**Gradients** [**https://tj.github.io/nib/**](https://tj.github.io/nib/)

Nib's gradient support is by far the largest feature it provides, not only is the syntax extremely similar to what you would normally write, it's more forgiving, expands to vendor equivalents, and can even produce a PNG for older browsers with [node-canvas](http://github.com/learnboost/node-canvas).

body {

background: linear-gradient(top, white, black);

}

yields:

body {

background: -webkit-gradient(linear, left top, left bottom, color-stop(0, #fff), color-stop(1, #000));

background: -webkit-linear-gradient(top, #fff 0%, #000 100%);

background: -moz-linear-gradient(top, #fff 0%, #000 100%);

background: linear-gradient(top, #fff 0%, #000 100%);

}

http://f.cl.ly/items/1q25061X2Q2U0p472L02/Screenshot.png

Any number of color stops may be provided:

body {

background: linear-gradient(bottom left, white, red, blue, black);

}

http://f.cl.ly/items/2I0k3D0A2y0n3i443g2W/Screenshot.png

Units may be placed before, or after the color:

body {

background: linear-gradient(left, 80% red, #000);

background: linear-gradient(top, #eee, 90% white, 10% black);

}

http://f.cl.ly/items/2B1U3m0t2T1B420I3C3I/Screenshot.png http://f.cl.ly/items/1T1P1x0n1X3k132o3V0F/Screenshot.png

**Position mixins**

The position mixins absolute, fixed, and relative provide a shorthand variant to what is otherwise three CSS properties. The syntax is as follows:

fixed|absolute|relative: top|bottom [n] left|right [n]

The following example will default to (0,0):

#back-to-top {

fixed: bottom right;

}

yielding:

#back-to-top {

position: fixed;

bottom: 0;

right: 0;

}

You may also specify the units:

#back-to-top {

fixed: bottom 10px right 5px;

}

yielding:

#back-to-top {

position: fixed;

bottom: 10px;

right: 5px;

}

**Clearfix**

The clearfix mixin currently takes no arguments, so it may be called as shown below:

.clearfix {

clearfix();

}

yielding:

.clearfix {

zoom: 1;

}

.clearfix:before,

.clearfix:after {

content: "";

display: table;

}

.clearfix:after {

clear: both;

}

**Border radius**

Nib's border-radius supports both the regular syntax as well as augmenting it to make the value more expressive.

button {

border-radius: 1px 2px / 3px 4px;

}

button {

border-radius: 5px;

}

button {

border-radius: bottom 10px;

}

yielding:

button {

-webkit-border-radius: 1px 2px/3px 4px;

-moz-border-radius: 1px 2px/3px 4px;

border-radius: 1px 2px/3px 4px;

}

button {

-webkit-border-radius: 5px;

-moz-border-radius: 5px;

border-radius: 5px;

}

button {

-moz-border-radius-topleft: 10px;

-webkit-border-top-left-radius: 10px;

border-top-left-radius: 10px;

-moz-border-radius-bottomright: 10px;

-webkit-border-bottom-right-radius: 10px;

border-bottom-right-radius: 10px;

}

**Responsive**

The image mixin allows you to define a background-image for both the normal image, and a doubled image for devices with a higher pixel ratio such as retina displays. This works by using a @media query to serve an "@2x" version of the file.

#logo {

image: '/images/branding/logo.main.png'

}

#logo {

image: '/images/branding/logo.main.png' 50px 100px

}

yields:

#logo {

background-image: url("/images/branding/logo.main.png");

}

@media all and (-webkit-min-device-pixel-ratio: 1.5) {

#logo {

background-image: url("/images/branding/logo.main@2x.png");

background-size: auto auto;

}

}

#logo {

background-image: url("/images/branding/logo.main.png");

}

@media all and (-webkit-min-device-pixel-ratio: 1.5) {

#logo {

background-image: url("/images/branding/logo.main@2x.png");

background-size: 50px 100px;

}

}

**Ellipsis**

The overflow property is augmented with a "ellipsis" value, expanding to what you see below.

button {

overflow: ellipsis;

}

yielding:

button {

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

**Reset**

Nib comes bundled with Eric Meyer's style reset support, you can choose to apply the global or any specifics that you wish. To view the definitions view [reset.styl](https://github.com/visionmedia/nib/blob/master/lib/nib/reset.styl)

* global-reset()
* nested-reset()
* reset-font()
* reset-box-model()
* reset-body()
* reset-table()
* reset-table-cell()
* reset-html5()

**Miscellaneous properties**

The following properties follow vendor expansion much like border-radius, however without augmentation, as well as some aliases such as whitespace instead of white-space.

* no-wrap == nowrap
* whitespace == white-space
* box-shadow
* user-select
* column-count
* column-gap
* column-rule
* column-rule-color
* column-rule-width
* column-rule-style
* column-width
* background-size
* transform
* border-image
* transition
* transition-property
* transition-duration
* transition-timing-function
* transition-delay
* backface-visibility
* opacity
* box-sizing
* box-orient
* box-flex
* box-flex-group
* box-align
* box-pack
* box-direction
* animation
* animation-name
* animation-duration
* animation-delay
* animation-direction
* animation-iteration-count
* animation-timing-function
* animation-play-state
* animation-fill-mode
* border-image
* hyphens
* appearance