mdn web docs

Viewport meta tag

This article describes how to use the "viewport" `<meta>` tag to control the viewport's size and shape.

Background

The browser's viewport is the area of the window in which web content can be seen. This is often not the same size as the rendered page, in which case the browser provides scrollbars for the user to scroll around and access all the content.

Some mobile devices and other narrow screens render pages in a virtual window or viewport, which is usually wider than the screen, and then shrink the rendered result down so it can all be seen at once. Users can then zoom and pan to look more closely at different areas of the page. For example, if a mobile screen has a width of 640px, pages might be rendered with a virtual viewport of 980px, and then it will be shrunk down to fit into the 640px space.

This is done because not all pages are optimized for mobile and break (or at least look bad) when rendered at a small viewport width. This virtual viewport is a way to make non-mobile-optimized sites in general look better on narrow screen devices.

However, this mechanism is not so good for pages that are optimized for narrow screens using media queries — if the virtual viewport is 980px for example, media queries that kick in at 640px or 480px or less will never be used, limiting the effectiveness of such responsive design techniques. The viewport `<meta>` element mitigates this problem of virtual viewport on narrow screen devices.

Viewport basics

The viewport is a comma-separated list of feature and value pairs. A typical mobile-optimized site contains something like the following:

HTML

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

Not all devices are the same width; you should make sure that your pages work well in a large variation of screen sizes and orientations.

The basic attributes of the "viewport" `<meta>` element include:

`width`

Controls the (minimum) size of the viewport (see viewport width and screen width). It can be set to a specific number of pixels like `width=600` or to the special value `device-width`, which is the physical size of the device screen in CSS pixels. This value establishes the value of the `vw` unit. Minimum: `1`. Maximum: `10000`. Negative values: ignored.

`height`

Controls the (minimum) size of the viewport (see viewport width and screen width). It can be set to a specific number of pixels like `height=400` or to the special value `device-height`, which is the physical size of the device screen in CSS pixels. This value establishes the value of the `vh` unit. Minimum: `1`. Maximum: `10000`. Negative values: ignored.

`initial-scale`

Controls the zoom level when the page is first loaded. Minimum: `0.1`. Maximum: `10`. Default: `1`. Negative values: ignored.

`minimum-scale`

Controls how much zoom out is allowed on the page. Minimum: `0.1`. Maximum: `10`. Default: `0.1`. Negative values: ignored.

`maximum-scale`

Controls how much zoom in is allowed on the page. Any value less than 3 fails accessibility. Minimum: `0.1`. Maximum: `10`. Default: `10`. Negative values: ignored.

`user-scalable`

Controls whether zoom in and zoom out actions are allowed on the page. Valid values: `0`, `1`, `yes`, or `no`. Default: `1`, which is the same as `yes`. Setting the value to `0`, which is the same as `no`, is against Web Content Accessibility Guidelines (WCAG).

`interactive-widget`

Specifies the effect that interactive UI widgets, such as a virtual keyboard, have on the page's viewports. Valid values: `resizes-visual`, `resizes-content`, or `overlays-content`. Default: `resizes-visual`.

> **Warning:** Usage of `user-scalable=no` can cause accessibility issues to users with visual impairments such as low vision. WCAG requires a minimum of 2× scaling; however, the best practice is to enable a 5× zoom.

Screen density

Screen resolutions have risen to the size that individual pixels are indistinguishable by the human eye. For example, smartphones often have small screens with resolutions upwards of 1920–1080 pixels (≈400dpi). Because of this, many browsers can display their pages in a smaller physical size by translating multiple hardware pixels for each CSS "pixel". Initially, this caused usability and readability problems on many touch-optimized websites.

On high dpi screens, pages with `initial-scale=1` will effectively be zoomed by browsers. Their text will be smooth and crisp, but their bitmap images may not take advantage of the full screen resolution. To get sharper images on these screens, web developers may want to design images – or whole layouts – at a higher scale than their final size and then scale them down using CSS or viewport properties.

The default pixel ratio depends on the display density. On a display with density less than 200dpi, the ratio is 1.0. On displays with density between 200 and 300dpi, the ratio is 1.5. For displays with density over 300dpi, the ratio is the integer floor (*density*/150dpi). Note that the default ratio is true only when the viewport scale equals 1. Otherwise, the relationship between CSS pixels and device pixels depends on the current zoom level.

Viewport width and screen width

Sites can set their viewport to a specific size. For example, the definition `"width=320, initial-scale=1"` can be used to fit precisely onto a small phone display in portrait mode. This can cause problems when the browser renders a page at a larger size. To fix this, browsers will expand the viewport width if necessary to fill the screen at the requested scale. This is especially useful on large-screen devices.

For pages that set an initial or maximum scale, this means the `width` property actually translates into a *minimum* viewport width. For example, if your layout needs at least 500 pixels of width then you can use the following markup. When the screen is more than 500 pixels wide, the browser will expand the viewport (rather than zoom in) to fit the screen:

HTML

```
<meta name="viewport" content="width=500, initial-scale=1" />
```

Other attributes that are available are `minimum-scale`, `maximum-scale`, and `user-scalable`. These properties affect the initial scale and width, as well as limiting changes in zoom level.

The effect of interactive UI widgets

Interactive UI widgets of the browser can influence the size of the page's viewports. The most common such UI widget is a virtual keyboard. To control which resize behavior the browser should use, set the `interactive-widget` property.

Allowed values are:

`resizes-visual`

The visual viewport gets resized by the interactive widget.

`resizes-content`

The viewport gets resized by the interactive widget.

`overlays-content`

Neither the viewport nor the visual viewport gets resized by the interactive widget.

HTML

```
<meta name="viewport" content="interactive-widget=resizes-content" />
```

When the viewport gets resized, the initial containing block also gets resized, thereby affecting the computed size of viewport units.

Common viewport sizes for mobile and tablet devices

If you want to know what mobile and tablet devices have which viewport widths, there is a comprehensive list of mobile and tablet viewport sizes here . This gives information such as viewport width on portrait and landscape orientation as well as physical screen size, operating system and the pixel density of the device.

Specifications

Specification
CSS Viewport Module Level 1 # viewport-meta

See also

- Article: Prepare for viewport resize behavior changes coming to Chrome on Android

Help improve MDN

Was this page helpful to you?

Yes No

Learn how to contribute.

This page was last modified on Dec 2, 2024 by MDN contributors.