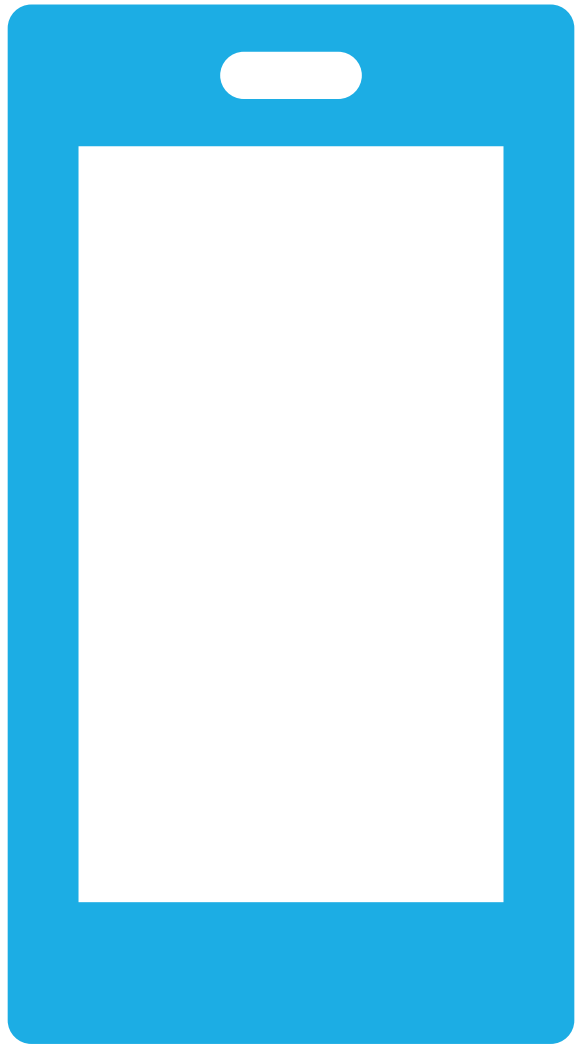
MEDIA QUERIES



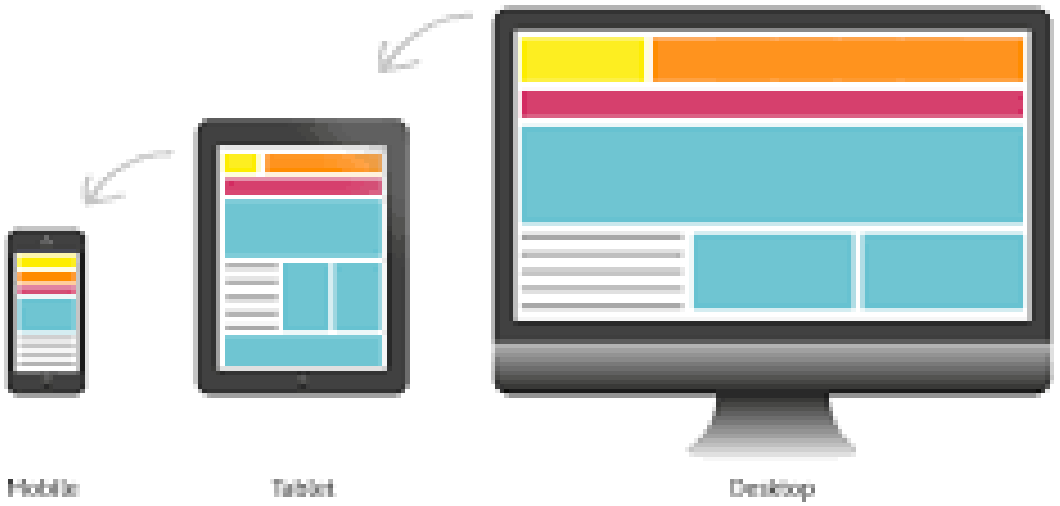
RESPONSIVE DESIGN

Before we start with the technical side of media queries, we should cover the idea of responsive design.

Responsive design is the practice of enabling our websites to change their style based on the current screen size. While the website will still hopefully (but not always) contain the same content, the layout may change drastically.

For example, you might see a very horizontal layout for a website on desktop but when you move to mobile, you see a very vertical layout. To give our users the best experience on our site, we need to adapt to all possible screen sizes to the best of our ability.

RESPONSIVE DESIGN CONT.



RESPONSIVE DESIGN CONT.

While tools like grid allow us to adapt to certain screen sizes, we still need a bit more control as we might want to change major elements of our page.

For example, you might want to hide an image on mobile that has enough room for desktop and tablet layouts. In general there are 3 layouts we want to design for at the start:

- Mobile
- Tablet
- Desktop

Note that this does not mean these are the only screen sizes! This can depend on your client. You may need to design for a huge screen if you know the client will be using the page primarily on large TVs.

RESPONSIVE DESIGN CONT.

So when it comes to designing for the masses, where do we start?

In recent years the use of mobile devices to browse the internet has dominated the market. For the most part, users are on their phones when visiting your site. This leads us as designers to want to design for this experience **first**

This idea is called **mobile first design.** We design our websites for the devices we expect our users to be on, and then move to larger screen sizes.

MOBILE FIRST / PROGRESSIVE ENHANCEMENT

Another name for mobile first design is **progressive enhancement**. The idea is the same (design for mobile first) but the motivation comes from a different source than designing for the most amount users.

The motivation for progressive enhancement is to make sure your design works for the absolute worst case device in terms of **power and size**. In most cases this means phones (although they have been getting more and more powerful).

When moving to tablet and eventually desktop design, you can start to include more intensive design elements (more stuff on the page, etc.).

DESKTOP FIRST / GRACEFUL DEGRADATION

Another school of design thought takes the opposite approach is called **graceful degradation**. Here you design for the **best case scenario** for your users in terms of size and power first.

This means assume they are on a top notch computer with a nice high resolution screen size and take full advantage. Fill the page with CPU intensive operations and animations and assume the user can handle it.

From there you can start cutting things out of the page as your users screen size and power goes down.

This is **not** the recommended way to design your sites. It forces you to ask the question “what do we cut out?” where as mobile first allows to you ask “what can we add in?”.

CSS MEDIA QUERIES

With all this theoretical talk about responsive design and mobile first, how do we actually implement these principles in our code?

The answer: CSS media queries and breakpoints! This is code that allows us to have rules that only apply to specific screen sizes. You can write rules that apply to mobile, tablet, and finally desktop.

As we dig into the technical code, keep in mind we want our default CSS rules to be for **mobile** and that CSS is read from **top to bottom** with higher rules being overwritten by lower rules.

CSS MEDIA QUERIES CONT.

So the general breakdown of our CSS files should be as follows:

1. Write CSS rules intended for **mobile** at the **top** of our CSS files
2. Write a **media query** for **tablet** sizes in the **middle** of our CSS files
3. Write a **media query** for **desktop** sizes at the **end** of our CSS files

With that being said, let's learn how to write a media query.

A media query can be thought of as a block of CSS rules that are only applied in certain scenarios. In normal speak, it can be thought as "only apply these rules when the screen size is X"

CSS MEDIA QUERIES CONT.

Let's take a look at an example that will apply only when the screen size is **at least** 600px (a decent tablet breakpoint):

```
<MOBILE RULES UP HERE>


@media only screen and (min-width: 600px) {
    #mobile-logo {
        display: none;
    }
}


<DESKTOP MEDIA QUERY  / RULES DOWN HERE>
```

KNOWLEDGE CHECK

Let's see if we can get specific rules to apply on certain screen sizes:

1. Create a folder called MediaQuery in your Scratch folder
2. Connect this folder to Git/GitHub
3. Create a simple index.html and style.css
4. Add a header tag that takes up the whole width of the screen and is 75px tall
5. Give the header a blue background color on mobile
6. Add in a media query for tablet
7. Give the header an orange background color on tablet
8. Test to verify the background color changes from one screen size to the next
9. Add, commit and push your code

CSS MEDIA QUERIES CONT.

You are not limited to having only 3 screen sizes, you can create as many breakpoints as you like. If you want to have a breakpoint for every 100 pixel increase you can do so.

To start I recommend 3 sizes (so 2 breakpoints)

- Default rules for mobile (no breakpoint)
- Breakpoint at somewhere between 600-768 pixels (you choose) for medium devices like tables
- Breakpoint at around 1000 pixels for large devices like laptops and desktops.

If you want to have a more specific layout for a 27 inch screen vs a 30 inch screen you are free to do so.

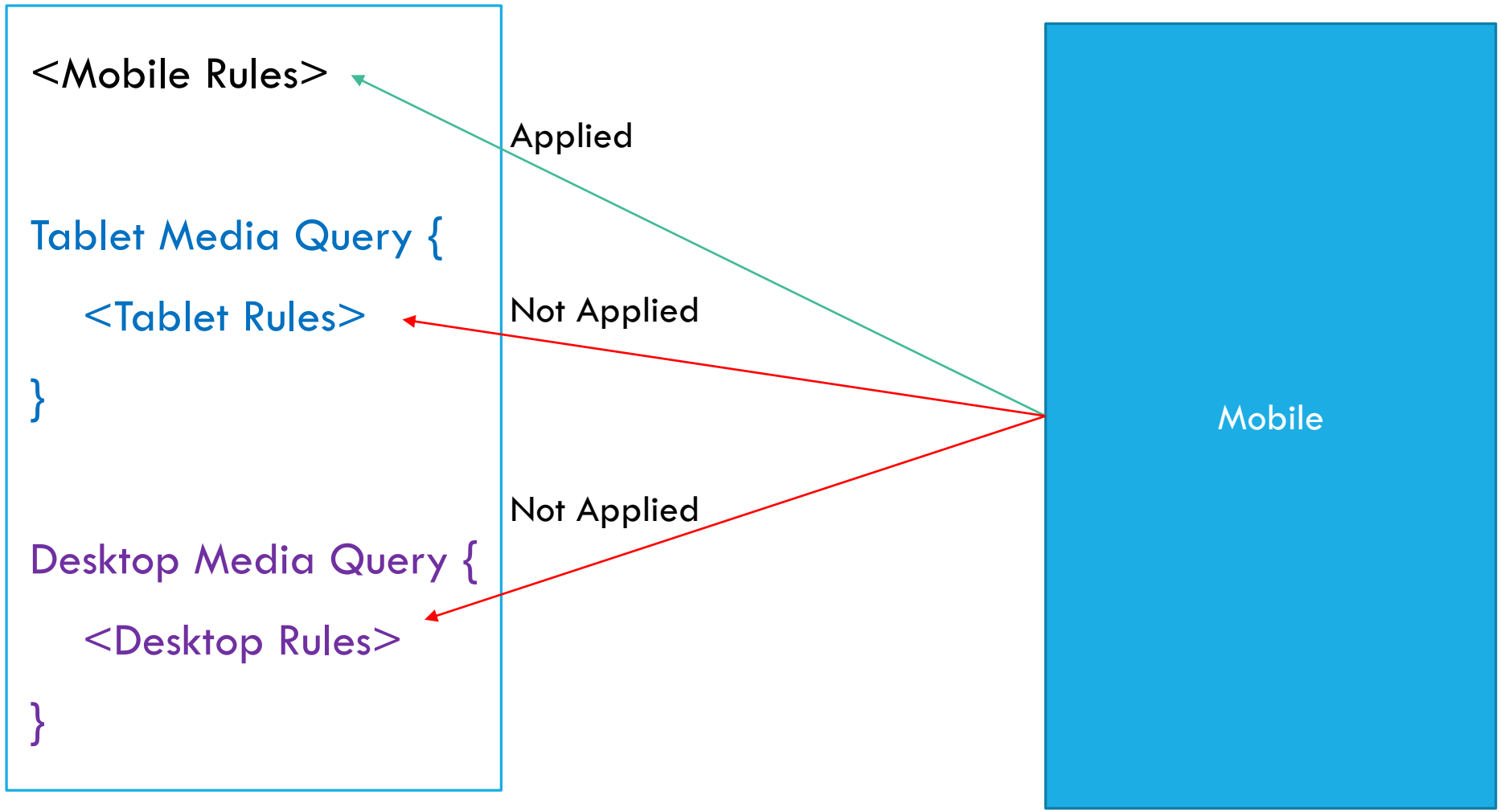
MEDIA QUERY GOTCHAS

Media queries tend to confuse new developers as it is usually the first time they need apply logic to their markup code. Instead of just writing the code, they now need to think ahead and plan where this code should be written.

Some things to remember about the way we are writing out media queries:

1. We are writing progressive enhancement CSS code. The confusing thing about this is that our rules written for mobile are **still applied** in our desktop code.
2. Specificity still matters! A more specific rule in mobile will overwrite a less specific rule in the desktop media query even when the user is on a desktop computer.
    1. Remember if there is a tie in specificity, the lower rule wins. This is great for overwriting mobile runs in the desktop section with the same selector.

CSS BEING APPLIED



CSS BEING APPLIED

<Mobile Rules>

Tablet Media Query {

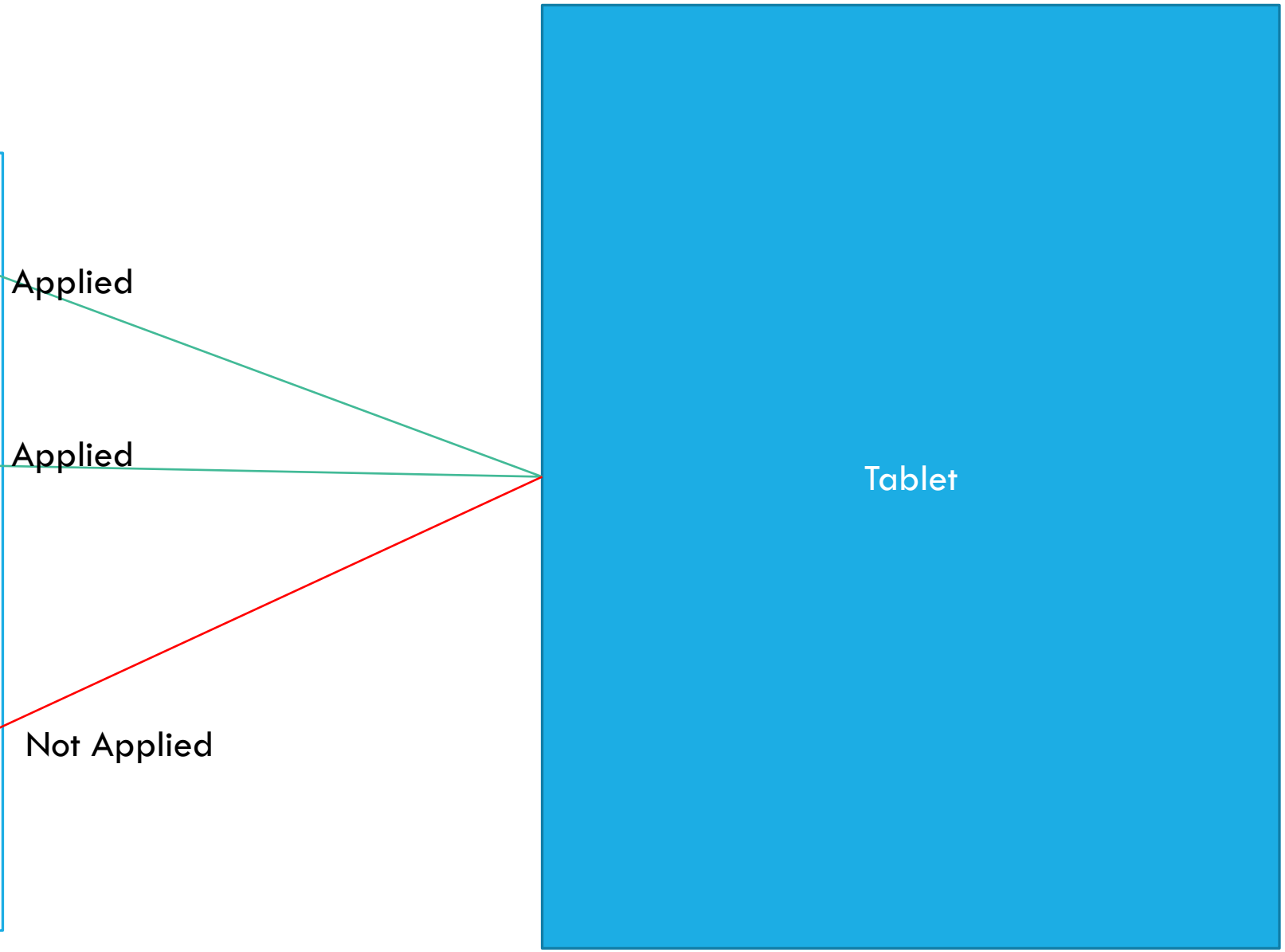
    <Tablet Rules>

}

Desktop Media Query {

    <Desktop Rules>

}



CSS BEING APPLIED

<Mobile Rules>

Tablet Media Query {

<Tablet Rules>

}

Desktop Media Query {

<Desktop Rules>

}

Applied

Applied

Applied

Desktop

KNOWLEDGE CHECK

Let's see if we can get specific rules to apply on certain screen sizes:

1. Use the MediaQuery folder from before
2. Add in a media query for desktop
3. Give the header a purple background color on desktop
4. Test to verify the background color changes from one screen size to the next
5. Add, commit and push your code

MEDIA QUERIES

Try to handle as much of the changes as possible using our responsive grid layout tools, but you will need to rely on media queries for many things.

This is also where you will start to see more logical errors so stay sharp!