



Elementos interactivos básicos y avanzados

Interactividad

La mayoría de los sitios web modernos incluyen alguna forma de interactividad. La interactividad suele integrarse en un sitio web mediante CSS y JavaScript, un lenguaje de programación web utilizado para mejorar un sitio web y crear interactividad.

Con CSS ya hemos utilizado animaciones y también hemos hecho algún ejercicio con Javascript, por lo tanto, en este capítulo veremos un framework nuevo: Bootstrap.

Bootstrap

Introducción

Bootstrap es un marco de trabajo gratuito para el diseño de sitios y aplicaciones web. Proporciona un conjunto de plantillas de diseño basadas en CSS y JavaScript para tipografía, formularios, botones, navegación y otros componentes de la interfaz. Bootstrap también incluye un sistema de cuadrícula adaptable que permite a los diseñadores crear fácilmente diseños que se ven bien en una variedad de dispositivos, incluyendo teléfonos, tabletas y ordenadores de sobremesa.

Introducción

Una de las principales ventajas de utilizar Bootstrap es que permite a los diseñadores crear rápidamente sitios web y aplicaciones de aspecto profesional sin tener que empezar desde cero. Incluye una amplia gama de elementos de interfaz de usuario prediseñados que se pueden personalizar fácilmente, lo que lo convierte en una opción popular para desarrolladores y diseñadores que desean crear sitios web hermosos y con capacidad de respuesta de forma rápida y eficiente.

Breakpoints

Breakpoints

Los puntos de interrupción son anchos personalizables que determinan cómo se comporta el diseño adaptable en función del tamaño del dispositivo o de la ventana gráfica.

Breakpoints

- Los puntos de interrupción son los componentes básicos del diseño responsive.
- Utiliza las media queries para diseñar tu CSS por puntos de interrupción. Permiten aplicar estilos de forma condicional en función de un conjunto de parámetros del navegador y del sistema operativo.
- El objetivo es el diseño responsive y mobile first. El CSS de Bootstrap tiene como objetivo aplicar el mínimo de estilos para hacer que un diseño funcione en el punto de interrupción más pequeño, y luego capas de estilos para ajustar ese diseño para dispositivos más grandes.

Breakpoints

Breakpoint	Class infix	Dimensions
Extra small	<i>None</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Contenedores

Cómo funcionan

Los contenedores son el elemento de diseño más básico en Bootstrap y son necesarios cuando se utiliza nuestro sistema de cuadrícula por defecto. Los contenedores se utilizan para contener, rellenar y (a veces) centrar el contenido que contienen. Aunque los contenedores pueden anidarse, la mayoría de los diseños no requieren un contenedor anidado.

Container

Bootstrap viene con tres contenedores diferentes:

- `.container`, que establece un `max-width` en cada punto de ruptura responsive
- `.container-{breakpoint}`, que es `width: 100%` hasta el punto de interrupción especificado
- `.container-fluid`, que es `width: 100%` en todos los puntos de interrupción.

Container

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

Container

Los contenedores responsive permiten especificar una clase con un ancho del 100% hasta que se alcanza el punto de rotura especificado, tras lo cual se aplican anchos máximos para cada uno de los puntos de rotura superiores.

```
<div class="container-sm">100% wide until small breakpoint</div>
```

```
<div class="container-md">100% wide until medium breakpoint</div>
```

```
<div class="container-lg">100% wide until large breakpoint</div>
```

```
<div class="container-xl">100% wide until extra large breakpoint</div>
```

```
<div class="container-xxl">100% wide until extra extra large breakpoint</div>
```

Grid

Grid

Grid es una cuadrícula flexbox mobile-first para crear diseños de todas las formas y tamaños gracias a un sistema de doce columnas, seis niveles responsivos predeterminados, variables y docenas de clases predefinidas.

Cómo funciona

La cuadrícula admite seis puntos de interrupción de respuesta. Los puntos de ruptura se basan en medias queries de anchura mínima, lo que significa que afectan a ese punto de ruptura y a todos los que están por encima de él (por ejemplo, `.col-sm-4` se aplica a `sm`, `md`, `lg`, `xl` y `xxl`).

Cómo funciona

Los contenedores centran y rellenan horizontalmente el contenido.

- `.container`: para un ancho de píxel adaptable.
- `.container-fluid`: para un `width: 100%` en todas las ventanas gráficas y dispositivos.
- `.container-md`: para una combinación de anchos de píxel y fluido.

Cómo funciona

Las filas son envoltorios de columnas. Cada columna tiene un relleno horizontal (llamado gutter) para controlar el espacio entre ellas. Este relleno se contrarresta en las filas con márgenes negativos para asegurar que el contenido de las columnas está visualmente alineado a la izquierda.

Las filas también soportan clases modificadoras para aplicar uniformemente el tamaño de las columnas y clases de gutter para cambiar el espaciado de tu contenido.

Cómo funciona

Las columnas son increíblemente flexibles. Hay 12 columnas de plantilla disponibles por fila, lo que te permite crear diferentes combinaciones de elementos que abarcan cualquier número de columnas.

Las clases de columna indican el número de columnas de la plantilla que deben abarcarse (por ejemplo, col-4 abarca cuatro). Las anchuras se establecen en porcentajes para que siempre tengas el mismo tamaño relativo.

Cómo funciona

Los gutter también son adaptables y personalizables. Las clases de gutter están disponibles en todos los breakpoints, con los mismos tamaños que nuestro espaciado de márgenes y relleno.

Cambia los gutter horizontales con las clases `.gx-*`, los gutter verticales con `.gy-*`, o todos los gutter con las clases `.g-*`. `g-0` también está disponible para eliminar los gutter.

Ejemplos

<https://getbootstrap.com/docs/5.2/examples/grid/#containers>

<https://getbootstrap.com/docs/5.2/layout/columns/>

<https://getbootstrap.com/docs/5.2/layout/gutters/>

Ejercicio

Quiero que me hagáis el siguiente ejercicio con Bootstrap:
Haced un grid con 50 monedas. Las monedas irán en filas de 10 en 10. Cada moneda ocupará como máximo el 300px y como mínimo 200px.

En el caso de que el tamaño total de la fila sea mayor que la ventana entonces se tendrán que crear 11 filas con 9 monedas cada fila.

Estos cambios se harán sucesivamente hasta llegar a filas con una sola columna.

Contenido

Reboot

Reboot, una colección de cambios CSS específicos puestos en marcha por Bootstrap para proporcionar un estilo base, coherente y sencilla sobre la que construir. Hay cambios en los siguientes apartados:

- **Page defaults:** en el body/main se han hecho varios cambios de estilos y de box-sizing.
- **Letra:** la letra por defecto está cambiada.
- **Forms:** hay estilos específicos que se han cambiado a todos los inputs
-

Imágenes

Documentación y ejemplos para optar por un comportamiento responsivo de las imágenes (para que nunca sean más grandes que sus elementos padre) y añadirles estilos ligeros, todo mediante clases.

Las imágenes en Bootstrap se hacen responsive con `.img-fluid`. Esto aplica `max-width: 100%;` y `height: auto;` a la imagen para que escale con el elemento padre.

Tablas

Existen una gran variedad de clases creadas para la estilización de tablas con Bootstrap.

```
<table class="table-primary">...</table>
```

```
<table class="table-secondary">...</table>
```

```
<table class="table-success">...</table>
```

....

Estos estilos se pueden modificar también a las etiquetas de tr y td.

Figure

Existen clases para las figuras también, un ejemplo:

```
<figure class="figure">  
    
  <figcaption class="figure-caption text-end">A caption.</figcaption>  
</figure>
```

Fin