



link: https://www.w3schools.com/css/css_rwd_images.asp

Responsive Web Design - Images

[< Previous](#)[Next >](#)

Resize the browser window to see how the image scales to fit the page.

Using The width Property

If the **width** property is set to a percentage and the **height** property is set to "auto", the image will be responsive and scale up and down:

Example

Dark mode



```
width: 100%;  
height: auto;  
}
```

[Try it Yourself »](#)

Notice that in the example above, the image can be scaled up to be larger than its original size. A better solution, in many cases, will be to use the `max-width` property instead.

Using The max-width Property

If the `max-width` property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:

Example

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

[Try it Yourself »](#)

Add an Image to The Example Web Page

Example

```
img {  
  width: 100%;  
}
```

Dark mode



Try it Yourself »

Background Images

Background images can also respond to resizing and scaling.

Here we will show three different methods:

1. If the **background-size** property is set to "contain", the background image will scale, and try to fit the content area. However, the image will keep its aspect ratio (the proportional relationship between the image's width and height):



Here is the CSS code:



HTML

CSS



```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-repeat: no-repeat;  
  background-size: contain;  
  border: 1px solid red;  
}
```

[Try it Yourself »](#)

2. If the **background-size** property is set to "100% 100%", the background image will stretch to cover the entire content area:



Here is the CSS code:

[Dark mode](#)



HTML

CSS



```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-size: 100% 100%;  
  border: 1px solid red;  
}
```

[Try it Yourself »](#)

3. If the **background-size** property is set to "cover", the background image will scale to cover the entire content area. Notice that the "cover" value keeps the aspect ratio, and some part of the background image may be clipped:



Here is the CSS code:

uporer oo nicher sada gap tuku thakbey na. pura image ta ovabei thakbey , just uporer tuku

[Dark mode](#)



```
div {  
  width: 100%;  
  height: 400px;  
  background-image: url('img_flowers.jpg');  
  background-size: cover;  
  border: 1px solid red;  
}
```

Try it Yourself »

Different Images for Different Devices

A large image can be perfect on a big computer screen, but useless on a small device. Why load a large image when you have to scale it down anyway? To reduce the load, or for any other reasons, you can use media queries to display different images on different devices.

Here is one large image and one smaller image that will be displayed on different devices:



Dark mode



Example

```
/* For width smaller than 400px: */  
body {  
    background-image: url('img_smallflower.jpg');  
}  
  
/* For width 400px and larger: */  
@media only screen and (min-width: 400px) {  
    body {  
        background-image: url('img_flowers.jpg');  
    }  
}
```

Try it Yourself »

You can use the media query `min-device-width`, instead of `min-width`, which checks the device width, instead of the browser width. Then the image will not change when you resize the browser window:

Example

```
/* For devices smaller than 400px: */  
body {  
    background-image: url('img_smallflower.jpg');  
}
```

Dark mode



```
body {  
  background-image: url('img_flowers.jpg');  
}
```

Try it Yourself »

The HTML <picture> Element

The HTML `<picture>` element gives web developers more flexibility in specifying image resources.

The most common use of the `<picture>` element will be for images used in responsive designs. Instead of having one image that is scaled up or down based on the viewport width, multiple images can be designed to more nicely fill the browser viewport.

The `<picture>` element works similar to the `<video>` and `<audio>` elements. You set up different sources, and the first source that fits the preferences is the one being used:

Example

```
<picture>  
  <source srcset="img_smallflower.jpg" media="(max-width: 400px)">  
  <source srcset="img_flowers.jpg">  
    
</picture>
```

Try it Yourself »

The `srcset` attribute is required, and defines the source of the image.

The `media` attribute is optional, and accepts the media queries you find in [CSS @media rule](#).

Dark mode