Fluid Layouts

- For a long time, we've built our websites to be oak trees: rigid and fixed-width. They look fantastic, until they meet with the inevitable unpredictably of the Web.
  - New Devices with different display characteristics
- To respond to this challenge we now embrace Fluid Layouts for web sites

Four Layout Options

- Understanding when a flexible layout might be the best choice requires that we review the other available options.
  - Fixed-width,
  - Liquid (or Fluid),
  - Elastic, and
  - Hybrid.

Fixed-width

- In fixed-width layouts the site width is constrained by a specific pixel measurement, 960px being the most commonly chosen width today.
- The number 960 is grid-based layout friendly.
  - It is easily divisible by 3, 4, 5, 6, 8, 10, 12 and 15 and therefore offers a variety of grid options
  - works well with the Interactive Advertising Bureau's (IAB) standard ad sizes.

Fixed-width

- Fixed-width layouts are the most common implementation on the Web.
- Knowing exactly the width at which your site will display lets you create graphically intense designs that will look fairly consistent across different screens.
- Simplifies the web site design

Fixed-width

- The biggest problem with fixed-width layouts is that you must operate under a lot of assumptions.
- When you determine how wide your site will be, you're making a guess as to what dimensions will best serve the largest percentage of visitors.
  - 13", 15" and 17" Displays
  - Mobile devices
  - Not all users have the browser at full size
- This helped drive the separate sites for web and mobile approach

Fixed-width

- The “consistency” that a fixed-width design provides is also a bit misleading. If your site is 960px wide and a visitor has a smaller screen (let’s say 800px wide), then she’ll see only a portion of your site and an ugly horizontal scrollbar
- Larger screens are not immune to issues either. If someone with a large monitor views your 960px-wide site, she’ll be met with a large amount of unplanned white space.

Fixed-width

- The rigidity of fixed-width layouts is even more of an issue in today's widely diverse ecosystem of devices.
- Many of the newest and most capable phones and tablets display sites zoomed out so they fit on the screen.
  - These devices provide a pinch-to-zoom feature to drill down from there.
- While this experience is better than not being able to zoom in or access the site at all, it is still cumbersome and far from enjoyable.

Fluid Layouts

- In fluid layouts, dimensions are determined by percentages, not pixel measurements.
- As a result, they're much more malleable.
- For example, you may have a primary column that's sized at 60% of the width of the container, a right sidebar column sized at 30%, and a gutter of 10% in between.

Fluid Layouts

- Using a layout defined in that manner means that it doesn't matter if the user is using a desktop computer with a browser width of 1024px, or a tablet sized at 768px wide: the widths of the elements in the page will adjust accordingly.
- A design built on a fluid layout avoids many of the problems that a fixed-width layout presents.
  - Horizontal scrollbars can be prevented, for the most part.

Fluid Layouts

- Since the site can adapt its width based on the width of the browser window, the design can adjust to better accommodate the available space, eliminating the unwanted white space that can occur with a fixed-width layout.
- However, a fluid layout by itself is not enough to ensure that a design looks good on everything from a smartphone to a TV.
  - Line lengths can become too wide on large displays, and too narrow on small displays.
- But it is a start!!!

Elastic Layouts

- Elastic layouts are very similar to fluid layouts, except that their constraints are determined by the type size— typically using the em measurement.
- One em is the equivalent of the currently defined font-size.
  - Say, for example, the body text is in a font-size of 16px. In that case, 1em equals 16px and 2em equals to 32px
- Elastic layouts provide strong typographic control. A large body of research recommends a line length between 45 and 70 characters for ideal readability.

Elastic Layouts

- With an elastic layout, you can define the width of the container to be, say, 55em.
- This would ensure that the width of the container always maintains an appropriate line length.
- Another benefit of elastic layouts is that as a user increases or decreases the font size, elements defined with elastic widths will scale in proportion to that font size.
  - Remember that Accessibility rule require that a user to change the size of a font

Elastic Layout

- Unfortunately, elastic layouts can bring back the dreaded horizontal scrollbar.
- If you have a font-size of 16px and you define the width of the container to be 55em, then any screen below 880px (16 × 55) will display a horizontal scrollbar.
- The issue can be even more unpredictable than with fixed widths. If the visitor increases the font-size to say, 18px, the width of your container just got changed to 990px (18 × 55).

Hybrid Layouts

- Hybrid layouts combines features from
  - Fixed-width Layouts
  - Fluid Layouts, and
  - Elastic Layouts

Which Approach is the most Responsive

- So which is the right method to be responsive to various devices and environments?
  - It depends
  - But Fluid, Elastic and Hybrid tend to win
- You can switch between fixed-width layouts using media queries
  - Here you have a finite number of layouts and pick the layout intended for a particular device
  - If you do not have a layout for a particular device the user would get a sub-par experience

Sizing Fonts

- You can size fonts on the Web using any of a number of different units, but the primary options are:
  - pixels,
  - percentages, and
  - ems.

Pixels

- For quite some time, pixels were the preferred method for sizing fonts.
- The reason is simple: you have precise control over how a browser sizes the text.
- If you set the font-size to 18px, each browser will display that size at precisely 18px.

Pixels

- This control comes at a cost. For starters, with pixel-sized pixel-sized fonts, there's no cascade— that is, the font size of the parent element has no effect on the font size of the child element.
- This means that you need to set a specific pixel size for each element where you want the text to display at a different size.
  - Maintenance Nightmare

Pixels

- More importantly, pixel-sized fonts are a potential accessibility concern.
- All major browsers allow the user to zoom in or out of a page. There are two ways a browser handles this. The first is to simply apply the zoom to everything on the page. So if a user zooms in, every element on a page will increase in size. This method allows users to zoom regardless of how the font is sized
- The other method is to resize the text itself, but not the other elements on the page. This was a common behavior for a long time, and it is still implemented by some browsers.

Pixels

- Pixel-sized fonts, unfortunately, do not scale in Internet Explorer. This means that for anyone using a version of the browser prior to IE9, where font resizing was the default (or if they have turned font resizing on in the latest version), the size of the fonts on your page will not be adjustable.

Pixels

- The ability to resize the text puts the user in control and also improves the accessibility of your site.
  - Some visitors may have difficulty reading text below a certain size.
  - Allowing them to increase the font size means that they can still enjoy your content.

Pixels

- Sizing fonts in pixels is also not a very future-friendly approach.
- Different devices have different screen sizes and pixel densities. As a result, a pixel-sized font that looks good on one device may be too small, or too large, on another screen

Ems

- A much more flexible, and increasingly popular, way of sizing fonts is to use the em unit instead.
- Ems are resizable across browsers.
  - They also cascade— which can be both a good thing and a bad thing
- It is good in the sense that it eases maintenance. Sizing the fonts of your elements relatively means you need only adjust the initial baseline and the rest of the content will adjust automatically— proportions intact.

Em - Example

```
body {
    font-size: 16px; /* base font size */
}
h1 {
    font-size: 1.5em; /* 24px / 16px */
}
span {
    font-size: 1em; /* 16px / 16px */
}
```

Em - Example

```
< h1 > Question One
  < span >
      Please choose an answer from below.
  </ span >
</ h1 >
```

Em - Example

- In the example the base font-size is set to `16px`.
- The `h1` element has a font size of `1.5em`, the equivalent of `24px`.
- We want our `span` to render at `16px`, so we set it to `1em`. The problem is that the context has changed.
  - The base is no longer the `16px` body font, it's the `24px` font size attached to the `h1` element.
  - Instead of our expected `16px` font, our `span` will actually display at `24px`.

Em

- In Internet Explorer, if the base font-size is defined using ems, then Internet Explorer greatly exaggerates how small or large the font should be when resized.

Percentages

- Fonts sized in percentages, like ems, are also resizable and they cascade.
- As with ems, if the base font-size is `16px`, then `100%` equals `16px` and `200%` equals `32px`.

Percentages

- While theoretically there isn't a major difference between ems and percentages, it's slowly becoming more and more common that ems are the preferred unit of measurement for fonts on the Web.
- There really isn't a technical reason for this, it just makes sense to use ems when sizing text since the em unit is directly related to the size of the text.
  - This is also the norm with text formatters such as LaTeX

Which Font sizing is most Responsive?

- There are some trade-offs here to consider when deciding which approach to take.
- Using ems not only lets your type scale, but it can also make maintenance easier.
- For portability reasons setting the base font size using percentages and thereafter using ems provides for the most most flexibility

Grid Layouts

- Setting designs to a grid is an incredibly popular practice, one that predates the Web by many decades.
- Grids help to achieve balance, spacing, and organization on a site.
- A well-implemented grid system makes your site feel less cluttered and improves its readability and scanability.

Benefits of Grid Layouts

- Grids add order, creativity, and harmony to the presentation of information.
- Grids allow an audience to predict where to find information, which aids in the communication of that information.
- Grids make it easier to add new content in a manner consistent with the overall vision of the original presentation.
- Grids facilitate collaboration on the design of a single solution without compromising the overall vision of the

Content-out

- The first thing to do when setting up a grid is determine the canvas.
- In graphic design, the canvas is your paper. Its dimensions determine the grid.
- You subdivide the canvas size into the number of columns you want (3, 5, 9, or 12) and work from there.

Content-Out

- As we have already discussed, on the Web, you do not have these kinds of dimensions to work with.
- Instead, you have to work content-out: let the content define the grid.
- Content takes many forms:
  - ads
  - videos
  - images
  - text
- Each of these different types of content can determine your grid.