HTML

```
<link rel="stylesheet" href="path/to/cssp/cssp.php?file1.cssp;file2.cssp;file3.cssp">
```

Only files ending with .cssp will be processed by cssp – all other format will be merged with the cssp output unchanged

Syntax

CSSP	Resulting CSS

`@media screen`
 	
`#foo, bar`
`        color:red`
`        font-size:10px`
`        em, i`
`                font-style:italic`
 	
`@media print`
 	
`#foo`
`        color:blue`
 	
`@media screen`
 	
`#foobar`
`        background:yellow`
	`@media screen {`
	`        #foo, bar {`
	`                color: red;`
	`                font-size: 10px;`
	`        }`
	`        #foo em, #foo i, bar em, bar i {`
	`                font-style: italic;`
	`        }`
	`        #foobar {`
	`                background: yellow;`
	`        }`
	`}`
	
	`@media print {`
	`        #foo {`
	`                color: blue;`
	`        }`
	`}`

Configuration

```
@cssp
        compress:1                      // Activiate minified css output
        plugins:plugin1, plugin2, plugin3 // List of plugins to use
```

Reserved Prefixes

Prefix	Context	Effect
$	Selector	Selector alias
	Value	Constant
$_	Value	Special constant
?	Selector	Prototype, will be removed before output
&	Value	Function

Aliases

CSSP	Resulting CSS

`@aliases`
`        header: #wrapper > div.header h1`
`$header`
`        color:red`
	`#wrapper > div.header h1 {`
	`        color: red;`
	`}`

Constants

CSSP	Resulting CSS

`@constants`
`        mainColor:#FF0`
`#foo`
`        color:$mainColor`
	`#foo {`
	`        color: #FF0;`
	`}`

Special Constants

`$_CSSPPATH // Path to css.php`	`$_FILEPATH // Path to current cssp file`

Extending elements

CSSP	Resulting CSS

`#foo`
`        color:red`
`#bar`
`        extends:#foo`
`        background:green`
	`#foo {`
	`        color: red;`
	`}`
	`#bar {`
	`        background: green;`
	`        color: red;`
	`}`

To extend from multiple sources quote the selectors e.g. `extends:"#foo", "#bar", "div.footbar"`

Copying values from other elements

CSSP	Resulting CSS

`#foo`
`        color:copy(#bar background)`
`#bar`
`        background:#F00`
	`#foo {`
	`        color: #F00`
	`}`
	`#bar {`
	`        background: #F00`
	`}`

Reset stylesheet plugin (`resetstyle`)

Includes a simple reset stylesheet

Border radius plugin (`borderradius`)

Automatic vendor-specific versions of `border-radius` plus new properties `border-top-radius`, `border-right-radius`, `border-bottom-radius` and `border-left-radius`.

CSSP	Resulting CSS

`#foo`
`        border-radius:4px;`
	`#foo {`
	`        -moz-border-radius: 4px;`
	`        -khtml-border-radius: 4px;`
	`        -webkit-border-radius: 4px;`
	`        border-radius: 4px;`
	`}`

Box shadow plugin (`boxshadow`)

Automatic vendor-specific versions of `box-shadow`.

CSSP	Resulting CSS

`#foo`
`        box-shadow:4px 4px 4px #666;`
	`#foo {`
	`        -moz-box-shadow: 4px 4px 4px #666;`
	`        -webkit-box-shadow: 4px 4px 4px #666;`
	`        box-shadow: 4px 4px 4px #666;`
	`}`

Bullet proof @font-face plugin (`fontface`)

Generates bullet proof @font-face declarations from a simplified syntax.

CSSP	Resulting CSS

`@font-face`
`        font-family:'Graublau Web';`
`        local:'Graublau Web Regular';`
`        otf:url('GraublauWeb.otf');`
`        eot:url('GraublauWeb.eot');`
`        font-weight:bold;`
`        font-style:italic;`
	`@font-face {`
	`        font-family: 'Graublau Web';`
	`        font-weight: bold;`
	`        font-style: italic;`
	`        src: url('GraublauWeb.eot');`
	`        src: local('Graublau Web Regular'),`
	`                local('Graublau Web'),`
	`                url('GraublauWeb.otf')`
	`                format(opentype);`
	`}`

Minifier plugin (`minifier`)

Performs a number of micro-optimizations (shortening hex codes and the like).

Data URI plugin (`datauri`)

Injects all images smaller than 24kb as a base64-encoded string into the CSS for all dataURI-capable browsers

Browser, engine and device detection plugin (`browser`)

Detects browsers, rendering engines and devices and excludes/includes CSS elements accordingly.

Match by browser	Negative matching

`// Will show up only in firefox`
`#foo`
`        color:red`
`        browser:firefox`
	`// Will show up everywhere EXCEPT in firefox`
	`#foo`
	`        color:red`
	`        browser:^firefox`

Match by browser version	Match by engine

`// Will not show up in IE6 and lower`
`#foo`
`        color:red`
`        browser:^msie<=6`
	`// Will show up only in browsers using gecko`
	`#foo`
	`        color:red`
	`        engine:gecko`

Match by device	

`// Will show up only on mobile devices`
`#foo`
`        color:red`
`        device:mobile`
	To show/hide whole cssp files for specific browsers, engines or devices (e.g. to create mobile-only designs or hide ie hacks from real browsers) simply add browser, engine or device rules to your @cssp configuration element.

Operating system detection plugin (`os`)

Detects operating systems and excludes/includes CSS elements accordingly.

Match by os	

`// Will show up only on windows`
`#foo`
`        color:red`
`        os:windows`
	To show/hide whole cssp files for specific operating systems, simply add an os rule to your @cssp configuration element.

Loader plugin (`loader`)

Inserts the contents of another cssp file at the location of the @load declaration before parsig.

CSSP
`@load url(path/to/file.cssp)`