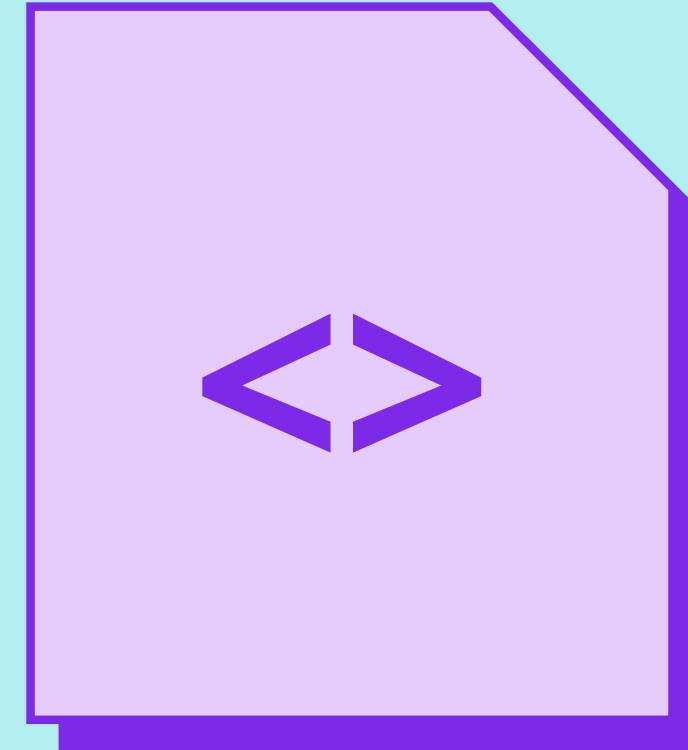




Mobile CSS



Prepared by
Mark Gurevich



Devices

Users could open a website on a lot of different devices: desktop, mobile, tablet, tv...

And developers responsibility is to build a website in the way that it would works properly wherever it has been opened

Mobile

Mobile web worldwide
traffic grows every year

And it's already ahead
of desktop

Desktop
49%



Mobile
51%

Mobile version

One of the ways to provide a good experience to mobile users is to build a separate version of the website which will be optimised to mobile devices

When a user opens your website from mobile device they will be redirected to mobile version (usually leaves on sub-domain which starts with "m." - m.website.com)

Mobile version

Pros:

- Well optimised for mobile
- Possible to build a different UX flow for mobile and desktop
- Easier to debug

Cons:

- Have to build 2 websites
 - Time
 - Make sure that changes were applied on both versions
- False detections
- Handle SEO problems

Responsive Website

The another way to build a website which will works good on both mobile and desktop is to make it responsive by CSS

It will change how it looks like based on the size of the screen

Responsive Website

Pros:

- Time
- Always have the same functionality whatever device is used

Cons:

- Extra code that used only for the one version of the website
- Harder to provide the best UX in all the cases

Media queries

Allows to create CSS rules which are applied to the document only when device reach specific criteria

```
.article {  
  padding: 5px 10px;  
}
```

```
@media (min-width: 600px) {  
  .article {  
    padding: 10px 20px;  
  }  
}
```


Media Types

```
// All the devices  
@media all { ... }
```

```
// Print mode  
@media print { ... }
```

```
// Screen devices  
@media screen { ... }
```

```
// Speech synthesizers  
@media speech { ... }
```

Media Features

```
// 500px and narrower (e.g. a phone)  
@media (max-width: 500px) { ... }
```

```
// 501px and wider  
@media (min-width: 501px) { ... }
```

```
// Primary input can hover  
@media (hover: hover) { ... }
```

```
// Dark mode preference  
@media (prefers-color-scheme: dark) ...
```

Multiple criteria

```
@media screen
  and (min-width: 320px)
  and (orientation: portrait) {
    ...
  }
```

Negate query

```
// Invert the whole media query
@media not screen
  and (min-width: 320px)
  and (orientation: portrait) {
  ...
}
```

Combine queries

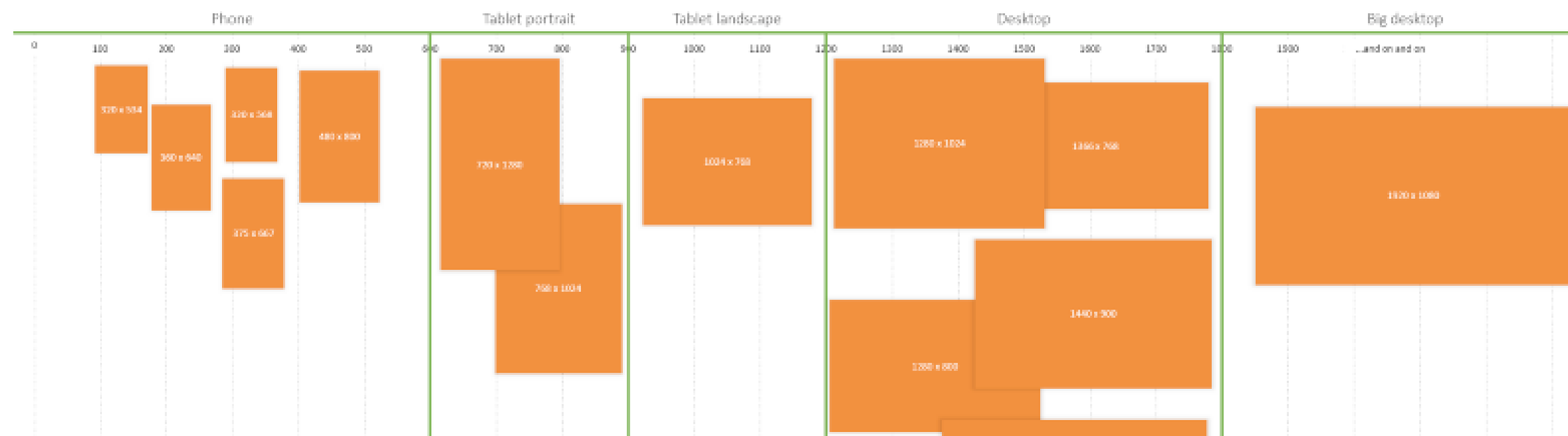
```
// Applied to both print mode  
// and screen with width >= 320  
@media print, screen  
  and (min-width: 320px) {  
  ...  
}
```

Viewport

Represents the currently viewed area of the page.

In CSS pixels. High resolution screens display multiple physical pixels per CSS pixels.

On mobile viewport not always equal to the size of the device, by default. It is wider than the screen and renders zoomed out.



Viewport Meta Tag

You can control the size and scale of the viewport by the meta tag

```
// Sets viewport size to the actual screen  
// size of the device
```

```
<meta  
  name="viewport"  
  content="width=device-width"  
>
```

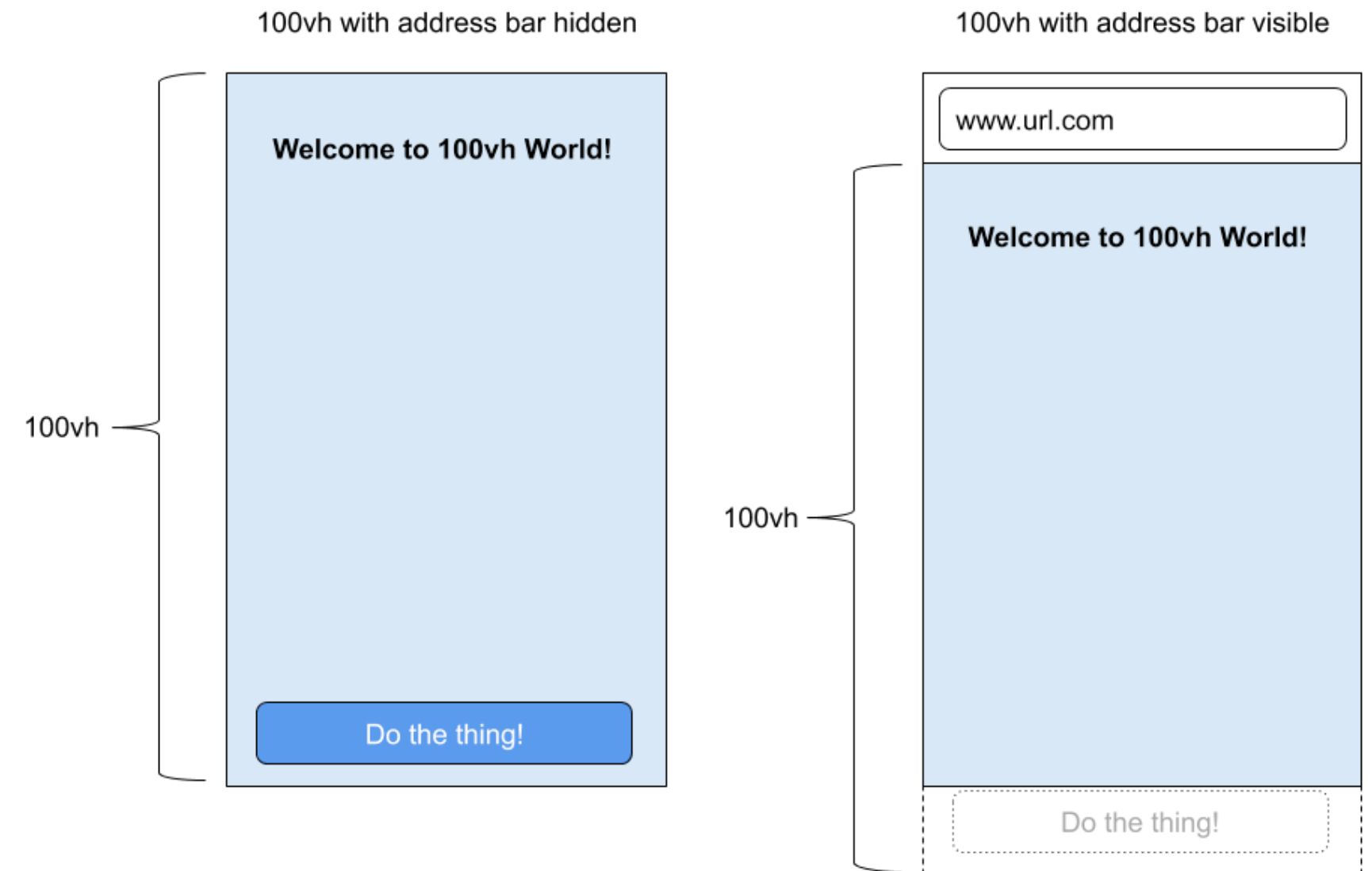
```
// Controls the zoom of the page
```

```
<meta  
  name="viewport"  
  content="initial-scale=1, maximum-scale=2"  
>
```

Viewport units

```
.box {  
  // 10% of viewport width  
  width: 10vw;  
  // 10% of viewport height  
  height: 10vh;  
}
```

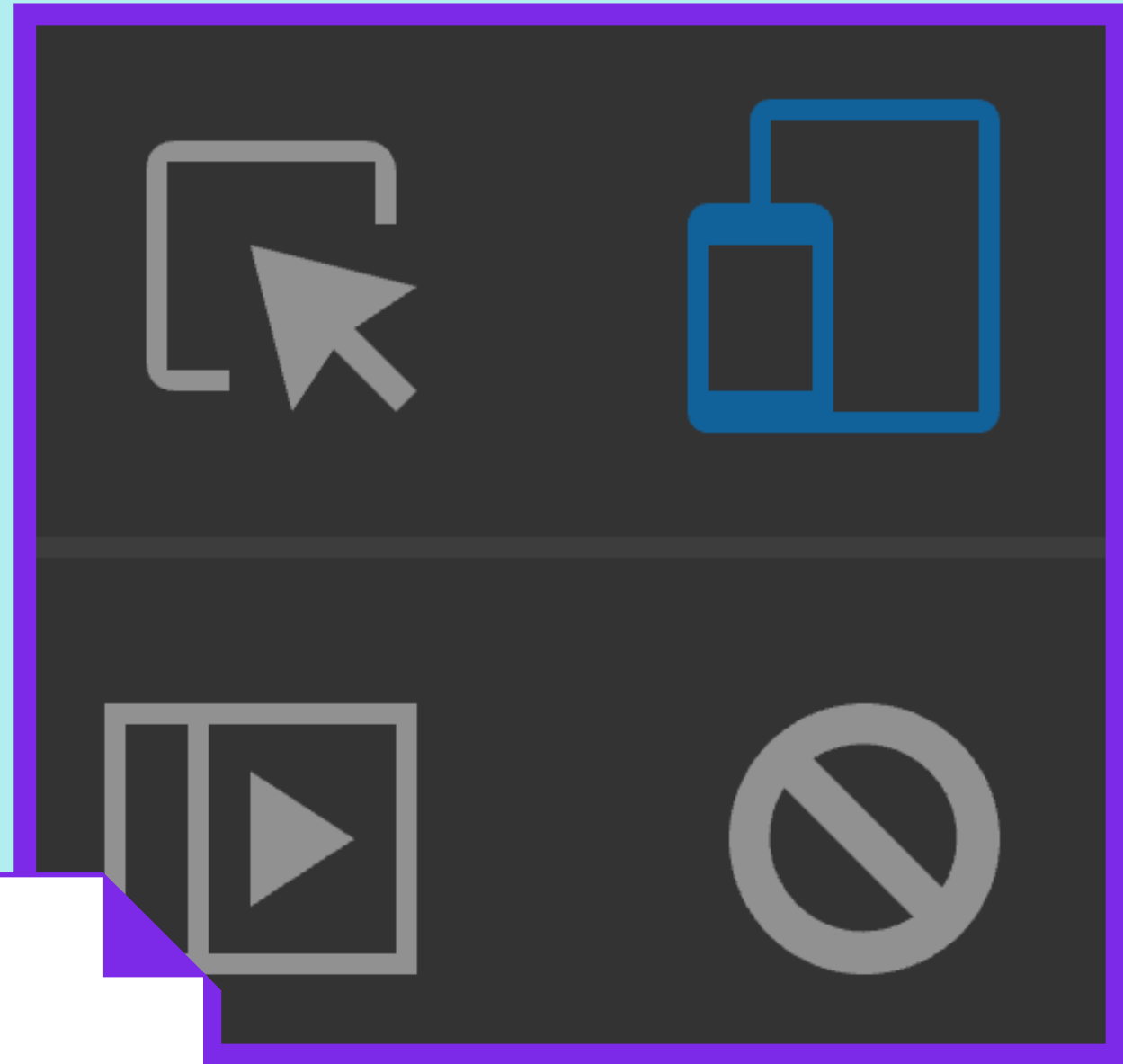
Warning, VH is counter-intuitive



<https://chanind.github.io/javascript/2019/09/28/avoid-100vh-on-mobile-web.html>

Dev Tools

Chrome DevTools has an instrument that allows to change the viewport



Device Mode

You can change the size of the viewport in Chrome device toolbar by choosing of the predefined devices or set the custom sizes

Note: its only change the viewport, not emulates the actual device

It also allows to change Orientation and Device Pixel Ratio