Viewport and Media Queries

The Complete Idiot's Guide

last updated on 4/4/2011

Viewport and Media Queries

Tiny little pixels can be big headaches (if you read *a pixel is not a pixel is not a pixel*, you know what I am talking about). Everybody talks about viewport and pixels, few knows what they are about, even fewer can explain them in an ***idiot*** and ***complete*** way.

Nevermind the pixels, here comes the ***Complete Idiot's Guide to Viewport and Media Queries****!!!*

viewport

1. viewport - width

You can define the width of viewport. If you don't use width=device-width, on iOS your page stretches over the full available width of the layout viewport, which is 980px, if you use width=device-width, your page will fit into screen size of 320px. So in boilerplate, we use:

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

1. viewport - width

Question:

Should I always use width=device-width?

Answer:

If your site is a fluid layout desktop design, using it is generally better when viewing the site on mobile devices.

If your site is a fixed layout desktop design, it’s up to you, doesn’t matter.

If your site is a **mobile specific** design, **just use it**.

2. viewport - target-densitydpi

Before talking about this property, let's briefly talk about pixel. The easiest way to explain this is to think about your desktop, on the same monitor if you adjust resolution, your desktop icons look smaller when you use high resolution, and bigger when you use low resolution.

Android introduced **`target-densitydpi`**. When you set **`target-densitydpi=device-dpi`**, with the same physical phone size, images/text will look smaller on a high resolution device, and bigger on a low resolution device.

2. viewport - target-densitydpi

Question:
So what's the problem with `target-densitydpi=device-dpi`? We are fine with it when we adjust our monitor right?

Answer:
The problem is that a high resolution device can be twice as high as a low resolution one, with the same physical device size, such big difference can make your pixel related everything shrink into half the size.

2. viewport - target-densitydpi

Question:
Is there a iPhone equivalent of target-densitydpi=device-dpi

Answer:
Not really. In iPhone, you can only specify width=device-width. Regardless of whether it's iPhone 3 or 4, the device-width is always 320px in portrait mode.

Question:
How about Window Phone?

Answer:
Windows phone also uses 320px as device-width in portrait mode, which is why it's good to make Android behave the same way as iPhone and Windows Phone. The consistency will make it much easier when styling.

3. viewport - scaling

On most smartphones, default scaling may occur that will 'zoom' your pages on mobile browser. To prevent user scaling, you can use **`initial-scale=1.0.`** So now, below is the full version of our '**viewport boilerplate**':

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

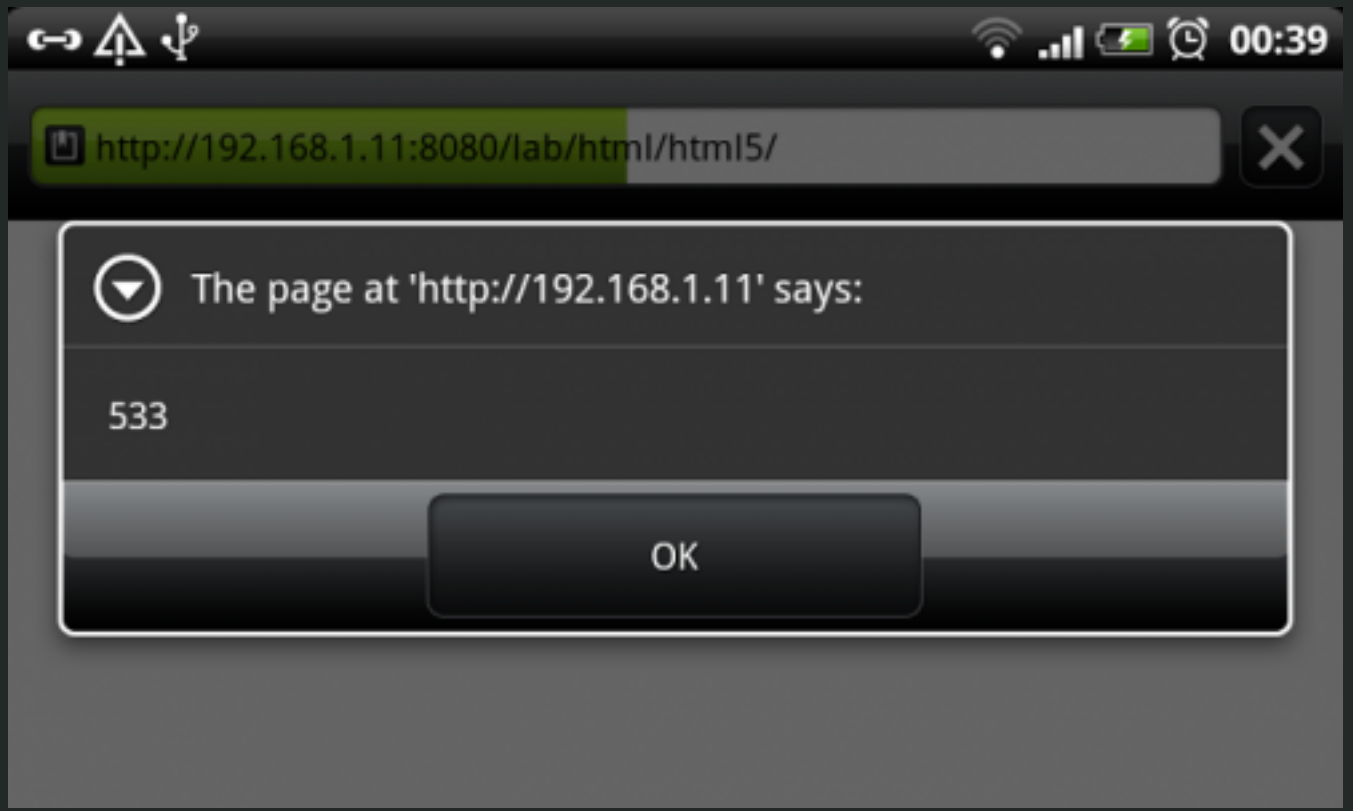
Media Queries

4. CSS Media Queries - max/min-device-width

If you want to have a CSS style that only smartphones will pick up, use:

`@media only screen and (min-device-width : 320px) and (max-device-width : 569px) {...}`

Example of HTC Desire HD, it has a width of 533 in landscape mode

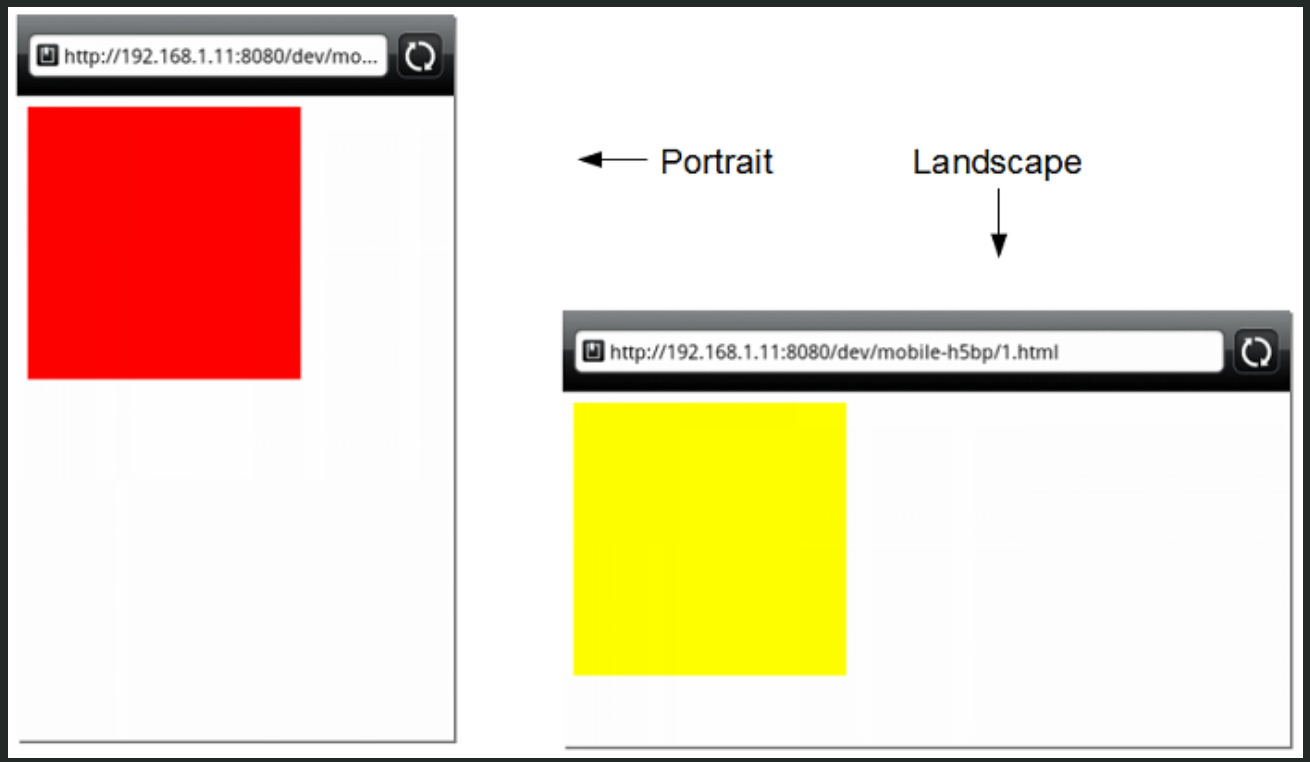


5. CSS Media Queries - max/min-width

max-width and min-width are updated when you rotate the screen, your max-width in portrait mode become max-height in landscape mode

```
@media only screen and (min-width : 480px) { .box { width:200px; height:200px;
background:yellow; } }
@media only screen and (max-width : 320px) { .box {width:200px; height:200px;
background:red; } }
```

below screen will be rendered when the above style applied:



5. CSS Media Queries - max/min-width

Question:

Is max/min-width the same as max/min-device-width?

Answer:
If you set width=device-width, then max/min-width is the same as max/min-device-width inportrait mode, and different when in landscape mode.

In short, the difference is max/min-width will update when you rotate the screen, whereas, max/min-device-width won't.

6. CSS Media Queries - device-pixel-ratio

device-pixel-ratio could help you know the resolution of the device screen, iPhone 4 and certain other smartphones have pixel ration equal or higher than 1.5, so if you want to target high resolution devices, you can use the media query below:

```
@media only screen and (-webkit-min-device-pixel-ratio : 1.5),
       only screen and (-o-min-device-pixel-ratio: 3/2),
       only screen and (min-device-pixel-ratio : 1.5) {
  .imagebox {background:(url:"images/high/demo.jpg");}
}
```

Be cautious!!! when you are using it in the above way, the image will be downloaded even if the page doesn't fit the rule above.

7. CSS Media Queries - orientation

iPad introduced 'orientation' to help with detecting orientation change, below is how you can specify landscape and portrait style

```
/* iPads (landscape) ----------- */
@media only screen and (min-device-width : 768px) and (max-device-width :
1024px) and (orientation : landscape) {
  /* Styles */
}
/* iPads (portrait) ----------- */
@media only screen and (min-device-width : 768px) and (max-device-width :
1024px) and (orientation : portrait) {
  /* Styles */
}
```

Thank you

Brought to you by Mobile Boilerplate http://html5boilerplate.com/mobile/

Shi Chuan, Paul Irish, Divya Manian