



EBook Gratis

APRENDIZAJE Shopify

Free unaffiliated eBook created from
Stack Overflow contributors.

#shopify

Tabla de contenido

Acerca de	1
Capítulo 1: Empezando con Shopify	2
Observaciones.....	2
No hay versiones para Shopify	2
Examples.....	2
Instalación o configuración.....	2
Configuración de tema local.....	2
Instalador ruby.....	3
Kit de desarrollo.....	3
Certificado SSL.....	3
Shopify Theme Gem.....	3
Crear una aplicación privada	4
Crear un archivo de configuración	4
Shopify desarrollo local con Theme Kit.....	5
Instalar Shopify Theme Kit.....	5
Instalacion de windows.....	5
Obtener acceso a la API.....	6
Capítulo 2: Implementando Secciones	10
Introducción.....	10
Observaciones.....	10
Cosas para recordar:	10
Más recursos:	10
Examples.....	10
Incluyendo una sección en una página.....	10
En esquema de archivo.....	11
Uso de bloques para elementos recurrentes.....	12
Usando JavaScript y hojas de estilo en secciones.....	12
Creditos	14

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [shopify](#)

It is an unofficial and free Shopify ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Shopify.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con Shopify

Observaciones

Esta sección proporciona una descripción general de qué es Shopify y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de Shopify y vincular a los temas relacionados. Dado que la Documentación para Shopify es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

No hay versiones para Shopify

Shopify no publica números de versión ni cambia los registros, debido a problemas de seguridad. Para mantenerse al día con los cambios en Shopify, los mejores lugares para verificar serían:

- [Shopify Blog](#)
- [Shopify Forums](#)
- [Shopify Open Source Projects](#)

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar Shopify.

Configuración de tema local

Hay algunas formas de configurar una copia local de su tema de Shopify:

- Node.js (gulp / grunt)
- Rubí
- Aplicación de tema (solo Mac)

Puede haber algunas otras formas, pero encontré la manera más fácil de trabajar con el rubí.

Para trabajar localmente en los archivos de tema de Shopify, necesitará algunas cosas:

- Instalador Ruby (<http://rubyinstaller.org/downloads/>)
 - Kit de desarrollo (<http://rubyinstaller.org/downloads/>)
 - Certificado SSL - (<https://curl.haxx.se/ca/cacert.pem>)
 - Shopify Theme Gem - (https://rubygems.org/gems/shopify_theme/versions/0.0.25)
-

Instalador ruby

Lo primero es instalar el instalador de Ruby, usar la última versión.

El proceso de instalación es sencillo, pero **le pedirá que agregue Ruby a la ruta global del entorno de Windows y debe hacer clic en la casilla de verificación y agregarlo allí**, para que no tenga que hacerlo manualmente más tarde.

Kit de desarrollo

Debes elegir la versión correcta del DevKit que va con el instalador de Ruby que agregaste.

El proceso de instalación es fácil también.

1. Descomprime el archivo en C: \ DevKit por ejemplo
2. Usted cmd en esa carpeta y escribe `ruby dk.rb init` que debería crear un `config.yml` en el archivador
3. Si abre el archivo `config.yml` debería ver en la parte inferior - C:/Ruby23-x64 (dependiendo de dónde instale su Ruby y qué versión es) si no está allí, debe agregarlo manualmente
4. `ruby dk.rb install` en la misma carpeta, ejecuta la siguiente `ruby dk.rb install`
5. Si todo está bien, debería poder ejecutar el siguiente comando: `gem install json --platform=ruby` que confirmará que su instalación de Ruby está funcionando

Aquí también se pueden ver instrucciones más detalladas:

<https://github.com/oneclick/rubyinstaller/wiki/Development-Kit>

Certificado SSL

Deberá agregar el archivo **cacert.pem** a las variables globales de su ventana.

1. Debe descargar el siguiente archivo: <https://curl.haxx.se/ca/cacert.pem> (guárdelo como `.pem` y no como `.pem.txt`)
 2. Debe ingresar a la pantalla Variables de entorno y en Variables del sistema, haga clic en Nuevo
 3. Para el nombre de la variable, ingrese **SSL_CERT_FILE** y para el valor de la variable ingrese la ruta al archivo, por ejemplo, **C: \ RailsInstaller \ cacert.pem** .
-

Shopify Theme Gem

Abra su cmd e ingrese el siguiente código: `gem install shopify_theme`

Esto debería instalar la gema del tema de Shopify que necesitaremos.

Si repasaste todos los pasos anteriores una vez, no los seguirás haciendo.

Ahora vienen los pasos que deberá seguir cada vez que inicie un nuevo proyecto de Shopify:

- Crear una aplicación privada (desde la tienda Shopify)
- Crear un archivo de configuración utilizando la información de la aplicación privada

Crear una aplicación privada

Debe iniciar sesión en el panel de administración de su tienda Shopify y acceder a la pantalla de la aplicación -> Aplicación privada:

<https://YOUR-STORE-NAME.myshopify.com/admin/apps/private>

Y haga clic en el botón **Crear aplicación privada** .

Te pedirá un título. (ingrese lo que quiera)

Lo importante es cambiar todos los permisos de lectura y escritura.

Una vez que cambie los permisos y guarde la aplicación, lo redirigirá a una página desde la que obtendrá la clave de API y la contraseña para esta aplicación. (los necesitas para el archivo de configuración que crearemos a continuación)

Crear un archivo de configuración

Vaya a una carpeta local donde le gusten los archivos de tema y cmd dentro.

Escribe el siguiente comando:

```
theme configure API-KEY PASSWORD DOMAIN ID
```

Dónde:

- Clave API - es tomada de ellos la aplicación privada que creamos
- Contraseña - es tomada de ellos la aplicación privada que creamos
- Dominio: este es su dominio **SU-STANCIA-NOMBRE.myshopify.com** , sin https: // antes de él
- ID: este es el ID de tema, si va a trabajar en el tema en vivo, puede omitirlo, pero si está trabajando en una clonación, deberá ingresar el ID de tema aquí.

Una vez que ejecute el código anterior, debe crear un archivo llamado **config.yml** donde toda la información proporcionada estará presente.

Ahora debe cmd (en caso de que cierre la consola) en la carpeta donde se encuentra el **archivo config.yml** y escriba el siguiente código: `theme download` que debe comenzar a descargar los

archivos uno por uno.

Una vez que complete el proceso de descarga, escribirá la `theme watch` y, cada vez que modifique los archivos descargados, actualizará los archivos en la tienda Shopify después de 1-2 segundos. Esto es cierto si también agrega archivos, pero no cuando borra archivos.

Se puede encontrar información más detallada sobre la Gema aquí:

https://github.com/Shopify/shopify_theme

Shopify desarrollo local con Theme Kit

Los temas de Shopify también se pueden desarrollar localmente. Siga los pasos a continuación para configurar el entorno de desarrollo local:

- Instalar Shopify Theme Kit
- Obtener acceso a la API
- Use un tema nuevo o configure un tema existente con el kit de temas

Explicación de los pasos anteriores son los siguientes:

Instalar Shopify Theme Kit

Instalacion automatica

Si está en Mac o Linux, puede usar el siguiente script de instalación para descargar e instalar automáticamente el Kit de temas para usted.

```
curl -s https://raw.githubusercontent.com/Shopify/themekit/master/scripts/install | sudo
python
```

Homebrew

Si tiene instalado Homebrew, puede instalar Theme Kit ejecutando los siguientes comandos.

```
brew tap shopify/shopify
brew install themekit
```

Instalacion de windows

Instalación automática de PowerShell

Ejecute los siguientes comandos en Powershell como administrador.

```
(New-Object
System.Net.WebClient).DownloadString("https://raw.githubusercontent.com/Shopify/themekit/master/scripts
| powershell -command -
```

Manual

- Cree una carpeta dentro de C: \ Archivos de programa \ llamado Kit de temas
- Descargue el kit de temas (a continuación) y copie el programa extraído en C: \ Archivos de programa \ Kit de temas
- Vaya a Panel de control > Sistema y seguridad > Sistema. Otra forma de llegar es haciendo clic con el botón derecho en Mi PC y elegir el elemento de propiedades
- Busque el botón o enlace llamado Variables de entorno
- En el segundo panel, busque el elemento llamado Ruta y haga doble clic en él. Esto debería abrir una ventana con un campo de texto que está repleto de contenido.
- Mueva el cursor hasta el final y agregue lo siguiente: C: \ Archivos de programa \ Kit de temas \
- Haga clic en Aceptar hasta que todas las ventanas se hayan ido.
- Para verificar que Theme Kit se haya instalado, abra cmd.exe y escriba theme.

Instalación manual

Descarga la última versión ejecutable.

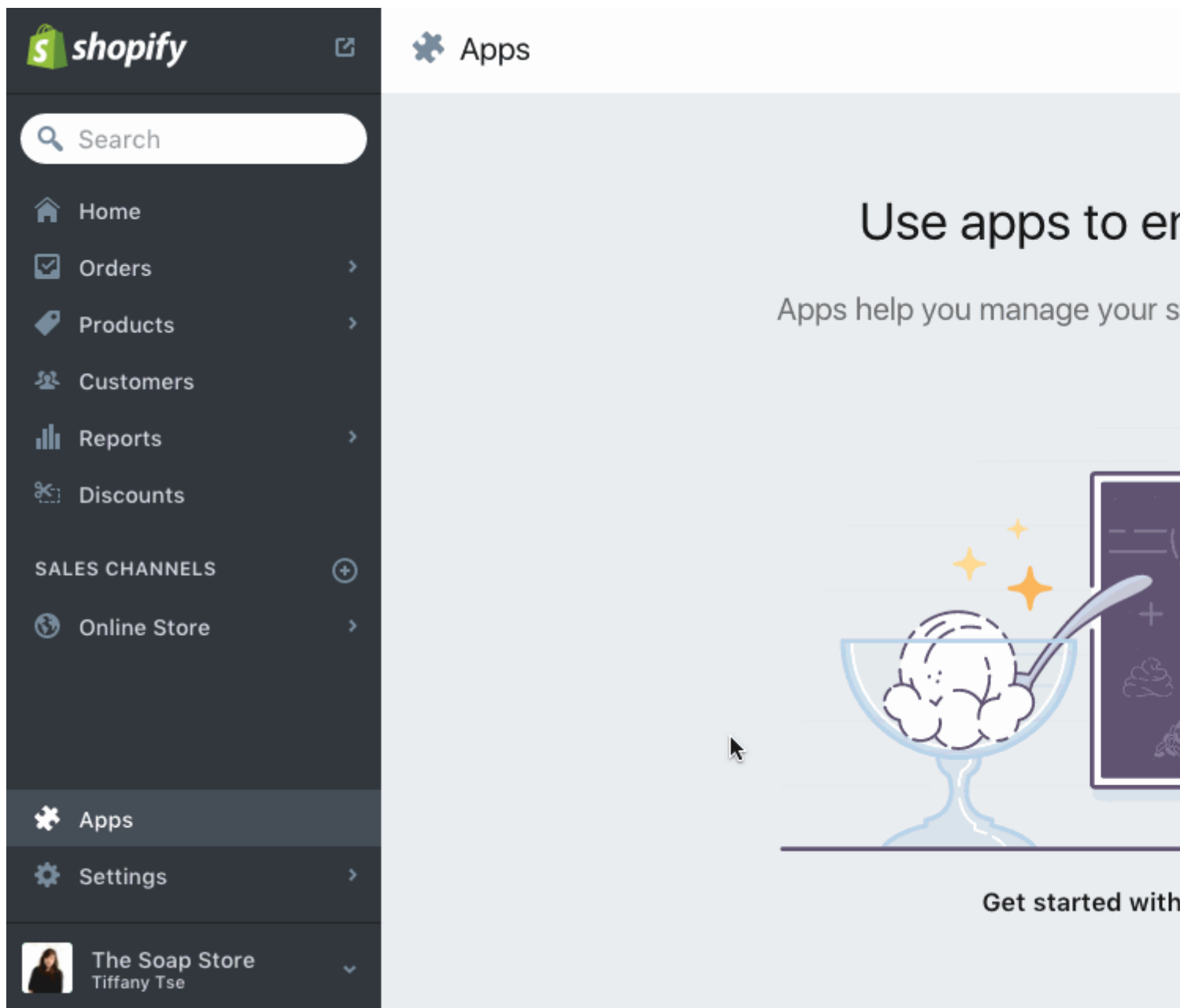
OS	Arquitectura	Enlazar
Mac OS	64 bits	Descargar
Windows	64 bits	Descargar
Windows	32 bits	Descargar
Linux	64 bits	Descargar
Linux	32 bits	Descargar

Obtener acceso a la API

Necesitará configurar una clave API para agregar a nuestra configuración y crear una conexión entre su tienda y el Kit de temas. La clave API le permite al Kit de temas hablar y acceder a su

tienda, así como a sus archivos de temas.

Para hacerlo, inicie sesión en la tienda Shopify y cree una aplicación privada. En Shopify Admin, vaya a Aplicaciones y haga clic en Ver aplicaciones privadas. Desde allí, haga clic en Generar credenciales API para crear su aplicación privada. Asegúrese de establecer los permisos de las plantillas de tema y los recursos del tema para tener acceso de lectura y escritura para generar las credenciales de API adecuadas, luego haga clic en Guardar.



Complete la información en la parte superior y configure los permisos de las plantillas del tema y los recursos del tema para el acceso de lectura y escritura. Presiona Guardar y aparecerá la siguiente pantalla. En ella podrás ver tus credenciales de acceso. Por favor, tome nota de la contraseña. Lo necesitarás más tarde.

Authentication

With these credentials you can use your [private app](#) to access your store through the [Shopify API](#).

API key
28c59b68796ff0bad3b53e042de8d557

Password
96d9631b4090589db4202c8a5f24cb5c

Shared secret
880c7b2c2d36aaa989f65650a0a69ce4

URL format
`https://apikey:password@hostname/admin/resource`

Example URL
`https://28c59b68796ff0bad3b53e042de8d557:96d9631b4090589db4202c8a5f24cb5c@can-i-buy-a-feeling.myshopify.com/admin/orders`

Utilice un nuevo tema.

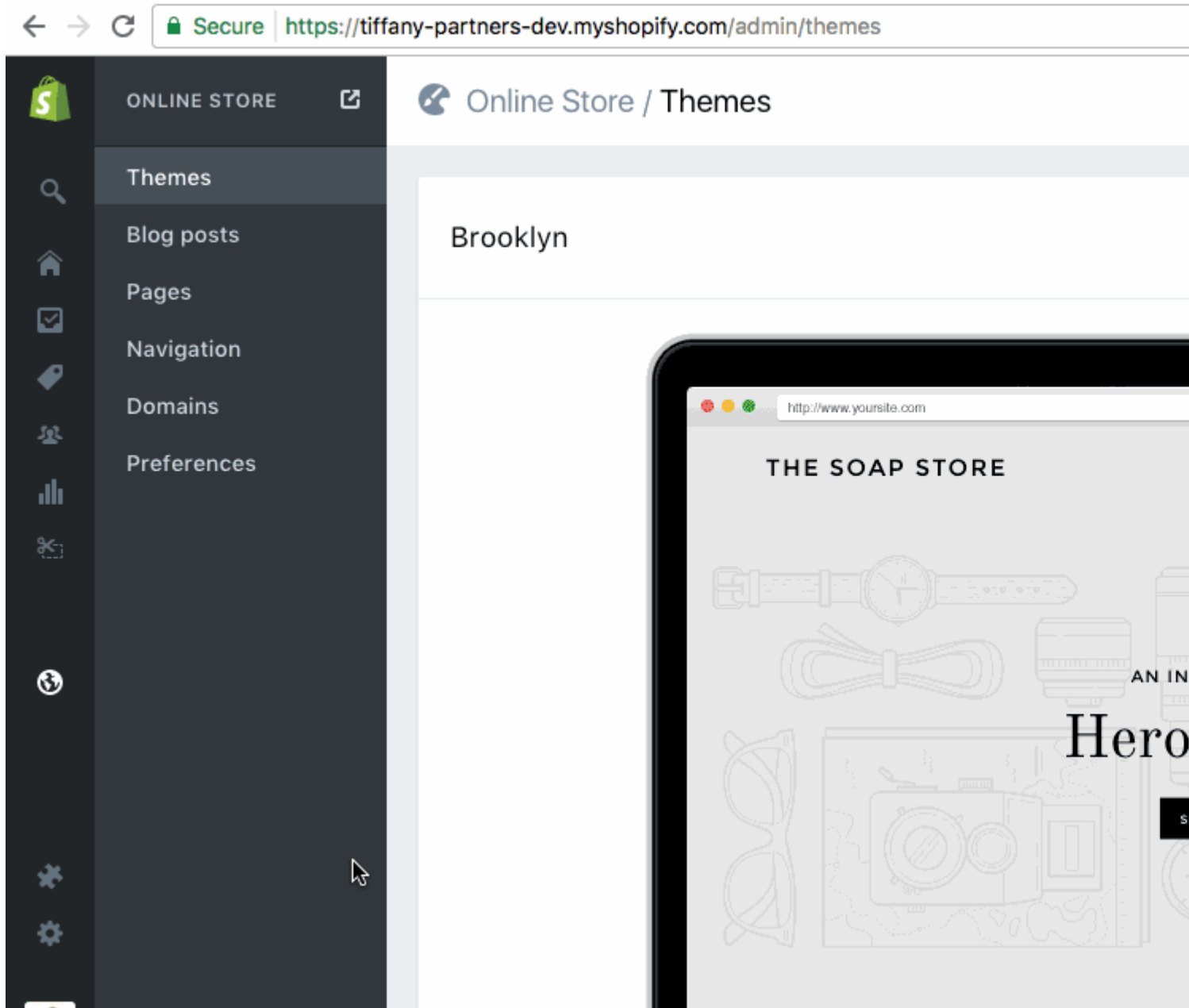
Si está comenzando desde cero y desea comenzar rápidamente, ejecute lo siguiente:

```
theme bootstrap --password=[your-password] --store=[your-store.myshopify.com]
```

Esto creará un nuevo tema para su tienda en línea desde la plantilla de [Timber](#). Luego, descargará todos esos recursos de Shopify y creará automáticamente un archivo `config.yml` para usted.

Configurar un tema existente.

Para conectar un tema existente, necesita el número de identificación del tema. La forma más fácil de obtener el número de identificación de su tema es ir al Editor de temas, haga clic en Editar HTML / CSS y copie el número de identificación del tema desde la URL: tendrán los últimos dígitos después de `mystore.myshopify.com/admin/themes/`.



Luego, una vez que haya anotado su ID de tema, ejecute los siguientes comandos:

```
# create configuration
theme configure --password=[your-password] --store=[you-store.myshopify.com] --themeid=[your-theme-id]
# download and setup project in the current directory
theme download
```

Lea Empezando con Shopify en línea: <https://riptutorial.com/es/shopify/topic/5772/empezando-con-shopify>

Capítulo 2: Implementando Secciones

Introducción

Las secciones son una función que permite a los propietarios de tiendas agregar, editar, eliminar y reordenar fácilmente el contenido de una página. Existen 2 tipos de secciones: dinámicas y fijas. Las secciones dinámicas se pueden reordenar en su totalidad con otras secciones en la página (página de inicio). Las secciones fijas no se pueden reordenar, pero el contenido dentro de ellas todavía se puede agregar, editar, eliminar y reordenar.

Observaciones

Cosas para recordar:

- El esquema de una sección está orientado a esa sección en particular. Esto significa que puede dar a sus variables nombres relativamente simples sin preocuparse de que entren en conflicto con nombres de variables de otras secciones en la misma página.

Más recursos:

- <https://help.shopify.com/themes/development/theme-editor/sections>
- <https://help.shopify.com/themes/development/theme-editor/settings-schema>

Examples

Incluyendo una sección en una página

Incluir una sección estática en una página es similar a incluir un fragmento. La única diferencia es que en lugar de usar la palabra `include`, usas la `section` palabras.

```
{% section 'section-file-name' %}
```

El uso de esta etiqueta "codificará" la sección de su tema en el lugar donde coloca el código y no se puede mover ni eliminar de la página.

Para usar las Secciones Dinámicas, debe usar el control `preset` en la parte inferior de la configuración del esquema para esa sección.

```
{% schema %}
{
  "presets": [
    {
      "category": "Custom Content",
```

```

    "name": "Text",
    "settings": {
      "heading": "Hello World"
    },
    "blocks": [
      {
        "type": "text",
        "settings": {
          "content": "Once upon a time..."
        }
      }
    ]
  }
]
}
{% endschema %}

```

Al usar los ajustes preestablecidos, esto le permitirá moverse y agregar su sección alrededor de su página según lo requiera. También le permite completar la sección con contenido ficticio para que pueda ver exactamente cómo se verá la sección en su página sin tener que agregarla a la página.

En esquema de archivo

Para las secciones, en lugar de utilizar un archivo JSON grande (como `settings_schema.json` , en cambio, mantienen su esquema dentro del archivo de la sección que lo está utilizando. Para hacer esto, se utilizan las etiquetas `{% schema %}` `{% endschema %}` y JSON se coloca entre ellos. Desde allí, el formato es similar a `settings_schema.json` .

```

{% schema %}
{
  "name": "Header Banner",
  "settings": [
    {
      "type": "checkbox",
      "id": "banner_enable",
      "label": "Enable Banner",
      "default": false
    },
    {
      "type": "color",
      "id": "banner_color",
      "label": "Banner Background Color",
      "default": "#000000"
    },
    {
      "type": "color",
      "id": "text_color",
      "label": "Banner Text Color",
      "default": "#ffffff"
    },
    {
      "type": "text",
      "id": "banner_text",
      "label": "Banner Text",
      "default": "Welcome to my Section"
    }
  ]
}

```

```
]
}
{% endschema %}
```

Para usar esto en su código HTML / líquido, debería extraerlo de `section.settings` . Algo como:

```
{% if section.settings.banner_enable %}
<div class="banner">
  <p>{{ section.settings.banner_text }}</p>
</div>
{% endif %}
```

Uso de bloques para elementos recurrentes

Una de las cosas más útiles que vinieron junto con las secciones son los **bloques** . Los bloques son básicamente un plan para algo que se puede crear una cantidad ilimitada de veces. Uno de los mejores ejemplos son las diapositivas de un control deslizante. Un bloque es un elemento de nivel superior en el esquema, lo que significa que está junto a cosas como `name` y `settings` , pero no dentro de ellos. Dentro de los bloques hay básicamente mini esquemas.

```
{% schema %}
{
  "name": "Slider",
  "max_blocks": 6,
  "blocks": [
    {
      "type": "slide",
      "name": "Slide",
      "limit": 4,
      "settings": [
        {
          "id": "image",
          "type": "image_picker",
          "label": "Image"
        }
      ]
    }
  ]
}
{% endschema %}
```

Observe que la diapositiva también tiene un tipo de `slide` . Puede asignar a sus bloques su propio tipo que no sea uno de los tipos provistos por Shopify.

Para usar esto, simplemente pasaría sobre cada bloque en su código HTML / Liquid.

```
<div class="slider">
  {% for block in section.blocks %}
    
  {% endfor %}
</div>
```

Usando JavaScript y hojas de estilo en secciones

Cada sección puede contener JavaScript y hojas de estilo, estos dos idiomas se usan en etiquetas líquidas: `{% javascript %}{% endjavascript %} & {% stylesheet %}{% endstylesheet %}`.

Al colocar el código dentro de estas dos etiquetas, Shopify compila cada pieza en `shopify_compiled.js` & `shopify_compiled.css`. Esto permite una mayor legibilidad dentro del código de cada sección, ya que no tiene que buscar en un documento largo para encontrar cada pieza.

Si no desea usar CSS, también puede usar SCSS utilizando `{% stylesheet 'scss' %}` lugar.

Sin embargo, hay inconvenientes en el uso de estos. El líquido no se puede utilizar en cualquiera de ellos, al igual que no se puede utilizar en normales `*.js`, `*.css`, `*.scss` archivos sin la `.liquid` extensión.

Además, al usar la etiqueta de hoja de estilo SCSS, esta no es una hoja de estilo global, por lo que no se podrá acceder a ninguna de las variables definidas en el documento `.scss` su tema y será necesario volver a definir las.

Ejemplos de uso

JavaScript

```
{% javascript %}
  $( "p" ).click(function() {
    $( this ).slideUp();
  });
{% endjavascript %}
```

CSS

```
{% stylesheet %}
  .container {
    width: 100%;
  }
  .container p {
    color: #ff0000;
  }
{% endstylesheet %}
```

HABLAR CON DESCARO A

```
{% stylesheet 'scss' %}
  $red: #ff0000;
  .container {
    width: 100%;
    p {
      color: $red;
    }
  }
{% endstylesheet %}
```

Lea Implementando Secciones en línea:

<https://riptutorial.com/es/shopify/topic/9198/implementando-secciones>

Creditos

S. No	Capítulos	Contributors
1	Empezando con Shopify	Community , drip , Inzamam Tahir , Sam
2	Implementando Secciones	BeesonBison , ConduciveMammal , Sam