pretium. Cras venenatis nisl aliquet neque. Maecenas nisi. Proin faucibus

odio nibh eu enim. Nullam posuere leo a lacus. Pellentesque odio. Maecenas tempus dui at purus tristique mattis. Mauris nisi ligula, commodo non, consetetuer nec, semper in, sem.



Quisque a nisi a mauris dignissim ultricies. Ut ipsum purus, hendrerit ut, varius eu, fringilla sodales, magna. Cras quis ipsum in lectus volutpat egestas. Sed eget nibh. Morbi commodo dapibus nunc. Pellentesque convallis tempor erat. Ut commodo. Aliquam condimentum. Praesent lacus tortor, dapibus sit amet, congue eget, blandit sit amet, arcu. Fusce euismod ultricies urna. Sed bibendum lectus vel nulla. Vivamus faucibus.

Donec mollis, ante at pretium scelerisque, pede magna imperdiet leo, sed condimentum velit augue ac metus.

Aliquam iaculis nonummy quam. Morbi elementum

quam in lectus. Vestibulum congue, lectus sed convallis malesuada, urna lectus rhoncus tellus, nec rutrum lectus magna a turpis. Suspendisse elementum sem a dui. Curabitur dictum mauris sit amet urna. Cras ornare. Donec sapien purus, vehicula eget, volutpat quis, ullamcorper

CSS floats are one of the most powerful tools we have to control the relative position of HTML elements within a page. You're going to be using them a lot for controlling page layouts but for now, let's consider their use for placing images within columns of text.

By default, images will just run inline with text (they are inline elements) but if we apply a CSS float property rule to them, we can control where they appear in relation to the surrounding text.

Floating an image to the right causes text to flow to the left and vice versa.

Left Placement

odio nibh eu enim. Nullam posuere leo a lacus. Pellentesque odio. Maecenas tempus dui at purus tristique mattis. Mauris nisi ligula, commodo non, consectetur nec, semper in, sem.



Quisque a nisi a mauris dignissim ultricies. Ut ipsum purus, hendrerit ut, varius eu, fringilla sodales, magna. Cras quis ipsum in lectus volutpat egestas. Sed eget nibh. Morbi commodo dapibus nunc. Pellentesque convallis tempor erat. Ut commodo. Aliquam condimentum. Praesent lacus tortor, dapibus sit amet, congue eget, blandit sit amet, arcu. Fusce euismod ultricies urna. Sed bibendum lectus vel nulla. Vivamus faucibus.

Donec mollis, ante at pretium scelerisque, pede magna imperdiet leo, sed condimentum velit augue ac metus. Aliquam iaculis nonummy quam. Morbi elementum

quam in lectus. Vestibulum congue, lectus sed convallis malesuada, urna lectus rhoncus tellus, nec rutrum lectus magna a turpis. Suspendisse elementum sem a dui. Curabitur dictum mauris sit amet urna. Cras ornare. Donec sapien purus, vehicula eget, volutpat quis, ullamcorper

```

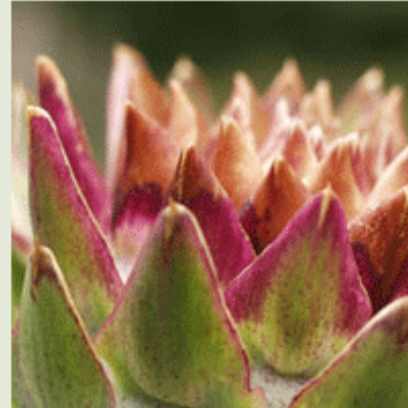
```

```
img {  
  float: left;  
  margin: 7px 20px 10px 0;  
}
```

Right Placement

Mauris rhoncus, justo vitae imperdiet sodales, tellus diam ultrices est, non condimentum felis est quis elit. Maecenas dolor. Nulla facilisi. Quisque velit. Nulla facilisi. Sed sed velit. Nam interdum.

Suspendisse congue risus et lectus. Proin pellentesque porttitor nisl. Duis aliquam. In eu felis. Aliquam erat volutpat. Aliquam scelerisque volutpat leo. Aenean tempor sapien quis lorem. Nulla lorem metus, tincidunt at, porttitor ut, vestibulum ut, nunc. Etiam ac erat non tortor pellentesque bibendum. Ut sodales volutpat ligula. Maecenas vestibulum suscipit sem. Nunc tristique augue laoreet ipsum. Phasellus quis dolor. Fusce pellentesque. Nam mollis ullamcorper nunc. Nam viverra sodales nunc. Praesent vulputate neque vel orci consequat tincidunt.



Sed elit. Nulla sapien risus, pulvinar vel, porttitor at, lobortis non, leo. Vivamus a odio. Sed vulputate quam a nunc. Integer aliquam. Ut ac enim nec ligula venenatis pretium. Cras venenatis nisl aliquet neque. Maecenas nisi. Proin faucibus

```

```

```
img {  
  float: right;  
  margin: 7px 0 10px 20px;  
}
```

We're floating the image to the right, so we are adding 20px margin to the left side of the image to give a reasonable separation from the text.

Centre Placement

libero odio ultrices felis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.



Ut tellus nisl, egestas eget, facilisis ac, consequat ac, leo. Etiam iaculis, quam hendrerit aliquet consequat, mauris lacus feugiat eros, sit amet sagittis felis nibh ac lectus. Proin et eros. Nulla erat nulla, aliquam at, rutrum et, fermentum ut, auque. Donec tempus magna id felis. Etiam

```

```

```
img {  
  display: block;  
  margin: 10px auto;  
}
```

Centring an image is a bit different, we can't float centre. We need two declarations, one to change the image behaviour from *inline* to *block* and another to make the left and right margins equal using the *auto* value.

Circular images

libero odio ultrices felis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.



Ut tellus nisl, egestas eget, facilisis ac, consequat ac, leo. Etiam iaculis, quam hendrerit aliquet consequat, mauris lacus feugiat eros, sit amet sagittis felis nibh ac lectus. Proin et eros. Nulla erat nulla, aliquam at, rutrum et, fermentum ut, auque. Donec tempus magna id felis. Etiam

```

```

```
img {  
  display: block;  
  margin: 10px auto;  
  border-radius: 50%;  
}
```

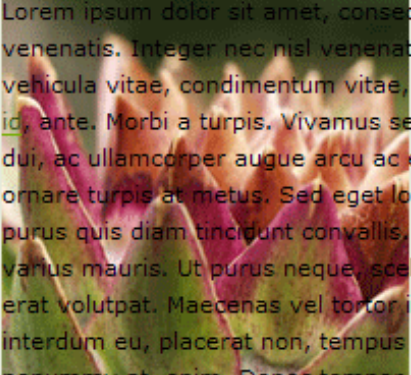
If we want to create a circular image, we **must** begin with a square image – the intrinsic height and width of our image must be the same (in this case, it's 200px). Then, set the border-radius to 50% (in other words, 100px).

Note: Do not use the height and width attributes to force an image to be square – use an image editor to crop it.

Background Images

Background Images

id, velit. Curabitur euismod, massa sed lobortis facilisis, libero odio ultrices tellis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla in ligula in risus fermentum venenatis. Integer nec nisl venenatis nisl ultrices auctor. Proin arcu dui, bibendum sit amet, vehicula vitae, condimentum vitae, lorem. Mauris enim est, vehicula eu, molestie ut, vulputate id, ante. Morbi a turpis. Vivamus semper, pede et lacinia accumsan, dolor lacus fermentum dui, ac ullamcorper augue arcu ac est. Quisque eget odio. Curabitur molestie leo a urna. Morbi ornare turpis at metus. Sed eget lorem at augue blandit rutrum. Nulla quis lectus. Maecenas a purus quis diam tincidunt convallis. Proin in nisi in sem rutrum dapibus. Sed pellentesque varius mauris. Ut purus neque, scelerisque vitae, varius vitae, fermentum ac, nulla. Aliquam erat volutpat. Maecenas vel tortor in dui tristique sagittis. Sed sed quam. Donec elit eros, interdum eu, placerat non, tempus et, nisl. In justo massa, aliquet ut, ullamcorper id, nonummy at, enim. Donec tempor. Sed congue. Donec in mauris. Vestibulum vestibulum euismod elit. Praesent ac risus. Nam tristique. Sed feugiat risus vel felis. Sed libero. Praesent nulla leo, mattis et, faucibus fermentum, venenatis sit amet, libero. Curabitur porta. Nam congue, turpis nec condimentum facilisis, risus quam eleifend dolor, ac fermentum massa dui in ipsum.

The CSS *background* property can be used to display an image in the background of an element.

Obviously, this should be used with caution because background images often make a design look too fussy or make the overlying text difficult to read.

```
<p class="background">Lorem ipsum dolor... </p>
```

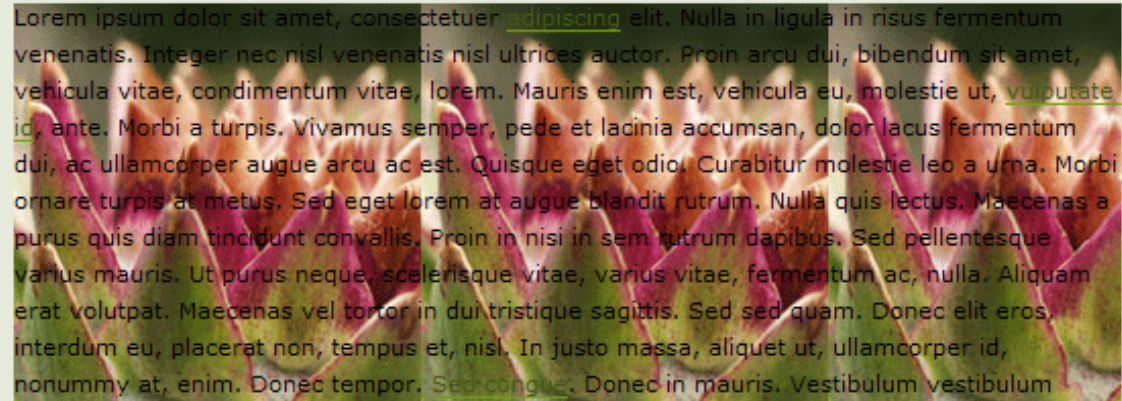
```
p.background {  
    background: url(artichoke.jpg) no-repeat;  
}
```

In this example, an image is shown in the background of a paragraph element.

<https://developer.mozilla.org/en-US/docs/Web/CSS/background>

Background Repeat

id, velit. Curabitur euismod, massa sed lobortis facilisis, libero odio ultrices tellis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla in ligula in risus fermentum venenatis. Integer nec nisl venenatis nisl ultrices auctor. Proin arcu dui, bibendum sit amet, vehicula vitae, condimentum vitae, lorem. Mauris enim est, vehicula eu, molestie ut, vulputate id, ante. Morbi a turpis. Vivamus semper, pede et lacinia accumsan, dolor lacus fermentum dui, ac ullamcorper augue arcu ac est. Quisque eget odio. Curabitur molestie leo a urna. Morbi ornare turpis at metus. Sed eget lorem at augue blandit rutrum. Nulla quis lectus. Maecenas a purus quis diam tincidunt convallis. Proin in nisi in sem rutrum dapibus. Sed pellentesque varius mauris. Ut purus neque, scelerisque vitae, varius vitae, fermentum ac, nulla. Aliquam erat volutpat. Maecenas vel tortor in dui tristique sagittis. Sed sed quam. Donec elit eros, interdum eu, placerat non, tempus et, nisl. In justo massa, aliquet ut, ullamcorper id, nonummy at, enim. Donec tempor. Sed congue. Donec in mauris. Vestibulum vestibulum euismod elit. Praesent ac risus. Nam tristique. Sed feugiat risus vel felis. Sed libero. Praesent nulla leo, mattis et, faucibus fermentum, venenatis sit amet, libero. Curabitur porta. Nam congue, turpis nec condimentum facilisis, risus quam eleifend dolor, ac fermentum massa dui in ipsum.

By default, all background images will repeat both vertically and horizontally (the X and Y directions) but we can control this behaviour using the *repeat* value.

In this example, the image repeats only in the horizontal direction (X).

```
<p class="background">Lorem ipsum dolor... </p>
```

```
p.background {  
    background: url(artichoke.jpg) repeat-x;  
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/background-repeat>

Background Position

id, velit. Curabitur euismod, massa sed lobortis facilisis, libero odio ultrices tellis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla in ligula in risus fermentum venenatis. Integer nec nisl venenatis nisl ultrices auctor. Proin arcu dui, bibendum sit amet, vehicula vitae, condimentum vitae, lorem. Mauris enim est, vehicula eu, molestie ut, vulputate id, ante. Morbi a turpis. Vivamus semper, pede et lacinia accumsan, dolor lacus fermentum dui, ac ullamcorper augue arcu ac est. Quisque eget odio. Curabitur molestie leo a urna. Morbi ornare turpis at metus. Sed eget lorem at augue blandit rutrum. Nulla quis lectus. Maecenas a purus quis diam tincidunt convallis. Proin in nisi in sem rutrum dapibus. Sed pellentesque varius mauris. Ut purus neque, scelerisque vitae, varius vitae, fermentum ac, nulla. Aliquam erat volutpat. Maecenas vel tortor in dui tristique sagittis. Sed sed quam. Donec elit eros, interdum eu, placerat non, tempus et, nisl. In justo massa, aliquet ut, ullamcorper id, nonummy at, enim. Donec tempor. Sed congue. Donec in mauris. Vestibulum vestibulum euismod elit. Praesent ac risus. Nam tristique. Sed feugiat risus vel felis. Sed libero. Praesent nulla leo, mattis et, faucibus fermentum, venenatis sit amet, libero. Curabitur porta. Nam congue, turpis nec condimentum facilisis, risus quam eleifend dolor, ac fermentum massa dui in ipsum.

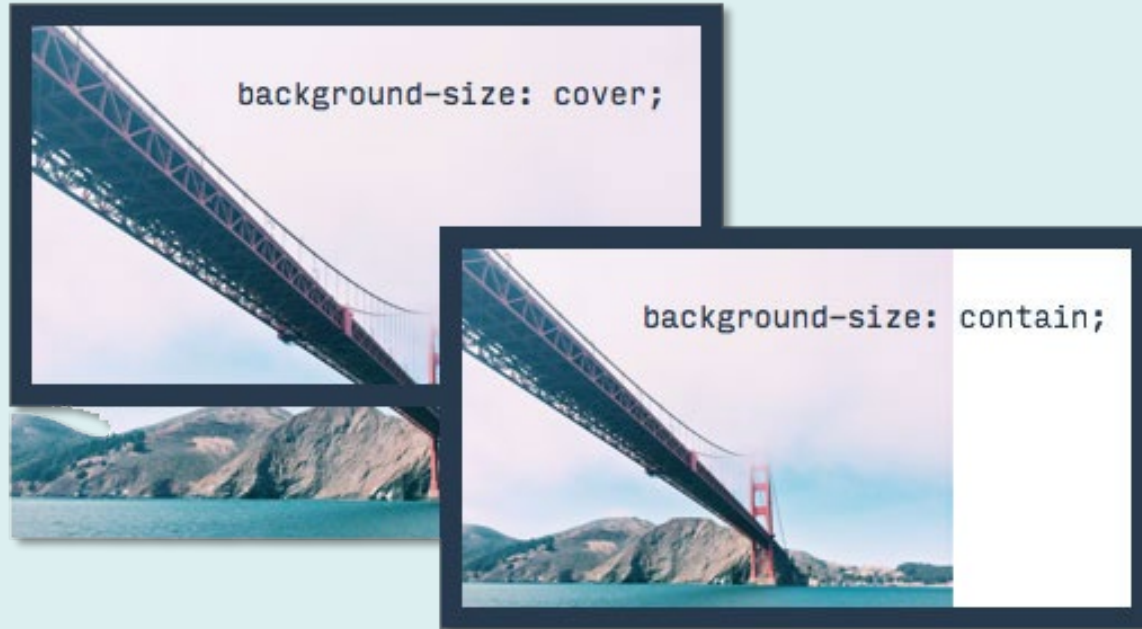
A single background image placed in the middle of the parent element. The position values (50% 50%) places the image half-way down and half-way across the element, effectively centring the image. The same effect can be created using the *center* keyword. Keywords *top*, *bottom*, *left* and *right* may also be used.

```
<p class="background">Lorem ipsum dolor... </p>
```

```
p.background {  
    background: url(artichoke.jpg) no-repeat 50% 50%;  
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/background-position>

Background Size



The *background* property is a shorthand for the many individual background properties in CSS...

background-image
background-repeat
background-position
background-size

```
<p class="background">Lorem ipsum dolor... </p>
```

```
p.background {  
    background-image: url(golden-gate.jpg);  
    background-repeat: no-repeat;  
    background-size: cover;  
}
```

<https://css-tricks.com/almanac/properties/b/background-size/>

Each can be set independently. The *background-size* property can be used with keywords to control how the image is displayed. This can be useful for responsive designs.

Background Image & Colour

id, velit. Curabitur euismod, massa sed lobortis facilisis, libero odio ultrices tellis, non sagittis lacus orci porta diam. Curabitur eu ipsum. Aenean nisl quam, hendrerit non, auctor a, semper eget, sapien. Proin sit amet dolor at leo ultrices vestibulum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla in ligula in risus fermentum venenatis. Integer nec nisl venenatis nisl ultrices auctor. Proin arcu dui, bibendum sit amet, vehicula vitae, condimentum vitae, lorem. Mauris enim est, vehicula eu, molestie ut, vulputate id, ante. Morbi a turpis. Vivamus semper, pede et lacinia accumsan, dolor lacus fermentum dui, ac ullamcorper augue arcu ac est. Quisque eget odio. Curabitur molestie leo a urna. Morbi ornare turpis at metus. Sed eget lorem at augue blandit rutrum. Nulla quis lectus. Maecenas a purus quis diam tincidunt convallis. Proin in nisi in sem rutrum dapibus. Sed pellentesque varius mauris. Ut purus neque, scelerisque vitae, varius vitae, fermentum ac, nulla. Aliquam erat volutpat. Maecenas vel tortor in dui tristique sagittis. Sed sed quam. Donec elit eros, interdum eu, placerat non, tempus et, nisl. In justo massa, aliquet ut, ullamcorper id, nonummy at, enim. Donec tempor. Sed congue. Donec in mauris. Vestibulum vestibulum euismod elit. Praesent ac risus. Nam tristique. Sed feugiat risus vel felis. Sed libero. Praesent nulla leo, mattis et, faucibus fermentum, venenatis sit amet, libero. Curabitur porta. Nam congue, turpis nec condimentum facilisis, risus quam eleifend dolor, ac fermentum massa dui in ipsum.

The *background* property allows us to have both a background image and a background colour on the same element (can be set independently with the *background-color* property).

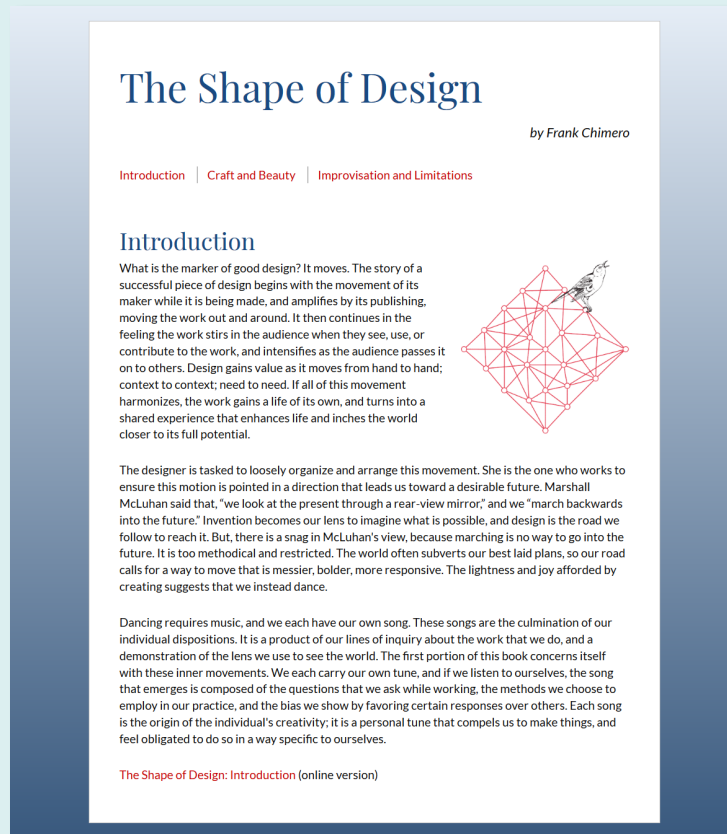
CSS3 introduced the ability to specify multiple background images. See link for details.

```
<p class="background">Lorem ipsum dolor... </p>
```

```
p.background {  
    background: #D0D5C6 url(artichoke.jpg) no-repeat 50% 50%;  
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Backgrounds_and_Borders/Using_multiple_backgrounds

Background Gradients



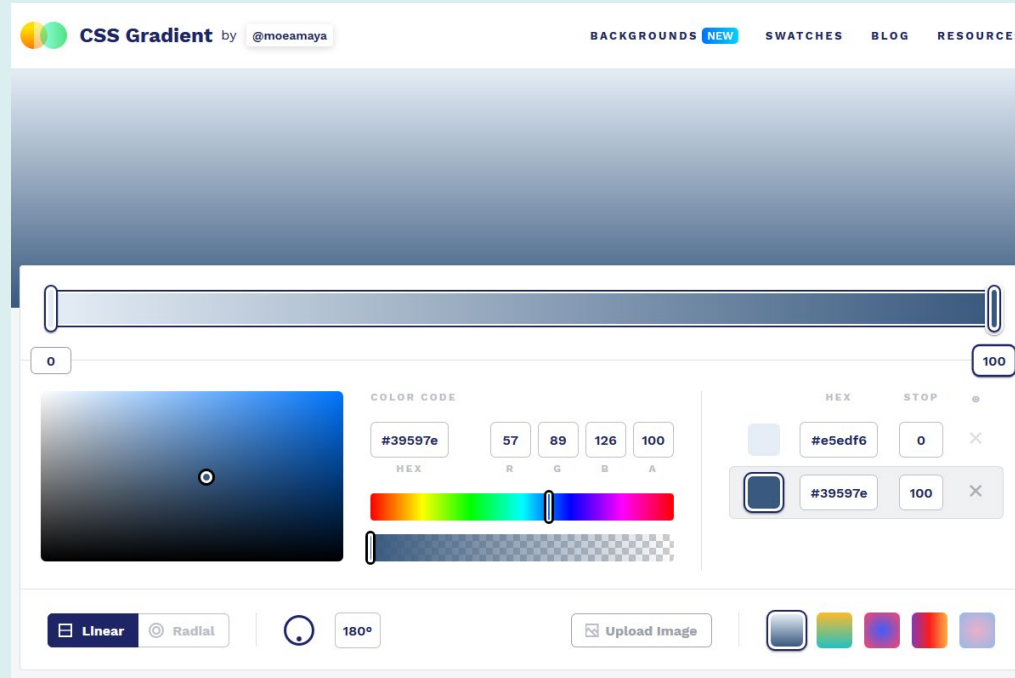
The *background* property can also be used to display a gradient. The options are comprehensive, so take a look at the documentation to find out what's possible and how to implement it.

As with background images, use gradients responsibly!

```
body {  
  background-color: #e5edf6; /*fall-back*/  
  background: linear-gradient(180deg, rgba(229,237,246,1) 0%, rgba(57,89,126,1) 100%);  
}
```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Images/Using_CSS_gradients

Gradient Generators



The syntax for CSS gradients can be quite complex so, inevitably, there are lots of gradient generator tools to help you.

<https://cssgradient.io/>

The End .