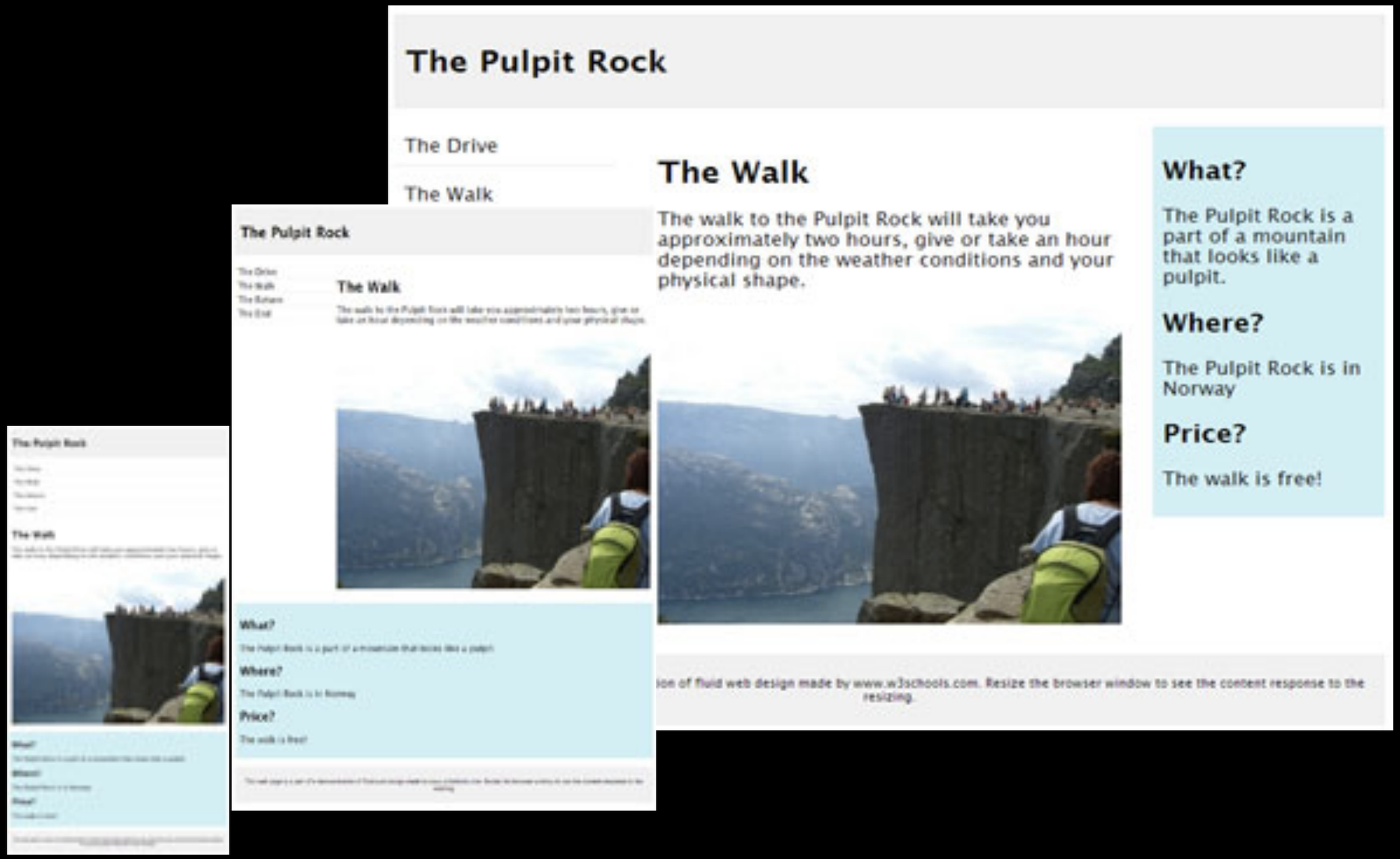
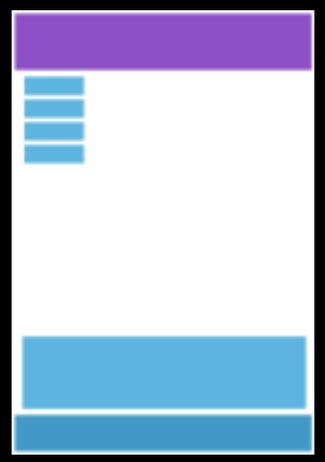
Responsive Web Design



Responsive Design slides in collaboration w/ Lee Tusman

Desktop



Tablet

Phone

Content
adapts to fit any device

Metadata: `viewport`

The user's visible area of a web page

HTML5 introduced a method to let web designers take control over the viewport, through the **<meta>** tag.

```
<!
—  - Tells the browser to match the device's width for the viewport
     - Sets an initial zoom value -->

<meta name="viewport" content="width=device-width, initial-scale=1”>
```

<meta name="viewport" content="width=device-width, initial-scale=1.0”>

without

with

Let's breakdown the `content` value of this responsive <meta> tag:

Values are comma separated, letting you specify a list of values for `content`

The `width` value is set to `device-width`. This will cause the browser to render the page at the same width of the device's screen size.

`initial-scale` set to `1` indicates the "zoom" value if your web page when it is first loaded. `1` means "no zoom."

There are other values you can specify for the `content` list -

Common device widths

There are a myriad of device screen sizes available that a user might view your website on. It is difficult (but not impossible) to develop a responsive website that looks good at any and all viewport widths and heights. For this reason, there are specific screen widths that are generally good practice to develop for and ensure your website is usable at these sizes. These widths roughly fall at the following sizes:

- 320px (x-small mobile)
- 375px (small mobile)
- 768px (tablet)
- 1024px (laptop)
- 1440px (desktop)

Although it is good practice to create and test your responsive design with these sizes in mind, it is also important that you test beyond the suggested sizes as well.

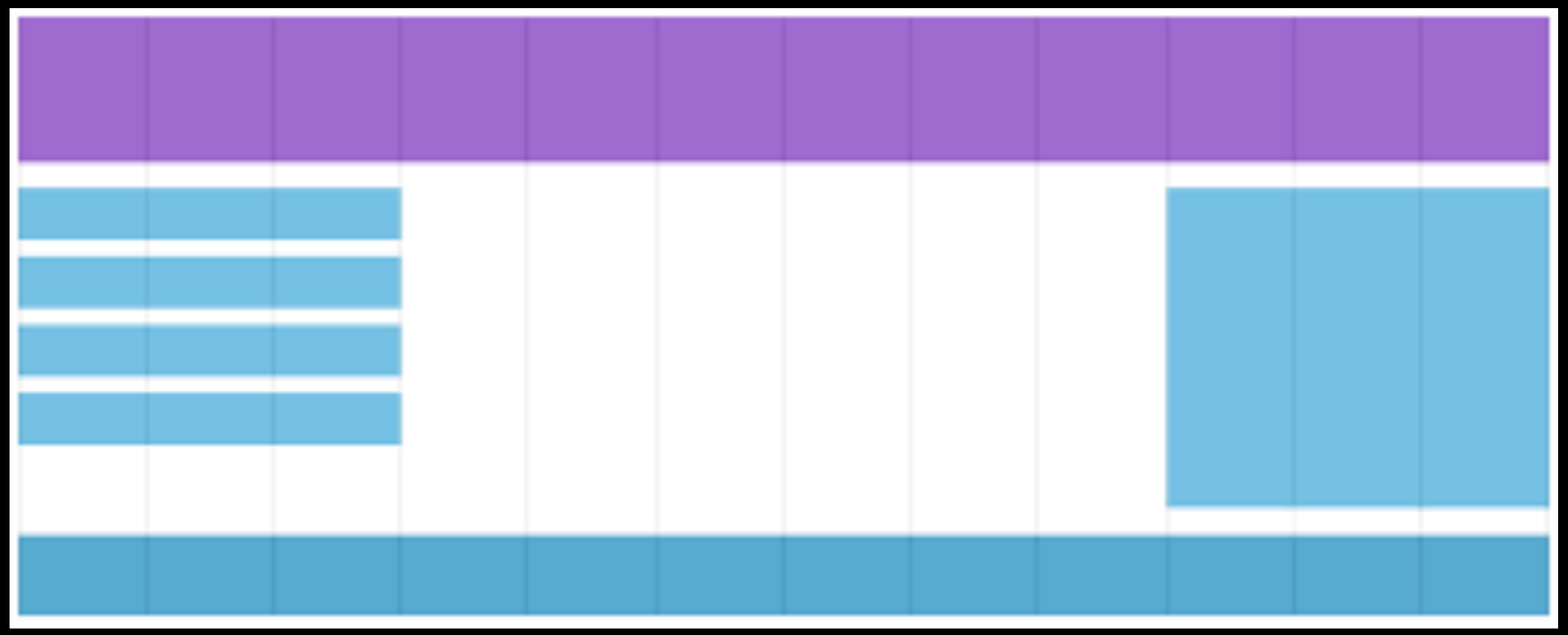
Mobile First

More and more users are visiting websites from mobile devices today. Because of this trend, there is a push to create implements what's known as a "mobile first" web design when creating sites. "Mobile first" is the practice of designing and developing your websites with a mobile layout first, before ever imagining it as a desktop site. The main reasoning behind this is that it is much easier to design and develop a website with a mobile mentality right from the start than it is to take an exisiting, non-mobile website and try to bring it into the 21st century.

Guidelines

Users are used to scrolling websites vertically, not horizontally!
Do NOT use large fixed—width elements (such as large images)
Do NOT let the content rely on a particular viewport width
Use CSS media queries to apply different styling for small and large screens

1 option: Use a grid



A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

example: https://www.w3schools.com/Css/css_rwd_grid.asp

Media Queries

Uses the @media rule to include a block of CSS properties only if a certain condition is true

Using `@media` queries

`@media` queries can be used to set conditions that must be met for certain styling to be applied to your page. `@media` queries should be used to specify the viewport conditions (generally the `min` or `max` screen widths) that you are styling for.

Example

```
@media only screen and (max-width)  {
    body {
        background-color: light blue;
    }
}
```

If max-width is 500 pixels,
make background color blue

Breakpoint

```
/* For mobile phones: */
[class*="col-"] {
    width: 100%;
}
@media only screen and (min-width: 768px) {
    /* For desktop: */
    .col-1 {width: 8.33%;}
    .col-2 {width: 16.66%;}
    .col-3 {width: 25%;}
    .col-4 {width: 33.33%;}
    .col-5 {width: 41.66%;}
    .col-6 {width: 50%;}
    .col-7 {width: 58.33%;}
    .col-8 {width: 66.66%;}
    .col-9 {width: 75%;}
    .col-10 {width: 83.33%;}
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;
}
```

add a **breakpoint** where certain parts of the design will behave differently on each side of the breakpoint

many examples: https://www.w3schools.com/Css/css_rwd_mediaqueries.asp

If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size

<img src="ashbery.jpg" style="max-width:100%;height:auto;">

Mobile-first! (Images)

```
/* For width smaller than 400px: */
body {
    background-image: url(‘void_newspaper.jpg’);
}

/* For width 400px and larger: */
@media only screen and (min-width: 400px) {
    body {
        background-image: url('void.jpg');
    }
}
```

Responsive Text

The text size can be set with a "vw" unit, which means the "viewport width".
That way the text size will follow the size of the browser window.

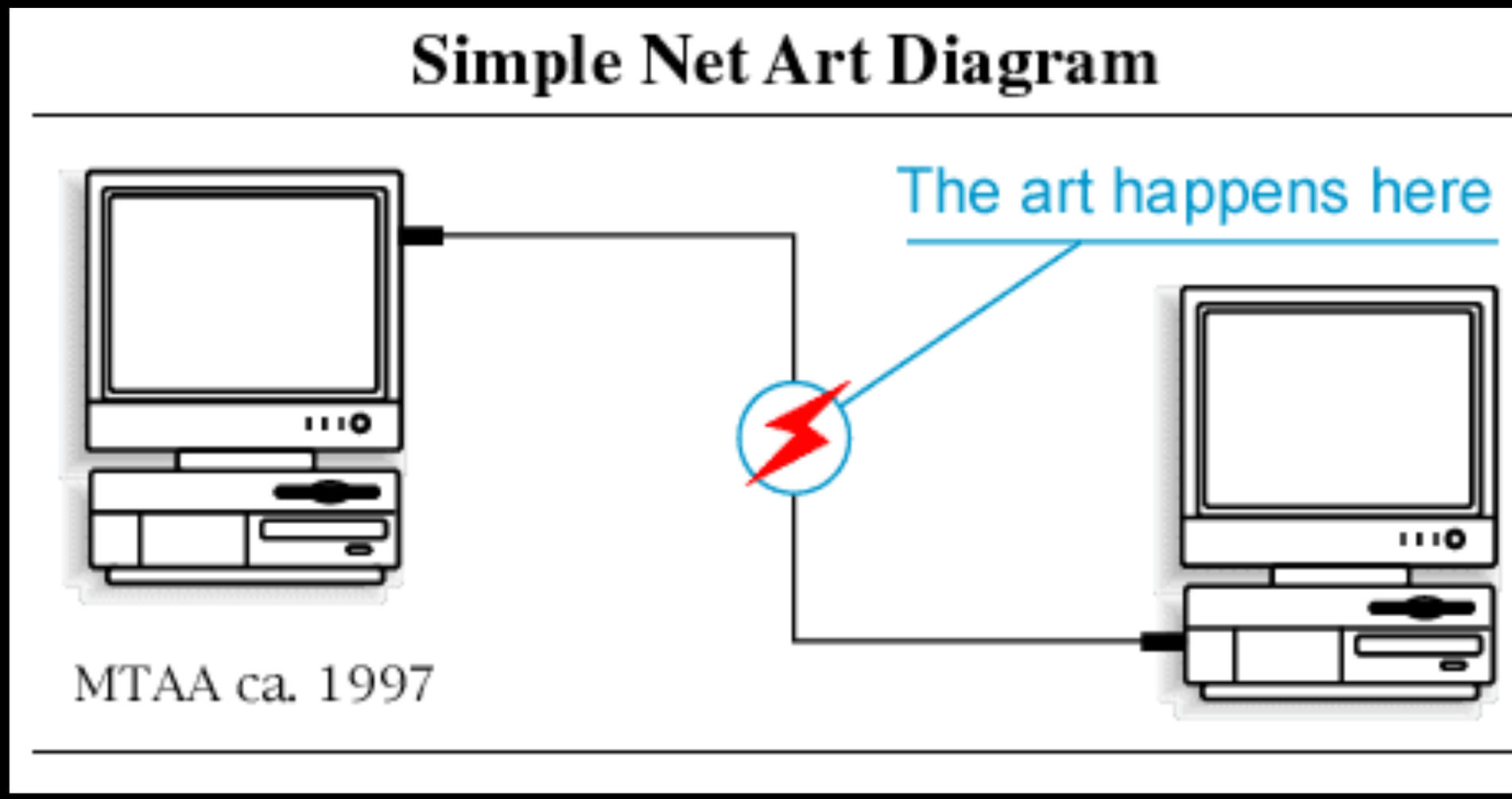
```
<h1 style="font-size:10vw">Hello World</h1>
```

Responsive Frameworks

Responsive frameworks are templates of responsive stylesheets

Examples: W3.CSS, Bootstrap (JavaScript)

Examples: https://www.w3schools.com/html/html_responsive.asp



"Art made online exists solely if realized by both author and viewer, since an experience of the work is possible only once, or if, it is accessed. Until then, its location remains undefined and unmapped, without any physical manifestation." (Tribe/Jana, 2006)

Midterm

Your midterm is to create a website using HTML + CSS. It must contain at least three layers of HTML file parenting, clean + heavily commented code, interactive CSS + thoughtful + appealing design and a title.

You must come up with a project proposal. A slide deck or pdf that illustrates the concept and purpose of your site as well as a sitemap and wireframes for the different pages. This is due **Thursday October 4th** + **October 11**. We will (quickly) look at everyone's proposal on those days.

Projects should be hosted on your Storm server in it's unique directory + url. URLs should posted to the wiki before **4pm** on **Monday October 15**. You will spend 5 minutes presenting your midterm to the class either Monday or Thursday the 18th.

Midterm

1. Create a hyper text narrative or net art piece
2. Invent a fictional or futuristic product (a machine that records your dreams, a shoe that plays music, etc) + create a website for it. **Inspiration**
3. Create a website for something / someone in your life. Your band, your Mom's ceramics hobby, etc. **Inspiration**