如何对 position absolute 进行水平对中

position: absolute;

margin-left: auto;

margin-right: auto;

left: 0;

right: 0;

如果屏幕居中： 水平和垂直都居中

1. 四周边框距离固定不变：

Display: block;

Position: absolute;

Top: 40px;

Left: 40px;

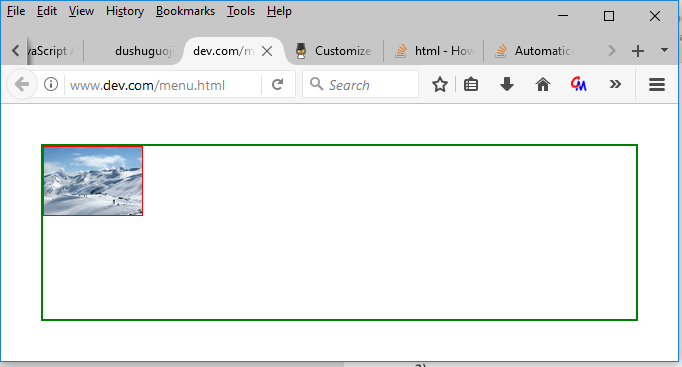
Right:40px;

Bottom: 40px;

<div style="display:block; position:absolute; width:200px; height:400px; left:0px; right:0px; margin:auto; border:2px solid green;">

    <img src="Canon03.jpg" class="img-responsive" style="position:absolute; width:100px; border:1px solid red;" />

</div>



1. 如果内容框大小不变，或者比例不变， 需要居中

Position: absolute;

Width: 200px;

Height: 300px;

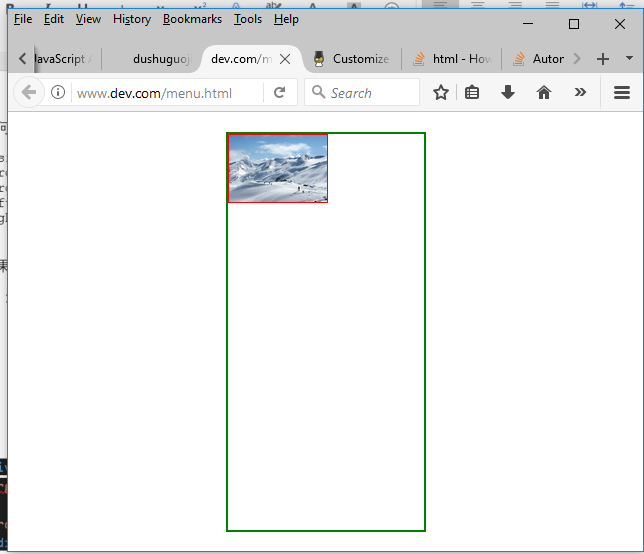
top: 0px; bottom:0px; left:0px; right:0px; //important

margin: auto; //important

<div style="display:block; position:absolute; width:200px; height:400px; top:0px; bottom:0px; left:0px; right:0px; margin:auto; border:2px solid green;">

    <img src="Canon03.jpg" class="img-responsive" style="position:absolute; width:100px; border:1px solid red;" />

</div>



1. 任何情况下的居中：

Top: 50%; 50% 父容器的高度，parent container height 50%

Left: 50%; 50% 父容器的宽度，parent container width 50%

Transform: translate(x, y) translate( -50%, -50%) 使用位移变化， 50% 是本身的宽度和高度

轻松实现任何元素的相对于父容器的居中问题：

<div style="display:block;position:relative; width:200px; height:400px; border:1px solid

purple;">

    <div style="display:inline-block;position:relative;border:1px solid navy; top:50%; left:50%; transform:translate(-50%, -50%);">Hello<br>You are good</div>

</div>

<div style="display:block; position:absolute; width:200px; height:400px; top:50%; left:50%;

transform:translate(-50%, -50%); border:1px solid purple;">

    <div style="display:inline-block;position:absolute;border:1px solid navy; top:50%; left:50%; transform:translate(-50%, -50%);">Hello<br>You are good</div>

</div>

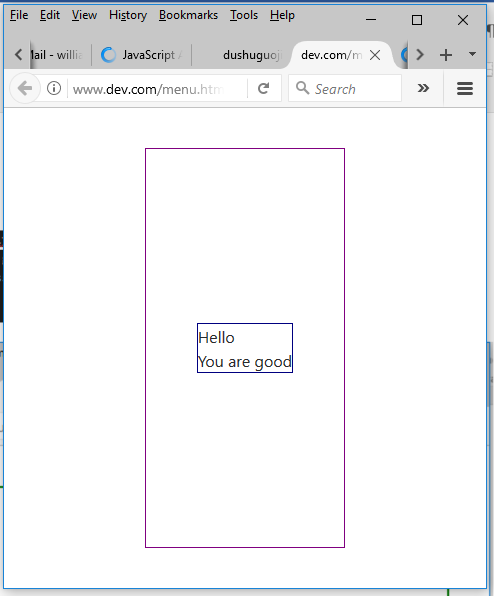


Image: 自动大小，而且不出界，同时屏幕居中的最佳解决方案：

1. image 跟随浏览器的resize 而自动调整大小，需要使用 bootstrap class=”img-responsive”

<img src="IMG\_3414.JPG" class="img-responsive" alt="Cinque Terre"/>

**以上不起作用因为还需要设置： max-width: 90% 或者 max-width: 90vw**

**注意我们如果只是设置 max-width: 90%; 但是高度没有设置，高度也有可能超界**

**max-width: 90%; -**

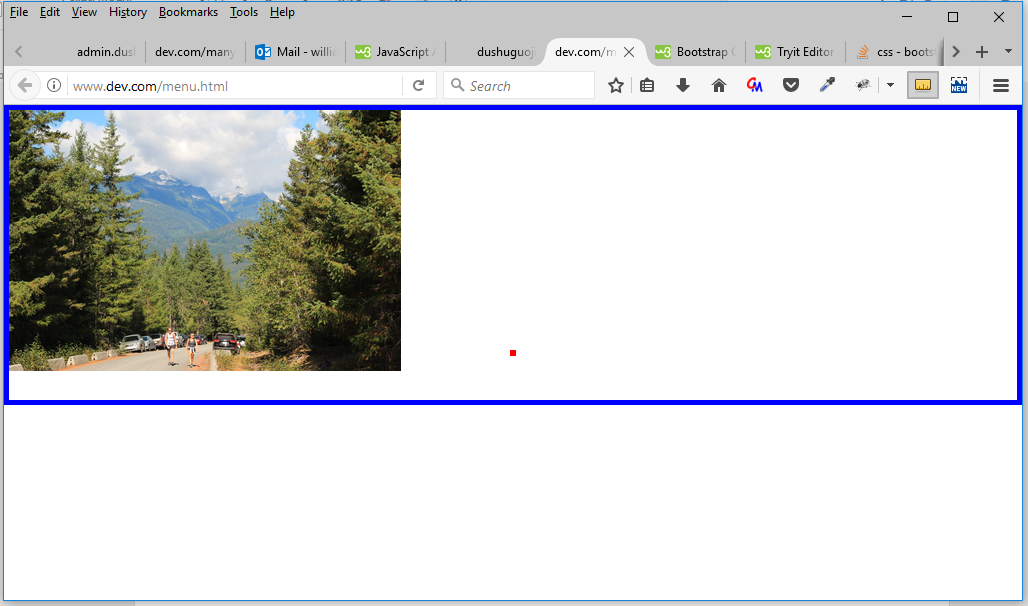
**max-height: 90%; - 高度仍然可能超级，因为 90% 是相对于整个 body 的高度， 而 body 高度是没有undefined , 所以图形高度相当于 auto;**

<body style="border:5px solid blue; height:300px;">

<img src="IMG\_3414.JPG" style="max-width:90%; max-height:90%;" alt="Cinque Terre"/>

</body>

**如果我们对 body 即 image 的父容器设置高度 300px; 那么 max-height: 270px;**



1. image 放入到 div 容器里： 如果image 宽和高 使用百分比 90% 会有问题

<body style="border:5px solid blue;">

<div style="

display:block;

position:absolute;

top: 50%;

left:50%;

transform: translate(-50%, -50%);

border:3px solid red;

box-sizing:content-box;">

     <img src="IMG\_3414.JPG" style="

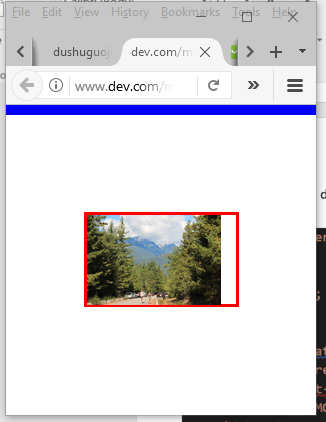
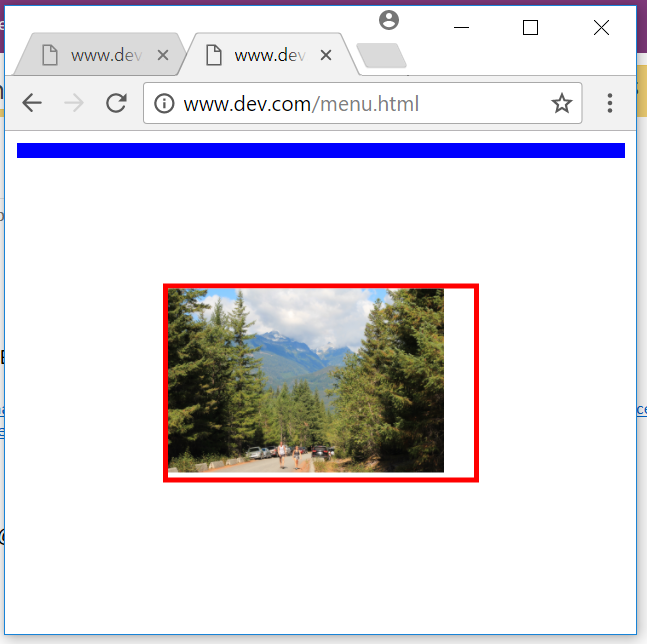
max-width:90%;

max-height:90%;"

alt="Cinque Terre"/>

    </div>

</body>

可以看到结果：

Image 与 div 容器包裹不完整, Image 会随容器大小而改变

1. Image 与 div 容器的完美融合， div容器的屏幕居中， image 的宽和高使用 viewport 百分比则完美解决

<div style="

display:block;

position:absolute;

top:50%; left:50%;

transform: translate(-50%, -50%);

border:1px solid red;

box-sizing:content-box;">

<img src="IMG\_3414.JPG" class="img-responsive"

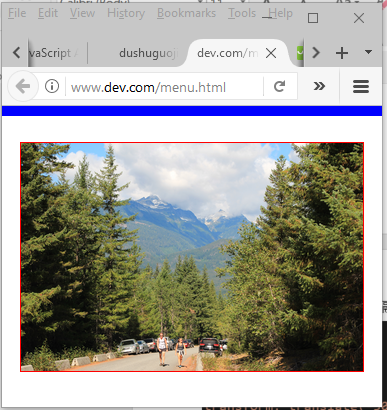
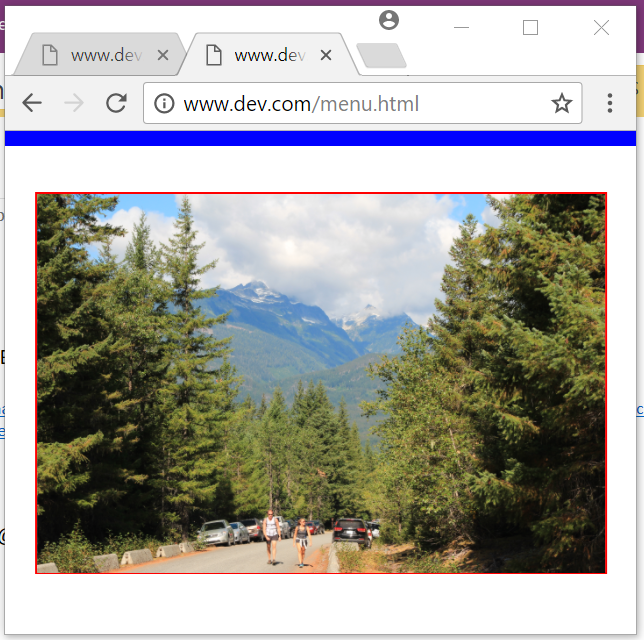
style="max-width:90vw; max-height:90vh;"

alt="Cinque Terre"/>

</div>

div 容器不需要制定最大的宽和高，只需要执行居中即可。

Image 需要 bootstrap img-responsive , max-width: 90vw max-height: 90vh 使用 viewport 的百分比更准确

CSS: 优先级问题：

1. !important – 优先级别最高，无论放在哪里都不会被覆盖掉，

在 style 里， 或者使用 all: initial 都不会被覆盖掉。

唯一可能被覆盖掉，就是同一个属性有多个 !important 那么也遵守优先级原则

1. 同一个位置： 后者定义优先级高过前者

            #d1 {

                background-color: pink;

                font-size:8px !important;

                font-size: 16px !important;

             }

最终是 16px

1. 同一文件里同一级别的定义： 后者定义的覆盖前者定义的， 以后者为主

    div.dc1 {

        text-align: center;

        font-size: 36px;

    }

    div.dc3 {

        text-align: left;

        font-size: 72px;

    }

    div.dc2 {

        text-align: right;

        font-size: 12px;

    }

最终是 font-size: 12px;

1. 相同等级的： 文件内部的定义 高于 外部文件的定义

如： 文件内部：

            div#dc4 {

                background-color: pink;

                font-size: 38px;

            }

Test1.css 和 Test2.css

    <link   href='/theme/wliu/test2.css' type='text/css' rel='stylesheet' />

    <link   href='/theme/wliu/test1.css' type='text/css' rel='stylesheet' />

Test1.css

div#dc4 {

font-size: 48px;

text-align: center;

}

Test2.css

div#dc4 {

font-size: 56px;

text-align: right;

}

最终是 font-size: 38px;

1. 相同级别在多个外部文件有定义: 后引用的覆盖前引用的文件定义

div id="tdiv" class="font56 font48"

class=”font56 font48” 这里的先后顺序和优先级无关

Test1.css 和 Test2.css

Test1.css

div.font48 {

font-size: 48px;

text-align: center;

}

Test2.css

div.font56 {

font-size: 56px;

text-align: right;

}

        <link   href='/theme/wliu/test1.css' type='text/css' rel='stylesheet' />

        <link   href='/theme/wliu/test2.css' type='text/css' rel='stylesheet' />

Font-size: 56px;

    <link   href='/theme/wliu/test2.css' type='text/css' rel='stylesheet' />

    <link   href='/theme/wliu/test1.css' type='text/css' rel='stylesheet' />

Font-size: 48px;

1. style -> #id -> .class style优先级高过 id , id 高过 class 定义
2. 同等级别的属性，限制条件越多的优先级越高

            div.dc1[lwh] {

                font-size: 48px;

            }

            div.dc1 {

                text-align: center;

                font-size: 36px;

            }

            div.dc3 {

                text-align: left;

                font-size: 72px;

            }

            div.dc2 {

                text-align: right;

                font-size: 12px;

            }

最终 div.dc1[lwh] font-size: 48px;

all: initial , inherit, unset

Initial – 设置成为初始值，浏览器对各种元素都有默认的初始值， 例如 width: auto; height:auto; float:none; div display:block

将所有属性设置为初始值：

All: initial 只会将此属性之前的属性重新设置，而之后的属性不受影响

<div style="

~~display:inline-block;~~

~~width:200px;~~

~~height:200px;~~

~~font-size:72px;~~

all:initial;

**后面的全部有效**

padding:20px;

border:20px solid blue;

box-sizing:content-box;

">Good Morning</div>

