FrontEndI

Módulo 3. Estructuración avanzada

Sesión 11. Cajas flexibles

Flexbox utiliza una estructura de filas y columnas. Es soportado por las versiones comerciales de todos los navegadores web desde 2015.

Un contenedor Flex posee dos ejes: el eje principal (main axis) y el eje transversal (cross axis).

La propiedad Flex-direction permite definir el main axis del contenedor:

flex-direction: row; // Los items se disponen en el eje x, de izquierda a derecha. Es el valor por defecto.

Flex-direction: row-reverse; // Los items se disponen en el eje x, de derecha a izquierda.

Flex-direction: column; // Los items se disponen en el eje y, de arriba a abajo.

Flex-direction: column-reverse; // Los items se disponen en el eje y, de abajo a arriba.

Para alinear los elementos a través del main axis se usa la propiedad justify-content, y para alinearlos a través del cross axis se utiliza align-items:

justify-content: flex-start; // Valor por defecto. Los items se ubican al inicio del main axis.

justify-content: flex-end; // Los items se ubican al final del main axis.

justify-content: center; // Los items se alinean en el centro del main axis.

justify-content: space-between; // Los items se distribuyen de manera uniforme.

justify-content: space-around; // Los items se distribuyen de manera uniforme, dejando un margen al inicio y un margen al final.

align-items: stretch; // Valor por defecto. Los items ocupan todo el espacio en el cross axis.

align-items: flex-start; //Los items se alinean al inicio del cross axis.

align-items: flex-end; //Los items se alinean al final del cross axis.

align-items: center; //Los items se alinean al centro del cross axis.

En casos de contenedores de una sola línea se establece la propiedad flex-flow con el valor no-wrap y se emplea la propiedad align-items. En el caso de contenedores multilínea se establece la propiedad flex-flow con los valores wrap o wrap-reverse y se utiliza align-content. Align-content admite los siguientes valores:

align-content: flex-start;

align-content: flex-end;

align-content: center;

align-content: stretch;

align-content: space-between;

align-content: space-around;

Estructura básica de Flexbox

Flexbox propone una estructura basada en el uso de un contenedor padre (flex-container) y sus elementos hijos (flex-items).

```
.contenedor-padre {
        display: flex;  // También puede adoptar el valor inline-flex.
        flex-wrap: wrap; ; // Para que el flex-container respete el ancho de los flex-items.
}
```

Items

order: 1; // Controla el orden en el que aparecen los items, independiente del orden establecido en el HTML. El valor por defecto es cero.

Flex-grow: 0,75; // Establece cuánto puede crecer un elemento si tiene espacio libre.

align-self: flex-end / flex-start, etc // Permite alinear el item sobre el cross axis, independiente del valor establecido por align-items.

Sincrónico

justify-content: Cómo se alinean los elementos en el eje principal

Box-sizing: border-box; // Generalmente se le aplica a todo el documento.

Flex-box soluciona problemas del position. Normalmente no se requieren usar en forma conjunta.
Es mala práctica usar las dos si se quiere usar solo una de ellas; solamente se aplicaría la que está más abajo.

Sticky y fixed tienen problemas de compatibilidad. Se recomienda que en proyectos grandes no se use, aunque normalmente se presentan más probelmas con fixed.

Recomendación

Se recomienda usar el normalize? No lo recomienda.
Pone en blanco la página; sin márgenes o paddings predeterminados, etc.

Una forma de 'normalizar' (revisar):

```
* {
	margin: 0;
	padding: 0;
	border: 0;
	font-size: 100%;
	font: inherit;
	vertical-align: baseline;
}
```

Para que las medidas relativas funcionen mejor, lo ideal es trabajar siempre con contenedores.
Recomendación: Utilizar medidas relativas, especialmente porcentajes.

Absolute se usa muy poco en pocos elementos; no se debe plantear el layout de todo un sitio con absolute.

Flex-box (revisar):

display: flex;

Casos:
justify-content: flexend;
justify-content: center;
justify-content: space-around;
justify-content: space-between;

align-items: flex-end; // Ubica en el eje secundario

Flex-box es ideal para armar layout
Flex-box es responsive.

Ej: para centrar un elemento, se combinan las dos propiedades, align-items y justify-content.

Ejercicios css:
https://mastery.games/flexboxzombies/
https://flexboxfroggy.com/#es
http://www.flexboxdefense.com/
https://flukeout.github.io/

Sesión 14. Diseño adaptativo.

Julio 1 de 2021.

Viewport

Etiqueta viewport:

<meta name="viewport" content="width=device-width, initial-scale=1">

Medidas relativas

Son aquellas que tienen en cuenta el contexto en el que se encuentran. Si el contexto cambia, estas medidas cambiarán con él.

- Porcentajes: relacionada con la medida del elemento padre en el mismo eje. No se recomienda usar porcentajes para el ancho de un elemento.

- rem y em. em toma como referencia la propiedad font-size del elemento padre. rem se calcula siempre con base en el font-size del elemento de referencia inicial, que es el <html>, cuyo font-size es 16px. Se recomienda usar rem en lugar de em.

- vw y vh. Medidas expresadas como porcentaje de las dimensiones del viewport.

Media Queries

Son un conjunto de reglas de CSS que permiten cambiar los estilos de los elementos en función de las características del dispositivo que esté visualizando el sitio.

```
@media (min-width: 460px){

body {
	background: red;
}

}
```

Orientación:

```
@media (max-width: 460px) and (orientation: landscape){
body {
```

```
        background: blue;
}
}
```

Estrategia de diseño: Mobile first. Recomendado.
Determinar de manera general las reglas CSS para pequeñas pantallas para luego, a través de media queries, ir aclarando el comportamiento en viewports más grandes.

```
body {
        background: red;
}

@media (min-width: 460px){
/* Tablets */
}

@media (min-width: 768px){
/* Laptop */
}
```

Breakpoints: Puntos de quiebre a partir de los cuales cambian las reglas de estilo.
Los más utilizados:

0 – 480px
481 – 768px
769 – 1279px
1280 -

Sincrónico

Analizar página de Nike.

Alinear botones a la izquierda para facilitar el uso con el pulgar

Para que no se vea alguna de las imágenes:
display: none
visibility:
width: 0%

Figma:

https://www.figma.com/file/wSyAvMlFnSM7eE3fufsm3I/Clase-14?node-id=0%3A1

Código de referencia:
https://github.com/juan351/heroes

Sesión 15

Sincrónico

Pensar primero en los teléfonos móviles: “Mobile first”.

Ver la página de Puma.
Se utiliza JavaScript para mostrar un número alto de elementos en una galería.
Algunos de los elementos quedan extra-canvas.

Flex-wrap: Permite que los elementos fluyen a la siguiente línea. (Revisar).

Figma:

https://www.figma.com/file/31NtnGFVE8XyUbfA8Esktw/Petshop?node-id=0%3A1

Pendiente:
- Completar clase 9 (PetShop)
- Completar Clase 11 (Pizza)
- Clase 12 (PetShop, flex)
- Clase 14 (Héroes, v2, flex, responsive)
- Clase 15 (Petshop, responsive)

visibility: hidden; // Oculta el elemento, pero sigue ocupando espacio
display: none; // Oculta el elemento y deja de ocupar espacio

Github Agustina:
https://github.com/agustinagarciarey/TpPetShop

Sesión 16. Formularios

Formulario en HTML

Etiquetas de uso común:

<form action=”/colors” method=”GET”></form>

action: Define la ruta en la que se va a procesar la información

method: Define cómo se envía la información. GET / POST.

<label for=”nombre”>: Texto que acompaña a cada campo del formulario. El atributo for permite asociar al label con un campo por su propiedad id.

<input type=”text”>

Valores de type: text, password, email, tel.

<textarea></textarea>: Campo de múltiples líneas.

<select></select>: Lista desplegable o ComboBox

```
<select name=”pais”>
        <option selected>Argentina</option>
        <option>Colombia</option>
        <option>Brasil</option>
</select>
```

<button type=”submit” value=”Enviar”></button>

Valores de submit: submit / reset / button. El type=”button” no realiza ninguna acción por defecto.

Otros atributos:

Name: Todo elemento debe tener este atributo. Los elementos que no tienen name no se envían (?)
Value: Cuando no permiten inserción de texto por el usuario
required:
placeholder:

Radio button y Checkboxes

Radio Button

```
<label>
	<input type=”radio” name=”forma-de-pago” value=”efectivo”>Pago en efectivo
</label>

<label>
	<input type=”radio” name=”forma-de-pago” value=”crédito” checked>Pago con tarjeta de crédito
</label>
```

El atributo name es lo que agrupa los elementos, haciendo que solo se pueda escoger una de las opciones.
Al incluir el input dentro del label, el usuario puede dar click en cualquier lugar del label para axctivar el control.
El checked (opcional) preselecciona el valor.

Checkbox

```
<label>
	<input type=”checkbox” name=”autorizacion” value=”js”>JavacScript
</label>

<label>
	<input type=”checkbox” name=”autorizacion” value=”html”>HTML
</label>
```

En este caso, el name permite guardar todos los value en una variable.

Otro caso: Preguntas Sí/No.

```
<label for=”terminos”>Acepto los términos.</label>
<input type=”checkbox” name=”terminos”>
```

En este caso no se requiere el atributo value.

Formulario avanzados

Input date

```
<input type=”date” name=”fechaNacimiento” value=”1985-08-28”>
<input type=”date” name=”fechaNacimiento” value=”1985-08-28” min=”1980-01-01” max=”2021-07-04”>
```

<input type=”file” name=”foto” accept=”.jpg, .png” multiple>

Validación de formularios

Agregar una etiqueta :

```
<div>
<label>Nombre Completo:</label>
        <input type=”text” name=”nombreUsuario”>
        <span class=”feedback”></span>
<div>
```

Formularios accesibles

* Situar correctamente las etiquetas label

- Antes de los campos para text, textarea, select, password, file.
(Arriba o a la izquierda del campo).

- Detrás del campo para los tipo checkbox y radio button

* Etiquetar los controles de forma correcta

```
<label for=”fname”>Nombre</label>
<input id=”fname” type=”text” name=”nombre”>
```

El atribute for debe ser igual al atribute id del input.

Label no debe usarse para los campos:
- Image (usar el atributo alt)
- Tipos submit y reset (usar value)
- Para los button en los que su contenido se usa como una etiqueta.

Etiquetas de agrupamiento

<fieldset> Agrupa semánticamente un número de controles del formulario
<legend> Agrega una descripción al filedset
<optgroup label=”nombreCategoria”></optgroup> Permite agrupar una serie de option dentro de un select

Navegación a través del formulario:
Atributo tabindex indica la posición del elemento en la navegación por tabulación. Si se establece un valor negativo, el elemento es ignorado.

Sincrónico

Sesiones se manejan desde el back porque responde a una necesidad de seguridad.

Cuando haya muchos campos de formulario, incluir botones de avance.

Filtrar, ordenar, etc, son funciones que cumple el back en los proyectos grandes, mientras que en proyectos pequeños y sin requerimientos especiales de seguridad, lo puede hacer el FrontEnd.

<form action=””>

Los labels son importantes

Etiqueta outgroup: Item que no puede seleccionar

Dentro de un formulario se recomienda usar las etiquetas <input type=”reset”> y <input type=”submit”> en lugar de la etiqueta <button>, ya que aquellas son específicas para formularios.

El atributo “name” es requerido en los formularios.

Para tener un formulario con varias páginas:
- Se puede utilizar el mismo div o el mismo form y ocultar todos los controles para mostrar otros.

Figma:
https://www.figma.com/file/7qh4iPVe98d7DfchQbu5PF/Ecommerce?node-id=1%3A2