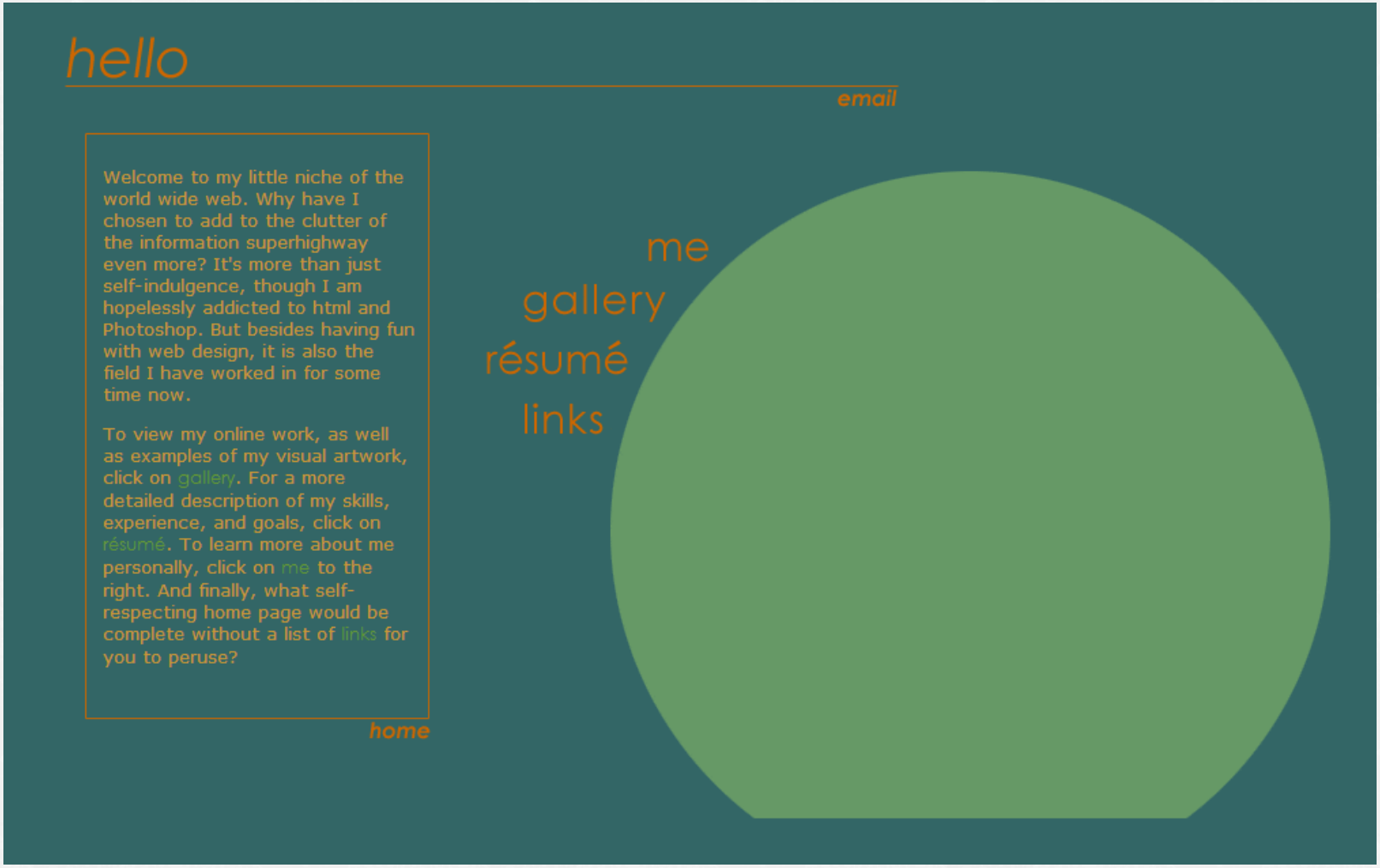
Enhancing Responsiveness WITH Flexbox

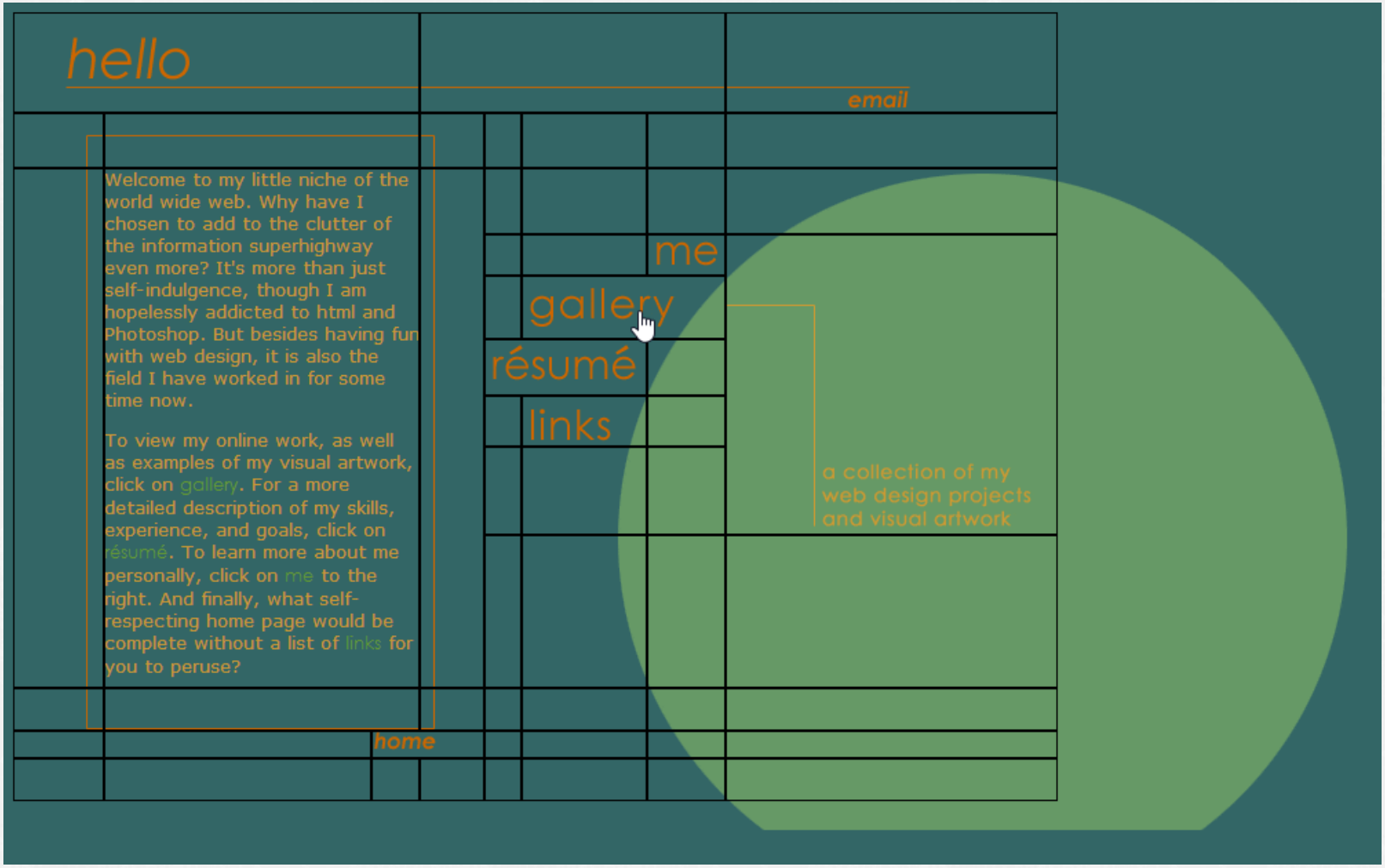
CSS Day
June 2015

Zoe Mickley Gillenwater @zomigi

I used to make fixed-width sites…



...with tables


hello
email
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?
me
gallery
résumé
links
a collection of my web design projects and visual artwork
home

Then I got into CSS “liquid” layout

I even wrote a book about it in 2008.

Problems with CSS 2 layout

- Float containment
- Wrapping hard to control
- Difficulty making boxes equal height
- Difficulty doing vertical centering
- Difficulty mixing units of measurement
- Visual location still tied to HTML order

Flexbox solves a lot of those issues

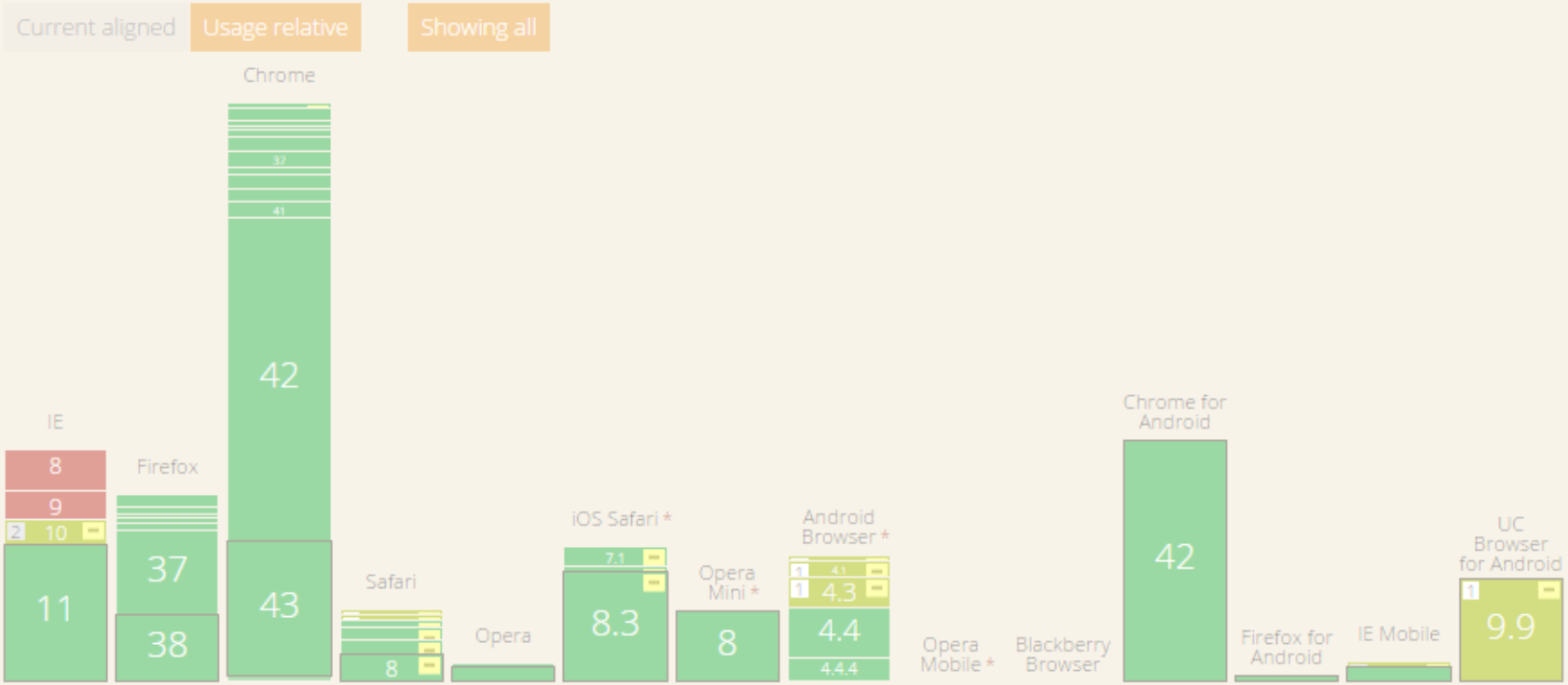
- Make boxes automatically grow to fill space or shrink to avoid overflow
- Give boxes proportional measurements
- Lay out boxes in any direction
- Align boxes on any side
- Place boxes out of order from HTML

Can I use flexbox ? Settings

1 result found

Global 81.44% + 11.8% = 93.24%

unprefixed: 70.92% + 0.34% = 71.26%



You can use flexbox now

Use flexbox now on UI components as progressive enhancement.

“Easy for you to say.
I have to support IE 8.”

– You, maybe

“I work for Booking.com,
and we support IE 7,
and I use flexbox.”

– Me

Progressive enhancement possibilities

- Align items in new ways
- Fill up the gaps in your layout
- Reorder decorative content
- Increase responsiveness

RWD is not binary

Responsiveness is a continuum.

Flexbox can help make your site **more** responsive.

Flexbox and RWD

Space

Make better use of the space at all screen sizes

Placement

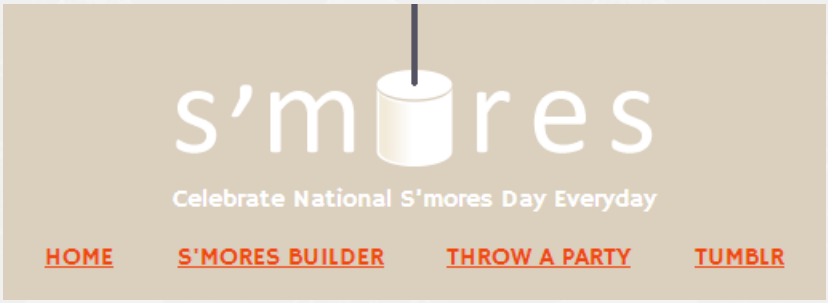
Reorder content at different screen sizes

Making better use of space

Demo: horizontal navigation

Without flexbox:

```
.list-nav {
  margin: 0;
  padding: 0;
  list-style: none;
  text-align: center;
}
.list-nav li {
  display: inline-block;
  padding: 0 .5em;
  text-align: center;
}
```



Demo: horizontal navigation

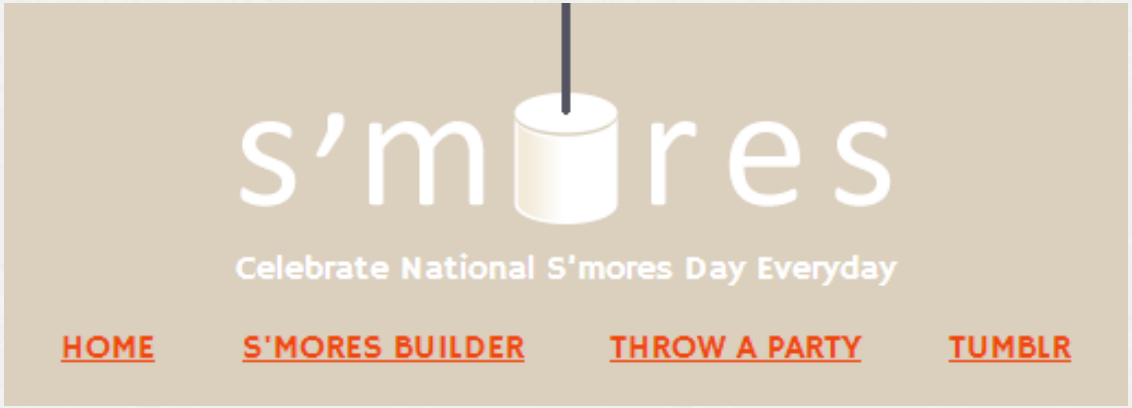
1. Turn `<ul>` into flex container:

```
.list-nav {
  display: flex;
  flex-direction: row; /* default */
  ...
}
```

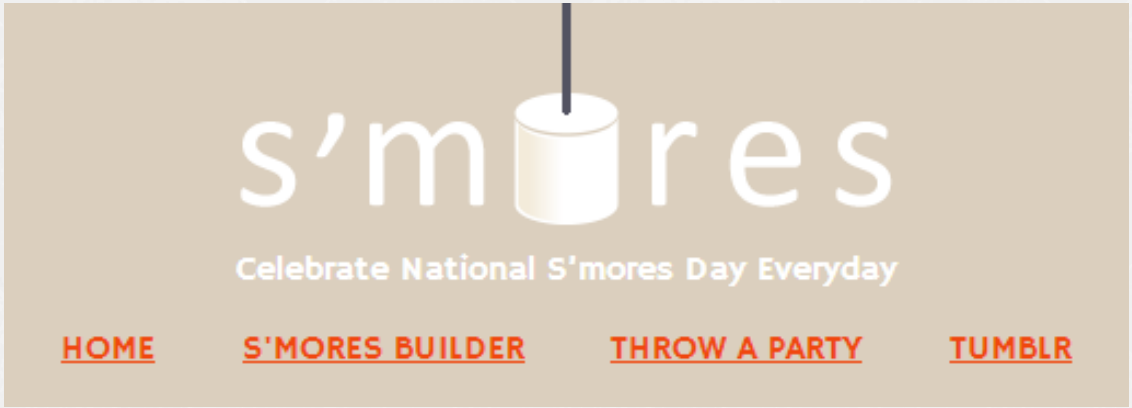
2. Children `<li>` become flex items laid out on single horizontal line

Demo: horizontal navigation

Non-flexbox fallback version



Flexbox version



Making it full-width

- All links on same line
- First link flush left, last link flush right
- Equal spaces between all links

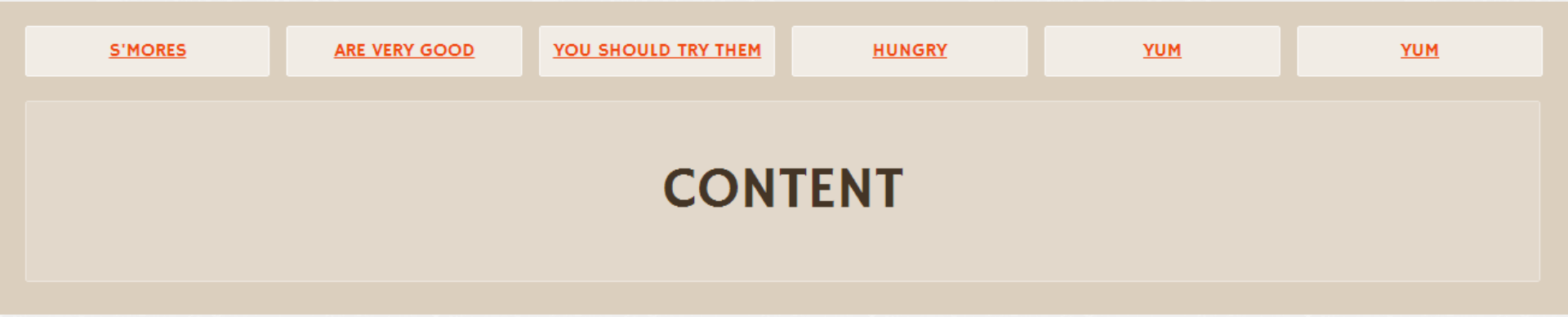
Trying display:table-cell

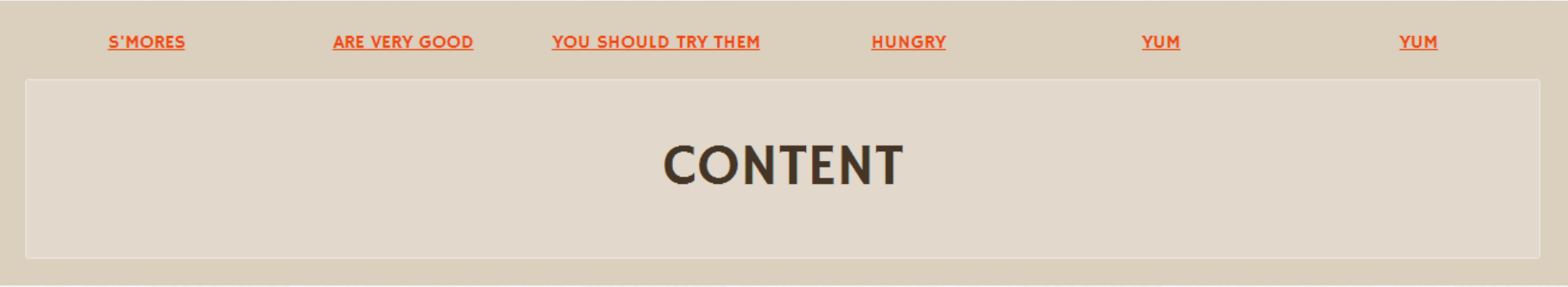
☺ All links on same line

☺ First link flush left, last link flush right

☹ Equal spaces between all links

Spacing with table-layout:fixed





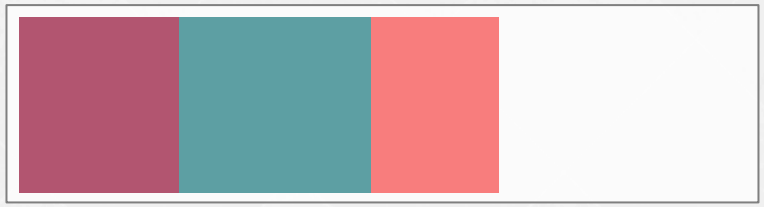
Making it full-width

```
.list-nav {
  display: flex;
  justify-content: space-between;
  margin: 0;
  padding: 0;
  list-style: none;
}
.list-nav li {
  text-align: center;
}
```

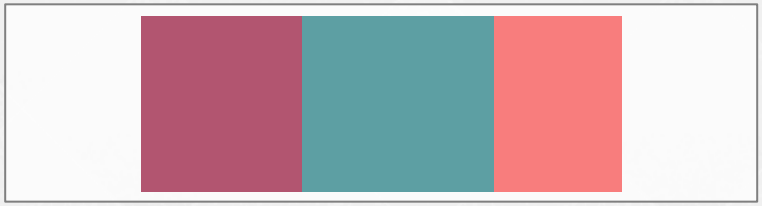
justify-content

aligns flex items along main axis

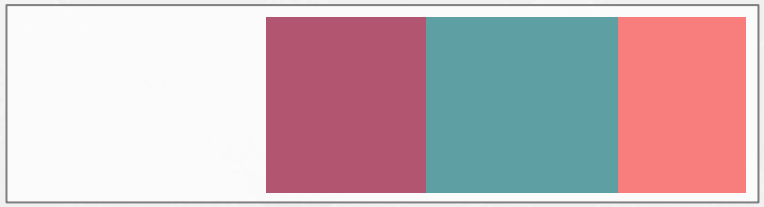
(flow direction set by flex-direction)



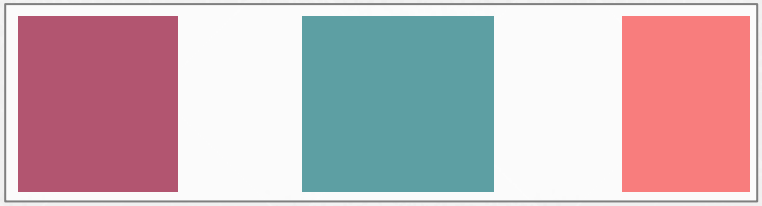
`flex-start`
(default)



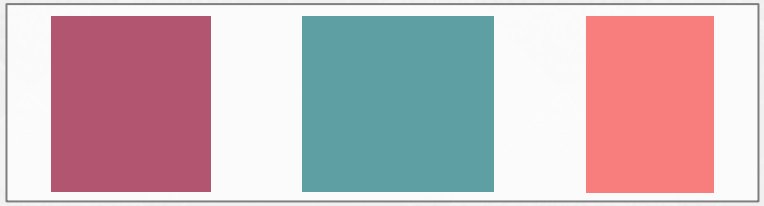
`center`



`flex-end`



`space-between`



`space-around`

Combine with inline-block

```
.list-nav {
  display: flex;
  justify-content: space-between;
  margin: 0;
  padding: 0;
  list-style: none;
  text-align: center; /* fallback */
}
.list-nav li {
  display: inline-block; /* fallback */
  padding: 0 .5em; /* fallback */
  text-align: center;
}
.list-nav li:first-child { padding-left: 0; }
.list-nav li:last-child { padding-right: 0; }
```

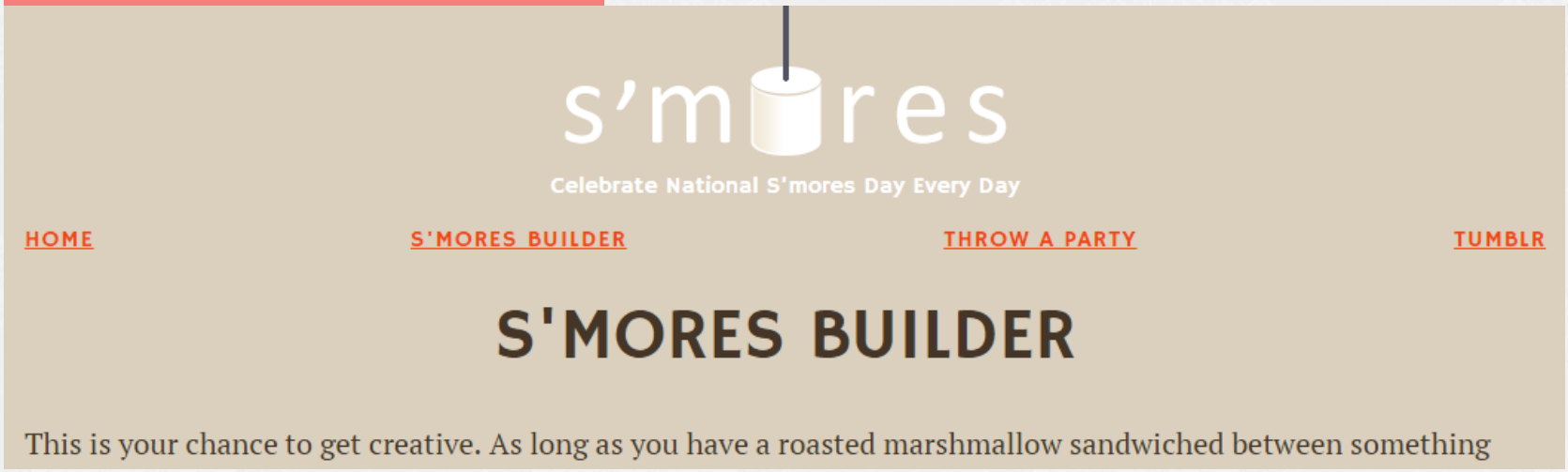
Combine with inline-block

Non-flexbox fallback version

Flexbox version

Improve the wide layout

Wide: too stretched out



A more responsive enhancement

Wide variation: two-piece main nav

1. Add media query for wide width:

```
@media (min-width:860px) {
}
```

2. Add link to Modernizr:

```
<script src="js/modernizr.js"></script>
```

Supporting browsers:

```
<html class="flexbox">
```

Non-supporting browsers:

```
<html class="no-flexbox">
```

Add Modernizr as needed with flexbox

- Flexbox and fallback styles can often co-exist, but sometimes need (or want) to isolate them
- Modernizr can add `flexbox`, `no-flexbox`, and `flexboxlegacy` classes to do this

Wide variation: two-piece main nav

3. Move nav bar up to overlap logo's line:

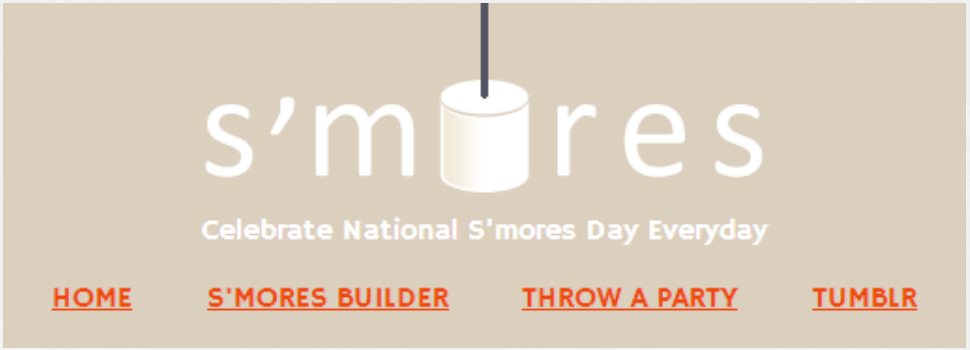
```
@media (min-width:860px) {
  .flexbox .list-nav {
      position: relative;
      top: -70px;
  }
}
```

Wide variation: two-piece main nav

4. Add margins to control extra space in line:

```
.flexbox .link-party {
    margin-left: auto;
}
.flexbox .link-home   { margin-right: 15px; }
.flexbox .link-tumblr { margin-left: 15px; }
```

A more responsive nav bar



This works vertically too

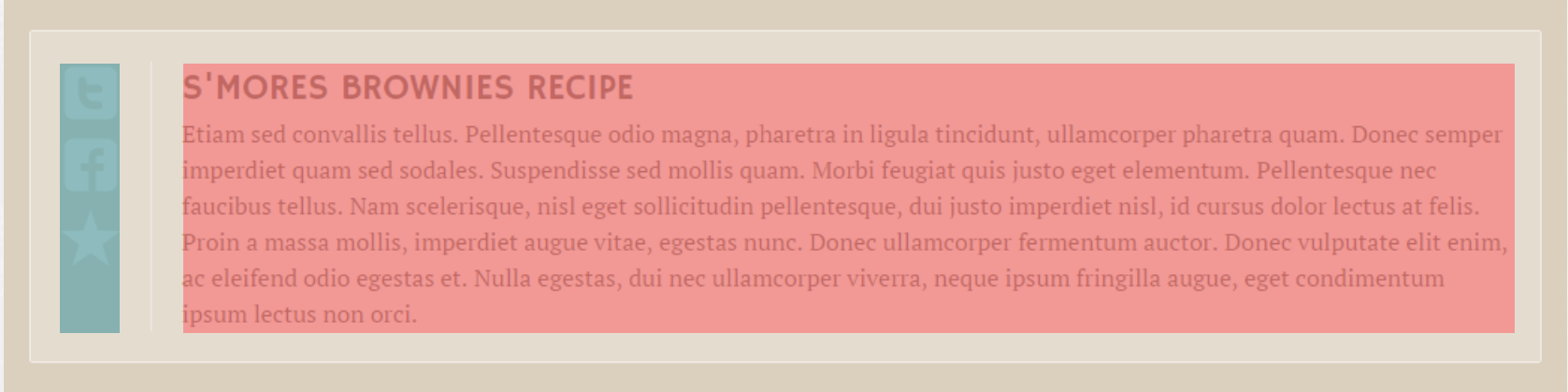
Content blocks' height varies in RWD, and flexbox can also help you make better use of varying vertical space

Demo: full-height stacked icons

Demo: full-height stacked icons

1. Turn children `.icons` and `.content` into side-by-side, equal-height flex items

```
.wrapper {
  display: flex;
  align-items: stretch; /* default */
}
```



align-items

aligns flex items in cross axis

(justify-content aligned in main axis)

`stretch`
(default)

`flex-start`

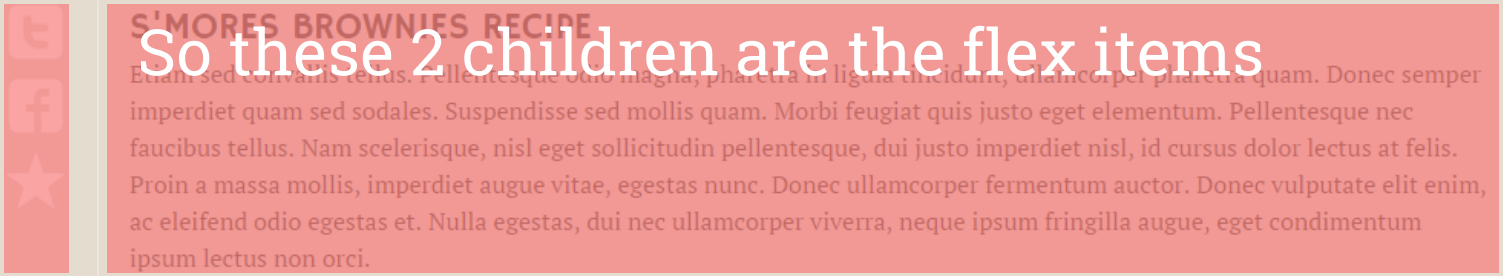
`flex-end`

`center`

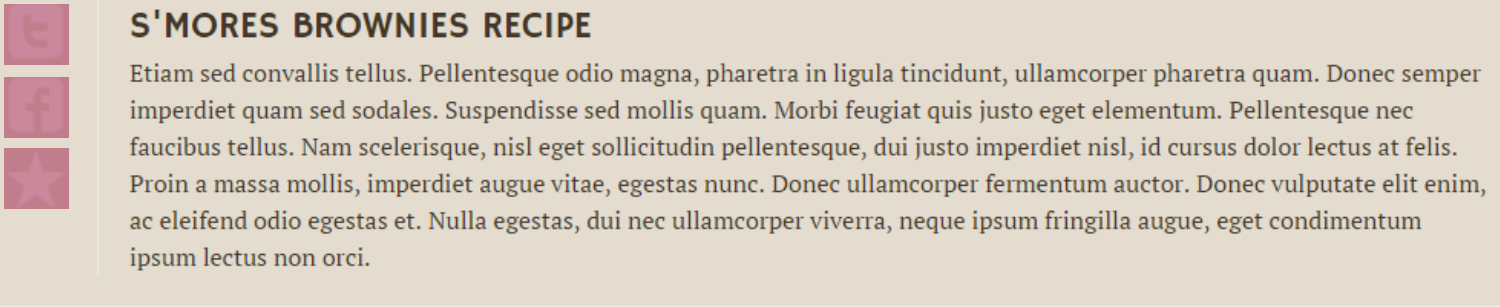
`baseline`

Only children become flex items

This is the flex container



These 3 grandchildren aren't flex items (yet)



Demo: full-height stacked icons

2. Turn `.icons` into flex container with vertically stacked children (the 3 icons):

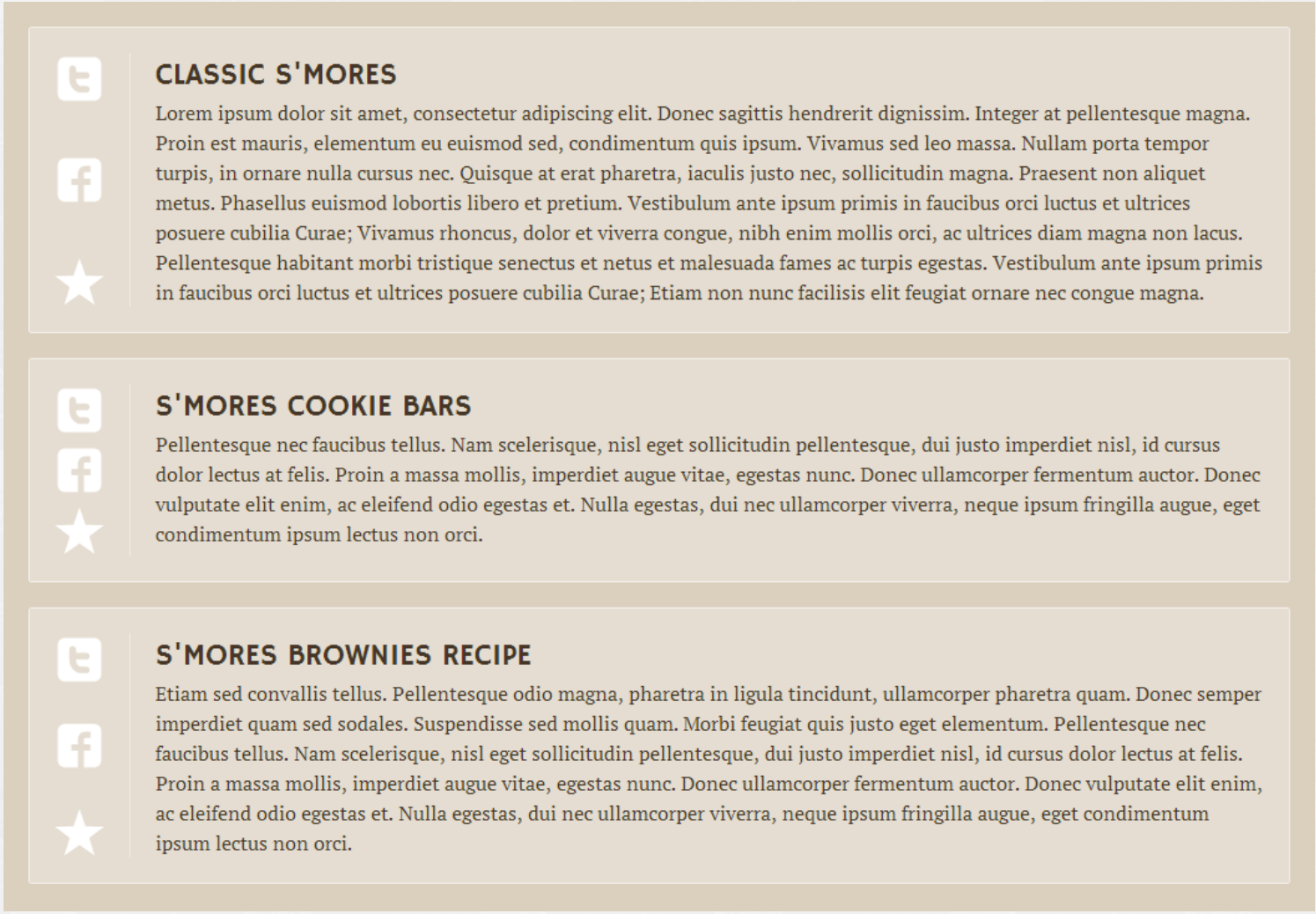
```
.icons {
  display: flex;
  flex-direction: column; /* main axis */
}
```

Demo: full-height stacked icons

3. Equally space the 3 icons along the vertical main axis:

```
.icons {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
}
```

Demo: full-height stacked icons



Combine with table-cell

```
.wrapper {
    display: table; /* fallback */
    display: flex;
}
.icons {
    display: table-cell; /* fallback */
    vertical-align: middle; /* fallback */
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}
.content {
    display: table-cell; /* fallback */
    vertical-align: top; /* fallback */
    flex: 1 0 0px;
}
```

Combine with float

```
.wrapper {
    display: flex;
}
.icons {
    float: left; /* fallback */
    position: relative; /* fix for old WebKit bug w/ floated flex items */
    width: 40px;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}
.content {
    flex: 1 0 0px;
}
.no-flexbox .content {
    margin-left: 60px; /* fallback */
}
```

Fallback alignment options

Top-aligned (float)

CLASSIC S'MORES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Done
Proin est mauris, elementum eu euismod sed, condimentum
turpis, in ornare nulla cursus nec. Quisque at erat pharetra, ia
metus. Phasellus euismod lobortis libero et pretium. Vestibul
posuere cubilia Curae; Vivamus rhoncus, dolor et viverra cong
Pellentesque habitant morbi tristique senectus et netus et ma
in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam

S'MORES COOKIE BARS

Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget s
dolor lectus at felis. Proin a massa mollis, imperdiet augue vi
vulputate elit enim, ac eleifend odio egestas et. Nulla egestas
condimentum ipsum lectus non orci.

S'MORES BROWNIES RECIPE

Etiam sed convallis tellus. Pellentesque odio magna, pharetra
imperdiet quam sed sodales. Suspendisse sed mollis quam. M
faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellent
Proin a massa mollis, imperdiet augue vitae, egestas nunc. Do
ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorper
ipsum lectus non orci.

Centered (table-cell)

CLASSIC S'MORES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Don
Proin est mauris, elementum eu euismod sed, condimentum
turpis, in ornare nulla cursus nec. Quisque at erat pharetra, ia
metus. Phasellus euismod lobortis libero et pretium. Vestibu
posuere cubilia Curae; Vivamus rhoncus, dolor et viverra con
Pellentesque habitant morbi tristique senectus et netus et m
in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam

S'MORES COOKIE BARS

Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget s
dolor lectus at felis. Proin a massa mollis, imperdiet augue vi
vulputate elit enim, ac eleifend odio egestas et. Nulla egestas
condimentum ipsum lectus non orci.

S'MORES BROWNIES RECIPE

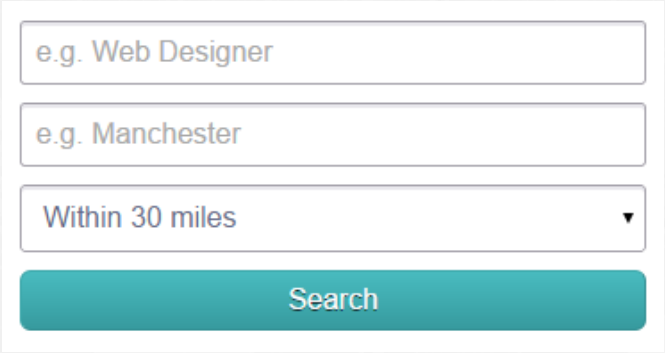
Etiam sed convallis tellus. Pellentesque odio magna, pharetra
imperdiet quam sed sodales. Suspendisse sed mollis quam. M
faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellent
Proin a massa mollis, imperdiet augue vitae, egestas nunc. Do
ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorpe
ipsum lectus non orci.

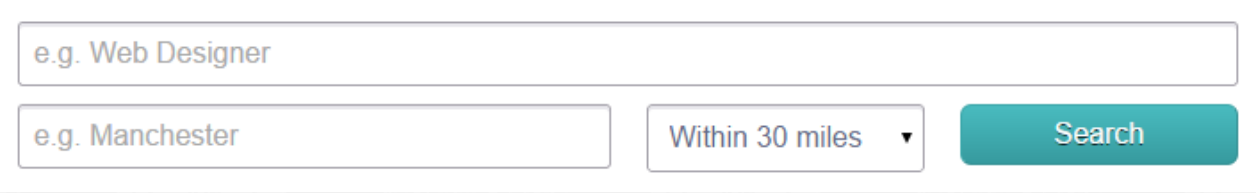
ems + % + px + [blank] = ???

Combining units of measurement across a line can make RWD tricky.

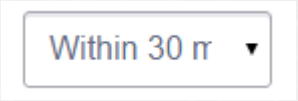
Demo: responsive form

Inspired by http://jobs.theguardian.com/, which uses floats and percentage widths

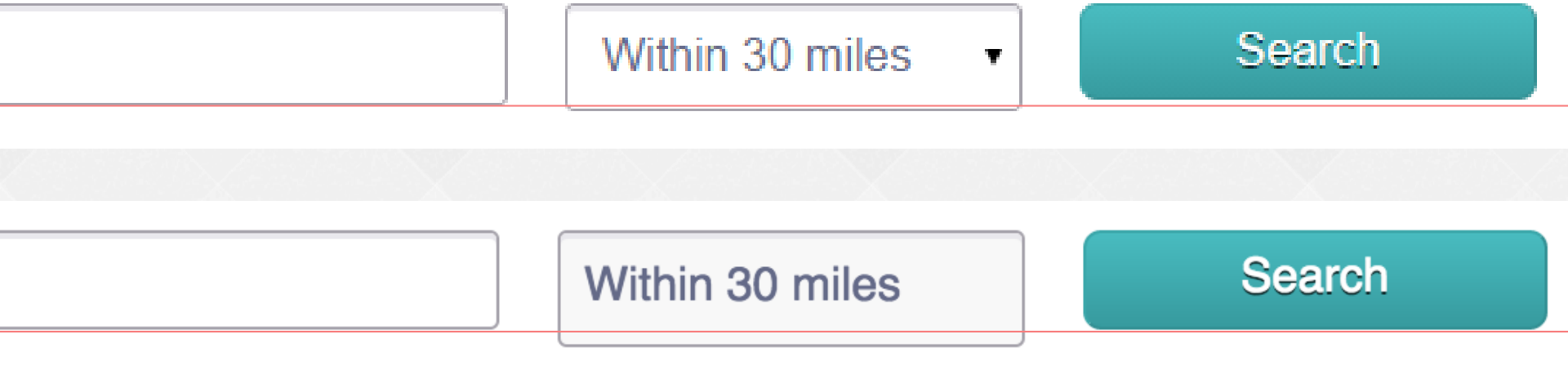




But it would be nicer if…

- The drop-down and button were sized automatically by their content, so this doesn't happen: 

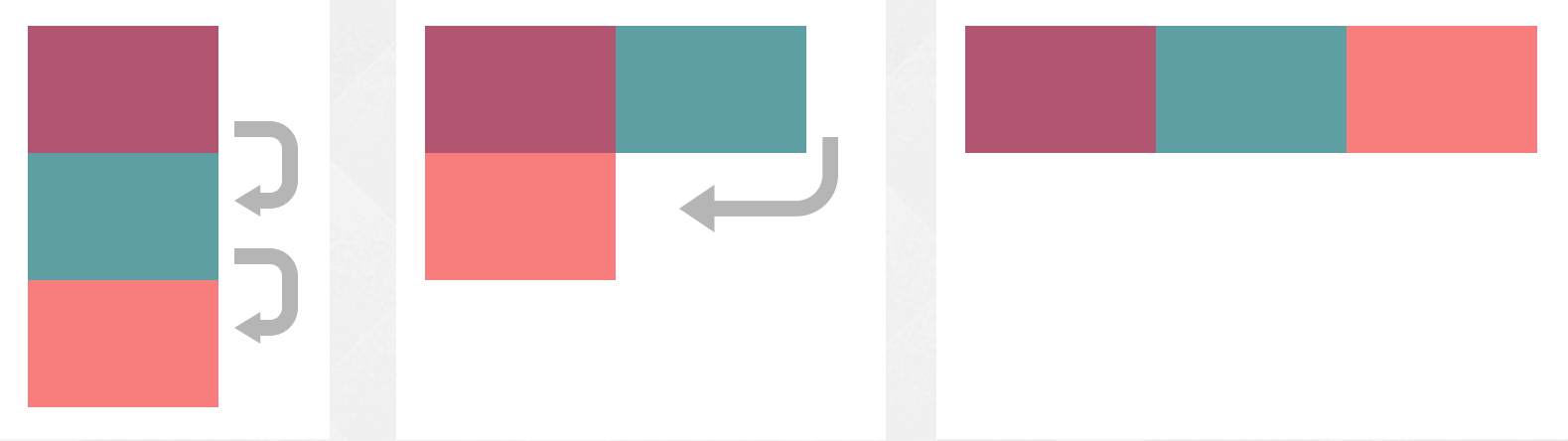
- The fields and button all matched each other exactly in height



Enhance form with flexbox

1. Let the fields wrap when needed:

```
.jobs-form {
    display: flex;
    flex-wrap: wrap;
}
```



Enhance form with flexbox

2. Override the % widths and use flex instead:

```
/* hide from non-flex browsers */
.flexbox .jobs-form_field-wrapper {
    width: auto;
    flex: 1 1 100%;
}
```

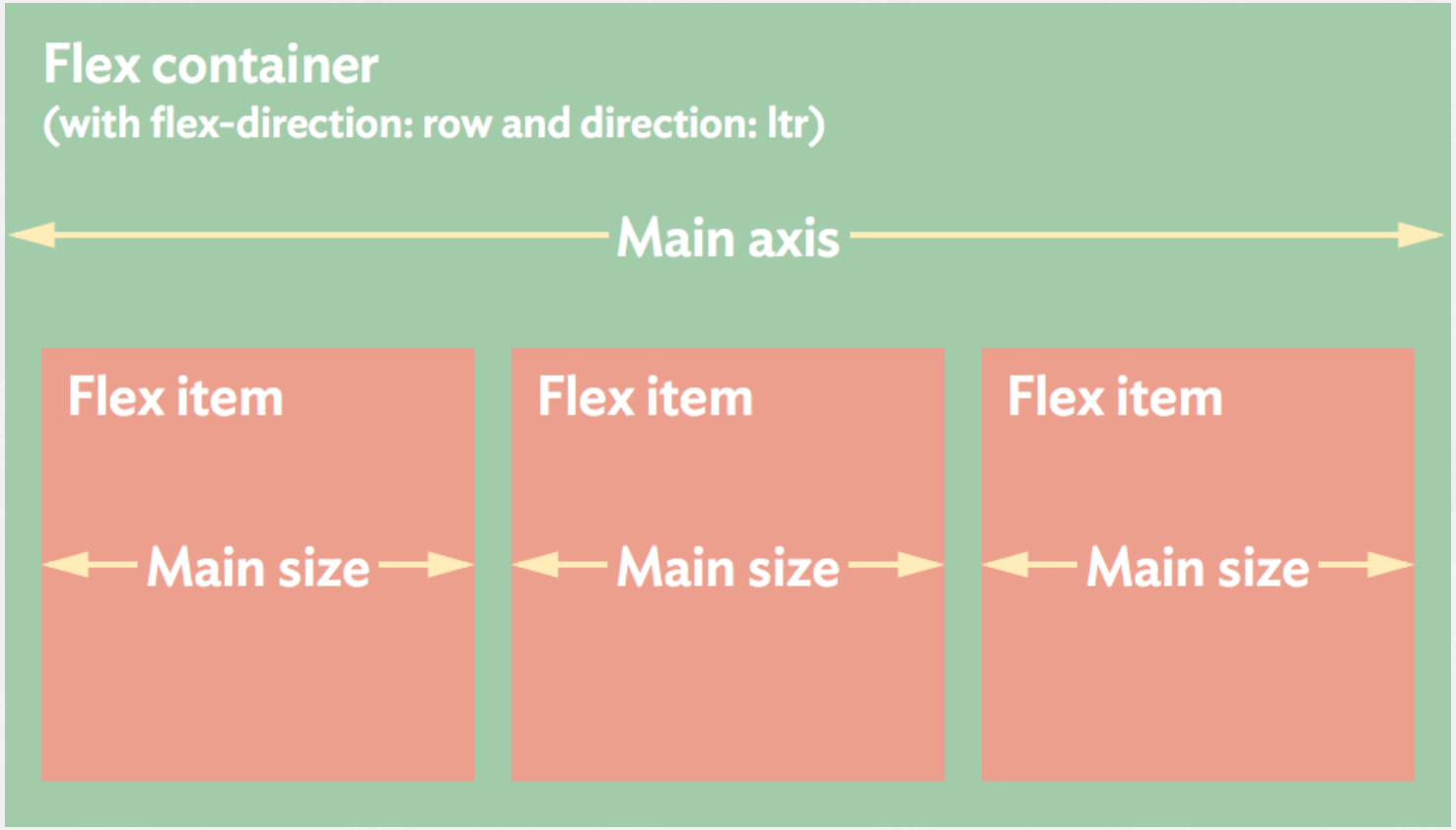
Enhance form with flexbox

3. When wider, let fields size to their content:

```
@media (min-width:40em) {
  .jobs-form_keywords,  /* the 2 text fields */
  .jobs-form_location {
    flex: 1 1 auto;
  }
  .jobs-form_distance,  /* select and button */
  .jobs-form_submit {
    flex: 0 0 auto;
  }
}
```

Defining the flex property

Makes flex items change their main size (width or height) to fit available space



Defining the flex property

`flex-grow`

how much flex item will grow relative to other items if extra space is available (proportion of extra space that it gets)

`flex-shrink`

how much item will shrink relative to others if there is not enough space (proportion of overflow that gets shaved off)

`flex-basis`

the initial starting size before free space is distributed (any standard width/height value, including `auto`)

Breaking down the flex property

```
@media (min-width:40em) {

  .jobs-form_keywords,
  .jobs-form_location {
    flex: 1 1 auto;
  }

  .jobs-form_distance,
  .jobs-form_submit {
    flex: 0 0 auto;
  }

}
```

`flex-basis = auto`

start field at “main size” value (in this case, `width`) or natural content size if main size not set

Breaking down the flex property

```
@media (min-width:40em) {

  .jobs-form_keywords,
  .jobs-form_location {
    flex: 1 1 auto;
  }

  .jobs-form_distance,
  .jobs-form_submit {
    flex: 0 0 auto;
  }

}
```

`flex-shrink = 1`

it's ok to shrink smaller than starting width if there's not enough space

`flex-shrink = 0`

don't shrink smaller than starting width

Breaking down the flex property

```
@media (min-width:40em) {

  .jobs-form_keywords,
  .jobs-form_location {
    flex: 1 1 auto;
  }

  .jobs-form_distance,
  .jobs-form_submit {
    flex: 0 0 auto;
  }

}
```

`flex-grow = 1`

give it 1 share of any extra width on its line

`flex-grow = 0`

don't grow bigger than starting width

In other words...

```
@media (min-width:40em) {

  .jobs-form_keywords,
  .jobs-form_location {
    flex: 1 1 auto;
  }

  .jobs-form_distance,
  .jobs-form_submit {
    flex: 0 0 auto;
  }

}
```

Text fields:

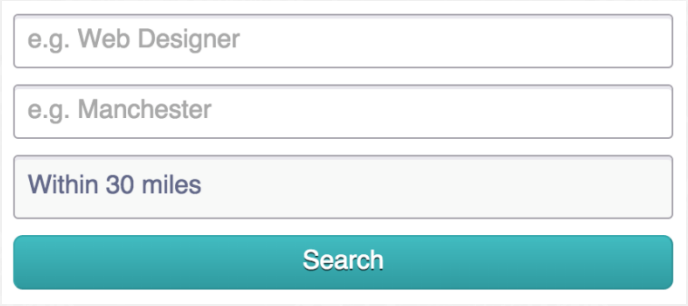
You guys adjust to the space available to fill the line.

Select and button:

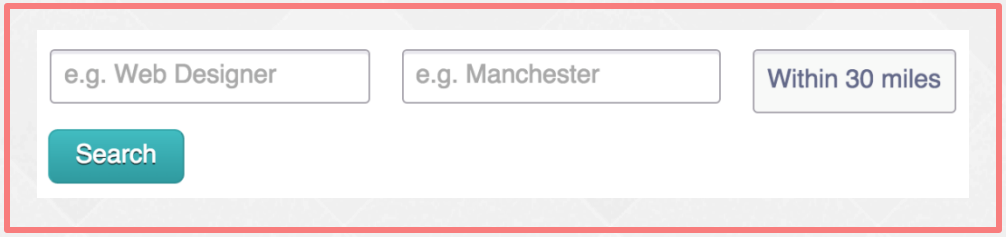
You guys just stay at your content width.

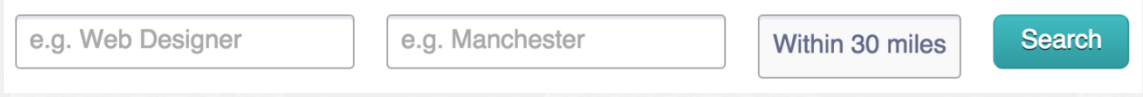
The flexbox form so far

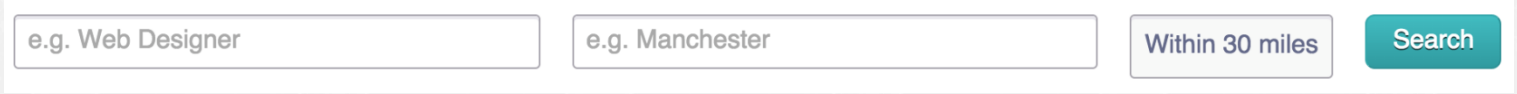
`flex-basis: 100%`



`flex-basis: auto`







Enhance form with flexbox

4. Refine keywords field's `flex-basis` values to improve the layout:

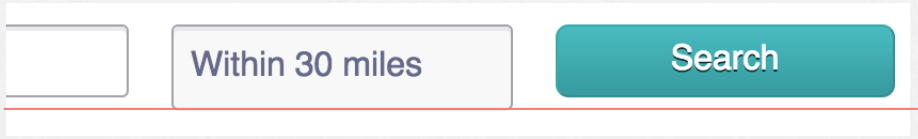
```
@media (min-width:40em) {
  .jobs-form_keywords { flex: 1 1 100%; }
  .jobs-form_location { flex: 1 1 auto; }
  .jobs-form_distance,
  .jobs-form_submit   { flex: 0 0 auto; }
}
@media (min-width:50em) {
  .jobs-form_keywords { flex-basis: auto; }
}
```

Enhance form with flexbox

5. Turn each field wrapper into flex container so field inside will stretch to match height of its line:

```
.flexbox .jobs-form_field-wrapper {
  display: flex; /* sets align-items:stretch */
  width: auto;
  flex: 1 1 100%;
}
```

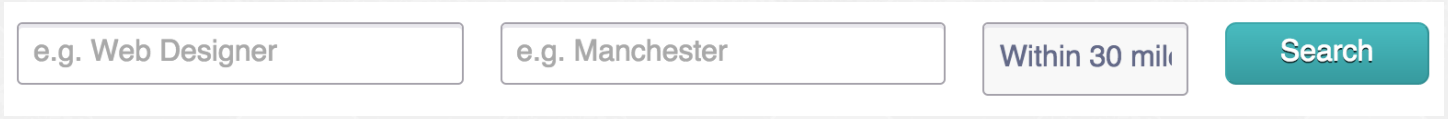
Fields misaligned without flexbox



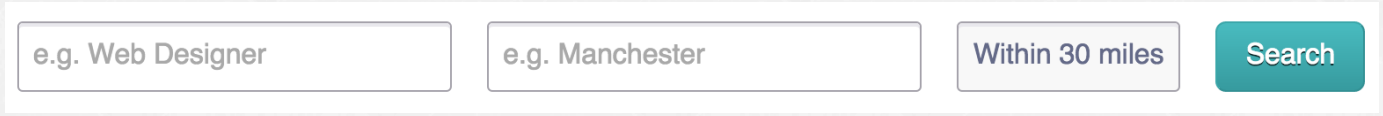
Fields match height due to align-items

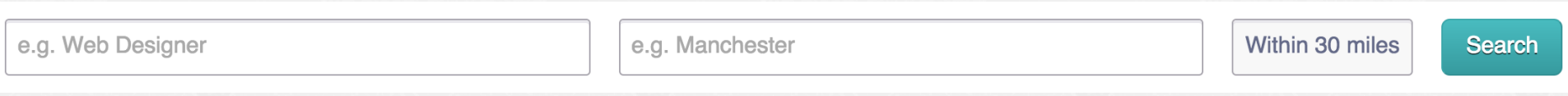
Smarter sizing

Non-flexbox



Flexbox enhanced

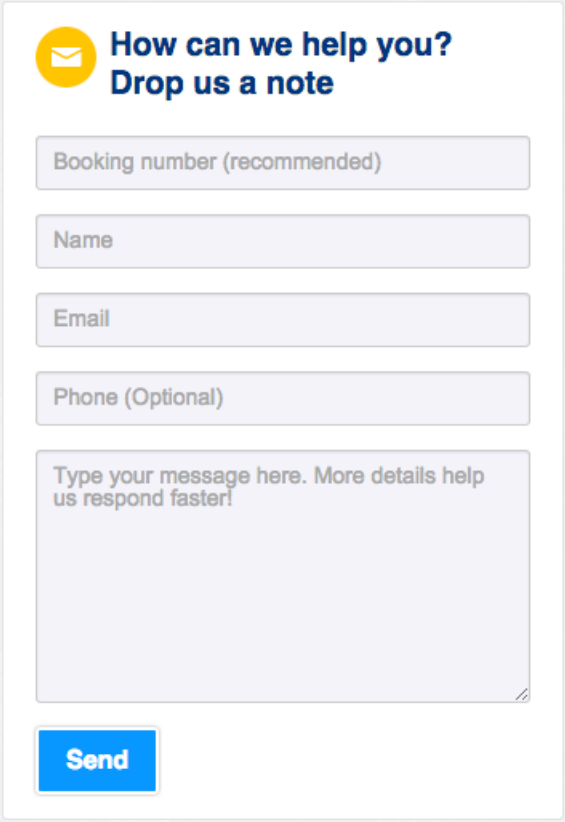


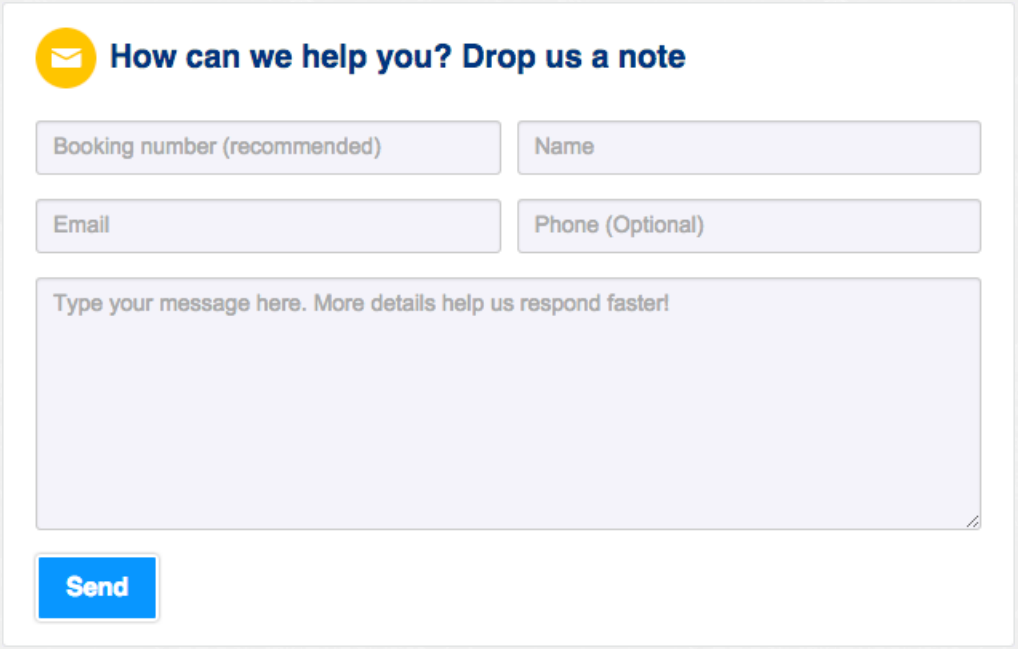


Content-driven breakpoints aren’t perfect.

Automatic orientation switch

Booking's responsive customer service form doesn't use any media queries





http://www.booking.com/content/cs.html

Automatic orientation switch

```
form.cs-message {
  display: flex;
  flex-flow: row wrap;
  margin-right: -10px;
}
input.cs-message__text {
  flex: 1 0 40%;
  width: 43%;
  float: left;
  margin-right: 10px;
  padding: 8px 10px;
}
```

1 property creates
2 responsive layouts,
both always full width

Demo: responsive article header

Narrow starting styles

- No media query
- `display: block`

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.

Wide starting styles

- 50em media query
- `display: table-cell`

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.

Enhance header with flexbox

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.

`.article-header-image`

`.article-header-text`

`.article-header`

Enhance header with flexbox

1. Make photo and text block automatically sit side-by-side when they can fit (300px + 20em):

```
.article-header {
  display: flex;
  flex-wrap: wrap;
  margin-left: -20px;
}
.article-header-image {
  flex: 1 1 300px;
  padding: 0 0 20px 20px;
}
.article-header-text {
  flex: 1 1 20em;
  padding: 0 0 20px 20px;
}
```

Enhance header with flexbox

2. Enhance alignment of text within the text block:

```
...
.article-header-text {
  display: flex;
  flex-wrap: wrap;
  align-items: baseline;
  align-content: space-between;
  flex: 1 1 20em;
  padding: 0 0 20px 20px;
}
.article-title {
  flex: 1 1 100%;
}
.article-category {
  flex: 1 1 auto;
}
```

Demo: responsive article header

Another School-free Snow Day for Hillsborough Kids

Weather 28 January 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit

`flex: 1 1 auto`

Another School-free Snow Day for Hillsborough Kids

Weather 28 January 2015

`align-content: space-between`

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

With float or text-align

Another School-free Snow Day for Hillsborough Kids

Hillsborough News

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.

With flex or justify-content

Another School-free Snow Day for Hillsborough Kids

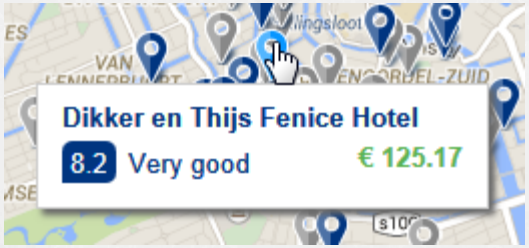
Hillsborough News

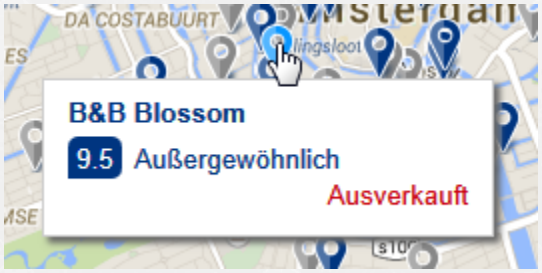
28 January 2015

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

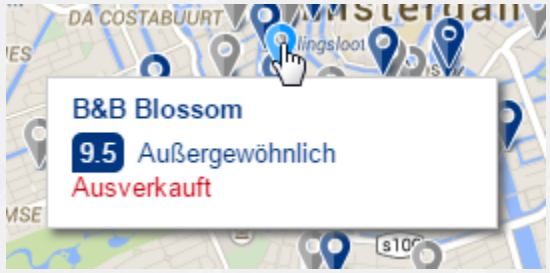
A real-world wrapping example

Without flexbox (IE 9)





With flexbox (Chrome)



Flexbox with float fallback

```
.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;
}
```

Reordering content

Remember this?

```
.flexbox .list-nav {
    position: relative;
    top: -70px;
}
.flexbox .link-party {
    margin-left: auto;
}
.flexbox .link-home   { margin-right: 15px; }
.flexbox .link-tumblr { margin-left: 15px; }
```

Nav overlaps logo's line, so link text could overlap logo if viewport too narrow or text too big

Use order property to move logo

1. Divide nav bar into order groups:

```
.link-home, .link-builder {
    order: 0; /* default, and first here */
}
.logo {
    order: 1; /* second */
}
.link-party, .link-tumblr {
    order: 2; /* last */
}
```

Use order property to move logo

2. Split extra space on line to center logo:

```
.logo {
    margin-left: auto;
}
.link-party {
    margin-left: auto;
}
```

Order only works on siblings

To move logo to middle of list, it needs to be part of list

~~ <img src="images/logo.png"> ~~

```
<ul class="list-nav">
    <li class="logo"><img src="images/logo.png"></li>
    <li class="link-home"><a>home</a></li>
    <li class="link-builder"><a>s'mores builder</a></li>
    <li class="link-party"><a>throw a party</a></li>
    <li class="link-tumblr"><a>tumblr</a></li>
</ul>
```

Accessibility implications

Pro

Can keep content in logical order in HTML instead of structuring HTML just to achieve a visual layout.

Cons

If HTML order is illogical, screen reader users still hear it.

Focus/tab order won't always match expected order, may jump around seemingly randomly.

Use the order property sparingly

If you’re using it for progressive enhancement, the content should make sense in both the HTML and visual order.

Reordering mobile content

Problem

In RWD, narrow-view (mobile) stacking order doesn't always match needed HTML order for wide-view (desktop) layout

Solution

Keep HTML order needed for desktop and use `order` property only on mobile, since browser support is great there

Demo: moving a photo on mobile

Mobile: reordered

S'MORES BARS

Recipe and photo by I Heart 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

Desktop: HTML order (no flexbox)

S'MORES BARS

Recipe and photo by I Heart 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 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!

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%;
}
```

Inspired by Jonathan Cutrell's example at http://webdesign.tutsplus.com/tutorials/tricks-with-flexbox-for-better-css-patterns--cms-19449

Turn off flexbox in desktop styles

```
@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 version

S'MORES BARS

Recipe and photo by I Heart 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

Non-flexbox version

S'MORES BARS

Recipe and photo by I Heart Nap Time

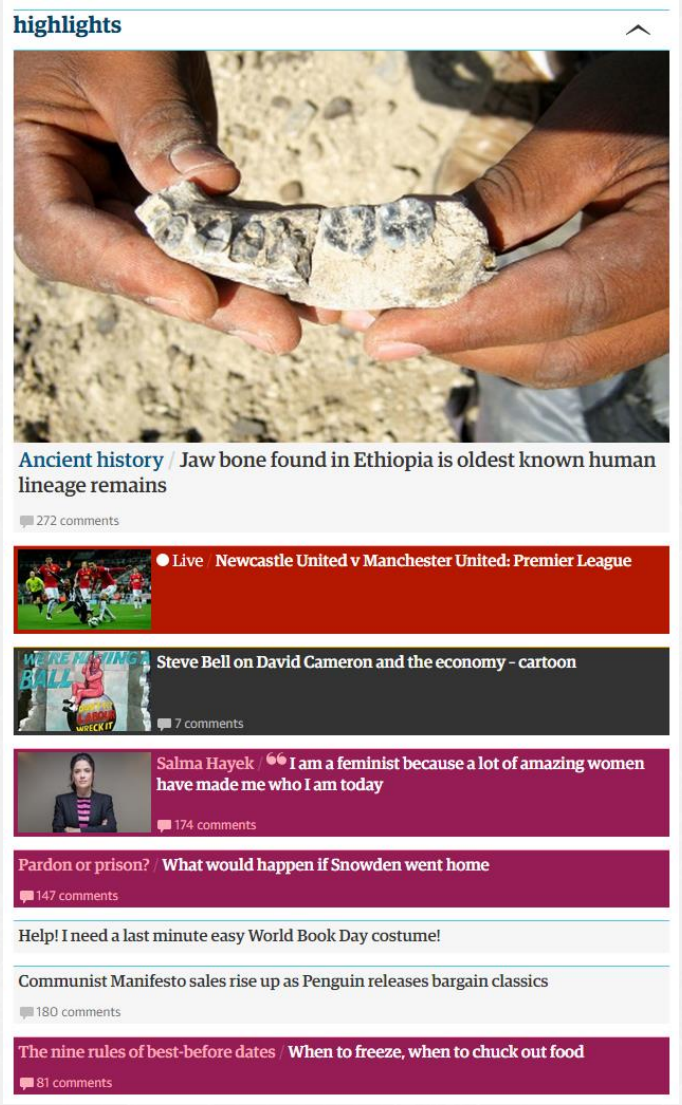
This is a demo of using the CSS flexbox `order` property for moving a photo out of the normal HTML flow. On narrow screens, such as you would see on mobile devices, the photo moves up over the title of the recipe—scrunch down your browser and check it out! Flexbox is only used on narrow screens because 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

The Guardian: opposite approach

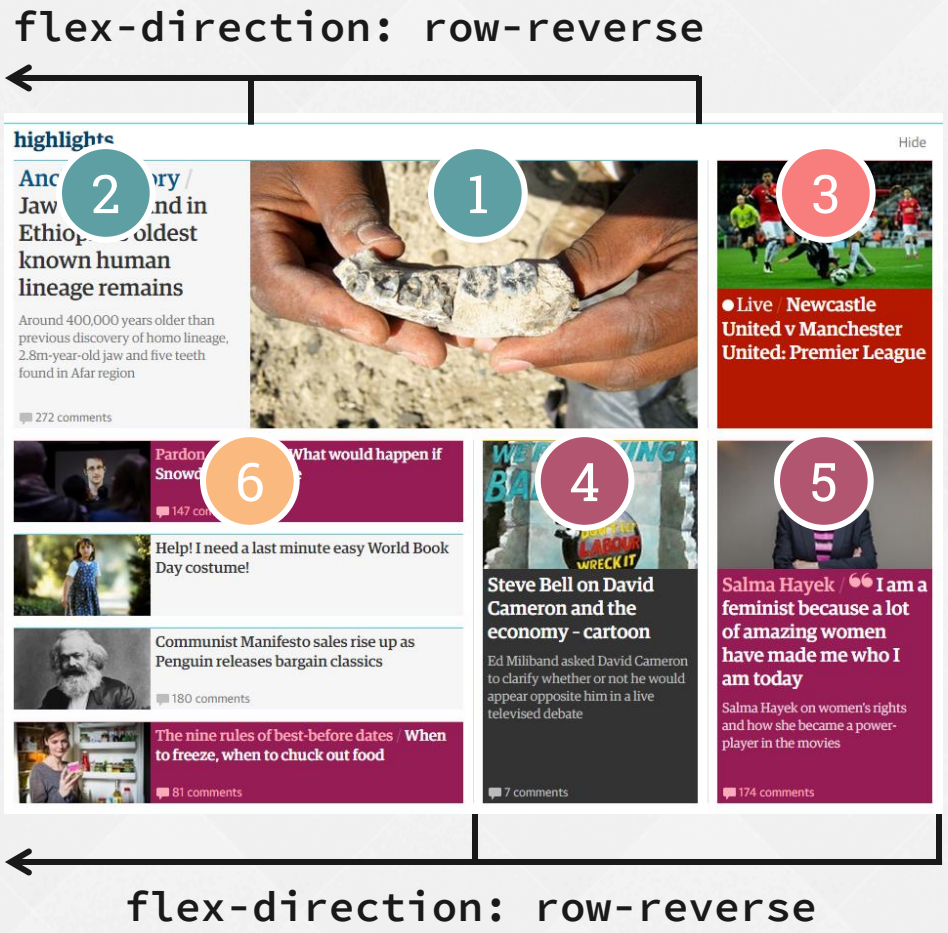
Stacking order you see when narrow is the HTML order, unchanged



1
2
3
4
5
6

The Guardian: opposite approach

Reordered when wide, but not using `order`



Using flexbox today

Pick which versions you'll use

2009 syntax
`(display:box)`

2011 syntax
`(display:flexbox)`

Current syntax
`(display:flex)`

* with -webkit- prefix

† with -ms- prefix

I recommend you skip the ‘09 syntax

- It’s slower to render than current syntax*
- Doesn’t support wrapping
- Its browsers have tiny market share
- You should be using flexbox in progressive enhancement sort of way regardless, so its browsers will just get same fallback you provide to non-supporting browsers

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

Set up your tools

- Let Autoprefixer, Sass, or LESS add the browser variants for you:
  - https://github.com/ai/autoprefixer
  - https://github.com/mastastealth/sass-flex-mixin
  - https://gist.github.com/cimmanon/4461470
  - https://github.com/thoughtbot/bourbon/blob/master/app/assets/stylesheets/css3/_flex-box.scss
  - https://github.com/annebosman/FlexboxLess
- Keep Modernizr on hand to help feed different styles to different browsers: http://modernizr.com/download/

Adding Modernizr classes with Sass

```
@mixin supports-flexbox($support) {
      @if $support == yes {
            .flexbox & { @content }
      }
      @if $support == no {
            .no-flexbox & { @content }
      }
}
```

Adding Modernizr classes with Sass

```
.container {
    display: flex;
}
.sidebar {
    float: left;
    width: 300px;
}
.main-content {
    margin-left: 320px;

    @include supports-flexbox(yes) {
        margin-left: 20px;
    }
}
```

Adding Modernizr classes with Sass

```
.container {
    display: flex;
}
.sidebar {
    float: left;
    width: 300px;
}
.main-content {
    margin-left: 320px;
}
.flexbox .main-content {
    margin-left: 20px;
}
```

Pick your starter/fallback layout CSS

No single right/best answer. Use whatever you normally would.

Flexbox will override:

- Floats
- table-cell
- inline-block

Flexbox will not override:

- Absolute positioning

Flexbox is not

or

Learn more

Download slides and get links at
www.zomigi.com/blog/rwd-flexbox

Thanks!

Zoe Mickley Gillenwater
@zomigi
design@zomigi.com
zomigi.com | stunningcss3.com | flexiblewebbook.com

Photo credits: “oh.my.goshk” by Abulic Monkey and “A Cone Undone” by patersor on Flickr.