Desarrollo de Aplicaciones Web

CSS (*Cascading Style Sheet*)

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

CSS. *Layout*.

- Layout Fijo.- se fija de forma absoluta la posición de las cajas, con independencia del tamaño de la ventana del navegador → Es necesario fijar el ancho del contenido total:

```
width:      800 px;
margin-left:        auto;
margin-right:       auto;
```

- Layout líquido.- fija de forma relativa la posición de las cajas. La colocación de las estas depende del ancho del “viewport” y de la posición donde decida ponerlas el navegador.

- Layout responsivo.- sería una variante de layout líquido,en donde el desplazamiento de las cajas es más controlado, al realizarse por bloques.

CSS. *Display*.

display → especifica la forma en que un elemento es "*renderizado" por parte del navegador.*

- display:none → elimina elemento a visualizar (incluido su renderizado).
- display:inline → transforma elemento de bloque en elemento de línea.
- display:block → transforma elemento de línea en elemento de bloque.
- display:inline-block → transforma elemento de bloque en elemento de línea, pero preservando los atributos width y height.

CSS. *Posicionamiento*.

Posicionamiento.- Por defecto los elementos de la página son distribuidos por el navegador, siguiendo el flujo natural del fichero fuente.

CSS proporciona seis formas de posicionamiento de una caja → *se modifica el flujo normal de posicionamiento en la página.*

- static → *default*
- absolute
- fixed
- relative

} position: ⇨

- left
- right
- top
- bottom

- float
- relative float

} float:

- z-index
- clear

CSS. *Posicionamiento*.

• Posicionamiento Estático.- el elemento en cuestión es colocado en la posición que le corresponde según el flujo natural.

• Posicionamiento Absoluto.- el elemento en cuestión es colocado en una posición relativa a la página abierta por el navegador.

• Posicionamiento Fijo.- el elemento en cuestión es colocado en una posición relativa a la ventana del navegador.

• Posicionamiento Relativo.- el elemento en cuestión es colocado en una posición relativa a la posición que tendría según el flujo natural.

• Posicionamiento Flotante.- el elemento en cuestión flota, elevándose sobre la página, propiciando el desplazamiento del resto de elementos que acuden a tapar el hueco que deja el elemento flotante.

• Posicionamiento Flotante Relativo.- idem flotante, pero partiendo de posición relativa.

CSS. *Posicionamiento*.

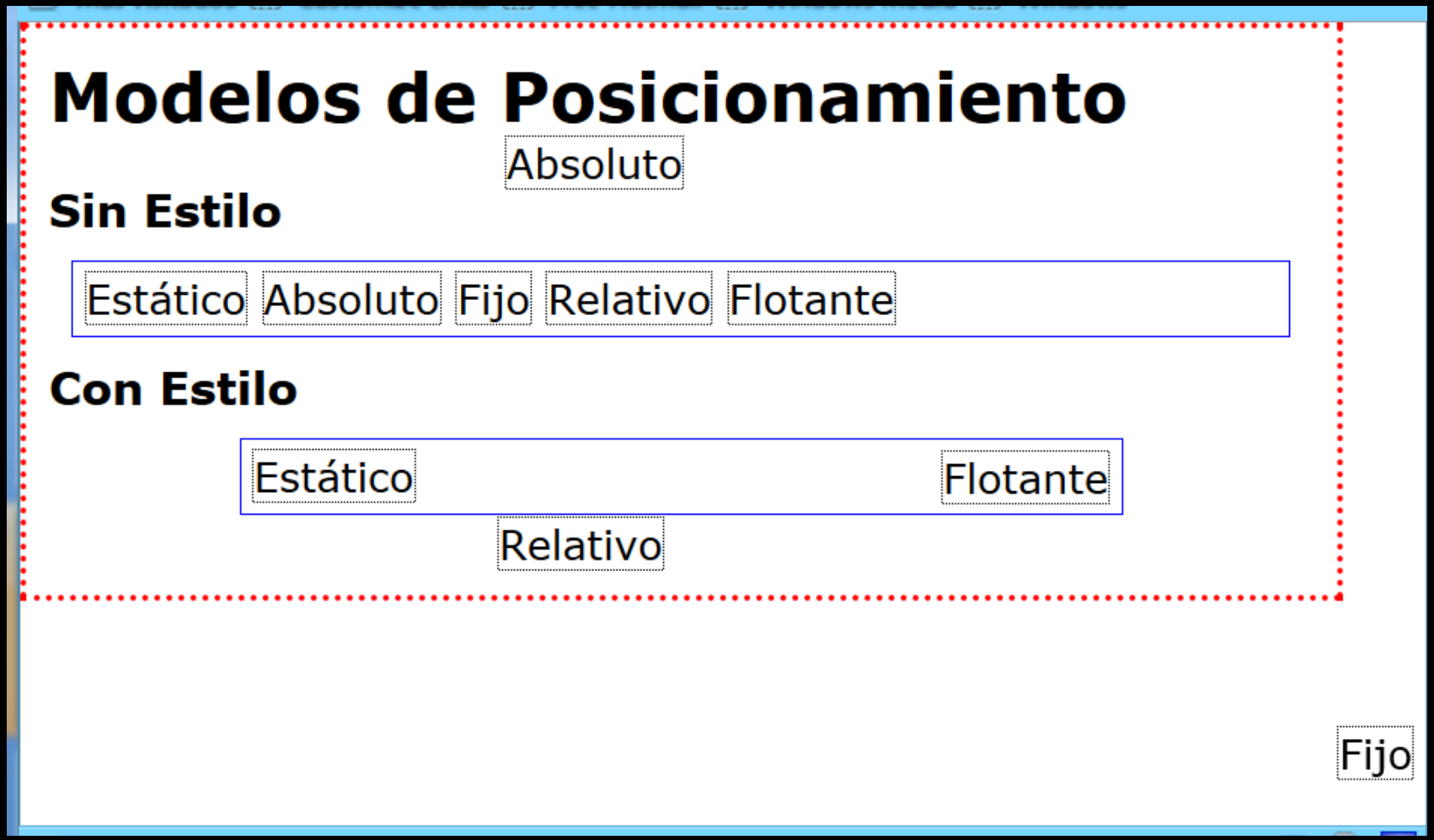
HTML

```
<body>
 <h1>Modelos de Posicionamiento</h1>
 <div class="section"><h2>Sin Estilo</h2>
  <p><span>Estático</span>      <span>Absoluto</span>
     <span>Fijo</span>       <span>Relativo</span>
     <span>Flotante</span>
  </p>
 </div>

 <div class="section"><h2>Con Estilo</h2>
  <p       class="static centered" >
     <span class="static"          >Estático</span>
     <span class="absolute"        >Absoluto</span>
     <span class="fixed"           >Fijo</span>
     <span class="relative"        >Relativo</span>
     <span class="float"           >Flotante</span>
  </p>
 </div>
</body>
```

CSS

```
.centered { width:380px; margin-left:auto; margin-right:auto; margin-bottom:35px;}
.static   { position:static;                                                      }
.absolute { position:absolute;  top:50px;     left:215px;     }
.fixed    { position:fixed;     bottom:20px;  right:5px;      }
.relative { position:relative;  top:30px;     left:30px;      }
.float    { float:right;                                      }
```

CSS. *Posicionamiento*.



CSS. *Display*.

Menús desplegables.- se consiguen combinando las propiedades de display, pasando de “none” a alguna de las opciones que lo hacen visible.

```
li ul {
        display: none;
}

li:hover ul {
        display: block;
}
```

```
<ul>
   <li>Menú
      <ul>
            <li><a href=".....">Enlace 1</a></li>
            <li><a href=".....">Enlace 2</a></li>
            <li><a href=".....">Enlace 3</a></li>
      </ul>
   </li>
</ul>
```

CSS3. *Display*.

Display Grid.- facilita el posicionamiento mediante la creación de "rejillas de cajas".

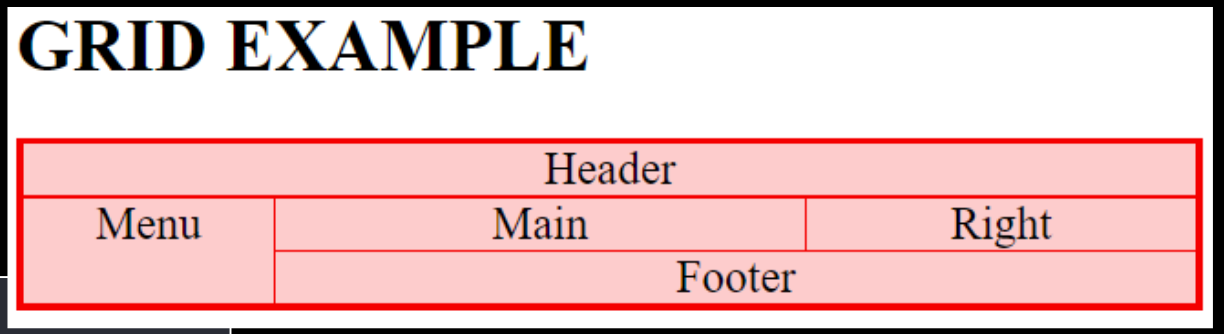
display: grid (inline-grid) → crea una rejilla de cajas de tamaño fijo (mínimo) que se adaptan mediante:

grid-template-colums: ……..…..;
grid-template-rows: ……..;
grid-template-areas: '……';
grid-area: …….;
grid-column-start:…….;
grid-column-end: …….;
grid-row-start: …….;
grid-row-end: …….;
grid-columna-gap: …….;
grid-row-gap: …….;
grid-gap: …….;

CSS3. *Display*.

```html
<body>

<h1>GRID EXAMPLE</h1>

<div class="grid-container">
  <div class="item1">Header</div>
  <div class="item2">Menu</div>
  <div class="item3">Main</div>
  <div class="item4">Right</div>
  <div class="item5">Footer</div>
</div>

</body>
```

```css
.item1 { grid-area: header; }
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }

.grid-container {
  display: grid;
  grid-template-areas:
    'header header header header header header'
    'menu main main main right right'
    'menu footer footer footer footer footer';
  grid-gap: 1px;
  background-color: red;
  padding: 3px;
}
.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  font-size: 20px;
}
```



USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces

CSS3. *Display*.

Display Flex.- facilita el posicionamiento mediante la creación de “rejillas de cajas” que se adaptan a diferentes formas de colocación: horizontal (row), vertical(columna), ancho *viewport* (wrap).

```
display:  flex;
          flex-wrap: row;
                     column;
                     wrap;
                     nowrap;
                     wrap-reverse;
                     otros;
```

y de justificado (*justify-content*), alineación (*align-ítems*)

CSS3 Diseño adaptativo (*responsive*)

Forma de construir una página web, adaptada a diferentes tipos de dispositivos.

A nivel de web se consigue mediante etiquetas HTML y CSS exclusivamente.

Se utilizarán valores relativos para fijar las dimensiones de las cajas.

viewport → área de una página web visible para el usuario.

HTML5 permite controlar el *viewport* mediante el uso de etiquetas *<meta>:*

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

width= device-width → fija el ancho de la página según el dispositivo con el que se abra.

initial-scale= 1.0 → fija el nivel de zoom inicial.

CSS3 Diseño adaptativo (*responsive*)

Reglas *@media* (CSS3).- permiten definir bloques de reglas según dispositivos.

```
@media not|only mediatype and (mediafeature and|or|notmediafeature) {
    CSS-Code;
}
```

- not.- revierte el significado de lo que viene a continuación.
- only.- previene a los navegadores que no soportan reglas @media. No tiene efecto en navegadores modernos.
- and.- permite combinar tipos (mediatype) y características (mediafeature) de los dispositivos.

CSS3 Diseño adaptativo (*responsive*)

Reglas *@media* (CSS3).- permiten definir bloques de reglas según dispositivos.

Las características de los dispositivos que se pueden chequear, son:

- width y height del viewport.
- width y height del dispositivo.
- Orientacion (landscape o portrait).
- Resolution.

```
@media only screen and (max-width: 1200px) {
    body {
        background-color: blue;
    }
}
```

CSS3 Frameworks

Bibliotecas de reglas CSS *propiedad:valor* predefinidas.

Ventajas:

- Facilitan compatibilidad entre navegadores y responsibidad entre dispositivos.
- Simplifican el desarrollo de una aplicación web.
- Garantizan cierto grado de fiabilidad y eficacia.

Inconvenientes:

- Se importa código innecesario, incrementando ancho de banda.
- Se pierde cierto control sobre lo que se está haciendo.
- Se limitan las posibilidades de elección del diseño.

W3.CSS

W3.CSS es un framework de hojas de estilo, de uso libre. Permite además la construcción de diseños “responsivos”.

Llamada:

```
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
```

O bajándola localmente y llamándola mediante:

```
<link rel="stylesheet" href="w3.css">
```

Incluye un conjunto de *plantillas*, pensadas para desarrollo de aplicaciones web en diferentes escenarios.

W3.CSS

Modo de funcionamiento:

Utilizar reglas asociadas a clases predefinidas que afectan a diferentes propiedades. Dado que se definen mediante “*class*”, es posible combinar varias de ellas en una misma regla.

```
<div class="w3-container w3-blue">
  <p>Caja Uno</p>
</div>
```

Nota.- los nombres de las clases predefinidas, empiezan siempre por w3-

W3.CSS

Clases predefinidas:

w3-container.- es la clase principal. Fija características de las cajas relativas a:

- Márgenes
- Rellenos
- Alineamientos horizontales y verticales
- Fuentes
- Colores

```
<div class="w3-container">
  <p>Desarrollo de Aplicaciones Web.</p>
</div>
```

W3.CSS

Otra clases predefinidas:

- w3-color.- proporciona colores típicamente usados en diseño.
- w3-panel.- pensado para visualizar diferentes tipos de notas de notas.
- w3-panel.- utilizado para visualizar alertas.
- w3-card.- pensado para uso en notas y texto de imágenes.
- w3-table.- utilizado para construir tablas.
- w3-ul.- utilizado para construir listas.
- w3-button y w3-btn.- utilizado para incluir botones.
- w3-tag y w3-badge.- utilizados para incluir etiquetas e “insignias”.
- w3-display.- permite colocar elementos HTML en posiciones específicas.
- w3-modal.- permite crear cajas “popup”.
- w3-tooltip y w3-text.- muestran contenido al hacer hoover sobre una etiqueta HTML.
- etc….

W3.CSS

W3.CSS cuanta con clases específicas para realizar diseños responsivos:

- w3-half.- ocupa media ventana/pantalla
- w3-third.- ocupa un tercio de ventana/pantalla
- w3-quarter.- ocupa un cuarto de ventana/pantalla
- w3-rest.- ocupa el resto de columnas.
- w3-col.- define una columna en un diseño a 12 columnas.
- etc….

Desarrollo de Aplicaciones Web

CSS (*Cascading Style Sheet*)

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Pablo García Tahoces