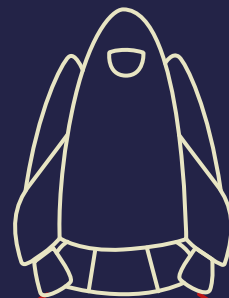


Programa académico CAMPUS



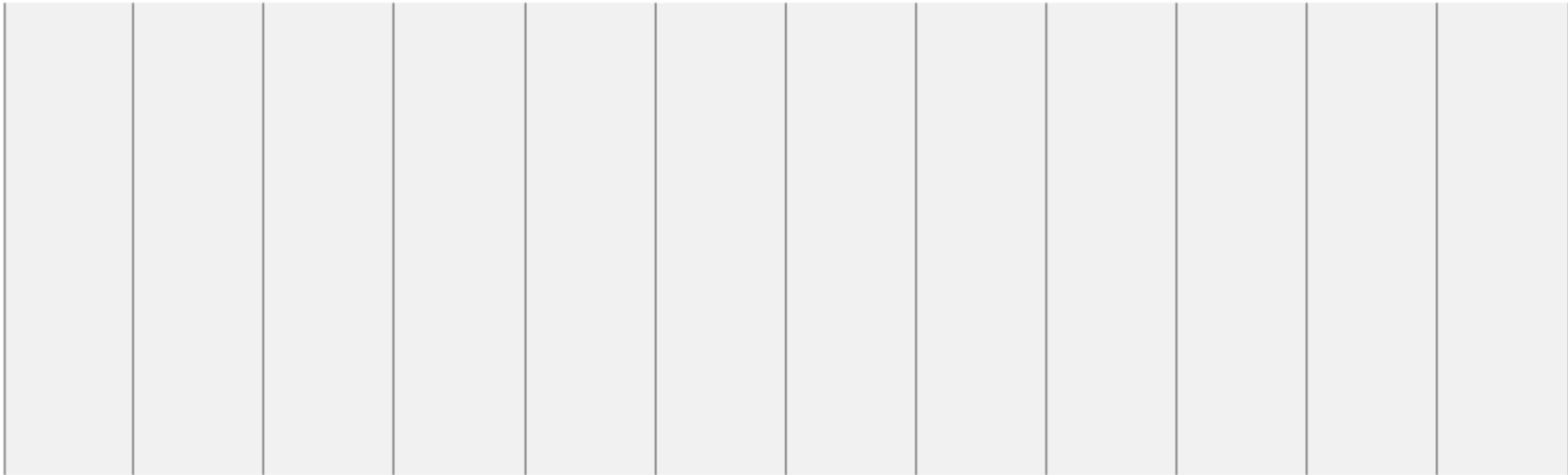
Ciclo 2:
CSS - Responsive Design –
Vista de Cuadrícula



DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



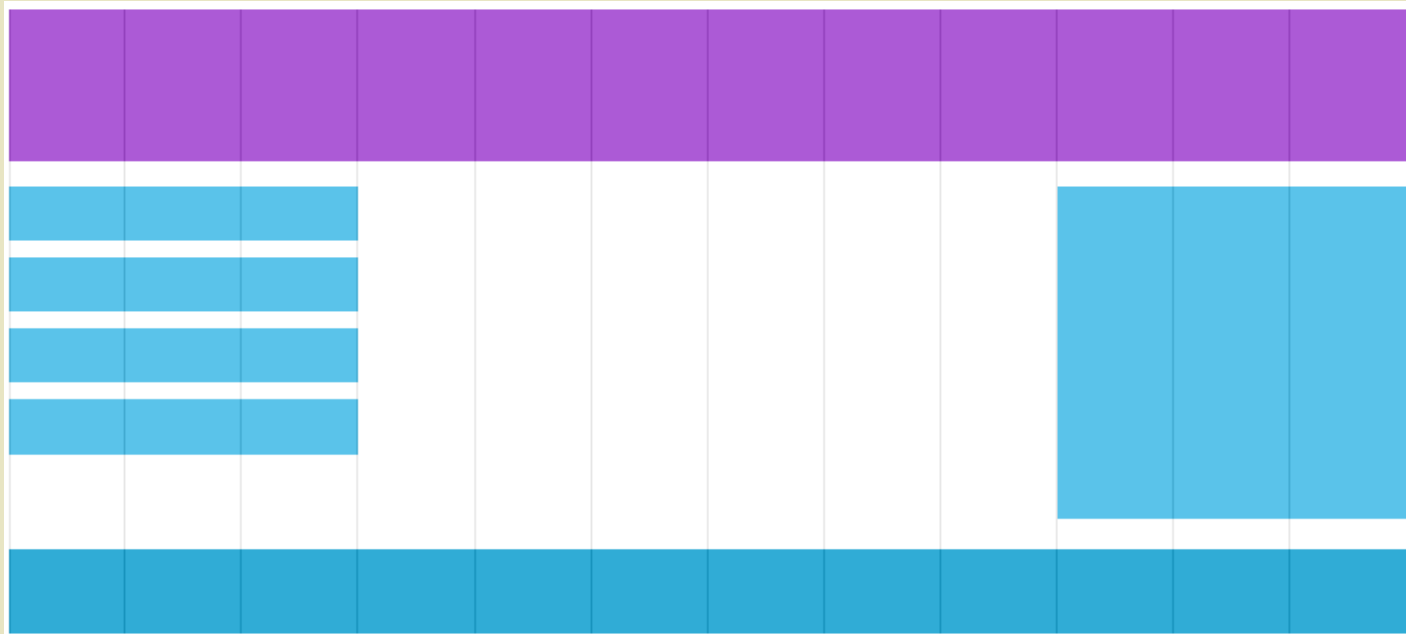
- Muchas páginas web se basan en una vista de cuadrícula, lo que significa que la página está dividida en columnas.



DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



El uso de una vista de cuadrícula es muy útil al diseñar páginas web. Facilita la colocación de elementos en la página..



DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



Una vista de cuadrícula receptiva a menudo tiene **12 columnas** y tiene un **ancho total del 100%**, y se reducirá y expandirá a medida que cambia el tamaño de la ventana del navegador.



DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



Primero asegúrese de que todos los elementos HTML tengan la `box-sizing` propiedad establecida en `border-box`. Esto asegura que el relleno y el borde estén incluidos en el ancho y alto total de los elementos.

Agregue el siguiente código en su CSS.

```
<style>
  * {
    box-sizing: border-box;
  }
</style>
```

BOX-SIZING



La propiedad CSS **box-sizing** nos permite incluir el relleno y el borde en el ancho y alto total de un elemento.

BOX-SIZING



Sin la propiedad de tamaño de caja CSS

De forma predeterminada, el ancho y la altura de un elemento se calculan así:

$\text{ancho} + \text{relleno} + \text{borde} = \text{ancho real de un elemento}$

$\text{alto} + \text{relleno} + \text{borde} = \text{altura real de un elemento}$

Esto significa: cuando establece el ancho/alto de un elemento, el elemento a menudo parece más grande de lo que ha establecido (porque el borde y el relleno del elemento se agregan al ancho/alto especificado del elemento).

BOX-SIZING



Sin la propiedad de tamaño de caja CSS

La siguiente ilustración muestra dos elementos `<div>` con el mismo ancho y alto especificados:

Este div es más pequeño (el ancho es de 300 px y la altura es de 100 px).

Este div es más grande (el ancho también es de 300 px y la altura es de 100 px).

Los dos elementos `<div>` anteriores terminan con diferentes tamaños en el resultado (porque `div2` tiene un relleno especificado)

BOX-SIZING



```
<style>
  .div1 {
    width: 300px;
    height: 100px;
    border: 1px solid blue;
  }

  .div2 {
    width: 300px;
    height: 100px;
    padding: 50px;
    border: 1px solid red;
  }
</style>
```

BOX-SIZING



```
<h1>SIN box-sizing</h1>
```

```
<div class="div1">
```

Este div es pequeño (width es 300px y height es 100px).

```
</div>
```

```
<br />
```

```
<div class="div2">
```

Este div es grande (width es also 300px y height es 100px).

```
</div>
```

BOX-SIZING



Con la propiedad CSS box-sizing

La **box-sizing** propiedad nos permite incluir el relleno y el borde en el ancho y alto total de un elemento.

BOX-SIZING



Con la propiedad CSS box-sizing

Si establece **box-sizing: border-box;** en un elemento, el relleno y el borde se incluyen en el ancho y la altura:

¡Ambos divs son del mismo tamaño ahora!

¡Hurra!

Aquí está el mismo ejemplo que el anterior, con **box-sizing: border-box;** agregado a ambos elementos `<div>`

BOX-SIZING



```
.div1 {  
  width: 300px;  
  height: 100px;  
  border: 1px solid blue;  
  box-sizing: border-box;  
}
```

```
.div2 {  
  width: 300px;  
  height: 100px;  
  padding: 50px;  
  border: 1px solid red;  
  box-sizing: border-box;  
}
```

BOX-SIZING



```
<body>
  <h1>CON box-sizing</h1>

  <div class="div1">¡Ambos divs son del
mismo tamaño ahora!</div>
  <br />
  <div class="div2">Hurra!</div>
</body>
```

BOX-SIZING



Dado que el resultado de usar el **box-sizing: border-box;** es mucho mejor, muchos desarrolladores quieren que todos los elementos de sus páginas funcionen de esta manera.

BOX-SIZING



El siguiente código garantiza que todos los elementos tengan el tamaño de esta forma más intuitiva. Muchos navegadores ya usan **box-sizing:border-box;** muchos elementos de formulario (pero no todos, por lo que las entradas y las áreas de texto se ven diferentes en el ancho: 100%;).

Aplicar esto a todos los elementos es seguro y sabio:

BOX-SIZING



```
body {  
    margin: 0;  
}  
  
* {  
    box-sizing: border-box;  
}  
  
input, textarea {  
    width: 100%;  
}
```

BOX-SIZING



```
<body>
  <form action="/action_page.php">
    Primer nombre:<br />
    <input type="text" name="firstname" value="Mickey"
/><br />
    Apellido:<br />
    <input type="text" name="lastname" value="Mouse"
/><br />
    Comentarios:<br />
    <textarea name="message" rows="5" cols="30">
</textarea>
    <br /><br />
    <input type="submit" value="Submit" />
  </form>
```

BOX-SIZING



```
<p>
```

```
  <strong>Tip:</strong> Intente eliminar  
la propiedad de tamaño de cuadro del elemento  
de estilo y observe lo que sucede. Tenga en  
cuenta que el ancho de la entrada, el área de  
texto y el botón de envío quedarán fuera de  
la pantalla.
```

```
  </p>
```

```
</body>
```

BOX-SIZING

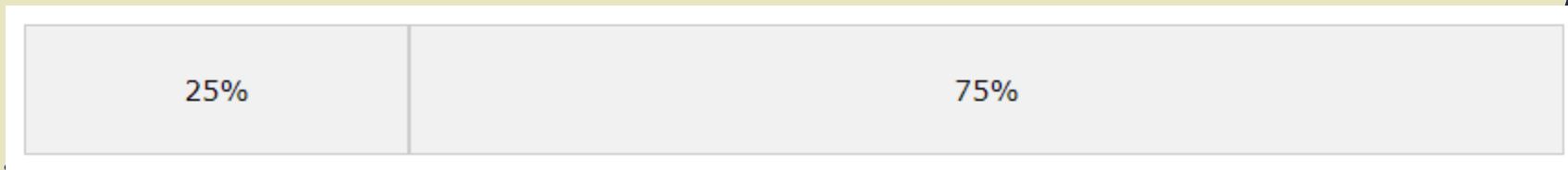


Property	Description
<u>box-sizing</u>	Define cómo se calculan el ancho y la altura de un elemento: si deben incluir relleno y bordes, o no

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



El siguiente ejemplo muestra una página web receptiva simple, con dos columnas:



DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
<style>
* {
  box-sizing: border-box;
}

.header {
  border: 1px solid red;
  padding: 15px;
}
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
.menu {  
  width: 25%;  
  float: left;  
  padding: 15px;  
  border: 1px solid red;  
}
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
.main {  
  width: 75%;  
  float: left;  
  padding: 15px;  
  border: 1px solid red;  
}  
</style>
```


DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
<body>
  <div class="header">
    <h1>Chania</h1>
  </div>
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
<div class="menu">
  <ul>
    <li>El Vuelo</li>
    <li>La Ciudad</li>
    <li>La Isla</li>
    <li>La Comida</li>
  </ul>
</div>
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
<div class="main">  
  <h1>La Ciudad</h1>  
  <p>
```

Chania es la capital de la región de Chania en la isla de Creta. La ciudad se puede dividir en dos partes, el casco antiguo y la ciudad moderna.

```
</p>
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



`<p>`

Cambie el tamaño de la ventana del navegador para ver cómo responde el contenido al cambio de tamaño.

`</p>`

`</div>`

`</body>`

· DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



- Ahora haremos una vista de cuadrícula responsiva con 12 columnas para tener más control sobre la página web.
- Primero debemos calcular el porcentaje para una columna:
 $100\% / 12 \text{ columnas} = 8,33\%$.
- Luego creamos una clase para cada una de las 12 columnas `class="col-"` y un número que define cuántas columnas debe abarcar la sección:

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



Todas estas columnas deben estar flotando a la izquierda y

- tener un relleno de 15px:

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
[class*="col-"] {  
    float: left;  
    padding: 15px;  
    border: 1px solid red;  
}
```


DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



Cada fila debe estar envuelta en un archivo `<div>`. El número de columnas dentro de una fila siempre debe sumar 12:

```
<div class="row">  
  <div class="col-3">...</div> <!-- 25% -->  
  <div class="col-9">...</div> <!-- 75% -->  
</div>
```

· DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



- Las columnas dentro de una fila están todas flotando hacia la izquierda y, por lo tanto, se eliminan del flujo de la página, y otros elementos se colocarán como si las columnas no existieran. Para evitar esto, agregaremos un estilo que borre el flujo:

```
.row::after {  
    content: "";  
    clear: both;  
    display: table;  
}
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



También queremos agregar algunos estilos y colores para que se vea mejor.

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA



```
html {  
    font-family: "Lucida Sans", sans-serif;  
}  
  
.header {  
    background-color: #9933cc;  
    color: #ffffff;  
    padding: 15px;  
}  
  
.menu ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
}
```

DISEÑO ADAPTATIVO – VISTA CUADRÍCULA

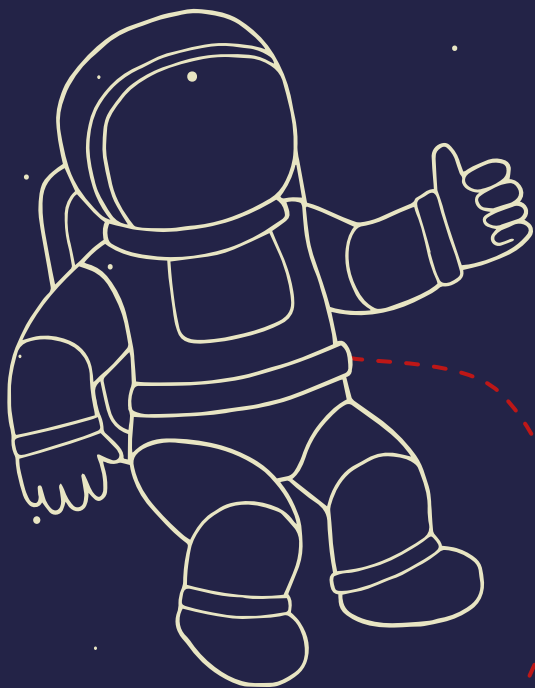


```
.menu li {  
    padding: 8px;  
    margin-bottom: 7px;  
    background-color :#33b5e5;  
    color: #ffffff;  
    box-shadow: 0 1px 3px rgba(0,0,0,0.12), 0 1px 2px rgba(0,0,0,0.24)  
}  
  
.menu li:hover {  
    background-color: #0099cc;  
}
```

· **DISEÑO ADAPTATIVO – VISTA CUADRÍCULA**



Ver ejemplo: [07-web-12-áreas.html](#)



Programa académico CAMPUS



Ciclo 2:
Programación WEB

