

Cómo utilizar Markdown para documentación rápida

Creada por [John Gruber](#), Markdown, es una sintaxis de formateo que facilita muchísimo la creación de elementos básicos de HTML.

En lugar de utilizar etiquetas HTML (lo cual, en verdad, toma mucho tiempo para escribir), el trabajo de Markdown es quitarte del camino el proceso laborioso, permitiéndote enfocarte en el contenido. Debido a que la sintaxis es muy básica, es fácil tanto *escribir* como *leer* Markdown. Más adelante en éste tutorial, te enseñaremos los pasos para crear la documentación para utilizar Markdown, así que ¡te darás cuenta lo ligero y rápido que es!

Convertir a Markdown

Una vez que te hayas familiarizado a escribir en Markdown, necesitarás alguna clase de aplicación para realizar ejercicios de análisis sintáctico para convertir tu Markdown en código HTML.

El convertidor original Markdown fue creado en Perl, pero desde entonces muchas aplicaciones (a través de múltiples plataformas) han surgido. Veamos algunas de las opciones, tanto en línea como que pueden ejecutarse en tu sistema.

Dingus (en línea)

[Dingus](#) es una aplicación web creada por las mismas personas que crearon Markdown. Simplemente copia y pega tu código Markdown desde un editor de texto (o ingrésalo en el área de texto) y haciendo clic en un botón tu código HTML(así como una vista previa) aparecerá. Nada elegante, pero una forma simple de convertir Markdown a HTML.



MarkdownPad (Windows)

[MarkdwonPad](#) es una gran aplicación Windows que te permite crear fácilmente archivos Markdown (¡y es gratuita!). Incluye asombrosas características como una vista previa instantánea de HTML, formateo fácil con atajos de teclado, personalización de CSS, exportar HTML e incluso un modo de “cero distracciones”.



MarkdownPad is a full-featured [Markdown](#) editor for Windows.

Mou (OS X)

[Mou](#) es una fabulosa aplicación gratuita para la plataforma Mac, que te permite escribir Markdown de manera sencilla y agradable. Tiene grandes prestaciones como aplicación de estilo personalizado, atajos de teclado, actualización instantánea del resultado en HTML, exportar (con aplicación de estilo CSS opcional), exportar a PDF, soporte para dictado y más. Para éste tutorial utilizamos Mou para crear nuestra documentación del tema.



MarkdownNote (OS X & iOS)

[MarkdownNote](#) es otra maravillosa aplicación para escribir Markdown, y la aplicación para iOS es perfecta en caso de que quieras escribir Markdown en cualquier sitio. Al igual que otras aplicaciones, hemos abordado algunas estupendas funcionalidades, incluyendo vista previa de HTML, atajos de teclado y sincronización con Dropbox.



MarkdownNote

Markdown editor for Mac and iOS

Crear la Documentación del Tema

Hasta ahora, hemos cubierto lo que es Markdown y un vistazo a algunas herramientas que puedes utilizar para escribir Markdown. Ahora, vamos a crear la documentación del tema para un tema no existente, cubriendo lo que típicamente debes incluir en la documentación, la sintaxis Markdown y las mejores prácticas.

Sintaxis de Markdown

Durante los siguientes pasos estaremos mirando cómo se aplica Markdown a nuestras necesidades. Para una mirada más detallada sobre la sintaxis de Markdown, revisa nuestra sección [“Los Pormenores de Markdown en Nettuts+”](#).

¿Qué debe incluir un archivo de Documentación?

Porque *tú* ya sabes lo que se puede y lo que no se puede hacer en nuestro tema/plantilla, aplicación web, podría parecer un poco tedioso cubrir los fundamentos. Es costumbre que haya instalación, personalización, y algunos cambios involucrados sobre lo que deben saber los usuarios-y es tu trabajo orientarlos. Esto es lo que deseábamos incluir:

- Información
 - Autor
 - Fecha de Creación
 - Versión
 - Donde Contactarnos
- Estructura del Archivo
 - HTML
 - CSS
 - Javascript
- Estructura Individual del Archivo
 - Estructura de HTML
 - Estructura de CSS
- Personalizar Estilos
- Modificar Archivos
 - Modificar CSS
 - Modificar Javascript
- Compatibilidad de Navegadores
- Archivo de Registro de Cambios

Recuerda, la documentación debería ser lo suficientemente simple para que cualquier persona pudiera entenderla, pero también abarcar como cambiar archivos importantes. Por ejemplo, no necesariamente tienes que mostrar cómo cambiar valores específicos de CSS, pero si deberías mostrar como acceder a esos archivos.

Paso 1: Hacer el archivo de Markdown

¡Ahora comienza la diversión! Abre tu editor de Markdown (usaré Mou para Mac). Crea un encabezado para tu archivo de documentación:

```
# Documentación del Tema
```

Los encabezados pueden escribirse simplemente al añadir de uno a seis numerales # antes del contenido. En el ejemplo de arriba, utilizamos un # para crear un elemento h1 con el texto “Documentación del Tema”. Si quieres crear un elemento h3, escribirías ###Encabezado3.

Ahora, hagamos una línea horizontal (<hr />) para separar el título del otro contenido. De nuevo, en Markdown esto es muy sencillo:

```
***
```

Paso 2: Información del Tema

Una sección importante para añadir dentro de tu documentación es una sección de información.

Nombre del Tema: Tema

Autor: Suhail Dawood

* Creado:* 08/08/2012

* Versión:* 1.01

¡Gracias por su compra! Si tiene alguna pregunta sobre éste tema, no dude en mandarme un correo electrónico a **email@tutsplus.com** o enviarme un tuit a **@tutsplus**

Veamos el código HTML equivalente del código Markdown de arriba:

```
<em>Nombre del Tema: </em>Tema<br />
```

```
<em>Autor: </em>Suhail Dawood<br />
```

```
<em>Creado: </em>08/08/2012<br />
```

```
<em>Versión: </em>1.0.1<br />
```

```
<hr />
```

¡Gracias por su compra! Si tiene alguna pregunta sobre éste tema, no dude en mandarme un correo electrónico a `email@tutsplus.com` enviarme un tuit `@tutsplus`

Sólo con ver las dos fuentes diferentes, puedes notar instantáneamente que Markdown es mucho más intuitivo de entender y crear. En lugar de tener que añadir constantemente signos de `<`, `y`, `>`, Markdown te permite enfocarte en el contenido. Para resaltar, añade un asterisco antes y después del texto (ejemplo `*Texto resaltado*`). Para poner en negritas, añade *dos* asteriscos antes y después del texto (ejemplo `**Texto en Negritas**`). Para agregar un salto de línea, simplemente añade dos espacios en el punto en que quieres insertar un salto de línea.

Paso 3: Añadir un Índice

Ahora agreguemos un índice a nuestro archivo Markdown:

##Índice

1. Estructura HTML
2. Archivos CSS
3. Archivos Javascript
4. Archivos PSD
5. Estilos de Tema
6. Modificar Javascript
7. Modificar CSS
8. Compatibilidad con Navegadores

Si fuéramos a crear esto en HTML, aparecería así:

```
<h2>Índice</h2>
```

```
<ol>
```

```
<li>Estructura HTML</li>
<li>Archivos CSS</li>
<li>Archivos Javascript</li>
<li>Archivos PSD</li>
<li>Estilos de Tema</li>
<li>Modificar Javascript</li>
<li>Modificar CSS</li>
<li>Compatibilidad con Navegadores</li>
</ol>
<hr />
```

En lugar de tener que crear una lista ordenada y separar elementos de la lista, simplemente escribimos `1. Primer Elemento` y continuamos añadiendo los elementos incrementándose en un número.

Paso 4: Estructura HTML

Es importante permitirles a tus usuarios saber cómo están maquetados los archivos del sitio. Ya que estamos creando documentación para un tema, deberíamos definir como se estructura el código HTML del tema. Podemos definir esto con el siguiente código:

```
##1. Estructura HTML
```

La Estructura HTML para cada página es como sigue:

```
* Meta
* Enlace a Archivos
* Encabezado
  *Logo
  *Enlace a Redes Sociales
*Navegación
*Contenido Principal
  *Carrusel de Contenido
  *Artículos
*Barra Lateral
  *Búsqueda
```

```
*Archivo de Post Anteriores
```

```
*Lista de Correo
```

```
* Enlace a Archivos Javascript
```

```
* Javascript
```

```
***
```

Fácil. Para crear nuestro índice, utilizamos una lista ordenada. En esta sección, usamos listas desordenadas *anidadas*. Para crear una lista ordenada en Markdown, simplemente añadimos un asterisco y un espacio antes del texto (ejemplo: `* Elemento 1`). Para anidar objetos en la lista, sólo tecleamos la tabulación y creamos otro objeto de lista.

Paso 5: Archivos CSS

Particularmente en temas, la documentación CSS es muy importante. Es buena idea definir los diferentes archivos que se incluyen en el tema, así como una breve descripción de cada archivo:

```
## 2. Archivos CSS
```

```
Hay 3 archivos CSS en este tema:
```

```
* principal.css
```

```
* colores.css
```

```
* skeleton.css
```

```
#### principal.css
```

```
Este archivo CSS es la hoja de estilos principal para el tema. Contiene todos los valores para los diferentes elementos del tema y el esquema de colores por defecto.
```

```
#### colores.css
```

```
Este archivo CSS contiene los estilos de todos los otros esquemas de colores que están incluidos en el tema.
```

```
#### skeleton.css
```

```
Este archivo CSS permite que el tema sea adaptable, ajustándose diferentes tamaños de pantalla.
```

```
***
```


Asegúrate de usar encabezados para separar diferentes secciones del archivo de documentación. Una buena maquetación del archivo de documentación provocará menos confusiones entre los usuarios.

Paso 6: Archivos Javascript

Si tu tema incluye archivos Javascript, es buena idea al menos especificarlos en la documentación:

```
## 3. Archivos Javascript
```

```
Hay 2 archivos Javascript en este tema:
```

```
* jquery.js
```

```
* slider.js
```

```
#### jquery.js
```

```
Este tema usa la biblioteca JQuery de Javascript.>
```

```
#### slider.js
```

```
Este tema usa la biblioteca JQuery de Javascript.>
```

```
El carrusel de contenido en el tema se ejecuta en este archivo Javascript. Hay  
varias configuraciones que pueden ser modificadas, esto se cubrirá en la sección  
"Modificar Javascript".
```

Asegurate de no solamente listar, sino describir los archivos en tu tema. Esto facilitará al usuario entender el contenido de cada archivo, y decida si quiere o no involucrarse en ellos.

Paso 7: Archivos PSD

Aunque hay una sección aparte para las plantillas PSD en [ThemeForest](https://www.themeforest.net), muchos autores deciden incluir archivos PSD con sus plantillas ya con código para permitirle al usuario realizar cambios libremente en el diseño. Si tu tema tiene archivos PSD incluidos, podría ser algo muy atinado especificarlos tan pronto como terminemos con los demás archivos:

```
##4. Archivos PSD
```

```
Incluidos en este tema hay 5 diferentes archivos PSD:
```

1. inicio.psd

2. acerca.psd

3. portafolio.psd

4. blog.psd

5. contacto.psd

Estos archivos PSD serán útiles si quisieras hacer cambios en el diseño del tema.

También puedes crear una nueva maquetación de página usando los archivos PSD

existentes como una base con la cual trabajar.

Lo mejor es que nombres tus archivos PSD claramente (ejemplo: página-ancho-total.psd) en lugar de nombres abstractos (ejemplo: plantilla-003.psd). De ésta forma, los usuarios pueden encontrar lo que necesitan sin tener que abrir cada archivo.

Paso 8: Estilos de Tema

Muy probablemente, tu tema incluirá una selección de esquemas de colores disponibles para los usuarios. Si éste es el caso, asegúrate de que especifiques como se hace esto:

##5. Estilos de Tema

Incluidos en este tema hay 10 diferentes estilos de tema:

1. Claro

2. Oscuro

3. Gris

4. Verde

5. Rojo

6. Naranja

7. Azul

8. Púrpura

9. Café

10. Azul Oscuro

Para cambiar estos estilos, dirígete al tablero de WordPress, selecciona **Opciones**

> **Estilos** y selecciona el estilo que prefieras.

En el ejemplo de arriba, he enlistado tres diferentes esquemas de colores incluidos en el tema y mostré como cambiarlos.

Paso 9: Modificar Javascript

Si tu código incluye archivos Javascript que tienen parámetros para personalizarlos (ejemplo: un script para el carrusel), una buena idea sería mostrar a tus usuarios como manipular estos valores:

```
##6. Modificar Javascript
```

El carrusel de contenido en el tema se ejecuta fuera de slider.js y hay un par de valores que pueden cambiarse para alterar el aspecto y hacer más notable el carrusel.

```
##### Cambiar valores
```

En slider.js, puedes alterar estos valores:

```
<code>auto: true</code>
```

Boolean: Animar Automáticamente, true o false

```
<code>speed: 1000</code>
```

Integer: Velocidad de la transición, en milisegundos

```
<code>pager: true</code>
```

Boolean: mostrar paginación, true o false

```
<code>nav: false</code>
```

Boolean: Mostrar navegación, true o false

```
***
```

El código de arriba define como un usuario puede cambiar valores. Para aplicar estilo al código, encierra el texto relevante dentro de etiquetas `<code>`. Aplicaciones como Mou agregan estilo a estos elementos en la vista previa instantánea, y también puedes exportar el código para mantener el estilo. Hablaremos sobre exportar tu documentación más adelante.

Paso 10: Modificar CSS

Los archivos CSS son frecuentemente personalizados por el usuario. Seguramente considerarán útil si agregas una sección a la documentación que muestre como acceder a los archivos CSS para poder editarlos.

##7. Modificar CSS

El tema es construido en un framework (esquema para el desarrollo y/o implementación de una aplicación) adaptable, que permite ver óptimamente el contenido en pantallas de todo tamaño.

####Cambiar el CSS

Empezar por el tablero de WordPress, ****Temas > Tema > Apariencia****. Selecciona el archivo `*estilos.css*` para ver y poder editar el código.

Aquí puedes cambiar cosas como las fuentes, tamaños, colores, etc.

Ahora que el usuario tiene acceso al archivo CSS, puede realizar los cambios que quiera.

Paso 11: Compatibilidad con Navegadores

Es buena idea notificar al usuario de cualquier problema de compatibilidad con navegadores:

##8. Compatibilidad con Navegadores

Este tema trabaja bien (-) o magníficamente (X) en los siguientes navegadores:

****IE 6-8**** -

****IE 9+**** X

****Chrome**** X

****Firefox**** X

****Safari**** X

****Opera**** X

Si quisieras expandirte en ésta sección, podrías explicar que características del tema tienen problemas en ciertos navegadores.

Paso 12: Terminar la Documentación

Para completar nuestra documentación, agregaremos una sección para que nuestros usuarios puedan contactarnos si tienen dudas. Añadamos este código:

```
####Tema de Suhail Dawood

Si tienes más preguntas que no están cubiertas en la documentación, no dudes en
mandarme un correo electrónico a <email@tutsplus.com>, o agreguemos un nuevo post en
el [foro] (http://forum.tutsplus.com/"visita el foro").
```

Para enlazar correos electrónicos utilizando Markdown, sólo encierra tu dirección de correo electrónico entre los signos de menor que y mayor que (ejemplo: <email@tutsplus.com>).

Agregar enlaces en Markdown es muy simple. `` Tuts+ es el hipervínculo que quieras que aparezca en el HTML. El texto debería ser colocado dentro de corchetes `[Tuts+]`. Inmediatamente después, escribimos la dirección del sitio entre paréntesis (ejemplo: `(http://www.tutsplus.com/)`) y el texto alternativo `alt` es colocado entre comillas dobles dentro del paréntesis (ejemplo: `(http://www.tutsplus.com/ "La red líder en el mundo en tutoriales")`).

Código Final en Markdown

Aquí está el aspecto final de nuestro código Markdown para ésta documentación:

```
#Documentación del Tema

***

*Nombre del Tema:* Tema
*Autor:* Suhail Dawood
*Creado:* 08/06/2012
*Versión:* 1.01

***

;Gracias por tu compra! Si tienes preguntas sobre éste tema, no dudes en mandarme un
correo electrónico a email@tutsplus.com o enviarme un tuit a @tutsplus

***
```

##Índice

1. Estructura HTML
2. Archivos CSS
3. Archivos Javascript
4. Archivos PSD
5. Estilos de Tema
6. Modificar Javascript
7. Modificar CSS
8. Compatibilidad con Navegadores

##1. Estructura HTML

La Estructura HTML para cada página es como sigue:

* Meta

* Enlace a Archivos

* Encabezado

*Logo

*Enlace a Redes Sociales

*Navegación

*Contenido Principal

*Carrusel de Contenido

*Artículos

*Barra Lateral

*Búsqueda

*Archivo de Post Anteriores

*Lista de Correo

* Enlace a Archivos Javascript

* Javascript

2. Archivos CSS

Hay 3 archivos CSS en este tema:

* principal.css

* colores.css

* skeleton.css

principal.css

Este archivo CSS es la hoja de estilos principal para el tema. Contiene todos los valores para los diferentes elementos del tema y el esquema de colores por defecto.

colores.css

Este archivo CSS contiene los estilos de todos los otros esquemas de colores que están incluidos en el tema.

skeleton.css

Este archivo CSS permite que el tema sea adaptable, ajustándose diferentes tamaños de pantalla.

3. Archivos Javascript

Hay 2 archivos Javascript en este tema:

* jquery.js

* slider.js

jquery.js

Este tema usa la biblioteca JQuery de Javascript.>

slider.js

Este tema usa la biblioteca JQuery de Javascript.>

El carrusel de contenido en el tema se ejecuta en este archivo Javascript. Hay varias configuraciones que pueden ser modificadas, esto se cubrirá en la sección "Modificar Javascript".

4. Archivos PSD

Incluidos en este tema hay 5 diferentes archivos PSD:

1. inicio.psd

2. acerca.psd

3. portafolio.psd

4. blog.psd

5. contacto.psd

Estos archivos PSD serán útiles si quisieras hacer cambios en el diseño del tema.

También puedes crear una nueva maquetación de página usando los archivos PSD

existentes como una base con la cual trabajar.

5. Estilos de Tema

Incluidos en este tema hay 10 diferentes estilos de tema:

1. Claro

2. Oscuro

3. Gris

4. Verde

5. Rojo

6. Naranja

7. Azul

8. Púrpura

9. Café

10. Azul Oscuro

Para cambiar estos estilos, dirígete al tablero de WordPress, selecciona **Opciones** > **Estilos** y selecciona el estilo que prefieras.

##6. Modificar Javascript

El carrusel de contenido en el tema se ejecuta fuera de slider.js y hay un par de valores que pueden cambiarse para alterar el aspecto y hacer más notable el carrusel.

Cambiar valores

En slider.js, puedes alterar estos valores:

```
<code>auto: true</code>
```

```
<code>\*Boolean: Animar Automaticamente, true o false*</code>
```

```
<code>speed: 1000</code>
```

Integer: Velocidad de la transición, en milisegundos

```
<code>pager: true</code>
```

Boolean: mostrar paginación, true o false

```
<code>nav: false</code>
```


Boolean: Mostrar navegación, true o false

##7. Modificar CSS

El tema es construido en un framework (esquema para el desarrollo y/o implementación de una aplicación) adaptable, que permite ver óptimamente el contenido en pantallas de todo tamaño.

####Cambiar el CSS

Empezar por el tablero de WordPress, ****Temas > Tema > Apariencia****. Selecciona el archivo `*estilos.css*` para ver y poder editar el código.

Aquí puedes cambiar cosas como las fuentes, tamaños, colores, etc.

##8. Compatibilidad con Navegadores

Este tema trabaja bien (-) o magníficamente (X) en los siguientes navegadores:

****IE 6-8**** -

****IE 9+**** X

****Chrome**** X

****Firefox**** X

****Safari**** X

****Opera**** X

####Tema de Suhail Dawood

Si tienes más preguntas que no están cubiertas en la documentación, no dudes en mandarme un correo electrónico a [<email@tutsplus.com>](mailto:email@tutsplus.com), o agreguemos un nuevo post en el [foro] (<http://forum.tutsplus.com/>"visita el foro").

Exportar la documentación

Ahora que hemos terminado de crear el archivo de documentación, es tiempo de convertir el archivo Markdown a archivo HTML ó PDF. Usaré Mou para exportar mi documentación, pero los pasos son habituales y se aplican a la mayoría de las aplicaciones.

8. Purple
9. Brown
10. Dark Blue

To change these styles, go to the WordPress backend, select **Options > Styles** and select the style you would like.

##6. Tweaking Javascript

The content slider in the theme runs off of slider.js, and there are a couple of values that can be changed to alter the look and feel of the slider.

#####Changing Values

In slider.js, you can alter these values:

```
<code>auto: true</code>
```

Boolean: Animate automatically, true or false

```
<code>speed: 1000</code>
```

Integer: Speed of the transition, in milliseconds

```
<code>pager: true</code>
```

Boolean: Show pager, true or false

8. Purple
9. Brown
10. Dark Blue

To change these styles, go to the WordPress backend, select **Options > Styles** and select the style you would like.

6. Tweaking Javascript

The content slider in the theme runs off of slider.js, and there are a couple of values that can be changed to alter the look and feel of the slider.

Changing Values

In slider.js, you can alter these values:

```
auto: true
```

Boolean: Animate automatically, true or false

```
speed: 1000
```

Integer: Speed of the transition, in milliseconds

```
pager: true
```

Boolean: Show pager, true or false

```
nav: false
```

Boolean: Show navigation, true or false

Exportar a HTML

Para exportar tu documentación a HTML, simplemente selecciona **Archivo>Exportar>HTML**. Nombra tu archivo de documentación y marca/desmarca ‘Incluir CSS’ dependiendo si quieres o no que el mismo estilo mostrado en la aplicación se aplique a tu archivo HTML. Mou no creará un archivo CSS separado, en cambio agrega el estilo al archivo HTML. Mou también crea las etiquetas necesarias tales como `html, doctype, head, etc.`

Exportar a PDF

Si quisieras exportar tu documentación a PDF, selecciona **Archivo>Exportar>PDF**. En el caso de Mou, todos los estilos mostrados en la vista previa instantánea de HTML, se incluirán en el archivo PDF.

¡Y terminamos! Perfectamente claro, documentación legible para los clientes, y colegas. [Descarga los archivos fuente para este tutorial](#) y revisa los archivos resultantes .md .html y .pdf.

Recapitulación de la Sintaxis Markdown

Aquí una rápida comparación de la sintaxis HTML y Markdown

<code><h1>Encabezado Uno</h1></code>	<code>#Encabezado Uno</code>
<code><h2>Encabezado Dos</h2></code>	<code>##Encabezado Dos</code>
<code><h3>Encabezado Tres</h3></code>	<code>###Encabezado Tres</code>
<code><h4>Encabezado Cuatro</h4></code>	<code>####Encabezado Cuatro</code>
<code><h5>Encabezado Cinco</h5></code>	<code>#####Encabezado Cinco</code>
<code><h6>Encabezado Seis</h6></code>	<code>#####Encabezado Seis</code>
<code><hr /></code>	<code>***</code>
<code>Texto Resaltado</code></code>	<code>*Texto Resaltado*</code>
<code>Texto en Negritas</code>	<code>**Texto en Negritas**</code>
<code>Elemento de Lista</code>	<code>1. Elemento de Lista</code>
<code>Elemento de Lista</code>	<code>* Elemento de Lista</code>
<code><code>Código</code></code>	<code><code>Código</code></code>
<code> email@tutsplus.com</code>	<code><email@tutsplus.com></code>
<code>forum</code>	<code>[forum] (http://www.tutsplus.com/ "Descripción")</code>

Recursos Adicionales

- Hay muchas más fuentes para Markdown, incluyendo formas alternativas para crear otros elementos. Puedes revisar una detallada información para toda la sintaxis de Markdown en [Nettuts+](#).
- [El Proyecto Oficial de Markdown](#) en Daring Fireball
- [Markdown](#) en Wikipedia
- [El plugin esencial para Vim: de Markdown a HTML](#) en Nettuts+.
- Más información en documentación de Themeforest, [aquí](#), [aquí](#) y [aquí](#) (¡Gracias a Ivor por esos!)