

No aporta nada interesante

USING

Flexbox

TODAY

I work for
Booking.com

Psst...you can too: www.workingatbooking.com

My portfolio site from 2000

The image shows a screenshot of a website with a dark teal header and a large green circular graphic in the center. The header contains the word "hello" in orange. Below the header is a rectangular box containing text. To the right of the box is a vertical list of links: "me", "gallery", "résumé", and "links". A mouse cursor is hovering over the "gallery" link. At the bottom of the page is a "home" link. The green circle contains the text "a collection of my web design projects and visual artwork".

Welcome to my little niche of the world wide web. Why have I chosen to add to the clutter of the information superhighway even more? It's more than just self-indulgence, though I am hopelessly addicted to html and Photoshop. But besides having fun with web design, it is also the field I have worked in for some time now.

To view my online work, as well as examples of my visual artwork, click on [gallery](#). For a more detailed description of my skills, experience, and goals, click on [résumé](#). To learn more about me personally, click on [me](#) to the right. And finally, what self-respecting home page would be complete without a list of [links](#) for you to peruse?

[home](#)

[email](#)

me
gallery
résumé
links

a collection of my web design projects and visual artwork

My portfolio site from 2000

The image shows a screenshot of a website from around 2000. The layout features a dark teal header and footer, and a white central content area.

Header: The word "hello" is displayed in orange text at the top left of the main content area.

Content Area: A large text block on the left side reads:

Welcome to my little niche of the world wide web. Why have I chosen to add to the clutter of the information superhighway even more? It's more than just self-indulgence, though I am hopelessly addicted to html and Photoshop. But besides having fun with web design, it is also the field I have worked in for some time now.

To view my online work, as well as examples of my visual artwork, click on [gallery](#). For a more detailed description of my skills, experience, and goals, click on [résumé](#). To learn more about me personally, click on [me](#) to the right. And finally, what self-respecting home page would be complete without a list of [links](#) for you to peruse?

Footer: The word "home" is displayed in orange text at the bottom center of the main content area.

Grid Menu: To the right of the main content area is a grid-based navigation menu. The columns are labeled "email" (top), "me" (second), "gallery" (third), "résumé" (fourth), and "links" (fifth). A mouse cursor is hovering over the "gallery" link. A large green circle highlights the "gallery" column, with the text "a collection of my web design projects and visual artwork" written inside it.

floats

flexbox

regions

shapes

tables

inline-block

exclusions

grid

multi-column

table-cell

positioning

flexbox

when

what

how

1. WHEN

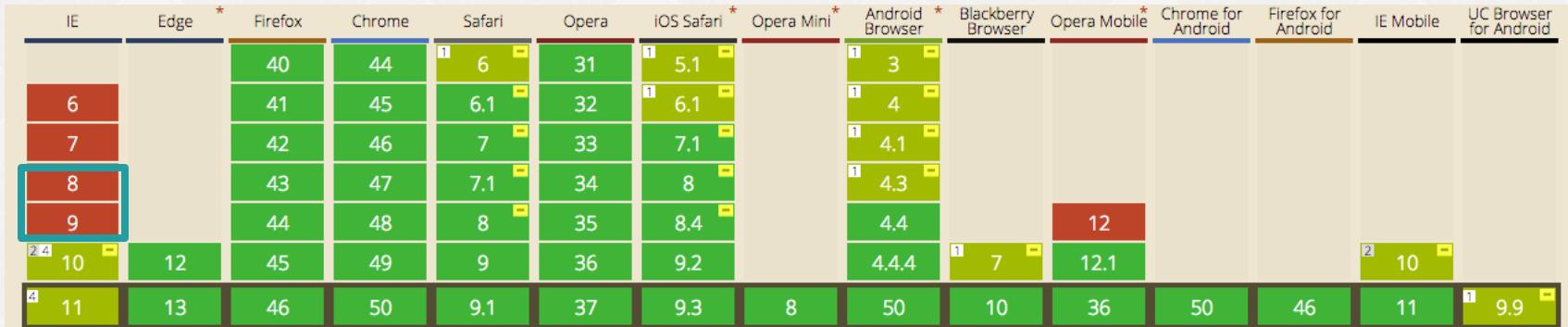
Deciding when to use and not use flexbox



What browsers do I need to support?

Don't ask yourself this—it's irrelevant here
(IMO)

Flexbox has 97% coverage worldwide



We support IE 8 and 9 at Booking, but still use flexbox as progressive enhancement.

Do I want to create a layout in
1 dimension (row OR column)
or 2 dimensions?

Hat-tip to Rachel Andrew:

<https://rachelandrew.co.uk/archives/2016/03/30/should-i-use-grid-or-flexbox/>

Flexbox is not a grid

No esta destinado para grandes diseños o todo el diseño de la distribución de la pagina

- Not meant for or great at whole page layout

Elemento flex solo se preocupan sobre el espacio dentro de las filas o columnas
- Flex items only care about space in their own **row/column**

A ellos no le importa sobre alinearse con otra dimension
- They don't care about **lining up** in the other dimension

Formación de fila
Alinearse
Fila

Card 1

Posuere varius ullamcorper ipsum adipiscing dignissim ipsum adipiscing a a quisque malesuada quam purus venenatis sagittis fermentum parturient curabitur montes a metus.

Card 2

Posuere varius ullamcorper ipsum adipiscing dignissim ipsum adipiscing a a quisque malesuada quam purus venenatis sagittis fermentum parturient curabitur montes a metus.

Card 3

Posuere varius ullamcorper ipsum adipiscing dignissim ipsum adipiscing a a quisque malesuada quam purus venenatis sagittis fermentum parturient curabitur montes a metus.

Card 4

Posuere varius ullamcorper ipsum adipiscing dignissim ipsum adipiscing a a quisque malesuada quam purus venenatis sagittis fermentum parturient curabitur montes a metus.

Card 5

Posuere varius ullamcorper ipsum adipiscing dignissim ipsum adipiscing a a quisque malesuada quam purus venenatis sagittis fermentum parturient curabitur montes a metus.

Demo by Rachel Andrew: <http://codepen.io/rachelandrew/pen/YqqdXL>

Flexbox is best for:

Define diseño visual , colores , diseño grafico , layout , tipografia

- **UI components**

Todo/Entero

diseño,distribución

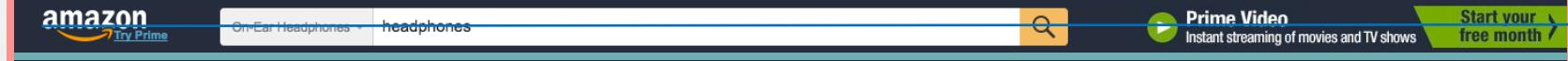
- **Simple whole page layouts (not grid-based)**

Mejorando

una completa alineación de la distribución , plano , diseño

- **Enhancing a complex layout's alignment, spacing, etc. (not controlling placement)**

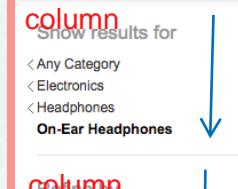
Colocacion , posicionamiento

row 

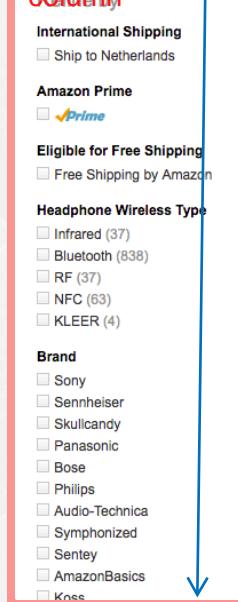
row 

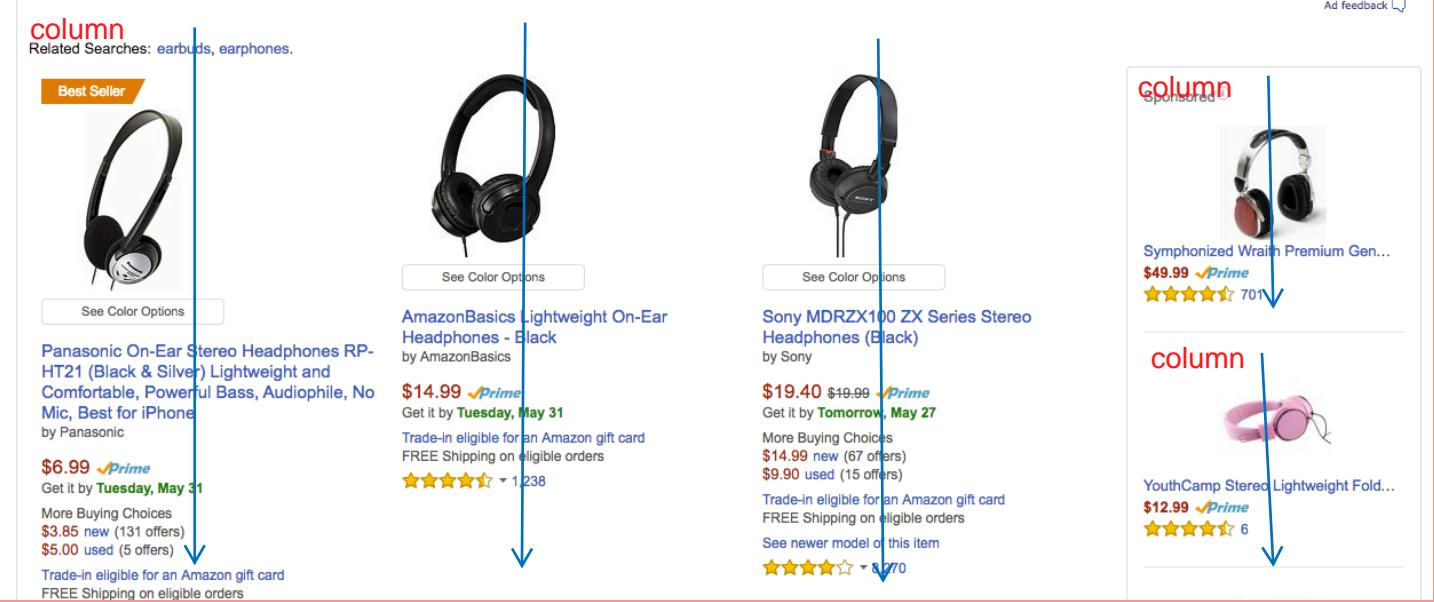
row 

row 

column 

column 

column 

column 

Do I need my content to
dictate sizing and placement,
or do I need to control these?

mandar ordenar

ubicar, posicionar

Flexbox

Content determines
boxes' size and
placement

(Mega-useful when
content is **unknown** and
variable, or **readability** is
^{legibilidad}
a top priority.)

Grids

Structure determines
content's size and
placement

(P.S. **Flexbox can do this too**, if you want. It's just the reverse that doesn't work so well.)

Es sólo que lo contrario no funciona muy bien.

ofrecerme

Does flexbox offer me anything I can't already get from an existing layout method?

diseño/maquetacion metodos

¿Me ofrece Flexbox algo que no pueda obtener de un método de diseño existente?

New things flexbox offers

Basado en el contenido

Significa que te enfocas primero en el material que estás tratando de mostrar y luego colocas elementos a su alrededor

- Content-driven, **unit-less sizes** unidad sin tamaños
- Content-driven, media-query-less layout changes menos cambios en el diseño/distribuciones de los media queries
- Mixed-unit layouts mezclando unit en el diseño/distribución
- Equal-height columns igualdad en el alto de la columnas
centrado vertical y otros alineamientos
- Vertical centering and other alignments Separación o estiramiento de elementos a llenar sin conocer el alto y ancho
- Spacing out or stretching items to fill unknown width/height envolviendo contenido combinado y envolviendo bloques
- Combining content wrapping and block wrapping fijacion de los elementos sin superposiciones
- Pinning items without overlaps
- Visual order different than HTML/reading order visualizar orden diferente como HTML u orden de lectura

2. WHAT

Components flexbox can enhance and
UI/UX problems it can help you solve

mejorar / aumentar

content-driven, unit-less sizes

Diseño primero los contenidos y luego
los colocas en las estructuras

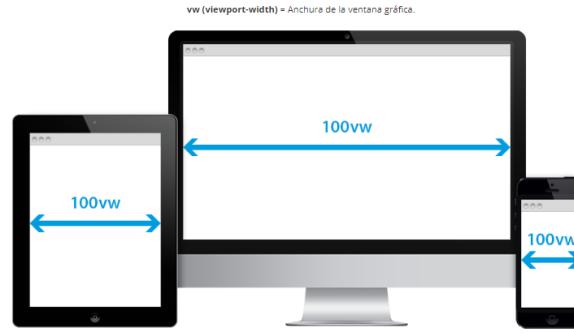
Como de grandes hare estas cosas

How big do I make this thing?

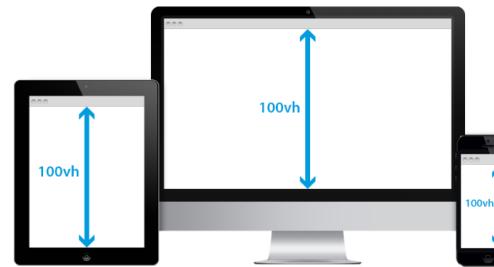
Las "unidades de longitud porcentuales de ventana de visualización" (viewport-percentage lengths) o más comúnmente llamadas viewport units son unidades CSS relativas a la ventana gráfica del navegador o visualizador.

El marco de referencia en anchura/altura de nuestro navegador donde se visualiza la web se establece desde el W3C como "Bloque inicial contenedor", y su medida nos permite definir las dimensiones porcentuales de anchura/altura de todos los elementos que contiene.

Este bloque inicial o ventana gráfica puede considerarse como la capa padre de toda nuestra web, y las unidades "vh" y "vw" son las unidades en altura y anchura que hacen referencia a esta ventana



vh (viewport-height) = Altura de la ventana gráfica.



¿Qué diferencias hay realmente entre vh/vw y %?

vh (Viewport Height)

Medición basada en la altura de la pantalla. Por ejemplo si damos una altura a un elemento de `height: 100vh;`, este elemento tendrá la misma altura que la pantalla.

vw (Viewport Width)

Medición basada en lo ancho de la pantalla. Por ejemplo si damos un ancho a un elemento de `width: 100vw;`, este elemento tendrá lo mismo ancho que lo ancho que la pantalla.

% (Porcentaje)

Medición basado en porcentaje. La medición en porcentaje será definida en relación a su elemento padre, a menos que éste se acompañe de una posición fija (`position:fixed;`) o absoluta (`position:absolute;`), que entonces en este caso su comportamiento ahora también respetará el tamaño de la pantalla.

em/rem
vw/vh
procedimiento
calc()

Ocupa todo el alto y ancho del viewport del dispositivo donde se haya establecido

Ejemplo con misma cantidad con las dos unidades: 60vw vs 60%

vw/vh: Esta unidad depende del tamaño del viewport, que es la ventana de visualización de donde se ve la página.

Ejemplo: Si mi navegador lo tengo abierto con un tamaño de 800px de ancho, 100vw serían 800px.

Porcentaje %: Esta unidad es el porcentaje de tamaño del contenedor padre o relativo a este.

Ejemplo: Tengo un de 1000px de ancho que contiene a adentro con `width: 20%`, este medirá 200px.

Te dejo un codepen para que veas la diferencia jugando con el tamaño de la pantalla: <https://codepen.io/bitxelus/pen/MvJGpZ>

aquellas cortará ello

When none of those will cut it,
go unit-free **with** flexbox.

Significa que te enfocas primero en el material que estás tratando de mostrar y luego colocas elementos a su alrededor

Content-driven sizing on Booking.com

Utilizar flexbox para una caja que almacene un modulo como por ejemplo el de fechas

Search

Destination/Hotel Name:
Chicago

Work Leisure ?

Check-in Date
Mon 2 November 2015

Check-out Date
Tue 3 November 2015

1-night stay

Guests More options

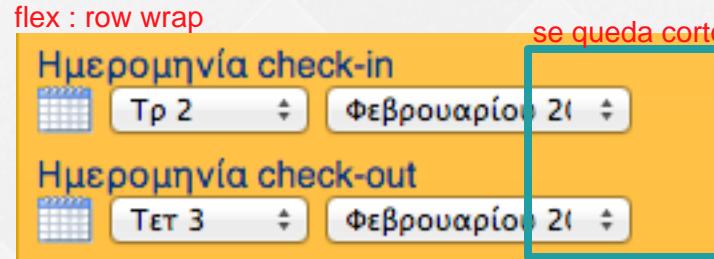
Rooms 1

Adults 3

Children 0

Search

Last year's sidebar searchbox design, with fixed-width select fields



Content-driven sizing on Booking.com

Non-flexbox

Ημερομηνία check-in



Ημερομηνία check-out



Flexbox enhanced

Utilizar flexbox para una caja que almacene un módulo como por ejemplo el de fechas

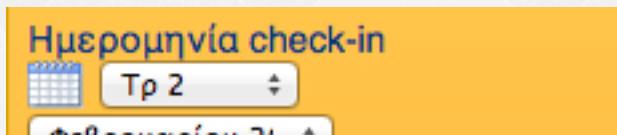
Ημερομηνία check-in



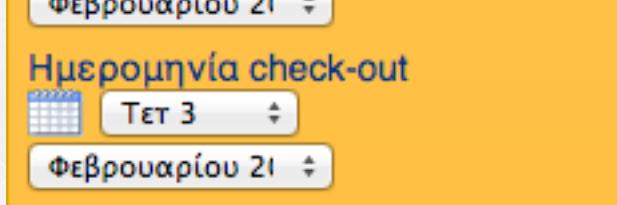
Ημερομηνία check-out



Ημερομηνία check-in



Ημερομηνία check-out



Ημερομηνία check-in



Ημερομηνία check-out



Content-driven sizing on booking.com

```
.sb-dates {  
    display: flex;  
}  
  
.sb-dates__icon {  
    flex: 0 0 23px;  
}  
  
.sb-dates__select-day {  
    flex: 1 0 auto;  
    margin: 0 6px;  
}  
  
.sb-dates__select-month {  
    flex: 1 1 auto;  
    flex-grow flex-shrink flex-basis  
}
```



flex-grow : 0 por defecto
flex-shrink : 1 por defecto
flex-basis : auto | number

Translating the flex property

```
.sb-dates {  
  display: flex;  
}  
  
.sb-dates__icon {  
  flex: 0 0 23px;  
}  
  
.sb-dates__select-day {  
  flex: 1 0 auto;  
  margin: 0 6px;  
}  
  
.sb-dates__select-month {  
  flex: 1 1 auto;  
}
```

Start out **23px** wide, and **don't grow** or **shrink** further
amplitud

Start out **sized to your content**, then **grow with 1 share** of any extra space available, but **don't ever shrink**

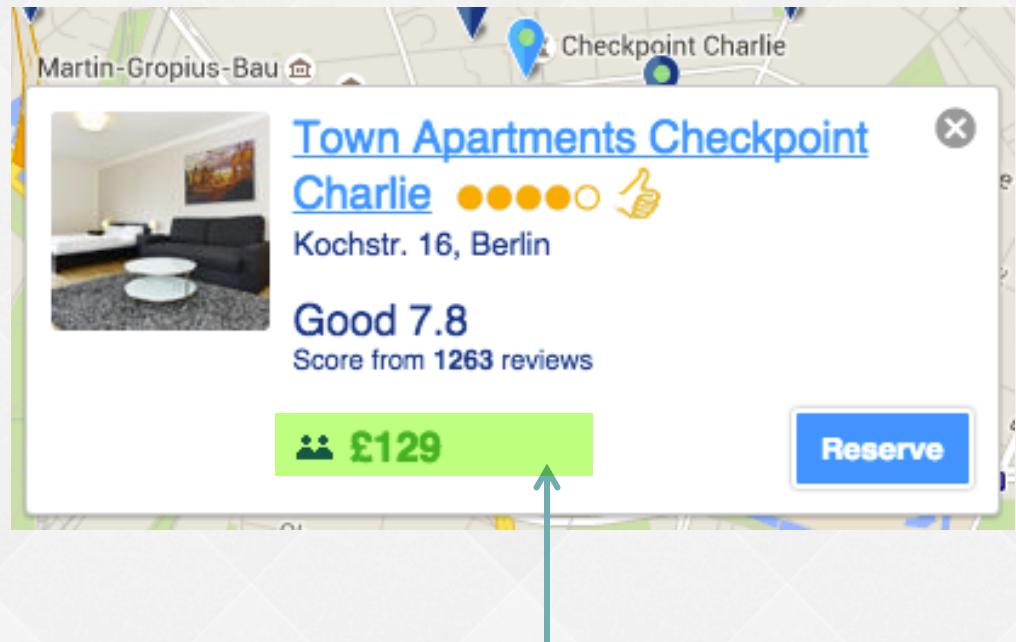
Start out **sized to your content**, then **grow with 1 share** of extra space, but if there's an **overflow shrink**

Mixed-unit layout is easier with
calc(), but ni siquiera not even it can do:

**calc(100% - 23px - the width of
the day field in Greek)**

Taking advantage of variable space

Task: add a message about low availability of the room price shown:
“We have only X left on our site!”



How about right here
in this lovely big gap?

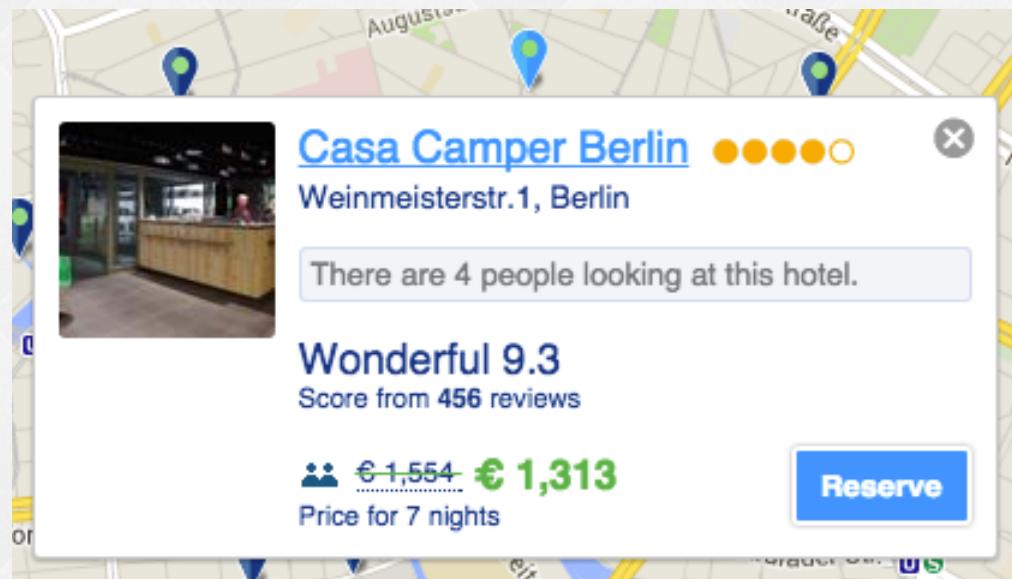
Taking advantage of variable space

Problem: the gap
is not always big
enough to hold a
sentence of text

espacio

suficiente

mantener



106 126 руб.
Цена за 7 ночей

Забронировать

Rp 23.301.844
Rp 19.680.612
Harga untuk 7 malam

Pesan sekarang

€ 1.554 - € 1.313
Harga untuk 7 malam

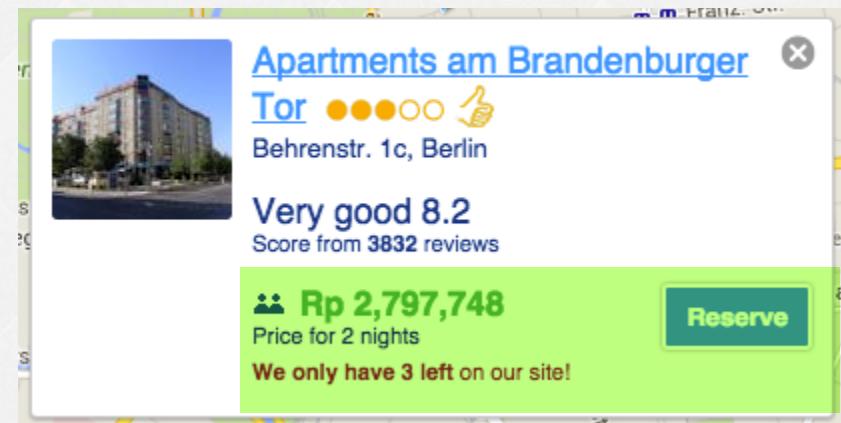
Pesan sekarang

Taking advantage of variable space

Solution: use **flexbox** to place text ^{ubicar texto} beside price ^{junto} when space allows; otherwise, it can wrap ^{sin embargo} below price and stretch ^{junto al precio} ^{estirar}



Not possible with floats, inline-block



Not possible with table-cell

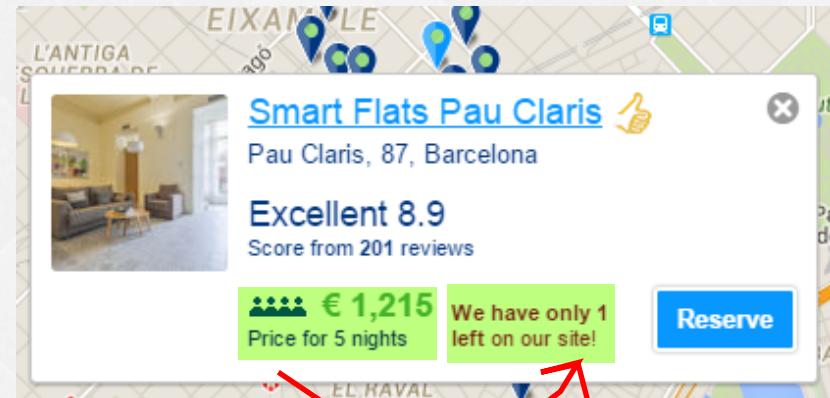
mejora

Progressive enhancement

Non-flexbox



Flexbox enhanced



RWD content-driven layout change

Narrow: 1 column



Another School-free Snow Day
for Hillsborough Kids

Weather

28 January 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed laoreet quam. Etiam vehicula vel enim non commodo. Integer quis eros sed ante semper ultricies id at justo. Integer vel cursus dui. Duis vitae rhoncus sem. Donec sed odio sodales, mollis ante id, dapibus dolor. Curabitur feugiat metus vel nisi vestibulum faucibus. Nunc iaculis felis eu arcu malesuada hendrerit. Duis vehicula dictum magna eu hendrerit. Vestibulum mauris nibh, hendrerit eget nisl suscipit, congue gravida urna. Morbi a eros aliquet, dictum sem vel, commodo neque. Phasellus sit amet nulla risus. Curabitur tempor lacinia nibh ac eleifend. Nam egestas et est tincidunt suscipit. Fusce facilisis venenatis erat, vitae varius lectus. Maecenas commodo, urna vitae tempor vulputate, lectus justo malesuada enim, non tincidunt ipsum arcu eget sem.

`display: flex
justify-content : column
align-content : wrap`

Wide: 2 columns



Another School-free Snow Day for
Hillsborough Kids

Weather

28 January 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed laoreet quam. Etiam vehicula vel enim non commodo. Integer quis eros sed ante semper ultricies id at justo. Integer vel cursus dui. Duis vitae rhoncus sem. Donec sed odio sodales, mollis ante id, dapibus dolor. Curabitur feugiat metus vel nisi vestibulum faucibus. Nunc iaculis felis eu arcu malesuada hendrerit. Duis vehicula dictum magna eu hendrerit. Vestibulum mauris nibh, hendrerit eget nisl suscipit, congue gravida urna. Morbi a eros aliquet, dictum sem vel, commodo neque. Phasellus sit amet nulla risus. Curabitur tempor lacinia nibh ac eleifend. Nam egestas et est tincidunt suscipit. Fusce facilisis venenatis erat, vitae varius lectus. Maecenas commodo, urna vitae tempor vulputate, lectus justo malesuada enim, non tincidunt ipsum arcu eget sem.

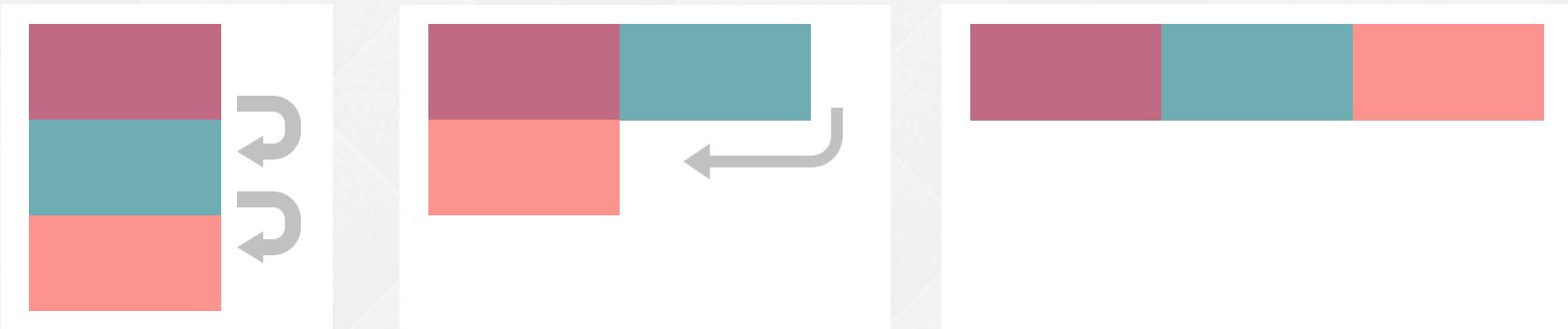
`display: flex
justify-content : row
align-content : wrap`

Layout change without media query

Deje que los bloques se envuelvan y se apilen cuando sea necesario

1. Let the blocks wrap and stack when needed:

```
.article-header {  
    display: flex;  
    flex-direction: row; /* default */  
    flex-wrap: wrap;  
}
```



Comutación

Switching between columns and rows

1 column when narrow

reducir/estrecho



Multiple rows when wide

ancho
extenso/amplio

`flex-direction: column` → `@media min-width...`

or

`flex-direction: row`

`flex-direction: row`

`flex-wrap: wrap`

(or the shorthand

`flex-flow: row wrap`

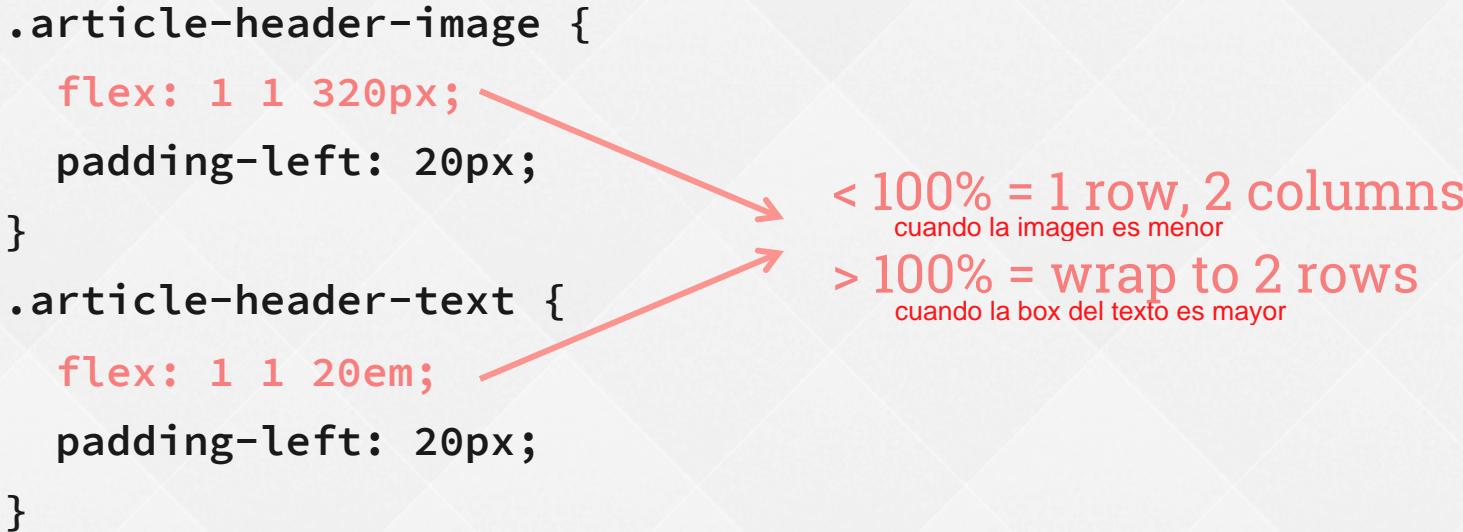


Same, no change needed

Layout change without media query

2. Size the blocks to control their wrapping point:

```
.article-header-image {  
    flex: 1 1 320px;  
    padding-left: 20px;  
}  
  
.article-header-text {  
    flex: 1 1 20em;  
    padding-left: 20px;  
}
```



The diagram consists of two red arrows pointing from specific CSS properties to explanatory text. The top arrow points from the 'flex' property of the '.article-header-image' rule to the text ' $< 100\% = 1 \text{ row}, 2 \text{ columns}$ ' and 'cuando la imagen es menor'. The bottom arrow points from the 'flex' property of the '.article-header-text' rule to the text ' $> 100\% = \text{wrap to } 2 \text{ rows}$ ' and 'cuando la box del texto es mayor'.

$< 100\% = 1 \text{ row}, 2 \text{ columns}$
cuando la imagen es menor

$> 100\% = \text{wrap to } 2 \text{ rows}$
cuando la box del texto es mayor

Stretching to fill unknown width/height



Another School-free Snow Day
for Hillsborough Kids

Weather

28 January 2015

flex: 1 1 auto

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed laoreet quam. Etiam vehicula vel enim non commodo. Integer quis eros sed ante semper ultricies id at justo. Integer vel cursus dui. Duis vitae rhoncus sem. Donec sed odio sodales, mollis ante id, dapibus dolor. Curabitur feugiat metus vel nisi vestibulum faucibus. Nunc iaculis felis eu arcu malesuada hendrerit. Duis vehicula dictum magna eu hendrerit. Vestibulum mauris nibh, hendrerit eget nisl suscipit, congue gravida urna. Morbi a eros aliquet, dictum sem vel, commodo neque. Phasellus sit amet nulla risus. Curabitur tempor lacinia nibh ac eleifend. Nam egestas et est tincidunt suscipit. Fusce facilisis venenatis erat, vitae varius lectus. Maecenas commodo, urna vitae tempor vulputate, lectus justo malesuada enim, non tincidunt ipsum arcu eget sem.



Another School-free Snow Day for
Hillsborough Kids

align-content:
space-between

Weather

28 January 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sed laoreet quam. Etiam vehicula vel enim non commodo. Integer quis eros sed ante semper ultricies id at justo. Integer vel cursus dui. Duis vitae rhoncus sem. Donec sed odio sodales, mollis ante id, dapibus dolor. Curabitur feugiat metus vel nisi vestibulum faucibus. Nunc iaculis felis eu arcu malesuada hendrerit. Duis vehicula dictum magna eu hendrerit. Vestibulum mauris nibh, hendrerit eget nisl suscipit, congue gravida urna. Morbi a eros aliquet, dictum sem vel, commodo neque. Phasellus sit amet nulla risus. Curabitur tempor lacinia nibh ac eleifend. Nam egestas et est tincidunt suscipit. Fusce facilisis venenatis erat, vitae varius lectus. Maecenas commodo, urna vitae tempor vulputate, lectus justo malesuada enim, non tincidunt ipsum arcu eget sem.

Improved wrapping in RWD layout

With float or text-align



Another School-free
Snow Day for
Hillsborough Kids

Hillsborough News

28 January 2015

With flex or justify-content



Another School-free
Snow Day for
Hillsborough Kids

Hillsborough News

28 January 2015

Flexbox is great for spacing
and aligning stuff, especially
shifting content in RWD.

Responsive web design (RWD)
Diseño adaptable

full width
full height
equally spaced

Full-width nav bar with equal spacing

<http://codepen.io/zomigi/pen/vKjbbY/>

Non-flexbox fallback version

2 opcion , retirada , alternativa



→ Pure CSS responsive flexbox nav

→ Resize your browser, both width and height, to see the effect

This page uses **flexbox** to control the layout so that when the off-canvas menu becomes visible on the left side of the screen (medium-width windows), the content area can **resize** to fit the remaining width in the viewport, instead of overflowing the viewport and getting cut off on the right side, as happens with most off-canvas menus.

→ Flexbox also makes it possible for the menu to be **full-width** when it's a horizontal bar at the top of the screen (wide windows), and **full-height** when it's vertical on the side of the screen.

Flexbox version con SASS



→ Pure CSS responsive flexbox nav

→ Resize your browser, both width and height, to see the effect

This page uses **flexbox** to control the layout so that when the off-canvas menu becomes visible on the left side of the screen (medium-width windows), the content area can **resize** to fit the remaining width in the viewport, instead of overflowing the viewport and getting cut off on the right side, as happens with most off-canvas menus.

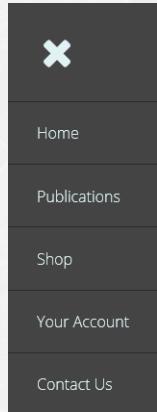
→ Flexbox also makes it possible for the menu to be **full-width** when it's a horizontal bar at the top of the screen (wide windows), and **full-height** when it's vertical on the side of the screen.

Full-width nav bar with equal spacing

```
.menu-list {  
    display: flex;  
    justify-content: space-between; → Equally spaces items  
    margin: 0; across main axis  
    padding: 0; (width in this case)  
    list-style: none;  
    text-align: center; /* fallback */  
}  
  
.menu-list-item {  
    propiedad valor  
    display: inline-block; /* fallback */  
    padding: 0 .5em; /* fallback */  
}
```

alternativa / segunda opcion / plan b

Full-height menu with equal spacing



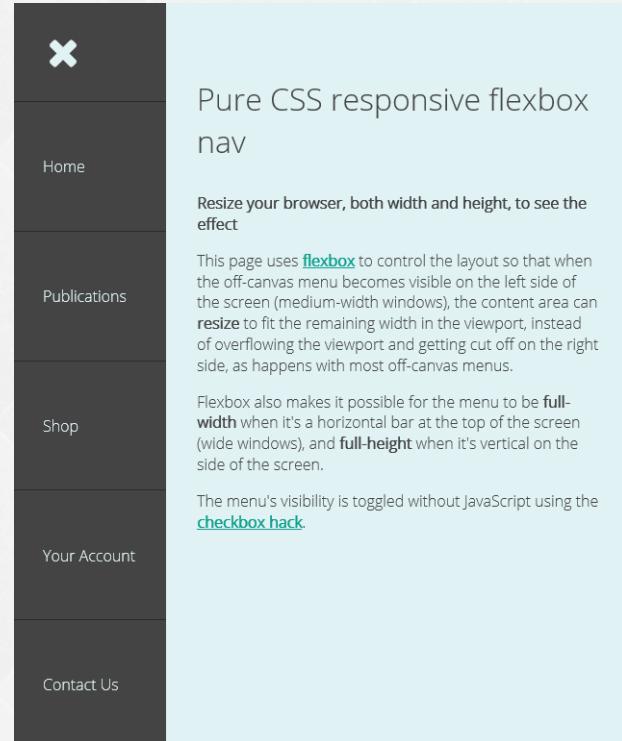
Pure CSS responsive flexbox nav

Resize your browser, both width and height, to see the effect

This page uses [flexbox](#) to control the layout so that when the off-canvas menu becomes visible on the left side of the screen (medium-width windows), the content area can **resize** to fit the remaining width in the viewport, instead of overflowing the viewport and getting cut off on the right side, as happens with most off-canvas menus.

Flexbox also makes it possible for the menu to be **full-width** when it's a horizontal bar at the top of the screen (wide windows), and **full-height** when it's vertical on the side of the screen.

The menu's visibility is toggled without JavaScript using the [checkbox hack](#).



Pure CSS responsive flexbox nav

Resize your browser, both width and height, to see the effect

This page uses [flexbox](#) to control the layout so that when the off-canvas menu becomes visible on the left side of the screen (medium-width windows), the content area can **resize** to fit the remaining width in the viewport, instead of overflowing the viewport and getting cut off on the right side, as happens with most off-canvas menus.

Flexbox also makes it possible for the menu to be **full-width** when it's a horizontal bar at the top of the screen (wide windows), and **full-height** when it's vertical on the side of the screen.

The menu's visibility is toggled without JavaScript using the [checkbox hack](#).

Full-height menu with equal spacing

justify-content version

```
.menu-list {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
}  
  
.menu-list-item {  
  height: 100%;  
  display: flex;  
  align-items: center;  
}
```

flex version

```
.menu-list {  
  display: flex;  
  flex-direction: column;  
}  
  
.menu-list-item {  
  flex: 1 0 auto;  
  display: flex;  
  align-items: center;  
}
```

Full-height menu with equal spacing

justify-content version

```
.menu-list {  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
}  
  
.menu-list-item {  
  height: 100%;  
  display: flex;  
  align-items: center;  
}
```

flex version

```
.menu-list {  
  display: flex;  
  flex-direction: column;  
}  
  
.menu-list-item {  
  flex: 1 0 auto;  
  display: flex;  
  align-items: center;  
}
```

Flexbox can also enhance
visual ordering.

mejorar

order

integer to specify flow order of flex items

default source order



re-ordered



re-ordered



re-ordered



Reorder for good, not evil.

Demo: moving a photo on mobile

S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.



iheartnaptime.com

INGREDIENTS

- 1/2 cup butter melted
- 1 1/2 cup graham cracker crumbs
- 2 cups brown sugar
- 2/3 cup salted butter (softened)
- 2 eggs
- 2 1/2 cup flour
- 2 1/4 tsp baking powder
- 1/2 tsp salt
- 1 tsp vanilla
- 1 cup semi-sweet chocolate chips
- 1 cup mini marshmallows
- 1 Milk Chocolate Candy Bar (broken into pieces)
- 1 Graham Cracker (broken into pieces)
- 1/2 cup marshmallows

S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.



iheartnaptime.com

INGREDIENTS

- 1/2 cup butter melted
- 1 1/2 cup graham cracker crumbs
- 2 cups brown sugar
- 2/3 cup salted butter (softened)
- 2 eggs
- 2 1/2 cup flour
- 2 1/4 tsp baking powder
- 1/2 tsp salt
- 1 tsp vanilla
- 1 cup semi-sweet chocolate chips
- 1 cup mini marshmallows
- 1 Milk Chocolate Candy Bar (broken into pieces)
- 1 Graham Cracker (broken into pieces)
- 1/2 cup marshmallows

INSTRUCTIONS

1. Pre-heat oven to 325 degrees.
2. Combine melted butter and graham cracker crumbs into a small bowl with a fork. Then line a 13x9 with foil and press down crust. Set aside.
3. Cream butter and sugar in a large bowl. Add in eggs and vanilla and stir until combined. Mix the baking powder and salt with 1 cup of flour. Pour into the butter mixture and stir in the rest of the flour. Fold in chocolate chips.
4. Press dough into pan with a spatula flour on your hands. The dough goes right on top of the crust. Bake for 30-35 minutes or until lightly browned. Oven temps will vary. Test with a tooth pick. The dough should be soft in the center.
5. Allow bars to cool and then enjoy with a scoop of ice cream or cold glass of milk!

Demo: moving a photo on mobile

Mobile: reordered



S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.

Desktop: HTML order (no flexbox)

S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.

INGREDIENTS

- 1/2 cup butter melted
- 1 1/2 cup graham cracker crumbs
- 2 cups brown sugar
- 2/3 cup salted butter (softened)
- 2 eggs
- 2 1/2 cup flour
- 2 1/4 tsp baking powder
- 1/2 tsp salt
- 1 tsp vanilla
- 1 cup semi-sweet chocolate chips
- 1 cup mini marshmallows
- 1 Milk Chocolate Candy Bar (broken into pieces)
- 1 Graham Cracker (broken into pieces)
- 1/2 cup marshmallows

INSTRUCTIONS

1. Pre-heat oven to 325 degrees.
2. Combine melted butter and graham cracker crumbs into a small bowl with a fork. Then line a 15x9 with foil and press down crust. Set aside.



ihearthaptme.com

Use flexbox order in mobile styles

```
.recipe {  
    display: flex;  
    flex-direction: column;  
}  
  
.recipe figure {  
    order: -1; /* before all items with default order: 0 */  
}  
  
.recipe figure img {  
    width: 100%;  
}
```

Apagar

Turn off flexbox in desktop styles

Todo lo que este por debajo de 800px

```
@media screen and (min-width:800px) {  
    .recipe {  
        display: block; /* turn off flexbox */  
    }  
  
    .recipe figure {  
        float: right;  
        width: 55%;  
    }  
}
```

Demo: moving a photo on mobile

Flexbox enhanced



S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.

INGREDIENTS

- 1/2 cup butter melted
- 1 1/2 cup graham cracker crumbs
- 2 cups brown sugar

Non-flexbox

S'MORES BARS

Recipe and photo by [IHeart Nap Time](#)

This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because it has much better support among mobile browsers than desktop browsers. Even if a mobile browser doesn't support it, the reordering here is simply a visual effect, in the spirit of progressive enhancement, so it's not a problem if some mobile users don't see the photo at the top.



INGREDIENTS

These examples don't look wrong
or broken without flexbox.

Flexbox just enhances their sizing
and spacing to look better.

3. HOW

Step-by-step process for adding flexbox
to your UI components effectively

Don't freak out



que o cual

Decide whether flexbox is the right tool for the job

Decide which versions of flexbox to support

standard, 2011/2012, and/or 2009

<http://caniuse.com/#feat=flexbox>

I recommend you skip the '09 syntax

- Slower to render than current syntax*
- Doesn't support wrapping
- Small market share
- Its browsers can get same fallback you give to non-supporting browsers

* <http://updates.html5rocks.com/2013/10/Flexbox-layout-isn-t-slow>

Let tools add browser variants for you

Which variants do you want?

Best tool to handle that

All the things!

[Autoprefixer](#)
[Bourbon \(Sass\)](#)

Only the standard syntax,
with and without prefixes

[Compass \(Sass\)](#)

Some other combination
(such as everything except 2009)

Sass or LESS mixins, such as
[https://github.com/mastastealth/
sass-flex-mixin](https://github.com/mastastealth/sass-flex-mixin)

Add Modernizr as needed with flexbox

plan b

Flexbox and fallback styles can often co-exist, but sometimes need to isolate them

The screenshot shows the Modernizr website interface. At the top, there's a navigation bar with links for DOWNLOAD, DOCUMENTATION, NEWS, and RESOURCES. Below the navigation, a search bar contains the word "flex". The main content area displays the results for the "flex" detect. On the left, a sidebar shows "4 checked" and "6 results" with a file size of "4.44 kB / 1.94 kB gzipped". Under "Options", there's a list of Modernizr methods like ".domPrefixes", ".prefixes", etc., each with a plus sign icon. The main results table has two columns: "Test" and "Result". The "Test" column lists "Flexbox", "Flexbox (legacy)", "Flexbox (tweener)", and "Flex Line Wrapping". The "Result" column for each row contains a pink circle with a white checkmark. To the right of the table, there's a "Flexbox" section with a "Description" and a "warning" note.

Test	Result
Flexbox	✓
Flexbox (legacy)	✓
Flexbox (tweener)	✓
Flex Line Wrapping	✓

Flexbox

Description

Detects support for the Flexible Box Layout model, a.k.a. Flexbox, which allows easy manipulation of layout order and sizing within a container.

warning

A 'true' result for this detect does not imply that the 'flex-wrap' property is supported; see the 'flexwrap' detect.

<http://zomigi.com/blog/using-modernizr-with-flexbox/>

Or use @supports

```
.gallery-item {  
    display: inline-block;
```

```
}
```

Consulta las características del navegador para comprobar si se puede aplicar las reglas agregadas dentro de esta hoja de estilos definida

```
@supports (flex-wrap: wrap) {
```



But IE 10-11, which do support flexbox but don't support @supports, won't get these styles

```
.gallery {
```

```
    display: flex;
```

```
    flex-wrap: wrap;
```

```
}
```

```
}
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/@supports>

Choose and add appropriate starter/fallback layout CSS

retroceder

Things to consider

necesito bloque de contenidos envueltos ?

Do I need content blocks to wrap?

not table-cell

necesito evitar bloques se envuelvan ?

Do I want to prevent blocks from wrapping?

floats, inline-block, but
table-cell best

crear el contenido y luego ajustarlo a la estructura

Do I need content-driven sizes?

floats, but table-cell or
inline-block best

necesitas alineamiento vertical

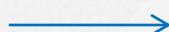
Do I need vertical alignment?



inline-block, table-cell

necesitas alineamiento horizontal

Do I need horizontal alignment?



floats, table-cell, inline-
block only with preset
sizes

Pick your starter layout CSS

Solo usa lo que normalmente usarías

Just use whatever you normally would;
flexbox plays nicely with most of them.

Flexbox will override:

- Floats**
- table-cell**
- inline-block**

Flexbox will not override:

- Absolute positioning**

A real example of this process

Split left-right layout

Task: ^{poner revisión} lay out review score and price, on opposite sides of same line

Needs:

- content-driven sizing basado en el contenido
- horizontal alignment dimensionado
- wrapping



score
marcador

price or
"sold out"

Adding the starter CSS

```
.iw_mini_review_score_wrapper {  
    float: left;  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
}
```

Start adding flexbox!



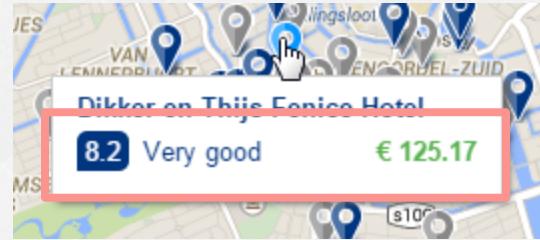
cual o que

Decide whether entire component needs to be block or inline-block

display:flex or inline-flex

Creating the block flex container

```
.iw_mini_details_wrapper {  
    display: flex; ——————→  
}  
  
.iw_mini_review_score_wrapper {  
    float: left;  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
}
```



sienta
Flex container sits
sobre debajo
on a new row below,
like a block element

Decide flow of flex items

Things to consider

poner

Lay out horizontally or vertically?

`flex-direction: row` or
`column`

permitir cajas envolver

Allow boxes to wrap?

`flex-wrap: wrap`,
`wrap-reverse` or `nowrap`

Order different than source?

`order values;`
`flex-direction:`
`row-reverse` or
`column-reverse`

Allowing wrapping

```
.iw_mini_details_wrapper {  
    display: flex;  
    flex-wrap: wrap; —————→  
}  
  
.iw_mini_review_score_wrapper {  
    float: left;  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
}
```

Allows second block to
wrap down if needed

Permite que el segundo bloque se
envuelva si es necesario

cual es el item que puede
Decide which items can
a llenar el espacio
crecer
grow to fill space,
encoger para evitar desbordamiento
shrink to avoid overflow,
debe de estar a un cierto tamaño
or must stay at a certain size

Tips for setting flex values

- Avoid single-digit and keyword values
 - flex: 1 1 0% instead of flex: 1
 - Hidden default values can lead to mistakes
 - Avoids IE 10-11 bugs
- Think about it backwards
 - First decide flex-basis
 - Then decide flex-shrink and flex-grow

Tips for setting flex-basis values

- Acts like min-width when wrapping on envolviendo
- Acts like max-width when flex-shrink on and flex-grow off
- Be careful with flex-basis:0 when wrapping
- Use flex-basis:auto whenever possible

Setting flex-shrink

- Always have at least 1 item per line that can shrink (or wrap, or both)
menos 1 elemento
- Yes, always

Setting flex-grow

- Decide what to do with extra space
 - ▣ Fill it up? (flex-grow: 1, 2, etc.)
 - ▣ Leave it? (flex-grow: 0)

Setting flex values

```
.iw_mini_details_wrapper {  
    display: flex;  
    flex-wrap: wrap;  
}  
  
.iw_mini_review_score_wrapper {  
    float: left;  
    flex: 0 1 auto; ——————→  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
    flex: 0 1 auto;  
}
```

Size to content, shrink
mas pequeño
smaller if you have to,
no hacer crecer el más grande
encogido
don't grow bigger
(default value)

Decide how to align flex items

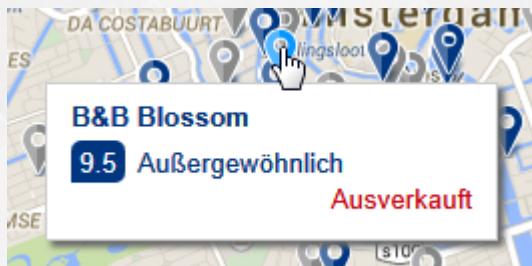
justify-content • align-items
align-self • align-content

Controlling alignment

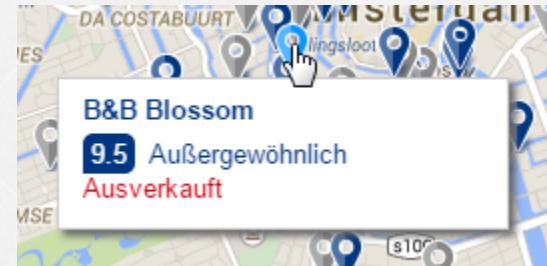
```
.iw_mini_details_wrapper {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: space-between; → Moves first item to left,  
    align-items: baseline;           last item to right  
}  
  
.iw_mini_review_score_wrapper {  
    float: left;  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
}
```

Improved wrapping

Non-flexbox



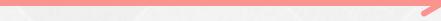
Flexbox enhanced



Flexbox with float fallback

retroceso o plan b

```
.iw_mini_details_wrapper {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: space-between;  
    align-items: baseline;  
}  
  
.iw_mini_review_score_wrapper {  
    float: left;  
}  
  
.iw_mini_price_wrapper {  
    float: right;  
}
```

Flexbox properties on container override floating automatically in supporting browsers

Floating gets used by old browsers

Test

Testing your flexbox

- If using Modernizr: Too Flexy bookmarklet for toggling Modernizr flexbox classes:
<http://chriswrightdesign.github.io/tooflexy/>
- If **reordering**: check **tabbing** and **screen reading order** to **make sure** it's still logical

Fix bugs

Fixing browsers' flexbox bugs

- Read explanations and workarounds by Philip Walton:
<https://github.com/philipwalton/flexbugs>
- Let PostCSS fix some for you:
<https://github.com/luisrudge/postcss-flexbugs-fixes>

Summing up the process

1. Decide which browser versions of flexbox to use
2. Choose and add starter layout CSS
3. Choose and add flexbox CSS
 1. Block or inline-block container
 2. Flow
 3. Flex to control sizing
 4. Alignment
4. Test and fix bugs

Flexbox is not for the future.
You can use flexbox today.

Thanks!

Zoe Mickley Gillenwater

@zomigi

design@zomigi.com

zomigi.com