# HTML

* 行内元素、块状元素、行内块状元素

[HTML行内元素、块状元素、行内块状元素的区别 - 歆轶 - 博客园 (cnblogs.com)](https://www.cnblogs.com/ljwk/p/7090320.html)

|  |
| --- |
| 1、行内元素：a img span b strong input select section  2、块级元素：div p table ul lo li h1-h6 dl dt  (1)display:inline;转换为行内元素  设置宽高无效  　　　对margin仅设置左右方向有效，上下无效；padding设置上下左右都有效，即会撑大空间  　　　不会自动进行换行   1. display:block;转换为块状元素   能够识别宽高  　　　margin和padding的上下左右均对其有效  　　　可以自动换行  　　　多个块状元素标签写在一起，默认排列方式为从上至下  　　(3)display:inline-block;转换为行内块状元素  不自动换行  　　　能够识别宽高  　　　默认排列方式为从左到右 |

* crossorigin 属性

[CORS settings attributes - HTML（超文本标记语言） | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Attributes/crossorigin)

[浅析html5中crossorigin属性 - 码农教程 (manongjc.com)](http://www.manongjc.com/detail/24-cvvmkobytgjbhdm.html)

[script crossorigin 属性 - 掘金 (juejin.cn)](https://juejin.cn/post/6969825311361859598)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | 关键字 | 描述 | | anonymous | 对此元素的 CORS 请求将不设置凭据标志。 | | use-credentials | 对此元素的 CORS 请求将设置凭证标志；这意味着请求将提供凭据。 | | "" | 设置一个空的值，如 crossorigin 或 crossorigin=""，和设置 anonymous 的效果一样。 | |
| · 设置了crossorigin就相当于开启了cors校验。  · 开启cors校验之后，跨域的script资源在运行出错的时候，window.onerror可以捕获到完整的错误信息。  · crossorigin=use-credentials可以跨域带上cookie。 |

* 禁止用户缩放

[利用meta标签禁止用户缩放 - 简书 (jianshu.com)](https://www.jianshu.com/p/4332bdddc2e6)

[html怎么禁止页面缩放-html教程-PHP中文网](https://www.php.cn/div-tutorial-475413.html" \l ":~:text=%E5%9C%A8html%E4%B8%AD%EF%BC%8C%E5%8F%AF%E4%BB%A5%E4%BD%BF%E7%94%A8meta%E6%A0%87%E7%AD%BE%E6%9D%A5%E7%A6%81%E6%AD%A2%E9%A1%B5%E9%9D%A2%E7%BC%A9%E6%94%BE%EF%BC%8C%E5%8F%AA%E9%9C%80%E8%A6%81%E5%9C%A8meta%E6%A0%87%E7%AD%BE%E4%B8%AD%E8%AE%BE%E7%BD%AE%E2%80%9Cmaximum-scale%E2%80%9D%E5%B1%9E%E6%80%A7%E7%9A%84%E5%80%BC%E4%B8%BA%E2%80%9C1.0%E2%80%9D%EF%BC%8C%E2%80%9Cuser-scalable%E2%80%9D%E5%B1%9E%E6%80%A7%E7%9A%84%E5%80%BC%E4%B8%BA%E2%80%9C0%E2%80%9D%E5%8D%B3%E5%8F%AF%E3%80%82 %E6%9C%AC%E6%95%99%E7%A8%8B%E6%93%8D%E4%BD%9C%E7%8E%AF%E5%A2%83%EF%BC%9Awindows7%E7%B3%BB%E7%BB%9F%E3%80%81HTML5%E7%89%88%E3%80%81Dell,G3%E7%94%B5%E8%84%91%E3%80%82 html5%E9%A1%B5%E9%9D%A2%E4%B8%AD%E9%BB%98%E8%AE%A4%E9%83%BD%E5%85%81%E8%AE%B8%E7%94%A8%E6%88%B7%E7%BC%A9%E6%94%BE%E9%A1%B5%E9%9D%A2%EF%BC%8C%E6%88%96%E8%80%85%E5%9C%A8%E5%B1%8F%E5%B9%95%E5%8F%8C%E5%87%BB%E6%94%BE%E5%A4%A7%E6%88%96%E7%BC%A9%E5%B0%8F%E3%80%82)

|  |
| --- |
| <meta name="viewport" content="width=device-width, initial-scale=1.0">  通常情况下，maximum-scale=1.0 与 user-scalable=no 一起使用。这样禁用缩放功能后，用户只能滚动屏幕，就能让您的网站看上去更像原生应用的感觉。 |

* script 标签，async，defer

[图解 script 标签中的 async 和 defer 属性 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/292953374)

|  |
| --- |
|  |

* 阻止冒泡、阻止默认事件

[(4条消息) JS阻止事件冒泡和默认事件\_姜无忧的博客-CSDN博客\_jquery阻止事件冒泡的方法](https://blog.csdn.net/xiasohuai/article/details/86496745)

|  |
| --- |
| window.event  ev.stopPropagation();  e.preventDefault(); |

* BOM

[HTML BOM Browser对象 - FangMu - 博客园 (cnblogs.com)](https://www.cnblogs.com/polk6/p/5045277.html)

|  |
| --- |
| **BOM**：Browser Object Model，即浏览器对象模型，提供了独立于内容的、可以与浏览器窗口进行互动的对象结构。  **Browser对象**：指BOM提供的多个对象，包括：Window、Navigator、Screen、History、Location等。  其中Window对象为顶层对象，其他对象都为Window对象的子对象。 |

* 浏览器渲染的过程

[浏览器渲染原理与过程 - 简书 (jianshu.com)](https://www.jianshu.com/p/e6252dc9be32)

|  |  |  |  |
| --- | --- | --- | --- |
| · 浏览器将获取的HTML文档解析成DOM树。   |  | | --- | | * DOM树在构建的过程中可能会被CSS和JS的加载而执行阻塞 * display:none的元素也会在DOM树中 * 注释也会在DOM树中 * script标签会在DOM树中 |   · 处理CSS标记，构成层叠样式表模型CSSOM(CSS Object Model)。  · 将DOM和CSSOM合并为渲染树(rendering tree)，代表一系列将被渲染的对象。   |  | | --- | | * Render Tree和DOM Tree不完全对应 * display: none的元素不在Render Tree中 * visibility: hidden的元素在Render Tree中 |   · 渲染树的每个元素包含的内容都是计算过的，它被称之为布局layout。浏览器使用一种流式处理的方法，只需要一次绘制操作就可以布局所有的元素。   |  | | --- | | float元素，fixed元素会发生位置偏移（会脱离文档流），absoulte元素（会脱离文档流），（CSS定位） |   · 将渲染树的各个节点绘制到屏幕上，这一步被称为绘制painting。 |

* 懒加载

[(4条消息) 前端优化——懒加载篇\_YooHoeh的博客-CSDN博客\_前端懒加载](https://blog.csdn.net/u011215669/article/details/80526441)

[懒加载和预加载 - 简书 (jianshu.com)](https://www.jianshu.com/p/4876a4fe7731)

|  |
| --- |
| 1)首先，不要将图片地址放到src属性中，而是放到其它属性(data-original)中。 2)页面加载完成后，根据scrollTop判断图片是否在用户的视野内，如果在，则将data-original属性中的值取出存放到src属性中。 3)在滚动事件中重复判断图片是否进入视野，如果进入，则将data-original属性中的值取出存放到src属性中。 |

* 可视化区域

[(3条消息) js判断元素是否进入可视化区域\_yunchong\_zhao的博客-CSDN博客\_js元素进入视野](https://blog.csdn.net/yunchong_zhao/article/details/122040469)

[javascript 如何判断一个元素是否在可视区域中？ - 简书 (jianshu.com)](https://www.jianshu.com/p/d9f5f19e78da)

|  |
| --- |
| el.offsetTop - document.documentElement.scrollTop <= viewPortHeight  元素距离顶部的距离 - 滚动的高度 <= 设备可视化高度 |
| 屏幕可视窗口大小  \*\*原生方法\*\*：  window.innerHeight 标准浏览器及IE9+ ||  document.documentElement.clientHeight 标准浏览器及低版本IE标准模式 ||  document.body.clientHeight 低版本混杂模式  \*\*jQuery方法\*\*：  $(window).height(); |

# CSS

* 选择器优先级

[CSS选择器的优先级（精讲版） (biancheng.net)](http://c.biancheng.net/view/7216.html)

[面试官：css选择器有哪些？优先级？哪些属性可以继承？ · Issue #95 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/95)

|  |
| --- |
| 内联 > ID选择器 > 类选择器 > 标签选择器  id选择器（#box），选择id为box的元素  类选择器（.one），选择类名为one的所有元素  标签选择器（div），选择标签为div的所有元素  后代选择器（#box div），选择id为box元素内部所有的div元素  子选择器（.one>one\_1），选择父元素为.one的所有.one\_1的元素  相邻同胞选择器（.one+.two），选择紧接在.one之后的所有.two元素  群组选择器（div,p），选择div、p的所有元素 |

* 盒子模型

[CSS 盒子模型 | 菜鸟教程 (runoob.com)](https://www.runoob.com/css/css-boxmodel.html)

[(3条消息) CSS-盒子模型，标准盒子模型，IE 盒子模型，盒模型之间的转换\_蓬莱老仙的博客-CSDN博客\_盒模型转换](https://blog.csdn.net/weixin_44486539/article/details/101977563)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 元素可以通过 CSS 中的 box-sizing 样式属性可用来转换盒子模型   | **属性值** | **说明** | | --- | --- | | content-box | 标准盒子模型 | | border-box | IE盒子模型 | |

* 重绘重排

[CSS重绘和重排(回流) - 一梦梦 - 博客园 (cnblogs.com)](https://www.cnblogs.com/mdr86553/p/12048742.html)

|  |
| --- |
| 重排（回流）:当render树中的一部分或者全部因为大小边距等问题发生改变而需要DOM树重新计算的过程  重绘:当元素的一部分属性发生改变，如外观、背景、颜色等不会引起布局变化，只需要浏览器根据元素的新属性重新绘制，使元素呈现新的外观叫做重绘。  （1）不要一条一条地修改 DOM 的样式。可以先定义好 css 的 class，然后修改 DOM 的 className。  （2）不要把 DOM 结点的属性值放在一个循环里当成循环里的变量。  （3）为动画的 HTML 元件使用 fixed 或 absoult 的 position，那么修改他们的 CSS 是不会 reflow 的。  （4）千万不要使用 table 布局。因为可能很小的一个小改动会造成整个 table 的重新布局。(table及其内部元素除外，它可能需要多次计算才能确定好其在渲染树中节点的属性，通常要花3倍于同等元素的时间。这也是为什么我们要避免使用table做布局的一个原因。)  （5）不要在布局信息改变的时候做查询（会导致渲染队列强制刷新） （6）用translate替代top改变 （7）用opacity替代visibility（在独立图层下优化重绘） |

* 浮动和清除浮动

[CSS中的浮动和清除浮动，梳理一下！ - 简书 (jianshu.com)](https://www.jianshu.com/p/09bd5873bed4)

|  |
| --- |
| **浮动会导致父元素高度坍塌**  clear属性不允许被清除浮动的元素的左边/右边挨着浮动元素，底层原理是在被清除浮动的元素上边或者下边添加足够的清除空间。  <div class="box-wrapper">  <div class="box"></div>  <div class="box"></div>  <div class="box"></div>  <div class="box" style="clear:both;"></div>一种方法  </div>  <style>  .box{  background-color: red;  float: left;  width: 20px;  height:20px;  }  .box-wrapper:after { 父div直接after用clear：both另一种方法  display: table;  content: " ";  clear:both;  }  </style>  **BFC的触发方式**  我们可以给父元素添加以下属性来触发BFC： ✦ float 为 left | right ✦ overflow 为 hidden | auto | scorll ✦ display 为 table-cell | table-caption | inline-block | flex | inline-flex ✦ position 为 absolute | fixed  所以我们可以给父元素设置overflow:auto来简单的实现BFC清除浮动，但是为了兼容IE最好用overflow:hidden。 |

* 样式单位

[「css基础」一次搞懂CSS 字体单位：px、em、rem 和 % - 云+社区 - 腾讯云 (tencent.com)](https://cloud.tencent.com/developer/article/1492058)

[面试官：说说em/px/rem/vh/vw区别? · Issue #96 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/96)

|  |
| --- |
| * px：绝对单位，代表屏幕中每个「点」( pixel )。 * em：相对单位，每个子元素通过「倍数」乘以父元素的px值。 * rem：相对单位，每个元素通过「倍数」乘以根元素的px值。 * %：相对单位，每个子元素通过「百分比」乘以父元素的px值。 |

* 画三角形

[css如何做三角形-css教程-PHP中文网](https://www.php.cn/css-tutorial-473454.html)

|  |
| --- |
| css做三角形的方法：首先创建一个div元素，设置div的width和height为0，只用边框宽来填充，边框样式设置为实线“solid”；然后顶部边框设置颜色，剩下的三个边框的颜色设置为透明“transparent”值即可。  .trangle{  　　width: 0px;  　　height: 0px;  　　border-top: 100px solid red;  　　border-left: 100px solid transparent;  　　border-right: 100px solid transparent;  }  <div class="trangle"></div> |

* 定位

[(3条消息) Position定位（fixed，absolute，relative），Float\_自在飞花轻似梦\_wzf的博客-CSDN博客](https://blog.csdn.net/weixin_42277214/article/details/89715847?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~default-1-89715847-blog-112252464.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2~default~CTRLIST~default-1-89715847-blog-112252464.pc_relevant_default&utm_relevant_index=2)

[CSS 定位 (w3school.com.cn)](https://www.w3school.com.cn/css/pro_css_positioning.asp)

|  |
| --- |
| position有3个属性是我们平常用的最多的，即fixed，absolute，relative。  1、absolute，绝对定位，是相当于父元素（或者父父元素）的定位，脱离文档流；也不占据空间。后来者居上，后边的元素会被顶到它使用绝对定位之前的位置。  2、relative，相对定位，是相当于定位之前自身的位置，遵循文档流，定位之前的区域不会被顶替，还在原来占据着位置。定位之后的区域，不占据任何位置  3、fixed，固定定位，顾名思义，会固定在某一个位置，是相对于浏览器窗口的定位。脱离文档流。用的最多的是APP等软件搜索框固定在最头上、最顶部，网页两端的广告固定. |

* 文字最多不超过三行

[文字超出三行省略...显示全文 - 简书 (jianshu.com)](https://www.jianshu.com/p/3bdf313606c7)

text-overflow:ellipsis;

|  |
| --- |
| WebKit内核的浏览器实现起来比较简单，可以通过添加一个-webkit-line-clamp的私有属性来实现。  overflow : hidden;  text-overflow: ellipsis;  display: -webkit-box;  -webkit-line-clamp: 2;  -webkit-box-orient: vertical;  比较靠谱的办法应该就是利用相对定位在最后面加上一个省略号了。  p {  position:relative;  line-height:1.5em;  /\* 高度为需要显示的行数\*行高，比如这里我们显示两行，则为3 \*/  height:3em;  overflow:hidden;  }  p:after {  content:"...";  position:absolute;  bottom:0;  right:0;  padding: 0 5px;  background-color: #fff;  } |

* 伪类，伪元素

[伪类和伪元素 - 学习 Web 开发 | MDN (mozilla.org)](https://developer.mozilla.org/zh-CN/docs/Learn/CSS/Building_blocks/Selectors/Pseudo-classes_and_pseudo-elements)

[伪元素&伪类 - 简书 (jianshu.com)](https://www.jianshu.com/p/8b610fdf0d48)

|  |
| --- |
| css引入伪类和伪元素概念是为了格式化文档树以外的信息。  伪元素用于创建一些不在文档树中的元素，并为其添加样式。比如说，我们可以通过:before来在一个元素前增加一些文本，并为这些文本添加样式。虽然用户可以看到这些文本，但是这些文本实际上不在文档树中。  伪类用于当已有元素处于的某个状态时，为其添加对应的样式，这个状态是根据用户行为而动态变化的。比如说，当用户悬停在指定的元素时，我们可以通过:hover来描述这个元素的状态。虽然它和普通的css类相似，可以为已有的元素添加样式，但是它只有处于dom树无法描述的状态下才能为元素添加样式，所以将其称为伪类。 |

* 水平垂直居中

[css 水平居中（8种方法）、垂直居中（8种方法） - 掘金 (juejin.cn)](https://juejin.cn/post/6994718555925119012)

[面试官：元素水平垂直居中的方法有哪些？如果元素不定宽高呢？ · Issue #102 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/102)

|  |  |
| --- | --- |
| **水平居中:**  行内元素：  <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <style>  .parent {  text-align: center;  }  </style>  </head>  <body>  <div class='parent'>  <span>123</span>  </div>  </body>  </html> | **垂直居中：**  行内元素：  <!DOCTYPE html>  <html lang="en">  <head>  <style>  .parent {  background: #ccc;  height: 200px;  }  .son{  width: 50px;  height: 50px;  line-height: 200px;  }  </style>  </head>  <body>  <div class='parent'>  <span class="son">123</span>  </div>  </body>  </html> |
| **块级元素**  **Flex ：**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <style>  .parent {  display: flex;  justify-content: center;  }  .son{  float: left;  background: pink;  width:50px;  height:50px;  }  </style>  </head>  <body>  <div class='parent'>  <div class="son">  </div>  </div>  </body>  </html>  **绝对定位 + tranform：**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <style>  .parent {  }  .son{  float: left;  background: pink;  width:50px;  height:50px;  position: absolute;  left: 50%;  transform: translate(-50%, 0);  }  </style>  </head>  <body>  <div class='parent'>  <div class="son">  </div>  </div>  </body>  </html> | **块元素：**  **flex布局**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <style>  .parent {  background: #ccc;  height: 200px;  width: 200px;  display: flex;  align-items: center;  }  .son{  width: 50px;  height: 50px;  background: pink;  }  </style>  </head>  <body>  <div class='parent'>  <div class="son">123</div>  </div>  </body>  </html>  **绝对定位：**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Title</title>  <style>  .parent {  }  .son{  width: 50px;  height: 50px;  background: pink;  position: absolute;  top: 50%;  transform: translate( 0, -50%);  }  </style>  </head>  <body>  <div class='parent'>  <div class="son">123</div>  </div>  </body>  </html> |

* grid 布局

[CSS Grid 网格布局教程 - 阮一峰的网络日志 (ruanyifeng.com)](http://ruanyifeng.com/blog/2019/03/grid-layout-tutorial.html)

|  |
| --- |
| .container {  display: grid;  grid-template-columns: 100px 100px 100px;  grid-template-rows: 100px 100px 100px;  } |

* bfc

[面试官：谈谈你对BFC的理解？ · Issue #99 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/99)

|  |
| --- |
| 触发BFC的条件包含不限于：   * 根元素，即HTML元素 * 浮动元素：float值为left、right * overflow值不为 visible，为 auto、scroll、hidden * display的值为inline-block、inltable-cell、table-caption、table、inline-table、flex、inline-flex、grid、inline-grid * position的值为absolute或fixed   BFC目的是形成一个相对于外界完全独立的空间，让内部的子元素不会影响到外部的元素 **防止margin重叠（塌陷）** 可以在p外面包裹一层容器，并触发这个容器生成一个BFC，那么两个p就不属于同一个BFC，则不会出现margin重叠 **清除内部浮动** BFC在计算高度时，浮动元素也会参与，所以我们可以触发.par元素BFC，则内部浮动元素计算高度时候也会计算 **自适应多栏布局** 而BFC的区域不会与浮动盒子重叠  所以我们可以通过触发main生成BFC，以此适应两栏布局 |

* trasition 和 animation

[面试官：css3动画有哪些？ · Issue #109 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/109)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| transition的属性如下：   * property:填写需要变化的css属性 * duration:完成过渡效果需要的时间单位(s或者ms) * timing-function:完成效果的速度曲线 * delay: 动画效果的延迟触发时间  | **属性** | **含义** | | --- | --- | | transition（过度） | 用于设置元素的样式过度，和animation有着类似的效果，但细节上有很大的不同 | | transform（变形） | 用于元素进行旋转、缩放、移动或倾斜，和设置样式的动画并没有什么关系，就相当于color一样用来设置元素的“外表” | | translate（移动） | 只是transform的一个属性值，即移动 | | animation（动画） | 用于设置动画属性，他是一个简写的属性，包含6个属性 | |

* CSS画骰子

[(2条消息) css使用flex布局实现骰子1~6\_摇曳de瘋丶的博客-CSDN博客](https://blog.csdn.net/qq_37000055/article/details/115410995)

* 隐藏元素

[opacity: 0、visibility: hidden、display: none 优劣和适用场景，以及隐藏元素的几种方法 - 掘金 (juejin.cn)](https://juejin.cn/post/6844904202867572749)

[面试官：css中，有哪些方式可以隐藏页面元素？区别? · Issue #98 · febobo/web-interview (github.com)](https://github.com/febobo/web-interview/issues/98)

[(2条消息) display:none和visibility:hidden两者的区别\_@Demi的博客-CSDN博客\_visibility:hidden和display:none](https://blog.csdn.net/qq_38128179/article/details/80794397)

|  |
| --- |
| display: none  * **DOM 结构**：浏览器不会渲染 display 属性为 none 的元素，会让元素完全从渲染树中消失，渲染的时候不占据任何空间； * **事件监听**：无法进行 DOM 事件监听，不能点击； * **性能**：修改元素会造成文档回流（reflow 与 repaint）,读屏器不会读取display: none元素内容，性能消耗较大；  visibility: hidden  * **DOM 结构**：不会让元素从渲染树消失，渲染元素继续占据空间，只是内容不可见； * **事件监听**：无法进行 DOM 事件监听，不能点击； * **性能**：修改元素只会造成本元素的重绘（repaint），是重回操作，比回流操作性能高一些，性能消耗较少；读屏器读取visibility: hidden元素内容；  opacity: 0  * **DOM 结构**：透明度为 100%，不会让元素从渲染树消失，渲染元素继续占据空间，只是内容不可见； * **事件监听**：可以进行 DOM 事件监听，可以点击； * **性能**：不会引发重排，一般情况下也会引发重绘。提升为合成层，是重建图层，和动画属性一起则不会产生repaint（不脱离文档流，不会触发重绘），性能消耗较少； |

* a标签去下划线

[html怎么去掉a标签下划线-前端问答-PHP中文网](https://www.php.cn/website-design-ask-475216.html)

|  |
| --- |
| * a{text-decoration: none;}    //这个是设置a标签的默认状态去除下划线 * a:visited{text-decoration: none;}    //这个是设置a标签的访问过后的状态去除下划线 * a:hover {text-decoration: none;}    //这个是设置a标签的鼠标覆盖状态去除下划线 * a:active{text-decoration:none;}    //这个是设置a标签的活跃状态去除下划线 |