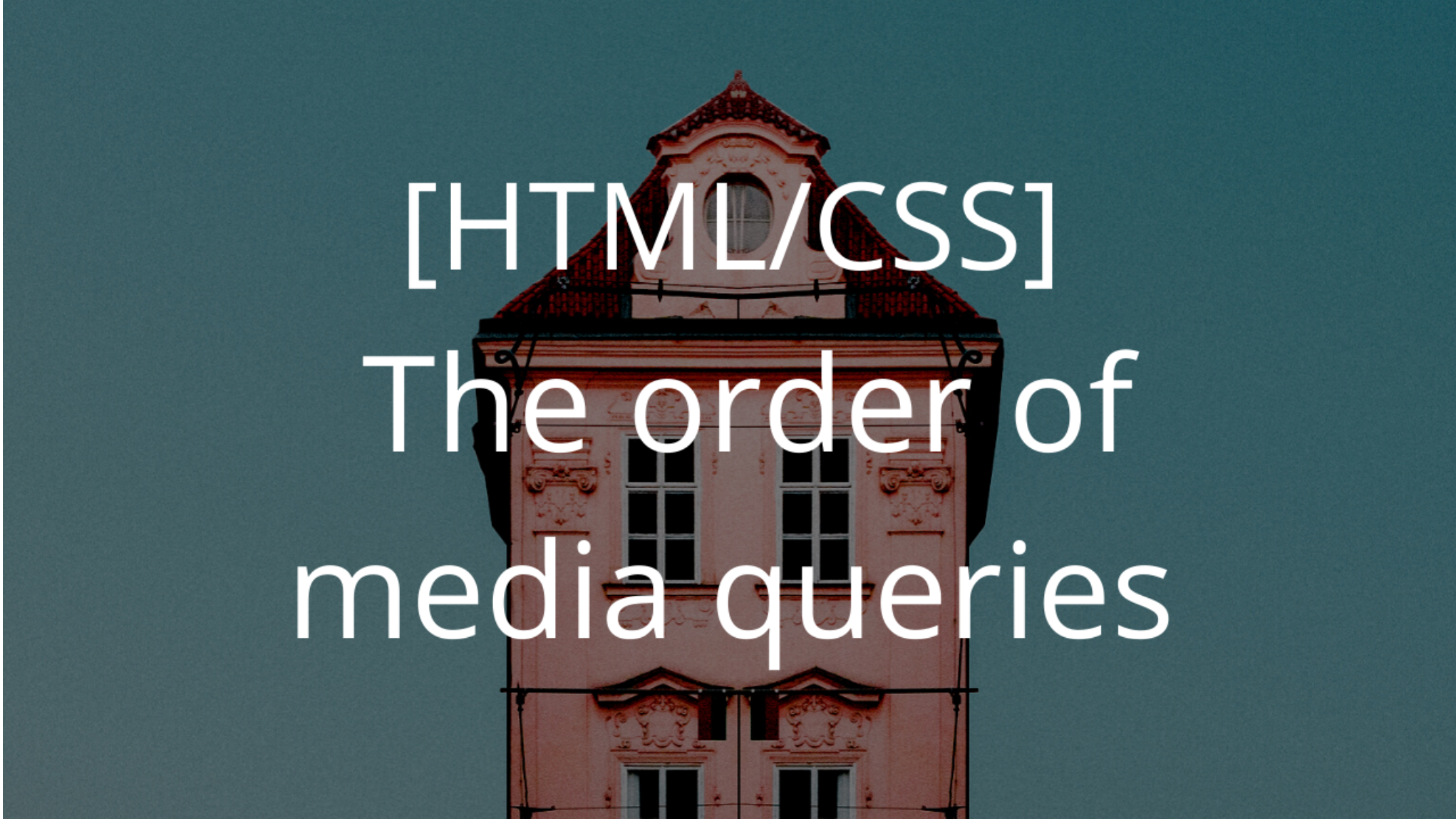


[HTML/CSS]The order of media queries

Published by **Shuhei** on 04/22/2020



In this article, I will explain how to decide the order of media queries. Here I will only use CSS to specify the media queries. There are 3 main ways to code media queries.

- Desktop-first media query
- Mobile-first media query
- Specify media queries individually

This article will explain each type of media queries and when to use them.

Table of Contents

1. Desktop-first media query
2. Mobile-first media query
3. Specify media queries individually
4. Be careful with the order of media queries!
5. Effects from other media queries

Desktop-first media query

```
1  /*
2     CSS for PC desktop (bigger than 1020px)
3  */
4
5  @media screen and (max-width:1020px){
6     /*CSS for tablet (bigger than 767px and smaller than 1020px)*/
7  }
8
9  @media screen and (max-width:767px){
10     /*CSS for mobile phone (bigger than 480px and smaller than 767px)*/
11  }
12
13 @media screen and (max-width:480px){
14     /*CSS for mobile phone(smaller than 480px)*/
15 }
```

CSS

This style is called Desktop first media query because the CSS code outside of those media queries defines the style for desktop view. In this style, we have to use "max-width" and we also have to code the larger view fast. This style is useful when your design is based on the desktop view. You can first complete the base CSS code for the desktop view, and when you need you can change some parts with these media queries.

Mobile-first media query

```
1  /*
2     CSS for mobile phone (smaller than 480px)
3  */
4
5  @media screen and (min-width:480px){
6     /*CSS for mobile phone (bigger than 480px and smaller than 767px)*/
7  }
8
9  @media screen and (min-width:767px){
10     /*CSS for tablet (bigger than 767px and smaller than 1020px)*/
11  }
12
13 @media screen and (min-width:1020px){
14     /*CSS for PC desktop (bigger than 1020px)*/
15 }
```

CSS

This style is called Mobile-first media query because the CSS code outside of those media queries defines the style for the mobile phone view. In this style, we have to use "min-width" and we also have to code the smaller view first. This style is useful when your design is based on the mobile phone view. You can first complete the base CSS code for the mobile phone view, and when you need you can change some parts with these media queries.

Specify media queries individually

```
1  @media screen and (min-width:1020px){
2     /*CSS for PC desktop (bigger than 1020px)*/
3  }
4
5  @media screen and (min-width:768px) and (max-width:1020px){
6     /*CSS for tablet (bigger than 768px and smaller than 1020px)*/
7  }
8
9  @media screen and (min-width:481px) and (max-width:767px){
10     /*CSS for mobile phone (bigger than 481px and smaller than 767px)*/
11  }
12
13 @media screen and (max-width:480px){
14     /*CSS for mobile phone(smaller than 480px)*/
15 }
```

CSS

In this style, we define each media query individually. This is also useful because you don't have to care about the order of the media queries and each media query is independent so other media queries have no effect on a specific media query. This style can be used when your design is very different between devices.

Search ...



Shuhei Tsuruya

A bachelor student majoring computer science at Osaka University. Learning web development and also interested in AI, ML. If you are interested, feel free to contact me!

Follow @ShuheiTsuruya

Archives

May 2020

April 2020

Be careful with the order of media queries!

When you use the desktop-first or mobile-first style, you have to be careful with the order of media queries. In desktop-first style, the larger view has to be coded earlier than a smaller view and in mobile-first style, the vice-versa.

If you fail to follow this order rule, it will cause styling error.

CSS

```
1  /*CSS for PC desktop (bigger than 1020px)*/
2  body{
3      color:red;
4  }
5
6  /* this media query comes earlier than the one below.
7     So this will cause an error and this style will not
8     applied to the view */
9  @media screen and (max-width:767px){
10     /*CSS for mobile phone (bigger than 480px and smaller than 767px)*/
11     color:green;
12 }
13
14 @media screen and (max-width:1020px){
15     /*CSS for tablet (bigger than 767px and smaller than 1020px)*/
16     color:blue;
17 }
18
19 @media screen and (max-width:480px){
20     /*CSS for mobile phone(smaller than 480px)*/
21     color:yellow;
22 }
```

If you specify media queries individually, this error will not happen so you don't need to be careful with this.

Effects from other media queries

When you use the desktop-first or mobile-first style, one media query will be affected by other media queries. In the desktop-first style case, the CSS for a larger view is automatically applied to the other smaller views and in the mobile-first style case, the vice-versa. For example, when you use the desktop-first style and when you specify a text color to be red in the PC view, the text is still red in tablet and mobile view.

CSS

```
1  h1{
2      color: red;
3  }
4
5  @media screen and (max-width:1020px){
6      h2{
7          color:blue;
8      }
9      /* h1 is still red */
10 }
11
12 @media screen and (max-width:767px){
13     h3{
14         color:green;
15     }
16     /* h1 is still red */
17     /* h2 is still blue */
18 }
19
20 @media screen and (max-width:480px){
21     h4{
22         color:yellow;
23     }
24     /* h1 is still red */
25     /* h2 is still blue */
26     /* h3 is still green */
27 }
```

Therefore, you should be careful with the effects from other media queries when you use desktop-first or mobile-first style. However, when you define each media queries individually, this problem will not happen and one media query will not be affected from any other media queries.

Categories: [HTML/CSS](#) [UNCATEGORIZED](#)

Tags: [HTML/CSS](#) [Media Query](#)



Related Posts

[HTML/CSS]
Prevent the child
element to overflow
its parent

[HTML/CSS](#)
**[HTML\CSS] Prevent the child
element to overflow its parent**
When we code HTML/CSS, we often see that the
child elements overflow its parent.