

# LEVELING UP WITH FLEXBOX

SmashingConf  
March 2014

Zoe Mickley Gillenwater

@zomigi



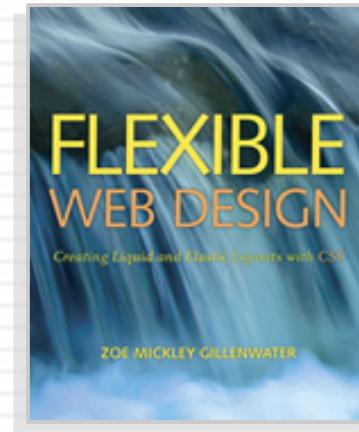
UX Designer

Booking.com

# Author



**Stunning CSS3:**  
A Project-based Guide  
to the Latest in CSS  
[www.stunningcss3.com](http://www.stunningcss3.com)



**Flexible Web Design:**  
Creating Liquid and Elastic  
Layouts with CSS  
[www.flexiblewebbook.com](http://www.flexiblewebbook.com)

# Speaker

“That’s cool! But I can’t use it because XYZ doesn’t support it. Maybe someday.”

why

how

“It’s OK if stuff doesn’t look the same everywhere.  
I’m just going to feed XYZ simpler styles.”

# The Old IE Acceptance Cycle

**Denial:** “No one uses IE 8 any more.”

**Anger:** “How can #\*!% IE not support this?!”

**Bargaining:** “If I use this script, it will almost be the same in IE, right?”

**Depression:** “I’ll never be able to make anything cool because of IE.”

**Acceptance:** “It’s OK if things look different in old IE.”

# You can use flexbox now

It doesn't need to be in the “can't use it yet” pile anymore.

Use flexbox now on small page components as progressive enhancement.

# Syntax is solid

The CSS Flexible Box Layout Module  
([www.w3.org/TR/css3-flexbox/](http://www.w3.org/TR/css3-flexbox/)) has been a Candidate Recommendation since September 2012.

# Browser support is good

2009 syntax  
(`display:box`)



2011 syntax  
(`display:flexbox`)



Current syntax  
(`display:flex`)



\* with -webkit- prefix

† with -ms- prefix

# Keeping track of variants

- Flexy Boxes code generator shows old and new syntaxes: [www.the-echoplex.net/flexyboxes/](http://www.the-echoplex.net/flexyboxes/)
- Let Sass or LESS do it for you, for instance:
  - <https://github.com/mastastealth/sass-flex-mixin>
  - <https://gist.github.com/cimmanon/4461470>
- Let Autoprefixer library do it for you:  
<https://github.com/ai/autoprefixer>

# Use the variants you want.

But for the sake of readability, I'm omitting them from the code samples on these slides. You can see them in the live demo.

# Summary: setting the stage

1. Create flex container using **display:flex**, children become flex items
2. Set its **flex-direction** to control orientation (horizontal or vertical)
3. Set its **flex-wrap** to control whether and in which direction to wrap  
(Or, set **flex-flow** as shorthand for **flex-direction** and **flex-wrap**)

# Problems with flex-wrap

- Firefox before version 28 didn't support it
- No browser supports 2009 equivalent **box-lines** property
- Limited control of where breaks occur without support for **break-before/break-after** properties (only IE 10+ and Opera support them)

# Demo site:

[www.smoresday.us](http://www.smoresday.us)

Use Chrome, Opera,  
Firefox 28, or IE 11  
for full effect

(Zoe really loves s'mores)



[HOME](#) [S'MORES BUILDER](#)

**s'mores**  
Celebrate National S'mores Day Every Day

[THROW A PARTY](#) [TUMBLR](#)

## S'MORES BUILDER

This is your chance to get creative. As long as you have a roasted marshmallow sandwiched between something with some chocolate added, I say you got a s'mores. So pick your frame, marshmallow, and candy, add an optional accessory or two, and build a crazy s'mores concoction.

FRAME	MARSHMALLOW	CANDY	ACCESSORIES
<input type="radio"/> graham crackers <input type="radio"/> cinnamon graham <input type="radio"/> chocolate graham <input type="radio"/> Fudge Stripe cookies <input type="radio"/> Oreos <input type="radio"/> chocolate chip cookies <input type="radio"/> Ritz crackers <input type="radio"/> other: <input type="text"/>	<input type="radio"/> plain marshmallow <input type="radio"/> chocolate 'mallow <input type="radio"/> strawberry 'mallow <input type="radio"/> giant 'mallow <input type="radio"/> other: <input type="text"/>	<input type="radio"/> Hershey's chocolate <input type="radio"/> dark chocolate <input type="radio"/> chocolate with almonds <input type="radio"/> Nutella <input type="radio"/> Reese's P.B. Cup <input type="radio"/> white chocolate <input type="radio"/> other: <input type="text"/>	<input type="radio"/> peanut butter <input type="radio"/> banana <input type="radio"/> strawberries <input type="radio"/> caramel sauce <input type="radio"/> cream cheese <input type="radio"/> bacon <input type="radio"/> other: <input type="text"/>

NAME YOUR S'MORES:  [BUILD IT](#)

### NEED SOME INSPIRATION?

<b>CLASSIC</b> plain graham crackers plain marshmallow Hershey's chocolate	<b>ELVIS</b> chocolate graham crackers plain marshmallow Hershey's chocolate peanut butter banana	<b>PB&amp;J</b> plain graham crackers plain marshmallow Hershey's chocolate peanut butter strawberries	<b>CHOCOLATE-COVERED STRAWBERRY</b> chocolate graham crackers plain marshmallow dark chocolate strawberries
<b>THANKSGIVING CASSEROLE</b> cinnamon graham crackers plain marshmallow Hershey's chocolate sweet potato puree	<b>NUTTY</b> almond thins cookies plain marshmallow Nutella	<b>BANANA PUDDING</b> Nilla Wafers plain marshmallow white chocolate banana	<b>THE MONSTER</b> chocolate graham crackers chocolate marshmallow Reese's peanut butter cup caramel sauce
<b>SALTY-SWEET</b> Ritz crackers plain marshmallow Hershey's chocolate peanut butter	<b>TRIPLE CHOCOLATE</b> chocolate graham crackers chocolate marshmallow dark chocolate	<b>FRUIT SALAD</b> plain graham crackers plain marshmallow white chocolate banana strawberries	<b>COOKIES 'N' CREAM</b> Oreos plain marshmallow Hershey's Cookies 'n' Creme bar

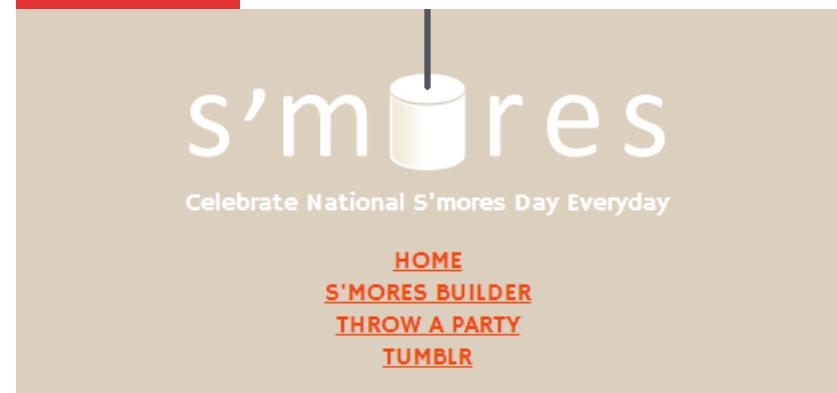
[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [Next](#)

# 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;  
}
```

Before



After



# Demo: horizontal navigation

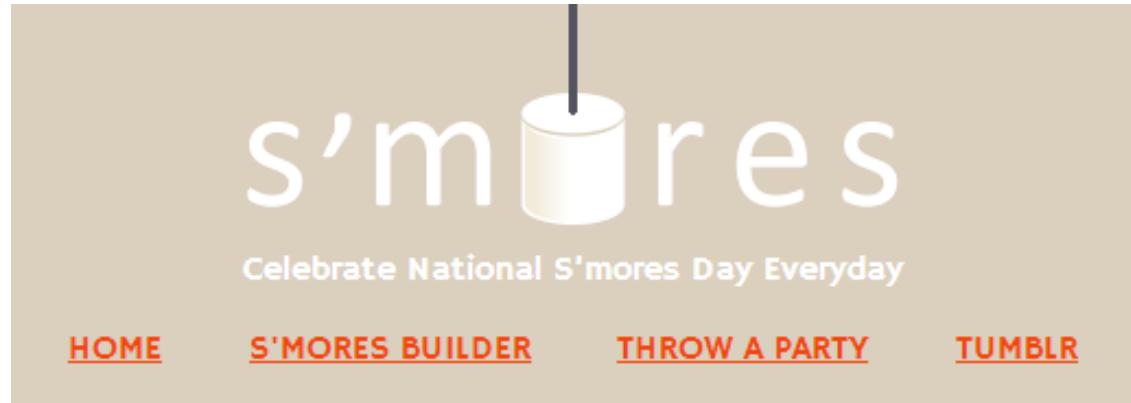
1. Turn <ul> into flex container:

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

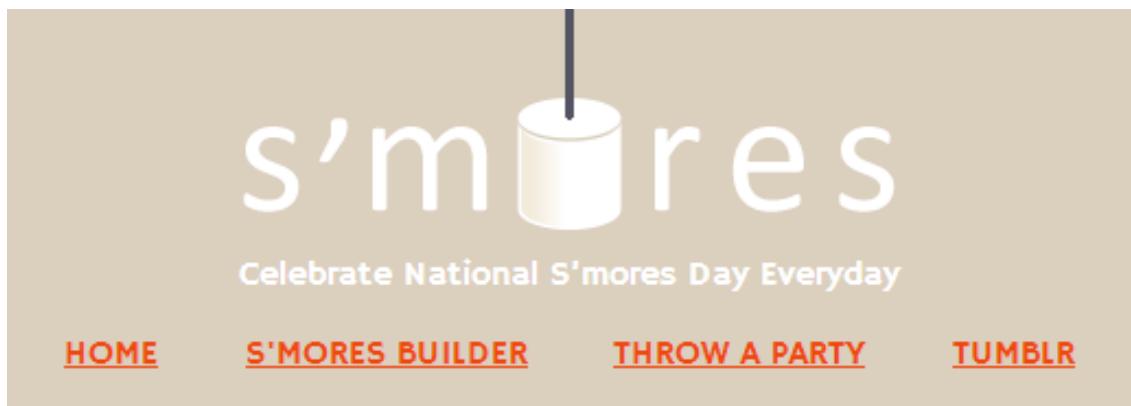
2. Children <li> become flex items laid out on single horizontal line

# Demo: horizontal navigation

Non-flexbox  
fallback version



Flexbox version



# Making it full-width

- Requirements:
  - All links on same line
  - First link flush left, last link flush right
  - Equal spaces between all links
- Using `display:table-cell` can do full-width but not equal spaces

# Spacing with table-layout:fixed

S'MORES

ARE VERY GOOD

YOU SHOULD TRY THEM

HUNGRY

YUM

YUM

## CONTENT

S'MORES

ARE VERY GOOD

YOU SHOULD TRY THEM

HUNGRY

YUM

YUM

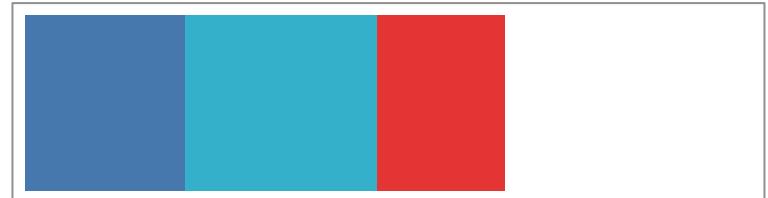
## CONTENT

# Nav with flexbox

```
.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



**flex-start**  
(default)



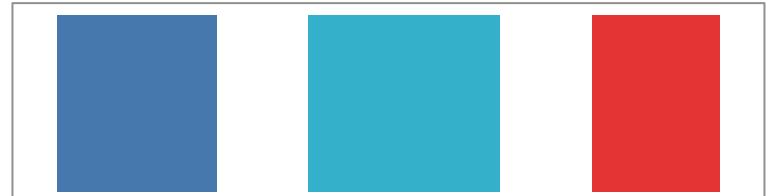
**center**



**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

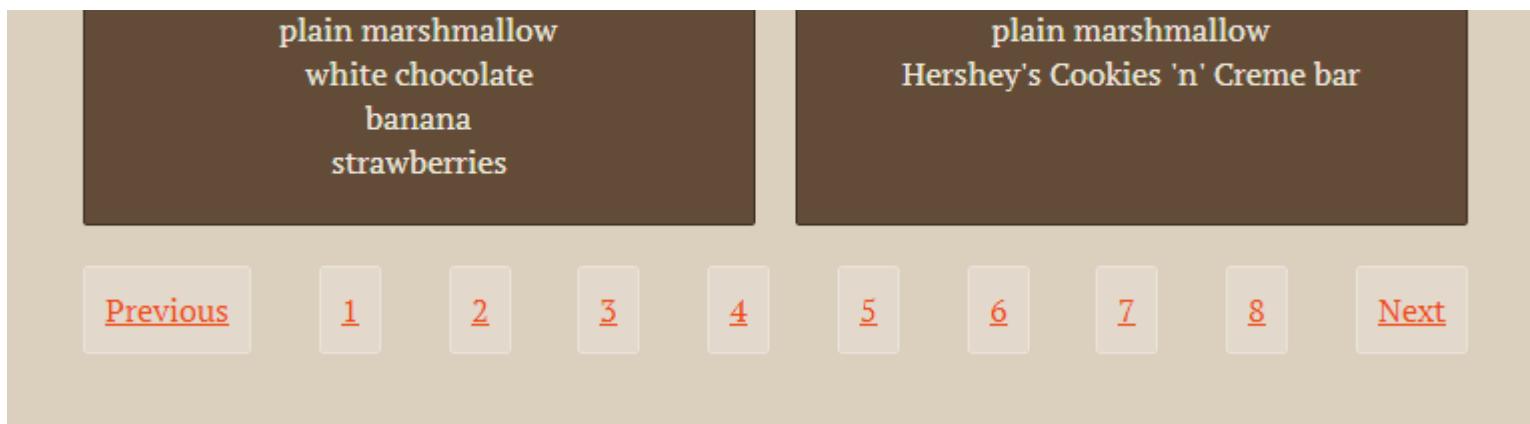
The screenshot shows the S'mores website with a tan background. At the top is a large logo 's'mores' where the 'o' is replaced by a white marshmallow on a stick. Below the logo is the tagline 'Celebrate National S'mores Day Every Day'. A horizontal navigation bar below the tagline contains four items: [HOME](#), [S'MORES BUILDER](#), [THROW A PARTY](#), and [TUMBLR](#). A large section title 'S'MORES BUILDER' is centered below the navigation. A descriptive text box at the bottom states: 'This is your chance to get creative. As long as you have a roasted marshmallow'.

Flexbox version

The screenshot shows the same S'mores website as above, but with improved layout due to flexbox. The text 'This is your chance to get creative. As long as you have a roasted marshmallow' is now positioned directly beneath the 'S'MORES BUILDER' title, without the need for a separate text box.

# Variation: pagination

- Wide view: all links on same line, centered
  - Set **justify-content:center**
- Medium view: all links on same line, full-width, equal spacing
  - Set **justify-content:space-between**



# Variation: stacked icons



## S'MORES BROWNIES RECIPE

Etiam sed convallis tellus. Pellentesque odio magna, pharetra in ligula tincidunt, ullamcorper pharetra quam. Donec semper imperdiet quam sed sodales. Suspendisse sed mollis quam. Morbi feugiat quis justo eget elementum. Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellentesque, dui justo imperdiet nisl, id cursus dolor lectus at felis. Proin a massa mollis, imperdiet augue vitae, egestas nunc. Donec ullamcorper fermentum auctor. Donec vulputate elit enim, ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorper viverra, neque ipsum fringilla augue, eget condimentum ipsum lectus non orci.

.icons

.content

.wrapper

# Demo: stacked icons

1. Turn .wrapper into flex container:

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

2. Children .icons and .content become side-by-side, equal-height flex items

# align-items

aligns flex items in cross axis



**stretch**  
(default)



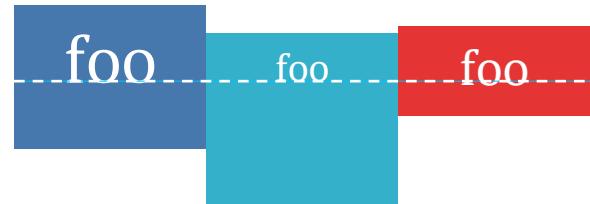
**flex-start**



**flex-end**



**center**



**baseline**

# Demo: stacked icons

3. Turn `.icons` into flex container with equally spaced, vertically stacked children:

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

# Demo: stacked icons



## CLASSIC S'MORES

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sagittis hendrerit dignissim. Integer at pellentesque magna. Proin est mauris, elementum eu euismod sed, condimentum quis ipsum. Vivamus sed leo massa. Nullam porta tempor turpis, in ornare nulla cursus nec. Quisque at erat pharetra, iaculis justo nec, sollicitudin magna. Praesent non aliquet metus. Phasellus euismod lobortis libero et pretium. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Vivamus rhoncus, dolor et viverra congue, nibh enim mollis orci, ac ultrices diam magna non lacus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Etiam non nunc facilisis elit feugiat ornare nec congue magna.



## S'MORES COOKIE BARS

Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellentesque, dui justo imperdiet nisl, id cursus dolor lectus at felis. Proin a massa mollis, imperdiet augue vitae, egestas nunc. Donec ullamcorper fermentum auctor. Donec vulputate elit enim, ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorper viverra, neque ipsum fringilla augue, eget condimentum ipsum lectus non orci.



## S'MORES BROWNIES RECIPE

Etiam sed convallis tellus. Pellentesque odio magna, pharetra in ligula tincidunt, ullamcorper pharetra quam. Donec semper imperdiet quam sed sodales. Suspendisse sed mollis quam. Morbi feugiat quis justo eget elementum. Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellentesque, dui justo imperdiet nisl, id cursus dolor lectus at felis. Proin a massa mollis, imperdiet augue vitae, egestas nunc. Donec ullamcorper fermentum auctor. Donec vulputate elit enim, ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorper viverra, neque ipsum fringilla augue, eget condimentum ipsum lectus non orci.

# 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 */  
}
```

# Add Modernizr as needed

- Flexbox and fallback styles can often co-exist, but sometimes need to isolate them
- Modernizr can add flexbox, no-flexbox, and flexboxlegacy classes to do this
- Example: add margin between icons and content only if flexbox is off and floating on:

```
.no-flexbox .content {  
    margin-left: 60px;  
}
```

# Fallback alignment options



## CLASSIC S'MORES

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



## S'MORES COOKIE BARS

Pellentesque nec faucibus tellus. Nam scelerisque, nisl eget semper, dolor lectus at felis. Proin a massa mollis, imperdiet augue vitae, 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. Maecenas faucibus tellus. Nam scelerisque, nisl eget sollicitudin pellentesque. Proin a massa mollis, imperdiet augue vitae, egestas nunc. Donec ac eleifend odio egestas et. Nulla egestas, dui nec ullamcorper, ipsum lectus non orci.

## CLASSIC S'MORES

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



## S'MORES COOKIE BARS

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



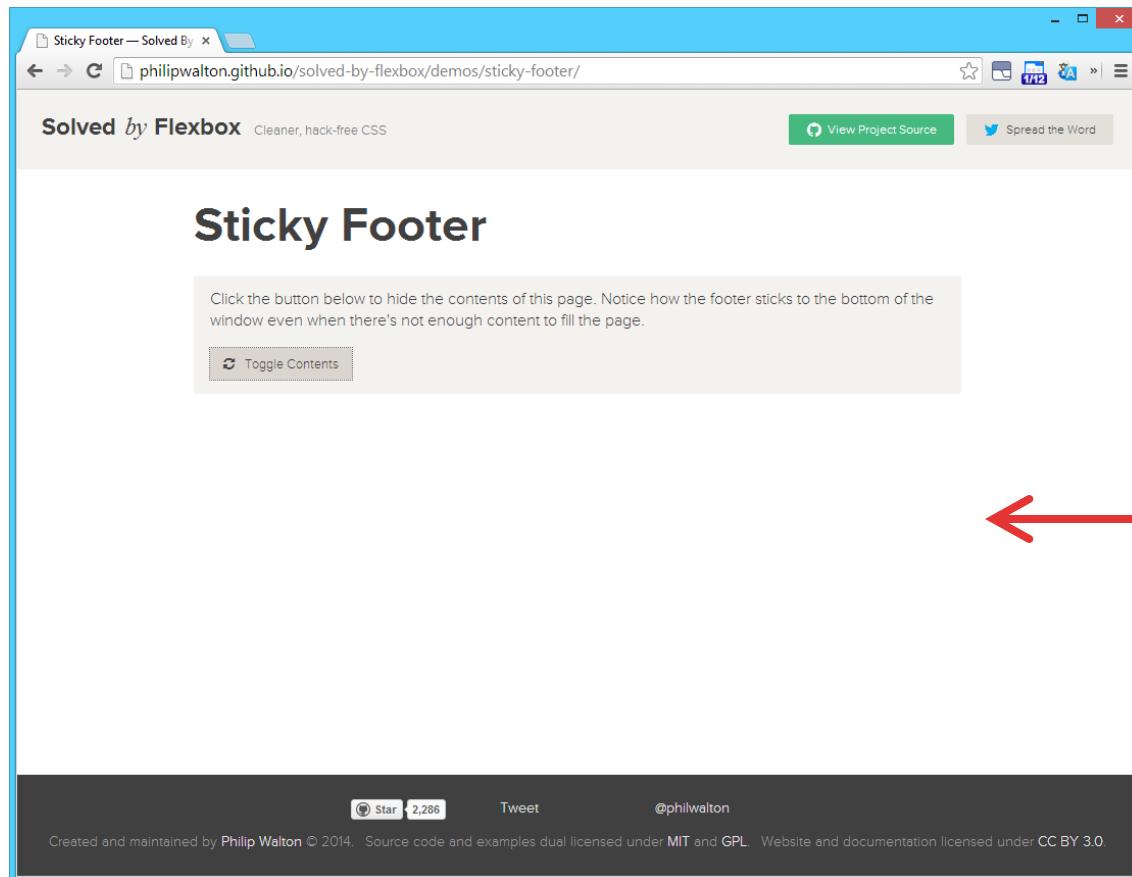
## S'MORES BROWNIES RECIPE

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

# Enhancement idea #1

Equally spaced items across full-width/height

# Demo: sticky footer



No content here, but footer stays at bottom of window anyway

Demo by Philip Walton at <http://philipwalton.github.io/solved-by-flexbox/demos/sticky-footer/>

# Demo: sticky footer

```
<body class="Site">
  <header>...</header>
  <main class="Site-content">...</main>
  <footer>...</footer>
</body>
```

```
.Site {
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}

.Site-content {
  flex: 1;
```

# The flex property

This is where flexbox gets flexible.  
And kinda confusing.

# Defining the flex property

- Makes flex items change their width or height (whichever is dimension along main axis) to fit available space
- Shorthand for 3 properties:
  - **flex-grow**
  - **flex-shrink**
  - **flex-basis**

# 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 it down

```
.gallery-item {  
  flex: 1 0 200px;  
}
```



**flex-grow = 1**

give each item 1 share of extra width

**flex-shrink = 0**

don't let items shrink at all

**flex-basis = 200px**

start items at 200 pixels wide

# Single-digit flex values

- Common to see `flex: 1` in demos
- `flex: [number]` is equivalent to  
`flex: [number] 1 0px`
- Be sure you really want `flex-basis` to be `0`
  - ▣ When wrap on, essentially min-width
  - ▣ `0px` therefore means items can shrink to `0px`
  - ▣ If everything can get down to `0px`, nothing ever has a reason to wrap

# Single-digit flex values

## Zoe's Brain Said:

“Since .action starts out at 100%, it won’t have space to sit on the first line with the content preceding it, and will wrap to a second line.”

```
.component {  
  flex: 1;  
}  
  
.action {  
  flex: 1 1 100%;  
}
```

The expected outcome:

FRAME

- graham crackers
- cinnamon graham
- chocolate graham
- Fudge Stripe cookies
- Oreos
- chocolate chip cookies
- Ritz crackers
- other:

MARSHMALLOW

- plain marshmallow
- chocolate 'mallow
- strawberry 'mallow
- giant 'mallow
- other:

CANDY

- Hershey's chocolate
- dark chocolate
- chocolate with almonds
- Nutella
- Reese's P.B. Cup
- white chocolate
- other:

ACCESSORIES

- peanut butter
- banana
- strawberries
- caramel sauce
- cream cheese
- bacon
- other:

NAME YOUR S'MORES:

BUILD IT

# Flexbox fail

## S'MORES BUILDER

This is your chance to get creative. As long as you have a roasted marshmallow sandwiched between something with some chocolate added, I say you got a s'mores. So pick your frame, marshmallow, and candy, add an optional accessory or two, and build a crazy s'mores concoction.



BUILDING S'MORES:  **BUILD IT**

- graham crackers
- cinnamon graham
- chocolate graham
- Fudge Stripe cookies
- Oreos
- chocolate chip cookies
- Ritzers: crackers
- other:

# What was really happening

## Reality:

Since it's fine for each .component to shrink to only 0px wide, a 100% wide element can and will sit on the same line as all the components.

```
.component {  
  flex: 1 1 0px;  
}  
  
.action {  
  flex: 1 1 100%;  
}
```

# Forcing the wrap

## Fixed:

.action will always wrap to new line, and .components will wrap to additional lines when there's less space than their combined flex-basis values (plus margin, etc.).

```
.component {  
  flex: 1 1 200px;  
}  
  
.action {  
  flex: 1 1 100%;  
}
```



# Why flex is great

# Less CSS & media queries needed

Layout changes that would previously have been hardcoded into media queries can now be done on the fly when browser determines stuff can fit

The image displays four mobile device screenshots of a "NAME YOUR S'MORES" application. Each screen shows a different layout of components based on the screen width, demonstrating how flexbox and media queries handle layout changes.

- Leftmost Device:** Shows a single column layout with three sections: "FRAME" (with graham crackers, cinnamon graham, chocolate graham, Fudge Stripe cookies, Oreos, chocolate chip cookies, Ritz crackers, other), "MARSHMALLOW" (with plain marshmallow, chocolate 'mallow, strawberry 'mallow, giant 'mallow, other), and "CANDY" (with Hershey's chocolate, dark chocolate, chocolate with almonds, Nutella, Reese's P.B. Cup, white chocolate, other).
- Second Device from Left:** Shows a two-column layout. The first column contains "FRAME" and "MARSHMALLOW" sections. The second column contains "CANDY" and "ACCESORIES" sections. The "ACCESORIES" section includes peanut butter, banana, strawberries, caramel sauce, cream cheese, bacon, and other.
- Third Device from Left:** Shows a three-column layout. The first column contains "FRAME". The second column contains "MARSHMALLOW" and "CANDY" sections. The third column contains "ACCESORIES" section.
- Rightmost Device:** Shows a four-column layout. The first column contains "FRAME". The second column contains "MARSHMALLOW". The third column contains "CANDY". The fourth column contains "ACCESORIES" section.

**Action Buttons:** Each screen features a "NAME YOUR S'MORES:" input field and a "BUILD IT" button at the bottom.

**Code Examples:**

```
.builder { display: flex; flex-wrap: wrap; }

.component { flex: 1 1 200px; }

.action { flex: 1 1 100%; }
```

# Flex adjusts for margin

box-sizing only takes care of padding and border added on to width, not margin

The boxes won't all fit

```
.component {  
  width: 25%;  
  margin-left: 20px;  
}
```

Works like a charm

```
.component {  
  flex: 1 1 200px;  
  margin-left: 20px;  
}
```

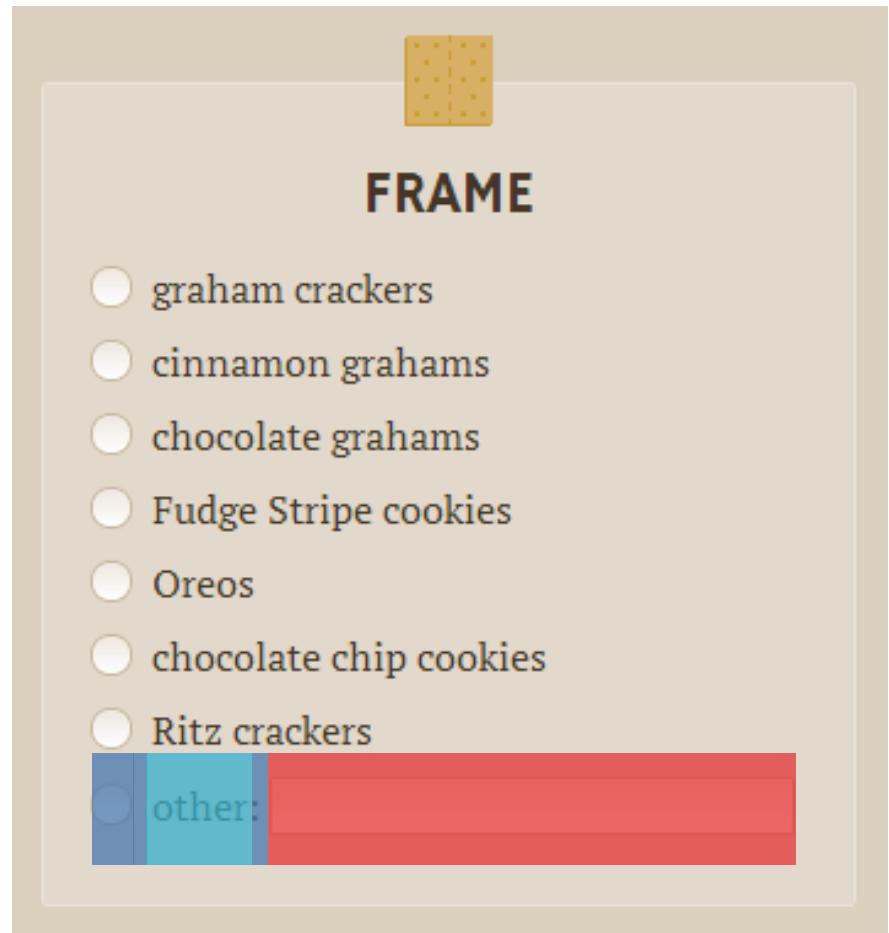
# Flex adjusts for quantity of items

- Great for sites with dynamic or frequently changing content blocks, e.g.:
  - ▣ News stories on home page vs inner page
  - ▣ Product or feature tiles
  - ▣ Pricing page with plans side by side
- Makes HTML/CSS more modular—an item can move anywhere and adjust in size as needed

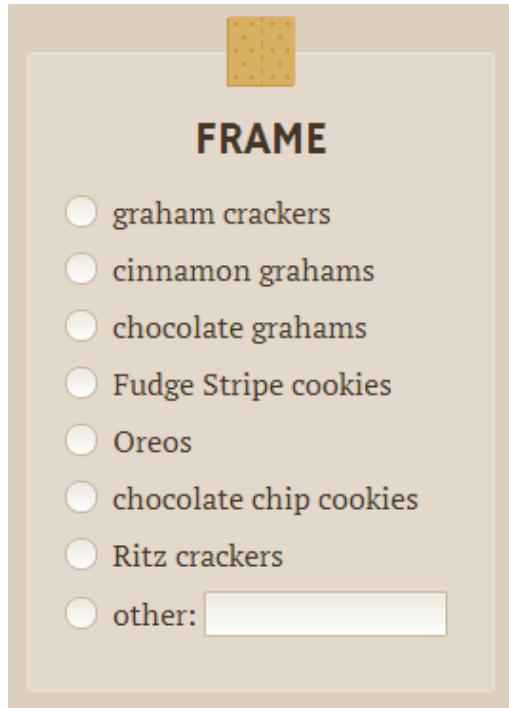
# Flex can combine different units

Items measured in different units can sit side-by-side and all fit perfectly

- Pixels
- Ems
- Mystery percentage

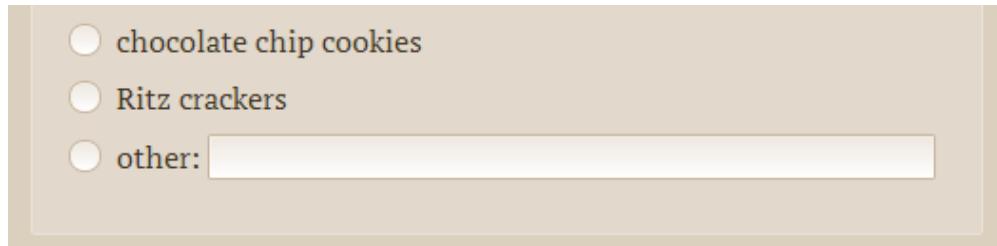


# Flex can combine different units



Set only the text field to flex:

```
.component li:last-child {  
  display: flex;  
}  
  
.component .other-name {  
  flex: 1;  
}
```



# Flex can be proportional

Setting `flex-grow`/`flex-shrink` to different values can make flex items size themselves relative to each other

```
flex: 1;    flex: 1;    flex: 2;
```



# But be careful!

Having widths be in multiples of each other only works if `flex-basis` is 0

```
flex: 1 0 0px; flex: 1 0 0px; flex: 2 0 0px;
```



If all start out 0px, then **all** the width on the line is extra, so the `flex: 2` item gets twice as much width as the others and is thus twice as wide as the others

# If flex-basis isn't 0px...

...the widths may not end up as you expect

```
flex: 1 0 10px;    flex: 1 0 10px;    flex: 2 0 10px;
```

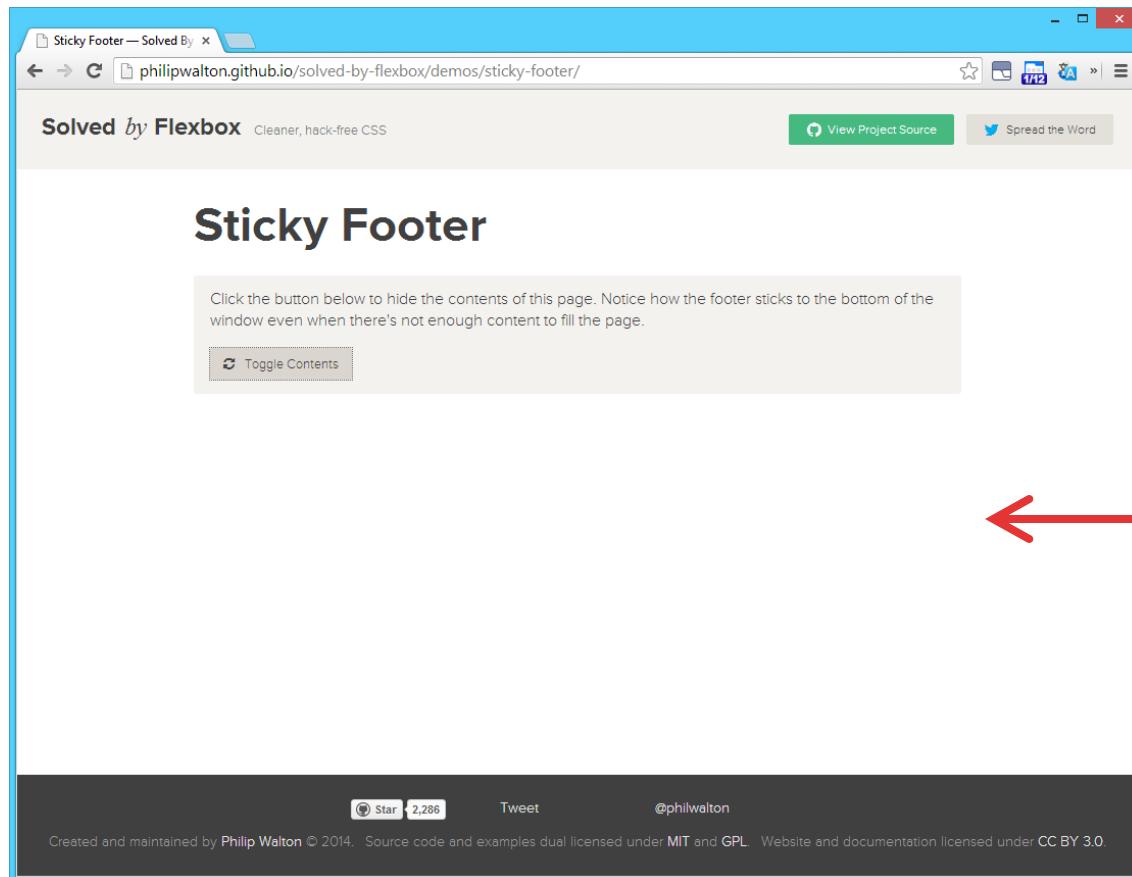


10px + 5px extra = 15px    10px + 5px extra = 15px    10px + 10px extra = 20px

if 50px available

The third box gets twice as much of the **extra**,  
but that doesn't make it twice as wide **overall**  
(20px is not twice as wide as 15px. Because math.)

# Demo: sticky footer



flex:1 on  
content div  
stretches it to  
fill leftover  
height

Demo by Philip Walton at <http://philipwalton.github.io/solved-by-flexbox/demos/sticky-footer/>

# New “auto” margin behavior

```
<body class="Site">
  <header>...</header>
  <main class="Site-content">...</main>
  <footer>...</footer>
</body>

.Site {
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}

.Site-content {
  flex: 1;
}

.footer {
  margin-top: auto;
}
```

CSS modified from Philip Walton's original

# New “auto” margin behavior

- Margins set to auto in main axis get all the free space left
- For instance: to pin flex item to bottom of its flex container:
  - set **flex-direction: column** on flex container so items can fill its height
  - set **margin-top: auto** on item you want to pin

# Pinned item at bottom

- All boxes equal in height
- Final item in each box pinned to the bottom so that all final items across grid appear to align

Different plans for different product needs

**Head Start**

Design review of your product

UI / UX action items

Framework for next steps

**Fast Track**

Design review of your product

UI / UX action items

Framework for next steps

Product development workshop

**Co-Bolt**

Design review of your product

UI / UX action items

Framework for next steps

Product development workshop

Finished design incl. logo, PSDs & UI kit

Front-end templates (built with Twitter Bootstrap)

**ENQUIRE**

**ENQUIRE**

**ENQUIRE**

# Pinned item at bottom

The image shows a user interface with four categories: FRAME, MARSHMALLOW, CANDY, and ACCESSORIES. Each category has a title with an icon above it and a list of items with radio buttons. There is also an 'other:' input field for each category.

FRAME	MARSHMALLOW	CANDY	ACCESSORIES
<input type="radio"/> graham crackers <input type="radio"/> cinnamon graham <input type="radio"/> chocolate graham <input type="radio"/> Fudge Stripe cookies <input type="radio"/> Oreos <input type="radio"/> chocolate chip cookies <input type="radio"/> Ritz crackers <input type="radio"/> other: <input type="text"/>	<input type="radio"/> plain marshmallow <input type="radio"/> chocolate 'mallow <input type="radio"/> strawberry 'mallow <input type="radio"/> giant 'mallow <input type="radio"/> other: <input type="text"/>	<input type="radio"/> Hershey's chocolate <input type="radio"/> dark chocolate <input type="radio"/> chocolate with almonds <input type="radio"/> Nutella <input type="radio"/> Reese's P.B. Cup <input type="radio"/> white chocolate <input type="radio"/> other: <input type="text"/>	<input type="radio"/> peanut butter <input type="radio"/> banana <input type="radio"/> strawberries <input type="radio"/> caramel sauce <input type="radio"/> cream cheese <input type="radio"/> bacon <input type="radio"/> other: <input type="text"/>

- ❑ Without flexbox, “other” fields disconnected from each other and thus can’t align
- ❑ With flexbox, they’re still disconnected, but their parents aren’t and can be equal height

# Pin the “other” fields

1. Make each `.component` match in height by making parent `.builder` a flex container

```
.builder {  
  display: flex;  
  align-items: stretch; /* default */  
  flex-wrap: wrap;  
  justify-content: space-between;  
  margin: 0 0 40px -20px;  
}
```

# Pin the “other” fields

2. Make each <ul> a flex item and stretch to full height, then make it a flex container with vertical direction so its <li> will stack

```
.component {  
  flex: 1 1 200px;  
  display: flex;  
  flex-direction: column;  
}
```

```
.component ul {  
  flex: 1;  
  display: flex;  
  flex-direction: column;  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}
```

# Pin the “other” fields

3. Give “other” <li> an auto top margin so all free space left in <ul> is put above that <li>, pushing it to bottom of <ul>

```
.component li:last-child {  
  margin-top: auto;  
}
```

# Pinning without flexbox

1. Use `display:table-cell` to create equal height boxes
2. Add bottom padding in ems to each box
3. Use absolute positioning to pin “other” row in space left by padding

# Variation: two-piece main nav



```
.flexbox .list-nav {  
  display: flex;  
  justify-content: flex-start;  
  position: relative;  
  top: -70px;  
}  
.flexbox #link-home { margin-right:20px; }  
.flexbox #link-tumblr { margin-left:20px; }  
  
.flexbox #link-party {  
  margin-left: auto;  
}
```

# Enhancement idea #2

Pinned items

# Single-line, full-width form

SIGN IN TO SAVE, SHARE AND CHECKOUT FASTER.

[CREATE ACCOUNT](#) Email Password [SIGN IN](#) OR  [LOGIN WITH FACEBOOK](#)

[Forgot your Password?](#)

Try searching a keyword or selecting a color  [Search](#)

- All items on same line
- Text input(s) stretches to fill remaining space
- All items vertically centered or equal height

# Form without flexbox

```
.action { text-align: center; }  
.action * {  
  display: inline; /* default */  
  vertical-align: middle;  
}  
  
NAME YOUR S'MORES:  BUILD IT
```

## NEED SOME INSPIRATION?

### CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

### ELVIS

chocolate graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
banana

### PB&J

plain graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
strawberries

### CHOCOLATE-COVERED STRAWBERRY

chocolate graham crackers  
plain marshmallow  
dark chocolate  
strawberries

# Form without flexbox

- ✓ All items on same line
- ✗ Text input stretches to take up remaining space
- ✓ All items vertically centered or equal height

NAME YOUR S'MORES:  BUILD IT

NEED SOME INSPIRATION?

## CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

## ELVIS

chocolate graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
banana

## PB&J

plain graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
strawberries

## CHOCOLATE-COVERED STRAWBERRY

chocolate graham crackers  
plain marshmallow  
dark chocolate  
strawberries

# Form with flexbox

```
.action {  
  flex: 1 1 100%;  
  display: flex;  
  align-items: stretch; /* default */  
}  
.action input {  
  flex: 1;  
}  
.action input, .action label {  
  margin-right: 10px;  
}
```

# Form with flexbox

- ✓ All items on same line
- ✓ Text input stretches to take up remaining space
- ✓ All items vertically centered or equal height

NAME YOUR S'MORES:

BUILD IT

## NEED SOME INSPIRATION?

### CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

### ELVIS

chocolate graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
banana

### PB&J

plain graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
strawberries

### CHOCOLATE-COVERED STRAWBERRY

chocolate graham crackers  
plain marshmallow  
dark chocolate  
strawberries

# Override alignment on label

```
.action label {  
  align-self: center;  
}
```

NAME YOUR S'MORES:

BUILD IT

## NEED SOME INSPIRATION?

### CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

### ELVIS

chocolate graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
banana

### PB&J

plain graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter  
strawberries

### CHOCOLATE-COVERED STRAWBERRY

chocolate graham crackers  
plain marshmallow  
dark chocolate  
strawberries

# Combine the two

```
.action {  
  flex: 1 1 100%;  
  display: flex;  
  text-align: center; /* fallback */  
}  
.action input {  
  flex: 1;  
}  
.action label {  
  align-self: center;  
}  
.action input, .action label {  
  margin-right: 10px;  
}
```

# Another option: stack, center

```
.action {  
  flex: 1 1 100%;  
  display: flex;  
  flex-wrap: wrap;  
  align-items: center;  
  text-align: center; /* fallback */  
}  
  
.action input {  
  flex: 1;  
  display: block; /* fallback */  
  width: 100%; /* fallback */  
  box-sizing: border-box; /* fallback */  
}  
  
.action button {  
  flex: 1 1 100%;  
  margin-top: 10px;  
}  
  
.flexbox .action label {  
  margin-right: 10px;  
}
```

# Narrow version

## Non-flexbox fallback version

NAME YOUR S'MORES:

BUILD IT

NEED SOME INSPIRATION?

CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

ELVIS

chocolate graham crackers

## Flexbox version

NAME YOUR S'MORES:

BUILD IT

NEED SOME INSPIRATION?

CLASSIC

plain graham crackers  
plain marshmallow  
Hershey's chocolate

ELVIS

chocolate graham crackers  
plain marshmallow  
Hershey's chocolate  
peanut butter

# Variation: input add-ons

Add-on Prepended

Amount	<input type="text"/>
Q	<input type="text"/>

Add-on Appended

<input type="text"/>	Go
<input type="text"/>	★

Appended and Prepended Add-ons

<input type="button" value="✉"/>	Example One	<input type="button" value="Send"/>	<input type="button" value="🔒"/>	Example One	<input type="button" value="Encrypt"/>
----------------------------------	-------------	-------------------------------------	----------------------------------	-------------	--

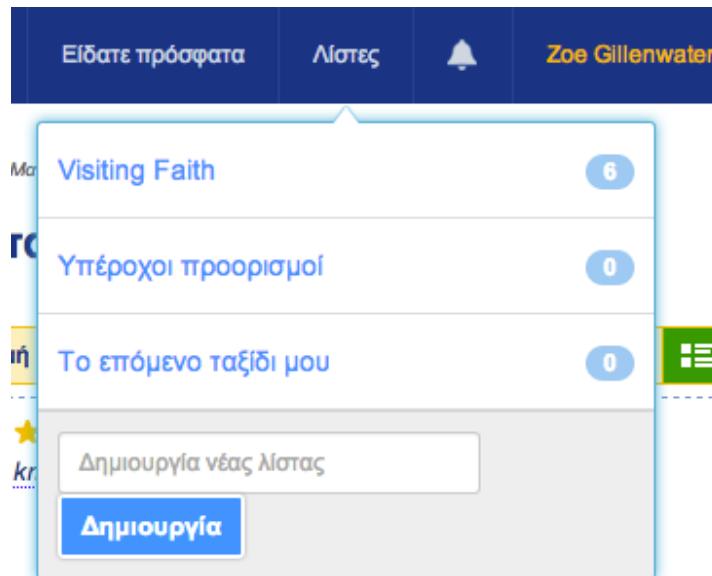


flex:1 on each input stretches it to fill width before/after any add-ons

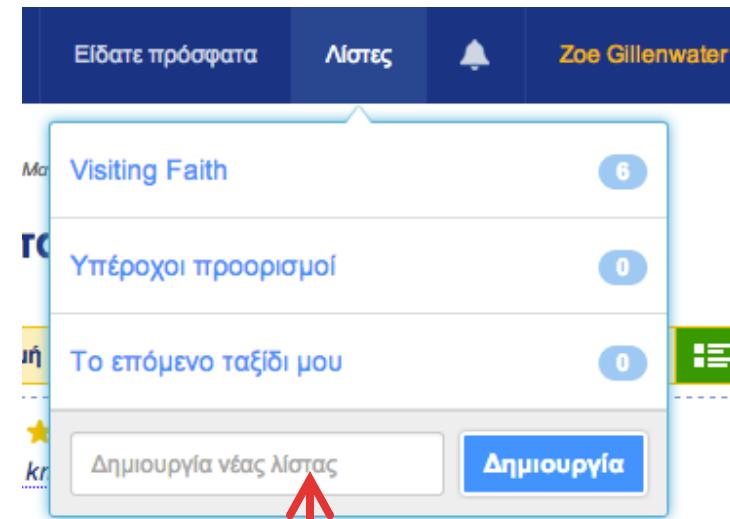
Demo by Philip Walton at <http://philipwalton.github.io/solved-by-flexbox/demos/input-add-ons/>

# How Booking.com uses it

## Before flexbox



## With flexbox



## New fallback



flex: 1 1 auto; on  
<input> lets it shrink to fit beside variable button

# Enhancement idea #3

Full-width forms

# Remember the pagination?

peanut butter

banana  
strawberries

[Previous](#)

[1](#)

[2](#)

[3](#)

[4](#)

[5](#)

[6](#)

[7](#)

[8](#)

[Next](#)

plain marshmallow  
white chocolate  
banana  
strawberries

plain marshmallow  
Hershey's Cookies 'n' Creme bar

[Previous](#)

[1](#)

[2](#)

[3](#)

[4](#)

[5](#)

[6](#)

[7](#)

[8](#)

[Next](#)

plain marshmallow  
Hershey's Cookies 'n' Creme bar

[Previous](#)

[Next](#)

[1](#)

[2](#)

[3](#)

[4](#)

[5](#)

[6](#)

[7](#)

[8](#)

# Visual reordering with flexbox

1. Make “previous” link come first visually, “next” link second, and all the rest third

```
.pagination li {  
    order: 2;  
    display: inline-block; /* fallback */  
}  
.pagination li:first-child { /* “Previous” link */  
    order: 0;  
    text-align: left;  
}  
.pagination li:last-child { /* “Next” link */  
    order: 1;  
    text-align: right;  
}
```

# Visual reordering with flexbox

Before

plain marshmallow  
Hershey's Cookies 'n' Creme bar

[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [Next](#)

After

plain marshmallow  
Hershey's Cookies 'n' Creme bar

[Previous](#) [Next](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#)

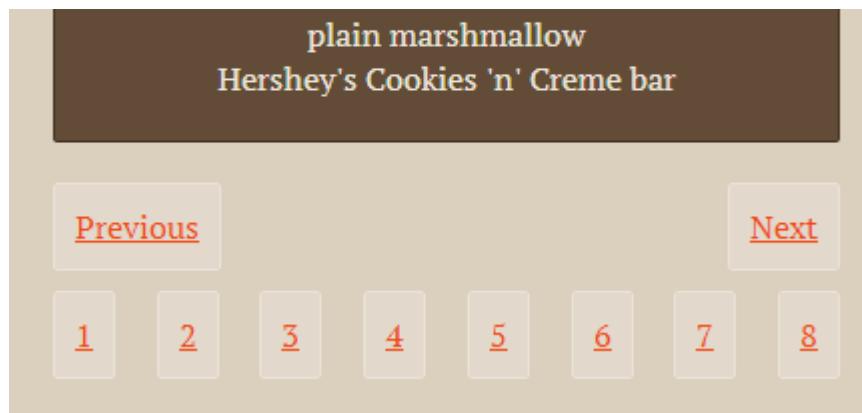
# Visual reordering with flexbox

2. Force links to wrap after “next” link by making it and “previous” link take up 100% of the first line together

```
.pagination {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: space-between;  
    text-align: center; /* fallback */  
}  
.flexbox .pagination li:first-child,  
.flexbox .pagination li:last-child {  
    width: 50%;  
}
```

# Demo: two-line pagination

- Narrow view: two lines with “previous” and “next” links on first row, full-width
  - Set `flex-wrap:wrap`
  - Set `justify-content:space-between`
  - Use `order` property to move “next” link up



# 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.

# Tab order = HTML order

“Next” won’t be second link tabbed to after “Previous” since it doesn’t follow it in HTML



# Exception: Firefox

Firefox changes tab order to match visual order when using flexbox to rearrange content



# Limitations of order property

- Potential to create confusing tab order
- Doesn't change reading order for screen readers
- Can only rearrange sibling elements
- Flexbox rows/cols can't overlap, so content may not always slide into the spot left by the re-ordered content

**So:** reserve flexbox order property for small moves that don't hurt accessibility, and use CSS3 Grid Layout, etc., for bigger re-ordering

# Trent Walton's CSS Zen Garden

```
<div class="page-wrapper">
  <section class="intro" id="zen-intro">...
  <div class="main supporting" id="zen-supporting">...
  <aside class="sidebar" role="complementary">...
</div>
```

```
.page-wrapper {
  display: flex;
  flex-direction: column;
}
.intro { order: 1; }
.main { order: 3; }
.sidebar { order: 2; }
```

See Trent's write-up at <http://blog.typekit.com/2013/11/26/typekits-second-css-zen-garden-theme-apothecary/>

# Flexbox version (reordered)



DR. SHEA'S MIRACULOUS ~

# CSS Zen Garden



THE BEAUTY OF CSS DESIGN

A demonstration of what can be accomplished through brilliant design. Select any one sheet from here to have it load into this page. Show off the example sheet you like.

## THE ROAD TO ENLIGHTENMENT

Learning a dark and dusty road by the pale light of browser compatibility, the CSS Zen Garden has emerged from the shadows. The CSS Zen Garden has been refined thanks to the tireless efforts of the W3C. Highly recommended.

The CSS Zen Garden has the tools to train and condition the beginner. It's a dark and dusty road, but the CSS Zen Garden is here to help you learn and prosper. Return to the CSS Zen Garden often to train and prosper.

### SELECT A DESIGN:

Germania — Sven Walz  
Athenaeum — Steven Hoober  
Saxon Pique — Eric Meyer  
Prairie Rose — Jason Beaird  
A Bohemian Jersey — Michael Scharf  
Vista Modern — Brad Neher  
Landscape — Matt Wilcox

[VIEW ALL DESIGNS](#)



SO WHAT IS THIS ABOUT?

This is a continuing work to show the power of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

css Zen Garden is continuing to do the great work of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

css Zen Garden is continuing to do the great work of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

## » PARTICIPATION «

Some cool things about our site. If you're reading this page, you're using CSS! We are massive fans of the simple elegance of CSS. We are also huge fans of the CSS Zen Garden as a starting point. Please see the [CSS Resources](#) for more information.

You may modify the site as often as you wish, but we ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

Demand the simple [HTML](#) and [CSS](#) to work on a copy. Once you've got that working, then add your own CSS. Don't submit half-baked work, and we'll be sick if you do. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

### BENEFITS

Why participate? Recognition, validation, and exposure are just a few of the perks for making CSS cool. By participating, you will be sharing your knowledge with others, learning from them, and becoming a part of something bigger than yourself.



## APPLY LIBERALLY AND REGULARLY!

### REQUIREMENTS

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

By [Sven Walz](#). Bestowed greatest honor by [Eric Meyer](#). You can help too: [designs](#), [links](#).



DR. SHEA'S MIRACULOUS ~

# CSS Zen Garden



THE BEAUTY OF CSS DESIGN

A demonstration of what can be accomplished through brilliant design. Select any one sheet from here to have it load into this page. Show off the example sheet you like.

## THE ROAD TO ENLIGHTENMENT

Learning a dark and dusty road by the pale light of browser compatibility, the CSS Zen Garden has emerged from the shadows. The CSS Zen Garden has been refined thanks to the tireless efforts of the W3C. Highly recommended.

The CSS Zen Garden has the tools to train and condition the beginner. It's a dark and dusty road, but the CSS Zen Garden is here to help you learn and prosper. Return to the CSS Zen Garden often to train and prosper.

### SO WHAT IS THIS ABOUT?

This is a continuing work to show the power of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

css Zen Garden is continuing to do the great work of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

css Zen Garden is continuing to do the great work of CSS. The CSS Zen Garden aims to teach, inspire, and entertain. It's a place where people can come to learn about CSS and those that are creating great things with it. The CSS Zen Garden is the place that has changed the world. You may never look at CSS the same again.

## » PARTICIPATION «

Some cool things about our site. If you're reading this page, you're using CSS! We are massive fans of the simple elegance of CSS. We are also huge fans of the CSS Zen Garden as a starting point. Please see the [CSS Resources](#) for more information.

You may modify the site as often as you wish, but we ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

Demand the simple [HTML](#) and [CSS](#) to work on a copy. Once you've got that working, then add your own CSS. Don't submit half-baked work, and we'll be sick if you do. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

### BENEFITS

Why participate? Recognition, validation, and exposure are just a few of the perks for making CSS cool. By participating, you will be sharing your knowledge with others, learning from them, and becoming a part of something bigger than yourself.



## APPLY LIBERALLY AND REGULARLY!

### REQUIREMENTS

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

When possible, we would like our visitors to have at least 1.5 or 2.0 range CSS 2.1 supported browsers only, though we do support IE5.5 and older. This is to ensure that the CSS Zen Garden is as close to browser neutral as possible. We ask that you keep the original design intact. We ask that you not break the site, but rather add the links to have more content added to the site.

By [Sven Walz](#). Bestowed greatest honor by [Eric Meyer](#). You can help too: [designs](#), [links](#).



# Non-flexbox (HTML 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

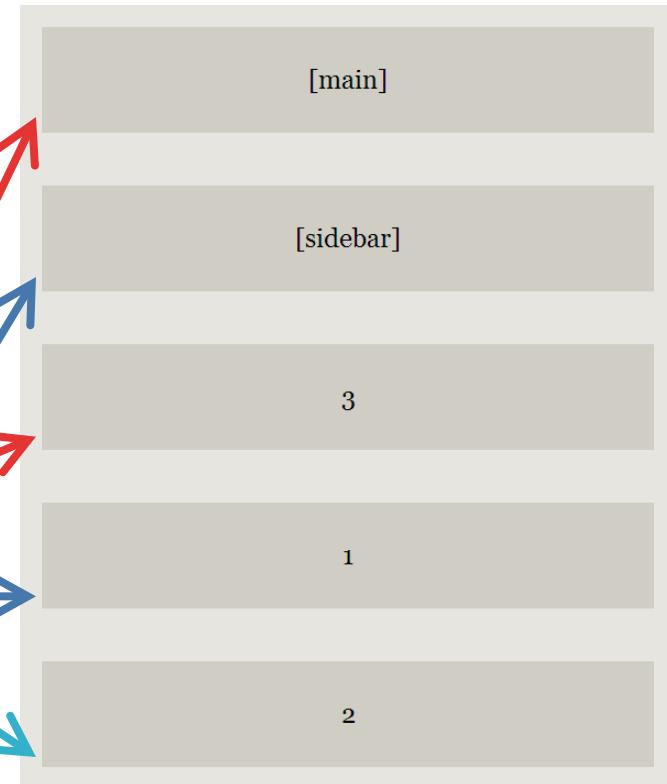
Keep HTML order needed for desktop and use `order` property only on mobile, since browser support is great there

# Jeremy Church's mobile example

HTML order (no flexbox)



Reordered with flexbox



```
@media screen and (max-width: 767px) {  
    .media_xs_order_one { order: 0; }  
    .media_xs_order_two { order: 1; }  
    .media_xs_order_three { order: 2; }  
}
```

See Jeremy's write-up at <http://j.eremy.net/flexbox-for-mobile-content/>

# Be careful with accessibility

- ❑ Reading order won't match visual order. This **may or may not** be bad.
- ❑ Tabbing order won't match visual order. This **is** bad.  
(And yes, tabbing matters even on mobile.)



# 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.

#### 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

## 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.
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 “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

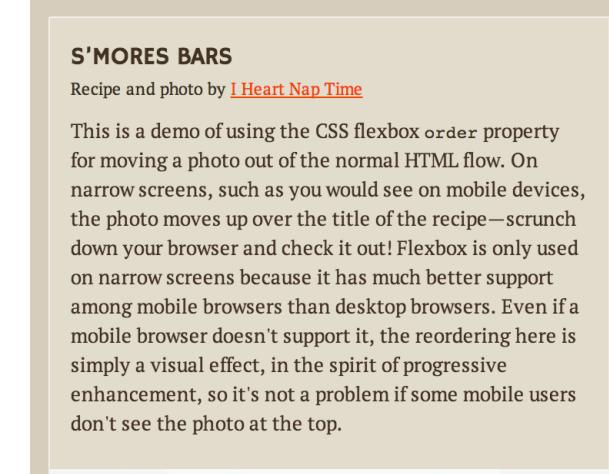
Recipe and photo by [I Heart Nap Time](#)

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

## 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

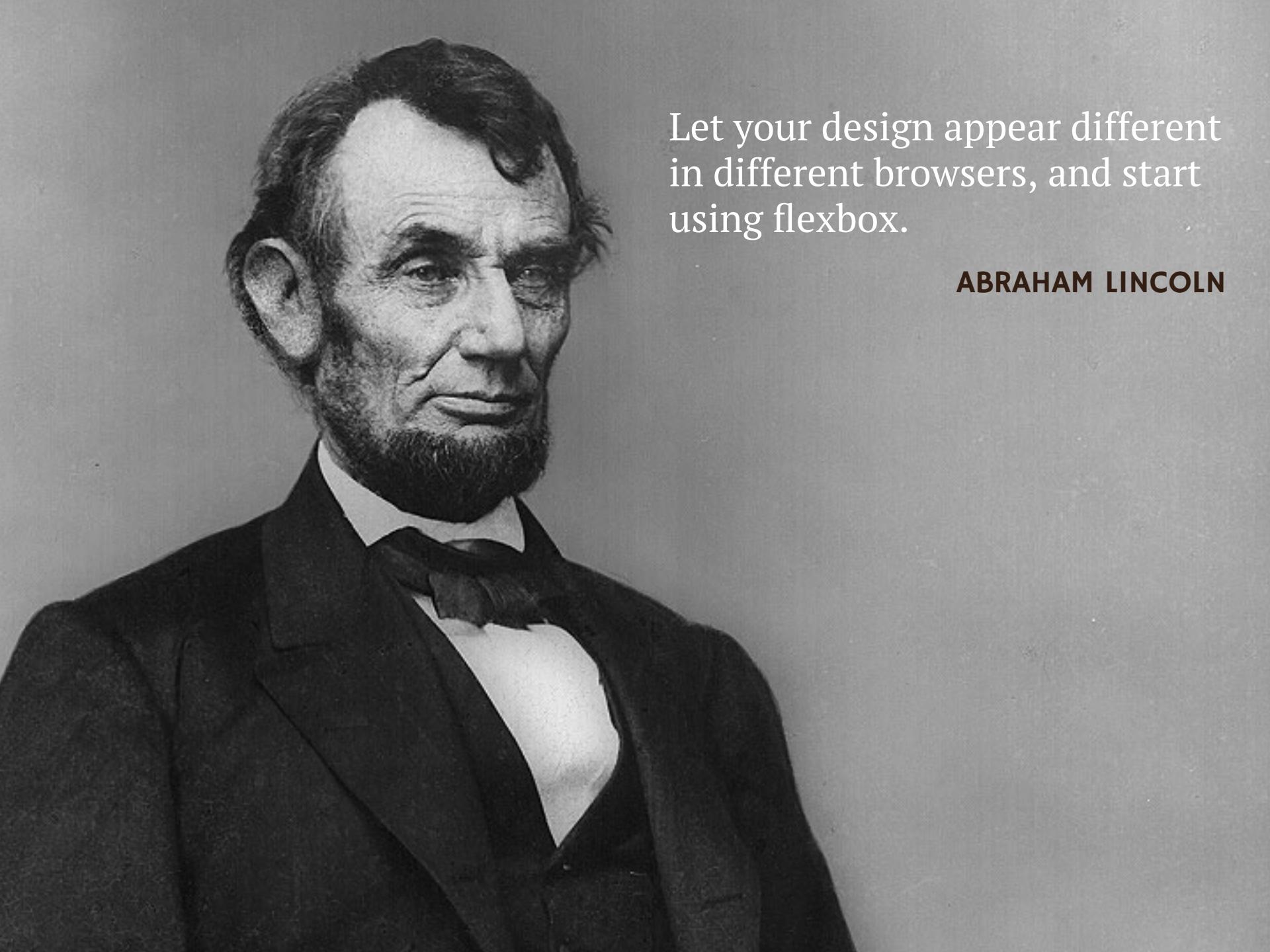
- 1/2 cup butter melted
- 1 1/2 cup graham cracker crumbs

## Enhancement idea #4

Reordering content blocks (esp. for mobile)

# Flexbox enhancement ideas

- Equally spaced items across full-width/height
- Pinned items
- Full-width forms
- Reordering content blocks

A black and white portrait of Abraham Lincoln, showing him from the chest up. He has dark hair and a full beard. He is wearing a dark suit jacket over a white collared shirt and a dark bow tie. The background is a plain, light color.

Let your design appear different  
in different browsers, and start  
using flexbox.

**ABRAHAM LINCOLN**

# Learn more

Download slides and get links at

[www.zomigi.com/blog/leveling-up-with-flexbox](http://www.zomigi.com/blog/leveling-up-with-flexbox)

Thanks!

Zoe Mickley Gillenwater

@zomigi

[design@zomigi.com](mailto:design@zomigi.com)

[zomigi.com](http://zomigi.com) | [stunningcss3.com](http://stunningcss3.com) | [flexiblewebbook.com](http://flexiblewebbook.com)

Credit: “S’more” photo by Christopher Penn on Flickr. Looks like that man really knows how to make a good s’mores!