

El plugin Scrollspy se usa para actualizar automáticamente , la posición de desplazamiento, de los enlaces en una lista de navegación basada en a menudo se usa junto con el plugin AFFIX

## data- \* Atributos

`data-spy="scroll"` para el elemento que debe usarse como área desplazable (a menudo este es el `<body>`).

Al desplazarse por elementos que no sean el `<body>` , asegúrese de tener una `height` establecida y `overflow-y: scroll;` aplicado.

`data-target` con un valor de id o el nombre de clase de la barra de navegación ( `.navbar`). Esto es para asegurarse de que la barra de navegación esté conectada con el área desplazable.

OJO!! **los elementos desplazables deben coincidir con la ID de los enlaces dentro de los elementos de la lista de la barra de navegación** ( `<div id="section1">coincidencias <a href="#section1">`).

El atributo `data-offset` especifica el número de píxeles a desplazar desde la parte superior cuando se calcula la posición del desplazamiento. Esto es útil cuando sientes que los enlaces dentro de la barra de navegación **cambian el estado activo demasiado pronto** al saltar a los elementos desplazables. El valor predeterminado es 10 píxeles.

El vinculo de con el body a lo mejor no encaja, y se activa antes de que se vea bien, por lo que se puede jugar con este atributo para que encaje.

Se trata de añadir Píxeles para compensar desde la parte superior al calcular la posición del desplazamiento.

**Requiere posicionamiento relativo:** el body o área desplazable, con `data-spy = "scroll"`, requiere la propiedad de **posición** CSS , con un valor de "relativo" para que funcione correctamente.

## Métodos

`.scrollspy('refresh')`

Cuando uses scrollspy junto con agregar o eliminar elementos del DOM, deberás llamar al método de actualización de la siguiente manera:

```
$ ( '[data-spy="scroll"]' ). each ( function () { var $spy = $ ( this ). scrollspy ( 'refresh' ) })
```

## Eventos

### activate.bs.scrollspy

Este evento se activa cada vez que se activa un nuevo elemento mediante scrollspy.

```
$ ( '#myScrollspy' ). on ( 'activate.bs.scrollspy' , function () { // do something... })
```

## ¿Cómo llevar a cabo esto?

1. En head:
  - a. Meta viewport en
  - b. Hoja de estilo de bootstrap
  - c. Librería genérica JQuery
  - d. Plugin genérica bootstrap
2. hacemos los enlaces con sus respectivas funciones
3. creamos el style con posición **relativa en el body**, y un pequeño estilo con margen entre las imágenes
4. elemento nav con **class="navbar navbar-inverse navbar-fixed-top"** con esto tenemos ya la barra de navegación fija, con fondo negro. Todavía los vínculos están con viñetas y en azul....
5. A cada uno de los vínculos le ponemos el class **class="navbar-brand"** con esto conseguimos que se quite el estilo predeterminado de los vínculos (se quita la viñeta, el color azul y subrayado)
- 6.

A continuación os explicaré cómo implementar Scrollspy. Para ello:

- Instala Bootstrap
- Añade una barra de navegación y contenido
- Configura scrollspy

## Instalar Bootstrap

Añade las librerías de bootstrap dentro de las etiquetas head de tu página web

```
<head>
...
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<!-- Optional theme -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap-theme.min.css">
<!-- Latest compiled and minified JavaScript -->
```

```
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.1
1.3/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/
js/bootstrap.min.js"></script>
        ...
</head>
```

## Añade una barra de navegación y contenido

### Barra de navegación

```
<div class="bs-docs-sidebar">
  <ul class="nav">
    <li><a href="#one">hello Bootstrp 3</a></li>
    <li><a href="#two">hello jQuery</a></li>
    <li><a href="#three">hello ScrollSpy</a></li>
  </ul>
</div>
```

### Contenido

```
<div id="one">
  <h1>This is section one.</h1>
  <p>...</p>
</div>
<div id="two">
  <h1>This is section two.</h1>
  <p>...</p>
</div>
<div id="three">
  <h1>This is section three.</h1>
  <p>...</p>
</div>
```

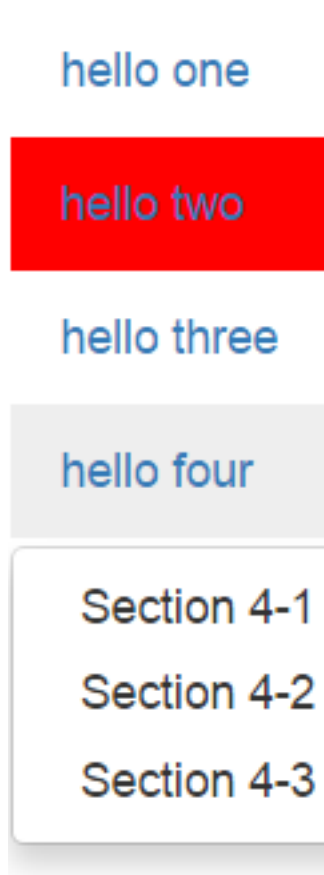
Hemos creado una estructura básica en html para que, al pinchar el anchor, salte al contenido.

## Añadir un menú dropdown (opcional)

Añadir un menú dropdown es sumamente sencillo. Para ello, insertamos la clase dropdown dentro del ítem, añadimos la clase dropdown-toggle y el atributo data-toggle="dropdown" al tag del enlace y por último, creamos una lista con todos los menús y submenús. Aquí lo podéis ver en este código:

```
<li class="dropdown open">
  <a href="#" class="dropdown-toggle" data-toggle="dropdown">hello four</a>
  <ul class="dropdown-menu">
    <li><a href="#four1">Section 4-1</a></li>
    <li><a href="#four2">Section 4-2</a></li>
    <li><a href="#four3">Section 4-3</a></li>
  </ul>
</li>
```

## Resultado Final



# This is section two

Nullam sagittis, ante eu elementum semper, ma  
lorem ante sed enim. In eu vestibulum quam. Se  
sodales metus varius. Morbi eget sodales sem.  
nec consectetur velit consectetur quis. Cras risu  
imperdiet a lorem. In ante tellus, mattis id elit et,  
gravida dui vel velit vulputate ultrices. Nunc sag

Semper tortor vitae arcu tincidunt, vitae tincidunt  
urna. Nullam sed lacinia ante. Proin ac purus nu  
ipsum dolor sit amet, consectetur adipiscing elit.  
justo blandit, sodales dui. Aliquam quis sagittis s  
varius tempor. Vestibulum non sodales lectus.

## Configurar scrollspy

### Método 1 - CSS puro

Añade dos atributos al body: data-spy="scroll" y data-target="target". Aquí el target puede ser el id de una clase o elemento de la barra de navegación:

```
<body data-spy="scroll" data-target=".bs-docs-sidebar">...</body>
```

### Método 2 - vía jQuery

Añade el método scrollspy() al contenido

```
$(function() {  
    $('.bs-docs-sidebar').scrollspy();  
});
```

## Cómo crear un scrollspy dinámico con PHP y MySQL

A large blue rectangle occupies the upper half of the page. On the right side of this rectangle, the letters 'pt' are written in a large, white, italicized sans-serif font.

Hay veces que necesitamos que se haga scroll directamente a cierta sección de nuestra web. Por ejemplo, tenemos contenido sobre un tema diferente y queremos mostrar el contenido de ese tema en el scroll y resaltar dicho tema al ver contenido relacionado o mostrar contenido relacionado al hacer clic en el tema. Si estás desarrollando un proyecto utilizando Bootstrap o piensas en implementarlo en tu próximo proyecto, lo tienes muy fácil. El componente Scrollspy ya está disponible en Bootstrap y actualiza automáticamente la navegación o enumera los componentes del grupo según la posición del scroll para indicar qué enlace está activo actualmente en la pantalla. Por lo tanto, en este tutorial aprenderá cómo crear Bootstrap Scrollspy dinámico con PHP y MySQL.

Habiendo dicho esto, vamos a entrar en materia. La estructura de archivos para este ejemplo es la siguiente.

- index.php
- scrollspy.js

## Crear la tabla en la base de datos

Para mostrar el menú de navegación y el contenido de la base de datos, primero hay que crear una tabla, a la cual llamaremos scrollspy, en la cual almacenaremos la información para después mostrarla...

```
CREATE TABLE `scrollspy` (
  `id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `description` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

## Obtener datos de la base de datos

Ahora obtendremos la información de nuestra base de datos, concretamente de la tabla scrollspy. Guardaremos todos los datos en un array, para después mostrar el título del menú y el contenido.

```
<?php
$sql = "SELECT id, title, description FROM scrollspy LIMIT 20";
$resultset = mysqli_query($conn, $sql) or die("database error:". mysqli_error($conn));
$records = array();
while( $rows = mysqli_fetch_assoc($resultset) ) {
  $records[] = $rows;
}
?>
```

## Crear el scrollspy con Bootstrap utilizando contenido dinámico

Primero crearemos el menú de navegación HTML utilizando el array de PHP para mostrar el título de la página. Para implementar Scrollspy, debemos inicializar el ID del target para llamar a data-target, por lo que crearemos navbar\_scrollspy para utilizarlo con data-target y crear el scrollspy.

```
<div id="navbar_scrollspy" class="navbar navbar-static">
<div class="navbar-inner">
<div class="container" style="width: auto;">
<a class="brand" href="#">PHPZAG</a>
<ul class="nav">
<?php
$count = 0;
foreach ($records as $record) {
  $count++;
  $class = '';
  if($count == 1) {
    $class = 'active';
  }
  echo "<li class='". $class. "'><a href='#". $record['title']. "'>". $record['title']. "</a></li>";
}
?>
</ul>
</div>
</div>
</div>
```

Ahora definiremos el comportamiento del scrollspy para crear nuestro menú de navegación mediante el atributo `data-spy="scroll"` y el `data-target="#navbar_scrollspy"`. También necesitaremos añadir algo de CSS, como un `position: relative;` para que este elemento tenga el comportamiento de un scrollspy. Añadiremos el contenido HTML utilizando el array que hemos creado antes.

```
<div data-spy="scroll" data-target="#navbar_scrollspy" data-offset="50"
    class="scrollspy-nav">
<?php
foreach ($records as $data) {
echo "<h4 id='". $data['title']. "'>". $data['title']. "</h4>";
echo "<p>". $data['description']. "</p>";
}
?>
</div>
```

## Utilizar las funciones de scrollspy

También puedes utilizar las funciones de scrollspy y así definir el comportamiento de un scrollspy, sin necesidad de utilizar los data attribute. Para ello, debes utilizar la función `.scrollspy()` tal que así:

```
$( 'body' ).scrollspy({ target: '#navbar_scrollspy' })
```

Cuando utilizas scrollspy, añades y eliminas elementos del DOM, por lo que debes llamar al método `refresh` de este modo:

```
$( '[data-spy="scroll"]' ).each(function () {
var $spy = $(this).scrollspy('refresh')
})
```

## JS Affix (affix.js)

El complemento Affix permite que un elemento se adhiera (bloquee) a un área de la página. Esto a menudo se usa con menús de navegación o botones de iconos sociales, para hacer que se "adhieran" a un área específica mientras se desplaza hacia arriba y hacia abajo de la página.

El complemento activa y desactiva este comportamiento (cambia el valor de la posición de CSS de estático a fijo), dependiendo de la posición de desplazamiento.

El plug-in affx alterna entre tres clases: `.affix`, `.affix-top`, y `.affix-bottom`. Cada clase representa un estado particular. Debe agregar propiedades de CSS para manejar las posiciones reales, con la excepción de `position: fixed` en la `.affix` clase.

Para obtener más información, lea nuestro [Tutorial de Bootstrap Affix](#).

**Consejo:** El plugin [Affix](#) se usa a menudo junto con el plugin [Scrollspy](#).

# A través de data-\* Atributos

Agregue `data-spy="affix"` al elemento que desea espiar y el atributo para calcular la posición del desplazamiento. `data-offset-top|bottom="number"`

## Opciones de Affix

Las opciones se pueden pasar a través de atributos de datos o JavaScript. Para los atributos de datos, agregue el nombre de la opción

Name	Type	Default	Description
offset	number   object   function	10	<p>Specifies the number of pixels to offset from screen when calculating position of scroll. When using a single number, the offset is added to both top and bottom directions. If you only want to control the top or the bottom, use an object, like <code>offset: {top:25}</code></p> <p>For multiple offsets, use <code>offset: {top:25, bottom:50}</code></p> <p><b>Tip:</b> Use a function to dynamically provide an offset (can be useful for responsive designs)</p>
target	selector   node   element	the window object	Specifies the target element of the affix

a data-, como en data-offset = "".

## Eventos de Affix

La siguiente tabla enumera todos los eventos de afijo disponibles.



Event	Description
affix.bs.affix	Occurs before fixed positioning is added to the element (e.g, when the <code>.affix</code> class is added)
affixed.bs.affix	Occurs after fixed positioning is added to the element (e.g., after the <code>.affix</code> class is added)
affix-top.bs.affix	Occurs before the top element returns to its original (non-fixed) position (e.g., when the <code>.affix-top</code> class is added)
affixed-top.bs.affix	Occurs after the top element returns to its original (non-fixed) position (e.g., after the <code>.affix-top</code> class is added)
affix-bottom.bs.affix	Occurs before the bottom element returns to its original (non-fixed) position (e.g., when the <code>.affix-bottom</code> class is added)
affixed-bottom.bs.affix	Occurs after the bottom element returns to its original (non-fixed) position (e.g., after the <code>.affix-bottom</code> class is added)

## Más ejemplos

### Barra de navegación fija

Crea un menú de navegación horizontal fijo:

#### Ejemplo

```
<nav class="navbar navbar-inverse" data-spy="affix" data-offset-top="197">
```

Inténtalo tú mismo "

### Usando jQuery para fijar automáticamente una barra de navegación

Utilice el método [outerHeight \(\) de jQuery](#) para fijar la barra de navegación después de que el usuario haya desplazado pasando un elemento especificado (<encabezado>):

## Ejemplo

```
$(".navbar").affix({offset: {top:  
$(".header").outerHeight(true)} } });
```

Inténtalo tú mismo "

## Scrollspy y Affix

Usando el plugin [Affix](#) junto con el plugin [Scrollspy](#) :

### Menú horizontal (barra de navegación)

```
<body data-spy="scroll" data-target=".navbar" data-offset="50">
```

```
<nav class="navbar navbar-inverse" data-spy="affix" data-offset-  
top="197">
```

```
...
```

```
</nav>
```

```
</body>
```

Inténtalo tú mismo "

### Menú vertical (Sidenav)

```
<body data-spy="scroll" data-target="#myScrollspy" data-  
offset="15">
```

```
<nav class="col-sm-3" id="myScrollspy">
```

```
  <ul class="nav nav-pills nav-stacked" data-spy="affix" data-  
offset-top="205">
```

```
...
```

```
</nav>
```

```
</body>
```

Inténtalo tú mismo "

## Barra de navegación animada en afijo

Use CSS para manipular las diferentes clases de .affix:

### Ejemplo: cambiar el color de fondo y el relleno de la barra de navegación en desplazamiento

```
.affix {  
  top: 0;  
  width: 100%;  
  -webkit-transition: all .5s ease-in-out;  
  transition: all .5s ease-in-out;  
  background-color: #F44336;  
  border-color: #F44336;  
}
```

```
.affix a {
  color: #fff !important;
  padding: 15px !important;
  -webkit-transition: all .5s ease-in-out;
  transition: all .5s ease-in-out;
}

.affix-top a {
  padding: 25px !important;
}
```

Inténtalo tú mismo "

## Ejemplo - Deslice en la barra de navegación

```
.affix {
  top: 0;
  width: 100%;
  -webkit-transition: all .5s ease-in-out;
  transition: all .5s ease-in-out;
}

.affix-top {
  position: static;
  top: -35px;
}
```

Inténtalo tú mismo "

[Anterior](#)

[Siguiente](#)

## Crear un elemento fijo con Bootstrap

En este tutorial aprenderá a aplicar pinning, es decir, [posicionamiento fijo](#) en una barra de navegación o cualquier otro elemento de forma automática cuando se desplazan más allá de una cierta distancia, así como al activar y desactivar su fijación con el complemento de afijo de Bootstrap. La fijación de un elemento se habilita cambiando el valor de su [position](#) propiedad de CSS de **static** a **fixed**.

Para hacer esto, el complemento **.affix** alterna entre tres clases: **.affix**, **.affix-top** y **.affix-bottom**. Cada clase representa un estado particular.

- Inicialmente, el complemento agrega **.affix-top** o **.affix-bottom** para indicar que el elemento está en su posición más alta o más baja.
  - Cuando el elemento se desplaza más allá del límite de compensación proporcionado por el atributo de compensación de **data-offset-** el complemento reemplaza la **.affix-top** o **.affix-bottom** con la clase **.affix** (establece la **position: fixed;**), que desencadena la colocación real.
  - En este punto, se requiere la propiedad apropiada [top](#) o [bottom](#) CSS para determinar la posición del elemento de afijo en la ventana gráfica.
- Veamos el siguiente ejemplo para verlo en acción real.

## Habilitar Affix a través de atributos de datos

Puede agregar fácilmente comportamiento de afijo a cualquier elemento; simplemente agregue `data-spy="affix"` al elemento que desea espiar. A continuación, utilice `data-offset-` atributos para definir cuándo alternar la fijación de un elemento 'on' y 'off'.

### Ejemplo

[Prueba este código »](#)

1. `<ul class = "nav nav-tabs nav-stacked" data-spy = "affix" data-offset-top = "195" >`
2. `<li class = "active" > <a href = "#one"> Sección uno </a> </li>`
3. `<li> <a href = "#two"> Sección dos </a> </li>`
4. `<li> <a href = "#three"> Sección tres </a> </li>`
5. `</ul>`

**Nota:** Los atributos de `data-offset-` solo especifican cuántos píxeles se deben desplazar para alternar la fijación de un elemento, no establecieron la posición del elemento anclado. Debe definir la propiedad de CSS `top` o `bottom` para el elemento anclado específicamente en su hoja de estilos para establecer su posición en la ventana gráfica.

## Habilitar Affix a través de JavaScript

También puede habilitar el complemento de afijo de forma manual con JavaScript: simplemente llame al método `affix()` con el `selector` de `id` o `class` del elemento requerido en su código JavaScript.

### Ejemplo

[Prueba este código »](#)

1. `<script type = "text / javascript" >`
2. `$ ( documento ). listo ( función () {`
3. `$ ( "#myNav" ). affix ({`
4. `desplazamiento : {`
5. `arriba : $ ( ".header" ). outerHeight ( verdadero )`
6. `}`
7. `});`
8. `});`
9. `</ script>`

## Opciones

Hay ciertas opciones que pueden pasarse al método de `affix()` Bootstrap para personalizar la funcionalidad del complemento affix.

Nombre	Tipo	Valor por defecto	Descripción
compensar	número   función	10	Especifique el número de píxeles para compensar desde la pantalla al calcular la posición del desplazamiento. Si se proporciona un solo número, el desplazamiento se aplicará en las direcciones superior e inferior. Para

Nombre	Tipo	Valor por defecto	Descripción
	objeto		establecer el desplazamiento para una sola dirección, o para múltiples compensaciones únicas, solo proporcione un objeto como <code>offset: {top:50, bottom:100}</code> También puede usar una función si desea proporcionar dinámicamente un desplazamiento en caso de diseños receptivos.
objetivo	selector   nodo   Elemento jQuery	el objeto <code>window</code>	Especifica el elemento objetivo del afijo.

También puede establecer estas opciones para el afijo utilizando los atributos de datos, simplemente agregue el nombre de la opción a `data-`, como `data-offset-top="195"`.

## Métodos

Estos son los métodos de fijación de bootstrap estándar:

### .affix (opciones)

Este método activa su contenido como contenido adjunto. Acepta un `object` opcional de opciones, como `offset: {top: 10}` o `offset: {top:50, bottom:100}`.

#### Ejemplo

[Prueba este código »](#)

```

1. <script type = "text / javascript" >
2. $( documento ). listo ( función () {
3. $( "#myNav" ). affix ({
4. desplazamiento : {
5. abajo : 195
6. }
7. });
8. });
9. </ script>
```

## Eventos

La clase de afijo de Bootstrap incluye pocos eventos para conectar a la funcionalidad modal.

Evento	Descripción
affix.bs.affix	Este evento se dispara inmediatamente antes de que el elemento se haya adherido.
affixed.bs.affix	Este evento se dispara después de que el elemento se haya adjuntado.

Evento	Descripción
affix-top.bs.affix	Este evento se dispara inmediatamente antes de que el elemento se haya colocado en la parte superior.
affixed-top.bs.affix	Este evento se dispara después de que el elemento se haya colocado en la parte superior.
affix-bottom.bs.affix	Este evento se dispara inmediatamente antes de que el elemento se haya fijado, abajo.
affixed-bottom.bs.affix	Este evento se dispara después de que el elemento se haya adherido a la parte inferior.

El siguiente ejemplo muestra un mensaje de alerta cuando se ha colocado el menú de navegación.

### ***Ejemplo***

**Prueba este código »**

```

1.  <script type = "text / javascript" >
2.  $ ( documento ). listo ( función () {
3.  $ ( "#myNav" ). on ( 'affixed.bs.affix', function () {
4.  alerta ( "El menú de navegación se ha colocado. Ahora no se desplaza con la página." );
5.  });
6.  });
7.  </ script>
```