MAKE it REAL
.camp

classroom session

MAKE it REAL .camp

classroom questions

MAKE it REAL .camp

FlexBox

MAKE it REAL .camp

- *Las cajas flexibles, o flexbox, es un nuevo modo de layouts en CSS3.*

- *El uso de flexbox asegura que los elementos se comporten de manera predecible cuando el diseño de la página debe acomodarse a diferentes tamaños de pantalla y a diferentes dispositivos de visualización.*

- *Para muchas aplicaciones, el modelo de caja flexible proporciona una mejora sobre el modelo de bloque en el sentido de que no utiliza floats, ni tampoco los márgenes del contenedor flexible se colapsan con los márgenes de su contenido.*

FlexBox

MAKE it REAL .camp

- *Flexbox se compone de flex containers y flex items.*

- *Se declara un flex container configurando la propiedad display de un elemento flex (renderizado como un bloque) o un inline-flex (renderizado como en línea).*

- *Dentro de un flex container hay uno o más flex items.*

- *Nota: todo lo que está fuera de un flex container y dentro de un flex item se procesa como de costumbre. Flexbox define cómo se colocan los elementos flexibles dentro de un flex container*

FlexBox

MAKE it REAL .camp

- *Los flex items se colocan dentro de un flex container a lo largo de un flex line. Por defecto, sólo hay un flex line por flex container.*

- *El siguiente ejemplo muestra tres flex items. Están posicionados de forma predeterminada: de forma horizontal en flex line, de izquierda a derecha*

FlexBox

MAKE it REAL .camp

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
    display: -webkit-flex;
    display: flex;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}

.flex-item {
    background-color: cornflowerblue;
    width: 100px;
    height: 100px;
    margin: 10px;
}
</style>
</head>
<body>

<div class="flex-container">
  <div class="flex-item">flex item 1</div>
  <div class="flex-item">flex item 2</div>
  <div class="flex-item">flex item 3</div>
</div>

</body>
</html>
```

FlexBox

MAKE it REAL .camp

- *También es posible cambiar la dirección del flex line.*

- *Si ajustamos la propiedad de direction a rtl (de derecha a izquierda), el texto se dibuja de derecha a izquierda, y también la flex line cambia de dirección, lo que cambiará el diseño de página*

FlexBox

MAKE it REAL .camp

```
body {
    direction: rtl;
}

.flex-container {
    display: -webkit-flex;
    display: flex;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}

.flex-item {
    background-color: cornflowerblue;
    width: 100px;
    height: 100px;
    margin: 10px;
}
```

FlexBox

MAKE it REAL .camp

Flex Direction

- *La propiedad flex-direction especifica la dirección de los flex-items dentro del flex-container. El valor predeterminado de flex-direction es row (de izquierda a derecha, de arriba a abajo).*

- *Los otros valores son los siguientes:*
  - *row-reverse - Si el modo de escritura (dirección) se deja a la derecha, los elementos flexibles se colocarán de derecha a izquierda*
  - *column - Si el sistema de escritura es horizontal, los artículos de flexión serán dispuestos verticalmente*
  - *column-reverse - Igual que la columna, pero invertida*

FlexBox

MAKE it REAL .camp

row-reverse

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-direction: row-reverse;
    flex-direction: row-reverse;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

column

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-direction: column;
    flex-direction: column;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

column-reverse

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-direction: column-reverse;
    flex-direction: column-reverse;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

justify-content

- *La propiedad justify-content alinea horizontalmente los elementos del flex-container cuando los elementos no utilizan todo el espacio disponible en el eje principal.*

- *Los valores posibles son los siguientes:*
  - *flex-start - Valor predeterminado. Los items se colocan al principio del contenedor*
  - *flex-end - Los items se colocan en el extremo del contenedor*
  - *center - Los items se colocan en el centro del contenedor*
  - *space-between - Los items se colocan con espacio entre las líneas*
  - *space-around - Los items se colocan con espacio antes, entre y después de las líneas*

FlexBox

MAKE it REAL .camp

flex-end

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-justify-content: flex-end;
    justify-content: flex-end;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

center

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-justify-content: center;
    justify-content: center;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

space-between

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-justify-content: space-between;
    justify-content: space-between;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

space-around

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-justify-content: space-around;
    justify-content: space-around;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

align-items

- *La propiedad align-items alinea verticalmente los items del flex-container cuando los items no utilizan todo el espacio disponible en el eje transversal.*

- *Los valores posibles son los siguientes:*
  - *stretch - Valor predeterminado. Los items se estiran para adaptarse al contenedor*
  - *flex-start - Los items se colocan en la parte superior del contenedor*
  - *flex-end - Los items se colocan en la parte inferior del contenedor*
  - *center - Los items se colocan en el centro del contenedor (verticalmente)*
  - *baseline - Los items se sitúan en la línea de base del contenedor*

FlexBox

MAKE it REAL .camp

stretch

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: stretch;
    align-items: stretch;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

flex-start

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: flex-start;
    align-items: flex-start;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

flex-end

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: flex-end;
    align-items: flex-end;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

center

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: center;
    align-items: center;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

baseline

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-align-items: baseline;
    align-items: baseline;
    width: 400px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

flex-wrap

- *La propiedad flex-wrap especifica si los flex-items se deben envolver o no, si no hay suficiente espacio para ellos en una flex-line*

- *Los valores posibles son los siguientes:*
  - *nowrap - Valor predeterminado. Los flex-items no se envolverán*
  - *wrap - Los flex-items se envolverán si es necesario*
  - *wrap-reverse - Los flex-items se envolverán, si es necesario, en orden inverso*

FlexBox

MAKE it REAL .camp

nowrap

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-wrap: nowrap;
    flex-wrap: nowrap;
    width: 300px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

wrap

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-wrap: wrap;
    flex-wrap: wrap;
    width: 300px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

wrap-reverse

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-wrap: wrap-reverse;
    flex-wrap: wrap-reverse;
    width: 300px;
    height: 250px;
    background-color: lightgrey;
}
```

FlexBox

MAKE it REAL .camp

align-content

- *La propiedad align-content modifica el comportamiento de la propiedad flex-wrap. Es similar a align-items, pero en lugar de alinear items flexibles, alinea líneas flexibles.*

- *Los valores posibles son los siguientes:*
  - *stretch - Valor predeterminado. Las líneas se extienden para ocupar el espacio restante*
  - *flex-start - Las líneas están empacadas hacia el comienzo del flex-container*
  - *flex-end - Las líneas se empaquetan hacia el extremo del flex-container*
  - *center - Las líneas están empaquetadas hacia el centro del flex-container*
  - *space-between - Las líneas están uniformemente distribuidas en el flex-container*
  - *space-around - Las líneas están distribuidas uniformemente en el flex-container, con espacios de tamaño medio en cada extremo*

FlexBox

MAKE it REAL .camp

center

```
.flex-container {
    display: -webkit-flex;
    display: flex;
    -webkit-flex-wrap: wrap;
    flex-wrap: wrap;
    -webkit-align-content: center;
    align-content: center;
    width: 300px;
    height: 300px;
    background-color: lightgrey;
}
```

Propiedades de flex-item

MAKE it REAL .camp

order

La propiedad order especifica el orden de un flex-item con relación al resto de elementos flexibles dentro del mismo contenedor:.

```
.flex-item {
    background-color: cornflowerblue;
    width: 100px;
    height: 100px;
    margin: 10px;
}

.first {
    -webkit-order: -1;
    order: -1;
}
```

Propiedades de flex-item

MAKE it REAL .camp

order

La propiedad order especifica el orden de un flex-item con relación al resto de elementos flexibles dentro del mismo contenedor:.

```
.flex-item {
    background-color: cornflowerblue;
    width: 100px;
    height: 100px;
    margin: 10px;
}

.first {
    -webkit-order: -1;
    order: -1;
}
```

Propiedades de flex-item

MAKE it REAL .camp

margin

cambiar a margin: auto; absorberá el espacio extra. Se puede utilizar para empujar flex-items en diferentes posiciones.

```
.flex-item {
    background-color: cornflowerblue;
    width: 75px;
    height: 75px;
    margin: 10px;
}

.flex-item:first-child {
    margin-right: auto;
}
```

Media Queries

MAKE it REAL .camp

CSS2

- *La regla @media, introducida en CSS2, permitió definir diferentes reglas de estilo para diferentes media types.*

- *Ejemplos: Puede tener un conjunto de reglas de estilo para pantallas de computadora, una para impresoras, una para dispositivos portátiles, otra para dispositivos de tipo televisión, etc.*

- *Desafortunadamente estos media types nunca recibieron mucha compatibilidad/soporte con estos dispositivos, aparte del media type de impresión*

Media Queries

MAKE it REAL .camp

CSS3 - Introduce media queries

- *Las media queries en CSS3 extienden la idea de los media types de CSS2: Pero en lugar de buscar un media type, buscan la capacidad del dispositivo.*
- *Las media queries se pueden utilizar para comprobar muchas cosas, tales como:*
  - *Anchura y altura de la ventana de visualización (viewport)*
  - *Anchura y altura del dispositivo*
  - *Orientación (esta la tableta / teléfono en modo horizontal o vertical?)*
  - *Resolución de la pantalla*
- *El uso de media queries es una técnica popular para entregar una hoja de estilo adaptada a tabletas, iPhone y Androids*

Media Queries

MAKE it REAL .camp

Sintaxis

- *Una media query consta de un media type y puede contener una o más expresiones, que se devuelven a true o false.*

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

Media Queries

MAKE it REAL .camp

Sintaxis

- *El resultado de una query (consulta) es true si el media type especificado coincide con el tipo de dispositivo en el que se muestra el documento y todas las expresiones en la consulta de medios (media query) son verdaderas. Cuando una consulta de medios es verdadera, se aplican las reglas de estilo o de hoja de estilo correspondientes, siguiendo las reglas normales de CSS.*
- *A menos que utilice los operadores not o only, el media type es opcional y todo el type estará implícito.*
- *También puede tener diferentes hojas de estilo para diferentes soportes:*

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)"
href="print.css">
```

Media Queries

MAKE it REAL .camp

Media Types

Value	Description
all	Used for all media type devices
print	Used for printers
screen	Used for computer screens, tablets, smart-phones etc.
speech	Used for screenreaders that "reads" the page out loud

Media Queries

MAKE it REAL .camp

Ejemplos sencillos

- *Una forma de utilizar media queries es tener una sección CSS alternativa dentro de su hoja de estilo.*
- *El siguiente ejemplo cambia el color de fondo a lightgreen si la ventana es de 480 píxeles de ancho o más (si la vista es menor de 480 píxeles, el color de fondo será el que tenga por defecto en body):*

```
@media screen and (min-width: 480px) {
    body {
        background-color: lightgreen;
    }
}
```

Media Queries

MAKE it REAL .camp

Refactorizamos

Para definir un **media query** utilizamos `@media` seguido de una expresión que define para qué dispositivos van a aplicar los estilos que estén dentro de ese **media query**.

Por ejemplo, el siguiente **media query** va a aplicar cuando el ancho del navegador supere los `992px`:

```
@media (min-width: 992px) {
  // acá van las reglas que aplican cuando el ancho es de más de 992px
}
```

Media Queries

MAKE it REAL .camp

Refactorizamos

Por ejemplo, si queremos que el fondo de nuestro documento sea amarillo en anchos menores a `992px` y rojo en anchos iguales o mayores podemos hacer lo siguiente:

```
// por defecto va a ser amarillo
body {
  background-color: yellow;
}

@media (min-width: 992px) {
  // esto solo aplica si el ancho es de más de 992px
  body {
    background-color: red;
  }
}
```

Media Queries

MAKE it REAL .camp

Mobile First

```
// mobile first
body {
  font-size: 14px;
}

// tabletas
@media (min-width: 768px) {
  body {
    font-size: 15px;
  }
}

// escritorio
@media (min-width: 992px) {
  body {
    font-size: 16px;
  }
}

// pantallas grandes
@media (min-width: 1200px) {
  body {
    font-size: 17px;
  }
}
```

Responsive Design

MAKE it REAL .camp

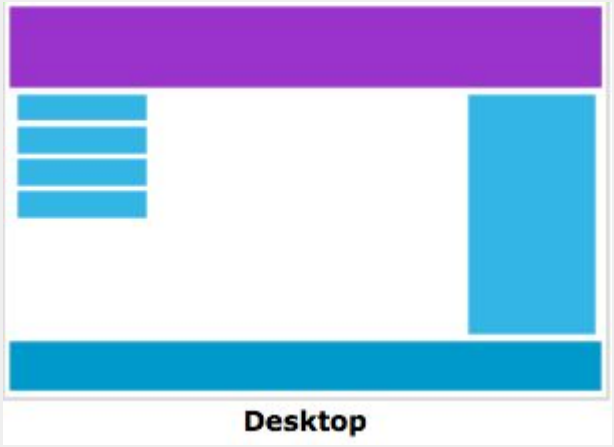
Intro

- *El diseño web responsable utiliza sólo HTML y CSS.*
- *Responsive web design no es un programa o un JavaScript.*
- *El diseño web Responsive hace que su página web se vea bien en todos los dispositivos.*
- *Las páginas web se pueden ver utilizando muchos dispositivos diferentes: escritorio, tabletas y teléfonos. Su página web debe verse bien, y ser fácil de usar, independientemente del dispositivo.*
- *Las páginas web no deben omitir la información para adaptarse a dispositivos más pequeños, sino adaptar su contenido a cualquier dispositivo:*

Responsive Design

MAKE it REAL .camp

Intro



Responsive Design

MAKE it REAL .camp

El viewport

- *El viewport es el área visible del usuario de una página web.*
- *El viewport varía con el dispositivo y será menor en un teléfono móvil que en una pantalla de computadora.*
- *Antes de las tabletas y teléfonos móviles, las páginas web estaban diseñadas sólo para pantallas de ordenador y era común que las páginas web tuvieran un diseño estático y un tamaño fijo.*
- *Entonces, cuando empezamos a navegar por Internet usando tabletas y teléfonos móviles, las páginas web de tamaño fijo eran demasiado grandes para caber en el viewport. Para solucionar esto, los navegadores de esos dispositivos redujeron la página web en su totalidad para ajustarse a la pantalla.*
- *Esto no fue perfecto! Pero si fue una solución rápida.*

Responsive Design

MAKE it REAL .camp

Configuración del viewport

- *HTML5 introdujo un método para permitir que los diseñadores de páginas web tomen el control sobre la ventana gráfica, a través de la etiqueta <meta>.*
- *Debe incluir el siguiente elemento viewport <meta> en todas sus páginas web:*

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

Responsive Design

MAKE it REAL .camp

Configuración del viewport

- *El elemento <meta> viewport le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.*
- *El width=device-width define el ancho de la página para seguir el ancho de pantalla del dispositivo (que variará dependiendo del dispositivo).*
- *El initial-scale=1.0 establece el nivel de zoom inicial cuando la página es cargada por primera vez por el navegador.*

Responsive Design

MAKE it REAL .camp

Configuración del viewport

Without the viewport meta tag

With the viewport meta tag

Responsive Design

MAKE it REAL .camp

Grid-view

- *Muchas páginas web se basan en una vista de cuadrícula, lo que significa que la página se divide en columnas:*

Responsive Design

MAKE it REAL .camp

Grid-view

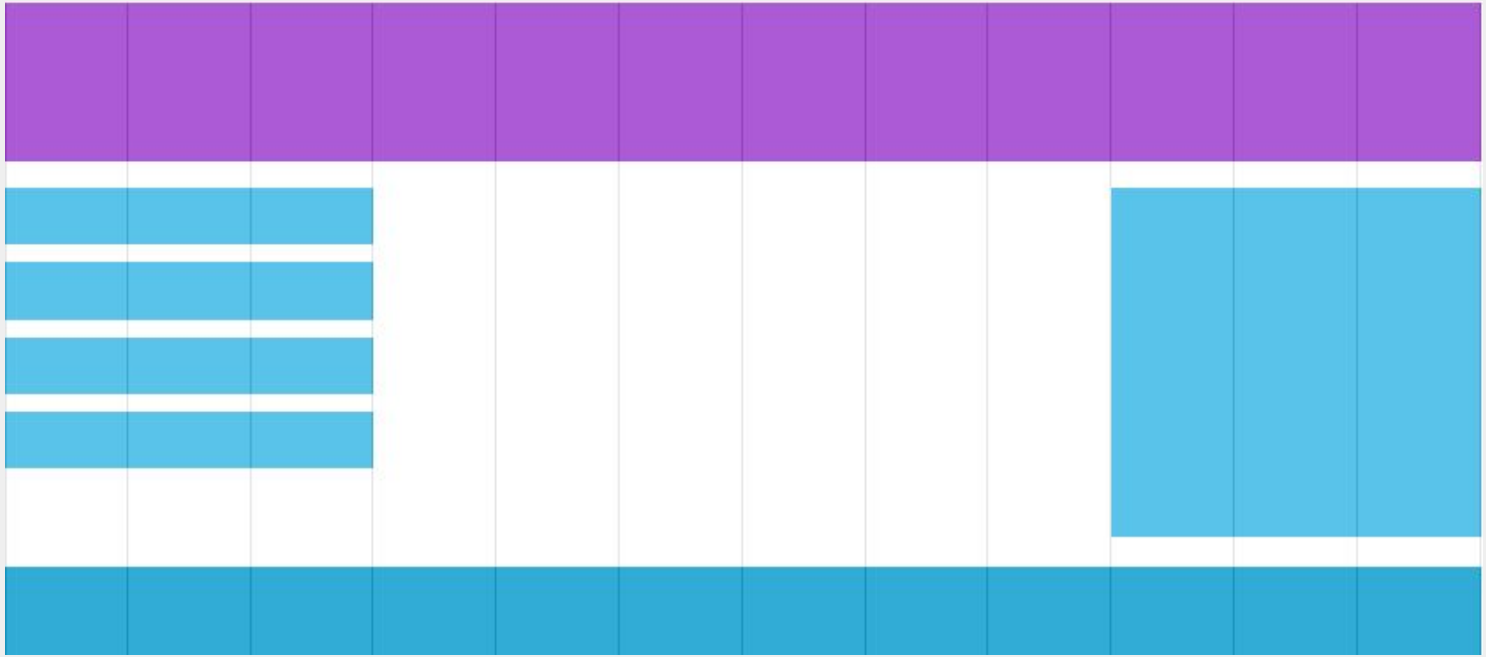
- *El uso de una vista de cuadrícula es muy útil al diseñar páginas web. Hace que sea más fácil colocar elementos en la página.*

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

Responsive Design

MAKE it REAL .camp

Grid-view



classroom rating