# CSS
# Layout

# CSS Layout

En general, podemos dividir las propiedades CSS en dos tipos:

- Apariencia (color, background, border, etc.)
- Organización (display, position, margin, etc.)

# CSS Layout

**display** controla el comportamiento general de un elemento.

- **block** ocupa todo el ancho disponible
- **inline** se comporta como una palabra
- **inline-block** combina los otros valores
- **none** oculta completamente la etiqueta

# CSS Layout

**margin** controla el espacio alrededor del elemento.

- **10px** dimensión en pixeles
- **10%** porcentaje relativo al contenedor
- **10em** unidades relativas al tamaño de fuente
- **10vw** relativo al ancho de la ventana
- **10vh** relativo al alto de la ventana
- **auto** centra el elemento (sólo horizontal)

# CSS Layout

Diferencias en la cantidad de valores en una misma propiedad.

- **20x;** 20 pixeles en todas las direcciones
- **10px 20x;** 10 arriba y abajo, 20 a ambos lados
- **10px 30x 20px;** 10 arriba, 30 a ambos lados y 20 abajo
- **10px 20x 30px 40px;** arriba, derecha, abajo, izquierda (como las manecillas del reloj)

# CSS Selectores y Metodología

```
<section> Esta sección estará centrada horizontalmente mientras la pantalla mida menos de 500px. </section>

section {
    margin: 20px auto;
    width: 500px;
}
```

# CSS Layout

**max-width** define el ancho máximo que podrá tener un elemento (útil para responsive).

```
section {
   margin: 20px auto;
   max-width: 500px;
}
```

# CSS Layout

**box-sizing** define cómo se calcula el tamaño de los elementos, dependiendo de varias propiedades.

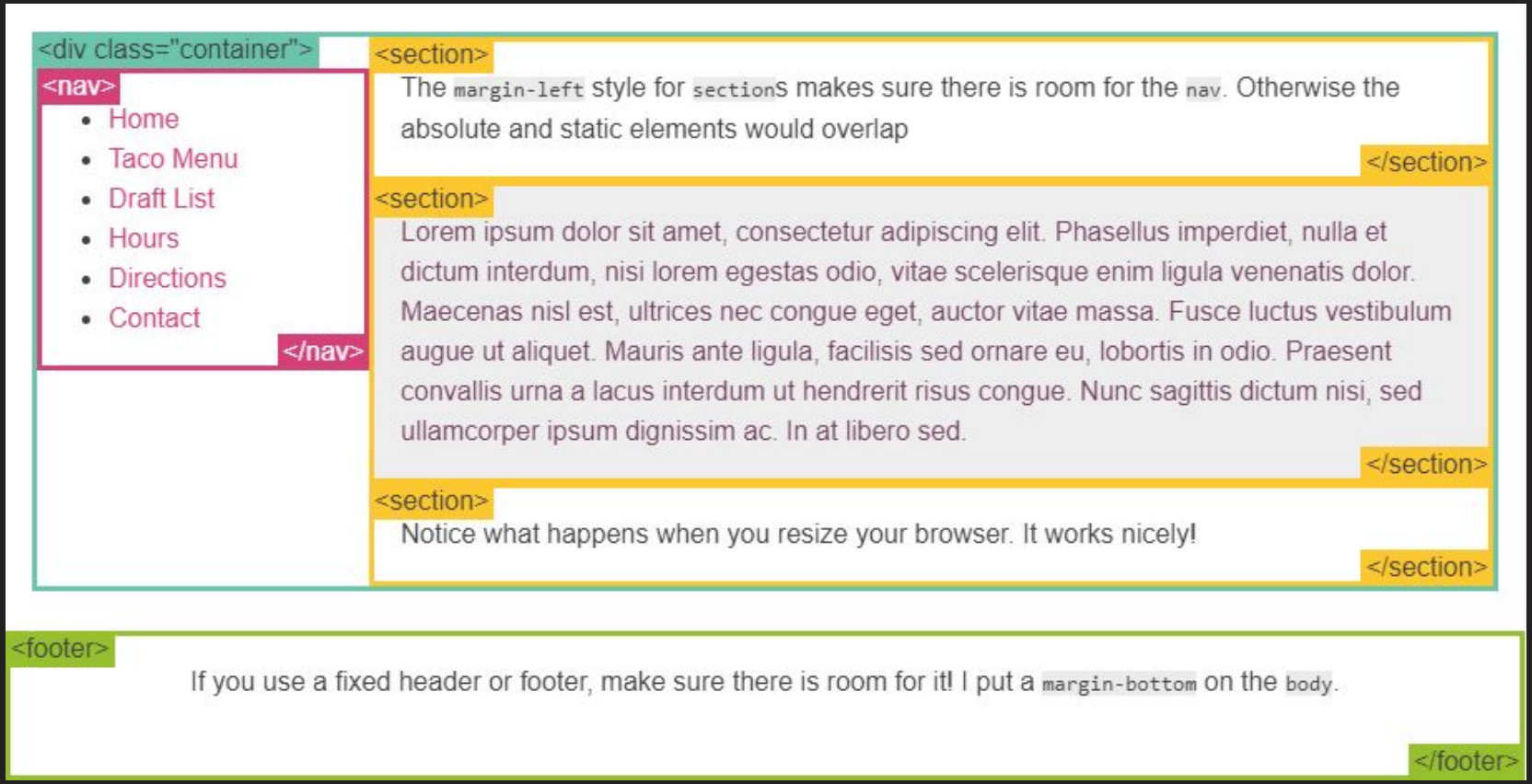
- **border-box** toma en cuenta el padding y border de un elemento dentro de su ancho.
- **content-box** suma el tamaño del padding y border al ancho que se ha definido con width.

# CSS Layout

**position** define el modo de posicionamiento que tiene un elemento.

- **static** es el valor por defecto, nada especial.
- **relative** permite utilizar las propiedades top, right, left y bottom para cambiar la posición.
- **fixed** ubica un elemento con relación a la ventana.
- **absolute** ubica un elemento con relación al primer padre que no tenga posición static.

# Ejercicio



<nav>
Home
Taco Menu
Draft List
Hours
Directions
Contact
</nav>
<section>
The margin-left style for sections makes sure there is room for the nav. Otherwise the absolute and static elements would overlap
</section>
<section>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed.
</section>
<section>
Notice what happens when you resize your browser. It works nicely!
</section>
<footer>
If you use a fixed header or footer, make sure there is room for it! I put a margin-bottom on the body.
</footer>

# CSS Selectores y Metodología

Guía sobre layout CSS (disponible en español)

http://learnlayout.com