(D)ynamic (C)ascading (S)tyle (S)heets

Vorteile:
-Mehr überblick
-Farben Ändern/Anpassen durch Variablen
-PHP Code (außer Ausgabetext) in der CSS-Datei für Besucher nicht sichtbar
-Verschiedene Layouts eines Media-Typs in einer CSS Datei
-Kein PHP in der Ausgangsdatei nötig

Nachteile:
-Man braucht ein wenig PHP Kenntnisse (Nichts Unmögliches selbst für Anfänger)
-wenn man nicht aufpasst oder die DCSS Datei sehr groß wird kann es auch unübersichtlich werden

Ein einfaches Beispiel mit Variablen:
style-css.php

Quellcode

```
<?php
header('Content-type: text/css'); //Damit der Browser weis das gleich eine CSS Datei kommt obwohl wir uns in einer PHP Datei befinden
$hintergrund = "#000000"; //Variable $hintergrund und der Wert ist ein Hex-Code in diesen Fall Farbe Schwarz
$weiss = "#ffffff";
?>
body {
background-color: <?=$hintergrund?>;
}
#menu {
color: <?=$weiss?>;
font-weight: bold;
}
```

Alles anzeigen

Oder so verschiedene Hintergrund Farben für verschiedene Benutzer:
style-css.php

Quellcode

```
<?php
header('Content-type: text/css');
$hintergrund = "#000000";
//wir prüfen ob eine Farbe in der Session gesetzt wurde, wenn nicht bleibt sie schwartz
if(isset($_SESSION['color'])) { $hintergrund = $_SESSION['color'] }
$weiss = "#ffffff";
?>
body {
background-color: <?=$hintergrund?>;
}
#menu {
color: <?=$weiss?>;
font-weight: bold;
}
```

Alles anzeigen

Und die HTML Datei (so wie gewohnt):
index.html

Quellcode

```
1. <html>
2. <head>
3. <link rel="stylesheet" type="text/css" href="css/style.css.php" />
4. </head>
5. <body>
6. ...
7. </body>
8. </html>
```

eine andere Möglichkeit währe:
index.html

Quellcode

```
1. <html>
2. <head>
3. <style type="text/css">
4. <?php
5. /* Include the style sheet */
6. require_once("style-css.php");
7. ?>
8. </style>
9. </head>
10. <body>
11. ...
12. </body>
13. </html>
```

Alles anzeigen

Da der Browser gerne CSS Dateien im Cache behält und wir hier mit DCSS arbeiten, müssen wir ihm noch sagen das er dies sein lassen soll

Erstes Beispiel verbietet dem Browser die Seite im Cache zu laden. Also wird bei jedem Reload alles neu geladen:

Quellcode

```
1. <meta http-equiv="cache-control" content="no-cache" />
```

Zweites Beispiel dem Browser befehlen, wann er spätestens den Cache neu laden muss.

Quellcode

```
1. <meta http-equiv="expires" content="Sat, 01 Dec 2001 00:00:00 GMT">
```

Setzen wir als Zeit eine 0 hat das den gleichen Effekt wie bei dem ersten Beispiel

Quellcode

1. <meta http-equiv="expires" content="0">

Setzen wir statt 0 z.B. 30 wird der Cache neu geladen wenn der letzte Aufruf länger als 30 Sekunden her ist