



02 CSS

What is CSS



1. Colors
2. Backgrounds
3. Font Sizes
4. Layouts
5.

推荐阅读 <https://www.w3schools.com/css/>



What is CSS

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

CSS describes how HTML elements are to be displayed on screen, paper, or in other media.

External stylesheets are stored in CSS files

推荐阅读 <https://www.w3schools.com/css/>



What Do Fancy Websites Become Without CSS?

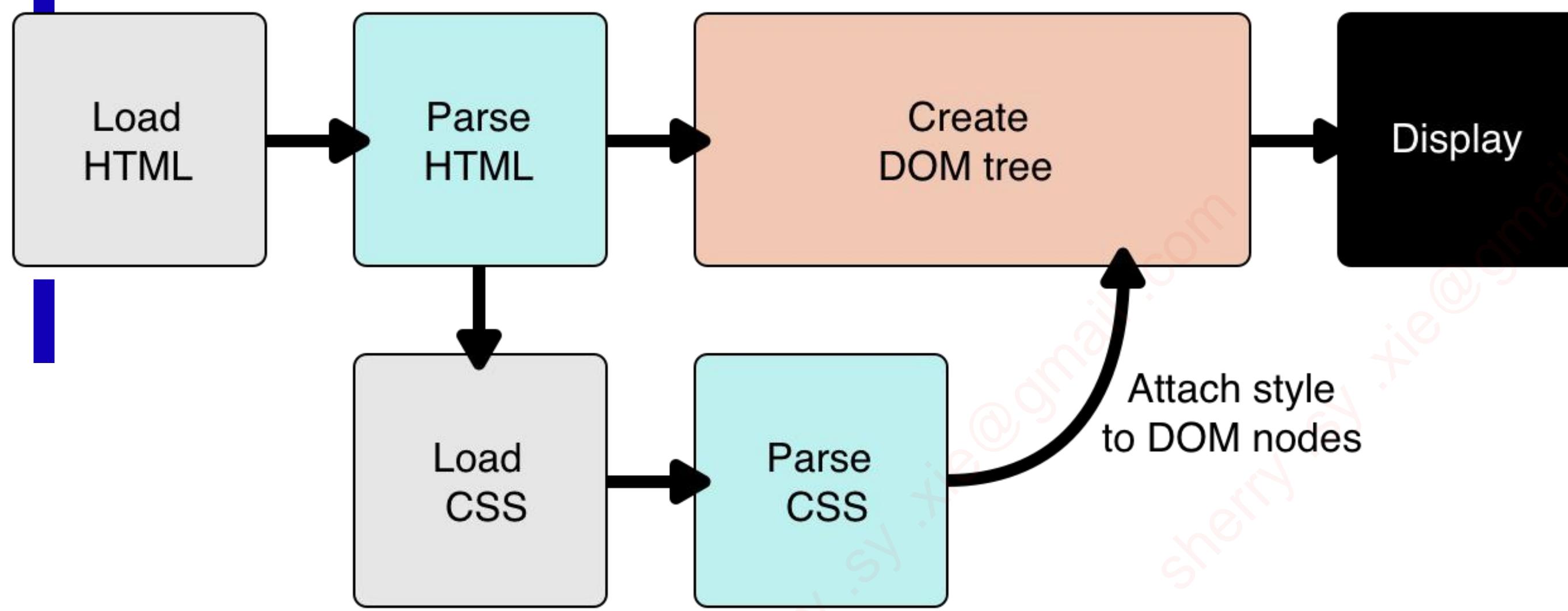


How does CSS actually work? - 1

<https://www.youtube.com/watch?v=tSv2KIF7uE4>



How does CSS actually work? - 2



1. The browser converts HTML and CSS into the DOM (Document Object Model). The DOM represents the document in the computer's memory. It combines the document's content with its style.
2. The browser displays the contents of the DOM.

推荐阅读 <https://www.w3schools.com/css/>



What is CSS

<http://www.w3.org/Style/CSS/>

CSS1 - 1996 - Fonts, Colors, Alignment, Spacing

CSS2.1 - 2004 - Layout, positioning

CSS3 - latest - Effects, sizing, speech

推荐阅读 <https://www.w3schools.com/css/>



External style sheet

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Internal style sheet

```
<head>
<style>
body {
    background-color: linen;
}

h1 {
    color: maroon;
    margin-left: 40px;
}
</style>
</head>
```



Inline style

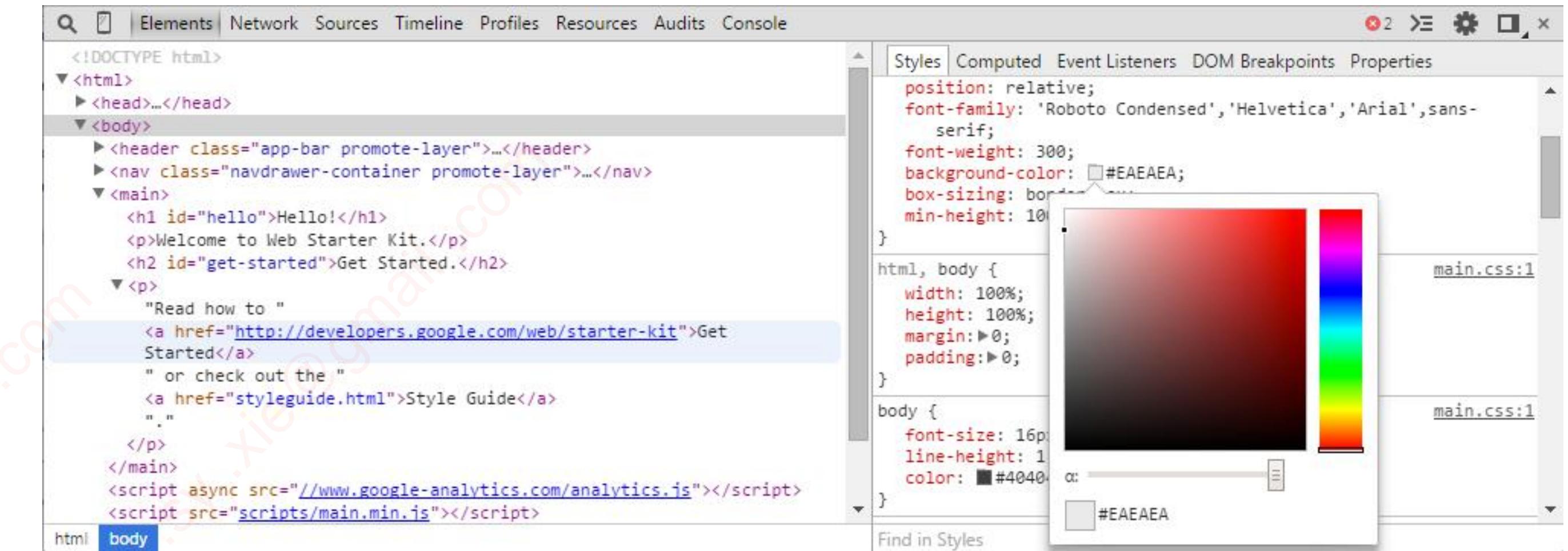
```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```

NOT recommend

Only for components



How to Access DevTools



- Select the **Chrome menu**  at the top-right of your browser window, then select **Tools > Developer Tools**.
- Right-click on any page element and select **Inspect Element**.

npm install live-server -g
live-server



Dev Tool

- 在 Elements 面板中检查和实时编辑 DOM 树中的任何元素。
- 在 Styles 窗格中查看和更改应用到任何选定元素的 CSS 规则。
- 在 Computed 窗格中查看和修改选定元素的框模型。



Style Rules

Web 浏览器将 CSS 规则应用于文档以影响它们的显示方式。一个 CSS 规则由以下组成：

- 一个 **选择器**，它选择元素，这（些）元素是你想应用这些最新的属性值于其上的元素。
- 一组 **属性**，属性的值更新了 HTML 的内容的显示方式。



Style Rules

Selector Declaration Declaration
h1 { color:blue; font-size:12px; }
 ↑ ↑ ↑ ↑
 Property Value Property Value

```
h2{  
  color: black;  
  font-size:24px;  
}  
.bold {  
  font-weight: bold;  
}
```





Selectors

- 简单选择器 (Simple selectors) : 通过元素类型、class 或 id 匹配一个或多个元素。
- 属性选择器 (Attribute selectors) : 通过 属性 / 属性值 匹配一个或多个元素。

Simple Selectors

Element Selector

```
body {  
    background-color:#cccc99;  
}
```

ID Selector

```
#menu {  
    background-color:#ffff00;  
}
```

Class Selector

```
.bookTitle {  
    font-style:italic;  
}
```



Simple Selectors

Universal selector

推荐阅读 [Universal selectors](#)

```
*{  
    color:green;  
}
```

Descendant selector

```
div p {  
    background-color: #3a3a3a;  
}
```

Child selector

```
div > p {  
    background-color: #3a3a3a;  
}
```

```
<div>  
    <form>  
        <p>I'm a descendant</p>  
    </form>  
</div>
```

```
<div>  
    <form>  
        <p>I'm a descendant,  
            but not a child</p>  
    </form>  
    <p>I'm a child</p>  
</div>
```



Simple Selectors

Attribute selector

推荐阅读 [Universal selectors](#)

```
img[alt=jiangren]{  
    padding:10px;  
}
```


Psuedo selector

```
a:hover {  
    color: #7E6432;  
}
```

Selectors

- 伪类 (Pseudo-classes) : 匹配处于确定状态的一个或多个元素, 比如被鼠标指针悬停的元素, 或当前被选中或未选中的复选框, 或元素是DOM树中一父节点的第一个子节点。

```
a:hover {}  
a:focus {}  
div:first-child {}  
div:nth-child(2) {}  
div:last-child {}
```

- **child** 和 **type** 的区别



Psuedo selector

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
    color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}  
  
/* selected link */  
a:active {  
    color: #0000FF;  
}
```



Selectors

- 伪元素 (Pseudo-elements) : 匹配处于相关的确定位置的一个或多个元素，例如每个段落的第一个字，或者某个元素之前生成的内容。

```
div::after {}  
div::before {}
```

Add emoji "\01F604", "\01F620"

CSS Box modal

This is a heading (for testing purposes)

Paragraphs are the bread and butter of HTML.

They can include **strong** tags.

1. Lists are cool!
2. The list things.
3. This one has an **i** inside.
4. This one contains a :

- o Item 1
- o Marked item
- o Final item

5. What happens with an h2?

6. Final order

A final paragraph with a dummy [link](#) inside it.



Selectors

- 组合器（Combinators）：这里不仅仅是选择器本身，还有以有效的方式组合两个或更多的选择器用于非常特定的选择的方法。例如，你可以只选择

的直系子节点的段落，或者直接跟在headings后面段落。
- 多重选择器（Multiple selectors）：这些也不是单独的选择器；这个思路是将以逗号分隔开的多个选择器放在一个CSS规则下面，以将一组声明应用于由这些选择器选择的所有元素。



Units

- 像素 (px) 是一种绝对单位 (absolute units) , 因为无论其他相关的设置怎么变化, 像素指定的值是不会变化的。其他的绝对单位如下:
- mm, cm, in
- pt, pc: 点 (Points (1/72 of an inch)) , 十二点活字 (picas (12 points.))



- Screen measurements
 - pixels(px)
- Relative measurements
 - %,em, pt
 - kyleschaeffer.com/development/css-font-size-em-vs-px-vs-pt-vs/
- Color
 - #rrggbb , rgb(r,g,b), rgba(r,g,b,a)
- Fonts
 - sans-serif, google fonts(<https://fonts.googleapis.com/>)
- Functional notation
 - **background:**
[url\(http://www.example.org/image\)](http://www.example.org/image);



Units

- vw, vh
- %
- 0
- auto



Units

em

- 1em与当前元素的字体大小相同（更具体地说，一个大写字母M的宽度）。CSS样式被应用之前，浏览器给网页设置的默认基础字体大小是16像素，这意味着对一个元素来说1em的计算值默认为16像素。但是要小心—em单位是会继承父元素的字体大小，所以如果在父元素上设置了不同的字体大小，em的像素值就会变得复杂。



Units

rem

- REM (root em) 和em以同样的方式工作，但它总是等于默认基础字体大小的尺寸；继承的字体大小将不起作用，所以这听起来像一个比em更好的选择



Well used properties

- width and height (min- and max-)
- margin and padding
- font
- border
- background and color
- text-align
- display



Less used properties

- visibility
- text-shadow
- box-shadow
- outline
- cursor
- text-decoration
- transform
- overflow



Other properties

- word-wrap
- word-break
- content



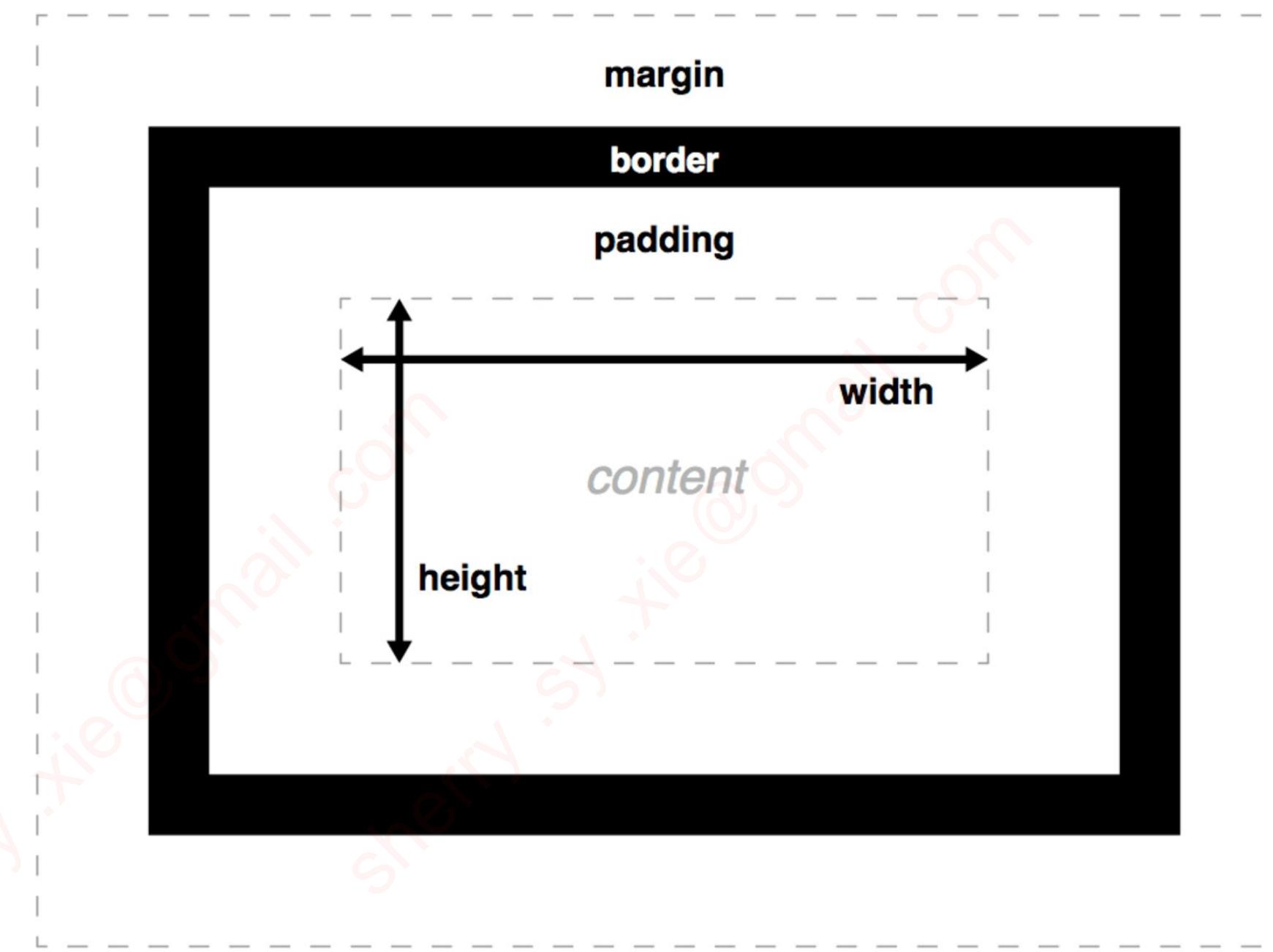
Normalize CSS

Normalize.css

makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.

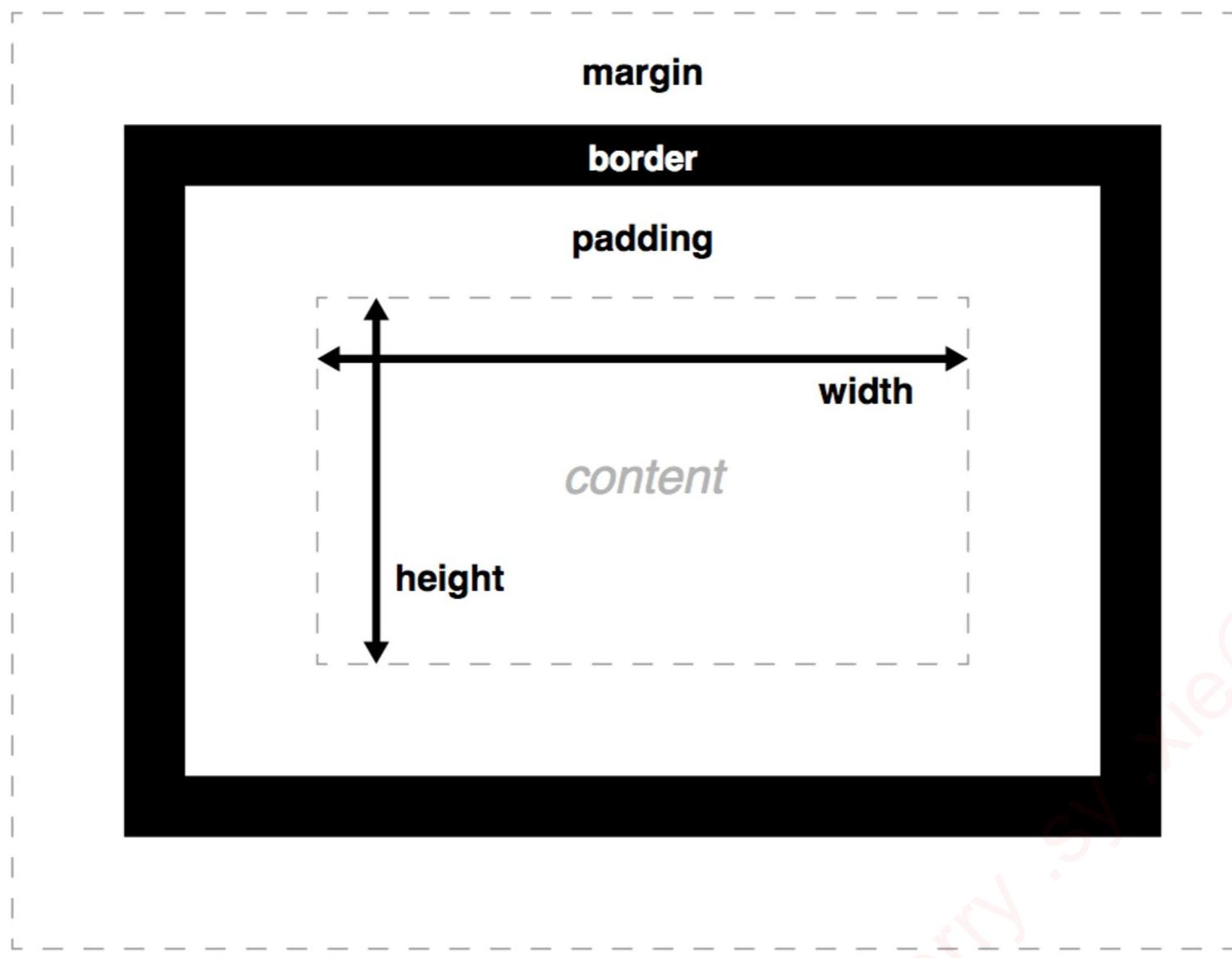


Box Model



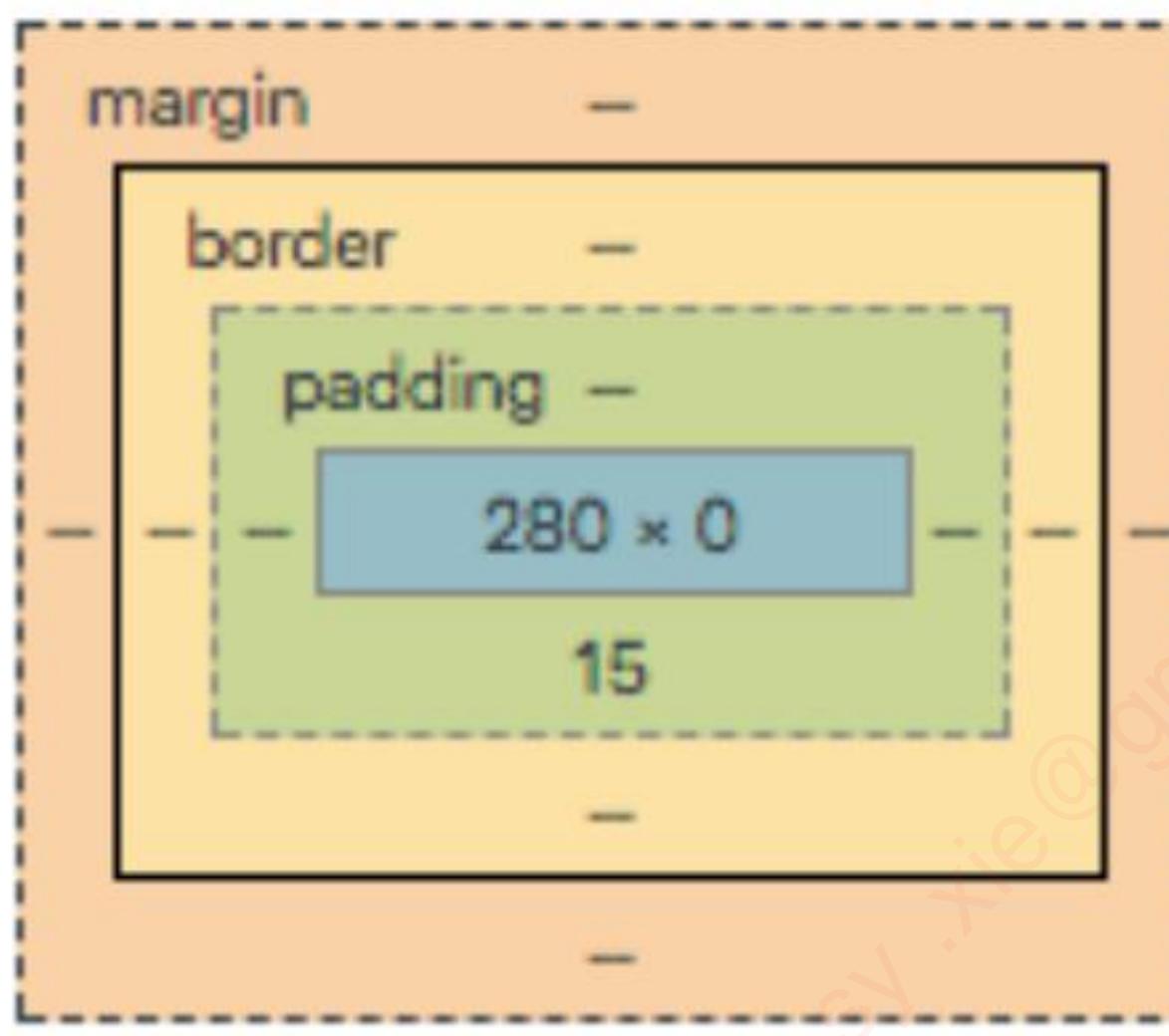
The CSS box model is the foundation of layout on the Web — each element is represented as a rectangular box, with the box's content, padding, border, and margin

Box Model



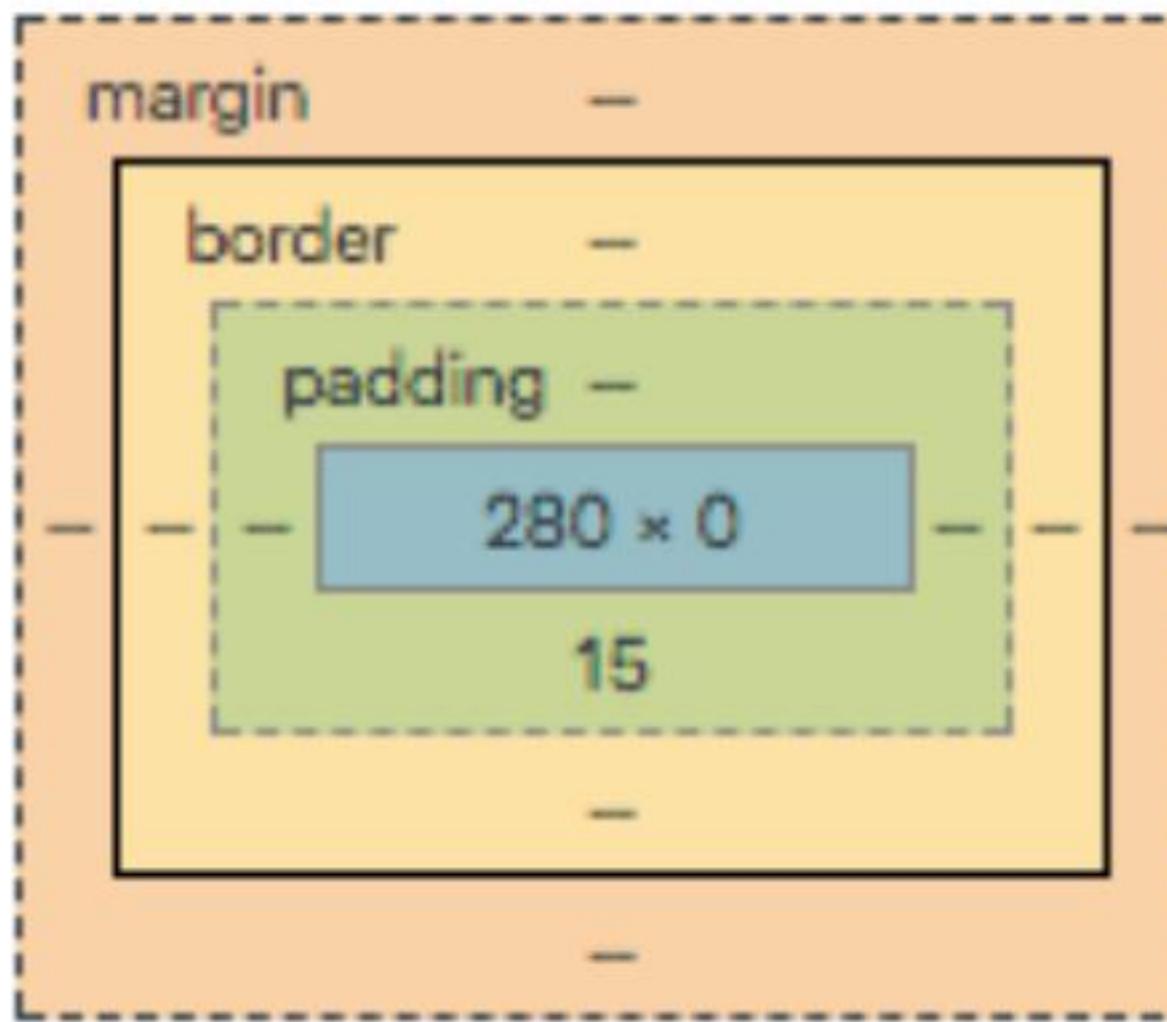
- Content - The content of the box, where text and images appear
- Padding - Clears an area around the content. The padding is transparent
- Border - A border that goes around the padding and content
- Margin - Clears an area outside the border. The margin is transparent

Margin, Border, Padding



- margin
 - margin-top
 - margin-bottom
 - margin-right
 - margin-left
- border
 - border-style
 - border-width
 - border-color
- padding
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left

Margin



- **margin: 25px 50px 75px 100px;**
 - top margin is 25px
 - right margin is 50px
 - bottom margin is 75px
 - left margin is 100px
- **margin: 25px 50px 75px;**
 - top margin is 25px
 - right and left margins are 50px
 - bottom margin is 75px
- **margin: 25px 50px;**
 - top and bottom margins are 25px
 - right and left margins are 50px
- **margin: 25px;**
 - all four margins are 25px



Style a button

```
button {  
    padding: 0;  
    outline: 0;  
    border: 0;  
    background: transparent;  
}
```



Padding

```
.box {  
    padding: <padding-top> || <padding-right> || <padding-bottom> || <padding-left>  
}
```

padding:10px 5px 15px 20px;

- top padding is 10px
- right padding is 5px
- bottom padding is 15px
- left padding is 20px

padding:10px 5px;

- top and bottom padding are 10px
- right and left padding are 5px

padding:10px 5px 15px;

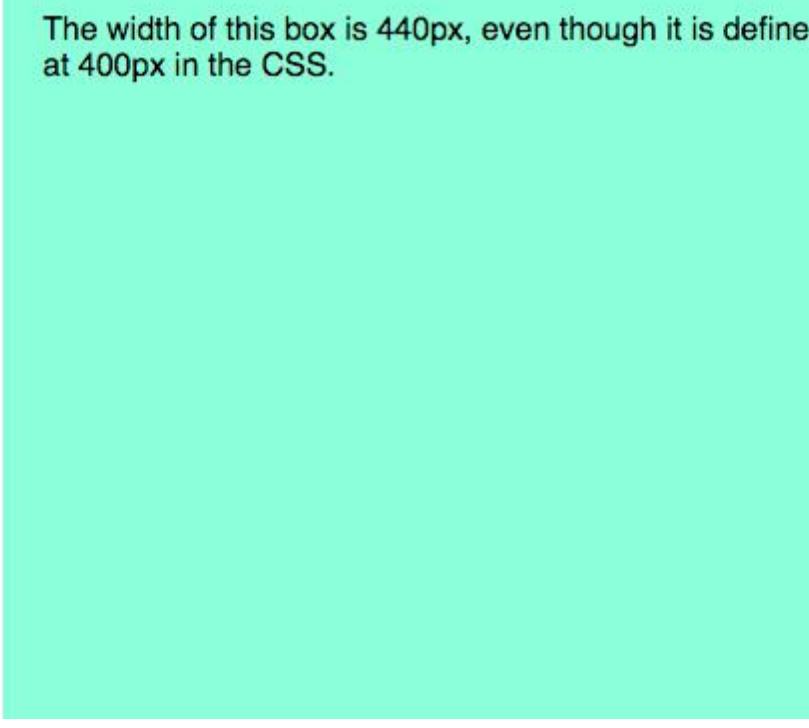
- top padding is 10px
- right and left padding are 5px
- bottom padding is 15px

padding:10px;

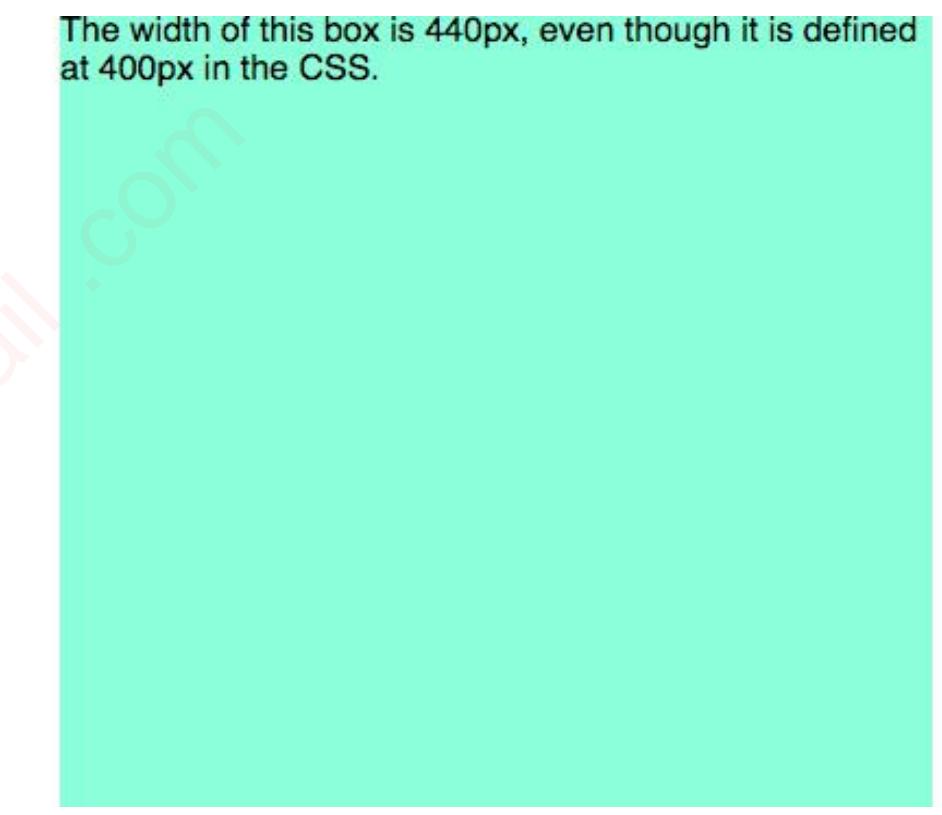
- all four paddings are 10px

Padding

The width of this box is 440px, even though it is defined at 400px in the CSS.



The width of this box is 440px, even though it is defined at 400px in the CSS.



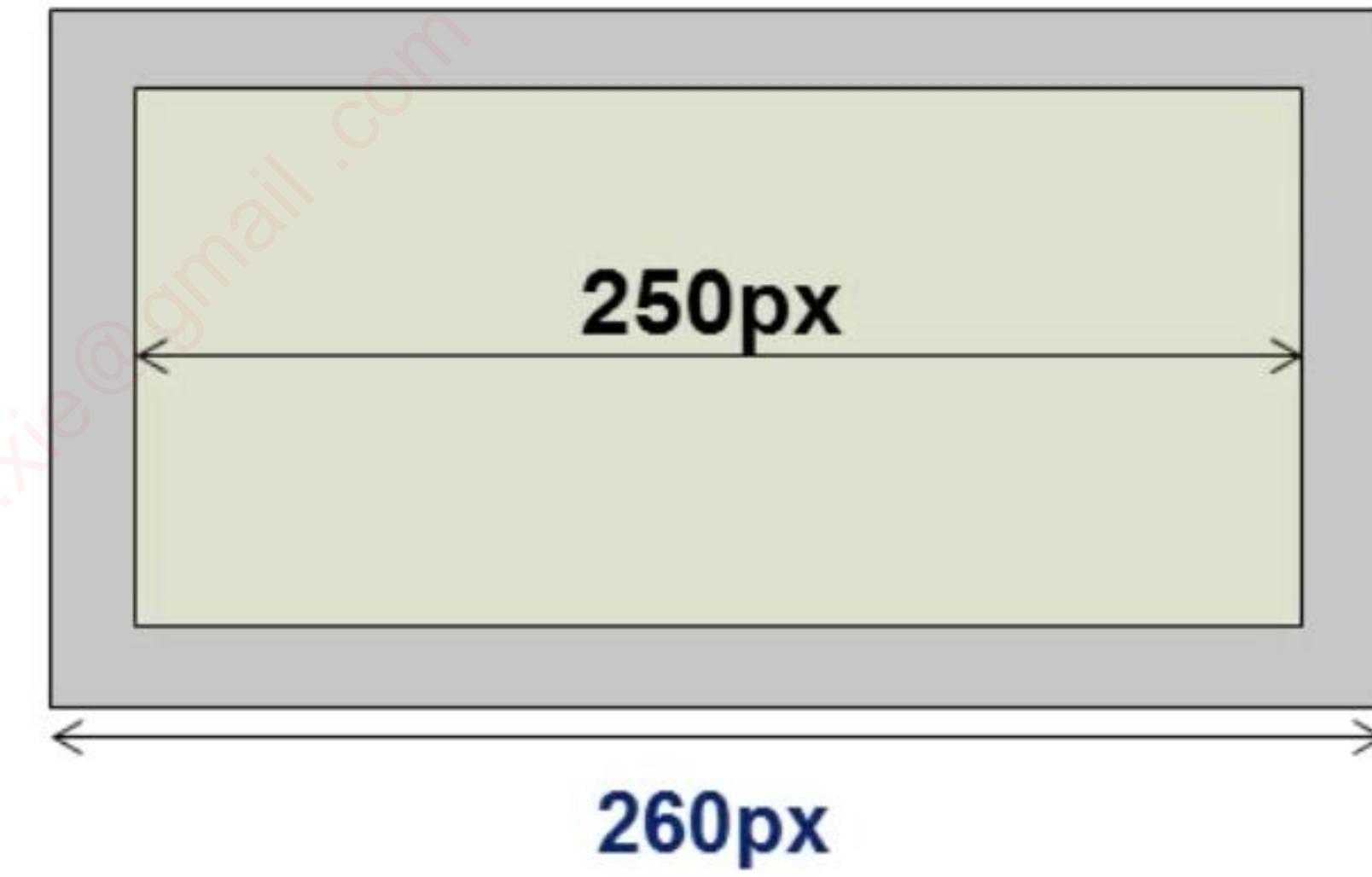
太丑

```
1. .box {  
2.   margin: 0 auto;  
3.   background-color: aquamarine;  
4.   width: 400px;  
5.   height: 400px;  
6.   padding: 20px;  
7. }
```

```
1. .box {  
2.   margin: 0 auto;  
3.   background-color: aquamarine;  
4.   width: 400px;  
5.   height: 400px;  
6.   padding: 0px;  
7. }
```

Box Width

```
ul li  
{  
    width: 250px;  
}
```



- Margin, padding, and border are additive



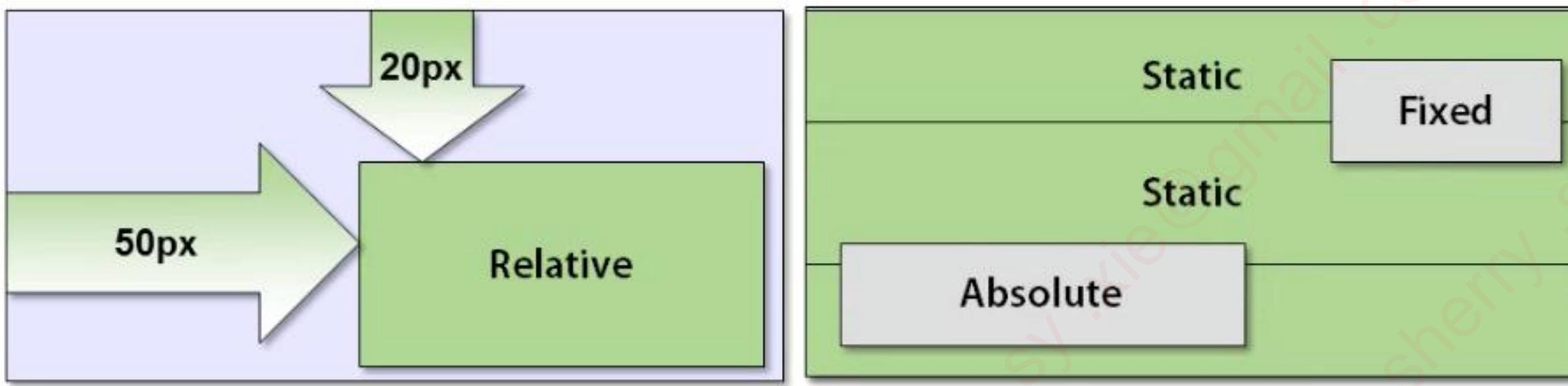
Display and visibility

- Display is generally block , inline , or none
 - block elements sit on top of each other
 - Inline elements only move downward when there is not enough space
 - Display of none removes an element
- Visibility
 - Hidden elements are not visible but reserve space



Position

```
#links  
{  
    position: relative;  
    top: 20px;  
    left: 50px;  
}
```

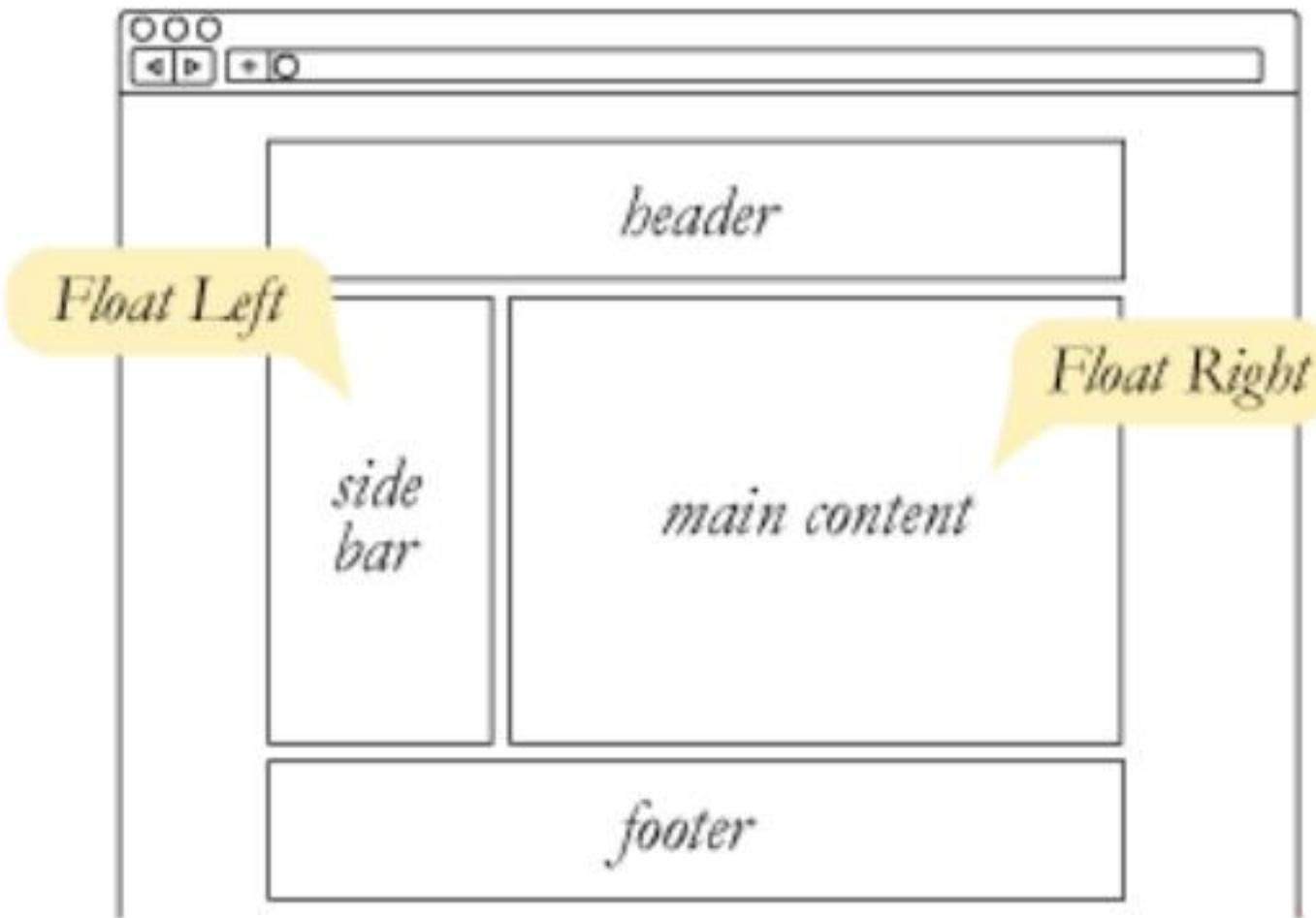


- **position: absolute**
 - 绝对定位的元素完全脱离。对于包围它的元素而言，视为不存在。可以通过top, right, bottom, left 控制定位
- **position: relative**
 - 相对定位
- **position: fixed**
 - 绝对定位，相对于浏览器窗口来定位
- **position: static (默认)**
 - 一个 static 定位的元素会忽略所有 top, right, bottom, left 以及 z-index 属性所声明的值
- **position: inherit (继承)**

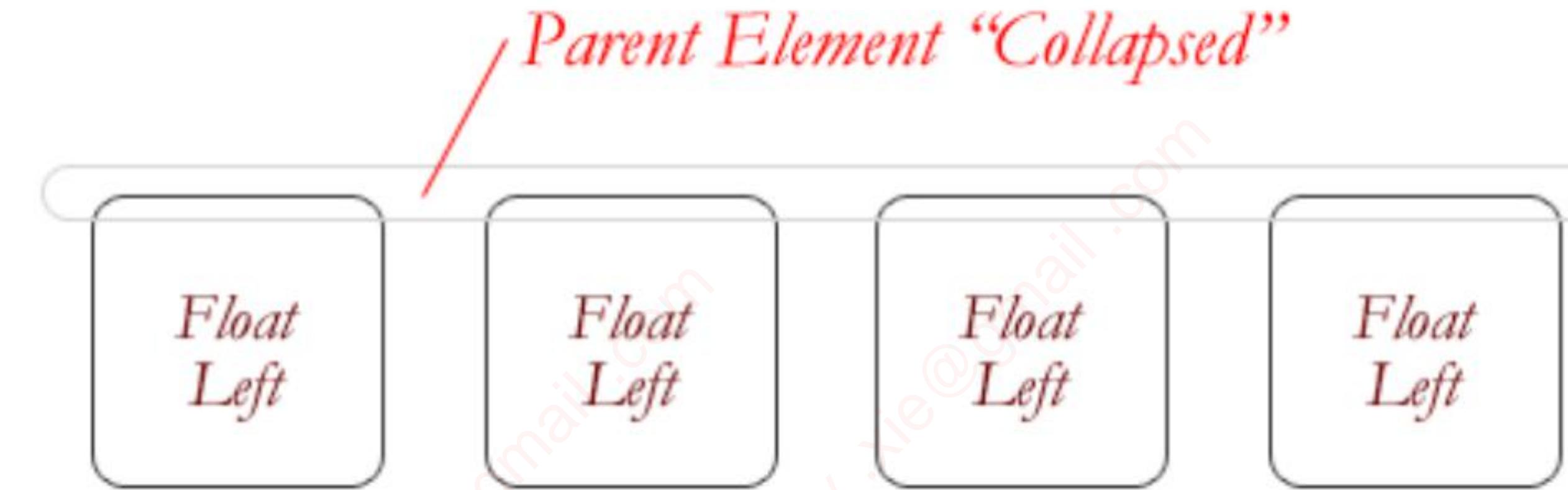
推荐阅读 [How Well Do You Understand CSS Positioning?](#)

Float and Clear

- float: right
- float: left
- float: noe
- float: clear



Float and Clear



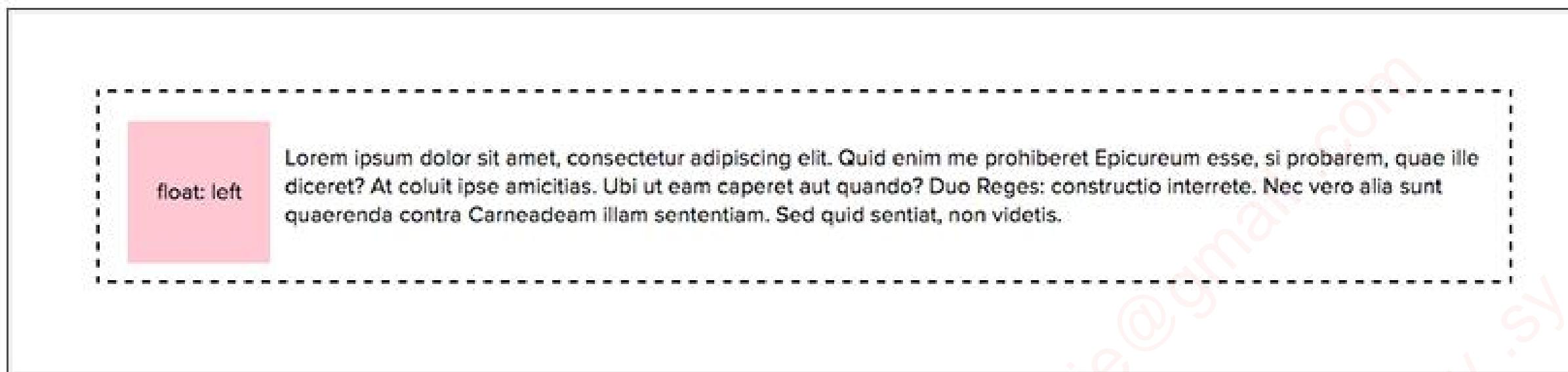
浮动元素会脱离文档流，不会继续停留在其父元素内。如果一个父元素只有一个子元素，那么它将会塌陷，就像是空的一样。就表现而言，就有点类似于子元素做了绝对定位。



Float and Clear

p 在 float 之后

p 在 float 之前



Block, Inline, Inline-block

This is a heading (for testing purposes)

Paragraphs are the bread and butter of HTML.

They can include **strong** tags.

1. Lists are cool!
2. The list things.
3. This one has an **i** inside.
4. This one contains a :

- o Item 1
- o Marked item
- o Final item

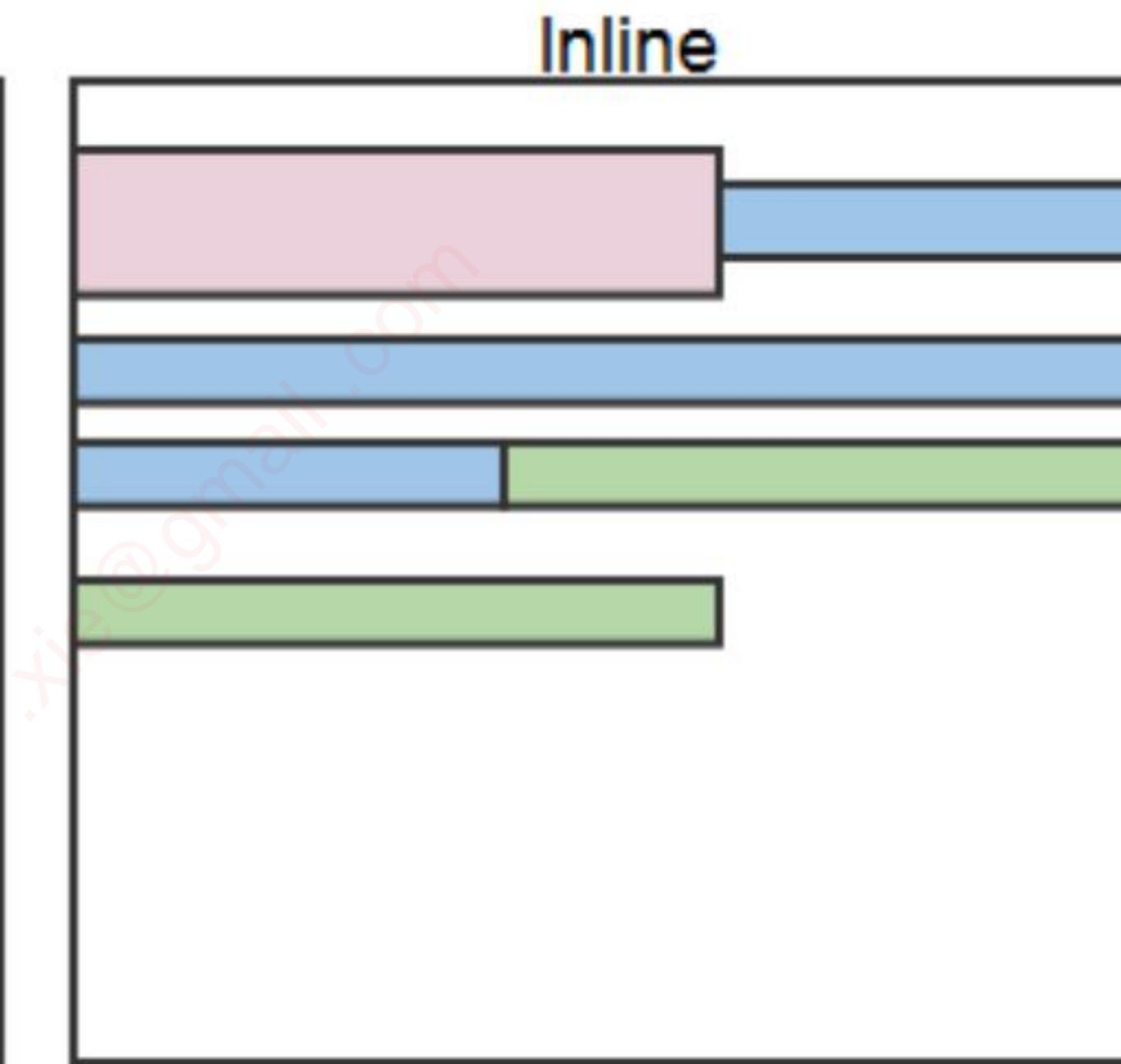
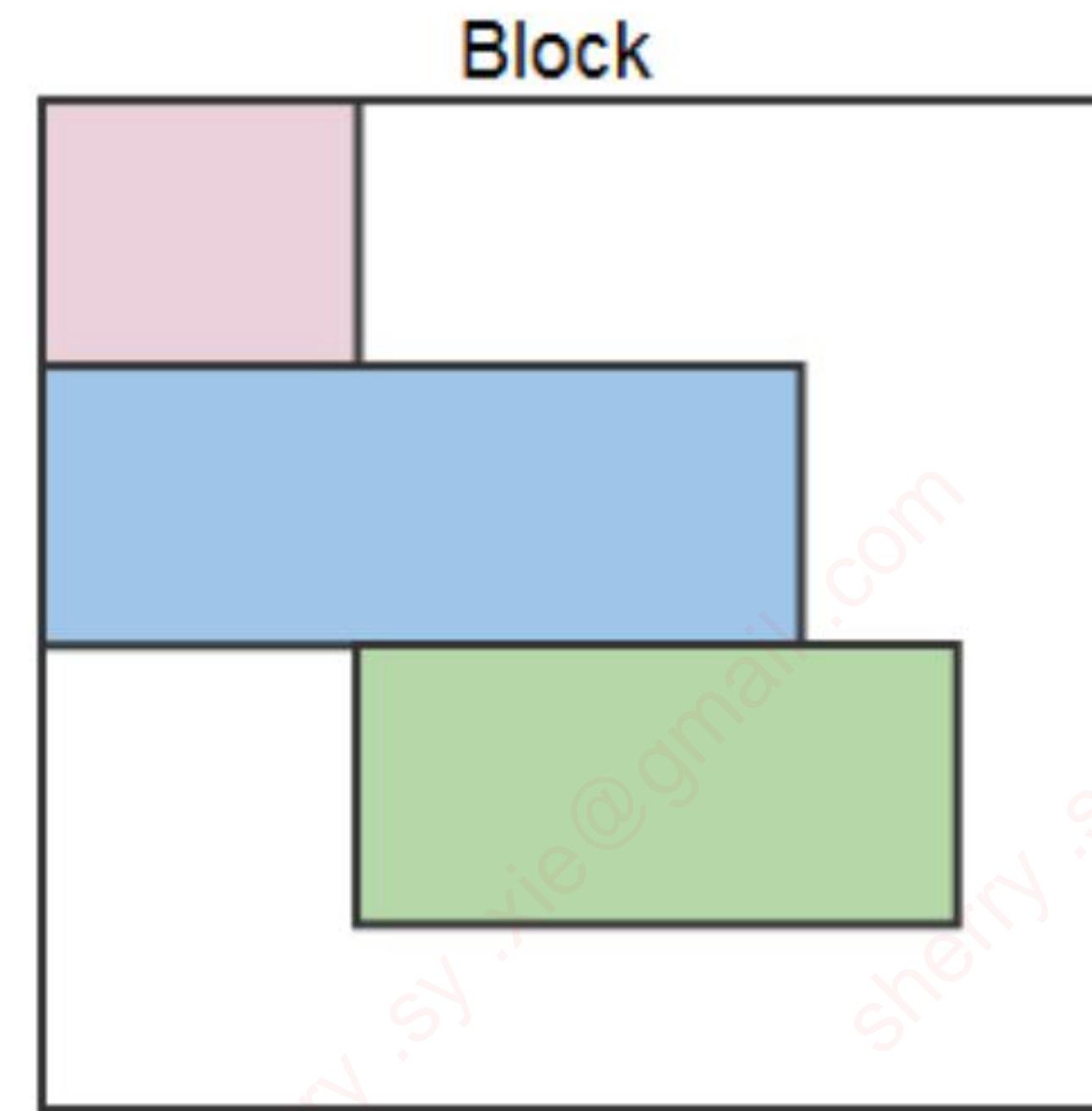
5. What happens with an h2?

6. Final order

A final paragraph with a dummy [link](#) inside it.



Block, Inline, Inline-block



height, width, padding, margin

Position

This is a heading (for testing purposes)

Paragraphs are the bread and butter of HTML.

They can include **strong** tags.

1. Lists are cool!
2. The list things.
3. This one has an **i** inside.
4. This one contains a :

- o Item 1
- o Marked item
- o Final item

5. What happens with an h2?

6. Final order

A final paragraph with a dummy [link](#) inside it.



Position

- **static**, 该关键字指定元素使用正常的布局行为，即元素在文档常规流中当前的布局位置。**此时 top, right, bottom, left 和 z-index 属性无效。**
- **relative**, 该关键字下，元素先放置在未添加定位时的位置，再在不改变页面布局的前提下调整元素位置（因此会在此元素未添加定位时所在位置留下空白）



Position

- absolute, 不为元素预留空间，通过指定元素相对于最近的非 static 定位祖先元素的偏移，来确定元素位置。绝对定位的元素可以设置外边距 (margins)，且不会与其他边距合并。
- fixed, 不为元素预留空间，而是通过指定元素相对于屏幕视口 (viewport) 的位置来指定元素位置。元素的位置在屏幕滚动时不会改变。

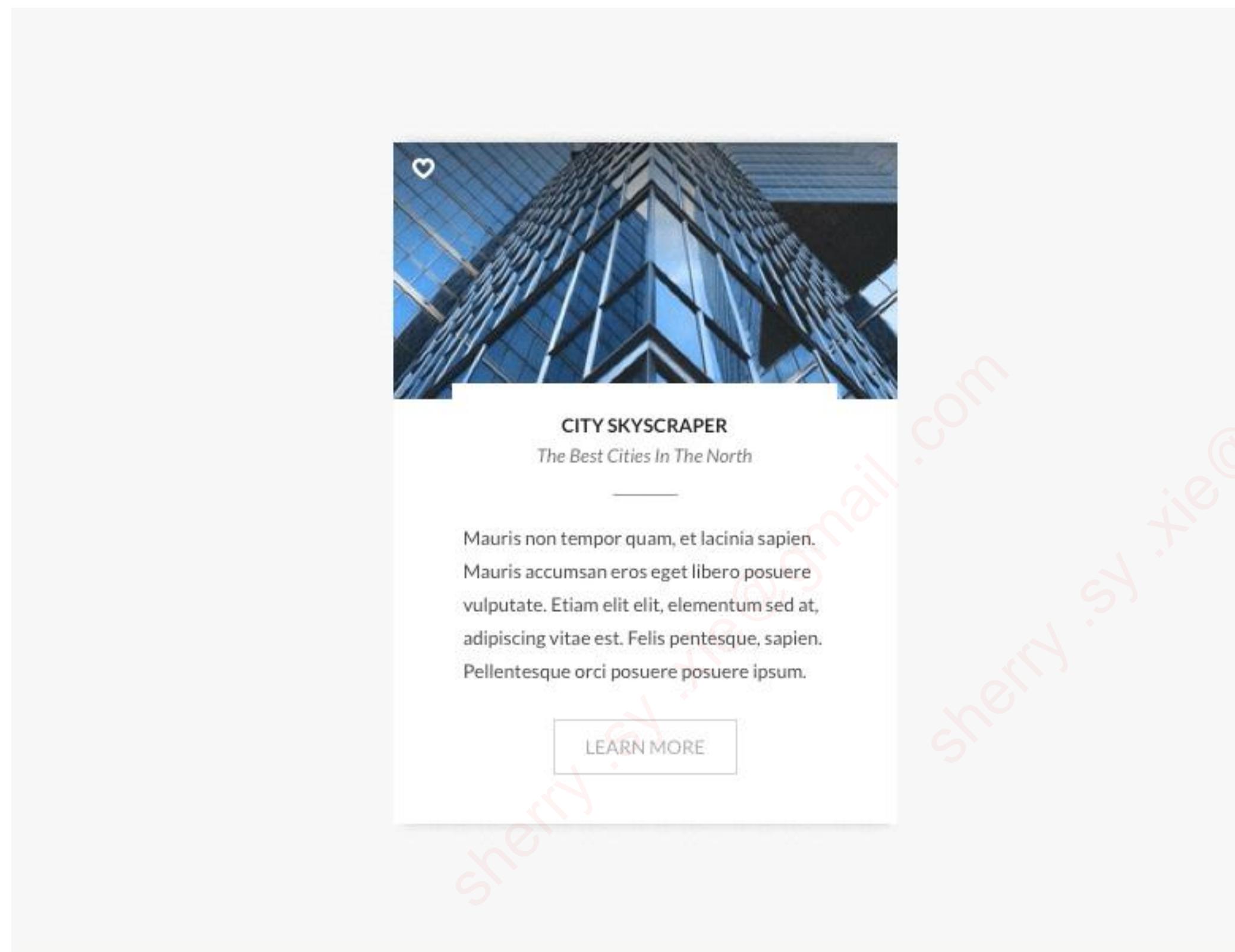


Position

demo

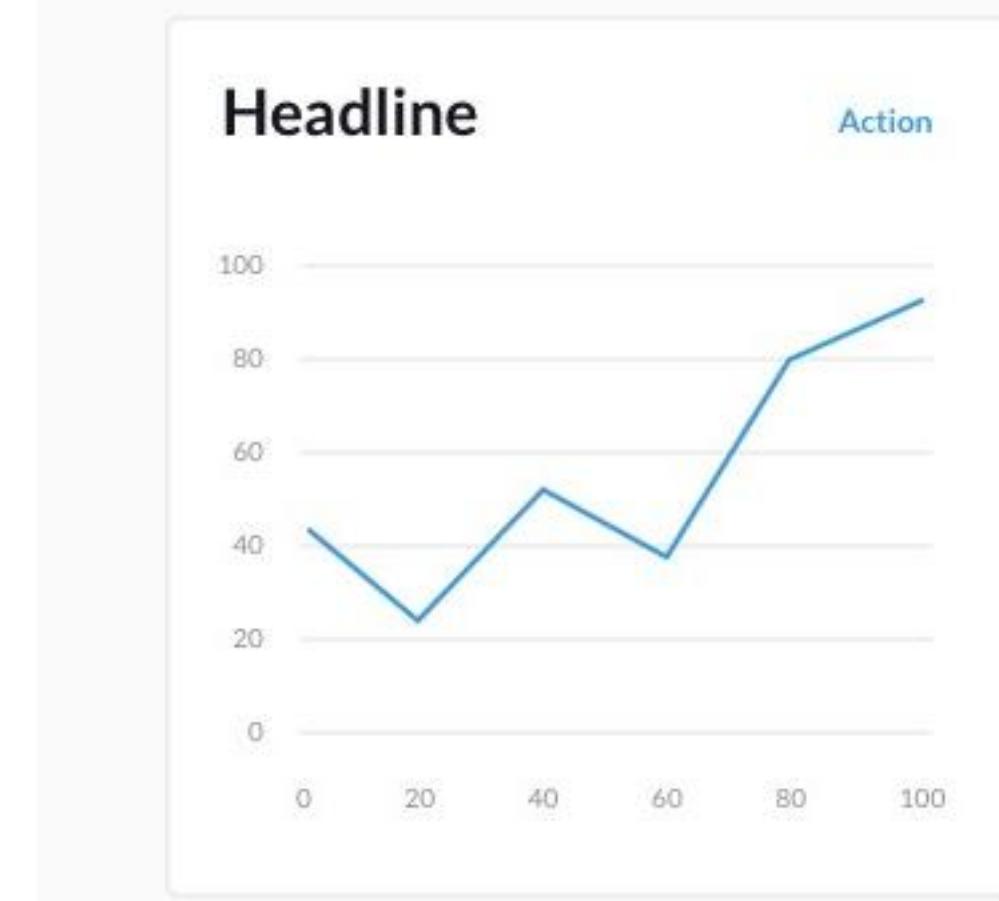
Position

- top
- bottom
- right
- left
- z-index



A dark vertical card template. At the top is a small white icon labeled "MEDIA". Below it is the word "Headline" in bold, followed by "Sub-headline" in a smaller font. Then there is a block of placeholder text: "Text non tempor quam, et lacinia sapien. Mauris accumsan eros eget libero posuere vulputate. Etiam elit elit, elementum sed at.". At the bottom are two light gray rectangular buttons, each with the word "Action" in a small font.

A dark vertical card template. At the top is a small white icon labeled "MEDIA". Below it is the word "Headline" in bold, followed by "Sub-headline" in a smaller font. Then there is a block of placeholder text: "Text non tempor quam, et lacinia sapien. Mauris accumsan eros eget libero posuere vulputate. Etiam elit elit, elementum sed at.". At the bottom are two light gray rectangular buttons, each with the word "Action" in a small font.





Styling from top to bottom

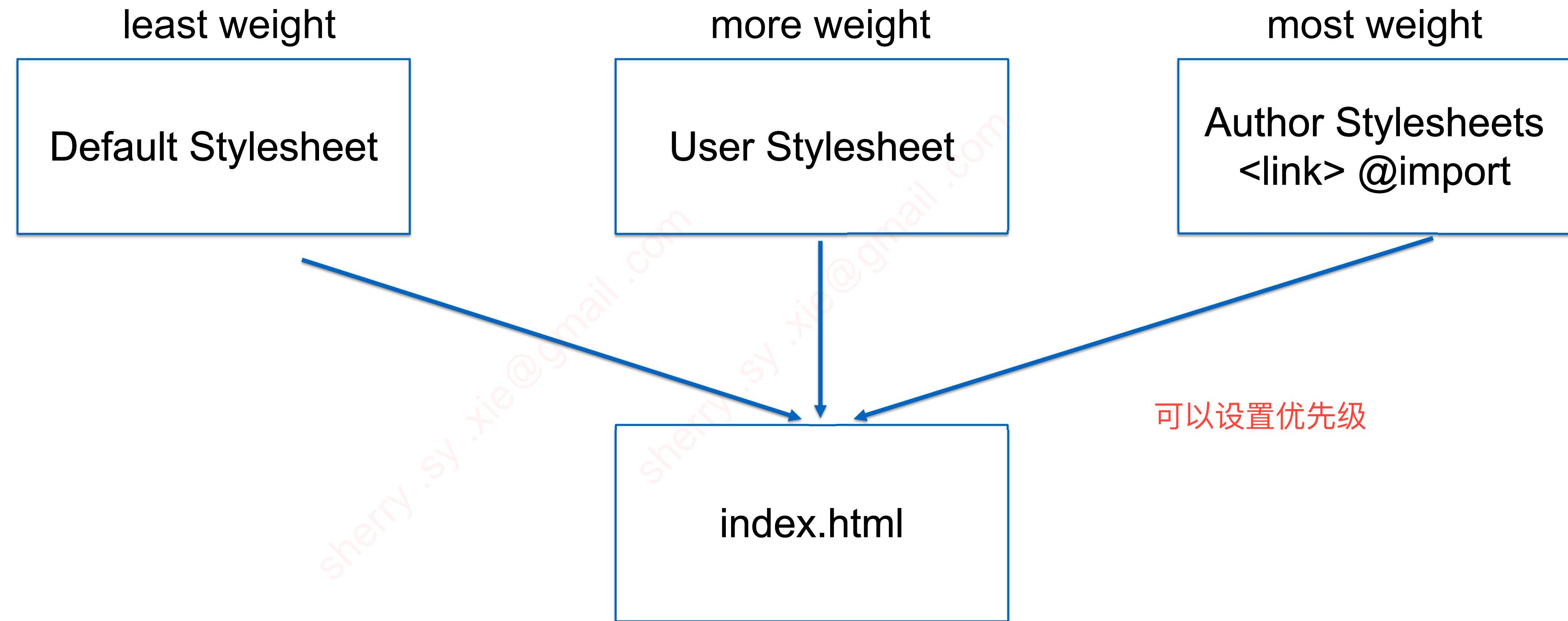
```
<div class="container">
  <div class="big">Big width</div>
  <div class="small">Small width</div>
</div>

.container {
  font-size: 16px;
}
.container > .big {
  width: 80%;
}

.container > .small {
  width: 20%;
}
```



Styles From Different Sources



<https://stackoverflow.com/questions/18252356/what-is-the-difference-between-default-user-and-author-style-sheets>



Why use css reset?

- 不同浏览器对element不一样

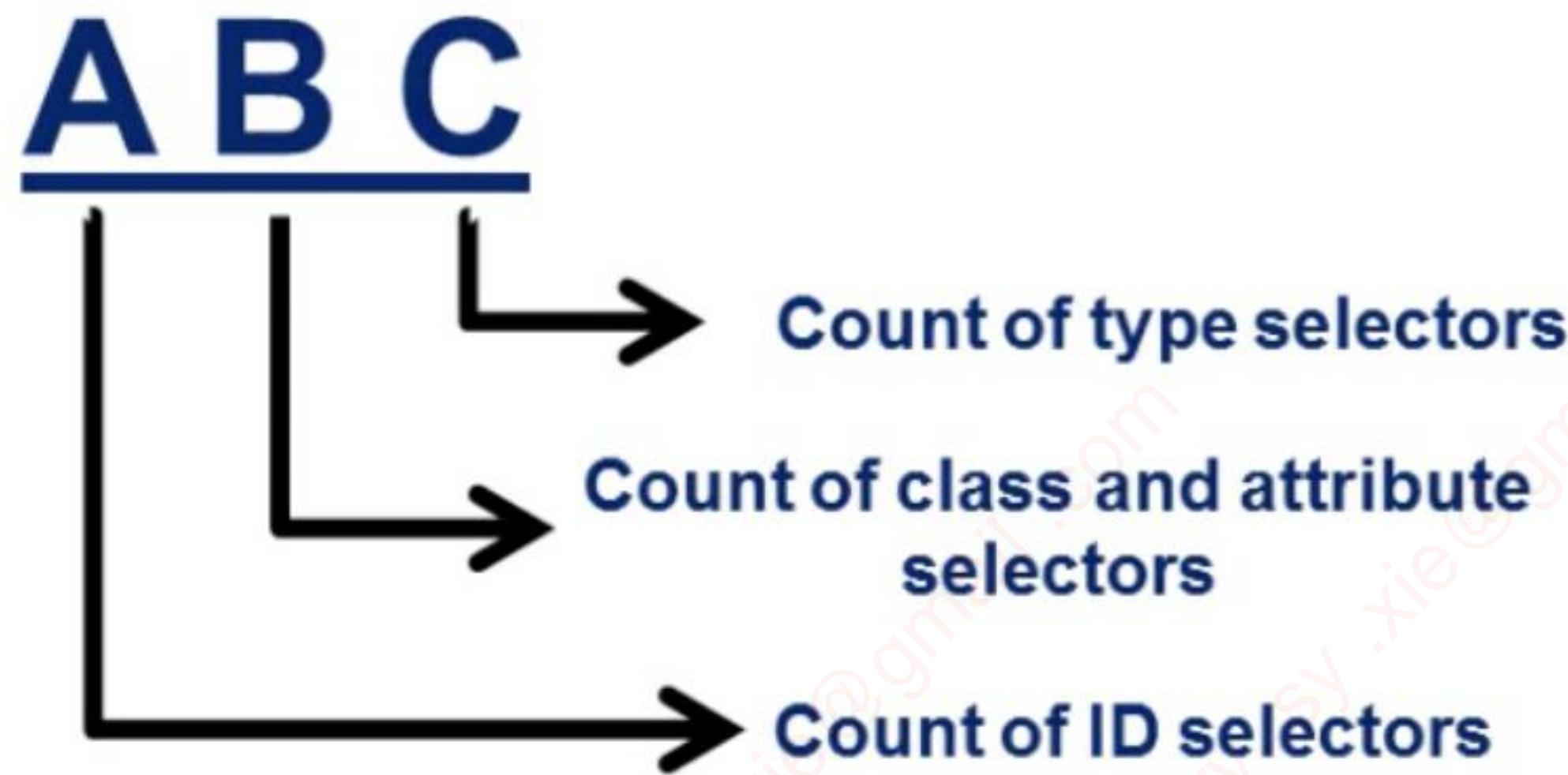
CSS Reset

- 面试点
- Eric Meyer
 - <https://meyerweb.com/eric/tools/css/reset/>

都设置成0，让自己去设定
不用universal selector是因为太危险了



Specificity

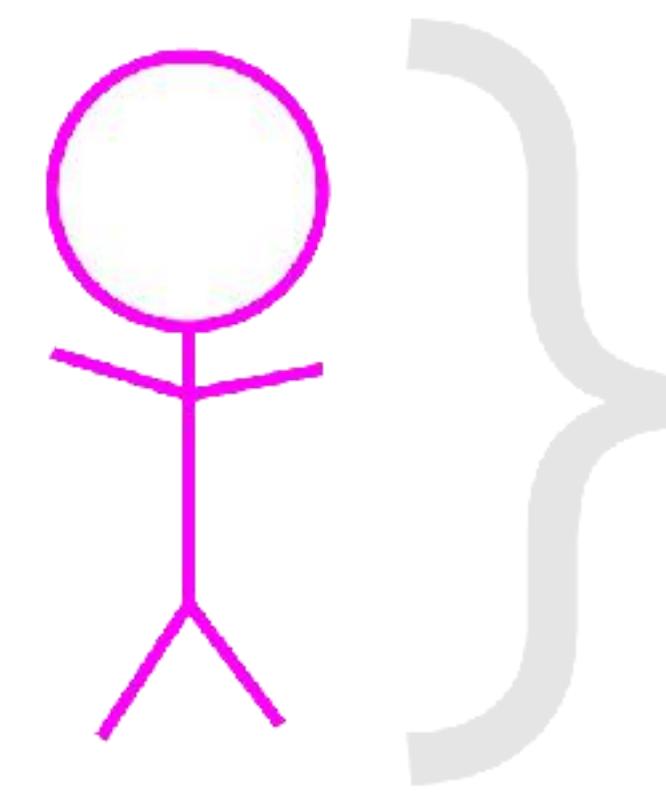


原则

越详细，优先级越高

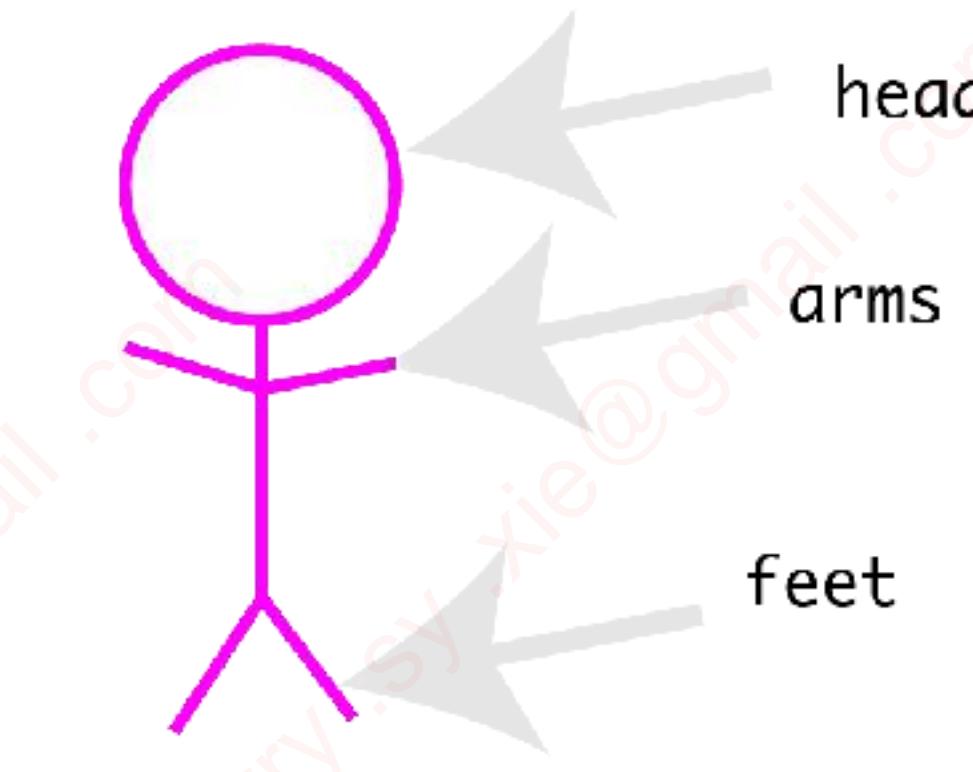
```
*                  /* a=0 b=0 c=0 -> specificity = 0 */
LI                 /* a=0 b=0 c=1 -> specificity = 1 */
UL LI              /* a=0 b=0 c=2 -> specificity = 2 */
LI.red             /* a=0 b=1 c=1 -> specificity = 11 */
#content          /* a=1 b=0 c=0 -> specificity = 100 */
```

BEM



.stick-man

B for 'Block'



E for Elements



blue stick man



red stick man

M for Modifiers

BEM



B for 'Block'

```
.stick-man {  
}
```

E for Elements

```
.stick-man__head {  
}  
.stick-man__arms {  
}  
.stick-man__feet {  
}
```

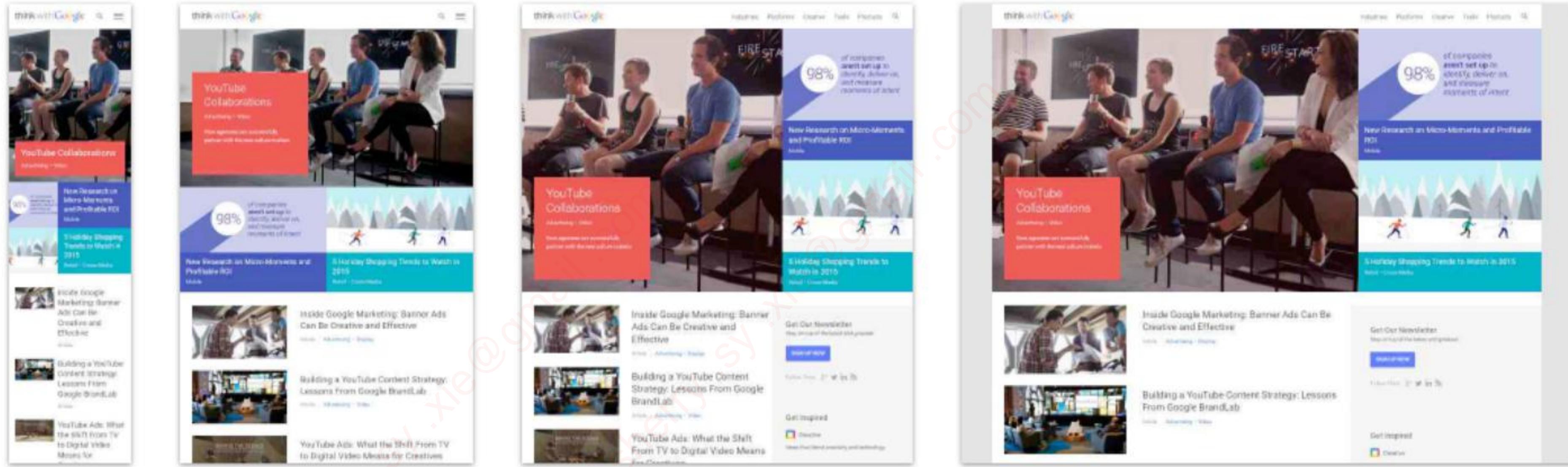
M for Modifiers

```
.stick-man--blue {  
}  
.stick-man--red {  
}
```



To adjust according to the viewpoint e.g view from laptop/pad/phone to make it user friendly

Responsive Web Design



[https://developers.google.com/web/fundamentals
als/design-and-
ui/responsive/videos/resize.webm](https://developers.google.com/web/fundamentals/als/design-and-ui/responsive/videos/resize.webm)



Responsive Web Design

px not mean 你屏幕的分辨率
set =device-width, can control the width

Set the viewport

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- Use the meta viewport tag to control the width and scaling of the browser's viewport.
- Include width=device-width to match the screen's width in device-independent pixels.
- Include initial-scale=1 to establish a 1:1 relationship between CSS pixels and device-independent pixels.
- Ensure your page is accessible by not disabling user scaling.



Responsive Web Design

Flexible Images

- img {
- max-width:100%;
- }

```
/* iPhone and Smartphones (portrait and landscape) */
@media screen and (min-device-width : 320px) and (max-device-width: 480px) {
    //你的样式放在这里...
}
```

CSS3 Media Queries

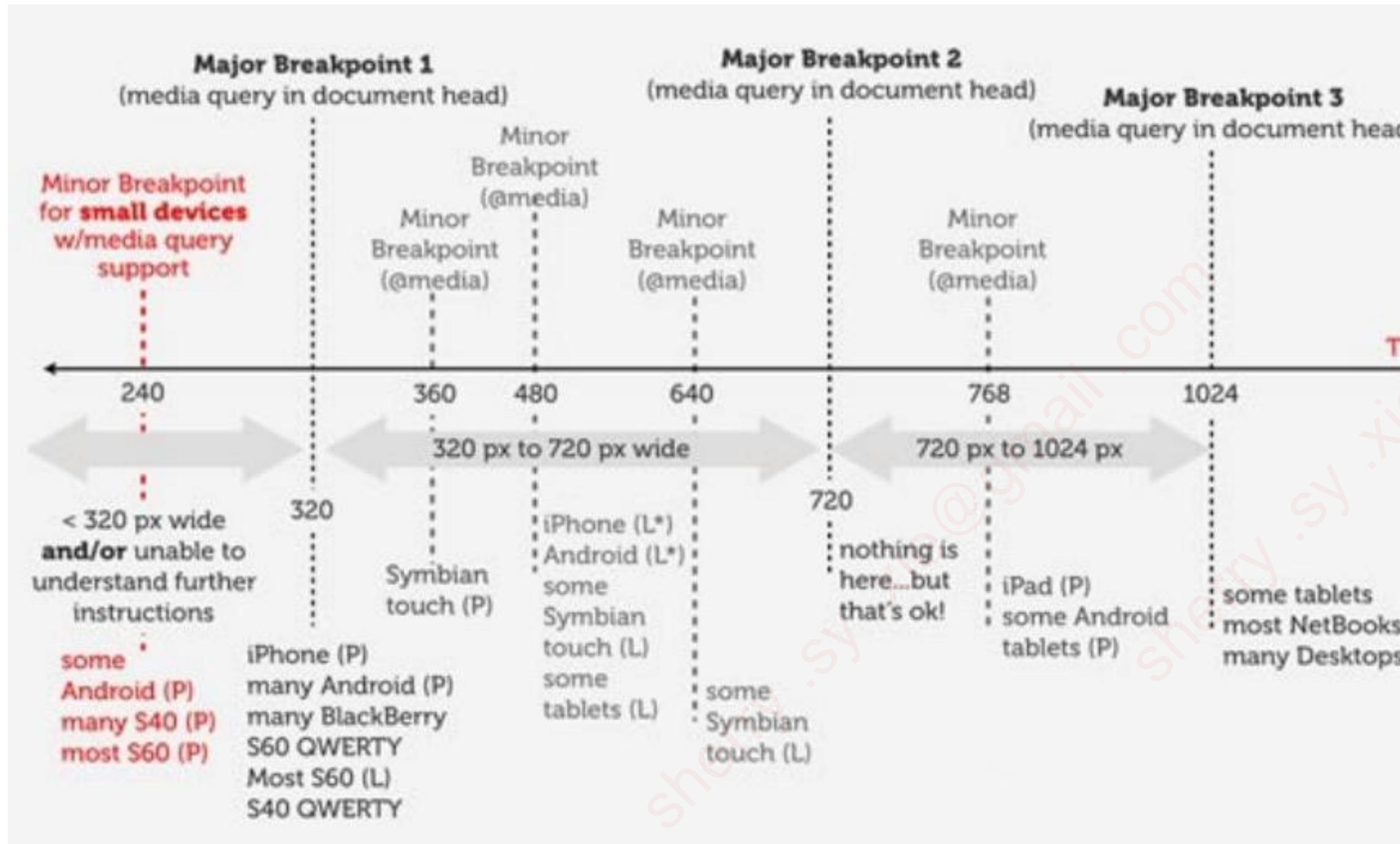
- @media screen and (max-width: 600px) {
• //你的样式放在这里....
• }

when $\leq 600\text{px}$

- @media screen and (min-width: 900px) {
• //你的样式放在这里....
• }

when $\geq 900\text{px}$

Responsive Web Design



```
/*===== Mobile First Method =====*/
/* Custom, iPhone Retina */
@media only screen and (min-width : 320px) {

}

/* Extra Small Devices, Phones */
@media only screen and (min-width : 480px) {

}

/* Small Devices, Tablets */
@media only screen and (min-width : 768px) {

}

/* Medium Devices, Desktops */
@media only screen and (min-width : 992px) {

}

/* Large Devices, Wide Screens */
@media only screen and (min-width : 1200px) {
}
```



Popular CSS Library



[Bootstrap](#)



[Foundation](#)



[Pure CSS](#)



[Mobi CSS](#)



[Semantic UI](#)

推荐阅读 [The 5 Most Popular Frontend Frameworks of 2017 Compared](#)



Bootstrap

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

grid system

<https://getbootstrap.com/docs/5.0/layout/grid/>
<https://get.foundation/sites/docs/grid.html>

grid 用来整个大页面的布局

flexbox 主要可以用来做里面 part 里的布局
或再不行就用position

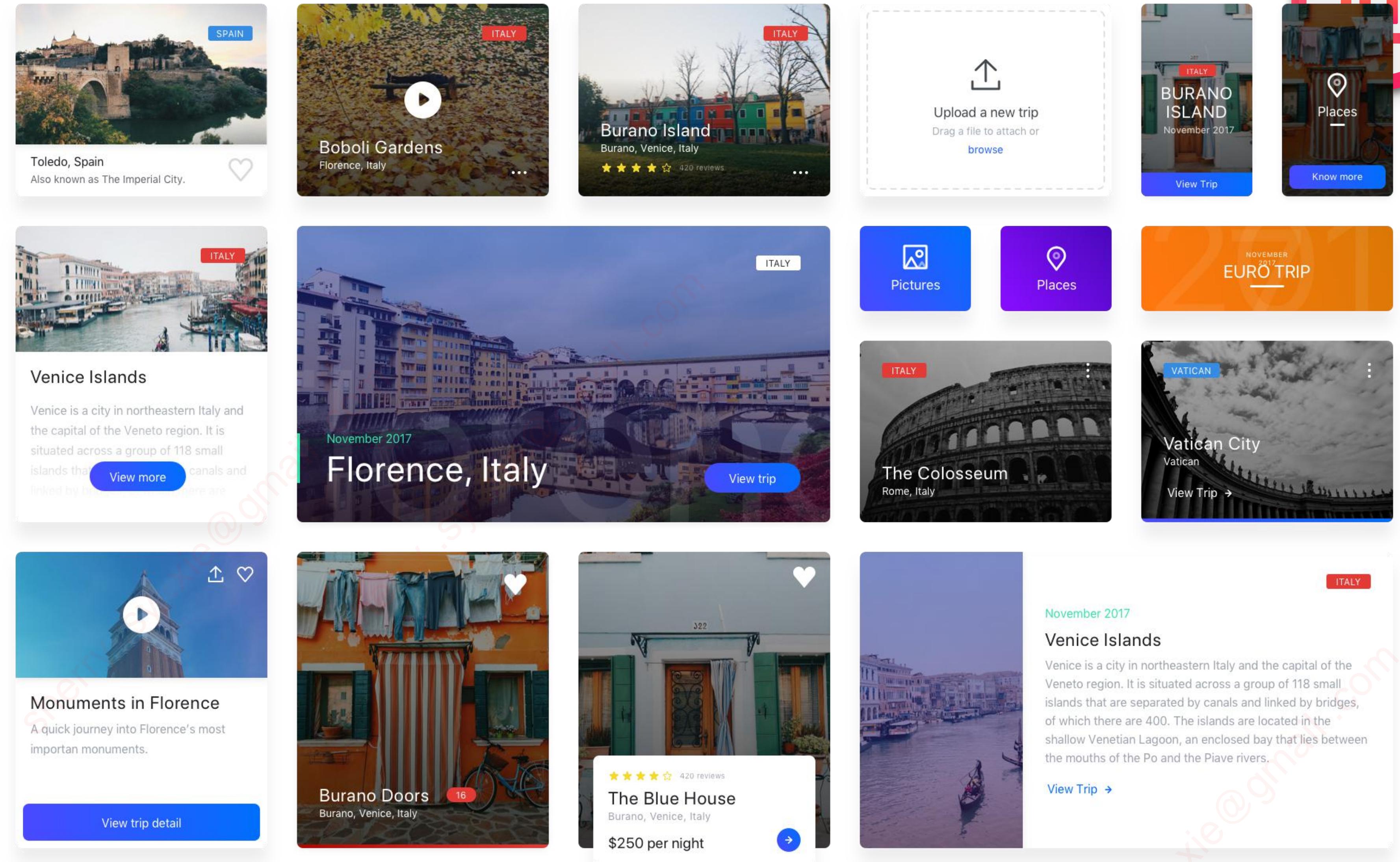
```
<div class="row">
  <div class="col-sm-8">col-sm-8</div>
  <div class="col-sm-4">col-sm-4</div>
</div>
<div class="row">
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
  <div class="col-sm">col-sm</div>
</div>
```

Assignment 1

Card Design

要求
挑选一个card
按照设计完成
使用html, CSS, BEM

Deliver:
上传到LMS





CSS Resources

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

<https://caniuse.com/>



联系方式

1. 微信联系求职顾问



匠人小花



悉尼匠人小嘿



墨尔本匠人小萌



阿德匠人小德

2. 匠人官网报名

Jiangren.com.au



澳洲匠人学院

电话: 0421-672-555

微信: jracademy

邮箱: hello@jiangren.com.au

网址: jiangren.com.au



匠人找工

免费发雇主招聘信息，免费发招聘者求职信息

微信: jiangren_jobs