THE PRACTICAL CSS3: DEVELOP AND DESIGN

CHEAT SHEET

This document is a handy reference to the syntax and browser support of all the CSS3 features I cover in my book. I will update this information as it changes; if you spot anything that is wrong/out of date, don't hesitate to let me know, on Twitter (@chrisdavidmills) or e-mail (chrisdavidmills@gmail.com).

CSS colors

www.w3.org/TR/css3-color

RGB colors: `rgb(255,0,0)`.
RGBA colors: rgba(255,0,0,0.5).
HSL colors: `hsl(0,100%,50%)`.
HSLA colors: hsla(0,100%,50%,0.5).

Fallback:

```
background-color: red;
background-color: rgba(255,0,0,0.5)
```

TABLE 1 Browser Support for CSS3 Color Units

Browser RGBA, HSL, HSLA, and Opacity	
Opera	Since 10
Firefox	Since 3.0
Safari	Since 3.1
Chrome	Since 4.0
Internet Explorer	Since 9
iOS	Since 3.2
Android	Since 2.1
Mobile Chrome	Since beta
Opera Mobile	Since 10
Opera Mini	Since 5

REM units

http://www.w3.org/TR/css3-values/

`font-size: 1.2rem`: rems work in exactly the same way as ems, except that they are always relative to the size of the root element — `<html>`.

Fallback:

```
font-size: 36px;
font-size: 3.6rem;
```

TABLE 2 Browser Support Matrix for Rem Units

Browser Rem Units	
Opera	Since 11.6
Firefox	Since 3.6
Safari	Since 5.0
Chrome	Since 6.0
Internet Explorer	Since 9
iOS	Since 4.0
Android	Since 2.1
Mobile Chrome	Since beta
Opera Mobile	12
Opera Mini	No

Selectors

http://www.w3.org/TR/selectors/

Fallback: Use Keith Clark's Selectivizr library to provide support for CSS3 selectors to old versions of IE: http://selectivizr.com.

TABLE 3 CSS Selectors Reference

Selector	Example	Description	Browser Support
Universal	`*`	Selects everything on the page.	All
Attribute	`img[alt]`	Selects all of the specified elements that have the specified attribute. Ideal for accessibility testing if you want to highlight images with and without `alt` attributes.	Not IE6 or earlier
	`img[src="alert.gif"]`	Selects all of the specified elements that have the specified attribute with the specified value. Useful for selecting specific images or other elements without needing extra IDs or classes.	Not IE6 or earlier
	`img[src^="alert"]`	Selects all of the specified elements that have the specified string at the start of the attribute value.	Not IE6 or earlier
	`img[src$="gif"]`	Selects all of the specified elements that have the specified string at the end of the attribute value.	Not IE6 or earlier
	`a[href*="uk"]`	Selects all of the specified elements that have the specified string somewhere inside the attribute value. These are useful for adding special styling or icons to specific content—for example, links to resources just about the UK or links to PDFs.	Not IE6 or earlier
	`article[class~="feature"]`	Selects all of the specified elements that have the specified string inside the attribute value, but only if it is a single value in a space-delimited list of values.	Not IE6 or earlier
	`article[id|="feature"]`	Selects all of the specified elements that have the specified string inside the attribute value, but only if it is a single value in a hyphen-delimited list of values. These last two selectors might be potentially useful if you are trying to select elements based on some kind of horrible tagging system inserted into attributes by a CMS.	Not IE6 or earlier
Descendant	`nav a`	Selects the element on the right only if it is nested somewhere inside the element(s) to the left. You can chain more than two together—for example, `nav li a`.	All
Child	`body>header`	Selects the element on the right only if it is a direct child of the element(s) to the left. You can chain more than two together—for example, `body>header>p`.	Not IE6 or earlier
Adjacent sibling	`h1 + p`	Selects the element on the right only if it comes immediately after the element on the left in the source order, and they are siblings at the same nesting level. It's perfect if, for example, you set paragraphs to have an indent on the first line but want to remove that indent for the first line after each heading.	Not IE6 or earlier

TABLE 3 CSS Selectors Reference *(continued)*

Selector	Example	Description	Browser Support
General sibling	`h1 ~ img`	Selects the element on the right only if it is a sibling (at the same nesting level) as the element on the left. It's great for setting that indent mentioned previously on each paragraph after a heading or giving a special styling only to images inside an article at the same level as a heading.	Not IE6 or earlier.
UI element pseudo-classes	`a:link`	Styles the default state of a link.	All
	`a:visited`	Styles links when they've already been visited.	All
	`img:hover`	Styles elements when they're hovered over.	All
	`input:focus`	Styles elements when they're given focus (e.g., with the keyboard).	All
	`a:active`	Styles links while they are being activated (e.g., by being clicked on).	All
	`input:valid`	Styles form inputs that contain valid data. These types of selectors are very useful for giving users hints about whether their form data is valid or not.	Not supported in IE
	`input:invalid`	Styles form inputs that contain invalid data.	Not supported in IE
	`input:enabled`	Styles enabled form inputs.	All
	`input:disabled`	Styles disabled form inputs.	All
	`input:in-range`	Styles form inputs that contain data that is inside the valid range.	Not supported in IE
	`input:out-of-range`	Styles form inputs that contain data that is outside the valid range.	Not supported in IE
Negation selector	`input:not([type="submit"])`	Styles the specified element if it isn't selected by the simple selector(s) inside the parentheses. This is useful in cases where you have several similar elements and want to select all but one or two. For example, when laying out a form you'll want to give most of the inputs an equal width but not the `submit` or `file` inputs. You can include multiple selectors to negate inside the parentheses in a comma-delimited list—for example, `input:not([type="submit"], [type="file"])`.	Not IE8 or earlier
Language selector	`p:lang(en-US)`	Styles the specified element only if it has the language inside the parentheses set on it via the `lang` attribute.	Not IE6 or earlier
Target selector	`article:target`	Styles the element only if it is the target of a link. It's incredibly cool for making content appear at the click of a button, such as overlays, information boxes, or different tabs in a tabbed interface, without needing JavaScript. The main problem is that each new state will be at a different URL, so it can break the expected Back button functionality.	Not IE8 or earlier

(continues)

Selectors *(continued)*

TABLE 3 CSS Selectors Reference *(continued)*

Selector	Example	Description	Browser Support
Structural pseudo-classes	`:root`	Styles the root element of the document, which is pretty much always `<html>`!	Not IE8 or earlier
	`li:nth-child(2n+1)`	In a series of child elements, styles the elements specified by the formula in the parentheses. So, for example, this formula would select all odd-numbered list items (1, 3, 5, etc.). It's great for zebra striping to enhance readability.	Not IE8 or earlier
	`li:nth-last-child(2n+1)`	Works the same as `nth-child` but starts at the last element and works backwards.	Not IE8 or earlier
	`p:nth-of-type(3)`	Works the same as `nth-child` but ignores elements not of the type specified. I usually use this just to select a single child element or a certain type.	Not IE8 or earlier
	`:nth-last-of-type(1)`	Works the same as `nth-of-type`, except that it counts backwards from the last child element.	Not IE8 or earlier
	`p:first-child`	Selects the first child in a series of child elements, if it is of the type specified.	Not IE8 or earlier
	`p:last-child`	Selects the last child in a series of child elements, if it is of the type specified.	Not IE8 or earlier
	`p:first-of-type`	Selects the first child of that type in a series of child elements.	Not IE8 or earlier
	`p:last-of-type`	Selects the last child of that type in a series of child elements.	Not IE8 or earlier
	`li:only-child`	Selects the only child of an element if it is of the type specified. This is useful if, for example, you want to give list items a special styling only if there is one list item present. You might want to omit the bullet point because it is pointless and looks silly if there is only one bullet!	Not IE8 or earlier
	`section p:only-of-type`	Selects the specified element if it is the only one of its type inside its parent.	Not IE8 or earlier
	`:empty`	Selects an element only if it has no children.	Not IE8 or earlier
Pseudo-elements	`p:first-letter`	Selects the first letter inside an element. It's perfect for making drop caps!	Not IE6 or earlier
	`p:first-line`	Selects the first line of text inside an element. This is good for giving an intro paragraph an interesting look—for example, putting the first line in small caps.	Not IE6 or earlier
	`a:before`	Allows you to generate content before the specified element—for example, adding icons before certain content.	Not IE7 or earlier
	`a:after`	Allows you to generate content after the specified content—for example, placing a copyright symbol after certain pieces of content.	Not IE7 or earlier

Web fonts

http://www.w3.org/TR/css3-fonts/

Including the font in your CSS:

```
@font-face {
    font-family: myfont;
    src: local("myfont"),url(myfont.ttf),url(myfont.woff);
    font-weight: bold;
    font-style: italic;
}
```

To use the font after you've included it:

```
font-face: myfont;
```

Note: You are more likely to use the bulletproof @font-face syntax, generated by http://www.fontsquirrel.com/fontface/generator.

TABLE 4 Browser Support Matrix for Font Formats

Browser/ Font Format	Support @ font-face	TrueType (.ttf)	OpenType (.otf)	Embedded OpenType (.eot)	SVG	Web Open Font Format (.woff)
Opera	Since 10	Yes	Yes	No	Yes	Yes
Firefox	Since 3.5	Yes	Yes	No	No	Yes
Safari	Since 3.1	Yes	Yes	No	Yes	Yes
Chrome	Since 4	Yes	Yes	No	Yes	Yes
Internet Explorer	.eot since 4, .woff since 9	No	No	Yes	No	Yes
iOS	SVG up to 4.1	Yes	Yes	No	Yes	Yes
Android	Since 2.2	Yes	Yes	No	Yes	No
Chrome Mobile	Since beta	Yes	Yes	No	Yes	Yes
Opera Mobile	Since beta	Yes	Yes	No	Yes	Yes

CSS text

http://dev.w3.org/csswg/css3-text/

text-shadow

```
text-shadow: 1px 1px 1px red;
```

Apply multiple text shadows separated by commas:

```
text-shadow: 1px 1px 1px red, 2px 2px 1px red;
```

text-overflow

```
overflow: hidden;
text-overflow: ellipsis;
```

word-wrap

```
word-wrap: break-word;
```

hyphens

```
hyphens: auto;
```

Available values: `none`, `manual`, `auto`

Polyfill available: Sweet Justice (https://github.com/aristus/sweet-justice)

TABLE 5 Browser Support Matrix for CSS3 Text Features

Browser	Text-shadow	Text-overflow	Word-wrap	Hyphenate
Opera	Since 9.5	Since 9 with -o-	Since 10.5	No
Firefox	Since 3.5	Since 7	Since 3.6	Since 6 with -moz-
Safari	Since 3.1	Since 3.2	Since 3.1	Since 5.1 with -webkit-
Chrome	Since 4	Since 4	Since 4	No
Internet Explorer	Since 10	Since 6	Since 5.5	Since 10 with -ms-
iOS	Since 3.2	Since 3.2	Since 3.2	Since 4.2 with -webkit-
Android	Since 2.1	Since 2.1	Since 2.1	No
Chrome Mobile	Since beta	Since beta	Since beta	No
Opera Mobile	Since 10	Since 10 with -o-	Since 10	No
Opera Mini	No	Since 5 with -o-	Since 5	No

CSS typography

http://www.w3.org/TR/css3-fonts/

text-rendering

```
text-rendering: optimizeLegibility
```

Available values `optimizeLegibility`, `optimizeSpeed`, `geometricPrecision`, `auto`.

font-feature-settings

```
font-feature-settings: "dlig" 1, "kern" 1, "frac" 1;
```

The different font features available are:

- `"liga" 1` — Ligatures
- `"dlig" 1` — Discretionary ligatures
- `"onum" 1` — Ordinals, or old-style numerals
- `"tnum" 1` — Tabular numberals
- `"frac" 1` — Fractions
- `"smcp" 1` — Small caps
- `"pcap" 1` — Petite caps
- `"kern" 1` — Kerning
- `"ss01" 1` — Alternative stylistic set
- `"swsh" 1` — Stylistic swashes
- `"cswh" 1` — Contextual swashes
- `"calt" 1` — Contextual alternatives

Note that sometimes there may be multiple sets of these features available in a font, which you can swap between using different numbers, so, for example, `"dlig" 1`, `"dlig" 2`, `"dlig" 3`.

TABLE 6 Browser Support Matrix for CSS3 Text Features

Browser	text-rendering: optimizeLegibility	Font-feature-settings
Opera	No	No
Firefox	Since 8	Since 8 with -moz-
Safari	Since 5.1	No
Chrome	Since 13	Since 17 with -webkit-, only on Windows
Internet Explorer	Since 10	Since 10 with -ms-*
iOS	No	No
Android	No	No
Chrome Mobile	Since beta	No
Opera Mobile	No	No
Opera Mini	No	No

*IE10 platform preview 6 saw the prefix removed

CSS3 bling features

http://www.w3.org/TR/css3-background/
http://dev.w3.org/csswg/css3-images/

border-radius

`border-radius: 10px;`
`border-radius: 10px/20px;` for separate horizontal and vertical radii

To add support for border-radius to older versions of IE, use http://css3pie.com.

box-shadow

```
box-shadow: inset 1px 1px 3px 4px black;
```

The `inset` keyword (which causes it to be an inner shadow) and the fourth unit value (spread) are optional.

To add support for border-radius to older versions of IE, use http://css3pie.com, but bear in mind that CSS3PIE has limited support for RGBA colours, dropping the alpha channel and rendering them as the full opaque version. It is often better to provide an alternative style with a nontransparent colour that might be more effective.

Multiple backgrounds

```
background: url(../images/castle.png) top left no-repeat,
            url(../images/clouds.png) top right no-repeat,
            linear-gradient(top right, #3B6498, #C1CDDB);
```

Images later in the property value appear behind those earlier on, which is rather contrary to the way CSS usually works.

For browsers that don't support multiple background images, include a fallback background or background color, for example:

```
background: url(../images/clouds.png) top right no-repeat; or
background-color: #C1CDDB;
```

Linear gradients

```
background-image: linear-gradient(to top,red,green);
```

With colour stop positions specified:

```
background-image: linear-gradient(to top, red 25%, green 75%);
background-image: linear-gradient(to top, red 100px, green 200px);
```

Repeating linear gradient:

```
background: repeating-linear-gradient(to top right, red 10px, green 20px, black 30px);
```

Radial gradients

```
background-image: radial-gradient(50% 50%, 60% 60%, red, black);
```

The different parts of this syntax are as follows:

Implicit gradient size:

```
background-image: radial-gradient(50% 50%, circle closest-side, red, black);
Implicit keywords: circle closest-side,
                   ellipse closest-side,
                   circle closest-corner,
                   ellipse closest-corner,
                   circle farthest-side,
                   ellipse farthest-side,
                   circle farthest-corner,
                   ellipse farthest-corner,
                   contain,
                   cover
```

`contain` is equivalent to `closest-side`; cover is equivalent to `farthest-corner`.

With colour stop positions specified:

```
background-image: radial-gradient(50% 50%, 60% 60%, red 25%, black 75%);
```

Repeating radial gradient:

```
background: -o-repeating-radial-gradient(50% 50%, 60% 60%, red 10px, green 20px, black 30%);
```

To provide fallbacks for gradients in old versions of IE, use http://css3pie.com, but again, beware of limited RGBA support. To use CSS3PIE, target a separate, nontransparent colour gradient to IE using a special `-pie-` prefixed background property (bear in mind that CSS3PIE doesn't add support for `background-image`, just the shorthand `background`).

CSS3 bling features *(continued)*

Border images

```
border-image: url(border.png) 30 30 30 30 round;
border-width: 30px;
```

`round` can be substituted with `repeat` or `stretch`.

Note that you need to specify a `border-width` for the `border-images` to be shown. If you change the `border-width`, the border image slices will adjust in size so that they fit inside the borders.

`border-image-slice: fill;` could be used to explicitly preserve the middle section of the image, inside the slices, except that no browser currently supports it.

background-clip

```
background-clip: border-box;
```

Available values: `border-box`, `padding-box`, `content-box`

box-decoration-break

```
box-decoration-break: clone;
```

TABLE 7 Browser Support Matrix for CSS3 "Bling Box" Features

Browser	RGBA/HSLA	Border-radius	Box-shadow	Multiple backgrounds	Gradients	Background-size	Border-image	Background-clip	Box-decoration-break
Opera	version 10.5	version 10.5	version 10.5	version 10.5	11.6 with -o-	version 10.5	11 with -o-	version 10.5	version 10.5
Firefox	version 3	3 with -moz-, 4 prefixless	3.5 with -moz-, 4 prefixless	version 3.6	3.6 with -moz-	version 4	3.5 with -moz-	version 4	Exact details unknown
Safari	version 3.1	3.1 with -webkit-, 5 prefixless	3.1 with -webkit-, 5 prefixless	version 3.1	5.1 with -webkit-	version 5	3.1 with -webkit-	version 5	Exact details unknown
Chrome	version 4	4 with -webkit-, 5 prefixless	4 with -webkit-, 10 prefixless	version 4	10 with -webkit-	version 4	15 with -webkit-	version 4	Exact details unknown
Internet Explorer	version 9	version 9	version 9	version 9	10 with -ms- *	version 9	10 with -ms-	version 9	Exact details unknown
iOS	version 3.2	version 4	3.2 with -webkit-, 5 prefixless	version 3.2	5.0 with -webkit-	version 5	3.2 with -webkit-	version 5	Exact details unknown
Android	version 2.1	2.1 with -webkit-, 2.2 prefixless	2.1 with -webkit-	version 2.1	4 with -webkit-	version 2.1	2.3 with -webkit-	version 2.1	Exact details unknown
Opera Mobile	version 10	version 11	version 11	version 10	11.5 with -o-	version 10	11 with -o-	version 10	Exact details unknown
Opera Mini	version 5	no	no	version 5	no	version 5.0	no	version 5	Exact details unknown

*IE10 platform preview 6 saw support without the prefix

Transforms, transitions, and animations

www.w3.org/TR/css3-2d-transforms
www.w3.org/TR/css3-3d-transforms
www.w3.org/TR/css3-transitions
www.w3.org/TR/css3-animations

Transforms

```
transform: rotate(45deg);
```

The available 2D transform functions are as follows:

- `translate(100px, 200px)`, `translateX(100px)`, `translateY(100px)`
- `rotate(45deg)`, `rotateZ(45deg)`
- `scale(1.25,1.1)`
- `scale(1.25)`, `scaleX(1.25)`, `scaleY(1.25)`
- `skew(10deg,13deg)`, `skewX(10deg)`, `skewY(10deg)`
- `matrix(1, 0, 0, 1, 150, 150)`

The available 3D transform functions are as follows:

- `translateZ(100px)`, `translate3D(100px, 200px, 300px)`
- `rotateX(180deg)`, `rotateY(180deg)`, `rotate3D(1,1,1,180deg)`
- `scaleZ(1.25)`, `scale3D(1.1, 1.25, 1.25)`
- `perspective(250px)`
- `matrix3D(.8, 0, 0, 0, 0, .5, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1);`
  — transform the element in three dimensions using a matrix.

Other transform-related properties are as follows:

- `transform-origin: center;`
  — Set in the same way as `background-position`
- `perspective: 800;`
- `perspective-origin: center;`
  — set in the same way as `background-position`
- `transform-style: preserve-3d;`
  — possible values are `flat` (default) and `preserve-3d`
- `backface-visibility: hidden;`
  — possible values are `visible` (default) and `hidden`

Applying multiple transforms:

```
transform: skew(-3deg,-10deg) translateX(-200px) scale(0.7)
           rotate(10deg);
```

TABLE 8 Browser Support for Transforms

Browser	2D Transforms	3D Transforms
Opera	10.6 with -o-	No
Firefox	3.5 with -moz-	10 with -moz-
Safari	3.1 with -webkit-	4 with -webkit-
Chrome	4 with -webkit-	12 with -webkit-
Internet Explorer	9 with -ms-	10 with -ms-
iOS	3.2 with -webkit-	3.2 with -webkit-
Android	2.1 with -webkit-	3 with -webkit-
Mobile Chrome	Since beta with -webkit-	Since beta with -webkit-
Opera Mobile	11 with -o-	No
Opera Mini	No	No

Note that in IE10 platform preview 6, support for non-prefixed transform properties has been added.

Transforms, transitions, and animations *(continued)*

Transitions

Transition properties:

```
transition-property: background-color;
transition-duration: 0.5s;
transition-delay: 0.5s;
```

`transition-timing-function;` — available timing functions are `linear`, `ease`, `ease-out`, `ease-in` `ease-in-out`, `steps(`*n*`)` and `cubic-bezier(a, b, x, y)`

Transition shorthand:

```
transition: opacity 3s ease-in 1s;
```

is equivalent to

```
transition-property: opacity;
transition-duration: 3s;
transition-timing-function: ease-in;
transition-delay: 1s;
```

Multiple transitions:

```
transition: opacity 3s ease-in 1s, height 4s linear;
transition-property: opacity, height;
```

TABLE 9 Browser Support for Transitions

Browser	Transitions
Opera	10.5 with -o-
Firefox	4 with -moz-
Safari	3.1 with -webkit-
Chrome	4 with -webkit-
Internet Explorer	10 with -ms-
iOS	3.2 with -webkit-
Android	2.1 with -webkit-
Mobile Chrome	Since beta with -webkit-
Opera Mobile	10 with -o-
Opera Mini	No

Note that in IE10 platform preview 6, support for non-prefixed transition properties has been added.

Animations

Specifying animation keyframes with `from` and `to`:

```
@keyframes menu-move {
   from { transform: translateY(-200px); }
   to { transform: translateY(0); }
}
```

Specifying animation keyframes with percentages:

```
@keyframes menu-move {
   0% { transform: translateY(-200px); }
   50% { transform: translateY(0); }
   100% { transform: translateY(-200px); }
}
```

You can group keyframes that are the same, for example:

```
@keyframes menu-move {
   0%, 100% { transform: translateY(-200px); }
   50% { transform: translateY(0); }
}
```

Animation properties:

- `animation-name: my-animation;`
- `animation-duration: 2s;`
- `animation-iteration-count: 5;`
- `animation-timing-function;` — available timing functions are `linear`, `ease`, `ease-out`, `ease-in` `ease-in-out`, `steps(`*n*`)`, and `cubic-bezier(a, b, x, y)`
- `animation-delay: 2s;`
- `animation-direction: alternate;`
- `animation-fill-mode: forwards;` — possible values are `none`, `forwards`, `backwards`, and `both`

Animation shorthand:

```
animation: whoosh 10s ease-in 3s 25 alternate backwards;
```

is equivalent to:

```
animation-name: whoosh;
animation-duration: 10s;
animation-timing-function: ease-in;
animation-delay: 3s;
animation-iteration-count: 25;
animation-direction: alternate;
animation-fill-mode: backwards;
```

If you don't explicitly specify all the values, their default values will come into play, which are:

```
animation-timing-function: ease;
animation-delay: 0s;
animation-iteration-count: 1;
animation-direction: normal;
animation-fill-mode: none;
```

Multiple animations:

```
animation: whoosh 10s, zap 5s;
animation-name: whoosh, zap;
animation-duration: 10s, 5s;
```

TABLE 10 Browser Support for Animations

Browser	Animations
Opera	12 with -o-
Firefox	5 with -moz-
Safari	4 with -webkit-
Chrome	4 with -webkit-
Internet Explorer	10 with -ms-
iOS	3.2 with -webkit-
Android	4 with -webkit-
Mobile Chrome	Since Beta with -webkit-
Opera Mobile	No
Opera Mini	No

Note that in IE10 platform preview 6, support for non-prefixed animation properties has been added.

Providing fallbacks for transforms, transitions, and animations

Polyfills and graceful degradation are often difficult with these CSS features, especially when they are being used to create effects that are relied on to display crucial content. I would recommend building your own graceful fallbacks, feature detection using Modernizr (www.modernizr.com) and then providing alternative styles/scripting.

CSS3 layout chops

Multi-col

Multi-col properties:

- `column-count: 3;`
- `column-width: 20rem`
- `column-rule: 3px solid #8B2101;` — column-rule-width: 3px;,
- `column-rule-style: solid;` — column-rule-color: #8B2101;
- `column-gap: 2rem;`
- `column-fill: balance;`
- `column-span: all;` — possible values are `1` (default) and `all`
- `break-before: always;` — possible values are `auto`, `always`, `avoid`, `column`, and `avoid-column`
- `break-after: always;` — possible values are `auto`, `always`, `avoid`, `column`, and `avoid-column`
- `break-inside: avoid;` — possible values are `auto`, `avoid`, and `avoid-column`

NOTE: There are other `break- property` values besides those listed, but they only apply to paged media: `avoid-page`, `left`, `page`, and `right`.

Flexbox

Flexbox properties:

- `display: flex;` — possible values are `flex` and `inline-flex`
- `flex-flow: row wrap;` — most common values are `row`, `row wrap`, `column` and `column wrap`

NOTE: `flex-flow` is a shorthand property for:
`flex-wrap` possible values — `no-wrap`, `wrap`, `wrap-reverse`
`flex-direction` possible values — `row`, `row-reverse`, `column`, `column-reverse`

- `order: 1;`
- `justify-content: center;` — possible values are `flex-start`, `flex-end`, `center`, `space-between` and `space-around`
- `align-items: center;` — possible values are `flex-start`, `flex-end`, `center`, `baseline` and `stretch`
- `flex: 1, 2, 200px;`

NOTE: `flex` is actually a shorthand property. You can set the three values it holds individually using `flex-grow`, `flex-shrink`, and `flex-basis`.

I wouldn't advise you try to use Flexbox in production projects right now, as it is not supported very well across browsers. If you decide to do so, you'll need to serve browsers that support the old 2009 Flexbox syntax (currently Firefox, IE, and WebKit browsers) using Modernizr. The site http://wiki.csswg.org/spec/flexbox-2009-2011-spec-propertymapping provides a fairly useful table showing the mapping between the old and new syntaxes, although this is now somewhat out of date. Peter Gasston, tech reviewer for this book, has written a useful guide to the old syntax at www.netmagazine.com/tutorials/css3-flexible-box-model-explained.

Grids

Grid properties:

- `display: grid;`
- `grid-columns: 30rem 40rem 30rem;`
- `grid-rows: 1fr 1fr 1fr 1fr;`
- `grid-column: 1;`
- `grid-column-span: 2;`
- `grid-row: 1;`
- `grid-row-span: 2;`

TABLE 11 Browser Support Matrix for CSS3 Layout Features

Browser	Multi-col	Flexbox	Grids	Regions	Exclusions	GCPM
Opera	Since 11.1	No	No	No	No	Since 12****
Firefox	2.0 with -moz-*	2.0 with -moz-**	No	No	No	No
Safari	3.2 with -webkit-*	3.1 with -webkit-**	No	6.0 with -webkit-***	No	No
Chrome	4.0 with -webkit-*	17 with -webkit-	No	16 with -webkit-***	No	No
Internet Explorer	10	10 with -ms-**	10 with -ms-	10 with -ms-	No	No
iOS	3.2 with -webkit-*	5.0 with -webkit-	No	No	No	No
Android	2.1 with -webkit-*	2.1 with -webkit-**	No	No	No	No
Mobile Chrome	Beta with -webkit-*	Beta with -webkit-	No	No	No	No
Opera Mobile	Since 11.1	No	No	No	No	Since 12****
Opera Mini	No	No	No	No	No	No

* Indicates limited multi-col support without support for break-before/after/inside.

** Indicates support for the old version of the Flexbox spec, which is deprecated (see the "Using Flexbox" section for more details).

*** Safari and Chrome at the time of this writing support regions with a different mechanism than that of IE10. See the "Regions" section for more details.

**** At the time of this writing, GCPM was only supported to a limited extent in a special Opera Labs build. See the "GCPM" section for more details.

Responsive design

Media queries

Basic syntax 1: in a stylesheet:

```
@media screen and (max-width: 800px) { ... }
```

Basic syntax 2: in a `<link>` element:

```
<link href="style.css" rel="stylesheet" type="text/css" media="screen and (max-width: 800px)">
```

Targeting all media types:

```
@media all and (max-width: 800px)
@media (max-width: 800px)
```

Targetting your media query NOT at browsers that pass the specified rule:

```
@media not screen and (max-width: 800px) { ... }
```

Enforcing multiple conditions to make the test stricter:

```
@media screen and (max-width: 800px) and (orientation: landscape) { ... }
```

Making the test pass under different sets of conditions:

```
@media screen and (max-width: 800px), print and (max-width: 29.7cm) { ... }
```

Most common tests:

- `min/max-width` and `min/max-height`
- `min/max-device-width` and `min/max-device-height`
- `aspect-ratio/device-aspect-ratio`
- `orientation`
- `min/max-resolution`
- `min/max-device-pixel-ratio`

To make media queries backwards compatible, consider the following two JavaScript polyfills:

- css3-mediaqueries.js: http://code.google.com/p/css3-mediaqueries-js
- respond.js: https://github.com/scottjehl/Respond

Viewport

Basic syntax:

```
<meta name="viewport" content="width=device-width, target-densitydpi=device-dpi">
```

Different available viewport settings:

- `width=320`
- `initial-scale=1.5`
- `target-densitydpi=device-dpi/target-densitydpi=160`
- `maximum-scale=2, minimum-scale=0.5`
- `user-scalable=no`

TABLE 12 Browser Support for Responsive Design Features

Browser	Media Queries	`<source>` media attribute	Viewport meta tag	@viewport CSS
Opera	Since 9.5	?*	n/a**	n/a**
Firefox	Since 3.5	?*	n/a**	n/a**
Safari	Since 4	?*	n/a**	n/a**
Chrome	Since 4	?*	n/a**	n/a**
Internet Explorer	Since 9	?*	n/a**	n/a**
iOS	Since 3.2	?*	Since 3.2	No
Android	Since 2.1	?*	Since 2.1	No
Mobile Chrome	Since beta	?*	Since beta	No
Opera Mobile	Since 10	?*	Since 10.0	Since 11.0
Opera Mini	Since 5	?*	Since 6.0	No

* It was difficult to find any concrete, detailed support data for the `<source>` media attribute. But from my tests, it seems to work well across all the latest desktop and mobile browsers.

** The Viewport meta tag and @viewport aren't relevant to nonmobile browsers.