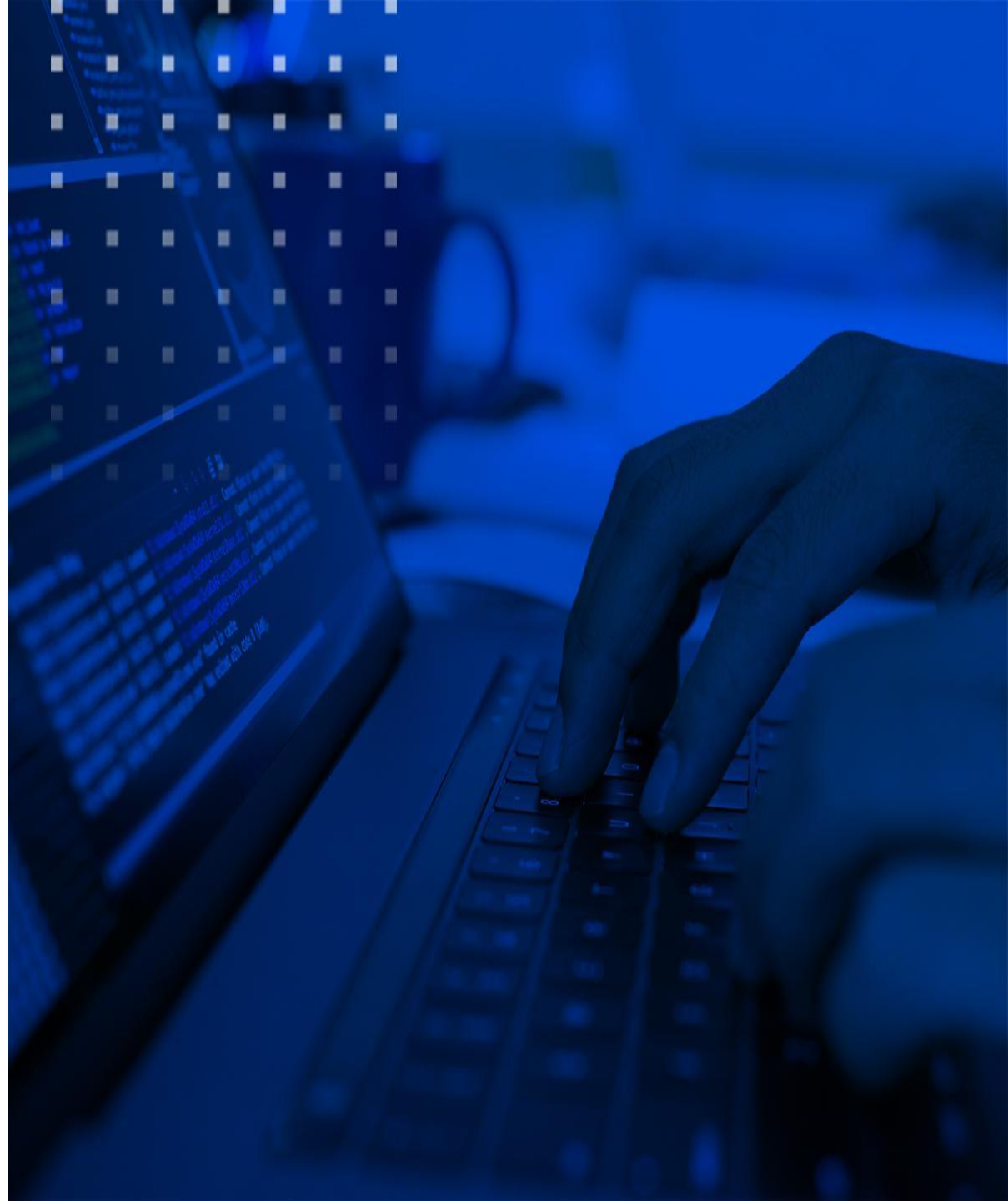


CURSO

FULL STACK DEVELOPER NIVEL INICIAL

UNIDAD 3:
FLEXBOX: El modelo de caja flexible



Enfoque tradicional

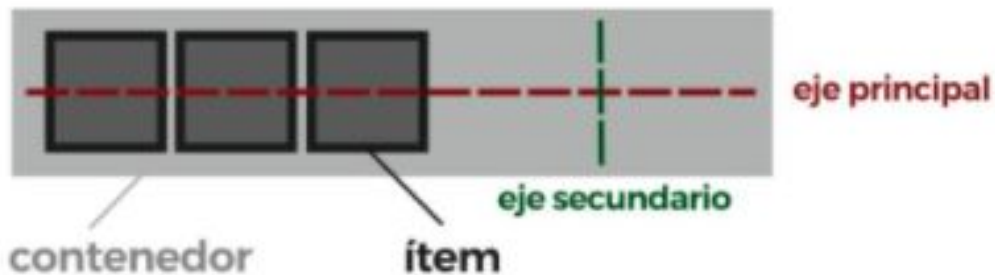
En CSS se ha utilizado el posicionamiento (static, relative, absolute...) o los float, lo que a grandes rasgos ya no encaja con los retos que tenemos hoy en día: sistemas de escritorio, dispositivos móviles, múltiples resoluciones, etc...

Flexbox

Flexbox es un sistema de elementos flexibles que aparece con la idea de reemplazar estos mecanismos y acostumbrarnos a una mecánica más potente, limpia y personalizable, en la que los elementos HTML se adaptan y colocan automáticamente haciendo más fácil personalizar los diseños.

Conceptos

Para empezar a utilizar flexbox debemos conocer algunos de los elementos básicos de este esquema.



Contenedor: Es el elemento padre que tendrá en su interior cada uno de los ítems flexibles y adaptables.

Ítem: Cada uno de los hijos flexibles que tendrá el contenedor en su interior.

Eje principal: La orientación por defecto, es en horizontal (row).

Eje secundario: La orientación secundaria es perpendicular a la principal.

Layouts usando flexbox

Para comenzar a usar **flexbox** debemos asignar el valor **flex** a la propiedad **display** del div que queremos que se comporte como el contenedor



```
#div-1 {
  background-color
  : width: 200px;
  height: 200px;
}
```

```
#div-2 {
  background-color
  : width: 150px;
  height: 150px;
}
```

```
#div-3 {
  background-color:
  #00bd9a; width: 200px;
  height: 200px;
}
```

```
#contenedor {
  display: flex;
  flex-direction: row;
  justify-content:
  space-around;
  align-items: center;

  background-color:
  #f1e7b6; width:
  100%;
  height: 300px;
}
```

Propiedades de flexbox: Contenedor

El modelo flexbox cuenta con varias propiedades que nos permitirán modificar la forma en la que se muestran los ítems dentro del contenedor.

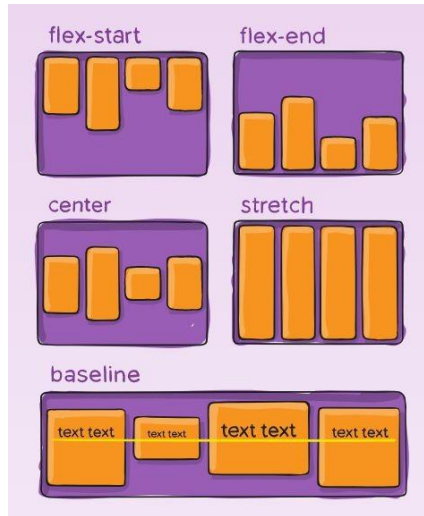
Flex-direction: Establece la orientación del eje principal (row, column, row-reverse, column-reverse).

Flex-wrap: Por defecto los items intentarán entrar en una sola línea. Esta propiedad permite el desbordamiento (nowrap, wrap, wrap-reverse).

Flex-flow: Combina las propiedades anteriores en una sola. Se especifica primero la dirección y luego si se permite o no el desbordamiento (ej: row wrap).

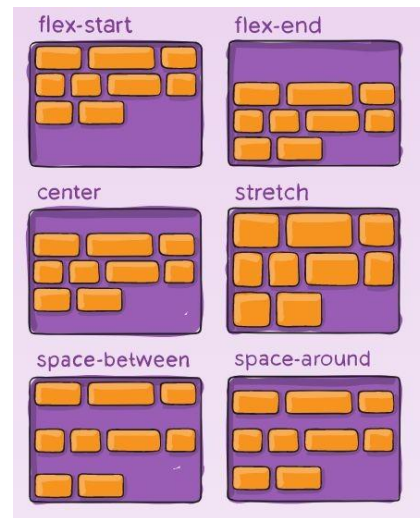
Align-items:

Establece la alineación de los items en el eje secundario (stretch por defecto).

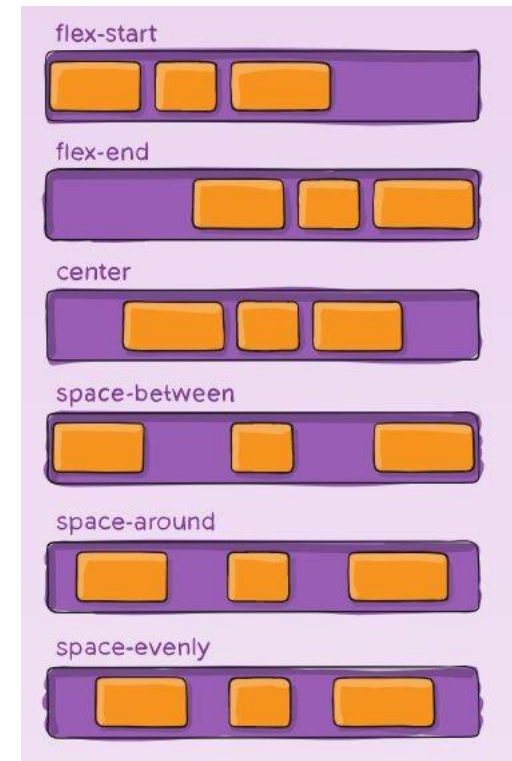


Align-content:

Establece la alineación del conjunto de items en el eje secundario.



Justify-content: Establece la alineación de los items en el eje principal (flex-start por defecto).



Propiedades de flexbox: Items

Order: Establece el orden en el que se muestran los ítems (0 por defecto).

Flex-grow: Permite a un ítem crecer o expandirse de ser necesario (0 por defecto).

Flex-shrink: Inversa a la propiedad anterior, permite a un ítem encogerse (1 por defecto).

Flex-basis: Define el tamaño por defecto de los ítems. Puede ser una longitud (ej: 25px) o una palabra clave (ej: auto) (auto por defecto).

Align-self: Define la alineación de un ítem en particular en el eje secundario (mismos valores que align-items y align-content).

Ejercicio 1

Realizar una página que contenga:

- Dos o más divs.
- A cada uno darle las propiedades vistas (borde, padding, margen, tamaño) y contenido (títulos, párrafo, imagen).
- Probar cómo se modifica la apariencia cambiando el tamaño, el padding, márgenes y bordes.
- Agregar div con tamaño en porcentaje, ver qué sucede cuando achicamos la ventana del navegador.

Ejercicio 2

Realizar una página que contenga:

- Dos o más divs, uno contenedor y dentro sus hijos.
- Aplicar las propiedades vistas de flexbox.
- Probar cómo se modifica la apariencia cambiando la dirección, justificación, alineación, etc.
- Agregar div con tamaño en porcentaje, ver qué sucede cuando achicamos la ventana del navegador.

Ejercicio 3

- Modificar el ejemplo para que la página quede con un modelo como el siguiente.
- Experimenten distintos valores de las propiedades de las cajas, de las propiedades de flexbox. Es importante que comprendan cómo funcionan.



Hacer un layout completo

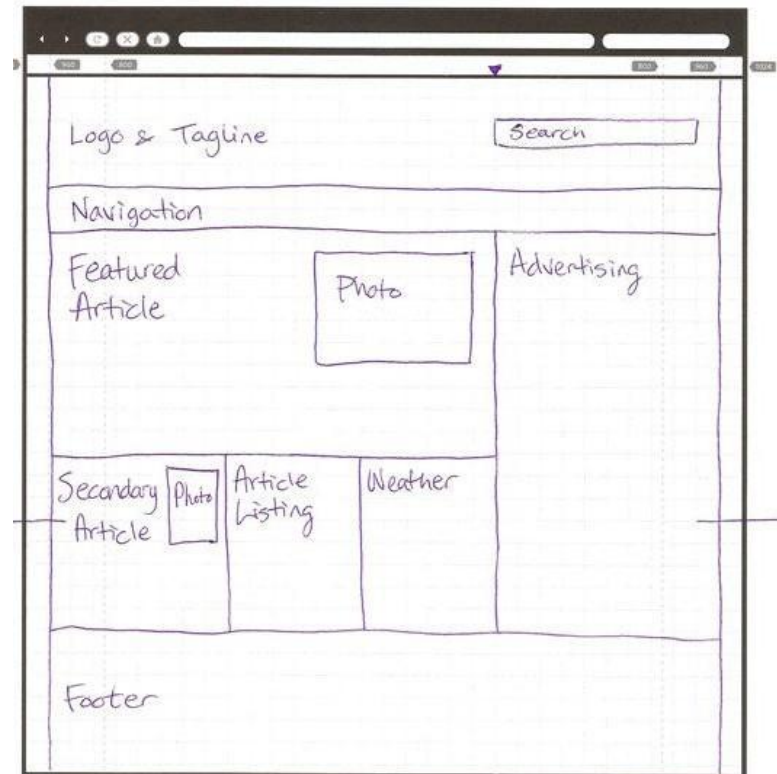
Actividad Final de módulo HTML y CSS

Debe contener: una página principal, menú, footer y al menos dos secciones. Aplicar los conceptos de HTML y CSS vistos en clase y el material de lectura. Pueden utilizar como base los ejercicios anteriores.

Sugerencias

Hacer un diagrama del layout en papel, lo más completo posible y con sus medidas. (wireframe)

Un vez que se tiene una idea clara del diseño que se desea lograr, comenzar a escribir código para ajustarlo al diseño.



Referencias



HTML & CSS - Design and Build Websites.

JON DUCKETT

CSS - the missing manual. DAVID SAWYER MCFARLAND



Unidades en CSS <https://www.w3.org/Style/Examples/007/units.en.html>

Flexbox <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

