Stylus CSS Preprocessor Cheatsheet

A comprehensive cheat sheet covering Stylus, a CSS preprocessor, with syntax, features, and practical examples to enhance your CSS workflow.

Getting Started with Stylus

Basic Syntax

CSS Syntax (with Stylus)

```
.box {
  color: blue;

  .button {
    color: red;
  }
}
```

Indent Syntax (Stylus)

```
.box
  color: blue

  .button
    color: red
```

Stylus supports both CSS and indented syntax.

Key Differences

- Colon is optional.
- Indentation defines nesting.
- Parent referencing with `&` is supported.

Variables

Defining Variables

```
royal-blue = #36a

div
  color: royal-blue
```

Variables store values for reuse.

Conditional Assignment

```
royal-blue ?= #89f
```

`?=` assigns only if the variable is not yet defined.

Mixins

Basic Mixin

```
red-border()
  border: solid 2px red

div
  red-border()
```

Reusable blocks of declarations.

Mixin with Arguments

```
border-radius(n)
  -webkit-border-radius: n
  border-radius: n

div
  border-radius(2px)
```

Mixins can accept arguments for customization.

Argument Defaults

```
border-radius(n = 2px)
  -webkit-border-radius: n
  border-radius: n
```

Arguments can have default values.

Block Mixins

```
mobile()
  @media (max-width: 480px)
    {block}

+mobile()
  width: 10px
```

Mixins that wrap blocks of code.

Rest Params

```
shadow(offset-x, args...)
  box-shadow: offset-x args

#login
  shadow: 1px 2px 5px #eee
```

Allows accepting variable number of arguments.

Functions and Operators

Functions

Defining Functions

```
add(a, b)
  a + b

body
  padding: add(10px, 5)
```

Functions perform calculations and return values.

Argument Defaults in Functions

```
add(a, b = 2)
  a + b
```

Arguments can have default values if not provided.

Named Parameters

```
shadow(x, y)
  x y (y * 1.5) #000

.button
  box-shadow: shadow(x: 2, y: 4)
```

Arguments can be passed by name.

Multiple Return Values

```
sizes()
  8px 16px

sizes()[0]  // → 8px
sizes()[1]  // → 16px
```

Functions can return multiple values as a list.

Values and Interpolation

Property Lookup

```
.logo
  width: w = 150px
  margin-left: -(w / 2)
  // or
  height: 80px
  margin-top: -(@height / 2)
```

Refer to other property values.

Interpolation

```
prefix = webkit

{
  -{prefix}-border-radius: 2px
}
```

Dynamically construct property names and values.

Color Operators

Stylus provides operators to manipulate colors:

- `#888 + 50%` → `#c3c3c3` (lighten)
- `#888 - 50%` → `#444` (darken)
- `#f00 + 50deg` → `#ffd500` (hue)

Casting

```
n = 5px

foo: (n)em
foo: (n * 5)%
```

Advanced Features

Conditionals

If/Else Statements

```
color = blue

if color == blue
  display: block
else if true and true
  display: inline
else
  display: none
```

Control flow based on conditions.

Aliases

- `==` is `is`
- `!=` is `is not`
- `!=` is `isnt`

Loops

For Loops

```
font-size-1 = 10px
font-size-2 = 20px
font-size-3 = 30px

for i in 1..3
  .text-{i}
    font-size: lookup('font-size-'
+ i)
```

Iterate over a range of values.

Definition and Type Checks

Definition Check

```
if ohnoes is defined
  color: blue
```

Check if a variable is defined.

Type Check

```
if val is a 'string'
if val is a 'ident'
if #fff is a 'rgba'    // →
true
```

Check the type of a value.

Built-in Functions and Utilities

Color Functions

Alpha

- `alpha(#fff)` → `1`
- `alpha(rgba(0, 0, 0, 0.2))` → `0.2`

Get the alpha (transparency) value of a color.

Dark/Light

- `dark(black)` → `true`
- `light(black)` → `false`

Check if a color is dark or light.

Hue/Saturation/Lightness

- `hue(#0a0)` → `120deg`
- `saturation(#f00)` → `100%`
- `lightness(#f00)` → `50%`

Extract color components.

Adjusting Color Components

- `hue(#0a0, 0deg)`
- `saturation(#f00, 50%)`
- `lightness(#f00)`

Set color components.

Color Manipulation

- `lighten(color, 10%)`
- `darken(color, 10%)`
- `saturate(color, 10%)`
- `desaturate(color, 10%)`
- `invert(color)`

Modify color properties.

Tint/Shade

- `tint(color, 50%)` // mix with white
- `shade(color, 50%)` // mix with black

Mix colors with white or black.

Utility Functions

Unquote

`unquote(string)`

Removes quotes from a string.

Image Size

```
width:  image-size('tux.png')[0]
height: image-size('tux.png')[1]
```

Returns the width and height of an image.

Caching

```
size($width)
  +cache('w' + $width)
    width: $width
.a { size: 10px }
.b { size: 10px }

// yields: .a, b { width: 10px }
```

Caches the result of a mixin.

Add Property

```
gradient(color)
  add-property('background-image', linear-gradient(top,
color, darken(color, 20%)))
  color

body
  background: gradient(red)
```

Adds a property to the current block.

Sprintf

```
'-webkit-gradient(%s, %s, %s)' % (linear (0 0) (0 100%))
// → -webkit-gradient(linear, 0 0, 0 100%)

s("rgba(0, 0, 0, %s)", 0.3)
```

Embed URL

```
background: embedurl('logo.png')
// → background: url("...")
```

Embeds a file as a data URL.