

entwickler.press

Praxisbuch

WordPress- Themes

Daniel Koch



schnell + kompakt

Diese Informationen definiert man in der *style.css* im oberen Seitenbereich innerhalb eines mehrzeiligen CSS-Kommentars. Die *style.css* muss im Root-Verzeichnis des Themes liegen.

```
/*  
Theme Name: Entwickler  
Theme URI: http://entwickler-press.de/  
Description: Ein Entwickler-Press-Themes  
Author: Entwickler-Press  
Version: 1.0  
Tags: three-columns, fixed-width, custom-header  
*/
```

Wie detailliert Sie diesen Kommentarbereich anlegen, hängt letztendlich auch davon ab, ob Sie das Template der WordPress-Community zur Verfügung stellen wollen. Wenn Sie das vorhaben, sollten Sie die Angaben so wie im gezeigten Beispiel machen.

Daran anschließend können Sie die eigentlichen Stylesheet-Eigenschaften und deren Werte notieren.

Eingebunden wird die CSS-Datei mittels des Template-Tags `bloginfo('stylesheet_url')`.

```
<link href="<?php bloginfo('stylesheet_url'); ?>"  
rel="stylesheet" type="text/css" media="screen" />
```

Selbstverständlich können Sie auch zusätzliche CSS-Dateien einbinden und dabei beispielsweise auf die Conditional Comments des IE setzen.

```
<!--[if lte IE 7]>
<link href="<?php bloginfo('template_url');
?>/ie7.css" rel="stylesheet" type="text/css"
      media="screen" />
<![endif]-->
```

Alternativ dazu können die Stylesheets aber auch auf mehrere Dateien aufgeteilt werden. Um diese einzubinden können Sie über `bloginfo('template_url')` den URL des Templates ermitteln und als Pfadangabe verwenden.

Ebenso kann auch auf CSS-Syntax zum Einbinden von Stylesheets zurückgegriffen werden. Da es sich hierbei um CSS-Syntax handelt, müssen die Angaben auch tatsächlich innerhalb des `style`-Elements stehen.

```
@import url("switcher.css");
@import url("element.css");
@import url("basemod.css");
```

Innerhalb der Klammer von `url()` werden die einzubindenden CSS-Dateien angegeben.

Auch dabei können Angaben zum verwendeten Medientyp gemacht werden.

```
@import url("print.css") print;
```

Bei der Gestaltung des Themes durch CSS gilt dasselbe wie beim Anlegen des HTML-Grundgerüsts. Normalerweise sollte man immer mit einem sehr reduzierten Code beginnen und diesen dann – wenn das Layout stabil läuft – Stück für Stück ausbauen. Im aktuellen Beispiel wird die folgende CSS-Syntax verwendet:

```
* {
  margin:0;
  padding:0;
}
body {
  background:#f4f4f4 top center repeat-y;
  color:#303;
  font:76% Arial, Verdana, Tahoma,
    sans-serif;
}
#wrap {
  background-color:#fff;
  color:#333;
  margin:0 auto;
  width:760px;
}
#header {
  position:absolute;
  margin:0 0 0 40px;
  width:350px;
  left:50%;
  text-align:right;
}
#leftside {
  clear:left;
  float:left;
  line-height:1.5em;

  margin:0 0 5px;
  padding:0;
  width:140px;
}
#extras {
  float:right;
  line-height:1.4em;
  margin:0 0 5px;
```

```
padding:0;
width:140px;
}
#content {
  line-height:1.6em;
  margin:0 155px 5px;
  padding:0;
}
#footer {
  border-top:1px solid #e0e;
  font-size:1.1em;
  clear:both;
  color:#777;
  font-weight:normal;
  line-height:1.7em;
  margin:0 auto;
  padding:10px 0;
  text-align:center;
  width:760px;
}
```

Diese Stylesheet-Datei muss nun mit der eigentlichen Template-Datei verbunden werden. Normalerweise würde man dabei auf die folgende Syntax zurückgreifen:

```
...
<link rel="stylesheet" href="style.css"
  type="text/css" media="screen" />
<body>
<div id="wrapper">
...

```

In WordPress gibt es allerdings eine interessante Alternative: die Template-Tags. Diese werden detailliert im nächsten Abschnitt

vorgestellt, schließlich bilden sie die Grundlage für Ihre tägliche Arbeit mit WordPress-Themes.

Im aktuellen Beispiel soll die *style.css* über ein solches Template-Tag eingebunden werden. Verwendet wird dafür das Template-Tag `bloginfo()`. Um das Stylesheet einzubinden, wird dabei auf dessen möglichen Parameter `stylesheet_url` zurückgegriffen. Dadurch wird automatisch der URL des Stylesheets ermittelt.

```
...  
<link rel="stylesheet" href="  
<?php bloginfo('stylesheet_url'); ?>"  
  type="text/css" media="screen" />  
<body>  
<div id="wrapper">  
...
```

Das Theme kann nun bereits im Administrationsbereich unter *Designs/Themes* aktiviert werden.

WordPress erkennt das Theme tatsächlich bereits jetzt, auch wenn es selbst noch keine Funktionalitäten aufweist.