



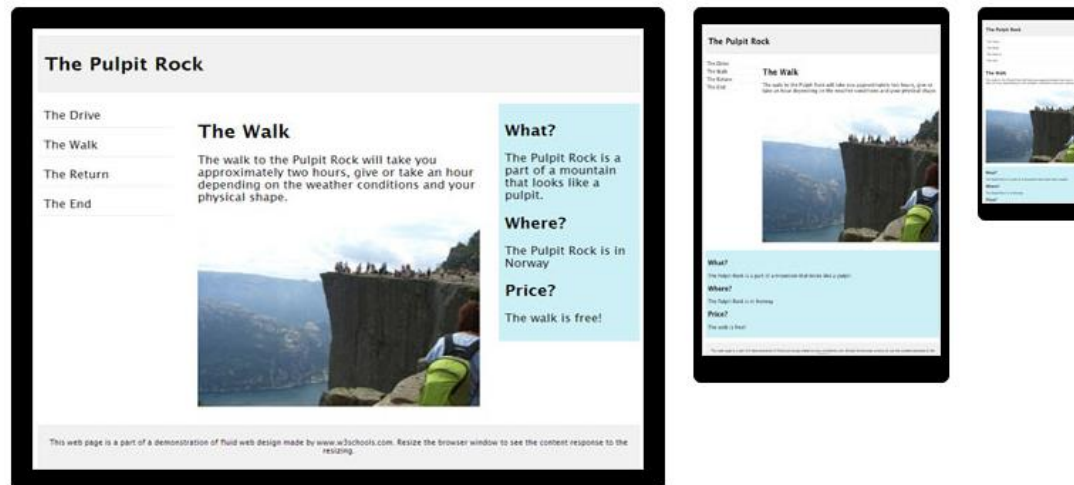
Agenda



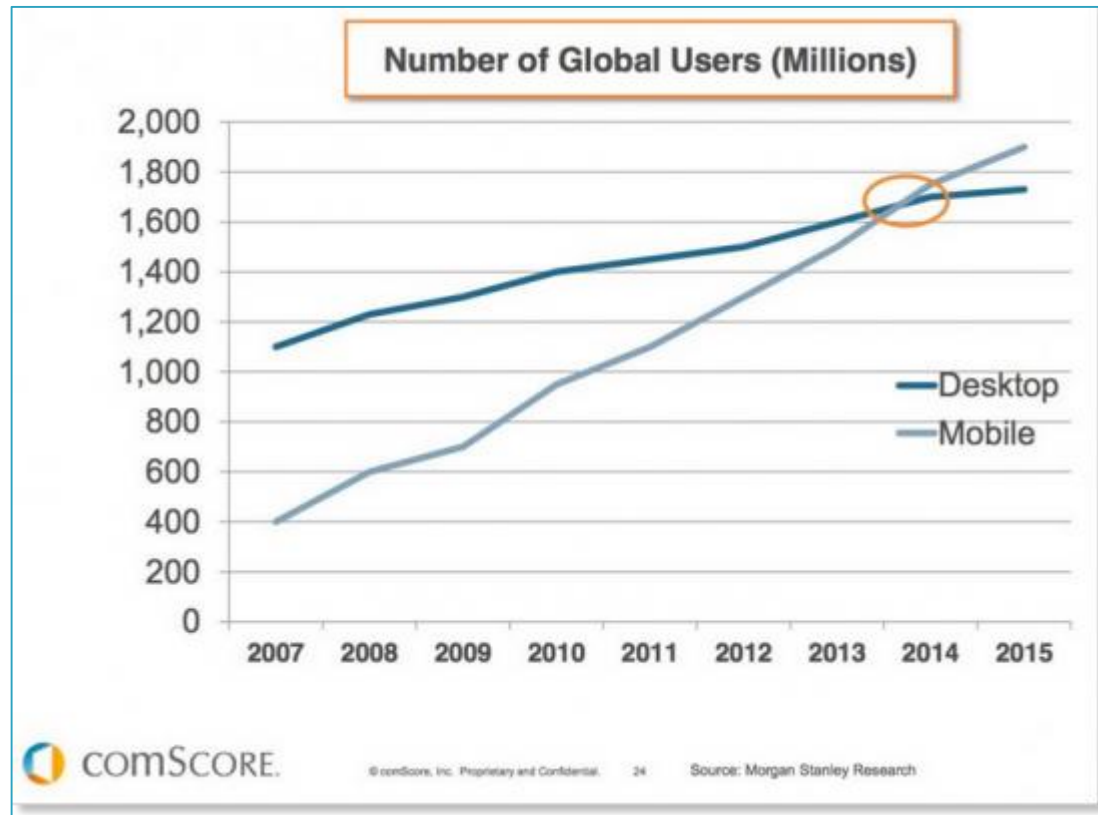
- What is responsive web design
- CSS and responsive design
 - Vertical align
 - Transform
 - Translate
- Hamburger menu
- Media queries
 - min/max-width

Reminder - Responsive Web Design?

- Make your app/website usable on any device, with any screen size



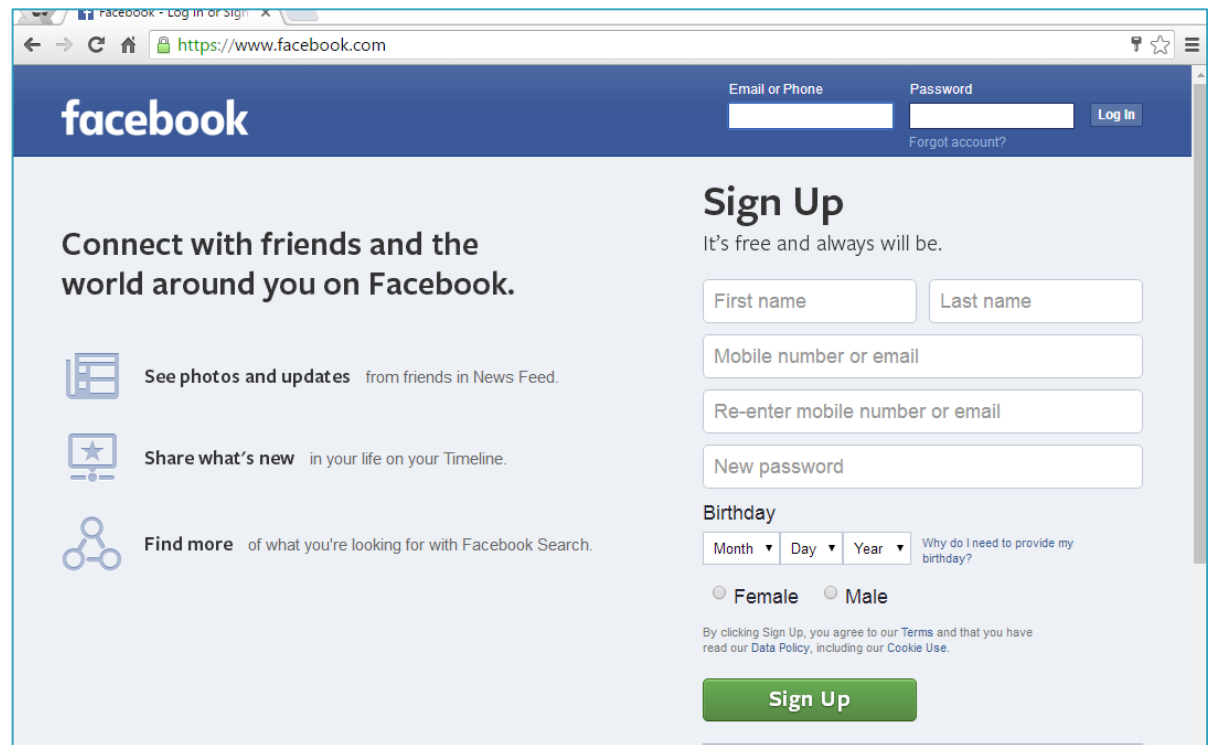
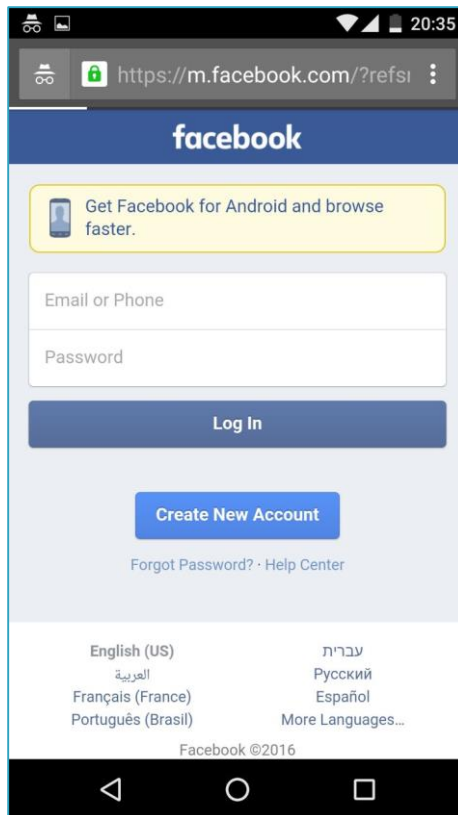
Why Bother?



Mobile sites

Optional Solution

- It's possible to redirect all mobile users to a different URL, based on their device



Mobile sites

- But then we'd have to maintain 2 different versions of every page
- Google also doesn't condone this, as this means that your main site is still not responsive.



Single Site for all Devices

A single site for all devices gives your website a few advantages:

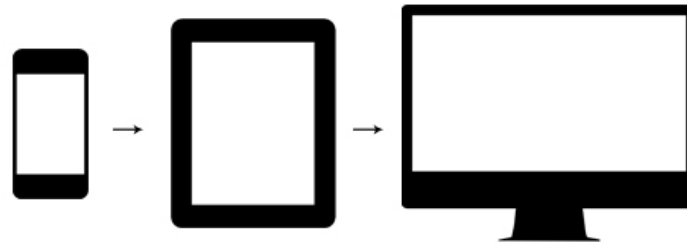
- Super Flexible
- Excellent User Experience
- Cost Effective
- Very Easy to manage
- It is Recommended By Google

One URL and the same HTML, regardless of the device, makes it easier and more efficient for Google to crawl, index, and organize content.



How to approach?

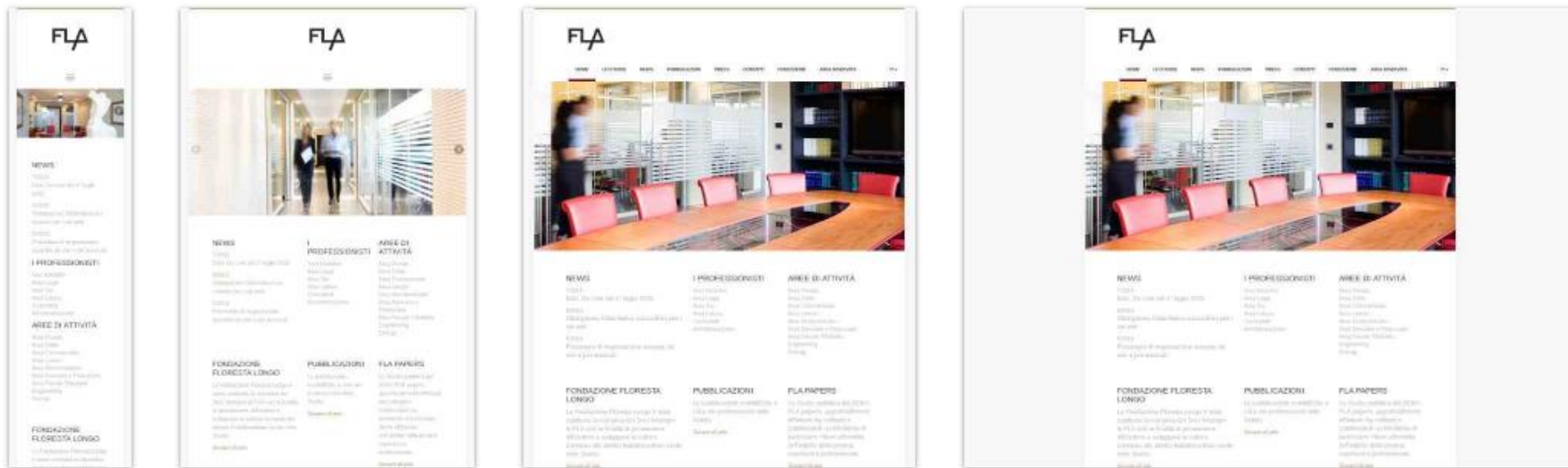
The mobile-first approach



- Designing for the smallest screen and working your way up.
- It is the best strategy to create either a responsive or adaptive design.

How to approach?

- Usually it means using a “Stripes” layout



Notice! Responsive design refers mostly to the screen width...

“Stripes” Revisited

- Render the following page in your head:

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="resp.css">
</head>

<body>
  <section class="menu stripe"></section>
  <section class="promo stripe"></section>
  <section class="features stripe"></section>
  <section class="gallery stripe"></section>
  <section class="footer stripe"></section>
</body>
</html>
```

HTML5

```
.stripe{
  height:200px;
  border:1px black solid;
}

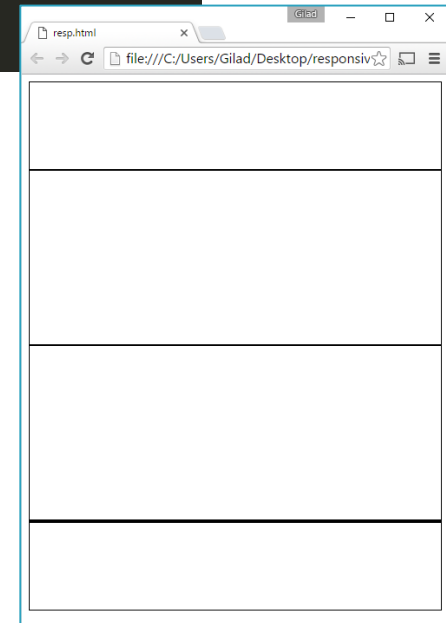
.menu.stripe , .footer.stripe{
  height:100px;
}

.gallery.stripe{
  height:auto;
}
```

Is it responsive? **SURE**

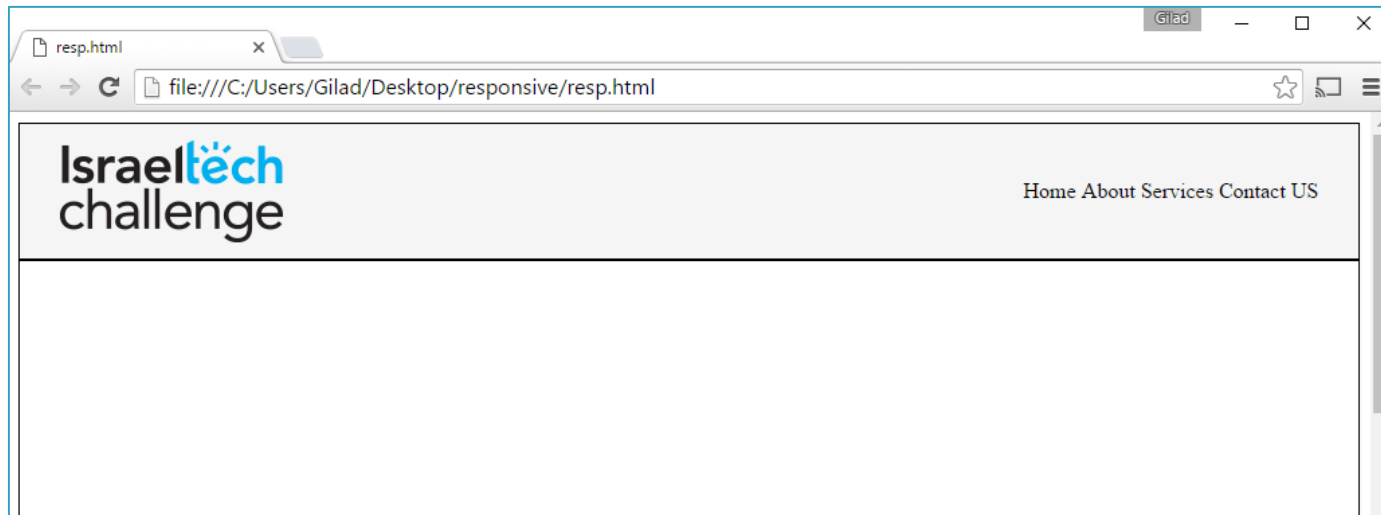


Using the default “display:block” of section/div in order to take up 100% width



The Menu Stripe

- The most common menu



The Menu Stripe



This is important if you want to “Absolute” position child elements. Read more [here](#) and [here](#).



The Menu Stripe

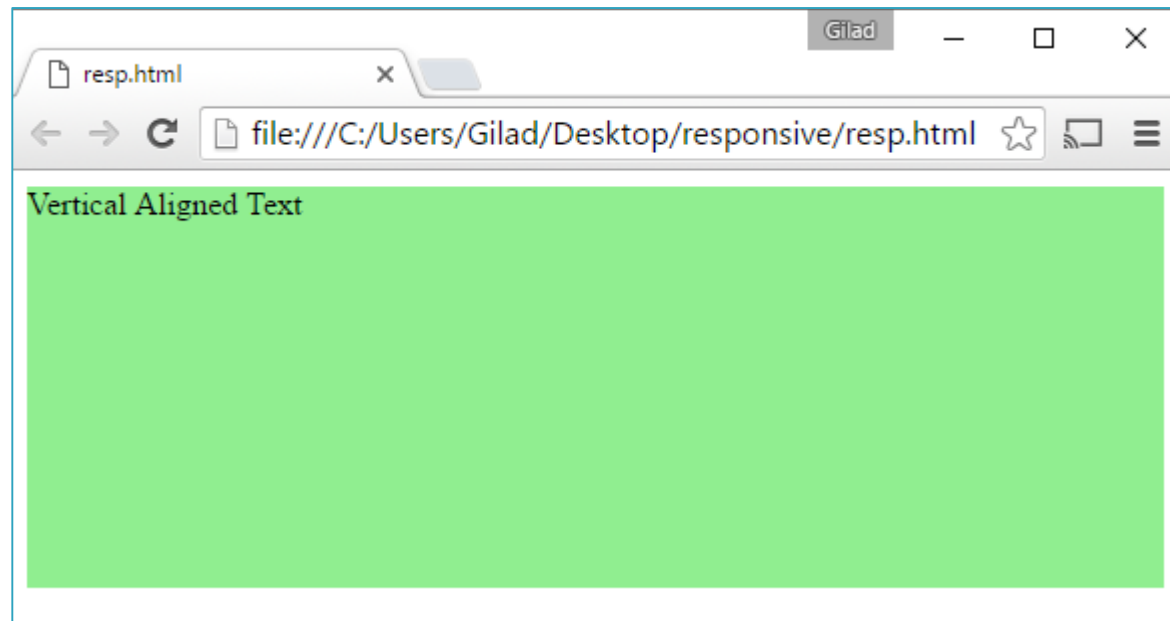


We don't want this div to take all of the menu's width



We are using percentage because we want smaller margins on mobile

Vertical Align - The Basic case

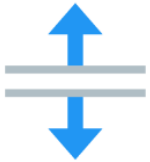


```
<section class="parent">
  <div class="valign">Vertical Aligned Text</div>
</section>
```

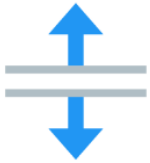
```
.parent {
  background-color: #90EE90;
  height: 200px;
}
```

resp.css:3

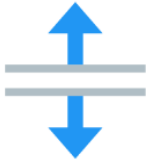
Solutions



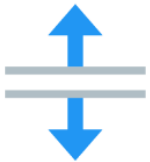
Using absolute positioning, top with percentages and negative margin



Using the vertical-align with `display:table` and `display:table-cell`



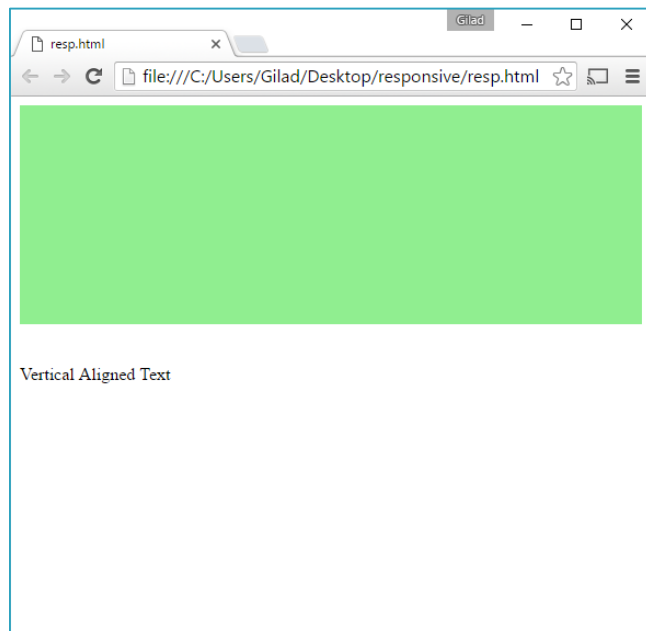
Using the CSS 3 transform magic



Using `display:flex`

First solution

- Using absolute positioning, top with percentages and negative margin



```
.parent{
  background-color:#90EE90;
  height:200px;
}

.valign{
  position:absolute;
  top:50%;
}
```

What happened?

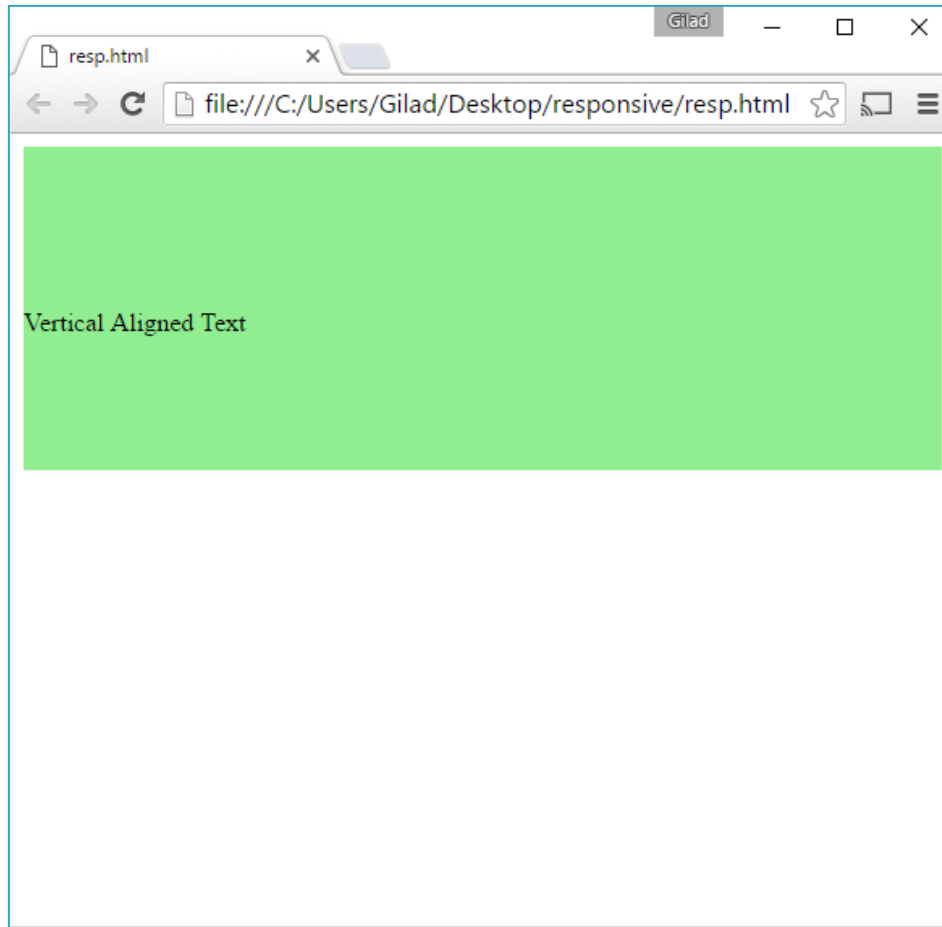
The child is absolutely positioned relative to the body

Solution?

Set the parent's position to relative

```
<section class="parent">
  <div class="valign">Vertical Aligned Text</div>
</section>
```


First solution



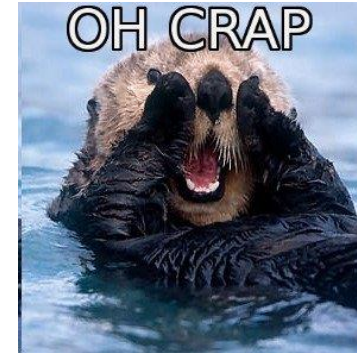
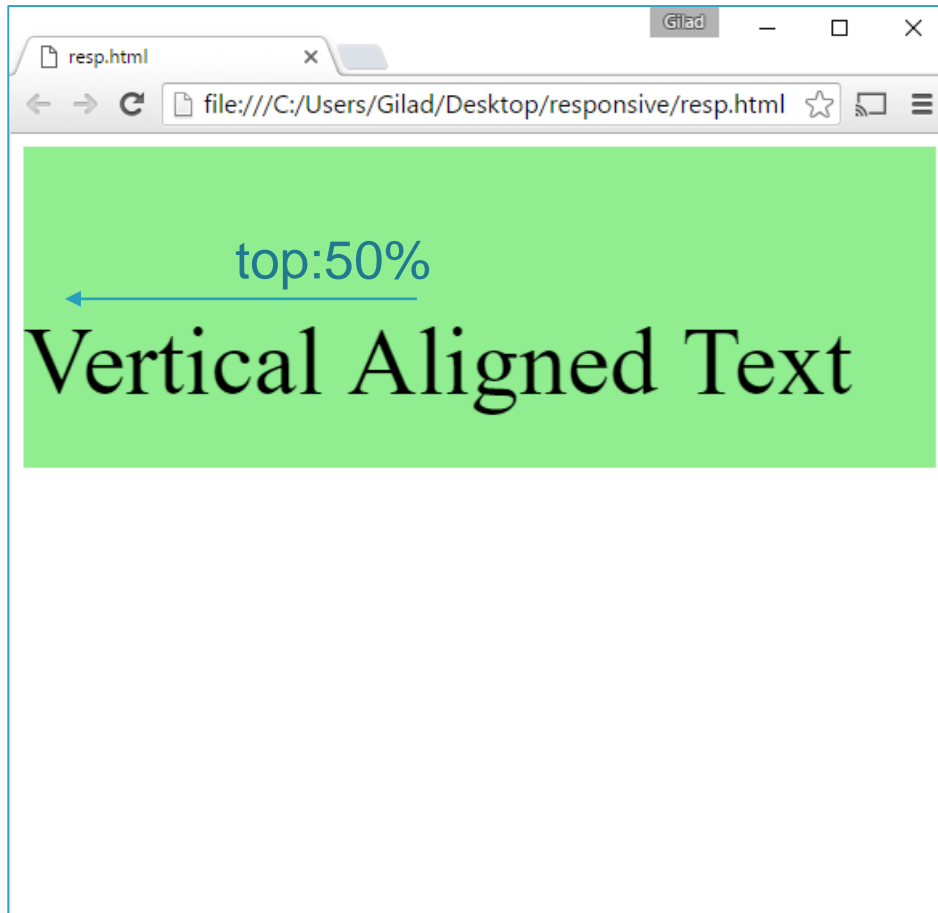
```
.parent{
  background-color:#90EE90;
  height:200px;
  position:relative;
}

.valign{
  position:absolute;
  top:50%;
}
```

Better!

What will happen if we increase the child's font-size to 60px?

First solution



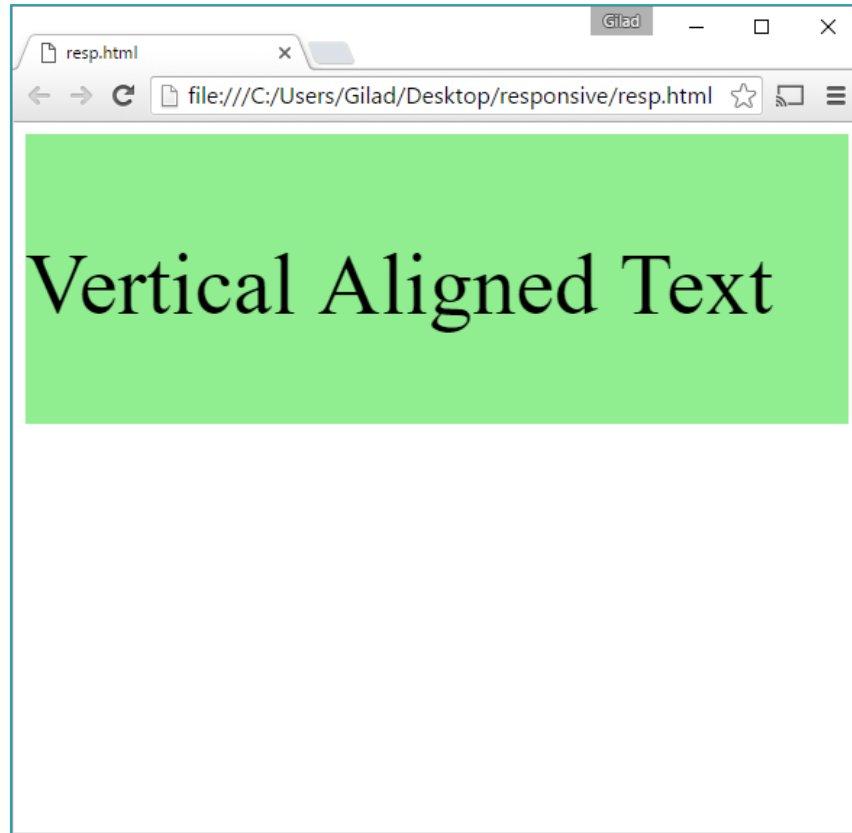
What happened?

The “top” property positions the element’s top...

The solution is :
to give the element a
negative top margin

How much?
Half of the child’s height

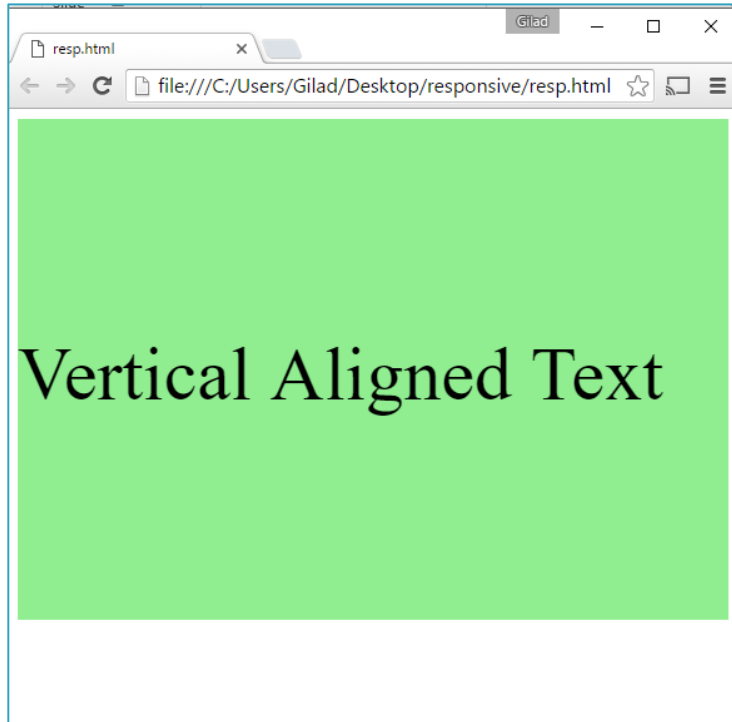
First solution



```
.parent{
  background-color:#90EE90;
  height:200px;
  position:relative;
}

.valign{
  position:absolute;
  top:50%;
  margin-top:-30px;
  font-size:60px;
}
```

First solution



What will happen if we change the parent's height to 400px?

Still good

What will happen if we change the child's font-size or height?

Or add text?

Or more content like images?

We must update the
margin
top value

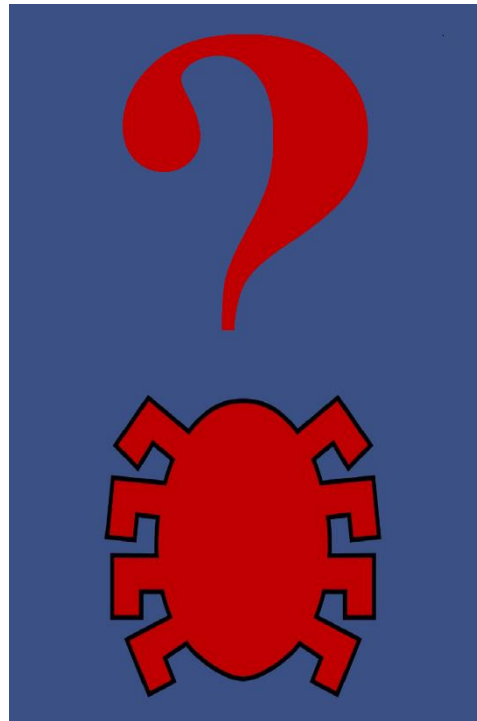
First solution

- Using absolute positioning, top with percentages and negative margin is a good and easy solution

BUT

maintenance is needed when the child's height changes.

Questions



Second solution

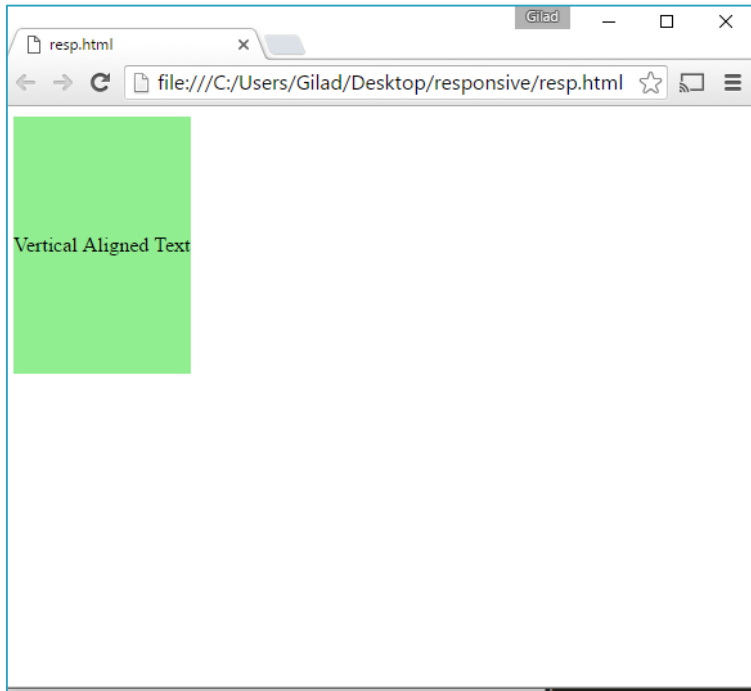
Parent => display:table + child => display:table-cell

- CSS allows elements to behave like the <table> tag without actually using it
- A <td> in a <table> implements the vertical-align attribute

So... we will make our left-side act like a table and it's content like a table cell

Second solution

- By setting the parent's display to "table"
And the child's display to "table-cell"
We are getting vertical alignment for free.



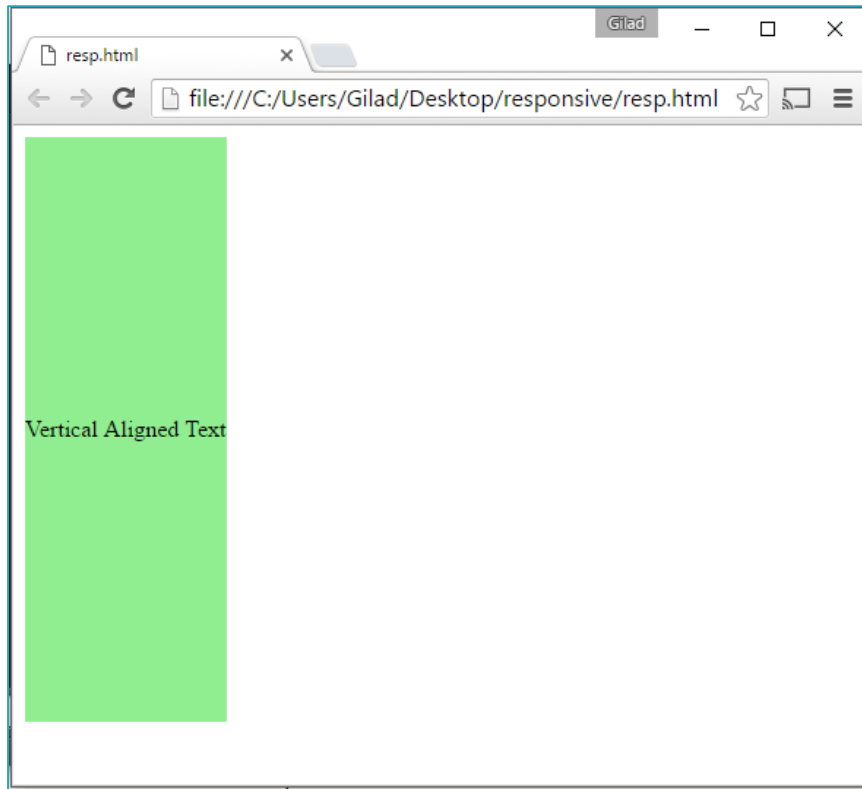
```
.parent{  
  background-color:#90EE90;  
  height:200px;  
  position:relative;  
  display:table;  
}  
  
.valign{  
  display:table-cell;  
  vertical-align:middle;  
}
```



Notice that we lost the "block" property of the parent and now it's width is as big as it's content.

Second solution

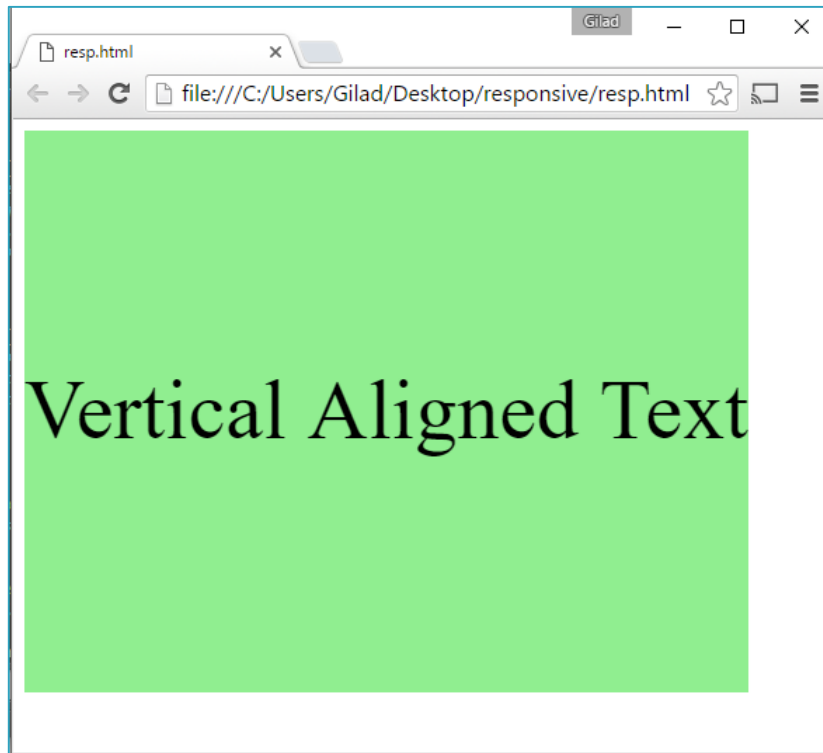
- What will happen if we change the parent's height?



All is well...

Second solution

- What will happen if we will change the child's font size?



Still good



Second solution

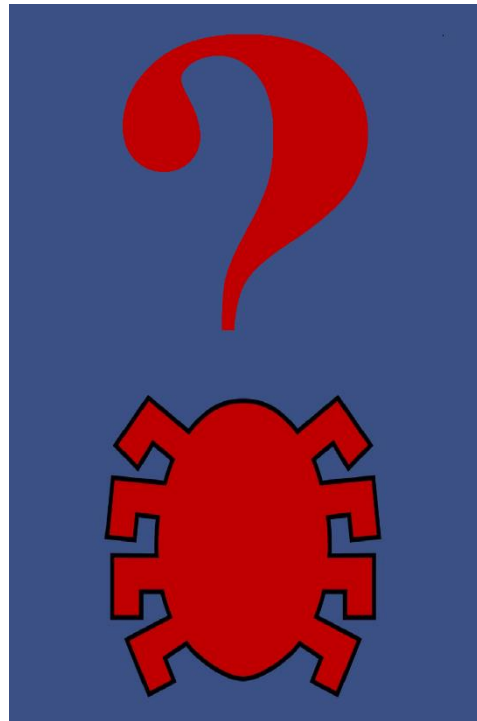
- Using the vertical-align with display:table and display:table-cell solves the problem without the need for maintenance

BUT

It is semantically wrong

And we lose the 100% width of the block element.

Questions



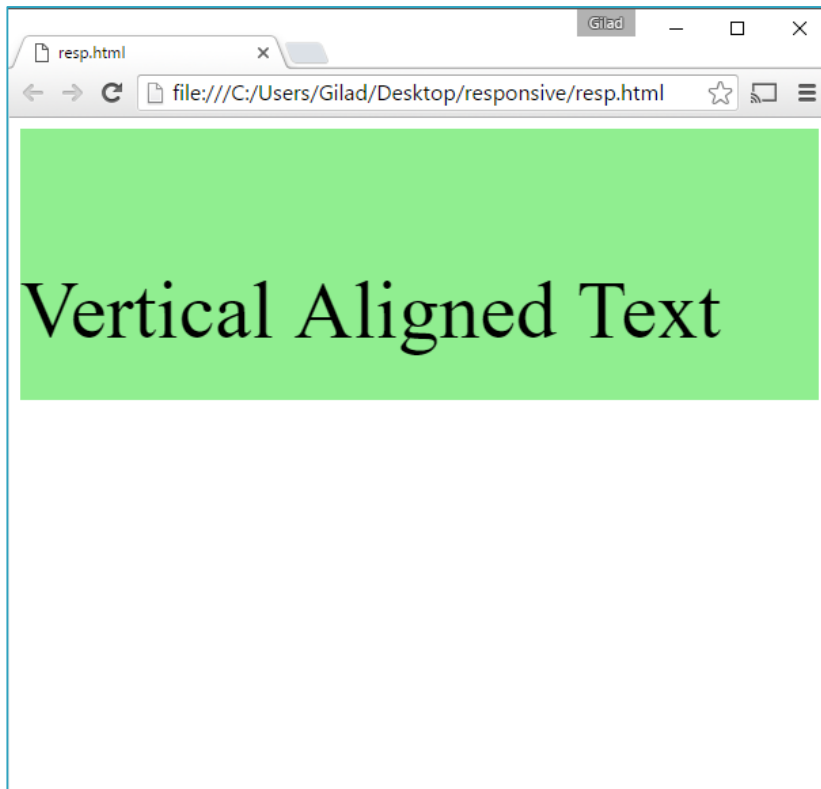
Third solution

Using the CSS 3 transform magic.

- We can solve the maintenance problem we encountered in the first solution by using the transform property.

Third solution

- Revisiting our problem: CSS top property is measured from the child's top and we don't know the child's height



```
.parent{
  background-color:#90EE90;
  height:200px;
  position:relative;
}

.valign{
  position:absolute;
  font-size:60px;
  top:50%;
}
```

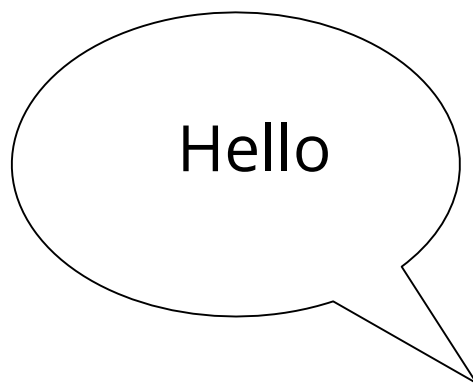
Third solution

- The “transform” property is a very rich property in CSS 3 that allows us to manipulate elements in many different ways. (we will learn about it later on).
- For now we will focus just on The translateY option.
- Translating an element (translateY) means we are moving it on the Y axis.
- Unlike setting the “top” of an element, which sets the starting point of an element (the edges), translate moves the element itself

Third solution

So by setting `translateY` to `-50%`, we will move it up by half of it's height

And this is how the third solution looks like



```
.parent{  
  background-color:#90EE90;  
  height:200px;  
  position:relative;  
}  
  
.valign{  
  position:absolute;  
  font-size:60px;  
  top:50%;  
  transform:translateY(-50%);  
}
```



Second solution

Using the CSS3 transform magic solves the problem most elegantly






BUT

Old browsers do not support CSS3
(Internet Explorer 9 and below, for example) See details [here](#)

Fourth solution

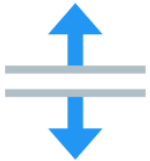
Display:flex

Flexible boxes, or flexbox, is a new layout mode in CSS3.

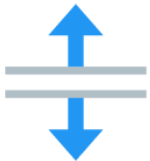
Property					
Basic support (single-line flexbox)	29.0 21.0 -webkit-	11.0	22.0 18.0 -moz-	6.1 -webkit-	12.1 -webkit-
Multi-line flexbox	29.0 21.0 -webkit-	11.0	28.0	6.1 -webkit-	17.0 15.0 -webkit- 12.1

```
.parent{
  background-color: #90EE90;
  height:200px;
  display: flex;
  flex-direction: row;
  align-items: center;
}
```

Solutions



Using absolute positioning, top with percentages and negative margin.



Using the vertical-align with display:table and display:table-cell.

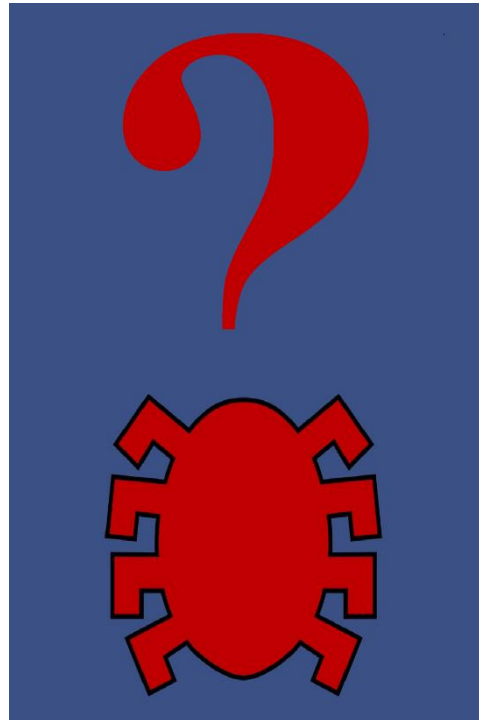


Using the CSS 3 transform magic.



Using display:flex.

Questions



<Break /> ?

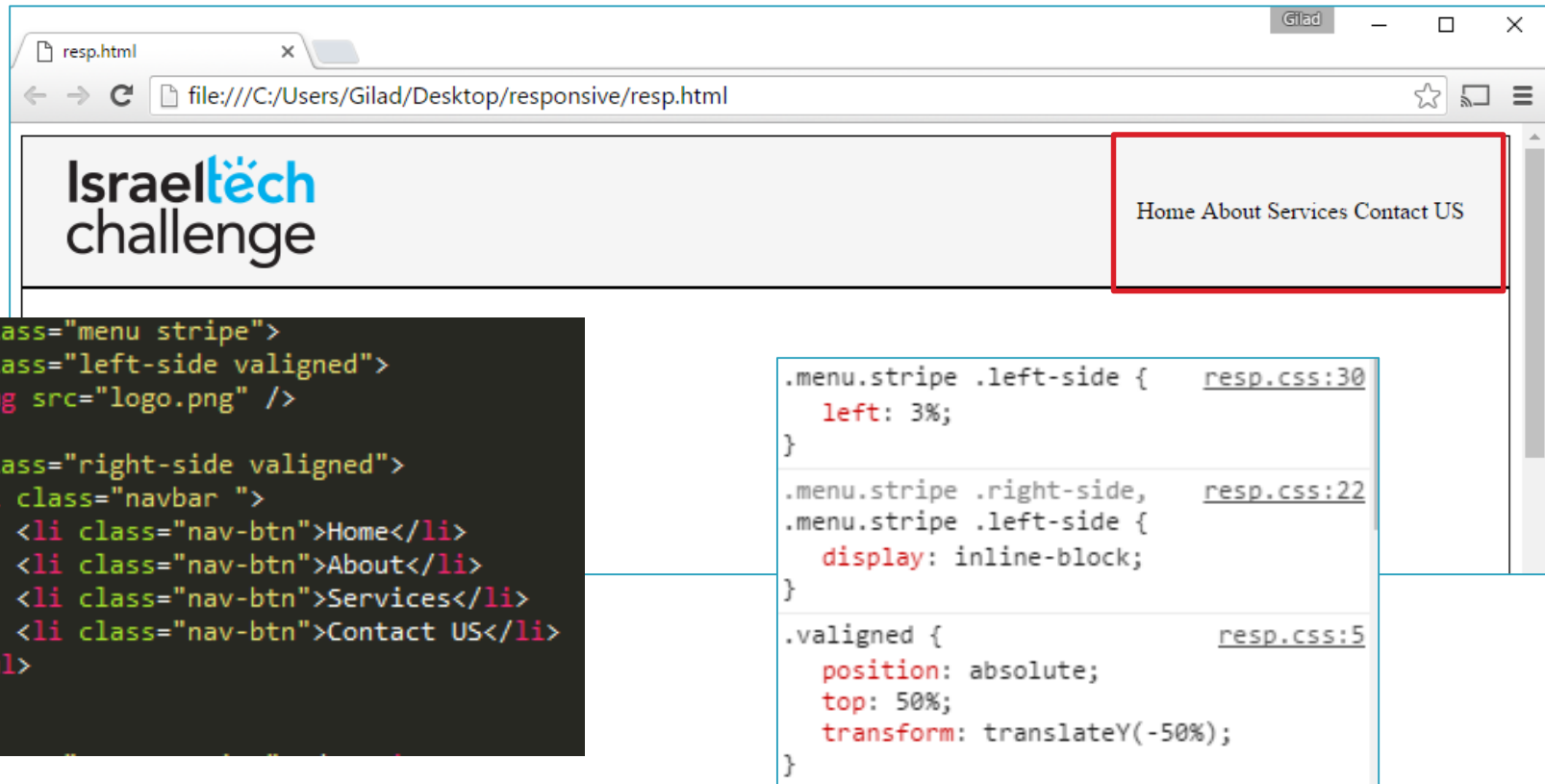
AAAND back to the menu



Using a general class name
.valigned will make it easier for
us to use it for other elements
in the document



AAAND back to the menu



The screenshot shows a web browser window with the title 'resp.html' and the address bar displaying 'file:///C:/Users/Gilad/Desktop/responsive/resp.html'. The website content includes a logo 'Israeltech challenge' on the left and a navigation menu 'Home About Services Contact US' on the right. A red box highlights the menu stripe. Below the browser window, there are two code blocks: one showing the HTML structure and another showing the CSS rules.

HTML Structure:

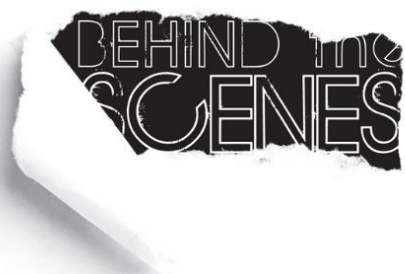
```
<section class="menu stripe">
  <div class="left-side valignned">
    
  </div>
  <div class="right-side valignned">
    <ul class="navbar ">
      <li class="nav-btn">Home</li>
      <li class="nav-btn">About</li>
      <li class="nav-btn">Services</li>
      <li class="nav-btn">Contact US</li>
    </ul>
  </div>
</section>
```

CSS Rules:

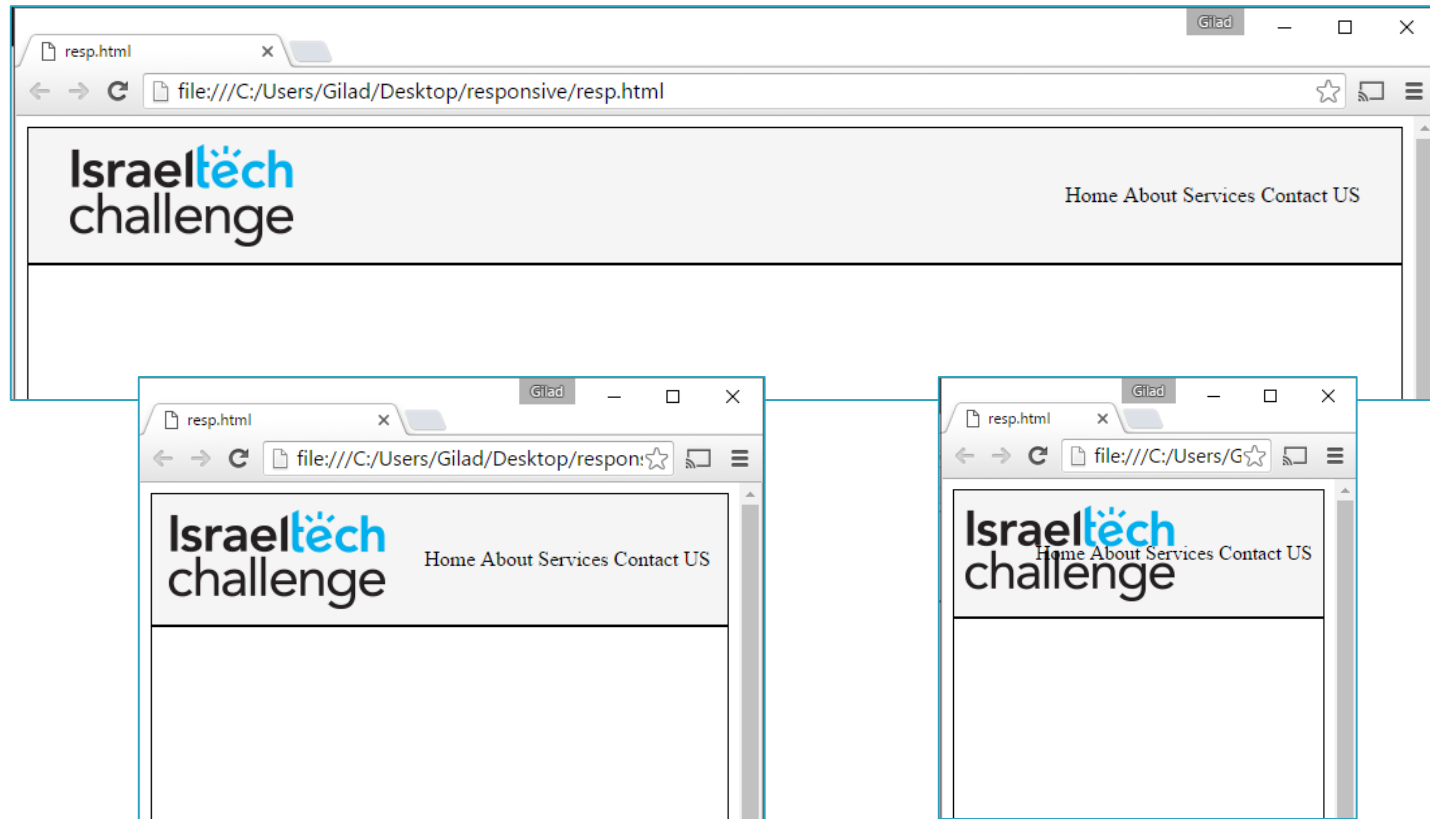
```
.menu.stripe .left-side { resp.css:30
  left: 3%;
}

.menu.stripe .right-side, resp.css:22
.menu.stripe .left-side {
  display: inline-block;
}

.valigned { resp.css:5
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
}
```



Is it responsive?



The solution?

Minify the menu

You probably know that menu style...

We call it the hamburger menu

For those of you who didn't get it...



The solution?

Hamburger recipe:

```
<ul class="hamburger clickable">
  <li class="bread slice"></li>
  <li class="meat slice"></li>
  <li class="bread slice"></li>
</ul>
```

```
.menu.stripe .right-side .hamburger .slice{
  width:30px;
  height:5px;
  background-color:black;
  margin:0px;
}

.menu.stripe .right-side .hamburger .meat.slice{
  margin: 5px 0px;
}
```

Israeltëch
challenge



Delicious

Is it good?

Well, it works...

But it doesn't respect people with large screens



Flexible Solution

We want more flexibility!



We want to give each device the best solution!

Another solution?

Introducing: Media queries

Media queries are logical “if” statements that test for properties of a device

```
if (this expression is true) {
  code block
}
```

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

← A Boolean statement

← The CSS that is rendered if the statement is true



Notice The syntax! we added a new scope by adding another pair of curly brackets

Media queries

- Min-width

```
@media screen and (min-width: 330px) {  
  
}
```

"If [device width] is greater than or equal to [specified #], then do {...}"

So if the actual "device width" is 320px, this condition will return false.

Media queries

- max-width

```
@media only screen and (max-width: 330px) {  
  
}
```

"If [device width] is less than or equal to [specified #], then do {...}"

So based on this second example, if the "device width" is 320px the condition is true

Back to our menu

Normal state

Hide the hamburger

Show the menu

Small screen state

(or menu is too large to fit)

Show the hamburger

Hide the menu

```
.menu.stripe .right-side .hamburger{
  display:none;
}

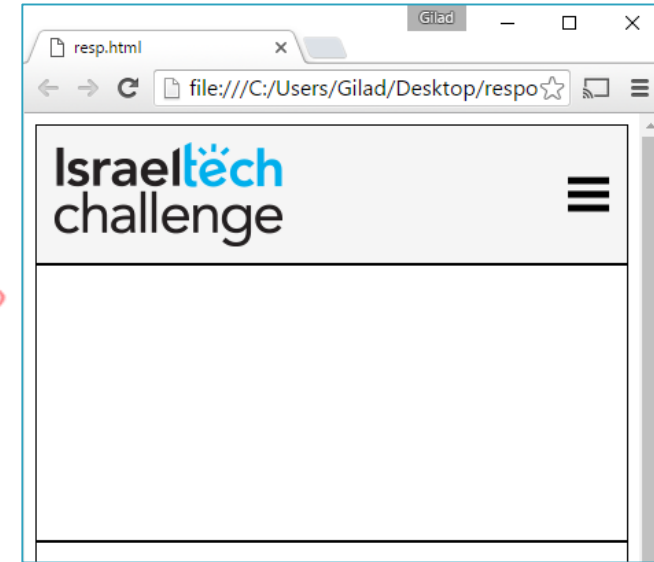
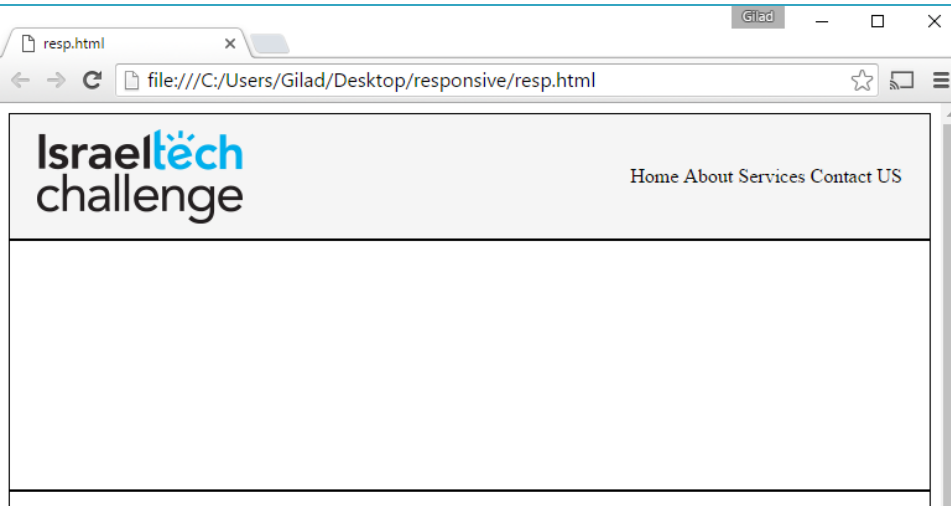
.menu.stripe .right-side .navbar {
  display:inline-block;
}

@media screen and (max-width: 500px) {
  .menu.stripe .right-side .navbar {
    display:none;
  }
  .menu.stripe .right-side .hamburger{
    display:inline-block;
  }
}
```



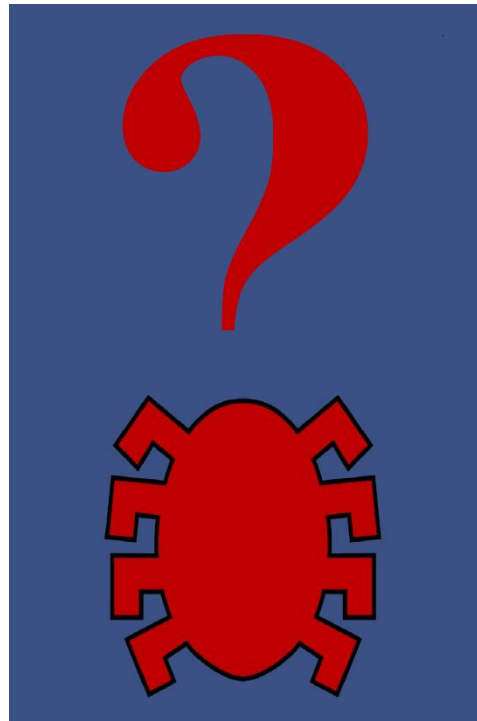
Notice the order: the “normal” state must come before the “mobile” state so the latter will override the properties

Back to our menu



We will learn how to show the original menu when clicking the hamburger when we will learn JS...

Questions



There's more?

Media queries have logical operators and can be much more complex:

- You can have multiple “break points”, i.e for desktops, tablets and mobile
- You can check the device type (tv, phone, pc...), it's resolution, it's orientation and many more

Getting Wacky

Media queries have logical operators and can be much more complex:

```
@media
only screen and (min-width: 100px),
not all and (min-width: 100px),
not print and (min-height: 100px),
(color),
(min-height: 100px) and (max-height: 1000px),
handheld and (orientation: landscape)
{
    html {
        background: red;
    }
}
```

This is not the way to go...

When to use?

- When you have a small number of cases and break points that you need to tackle
- Usually referring to the screen width



Let's see another example

Features Stripe

“Features” or “featured” or “team” is a section in the document that presents a row of several important items

2004

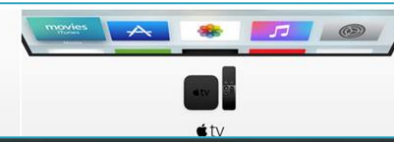
Squarespace was launched from Anthony's dorm room at the University of Maryland.

554

Squarespace has grown to a team of 554 talented individuals.

Millions

Since 2004, millions of websites have been created on the Squarespace platform.



6

weeks in total
June 11 to August 20

9

hours
of class a day

7

days of traveling
and trips

5,500

dollars
including accommodations, medical coverage
and educational and social activities

Features Stripe

In order for it to be responsive, we would like to give each item a width based on a percentage.

Let's say we have 4 items, each one will take up 20% of the parent's width.

so we also need to take into account the space between the items

Features Stripe

```
<section class="features stripe">
  <div class="item">
    <div class="title">First Item</div>
    <div class="text">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a
    sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet
    magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris.
    Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
    </div>
  </div>
  <div class="item">
    <div class="title">First Item</div>
    <div class="text">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a
    sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet
    magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris.
    Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
    </div>
  </div>
  <div class="item">
    <div class="title">First Item</div>
    <div class="text">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a
    sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet
    magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris.
    Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
    </div>
  </div>
  <div class="item">
    <div class="title">First Item</div>
    <div class="text">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a
    sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet
    magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris.
    Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
    </div>
  </div>
</section>
```

Features Stripe

First Item	Second Item	Third Item	Fourth Item
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.</p>	<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.</p>

The default alignment for inline elements is bottom.

Items should align to their top if they are not of the same height

```
.features.stripe .item{
    vertical-align:top;
    text-align:left;
    color:white;
    display:inline-block;
    width:20%;
    background-color:#333;
}

.features.stripe .item .title{
    font-size:20px;
}
```


Is it responsive?

First Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Second Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Third Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Fourth Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
---	--	---	--

First Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Second Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Third Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Fourth Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
---	--	---	--



First Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Second Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Third Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.	Fourth Item Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.
---	--	---	--

This is not usable...

The Solution?

Media queries

```
@media screen and (max-width: 500px) {  
  .features.stripe .item {  
    display: block;  
    width: 80%;  
    margin: 20px auto;  
  }  
  .menu.stripe .right-side .navbar {  
    display: none;  
  }  
  .menu.stripe .right-side .hamburger {  
    display: inline-block;  
  }  
}
```

In English:

- When the screen is smaller than 500px, show one item per line.
- Each item should take up 80% of the screen's width and be horizontally centered.

Responsive features

First Item

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.

First Item

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.

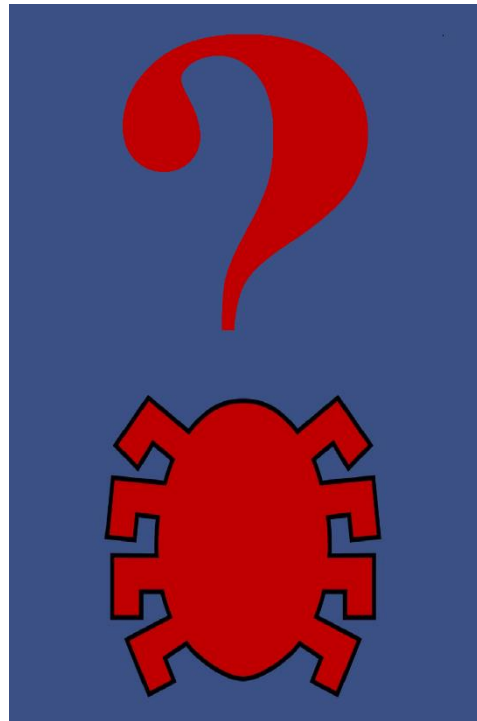
First Item

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam lobortis lorem a sagittis laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras et nunc sit amet magna feugiat posuere. Nulla facilisi. In eget est non ante scelerisque elementum eget ac mauris. Nam arcu justo, dapibus at erat sed, hendrerit molestie augue.

First Item



Questions

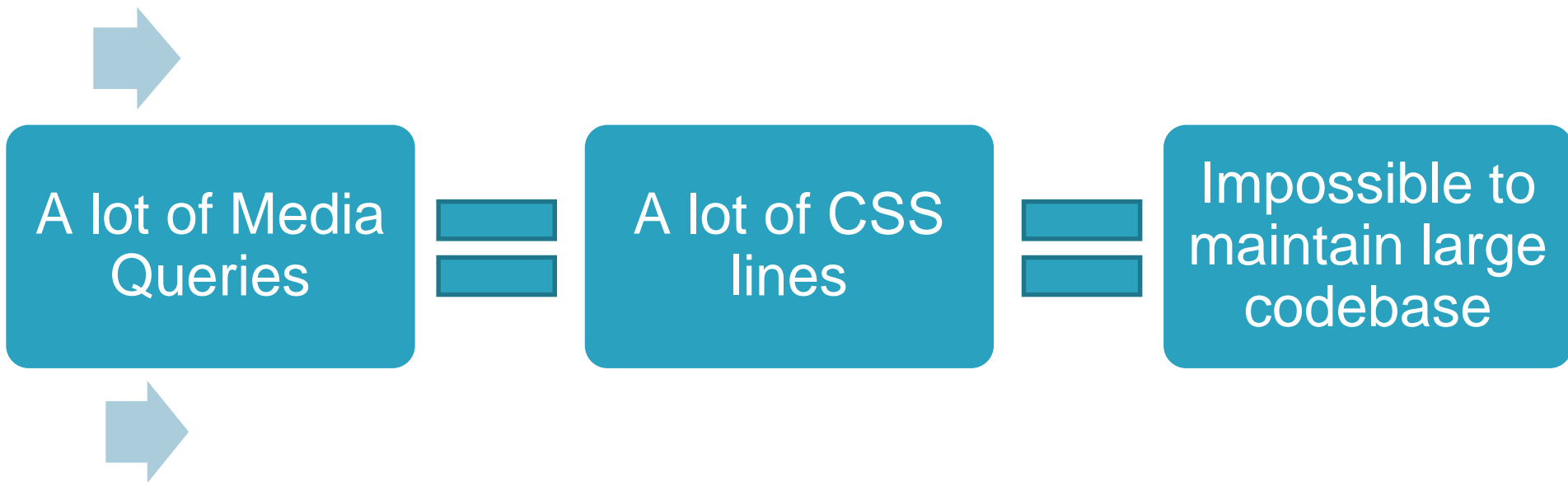


Why media queries are not so great?

The problem with Media Queries is the responsive work that they demand.

Think about it – We will have to make a different Media Query for each resolution! (a lot of work).

Why media queries are not so great?



Why media queries are not so great?

- Until recent years, the poor web developer had to make a Media Query for each of the screen sizes.
- Fortunately, we can use a set of utilities that other great programmers made for us!
- These set of utilities are called Frameworks...



Summary

- ▶ You needed to understand:
 - Pros and cons of different ways to align content in a responsive way
- ▶ You need to remember:
 - The importance of responsive design
- ▶ You need to be able to do:
 - Use vertical-align
 - Use media queries

Questions?



Cheat Sheet

Vertical align

Margin top

```
.parent {
  position: relative;
}
.vertical-align {
  position: absolute;
  top: 50%;
  font-size: 60px;
  /* half of its size */
  margin-top: -30px;
}
```

Flex box

```
.parent {
  display: flex;
  flex-direction: row;
  align-items: center;
}
```

Table Display

```
.parent {
  position: relative;
  display: table;
}
.vertical-align {
  display: table-cell;
  vertical-align: middle;
}
```

Translate Y

```
.parent {
  position: relative;
}
.vertical-align {
  position: absolute;
  top: 50%;
  transform: translateY(-
50%);
}
```

Hamburger Menu

HTML

```
<ul>
  <li class="hamburger-slice"></li>
  <li class="hamburger-slice meat"></li>
  <li class="hamburger-slice"></li>
</ul>
```

CSS

```
.hamburger-slice {
  width: 30px;
  height: 5px;
  background-color: black;
  margin: 0px;
}

.hamburger-slice.meat {
  margin: 5px 0;
}
```

Media Queries

```
@media screen and (min-width: 330px) {
  .navbar {
    display: inline-block;
  }
}
```

```
@media screen and (max-width: 330px) {
  .navbar {
    display: none;
  }
}
```