

# CSS

## CSS基础规则

### CSS 是啥 —— 基础规则

让我们来仔细看一看上述CSS：

整个结构称为 规则集（通常简称“规则”），各部分释义如下：

- 选择器：选择了一个或多个需要添加样式的元素（在这个例子中就是 p 元素）
- 声明：一个单独的规则，如 color: red; 用来指定添加样式元素的属性。
- 属性：指定要改变 HTML 元素样式
- 属性的值：从指定属性的众多外观中选择一个值（除了 red 之外还有很多 color 的值）。

The diagram illustrates a CSS rule structure. It shows the selector 'p' followed by a brace '{' and then the declaration block. The declaration block contains the property 'color' followed by a colon and the value 'red'. Below the declaration block is a bracket labeled '声明' (declaration). To the left of the selector, an arrow points to it with the label '选择器' (selector). To the right of the declaration block, two arrows point to the property name 'color' and its value 'red', both labeled '属性' (property) and '属性的值' (value of the property) respectively.

### CSS 是啥 —— 基础规则

注意其他重要的语法：

如果要同时修改多个属性，只需要将它们用分号隔开

也可以选择多种类型的元素并为它们添加一组相同的样式。将不同的选择器用逗号分开

The diagram shows a block of CSS code: 'p, h2 { color: red; background: white; font-style: italic; }'. This represents multiple selection rules. The first part 'p, h2' is the selector, followed by a brace '{' and then a list of three properties: 'color', 'background', and 'font-style', each with its corresponding values 'red', 'white', and 'italic' respectively. A bracket below the properties is labeled '声明' (declaration).



# CSS 是啥 —— 基础规则

ByteDance  
字节跳动

## 选择器

选择器有许多不同的类型。上面只介绍了元素选择器，用来选择 HTML 文档中给定的元素。但是选择操作可以更加具体。下面是一些常用的选择器类型：

选择器名称	选择的内容	示例
元素选择器（也称作标签或类型选择器）	所有指定(该)类型的 HTML 元素	<code>p</code> 选择 <code>&lt;p&gt;</code>
ID 选择器	具有特定 ID 的元素（单一 HTML 页面中，每个 ID 只对应一个元素，一个元素只对应一个 ID）	<code>#my-id</code> 选择 <code>&lt;p id="my-id"&gt;</code> 或 <code>&lt;a id="my-id"&gt;</code>
类选择器	具有特定类的元素（单一页面中，一个类可以有多个实例）	<code>.my-class</code> 选择 <code>&lt;p class="my-class"&gt;</code> 和 <code>&lt;a class="my-class"&gt;</code>



# CSS 是啥 —— 基础规则

ByteDance  
字节跳动

## 选择器

选择器有许多不同的类型。上面只介绍了元素选择器，用来选择 HTML 文档中给定的元素。但是选择操作可以更加具体。下面是一些常用的选择器类型：

属性选择器	拥有特定属性的元素	<code>img[src]</code> 选择 <code>&lt;img src="myimage.png"&gt;</code> 而不是 <code>&lt;img&gt;</code>
伪 (Pseudo) 类选择器	特定状态下的特定元素（比如鼠标指针悬停）	<code>a:hover</code> 仅在鼠标指针悬停在链接上时选择 <code>&lt;a&gt;</code> 。



## CSS 是啥 —— 基础规则

### 层叠与继承

CSS代表层叠样式表(Cascading Style Sheets), 理解第一个词cascading很重要— cascade 的表现方式是理解CSS的关键。

下面的例子中, 我们有两个关于 h1 的规则。h1 最后显示蓝色 — 这些规则有相同的优先级, 但顺序在最后的生效。

```
h1 {  
    color: red;  
}  
h1 {  
    color: blue;  
}  
  
<h1>This is my heading.</h1>
```

This is my heading.



## CSS 是啥 —— 基础规则

### 层叠与继承

浏览器有个选择器优先级来决定规则, 基本上是一个选择器越具体越优先。

- 一个元素选择器不是很具体 — 会选择页面上该类型的所有元素
- 一个类选择器稍微具体点 — 它会选择该页面中有特定 class 属性值的元素 — 所以它的优先级就要高一点。

下面的 h1 最后会显示红色

```
.main-heading {  
    color: red;  
}  
  
h1 {  
    color: blue;  
}  
  
<h1 class="main-heading">This is my heading.</h1>
```

This is my heading.



# CSS 是啥 —— 基础规则

字节跳动

## 层叠与继承

一些设置在父元素上的 css 属性是可以被子元素继承的，有些则不能。

举一个例子，如果你设置一个元素的 color 和 font-family，每个在里面元素也都会有相同的属性值，除非你直接在元素上设置属性。

```
p {  
    color: blue;  
}  
  
span {  
    color: black;  
}  
  
<p>text in p <span>text in span</span>.</p>
```

text in p text in span.



# CSS 是啥 —— 基础规则

字节跳动

## 层叠与继承

这些概念一起来控制css规则应用于哪个元素；有时候会感觉有些复杂，但是当你对css有更多经验的时候，你就可以记住它们，即便忘记了细节，可以在网上查到。

看一个更复杂的例子。颜色和字重都被继承了；border 就没有（不然每个子元素都有框框得丑），说明浏览器会根据常识判断哪些属性该继承。

```
.main {  
    color: rebeccapurple;  
    font-weight: bold;  
    border: 2px solid #ccc;  
}  
  
.special {  
    color: black;  
}
```

```
<ul class="main">  
    <li>Item One</li>  
    <li>Item Two  
        <ul class="special">  
            <li>2.1</li>  
            <li>2.2</li>  
        </ul>  
    </li>  
</ul>
```

- Item One
- Item Two
  - 2.1
  - 2.2

## CSS怎么学？



# CSS 怎么学——免费课程

字节跳动

在进一步讲解 CSS 有哪些规则之前，先给大家分享几个学习资源。因为 CSS 的内容非常多，也非常适合自学，这里给大家几个课后能不断耕耘 CSS 知识的免费网站。

codecademy.com

老牌在线编程教育网站。  
全程交互式自助式学习  
体验。非常适合 CSS  
边学边练的场景。

The screenshot shows a Codecademy course page for CSS. It features a code editor with HTML and CSS code, step-by-step instructions, hints, and a concept review section.

```
index.html style.css
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <link href="https://fonts.googleapis.com/css?family=Roboto:400,300,500,100" rel="stylesheet" type="text/css">
5 
6 </head>
7 <body>
8   <div class="container">
9     <h2>Innovation Cloud2</h2>
10    <p>Connect your ideas globally</p>
11    <a href="#">Learn More</a>
12  </div>
13 
14  <div class="row">
15    <div class="col-6">
16      <h3>The Innovation Cloud Conference</h3>
17      <p>Connect with the best minds across a wide range of industries to share ideas and brainstorm new solutions to challenging problems.</p>
18      <p>Hear industry leaders talk about what worked (and what didn't) so that you can save time on your most challenging projects.</p>
19      <p>Learn about the latest research and technologies that you can
20        ...
21      </p>
22    </div>
23    <div class="col-6">
24      <h3>Register</h3>
25    </div>
26  </div>
27 
28  <div>
29    <small>Did you copy the <code><div> tag from the instructions and paste it inside the <cloud> tag?</code>
30  </div>
31 
32  <div>
33    <small>Run</small>
34  </div>
35 
36  <div>
37    <small>Concept Review</small>
38  </div>
39 
40  <div>
41    <small>Want to quickly review some of the concepts you've been learning? Take a look at this material's <a href="#">cheatsheet!</small>
42  </div>
```

# CSS 怎么学——免费课程

字节跳动

udacity.com

也是老牌在线编程教育网站。虽然是外文站点，但进入课程内中文是拉满的。推荐理由一是免费；  
二是视频式教学 + 课后实验互动的形式，给喜欢看着老师讲解的同学推荐。

The screenshot shows a Udacity course page for CSS. It features a video player with two instructors, a sidebar with course navigation, and a code editor with a challenge assignment.

**左侧导航栏:**

- 1. 什么是 CSS
- 2. CSS 基本
- 3. 选择器
- 4. 盒模型
- 5. CSS 基本
- 6. 选择
- 7. 相互作用
- 8. 先进的选择器
- 9. 从零开始项目
- 10. 使用 CSS 工具
- 11. 开发工具
- 12. 并行开发与部署上的最佳实践
- 13. 使用 CSS 正则
- 14. CSS 未来

**右侧内容区:**

**视频:** Hey everybody, welcome back to this lesson on style. 大家好 欢迎来一起学习“样式”

**代码编辑器:**

```
Hello, world!
Are you ready for your first challenge?
Let's add some style to this webpage!
For this quiz, I want you to change the color of Hello, World to green.
这个任务里 我要你把 Hello, World 变成绿色。
任务
使用 CSS 选择器将 "Hello, world!" 的颜色设置为绿色。
```

**CSS 怎么学 —— 免费课程**

w3schools.com & runoob.com

一个英文一个中文。免费，交互式，内容相当全面，不错的实操演练场。

The screenshot shows two side-by-side browser windows. The left window is from w3schools.com, titled 'CSS Example' under 'CSS Selectors'. It contains a code snippet: 

```
p { color: red; text-align: center; }
```

 and a 'Try it Yourself' button. The right window is from runoob.com, titled 'CSS 语法' (Syntax). It shows a similar example: 

```
h1 { color:blue; font-size:12px; }
```

 with a breakdown of '选择器' (Selector), '声明' (Declaration), '属性' (Property), and '值' (Value).

**CSS 怎么学 —— 免费课程**

freecodecamp.org

Github 上 star 最多的项目。由社区贡献而成的全面的程序员自学课程。

The screenshot shows a freeCodeCamp.org tutorial titled 'Change the Color of Text'. It includes a sidebar with search and filter options like 'Responsive Web Design' and 'Basic CSS'. The main content area shows a code editor with the following code:

```
1 <h2>CatPhotoApp</h2>
2 <main>
3   <p>Click here to view more <a href="#">cat photos</a>.</p>
4
5   <a href="#"></a>
6
7   <div>
8     <p>Things cats love:</p>
9     <ul>
10    <li>cat nap</li>
11    <li>laser pointers</li>
12    <li>lasagna</li>
13  </ul>
14  <p>Top 3 things cats hate:</p>
15  <ol>
16    <li>flea treatment</li>
17    <li>thunder</li>
18    <li>other cats</li>
19  </ol>
20 </div>
```

To the right, there's a preview of a cat photo with the caption 'CatPhotoApp' and a link 'Click here to view more cat photos.' Below the preview, there are lists for 'Things cats love:' and 'Top 3 things cats hate:', each with three items. At the bottom, there are checkboxes for 'Indoor' (unchecked), 'Outdoor' (checked), 'Loving' (checked), 'Lazy' (unchecked), 'Energetic' (unchecked), and a 'Submit' button.

**CSS 怎么学——持续学习**

下面这些真学习资料，如果你干前端这行，它们会陪你到 35 岁一辈子。

**MDN 文档**  
<https://developer.mozilla.org/zh-CN/docs/Web/CSS>

css 文档库。基本会是前端程序员最常来的地方 (html、js 规则也在里面)。同时，里面还有丰富的各式教程，作为一个自由探索的网站也是相当不错。

**MDN Web Docs** mozilla.org

Web 开发技术 > CSS (层叠样式表)

This page was translated from English by the community. Learn more and join the MDN Web Docs community.

Table of contents

- 教程
- 相关链接

Related Topics

- Learn CSS
  - CSS first steps
  - CSS building blocks
  - 样式化文字

**CSS (层叠样式表)**

教程

层叠样式表 (Cascading Style Sheets, 缩写为 CSS)，是一种 层叠式语言，用来描述 HTML 或 XML (包括 SVG、MathML、XHTML 之类 XML 分支语言) 文档的呈现。CSS 描述了在屏幕、纸张、音频等其它媒体上的元素应该如何被渲染的问题。

CSS 是开放网络的核心语言之一，由 W3C 维护，实现跨浏览器的标准。CSS 节省了大量的工作。样式可以定义保留在外部.css文件中，同时控制多个网页的布局。这意味着开发者不必经历在所有网页上编辑布局的麻烦。CSS 被分为不同等级：CSS1 现已废弃，CSS2.1 是推荐标准，CSS3 分成多个模块且正在标准化中。

**CSS 怎么学——持续学习**

前面的课程挑一个来学就行。但下面这些真学习资料，如果你干前端这行，它们会陪你一辈子。

**CSS-TRICKS**  
<https://css-tricks.com/almanac/>

另一个 css 文档库，用有别于 mdn 的形式组织 css 知识，还有非常活跃的 css 社区，里面有各种奇技淫巧。作为 mdn 文档的补充相当合适。

**\* CSS-TRICKS**

ARTICLES VIDEOS ALMANAC NEWSLETTER GUIDES BOOKS

## ALMANAC

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

**Selectors**

CSS selectors are patterns used to select elements in the DOM.

**Properties**

CSS properties style elements through their values.



# CSS 怎么学 —— 持续学习

\*w3c CSS 标准  
w3.org/Style/CSS

前面的不管是教程还是文档，基本只会教你用法。大部分用法其实也是自说明的，比如 color: red; 就是文本变成红色，还要啥自行车？但：

1. 现实中有很多 css 规则，在不同的场景下会不同的表现，而普通文档不会告诉你为什么（甚至不告诉你有这事儿）
2. 不像 color 这种，有很多 css 规则组合起来使用会实现很神奇的效果，你可能会好奇为什么（其实 color 能影响的远不止文本的颜色，你想知道么？）

w3c 文档，就是解答这些问题

## CSS 常用规则

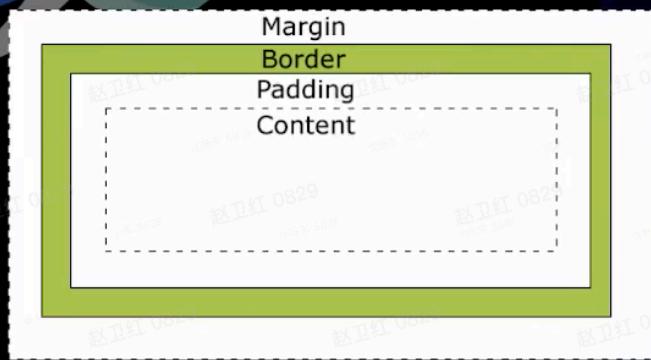
### 布局



# CSS 常用规则 —— 布局

### 盒模型

在具体介绍布局相关的属性之前，先要了解：在 CSS 中，所有的元素都被一个个的“盒子（box）”包围着，理解这些“盒子”的基本原理，是我们使用 CSS 实现准确布局、处理元素排列的关键。





# CSS 常用规则 —— 布局

赵卫红 0829 ByteDance 字节跳动

## 盒模型

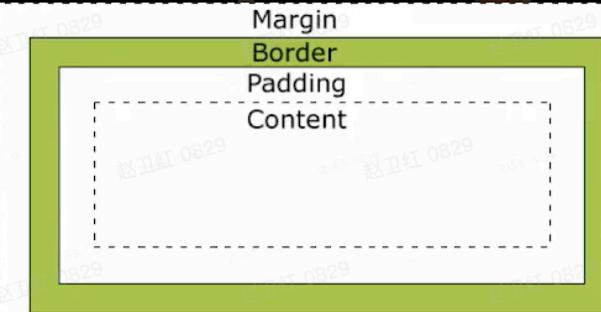
CSS 中组成一个块级盒子需要：

Content box: 这个区域是用来显示内容，大小可以通过设置 width 和 height。

Padding box: 包围在内容区域外部的空白区域；大小通过 padding 相关属性设置。

Border box: 边框盒包裹内容和内边距。大小通过 border 相关属性设置。

Margin box: 这是最外面的区域，是盒子和其他元素之间的空白区域。大小通过 margin 相关属性设置。



# CSS 常用规则 —— 布局

赵卫红 0829 ByteDance 字节跳动

## 盒模型

“盒子”——块级盒子 (block box) 和 内联盒子 (inline box)。这两种盒子会在页面流 (page flow) 和元素之间的关系方面表现出不同的行为：

```
<p>
  I am a paragraph. Some of the
  <span>words</span> have been wrapped in a
  <span>span element</span>.
</p>
```

I am a paragraph. Some of the words have been wrapped in a span element.



## CSS 常用规则 —— 布局

ByteDance

字节跳动

### 盒模型

一个被定义成块级的 (block) 盒子会表现出以下行为:

- 盒子会在内联的方向上扩展并占据父容器在该方向上的所有可用空间，在绝大多数情况下意味着盒子会和父容器一样宽
- 每个盒子都会换行
- width 和 height 属性可以发挥作用
- 内边距 (padding)，外边距 (margin) 和边框 (border) 会将其他元素从当前盒子周围“推开”

除非特殊指定，诸如标题(<h1>等)和段落(<p>)默认情况下都是块级的盒子。

```
<p>  
I am a paragraph. Some of the  
<span>words</span> have been wrapped in a  
<span>span element</span>.  
</p>
```

I am a paragraph. Some of the words have been  
wrapped in a span element.



## CSS 常用规则 —— 布局

ByteDance

字节跳动

### 盒模型

如果一个盒子对外显示为 inline，那么他的行为如下:

- 盒子不会产生换行。
- width 和 height 属性将不起作用。
- 垂直方向的内边距、外边距以及边框会被应用但是不会把其他处于 inline 状态的盒子推开。
- 水平方向的内边距、外边距以及边框会被应用且会把其他处于 inline 状态的盒子推开。

用做链接的 <a> 元素、<span> 都是默认处于 inline 状态的。

```
<p>  
I am a paragraph. Some of the  
<span>words</span> have been wrapped in a  
<span>span element</span>.  
</p>
```

I am a paragraph. Some of the words have been  
wrapped in a span element.



# CSS 常用规则 —— 布局

## 盒模型

让我们继续看看别的例子

```
p {  
    border: 2px solid purple;  
    padding: .5em;  
}  
span {  
    color: green;  
}  
.block {  
    display: block;  
    border: 2px solid blue;  
    padding: .5em;  
}  
  
<p>I am a paragraph. A short one.</p>  
<p>I am another paragraph. Some of the <span> have been wrapped in a <span> element</span>.</p>
```

I am a paragraph. A short one.

I am another paragraph. Some of the  
words

have been wrapped in a span element.

Styles Computed Layout Event Listener  
Filter  
p {  
 display: block;  
 margin-block-start: 1em;  
 margin-block-end: 1em;  
 margin-inline-start: 0px;  
 margin-inline-end: 0px;  
}



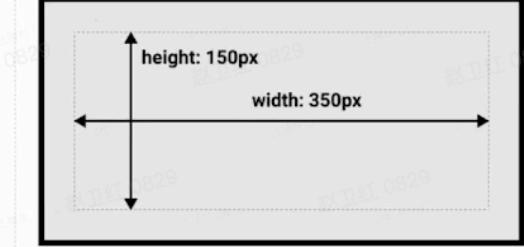
# CSS 常用规则 —— 布局

## 盒模型

这个例子中，box 元素真正占用的面积是多少？

模型宽度 = 410px (350 + 25 + 25 + 5 + 5)，高度 = 210px (150 + 25 + 25 + 5 + 5)，padding 加 border 再加 content box。

```
.box {  
    width: 350px;  
    height: 150px;  
    margin: 25px;  
    padding: 25px;  
    border: 5px solid black;  
}
```





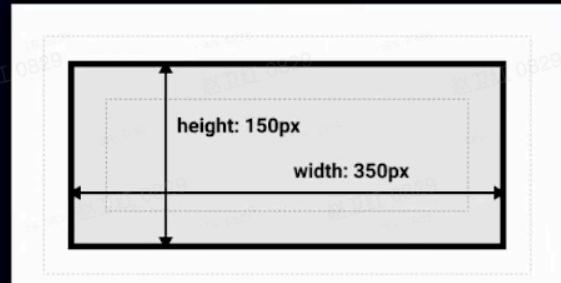
# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 盒模型

你可以修改浏览器这一默认行为，通过设置 `box-sizing: border-box` 来实现（默认是 `content-box`）

```
.box {  
    width: 350px;  
    height: 150px;  
    margin: 25px;  
    padding: 25px;  
    border: 5px solid black; }  
  
.box {  
    box-sizing: border-box; }
```



# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 盒模型

如果你希望所有元素都使用替代模式，而且确实很常用，设置 `box-sizing` 在 `<html>` 元素上，然后设置所有元素继承该属性，正如下面的例子

```
html {  
    box-sizing: border-box;  
}  
  
*, *::before, *::after {  
    box-sizing: inherit; }
```



# CSS 常用规则 —— 布局

字节跳动

## 盒模型

内联盒子只适用部分块级盒子属性。

可以看到，宽度和高度被忽略了。外边距、内边距和边框是生效的，但它们不会改变其他内容与内联盒子的关系

```
span {  
    margin: 20px;  
    padding: 20px;  
    width: 800px;  
    height: 0px;  
    border: 2px solid blue;  
}  
  
<p>  
    I am a paragraph and this is a  
<span>span</span> inside that paragraph.  
    A span is an inline element and so does  
    not respect width and height.  
</p>
```

I am a paragraph and this is a span inside that paragraph. A span is an inline element and so does not respect width and height.



# CSS 常用规则 —— 布局

字节跳动

## 盒模型

display有一个特殊的值，它在内联和块之间提供了一个中间状态。这对于以下情况非常有用：您不希望一个项切换到新行，但希望它可以设定宽度和高度

```
span {  
    margin: 20px;  
    padding: 20px;  
    width: 80px;  
    height: 50px;  
    border: 2px solid blue;  
    display: inline-block;  
}
```

I am a paragraph and this is a span inside that paragraph. A span is an inline element and so does not respect width and height.



# CSS 常用规则 —— 布局

字节跳动

## 盒模型

本章所学 CSS 规则：

- margin：设置外边距
- border：设置边框
- padding：内边距
- width & height：内容宽高
- box-sizing：修改宽高的定义范围
- display：改变元素是块级盒子还是内联盒子的特征

关于盒模型和上述属性，还有更多的使用技巧和延伸属性，可以进一步前往 mdn 学习。



# CSS 常用规则 —— 布局

## 正常文档流

盒模型解释了一个元素块的尺寸；那元素块在整篇文档中的位置是如何确定的呢？

```
<p>  
    这是 p 标签内容  
</p>  
<p>  
    这是 p 标签内容  
    <span>这是 span 标签内容</span>  
</p>
```

这是 p 标签内容

这是 p 标签内容 这是 span 标签内容

```
p {  
    background: rgba(255, 84, 104, 0.3);  
    border: 2px solid rgb(255, 84, 104);  
    padding: 10px;  
    margin: 10px;  
}  
  
span {  
    background: white;  
    border: 1px solid black;  
}
```



# CSS 常用规则 —— 布局

## 正常文档流

默认的，块级元素按照基于其父元素的书写顺序(默认值: horizontal-tb)的块流动方向(block flow direction)放置

每个块级元素会在上一个元素下面另起一行，它们会被设置好的margin 分隔。块级元素是垂直组织的。

```
<p>  
    这是 p 标签内容  
</p>  
<p>  
    这是 p 标签内容  
    <span>这是 span 标签内容</span>  
</p>
```

这是 p 标签内容

这是 p 标签内容 这是 span 标签内容

```
p {  
    background: rgba(255, 84, 104, 0.3);  
    border: 2px solid rgb(255, 84, 104);  
    padding: 10px;  
    margin: 10px;  
}  
  
span {  
    background: white;  
    border: 1px solid black;  
}
```



# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 正常文档流

默认的，块级元素按照基于其父元素的书写顺序(默认值: horizontal-tb)的块流动方向(block flow direction)放置

每个块级元素会在上一个元素下面另起一行，它们会被设置好的margin 分隔。块级元素是垂直组织的。

```
<p>  
    这是 p 标签内容  
</p>  
<p>  
    这是 p 标签内容  
<span>这是 span 标签内容</span>  
</p>
```

```
p {  
background: rgba(255, 84, 104, 0.3);  
border: 2px solid #rgb(255, 84, 104);  
padding: 10px;  
margin: 10px;  
}  
  
span {  
background: white;  
border: 1px solid black;
```

这是 p 标签内容

这是 p 标签内容 这是 span 标签内容



# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 弹性布局

弹性盒子是一种用于按行或按列布局元素的一维布局方法。元素可以膨胀以填充额外的空间, 收缩以适应更小的空间

### Sample flexbox example

#### First article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Second article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

#### Third article

Tacos actually microdosing, pour-over semiotics banjo chicharrones retro fanny pack portland everyday carry vinyl typewriter. Tacos PBR&B pork belly, everyday carry ennui pickled sriracha normcore hashtag polaroid single-origin coffee cold-pressed. PBR&B tattooed trust fund twee, leggings salvia iPhone photo booth health goth gastropub hammock.

Cray food truck brunch, XOXO +1 keffiyeh pickled chambray waistcoat ennui. Organic small batch paleo 8-bit. Intelligentia umami wayfarers pickled, asymmetrical kombucha letterpress kitsch leggings cold-pressed squid chartreuse put a bird on it. Listicle pickled man bun cornhole heirloom art party.



# CSS 常用规则 —— 布局

字节跳动

## 弹性布局

只需要在容器上加 `display: flex` 属性。你就可以得到一个横排的布局。

就盒模型而言，此时容器类似于块级元素 (`display: block`)，但宽度默认由子元素决定；

子元素类似于 `inline-block` 元素，可以设宽高且不换行。其原本的 `display` 属性基本被无视了，这是一种隐含的盒模型状态。

```
.container {  
    display: flex;  
}
```



# CSS 常用规则 —— 布局

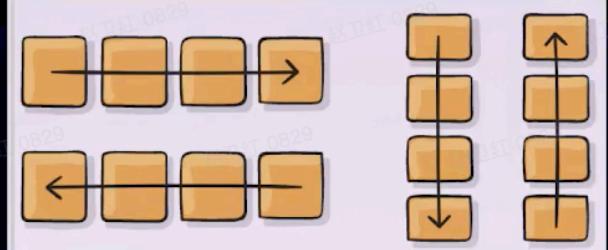
字节跳动

## 弹性布局

这种模型非常灵活。可以完成 90% 的布局要求。

`flex-direction` 属性控制子元素的排布顺序

```
.container {  
    display: flex;  
}  
  
.container {  
    flex-direction: row | row-reverse | column | column-reverse;  
}
```





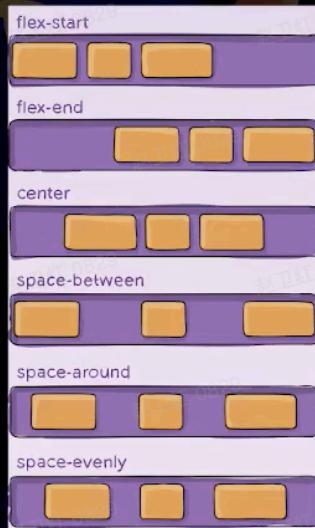
# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 弹性布局

flex 容器类似块级盒子，可以设置比子元素所需更大的宽度。  
justify-content 属性，帮你确定此时子元素如何分配空间

```
.container {  
    display: flex;  
}  
  
.container {  
    justify-content: flex-start | flex-end | center | space-between  
}
```



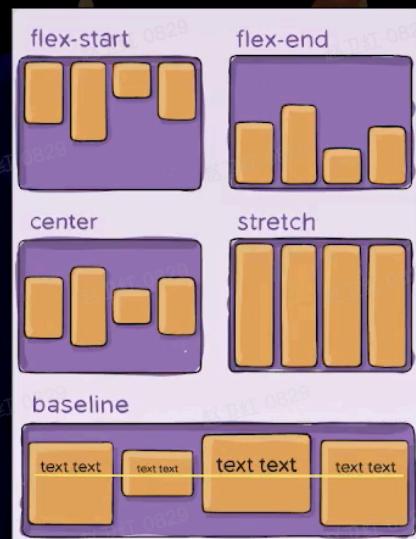
# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 弹性布局

同样的，当父容器高度溢出时，align-items 帮你确定子元素如何垂直对齐

```
.container {  
    display: flex;  
}  
  
.container {  
    align-items: stretch | flex-start | flex-end | center |  
}
```



# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 弹性布局

以上是最常用的弹性盒子用法。50% 的情况下，浏览器会恰到好处地帮你分配好各个元素所占的空间；但另外一半是需要你做额外调整的。这时你需要了解弹性盒子的更多属性

[CSS-TRICKS —— 弹性盒子介绍](#)



# CSS 常用规则 —— 布局



弹性盒子所学回顾：

- `display: flex;` 一键开启弹性盒子模式，替代正常文档流
- `flex-direction:` 更改弹性盒子的「主轴」
- `justify-content:` 子元素主轴线上如何对齐
- `align-items:` 子元素横轴线上如何对齐

```
<div style="display: flex; align-items: center; justify-content: center; width: 200px; height: 200px; background: blue;>
  <div style="color: white; border: 1px solid white;">Hello</div>
</div>
```



# CSS 常用规则 —— 布局



## 定位 (position)

不管是正常的文档流，还是弹性盒子，内里元素排布都是相互影响的。前面的元素占了一块地，后面的元素肯定得往后稍稍。

这其实只是其中一种「定位情况」，叫「静态定位」  
具体到 CSS 就是 `position: static;` 默认给到每个元素

```
<p>
  这是 p 标签内容
</p>
<p>
  这是 p 标签内容
  <span>这是 span 标签内容</span>
</p>
```





# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 定位 (position)

position 还能有其他值，比如「相对定位」，position: relative

相对定位在文档流中仍然占据位置，但可以用 top, left, bottom, right 属性做一些偏移的操作

```
<p>
    这是 p 标签内容
</p>
<div>
    这是 p 标签内容
    <span style="position: relative; top: 16px; left: 8px">
        这是 span 标签内容
    </span>
    内容
</p>
```

这是 p 标签内容

这是 p 标签内容

这是 span 标签内容



# CSS 常用规则 —— 布局

ByteDance  
字节跳动

## 定位 (position)

position: absolute，这种叫「绝对定位」。绝对定位的元素完全脱离了文档流和什么弹性盒子。绝对定位的盒子的定位、大小，你可以完全指定。

此时 top, left, 已经不是相对于原位置，而是相对于一个非 static 定位的父元素容器  
顺便讲一个类似的 position: fixed，其 top, left 等属性是相对于浏览器窗口

```
<p>
    这是 p 标签内容
</p>
<div>
    这是 p 标签内容
    <span style="position: absolute; top: 16px; left: 8px">
        这是 span 标签内容
    </span>
    内容
</p>
```

这是 span 标签内容

这是 p 标签内容 内容



# CSS 常用规则 —— 布局

赵卫红 0829 ByteDance 字节跳动

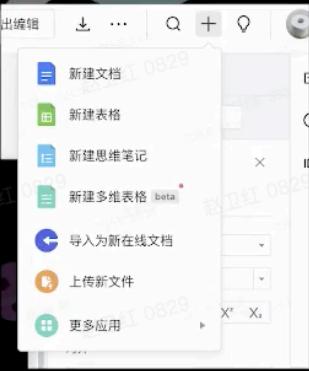
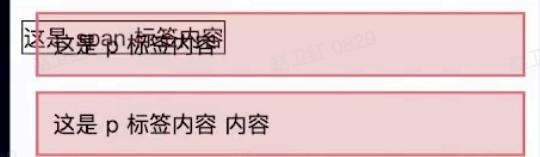
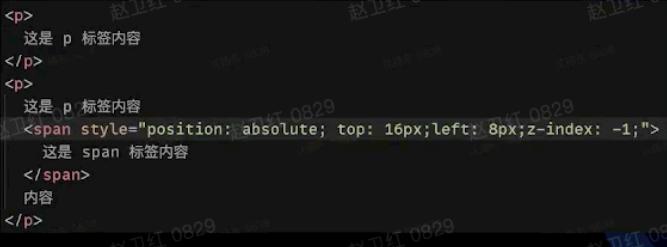
## 定位 (position)

因为非 static 值的 position 属性让元素之间可以相互覆盖, CSS 提供了 z-index 属性来控制哪个元素覆盖在最上层

# CSS 常用规则 —— 布局

## 定位 (position)

非 static 值的定位让元素非常独立可控, 广泛运用在弹窗、下拉选项、固定导航栏等场景。



## 装饰相关



# CSS 常用规则 —— 装饰相关



关于布局流、弹性盒子和绝对定位的使用，还有很多场景和使用技巧没有提及。另外还有 float、table、grid 布局。这些有待同学们在线上持续去探索。

下面，我们来看看在装饰领域，CSS 可以在哪些方面帮助我们装点页面吧



# CSS 常用规则 —— 装饰相关



## 文字

```
<div style="border: 1px solid black; width: 250px">  
  <div style="">Basic document flow</div>  
  <div style="font-family: serif;">Basic document flow</div>  
  <div style="color: purple;">Basic document flow</div>  
  <div style="font-weight: bold;">Basic document flow</div>  
  <div style="font-style: italic;">Basic document flow</div>  
  <div style="text-decoration: underline;">Basic document flow</div>  
  <div style="text-align: center;">Basic document flow</div>  
  <div style="font-size: 24px">Basic document flow</div>  
</div>  
  
<a href="https://douyin.com/sports">hello</a>
```

Basic document flow  
Basic document flow

hello hello hello

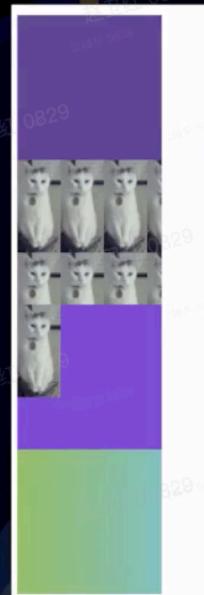


# CSS 常用规则 —— 装饰相关

ByteDance  
字节跳动

## 背景

```
<style>
  .bg {
    width: 100px;
    height: 100px;
  }
  .bg1 {
    background-color: #rebeccapurple;
  }
  .bg2 {
    background-image: url("https://mdn.mozilla..."); 
  }
  .bg22 {
    background-repeat: no-repeat;
    background-color: #blueviolet;
  }
  .bg3 {
    background-image: linear-gradient(
      to right,
      hsl(100, 50%, 50%),
      hsl(180, 50%, 50%)
    );
  }
</style>
<div class="bg bg1"></div>
<div class="bg bg2"></div>
<div class="bg bg2 bg22"></div>
<div class="bg bg3"></div>
```



# CSS 常用规则 —— 装饰相关

ByteDance  
字节跳动

## 边框

```
border: solid;
border: dashed red;
border: 1rem solid;
border: thick double #32a1ce;
border: 4mm ridge rgba(170, 50, 220, .6);
```

This is a box with a border around it.

This is a box with a border around it.

This is a box with a border around it.

This is a box with a border around it.

This is a box with a border around it.



## CSS 常用规则 —— 装饰相关

ByteDance

字节跳动

### 阴影

```
box-shadow: 10px 5px 5px red;
```

```
box-shadow: 60px -16px teal;
```

```
box-shadow: 12px 12px 2px 1px rgba(0, 0, 255, .2);
```

```
box-shadow: inset 5em 1em gold;
```

```
box-shadow: 3px 3px red, -1em 0 .4em olive;
```

This is a box with a box-shadow around it.

This is a box with a box-shadow around it.

This is a box with a box-shadow around it.



## CSS 常用规则 —— 装饰相关

ByteDance

字节跳动

### 阴影

```
<div style="text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.6)">你好</div>
<div style="text-shadow: 4px 4px 4px rgba(0, 0, 0, 0.6)">你好</div>
```

你好  
你好



## CSS 常用规则 —— 装饰相关

ByteDance

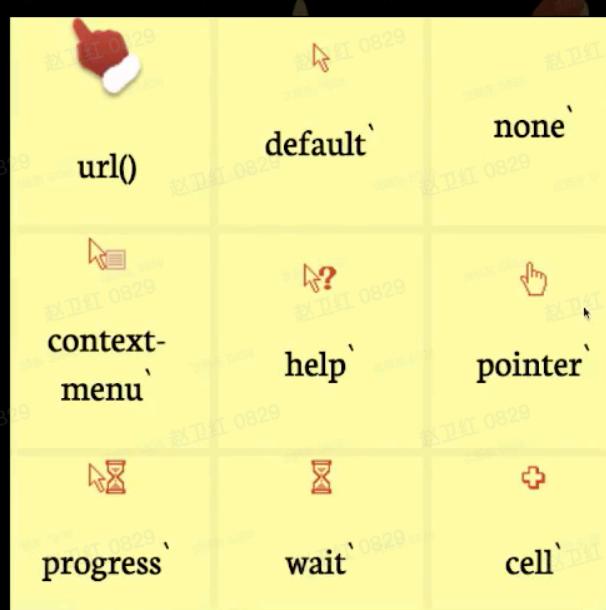
字节跳动

交互相关

cursor-pointer

可交互体验

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/cursor>



## CSS 常用规则 —— 装饰相关

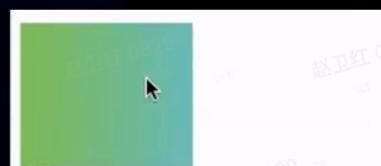
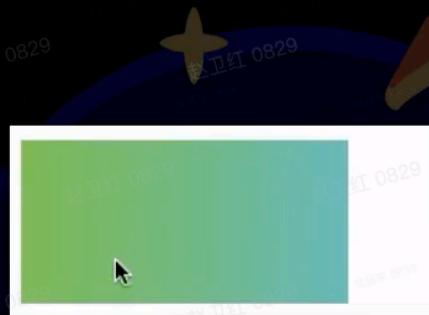
ByteDance

字节跳动

交互相关

```
<style>
.tran:hover {
    width: 200px;
}
</style>
<div class="bg bg3 tran"></div>
```

```
.tran {
    transition: width 300ms ease-in;
}
```





## CSS 常用规则——装饰相关

赵卫红 0829

字节跳动

### 动画

```
.tran {  
    transition: width 300ms ease-in;  
    animation: spin 5s ease-in-out infinite;  
}  
  
@keyframes spin {  
    from {  
        transform: rotate(0deg);  
    }  
    to {  
        transform: rotate(360deg);  
    }  
}
```



@keyframes帧动画

## 调试

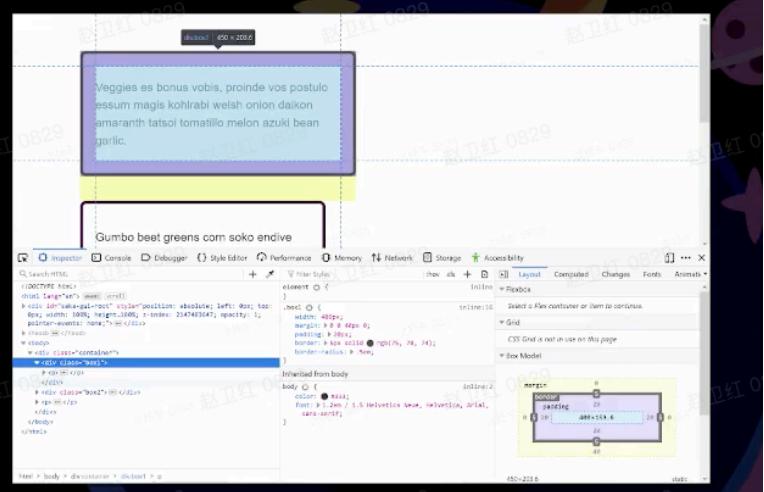


## CSS 精益求精——调试

赵卫红 0829  
字节跳动

### CSS 调试

你有时写 CSS 会碰到这样的问题：结果看起来和你想的不太一样。你可能会认为 selector (选择器) 匹配到了元素但是什么都没发生，还可能会觉得盒子的大小与你想的有出入。这章会教你着手调试 CSS，向你展示现代浏览器中的 DevTools 是怎样让你更方便地获悉发生了什么。



## CSS 扩展



## CSS 精益求精 —— CSS 扩展

赵卫红 0829

ByteDance

字节跳动

### CSS 预处理器

另一种组织CSS的方法是利用一些对于前端开发者可用的工具，它们让你可以稍微更程式化地编写CSS。预处理工具以你的原文件为基础运行，将它们转化为样式表。代表工具有：

- less
- sass
- stylus

# UI设计



## 功能导向 —— 设计中最重要的事

赵卫红 0829

ByteDance

字节跳动

### 设计简单的、完整的功能

- 一个联系人列表，是否要考虑超过 2000 人的展示情况？如何优化交互？
- 数据不存在时，根据不同的错误情况，应该如何准备错误信息展示，和用户引导？
- 一个图片上传模块，其复制粘贴、拖拽、点击选择文件上传，这些路径是否 MVP 版本都要做，是否都要先做好布局和引导设计？

在开周期间，时刻提醒自己，优先把用户关键路径上的主流场景设计完成，迅速迭代能力。  
先让游戏开始。

Add a new comment



Type your message...

Attach files by dragging & dropping or selecting them.

Some HTML is OK.

Post Comment

功能优先！



## 设计原则 —— 层级

赵卫红 0829

ByteDance

字节跳动

### 层级 —— 你可能唯一需要关心的原则

这一章我们来分享你可能唯二需要知道的设计原则。

如果要从「唯二」变成「唯一」，那么层级区分，就是你最需要掌握的原则和技巧。

- 一个产品要好用，就要让用户很容易地，抓到产品重点。
- 在重点里，用户能自在地进入沉浸式阅读、和使用的状态；
- 当用户想探索其他内容时，ta 能轻松地识别网站的次要板块还有哪些，焦点能迅速转移，并迅速沉浸；
- 辅助提示信息，精确而又不会打扰各主要模块的呈现。



# 设计原则 —— 层级

ByteDance  
字节跳动

层级 —— 你可能唯一需要关心的原则

哪张图片展示了更好的阅读体验?

Overview As of December 11, 2018, 8:00 AM PT		Overview As of December 11, 2018, 8:00 AM PT																					
Total Balance:	\$16,412.92	INVESTMENTS:	15,606.14																				
+\$648.83		+\$648.83																					
<table border="1"> <thead> <tr> <th>Holdings</th> <th>Activity</th> <th>Performance</th> <th>Gain &amp; Loss</th> </tr> </thead> <tbody> <tr> <td>TDB900 - TD CDN INDEX-E...</td> <td>140.143</td> <td>\$25.70</td> <td>\$24.49</td> </tr> <tr> <td>TDB902 - TD US INDX CS...</td> <td>65.218</td> <td>\$54.86</td> <td>\$51.95</td> </tr> <tr> <td>TDB909 - TD CD BD IDX-E...</td> <td>415.872</td> <td>\$11.64</td> <td>\$11.63</td> </tr> <tr> <td>TDB911 - TD INTL IDX E SE...</td> <td>271.861</td> <td>\$13.19</td> <td>\$12.15</td> </tr> </tbody> </table>				Holdings	Activity	Performance	Gain & Loss	TDB900 - TD CDN INDEX-E...	140.143	\$25.70	\$24.49	TDB902 - TD US INDX CS...	65.218	\$54.86	\$51.95	TDB909 - TD CD BD IDX-E...	415.872	\$11.64	\$11.63	TDB911 - TD INTL IDX E SE...	271.861	\$13.19	\$12.15
Holdings	Activity	Performance	Gain & Loss																				
TDB900 - TD CDN INDEX-E...	140.143	\$25.70	\$24.49																				
TDB902 - TD US INDX CS...	65.218	\$54.86	\$51.95																				
TDB909 - TD CD BD IDX-E...	415.872	\$11.64	\$11.63																				
TDB911 - TD INTL IDX E SE...	271.861	\$13.19	\$12.15																				



# 设计原则 —— 一致性

ByteDance  
字节跳动

以飞书文档为例

当页面中，主要的交互、视觉元素都采用同一主题色（图中为蓝色）来表示时，用户可以迅速知道：

- 页面中有哪些元素是可交互的
- 我需要的提示信息在哪

前端 UI 设计基础 \*

没设置

已经保存到云端

发表

前端 UI 设计基础  
背景  
做好作品  
没有专业 UI 来帮助  
小组  
功能架构 —— 设计中最重要的事  
简约设计  
用简单的创作工具  
设计简单的、完整的功能  
设计准则  
层级 —— 你可能唯一需要关心的原则  
一致性 —— 如何表现得专业  
雷同  
设计体系 —— 开始工作  
间距  
字体  
△ △

在我们学习具体设计手段之前，本章只想传递一个观点：  
It's almost always about 基层。

一致性 —— 如何表现得专业  
让应用各个层级有整体性  
依循体系  
克制  
亲切感  
对齐

一致性指的是：用户在站点的各个角落，观察到颜色、间距、阴影、位置、字体和字重的运用，都在一套有组织的框架里，一套能迅速建立亲切感的框架内。什么意思呢？



# 设计原则 —— 一致性



能否克制，且重复、精确地运用着拟定好的设计体系，会在方方面面影响着一个站点的质感，也是体现专业性的考量。

- 随意地配置间距 vs 克制地运用间距体系

