

Ocultar <<

Módulo: 4

< **Anatomía de un sitio web**

Clase 10 - Introducción a Servidores ▾

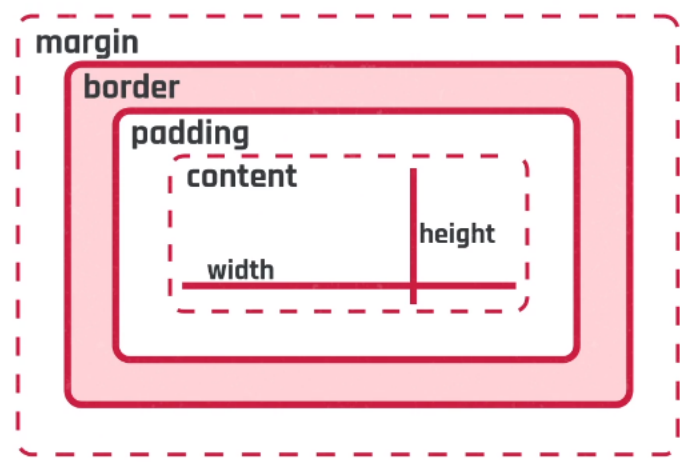
📺 Clase 10 - Clase en vivo: Introducción a Servidores ▾

Clase 11 - Creando la estructura de un sitio web ▾

📺 Clase 11 - Clase en vivo: Creando la estructura de un sitio web ▾

Clase 12 - Agregando estilos ▾

📺 Clase 12 - Clase en vivo: Agregando estilos ▾



¡Excelente! Gran trabajo.

Módulo: 4



## Anatomía de un sitio web

Clase 10 - Introducción a Servidores

Clase 10 - Clase en vivo: Introducción a Servidores

Clase 11 - Creando la estructura de un sitio web

Clase 11 - Clase en vivo: Creando la estructura de un sitio web

Clase 12 - Agregando estilos

Clase 12 - Clase en vivo: Agregando estilos

Clase 13 - Diseño adaptativo

Para este ejercicio contamos con una pequeña estructura HTML definida con dos etiquetas `<div>`.

Nuestro trabajo será agregarles a estas cajas propiedades de CSS que solo entiendan los elementos de bloque. Estas son las características que deberán tener:

- Ancho fijo de 100px
- Borde sólido rojo de 1px
- Padding de 5px
- Margen de 10px

Por último, queremos que el ancho final de cada `<div>` sea de 100px incluyendo el borde. ¿Con qué propiedad podemos configurar eso?

Dame una pista

Con la propiedad `box-sizing: border-box` el navegador se encargará de respetar el ancho definido de cada elemento.

```
PlaygroundCode
index.html x style.css x
SRC PATH: SRC > STYLE.CSS
index.html
style.css
1 div {
2   width: 100px;
3   padding: 5px;
4   border: solid 1px red;
5   margin: 10px;
6   box-sizing: border-box;
7 }
```

¡Excelente! Gran trabajo.

Ocultar <<

Módulo: 4



## Anatomía de un sitio web

Clase 10 - Introducción a Servidores

Clase 10 - Clase en vivo: Introducción a Servidores

Clase 11 - Creando la estructura de un sitio web

Clase 11 - Clase en vivo: Creando la estructura de un sitio web

Clase 12 - Agregando estilos

Clase 12 - Clase en vivo: Agregando estilos

Clase 13 - Diseño adaptativo

Box Model



http://playground.digitalhouse.com



Caja 1

Caja 2

Ocultar <<

Módulo: 4



## Anatomía de un sitio web

✓ Agregando estilos

### Clase 13 - Diseño adaptativo

✓ ¿Qué vamos a ver en esta clase?

✓ Modelo de caja

✓ Viewports

✓ Medidas relativas

⦿ **Media queries**

⦿ Proyecto: diseño adaptativo

⦿ Proyecto integrador: Se viene el segundo sprint

⦿ Hacia la clase en vivo



**Clase 13 - Clase en vivo: Diseño adaptativo**



Mira las respuestas de cada pregunta:

1

¿Qué hacemos cuando escribimos nuestras reglas de CSS **mobile-first**?



Primero escribimos reglas de CSS en el ámbito global para teléfonos móviles y luego hacemos reglas más específicas dentro de media queries.



Las reglas de CSS para celulares tienen prioridad.



Queremos que el CSS cargue más rápido en dispositivos móviles.



¡Correcto! Mobile-first es una corriente de diseño que piensa primero en la visualización del sitio en dispositivos móviles. Por lo cual, escribimos las reglas generales del sitio o de la página pensando en estos dispositivos y luego las reglas de CSS para otros dispositivos en las media queries necesarias.

Ocultar <<

Módulo: 4

< **Anatomía de un sitio web**

✓ Agregando estilos

**Clase 13 - Diseño adaptativo** ^

✓ ¿Qué vamos a ver en esta clase?

✓ Modelo de caja

✓ Viewports

✓ Medidas relativas

⦿ **Media queries**

⦿ Proyecto: diseño adaptativo

⦿ Proyecto integrador: Se viene el segundo sprint

⦿ Hacia la clase en vivo

📺 **Clase 13 - Clase en vivo: Diseño adaptativo** 🔒

Mira las respuestas de cada pregunta:

2 ¿Para que sirven las **media queries**?

☐ Para definir reglas de CSS que se adapten al viewport.

☒ Para definir reglas de CSS que se apliquen en algunos viewport, pero no en otros.

☐ Para darle opciones de visualización a elegir por el usuario

✓ ¡Bien hecho! Cuando definimos reglas de CSS dentro de las media queries, estas solo afectan a algunos viewports que cumplen las condiciones que nosotros definimos.

⬅ Anterior

Siguiente ➡

1 2 3  
✓ ✓ ✓

Ocultar <<

Módulo: 4



## Anatomía de un sitio web

✓ Agregando estilos

### Clase 13 - Diseño adaptativo

- ✓ ¿Qué vamos a ver en esta clase?
- ✓ Modelo de caja
- ✓ Viewports
- ✓ Medidas relativas
- ⦿ **Media queries**
- ⦿ Proyecto: diseño adaptativo
- ⦿ Proyecto integrador: Se viene el segundo sprint
- ⦿ Hacia la clase en vivo



**Clase 13 - Clase en vivo: Diseño adaptativo**



Mira las respuestas de cada pregunta:

3

¿Qué hace la siguiente media query? **@media(min-device-width: 320px) and (max-device-width: 480px)**



Las reglas de CSS que escribamos adentro solo aplicarán si el viewport tiene entre 320px y 480px de ancho.



Las reglas de CSS que escribamos adentro solo aplicarán si el viewport no tiene entre 320px y 480px de ancho.



Las reglas de CSS dentro del media query se cargarán por default, pero tomarán prioridad si se cumplen las condiciones.



¡Correcto! Estamos indicando que las reglas que vamos a escribir solo van a ser aplicadas cuando el dispositivo cumpla las condiciones de tener entre 320px y 480px (inclusive) de ancho.



Anterior

Módulo: 4

< Anatomía de un sitio web

Clase 10 - Introducción a Servidores	▼
Clase 10 - Clase en vivo: Introducción a Servidores	▼
Clase 11 - Creando la estructura de un sitio web	▼
Clase 11 - Clase en vivo: Creando la estructura de un sitio web	▼
Clase 12 - Agregando estilos	▼
Clase 12 - Clase en vivo: Agregando estilos	▼
Clase 13 - Diseño adaptativo	▲

## Proyecto: diseño adaptativo



### Respuesta incorrecta

Tener en cuenta que la única función que cumple el viewport es la de informar al navegador que deberá ajustar el contenido del documento al tamaño que tenga el viewport. Todo lo demás lo deberemos generar con nuestras reglas de CSS.

Continuar



[< Anterior](#) [Siguiente >](#)



Clase 10 - Introducción a Servidores	▼
Clase 10 - Clase en vivo: Introducción a Servidores	▼
Clase 11 - Creando la estructura de un sitio web	▼
Clase 11 - Clase en vivo: Creando la estructura de un sitio web	▼
Clase 12 - Agregando estilos	▼
Clase 12 - Clase en vivo: Agregando estilos	▼
Clase 13 - Diseño adaptativo	^

A. Siempre píxeles.

B. En porcentajes o unidades del viewport.

C. Es indistinto, CSS es siempre CSS.

Anterior

Siguiente