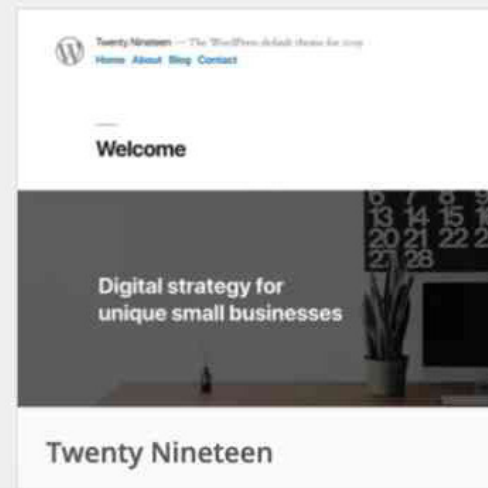
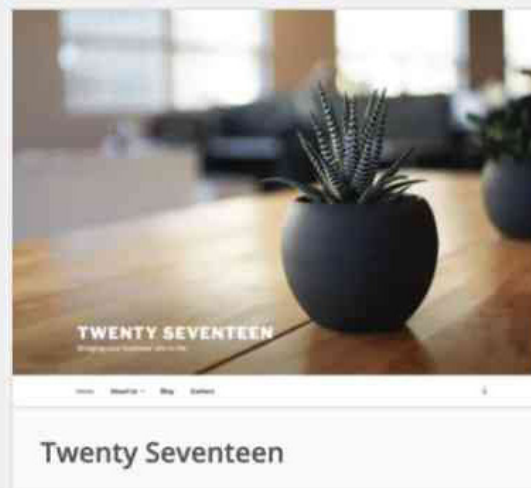
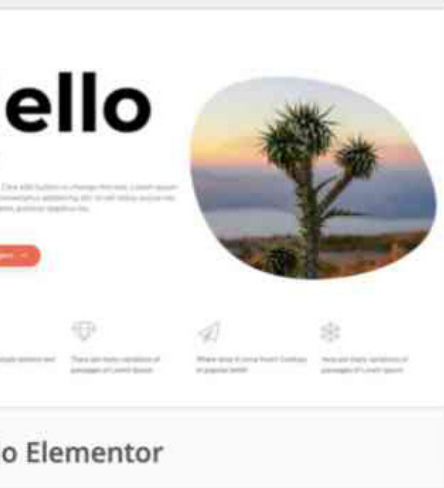
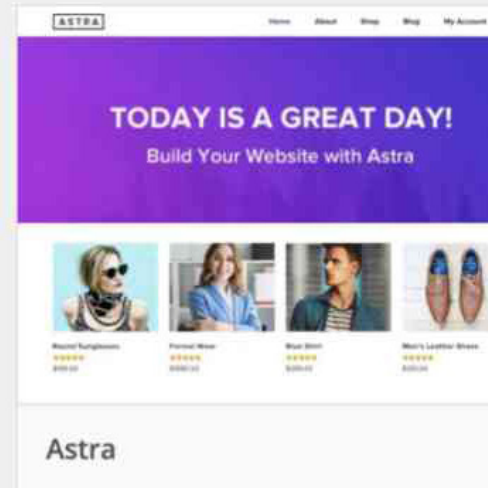
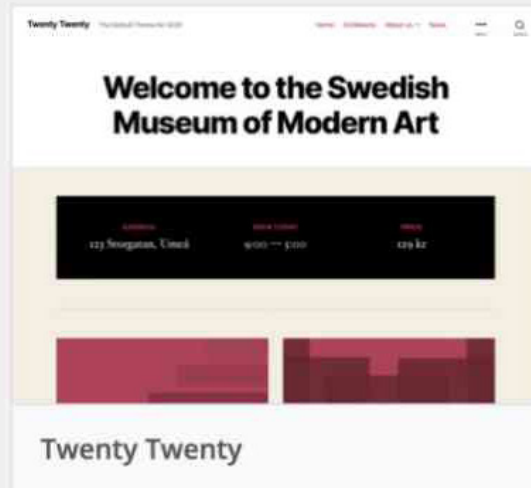




# WordPress-Theme erstellen



# 1. Schritt: Vorbereitung

Bevor gestartet werden kann, braucht es vorher noch ein paar Installationen, um eine Entwicklungsumgebung aufzugauen. Diese wären:

- [Code-Editor VS Code](#)
- Einen Webserver mit der Domain und einer Datenbank (jeweilige Benutzername und Passwort merken)
- [Wordpress-Installationsdatei](#)
- [FileZilla](#)

Die Programme Code-Editor und FileZilla sollen heruntergeladen und installiert werden. Die heruntergeladene Datei von Wordpress soll entzippt werden. Als nächstes müssen folgende Punkte erledigt werden:

- einen Ordner auf dem Computer erstellen, in welchem das Theme abgespeichert wird. Diesem Ordner einen Namen ohne Umlaute oder Abstände geben.
- die entpackten Wordpress-Dateien in den erstellten Ordner verschieben (nur den Inhalt des entzippten Ordners)

Als nächstes wird der neue Ordner mit dem Webserver durch FileZilla verbunden. Es wird FileZilla geöffnet.



Hier werden folgende Daten gebraucht:

- Servername
- Benutzername
- Passwort
- Port

Diese Daten bekommt man, sobald eine Domain gekauft oder erstellt wurde. Am Beispiel vom Anbieter hosttech bekommt man ein PDF mit Zugangsdaten. Die Angaben für FileZilla sind unter FTP-Zugangsdaten vorhanden. Sobald diese Daten auf FileZilla eingegeben wurden, kann auf „Verbinden“ gedrückt werden.

FileZilla nimmt die Verbindung auf und es erscheint auf der rechten Hälfte des Programms eine Ordnerstruktur. Unter dem Ordner „web“ kann nun die entzippten Dateien von Wordpress hochladen durch Drag & Drop.

In einem nächsten Schritt wird nun die ganze Wordpress-Installation vorgenommen.

## 2. Schritt: Installation von Wordpress

Jetzt kann man auf die Webseite gehen. Es erscheint eine grosse Auswahl an verschiedenen Sprachen. Es kann die gewünschte Sprache ausgewählt werden und auf den Button Weiter gedrückt werden.

Jetzt muss die Datenbank mit Wordpress verbunden werden. Auf „Los geht's“ drücken. Auf der nächsten Seite erscheinen fünf auszufüllende Felder mit:

- Datenbank-Name
- Benutzername
- Passwort
- Datenbank-Host
- Tabellen-Präfix

Diese Daten können vom jeweiligen Server abgerufen werden und hier eingegeben werden. Beim Tabellen-Präfix kann etwas beliebiges eingegeben werden. Am Besten etwas, das nicht leicht zu hacken ist. Danach kann auf „Senden“ gedrückt werden. Wenn danach ‚Alles klar!‘ steht, hat alles funktioniert und die Installation kann durchgeführt werden.

Danach erscheint eine Willkommensseite. Hier kann der Titel der Webseite eingegeben werden. (Es spielt keine Rolle, was hier eingegeben wird.)

Der Benutzername und das Passwort ist für das einloggen in Wordpress für diese Seite. Diese beiden Sachen am Besten irgendwo notieren, damit sie nicht vergessen gehen. Danach kann WordPress installiert werden.

Es soll eine Seite erscheinen, bei der steht: „Installation erfolgreich!“ Darunter hat es einen Anmeldebutton. Dort drauf drücken und danach mit den notierten Daten anmelden.

Nach dem Anmelden erscheint die Startseite von Wordpress.

Unter Design → Themes sind normalerweise drei Standard-Themes vorinstalliert.

Es soll jetzt ein neues Theme erstellt werden.

### 3. Schritt: Theme-Aufbau

Als erstes wird ein Ordner auf dem Computer erstellt. Auch auf dem Server im FileZilla soll der gleiche Ordner unter `web / wp-content / themes /` erstellt werden. Auf die Rechtschreibung achten! Es ist sehr wichtig, dass diese beiden Ordner genau identisch heißen.

In diesem Ordner sieht man auch schon drei andere Ordner. Dies sind die schon vorinstallierten Themes, die vorher schon in Wordpress gesehen wurden.

Sobald man in FileZilla in dem them-Ordner drin ist, einen Rechtsklick machen und dem neuen Verzeichnis den gleichen Namen geben.

Als nächstes kann das Programm VS Code geöffnet werden. Unter `File → Open Folder` den lokal erstellten Ordner auswählen.

Danach wird in unter `File → New File...` erstellt, welches mit `index.php` benannt wird. Es braucht immer eine `index`-Datei, damit eine Webseite läuft. Zusätzlich wird noch einmal ein neues File erstellt. Dieses wird `style.css` benannt. Dieses ist für das Design zuständig.

Im File `index.php` kann nun folgender Code eingefügt werden:

```
<!DOCTYPE html>
<html lang="de-CH" dir="ltr">
<head>
  <meta charset="utf-8">
  <title>Das neue Theme</title>
</head>
<body>
<header>
  <h1>Das neue Theme</h1>
  <p>Slogan</p>
  <nav>
    <ul>
      <li>Das ist ein Navigationspunkt</li>
      <li>Das ebenfalls</li>
    </ul>
  </nav>
</header>
<main>
  <article>
    <p>Hier kommt der Hauptinhalt</p>
  </article>
</main>
<footer>
  <p>© Dein Name - Juni 2023</p>
</footer>
</body>
</html>
```

Als nächstes kann der folgende Code ins style.css kopiert werden.

```
/*  
Theme Name: Name des Themes  
Author: Dein Name  
Author URL: Deine URL  
Description: Beschreibung des Themes  
Version: 1.0  
*/
```

Diese Angaben werden nicht sichtbar sein auf der Webseite, aber sind hilfreich für den Benutzer, um welches Theme oder welche Version es sich handelt. Hier können eigene Angaben abgefüllt werden.

Jetzt wurde das erste Theme erstellt. Es kann noch nicht sehr viel, aber dies wird sich in den nächsten Schritten ändern.

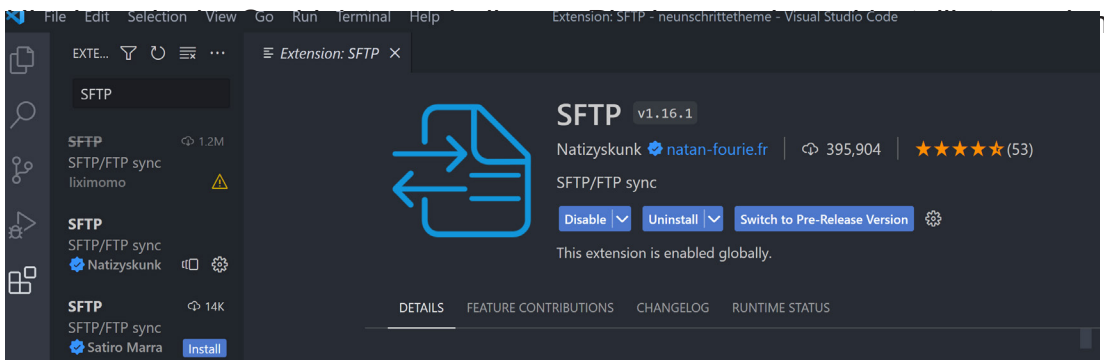
Im nächsten Schritt geht es darum, diese Dateien auf den Server laden zu können.

## 4. Schritt: Hochladen auf Server

Um die Dateien automatisch auf den Server zu laden, wird zuerst ein neues File erstellt und ein Plugin in VS Code heruntergeladen. Die neue Datei wird mit „.ftplib“ benannt. In dieses File wird folgender Code abgefüllt:

```
{
  „protocol“: „ftp“,
  „host“: „deine Domain“,
  „port“: 21,
  „user“: „dein user“,
  „pass“: „dein Passwort“,
  „promptForPass“: false,
  „remote“: „/“,
  „local“: „“,
  „secure“: false,
  „secureOptions“: null,
  „connTimeout“: 10000,
  „pasvTimeout“: 10000,
  „keepalive“: 10000,
  „watch“: [],
  „watchTimeout“: 500
}
```

Die grünen Texte werden durch die Zugangsdaten vom host abgefüllt. Um jetzt das Ganze auf den Server laden zu können, muss das Plugin installiert werden. Dazu geht man auf der Linken Seite auf dieses Symbol.



Sobald dies installiert ist, kann man wieder zu den Files gehen. Mit der Tastenkombination Ctrl + Shift + P wird in die Command Palette zugegriffen. Dort wird der Suchbegriff „SFTP“ eingegeben und den Befehl „SFTP: Sync Local → Remote“ gesucht und ausgewählt. Jetzt werden alle Dateien auf den Webserver geladen und sobald eine Datei neu gespeichert wird, passiert dies auch automatisch auf dem Server.

Um zu überprüfen, ob es funktioniert hat, kann auf dem SFTP-Server und auch auf der Webseite nachgeschaut werden.

## 5. Schritt: Die Dynamik

Als erstes wird der Titel geändert. Dieser wurde in Wordpress unter Einstellungen → Allgemein schon einmal vergeben. Doch im Code und auf der Webseite steht „Das neue Theme“. Im Code von index.php wird also der h1 folgend angepasst. Auch wird hier gleichzeitig der p-Tag mit dem aktuellen Slogan angepasst.:

Alt	Neu
<pre>&lt;header&gt;   &lt;h1&gt;Das neue Theme&lt;/h1&gt;   &lt;p&gt;Slogan&lt;/p&gt;   &lt;nav&gt;     &lt;ul&gt;       &lt;li&gt;Das ist ein Navigationspunkt&lt;/li&gt;       &lt;li&gt;Das ebenfalls&lt;/li&gt;     &lt;/ul&gt;   &lt;/nav&gt; &lt;/header&gt;</pre>	<pre>&lt;header&gt;   &lt;h1&gt;&lt;?php bloginfo(,name') ;?&gt;&lt;/h1&gt;   &lt;p&gt;Slogan&lt;/p&gt;   &lt;nav&gt;     &lt;ul&gt;       &lt;li&gt;Das ist ein Navigationspunkt&lt;/li&gt;       &lt;li&gt;Das ebenfalls&lt;/li&gt;     &lt;/ul&gt;   &lt;/nav&gt; &lt;/header&gt;</pre>

Nachdem die Datei gespeichert wurde, kann ein refresh auf der Webseite ausgeführt werden und der Titel wurde angepasst.

Dies wird auch für den title-Tag gemacht.

Alt	Neu
<pre>&lt;head&gt;   &lt;meta charset="utf-8"&gt;   &lt;title&gt;Das neue Theme&lt;/title&gt; &lt;/head&gt;</pre>	<pre>&lt;head&gt;   &lt;meta charset="utf-8"&gt;   &lt;title&gt;D&lt;?php bloginfo(,name') ;?&gt;&lt;&lt;/title&gt; &lt;/head&gt;</pre>

Als nächstes wird die Webseite noch ein wenig gestaltet. Doch zuerst muss das CSS mit der HTML-Datei verbunden werden. Die folgende Zeile kann direkt unter dem title-Tag eingefügt werden.

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

Nun ist das CSS mit dem HTML verknüpft. Jetzt kann im CSS unter dem schon vorhandenen Code folgendes eingegeben werden:



```
@import url(,https://fonts.goog-  
leapis.com/css2?family=Pop-  
pins:wght@300;400;500;600&display=swap');
```

```
/* --- Grundlagen */
```

```
* {  
  margin: 0 0;  
  box-sizing: border-box;  
  letter-spacing: 0.5px;  
  line-height: 1.4;  
  font-family: Poppins, sans-serif;  
}
```

```
body {  
  width: 100%;  
  background-color: white;  
  overflow-x: hidden;  
}
```

```
/* --- Header */
```

```
header h1 {  
  font-size: 3rem;  
  font-weight: 700;  
  text-align: center;  
}
```

```
header p,  
header nav {  
  font-size: 1.1rem;  
  font-weight: 300;  
  text-align: center;  
}
```

```
header {  
  background: linear-gradient(0deg, orange  
0%, darkorange 100%);  
  color: white;  
  padding-top: 40px;  
  width: 100%;  
}
```

```
header a {  
  color: white;  
  text-decoration: none;  
}
```

```
header a:hover {  
  opacity: 0.5;  
}
```

```
header nav {  
  margin-top: 40px;  
  background-color: darkorange;
```

```
padding: 40px 0;  
}  
header nav ul {  
  list-style-type: none;  
  padding-left: 0px;  
}  
header nav li {  
  display: inline;  
}  
header nav li:not(:last-of-type)::after {  
  content: „ | „;  
}
```

```
/* --- Main */
```

```
main {  
  width: 50vw;  
  padding: 100px 0;  
  margin: 0 auto;  
  min-height: 60vh;  
  color: #3a3a3a;  
}
```

```
main h2 {  
  font-size: 2.5rem;  
  padding-top: 20px;  
}
```

```
main h4 {  
  color: orange;  
}
```

```
main p {  
  font-size: 1.2rem;  
  line-height: 140%;  
  margin: 15px 0;  
}
```

```
main a {  
  font-size: 1.2rem;  
  line-height: 140%;  
  color: darkorange;  
  display: inline-block;  
}
```

```
main a:hover {  
  text-decoration: none;  
}
```

```
/* --- Footer */
```

```
footer {  
  padding: 40px 0;  
  text-align: center;  
  background-color: #3a3a3a;  
  color: white;  
}
```

Mit speichern und einem refresh auf der Webseite ist das CSS nun ersichtlich. Die Farben und alles kann auch nachträglich noch geändert werden. Damit man, wenn man auf Unterseiten ist, mit einem Klick auf das Logo (bei uns jetzt einfach der Seitentitel) immer zurück auf die Startseite kommt, muss man diesen dynamisch mit der Startseite verlinken. Der h1 wird somit ersetzt

Alt	Neu
<pre>&lt;header&gt;   &lt;h1&gt;&lt;?php bloginfo('name') ;?&gt;&lt;/h1&gt;   &lt;p&gt;Slogan&lt;/p&gt;   &lt;nav&gt;     &lt;ul&gt;       &lt;li&gt;Das ist ein Navigationspunkt&lt;/li&gt;       &lt;li&gt;Das ebenfalls&lt;/li&gt;     &lt;/ul&gt;   &lt;/nav&gt; &lt;/header&gt;</pre>	<pre>&lt;header&gt;   &lt;a href="&lt;?php echo home_url('/') ;?&gt;"&gt;   &lt;h1&gt;&lt;?php bloginfo('name') ;?&gt;&lt;/h1&gt; &lt;/a&gt;   &lt;p&gt;Slogan&lt;/p&gt;   &lt;nav&gt;     &lt;ul&gt;       &lt;li&gt;Das ist ein Navigationspunkt&lt;/li&gt;       &lt;li&gt;Das ebenfalls&lt;/li&gt;     &lt;/ul&gt;</pre>

Als nächstes werden unterstützende Tags eingefügt. Sie verändern auf den ersten Blick nicht viel, sind aber für die Funktionalität von Wordpress essenziell! Das Einzige, was nach diesen Schritten auf den ersten Blick als Veränderung erkennbar ist, ist die Bearbeitungsleiste, die angezeigt wird.

Alt	Neu
<pre>&lt;body&gt;</pre>	<pre>&lt;body &lt;?php body_class() ;?&gt;&gt;</pre>

Alt	Neu
<pre>&lt;head&gt;   &lt;meta charset="utf-8"&gt;   &lt;title&gt;&lt;?php bloginfo('name') ;?&gt;&lt;/title&gt; &lt;/head&gt;</pre>	<pre>&lt;head&gt;   &lt;meta charset="utf-8"&gt;   &lt;title&gt;&lt;?php bloginfo('name') ;?&gt;&lt;/title&gt;   &lt;?php wp_head() ;?&gt; &lt;/head&gt;</pre>

Alt	Neu
<pre>&lt;footer&gt;   &lt;p&gt;© Dein Name - Juni 2023&lt;/p&gt; &lt;/footer&gt; &lt;/body&gt; &lt;/html&gt;</pre>	<pre>&lt;footer&gt;   &lt;p&gt;© Dein Name - Juni 2023&lt;/p&gt; &lt;/footer&gt;   &lt;?php wp_footer() ;?&gt; &lt;/body&gt; &lt;/html&gt;</pre>

## 6. Schritt: Aufsplitten

Eine Webseite ist ja meistens so aufgebaut, dass sie auf der ganzen Seite immer denselben Header und denselben Footer hat. Genau dies wird jetzt gemacht, indem das Theme aufgesplittet wird.

In VS Code werden nun neue Dateien namens header.php und footer.php erstellt.

Jetzt werden im index.php alle Inhalte von zuoberst bis und mit zum öffnenden main-Tag ausgeschnitten und werden in das header.php eingefügt.

Danach wird alles ab dem schliessenden main-Tag bis zum Schluss ausgeschnitten und in die footer.php-Datei eingefügt. Nachdem werden alle Dateien mal gespeichert.

In der index.php Datei werden nun folgende Schnipsel eingefügt.

Dieses hier wird zuoberst eingefügt:

```
<?php get_header(); ?>
```

Dieses hier wird zuunterst eingefügt:

```
<?php get_footer(); ?>
```

Wenn nun die Seite aktualisiert wird, sollte sich nichts verändert haben,

## 7. Schritt: Die Navigation

In diesem Schritt wird eine Navigation eingefügt, welche nachher im Backend frei anpassbar ist. Im Moment ist dies noch nicht möglich, da unter den Einstellungen im Design von Wordpress das Menü fehlt.

Zuerst wird eine neue Datei erstellt, damit die Navigation im Backend registriert ist. Diese wird functions.php benannt. In diese Datei wird folgender Code eingefügt:

```
<?php
// --- navigation registrieren
add_action('after_setup_theme', 'navigation_registrieren');
function navigation_registrieren(){
    register_nav_menu('hauptnavigation', 'Hauptnavigation oben');
};
?>
```

Im Backend erscheint nun unter Design der Menü-Punkt. Auf diesen klicken.

Hier hat es momentan noch nicht so viele Navigationspunkte drin. Den Menüpunkt Beispiel-Seite kann unter der Menü-Struktur entfernt werden.

Es wird ein neuer Menüpunkt erstellt, indem man einen z.B. individuellen Link anlegt und den Linktext abfüllt. Danach wird auf den Button „Zum Menü hinzufügen“ geklickt.

Diesem Menü kann unter Name des Menüs auch noch einen Namen gegeben werden, nachdem alle gewünschten Navigationspunkte erfasst wurden.

Als nächstes klickt man auf den Button Menü erstellen.

Jetzt kann man auch auswählen, wo dieses Menü positioniert werden soll. Hauptnavigation oben auswählen.

Danach wird im Backend unter header.php noch ein kleines Codeschnipsel eingefügt.

Alt	Neu
<pre>&lt;nav&gt; &lt;ul&gt;   &lt;li&gt;Das ist ein Navigationspunkt&lt;/li&gt;   &lt;li&gt;Das ebenfalls&lt;/li&gt; &lt;/ul&gt; &lt;/nav&gt;</pre>	<pre>&lt;nav&gt; &lt;?php wp_nav_menu(array('theme_location' =&gt; 'hauptnavigation')) ;?&gt; &lt;/nav&gt;</pre>

Jetzt wird die Navigation nach einem refresh auf der Webseite angezeigt.

## 8. Schritt: Der Loop

Bevor mit dem achten Schritt gestartet werden kann, müssen ein paar Testinhalte erstellt werden. Am Besten werden ein paar neue Seiten und Beiträge erstellt. Es können auch fürs erste noch ein paar Beispielseiten erstellt werden.

Jetzt wird unter index.php ein Codeschnipsel zwischen den article-Tags eingefügt, welches ein sogenannter Loop ist. Der p-Tag wird gelöscht.

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
  <h2><?php the_title() ?></h2>
  <p><?php the_content() ?></p>
<?php endwhile; else : ?>
  <p>Es können keine Posts geladen werden.</p>
<?php endif; ?>
```

Im Wordpress unter Design → Menüs werden die neu erstellten Seiten ausgewählt und zum Hauptmenü hinzugefügt. Das Ganze nicht vergessen zu speichern.

Danach wird die Startseite aktualisiert und es soll in der Navigation die neuen Navigationspunkte erscheinen und auch die Beiträge, welche erstellt wurden.

## 9. Schritt: Eigene Ansichten für die Blogposts

Momentan werden alle Blog-Posts auf der Startseite angezeigt. Dies ist nicht unbedingt sehr schön und deswegen werden im Backend drei weitere Files erstellt.:

- page.php
- single.php
- front-page.php

In alle diese Seiten wird der Code von index.php reinkopiert. Jedoch wird auf der zweiten Linie eine Markierung eingefügt:

```
<mark>front-page.php</mark>
```

Für page.php und single.php wird der entsprechende Name zwischen die mark-Klammern geschrieben.

Als nächstes wird unter front-page.php zwischen den p-Tags folgendes Schnipsel eingefügt:

```
<?php the_excerpt() ?>
```

Auch noch auf front-page.php wird nach dem p-Tag dieses Codeschnipsel eingefügt:

```
<a href="<?php the_permalink() ?>">mehr lesen</a>
```

Als nächstes wird bei single.php folgendes oberhalb vom h2-Tag eingefügt:

```
<a href="<?php the_permalink() ?>">mehr lesen</a>
```

Am Schluss werden die am Anfang erstellten mark-Tags gelöscht, da diese nur zur Übersicht dienen.

Jetzt kann alles gespeichert werden und auf der Webseite begutachtet werden.

## 10. Schritt: Eigene Beiträge erstellen

Jetzt ist es geschafft und ein eigenes Wordpress-Theme wurde erstellt. Dies kann nun beliebig abgefüllt und mit den eigenen Beiträgen angepasst werden. Auch im Code können noch einige Sachen bearbeitet werden, wie z. B. die Farben. Doch fürs erste wurde ein Wordpress-Theme erstellt.

Herzlichen Glückwunsch!