1 CSS介绍

1. CSS 全称为层叠样式表（Cascading Style Sheets），又叫级联样式表，简称样式表。
2. CSS 用于 HTML 文档中元素样式的定义，实现了将内容与表现分离，提高代码的可重用性和可维护性。
3. CSS 文件的后缀是 `.css`。
4. CSS 由浏览器解析执行。
5. CSS 不区分大小写，建议小写。
6. CSS 注释的语法：`/*注释内容*/`。注释之间不能嵌套。
7. CSS 的特点：
   （1）继承性：子元素可以继承父元素的样式。
   （2）层叠性：一个元素可以设置多个样式。当样式冲突时，优先级高的样式生效；优先级相同时，写在后面的样式生效。

2 CSS的引入方式

2.1 内联样式（行内样式）

在 HTML 标签内使用 `style` 属性，属性值是 CSS 代码，该样式只对当前元素生效。例如：

```
<p style="background: orange; font-size: 24px;">内容</p>
```

特点：缺乏整体性和规划性，不利于维护，维护成本高。

2.2 内部样式

当单个 HTML 文档需要特殊的样式时，就应该使用内部样式表。在 HTML 文档中使用 `style` 标签，在 `style` 标签中写 CSS 代码，该样式只对当前页面生效。`style` 标签通常写在 `head` 标签中，例如：

```
<head>
    <style>
        h1 {background: red;}
    </style>
</head>
```

特点：单个页面内的 CSS 代码具有统一性和规划性，便于维护，但是在多个页面之间容易混乱。

2.3 外部样式

在`head`标签中使用`link`标签引入 CSS 文件。`rel="stylesheet"`属性值表示引入的是样式表，`href`属性指定 CSS 文件的路径。例如：

```
<head>
    <link rel="stylesheet" href="" />
</head>
```

特点：实现了内容与表现的完全分离，提高了代码的可重用性和可维护性。

2.4 导入式

```
<style>
    @import "CSS文件路径";
    @import url("CSS文件路径");
</style>
```

`link`和`@import`的区别：

1. `@import`是 CSS 提供的加载样式的一种方式，只能用于加载 CSS；`link`可以引入多种内容。
2. `@import`先加载 HTML 文件，后加载 CSS 文件；`link`同时加载 HTML 文件和 CSS 文件。
3. `@import`会增加页面的 http 请求。
4. JavaScript 操作 DOM 时，只能操作`link`引入的 CSS 文件。
5. IE5 以下不支持`@import`，而`link`没有兼容性问题。

由于以上原因，一般不使用导入式。

2.5 引入方式的优先级

行内样式 > 内部样式 = 外部样式

内部样式和外部样式优先级相同，写在后面的生效。

3 选择器

选择器的作用：选中某些元素，为它们添加样式。

选择器的语法：`选择器{属性:属性值;}`

3.1 基础选择器

3.1.1 全局选择器

选择页面中的所有元素，优先级最低，不推荐使用。语法：`*{}`。

3.1.2 元素选择器

HTML标签名+大括号，选择页面上所有同种类型的标签，描述它们的“共性”。语法：`标签名{}`。

3.1.3 类选择器

在 HTML 标签中使用`class`属性定义类名，在 CSS 中使用`.类名`选择相应类名的所有标签。例如：

```html
<h2 class="oneclass"></h2>
<style>
    .oneclass {
        /* 属性:属性值; */
    }
</style>
```

`class`属性的特点：

1. 同一类名可以被多种标签使用。
2. 命名规则：可以包含数字、字母、下划线、`-`，不能以数字开头。
3. 同一个标签可以使用多个类名，用空格隔开。例如：

```html
<h3 class="classone classtwo"></h3>
```

3.1.4 ID选择器

在 HTML 标签中使用`id`属性定义 ID，在 CSS 中使用`#ID`选择相应 ID 的标签。例如：

```html
<h2 id="title"></h2>
<style>
    #title {
        /* 属性:属性值; */
    }
</style>
```

在 HTML 页面中不能出现相同的 ID，ID 具有唯一性。

3.1.5 合并选择器

为多个选择器设置公共样式，多个选择器之间用逗号隔开。

```
选择器1,选择器2,...,选择器n {
    /* 公共样式 */
}
```

3.1.6 交集选择器

`标签名.类名{}`，选择指定标签中具有指定类名的元素。

```
标签名.类名 {
    /* 属性:属性值; */
}
```

3.1.7 选择器的优先级

行内样式 > ID选择器 > 类选择器 > 元素选择器 > 全局选择器（范围越小，优先级越高）

选择器的优先级还可以通过权重计算：

选择器	权重
行内样式	1000
ID选择器	100
类选择器	10
元素选择器	1

多个选择器组合使用时（如合并选择器），其权重等于各个选择器权重之和。

3.2 关系选择器

3.2.1 后代选择器

选择所有被 E 元素包含的 F 元素。父元素和后代元素之间用空格隔开，语法如下：

```
E F {
    /* 属性:属性值; */
}
```

3.2.2 子代选择器

选择 E 元素的直接子元素 F，对更深层的子元素不起作用。父元素和子元素之间用`>`隔开，语法如下：

```
E>F {
    /* 属性:属性值; */
}
```

3.2.3 相邻兄弟选择器

选择紧跟在 E 元素后面的 F 元素，相邻元素之间用`+`连接。语法如下：

```
E+F {
    /* 属性:属性值; */
}
```

3.2.4 通用兄弟选择器

选择 E 元素之后的所有兄弟元素 F，两种元素之间用`~`连接。语法如下：

```
E~F {
    /* 属性:属性值; */
}
```

3.3 伪类选择器

同一个标签，根据其不同的状态有不同的样式，就叫做“伪类”。伪类用冒号`:`表示。

3.3.1 常用伪类选择器

1. `:link`：设置超链接点击之前的样式（只适用于`a`标签）
2. `:visited`：设置超链接被访问后的样式（只适用于`a`标签）
3. `:hover`：设置鼠标悬停在元素上时的样式（适用于所有标签）
4. `:active`：设置鼠标点击元素不松手时的样式（适用于所有标签）

当这四个选择器同时作用于一个元素上时，必须按照`:link`、`:visited`、`:hover`、`:active`依次从前到后的顺序书写。

3.3.2 CSS3新增伪类选择器

1. `:first-child`：当第一个子元素是给定的类型时，选择第一个子元素。
   例如：`div>p:first-child`表示当`div`的第一个子元素是`p`标签时，选择这个`p`标签，否则此样式无效。
2. `:last-child`：当最后一个子元素是给定的类型时，选择最后一个子元素。
3. `:nth-child(n)`：当第 n 个子元素是给定的类型时，选择第 n 个子元素。n 可以是数字、关键字或公式。
   关键字包括：`odd`表示奇数，`even`表示偶数。
4. `:first-of-type`：选择第一个给定类型子元素。
   例如：`div>p:first-of-type`表示选择`div`的子元素中第一个`p`标签。
5. `:last-of-type`：选择最后一个给定类型子元素。
6. `:nth-of-type(n)`：选择第 n 个给定类型子元素。
7. `:only-child`：当唯一子元素是给定的类型时，选择这个子元素。
8. `:nth-last-child(n)`：当倒数第 n 个子元素是给定的类型时，选择倒数第 n 个子元素。
9. `:nth-last-of-type(n)`：选择倒数第 n 个给定类型子元素。
10. `:empty`：选择空元素。
11. `:not(选择器)`：选择除了括号中选中的元素之外的其他元素。
12. `:focus`：选择获取焦点的`input`元素。
13. `:checked`：选择被选中的`input`元素。
14. `::selection`：选择被光标选中的元素部分。

3.4 伪对象选择器

伪对象也叫伪元素。伪对象用双冒号`::`表示，只写一个冒号也能被解析。

伪类用于定义元素状态；而伪对象则是改变文档结构，在结构外另加一个没有实际存在的元素（伪元素）。

常用伪对象选择器：

1. `::before`：在选中元素的内容前面插入其他内容，在`content`属性中设置要插入的内容。
2. `::after`：在选中元素的内容后面插入其他内容，在`content`属性中设置要插入的内容。

例如：

```
<style>
    /* 1.插入文字 */
    .box::before{
        content: "插入文字";
    }

    /* 2.插入图片 */
    .box::after{
```

```
        content: url("");
    }

    /* 3.自定义内容 */
    .box::before{
        content: ""; /* 默认为行内元素 */
        display: block; /* 转换为块级元素 */
        /* 设置其他属性 */
    }
</style>

<div class="box">内容</div>
```

4 CSS属性

4.1 字体属性

属性	描述	取值
`color`	文字颜色	颜色值
`font-size`	文字大小	像素数，单位为 `px`。 普通文字的默认大小为16px。
`font-weight`	文字的粗细	`normal`：标准字符 `bold`：粗体 `bolder`：更粗的字符 `lighter`：更细的字符 也可以取100到900之间的数字，其中400等同于`normal`，700等同于`bold`。
`font-style`	文字倾斜	`normal`：标准字符 `italic`：斜体
`font-family`	字体	可以设置多个值，多个值之间用逗号隔开。浏览器从左往右尝试，如果浏览器不支持第一个字体，则会尝试下一个，以此类推。如果所有字体都不支持，就使用浏览器的默认字体。 如果字体名称包含空格，必须加上引号。

颜色的取值：

1. 关键词。例如：`red`、`blue`、`green`等。
2. 6 位十六进制数，由`#`引导。例如：`#000000`（黑色）、`#ffffff`（白色）。
   当 6 位数字相同时，可以只写 3 位。例如，`#000000`可以简写为`#000`。
3. `rgb()`。RGB 色彩模式，有 3 个参数，分别代表红、绿、蓝，每个参数取值为 0-255 的整数。

4. `rgba()`。在`rgb()`的基础上增加第四个参数，表示透明度，取值为 0-1 之间的数字，0 表示完全透明，1 表示不透明。

字体属性具有继承性，如果元素没有设置样式，就会继承父元素的样式。

4.2 文本属性

属性	描述	取值
`text-align`	文本的水平对齐方式	`left`：默认值。左对齐 `right`：右对齐 `center`：水平居中
`text-decoration`	文本的修饰	`none`：没有修饰线 `underline`：下划线 `overline`：上划线 `line-through`：删除线
`text-transform`	英文大小写	`none`：默认样式 `capitalize`：每个单词以大写字母开头 `uppercase`：全部大写 `lowercase`：全部小写
`text-indent`	文本首行缩进	像素数。正值表示向右缩进，负值表示向左缩进。

文本属性具有继承性。

4.3 列表属性

属性	描述	取值
`list-style-type`	列表项标记的类型	`none`：无标记 `disc`：实心圆 `circle`：空心圆 `square`：实心方块 `decimal`：数字 `decimal-leading-zero`：0开头的数字（如01、02等） `lower-roman`：小写罗马数字 `upper-roman`：大写罗马数字 `lower-alpha`：小写英文字母 `upper-alpha`：大写英文字母 `lower-greek`：小写希腊字母 `lower-latin`：小写拉丁字母 `upper-latin`：大写拉丁字母
`list-style-image`	将列表项标记设置为图像	`list-style-image:url("");`
`list-style-position`	列表项标记相对于内容的位置	`inside`：列表项标记放在文本以内，且环绕文本根据标记对齐 `outside`：列表项标记放在文本以外，且环绕文本不根据标记对齐

可以使用`list-style`属性同时设置以上三个属性，多个值用空格隔开，不区分先后顺序。

4.4 表格属性

属性	描述	取值
`width`	表格的宽度	像素数
`height`	表格的高度	像素数
`border-spacing`	单元格之间的距离	像素数
`border-collapse`	边框折叠	`collapse`：边框折叠
`text-align`	单元格内容的水平对齐方式	`left`：默认值。左对齐 `right`：右对齐 `center`：水平居中
`vertical-align`	单元格内容的垂直对齐方式	`top`：垂直居上 `middle`：垂直居中 `bottom`：垂直居下 只能对`tr`和`td`标签设置，对`table`标签无效。

4.5 背景属性

属性	描述	取值
`background-color`	背景颜色	`transparent`：默认值。背景透明 还可以取颜色值
`background-image`	背景图片	`url("")`
`background-attachment`	设置背景图片是随内容滚动还是固定	`scroll`：默认值。滚动 `fixed`：固定 必须先设置`background-image`属性。
`background-position`	背景图片的位置	取值由两个值组成，第一个表示水平方向的位置，第二个表示垂直方向的位置，两个值之间用空格隔开。这两个值可以取像素数、百分数或关键词。 关键词：`left`水平居左，`right`水平居右，`top`垂直居上，`bottom`垂直居下，`center`居中。 在水平和垂直两个方向上只设置其中一个时，另一个方向为`center`。 必须先设置`background-image`属性。
`background-repeat`	设置背景图片如何铺排填充	`repeat`：默认值。平铺 `no-repeat`：不平铺 `repeat-x`：水平平铺 `repeat-y`：垂直平铺 必须先设置`background-image`属性。
`background-size`	背景图片大小	取值由两个值组成，第一个表示宽度，第二个表示高度，两个值之间用空格隔开。也可以只设置宽度，图片按比例缩放。 `cover`：覆盖整个背景区域，但背景图可能显示不全 `contain`：背景图显示完整，但可能无法覆盖整个背景区域 必须先设置`background-image`属性。

可以使用`background`简写属性设置以上属性，多个值用空格隔开，不区分先后顺序。没有设置的属性取默认值。

当背景图片的大小和位置取值为像素数或百分数时，一般不使用简写属性。

当简写属性与单个属性同时出现时，单个属性写在简写属性下面，避免被覆盖。

4.6 内容溢出

4.6.1 overflow

`overflow`属性指定如果内容溢出一个元素的框会发生什么。

取值	描述
`visible`	默认值。内容不会被修剪，溢出内容会呈现在元素框之外
`hidden`	内容会被修剪，并且溢出内容是不可见的。 开启 BFC 格式，变成独立的一块，子元素的变动对父元素没有影响。
`scroll`	内容会被修剪，但是浏览器会显示滚动条以便查看溢出内容
`auto`	如果内容被修剪，则浏览器会显示滚动条；如果未修剪，则不显示滚动条
`inherit`	从父元素继承 `overflow` 属性的值

4.6.2 white-space

`white-space` 属性定义元素内空白的处理方式。空白包括空格、换行符（回车）、制表符。

取值	描述
`normal`	默认值。忽略多余的空白，多个空白连在一起时看做一个空白。 不影响自动换行，一行文字放不下时自动换行。
`nowrap`	忽略多余的空白，多个空白连在一起时看做一个空白。 文本不会自动换行，只有遇到 `br` 标签才换行。
`pre`	保留空白，有几个空白就显示几个。文本不会自动换行，只有遇到 `br` 标签才换行。
`pre-wrap`	保留空白，有几个空白就显示几个。不影响自动换行，一行文字放不下时自动换行。
`pre-line`	忽略多余的空格，多个空格连在一起时看做一个空格。换行符正常处理，换行符后的内容从下一行开始。不影响自动换行。
`inherit`	继承父元素的 `white-space` 属性值

4.6.3 text-overflow

`text-overflow` 规定文本溢出时发生什么。

取值	描述
`clip`	修剪文本
`ellipsis`	显示省略号来代表被修剪的文本
字符串	用给定的字符串来代表被修剪的文本

`text-overflow` 属性必须配合 `overflow` 和 `white-space` 才能生效，并且 `overflow` 属性值必须为 `hidden`，`white-space` 属性值必须为 `nowrap`。例如，要设置单行文字溢出时显示省略号，可以使用如下代码：

```css
/* 设置文字单行显示 */
white-space: nowrap;
/* 溢出部分隐藏 */
overflow: hidden;
/* 设置文字溢出时的表现 */
text-overflow: ellipsis;
```

4.7 浮动

浮动的作用：布局，使块级元素水平显示。

浮动的原理：

1. 元素浮动后排除到普通文档流之外，在页面中不占据位置（原位置不保留），其他元素就可以占据浮动元素的位置。文字不会被浮动元素覆盖，而是环绕在浮动元素周围。
2. 浮动时碰到父元素的边框或其他浮动元素的边框就会停止，因此浮动不会重叠。
3. 元素浮动后转换为块级元素。
4. 元素浮动后，元素的默认宽度由内容撑开。
5. 元素浮动后，开启 BFC 格式。

`float`属性用于设置浮动，取值包括`left`、`right`、`none`，默认值为`none`。

当父元素没有设置高度时，默认高度由内容撑开。此时当子元素浮动后，子元素不占据位置，父元素的高度无法撑开，发生高度坍塌，对后面的元素产生影响。清除这种影响的方法有：

1. 给父元素设置高度。（要求父元素高度已知）
2. 对父元素设置`overflow: hidden;`。（父元素自动找高）
3. 对受影响的元素设置`clear`属性，清除浮动的影响。取值包括`left`、`right`、`both`。（父元素的高度无法恢复）
4. 空`div`法：在浮动元素后面加一个空`div`，设置空`div`的`clear`属性。
5. 伪对象法：用伪对象选择器写空`div`。

4.8 定位

`position`属性指定了元素的定位类型。

取值	描述	应用场景
`static`	默认值。没有定位，元素出现在正常的流中，忽略`left`、`right`、`top`、`bottom`、`z-index`属性。	
`relative`	相对定位，相对于其正常位置进行定位，定位后原位置保留。 元素的位置通过`left`、`right`、`top`、`bottom`属性进行设置。	1、配合`absolute`使用。 2、元素小范围移动。
`absolute`	绝对定位，相对于`static`以外的第一个父元素进行定位，原位置不保留。 如果父元素没有定位，则逐级往上找，最后相对于body定位。 元素的位置通过`left`、`right`、`top`、`bottom`属性进行设置。	形成独立的一层
`fixed`	固定定位，相对于浏览器窗口进行定位，原位置不保留。 元素的位置通过`left`、`right`、`top`、`bottom`属性进行设置。	固定在页面某个位置
`sticky`	CSS3新增属性。`fixed`与`relative`的结合。 当目标区域在屏幕中可见时，表现为`relative`；当页面滚动超出目标位置时，表现为`fixed`。	CSDN导航条

`left`和`right`属性同时出现时，`left`优先；`top`和`bottom`属性同时出现时，`top`优先。

`z-index`属性设置堆叠顺序，配合元素定位使用。在元素定位过程中，可能出现元素重叠（比如`relative`、`absolute`、`fixed`），此时需要设置元素的堆叠顺序。`z-index`属性的默认值为`auto`，表示与父元素相同。取值为数字，取值越大，层级越高，可以取负值。默认情况下，当两个元素同时定位时，写在后面的元素层级较高。

4.9 其他属性

4.9.1 行高

`line-height`属性设置文字的行高。取值可以是像素数或数字，当取值为数字时，行高=数字×字体大小。

要想设置单行文字垂直居中，可以设置`line-height`属性等于容器的高度。例如：

```
.box {
    height: 50px;
    line-height: 50px;
}
```

4.9.2 透明度

`opacity`属性设置元素的透明度。取值为 0-1 之间的数字，0 表示完全透明，1 表示不透明。当取值为 0-1 之间的小数时，小数点之前的 0 可以省略。

4.9.3 字符间距

`letter-spacing` 属性设置字符间距。取值为像素数，可正可负。

4.9.4 display

`display` 属性设置元素生成的框的类型。

取值	描述
`none`	此元素不显示，原位置不保留
`block`	此元素显示为块级元素，元素前后带有换行符
`inline`	此元素显示为行内元素，元素前后没有换行符
`inline-block`	行内块元素。表现为行内元素，但可以设置宽高；行内块元素可以识别代码中的空白。 常见的行内块元素：`img`、`input`、`button`、`video` 等。
`table`	此元素显示为表格
`table-cell`	此元素显示为表格单元格
`flex`	弹性盒模型

块级元素和行内元素的区别：

1. 块级元素独占一行，行内元素在同一行显示。
2. 块级元素默认宽度为 100%，行内元素默认宽度由内容撑开。
3. 块级元素可以设置宽高，行内元素设置宽高不生效。
4. 块级元素可以设置 `margin` 和 `padding` 的四周；行内元素只能设置 `margin` 和 `padding` 的左右，上下不生效。
5. 块级元素可以包含块级元素和行内元素，行内元素一般不包含块级元素。
6. 块级元素 `display` 属性默认值为 `block`，行内元素 `display` 属性默认值为 `inline`。

块级元素：`div`、`p`、`h1-h6`、`hr`、`body`、`html`、`ul`、`ol`、`li`、`dl`、`dt`、`dd`、`form`、`header`、`nav`、`footer`、`aside`、`article`、`section` 等。

行内元素：`span`、`img`、`a`、`i`、`b`、`u`、`s`、`del`、`sup`、`sub`、`em`、`strong`、`input`、`button`、`textarea`、`video` 等。

4.9.5 鼠标光标

`cursor` 属性设置鼠标指针放在元素上时光标的形状。

取值	描述
`default`	默认样式
`pointer`	指示链接的指针（手形）
`url("")`	图片
`crosshair`	十字线
`text`	指示文本
`wait`	指示程序正忙（通常是一只表或沙漏）
`help`	指示可用的帮助（通常是问号或气球）
`move`	指示元素可被移动
`e-resize`	指示矩形框的边缘可以向右（东）移动
`ne-resize`	指示矩形框的边缘可以向右上（东北）移动
`nw-resize`	指示矩形框的边缘可以向左上（西北）移动
`n-resize`	指示矩形框的边缘可以向上（北）移动
`se-resize`	指示矩形框的边缘可以向右下（东南）移动
`sw-resize`	指示矩形框的边缘可以向坐下（西南）移动
`s-resize`	指示矩形框的边缘可以向下（南）移动
`w-resize`	指示矩形框的边缘可以向左（西）移动

4.10 CSS3新增属性

4.10.1 圆角

`border-radius`属性设置圆角，取值为像素数或百分数，取值越大弧度越大。

`border-radius`属性可以设置 1-4 个值：取 1 个值表示四个角；取 2 个值，第 1 个表示左上角、右下角，第 2 个表示右上角、左下角；取 3 个值，第 1 个表示左上角，第 2 个表示右上角、左下角，第 3 个表示右下角；取 4 个值，从左到右依次表示左上角、右上角、右下角、左下角。

画圆：宽度和高度相等，设置`border-radius: 50%;`。

4.10.2 阴影

`box-shadow`属性向框添加阴影，`text-shadow`属性向文字添加阴影。

语法：

```
box-shadow: h-shadow v-shadow blur spread color position;
text-shadow: h-shadow v-shadow blur color;
```

取值	描述
h-shadow	必需。水平阴影的位置，可正可负。正值表示右侧，负值表示左侧。
v-shadow	必需。垂直阴影的位置，可正可负。正值表示下方，负值表示上方。
blur	可选。模糊距离，取正值。
spread	可选。阴影的尺寸，可正可负。
color	可选。阴影的颜色，默认为黑色。
position	可选。设置阴影位置，取值`outset`表示外部阴影，`inset`表示内部阴影。默认值为`outset`。

4.10.3 背景渐变

CSS3 定义了两种类型的渐变：

1. 线性渐变：沿直线方向渐变。
2. 径向渐变（射线渐变）：从中心向四周渐变。

线性渐变语法：`background-image: linear-gradient(direction, color1, color2, ...);`

参数说明：

1. 方向：写在颜色前面，可以省略。有两种取值：
   （1）关键词：`to 方向`。例如：`to right`表示向右，`to left`表示向左，`to bottom`表示向下，`to top`表示向上，`to right bottom`表示向右下，等等。默认值为向下。
   （2）角度：单位为`deg`，`0deg`表示向上，取正值表示顺时针旋转，取负值表示逆时针旋转。例如：`90deg`表示向右，`180deg`表示向下，`-90deg`表示向左。
2. 颜色：至少有 2 个，可以有多个。

径向渐变语法：`background-image: radial-gradient(center, shape, color1, color2, ...);`

参数说明：

1. 渐变中心：第一个参数，可以省略。默认在元素的正中心，取值为关键词，如`left`、`right`、`top`、`bottom`、`left top`等。
2. 形状：第二个参数，可以省略。默认值为椭圆，`circle`表示圆形。
3. 颜色：至少有 2 个，可以有多个。

4.10.4 转换

转换的效果是改变元素的形状、大小和位置。

`transform`属性对元素施加 2D 或 3D 转换。

属性值	效果	语法
`translate()`	位移	`transform: translate(x, y);`（2D位移，只有一个参数时表示水平位移） `transform: translateX(x);`（水平位移） `transform: translateY(y);`（垂直位移） `transform: translate3D(x, y, z);`（3D位移） `transform: translateZ(z);`（z轴位移） 参数取值为像素数或百分数，可正可负。
`rotate()`	旋转	`transform: rotate(r);`（绕中心点旋转） `transform: rotateX(x);`（绕X轴旋转） `transform: rotateY(y);`（绕Y轴旋转） 参数取值为角度，取正值时顺时针旋转，取负值时逆时针旋转。 注意：元素旋转时，坐标轴也会旋转。当位移和旋转同时使用时，推荐先位移后旋转。
`scale()`	缩放	`transform: scale(x, y);`（只有一个参数时，另一个方向等比例缩放） `transform: scaleX(x);`（水平缩放） `transform: scaleY(y);`（垂直缩放） `transform: scale3D(x, y, z);`（3D缩放） `transform: scaleZ(z);`（z轴缩放） 参数取值为数字，0-1表示缩小，大于1表示放大，默认值为1。
`skew()`	倾斜	`transform: skew(x, y);`（只有一个参数时表示水平倾斜） `transform: skewX(x);`（水平倾斜） `transform: skewY(y);`（垂直倾斜） 参数取值为角度，单位为`deg`。

4.10.5 过渡

过渡是从一种样式逐渐改变为另一种样式的效果。过渡必须有触发事件，如鼠标悬停、鼠标点击等。

`transition`属性设置过渡，它是一个复合属性，包括`transition-property`、`transition-duration`、`transition-timing-function`、`transition-delay`四个子属性，通过这四个子属性的配合来完成一个完整的过渡效果。

属性	描述	语法
`transition-property`	参与过渡的属性	`transition-property: 属性1, 属性2, ...;` 必需。取值`all`表示所有属性。 可以过渡的属性：取值为数值（如宽度），取值为颜色（如背景颜色），转换（`transform`），阴影（`box-shadow`、`text-shadow`）。
`transition-duration`	持续时间	`transition-duration: 时间1, 时间2, ...;` 必需。默认值为`0s`或`0ms`。每个属性对应一个时间。
`transition-timing-function`	速度变化类型	可选。 `ease`：默认值。先加速后减速 `ease-in`：加速 `ease-out`：减速 `ease-in-out`：先加速后减速 `linear`：匀速
`transition-delay`	延迟时间	可选。默认值为`0s`或`0ms`。 可以取负值，表示跳过这段时间的效果。

一般在`transition`简写属性中同时设置这四个子属性。

一般将过渡写在元素中，这样可以在鼠标悬停和移开时都产生过渡效果。

4.10.6 动画

动画是使元素从一种样式逐渐改变为另一种样式的效果。与过渡不同，动画可以改变任意数量的样式，可以控制动画的执行次数，可以没有触发事件。

使用`@keyframes`定义动画，在`@keyframes`后面指定动画名称，如下所示：

```
@keyframes name {

}
```

在`@keyframes`的大括号中，使用`0%`或`from`定义开始状态，`100%`或`to`定义结束状态，二者之间可以用百分数定义中间状态。例如：

```
@keyframes name {
    0% {

    }
    50% {

    }
    100% {

    }
}
```

动画定义完成后，可以在需要的地方调用动画。使用`animation`属性调用动画，这是一个复合属性，有若干子属性，通过这些子属性的配合完成完整的动画。

属性	描述	语法
`animation-name`	动画名称	必需。指定要调用的动画的名称。
`animation-duration`	持续时间	必需。类似于`transition-duration`。
`animation-timing-function`	速度变化类型	可选。类似于`transition-timing-function`。
`animation-delay`	延迟时间	可选。类似于`transition-delay`。
`animation-iteration-time`	播放次数	可选。取值为数字。取值`infinite`表示无限循环播放。
`animation-direction`	播放方向	可选。`alternate`表示偶数次播放时倒序播放。
`animation-fill-mode`	停在最后一帧	可选。`forwards`表示停在最后一帧。
`animation-play-state`	动画播放状态	可选。`running`表示播放，`paused`表示暂停。一般不简写，配合触发事件使用。

4.10.7 字体图标

字体图标本质上是文字，其样式可以使用字体属性来修改。

阿里巴巴矢量图标库提供了大量字体图标，可以根据需要下载使用。

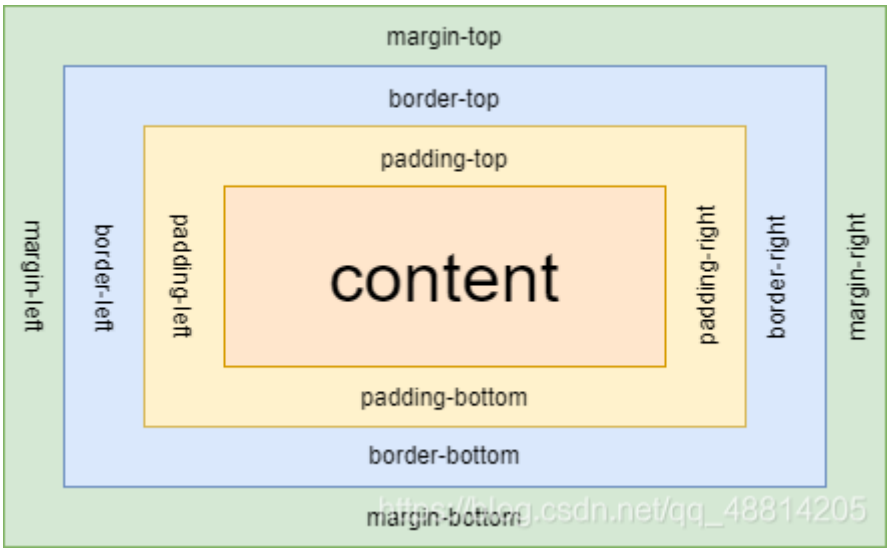
阿里巴巴矢量图标库使用方法：

1. 搜索需要的图标，加入购物车。
2. 打开购物车，选择“下载代码”。（需要登录）
3. 将下载到的压缩包解压。
4. 将`iconfront.css`文件引入项目中。
5. 根据`demo_index.html`文件中的提示，在代码中使用 Unicode 编码引入图标。

5 盒模型

5.1 标准盒模型

所有 HTML 元素都可以看做盒子，标准盒模型如下图所示：



盒模型包括四部分：content（内容）、padding（填充）、border（边框）、margin（外边距）。

5.1.1 content

content 的属性有：

1. `width`：元素的宽度
2. `height`：元素的高度
3. `min-width`：最小宽度
4. `max-width`：最大宽度
5. `min-height`：最小高度
6. `max-height`：最大宽度

5.1.2 border

border 的属性有：

1. `border-style`：边框的样式。必需属性。
   取值：`solid` 表示实线，`dashed` 表示虚线，`dotted` 表示点线，`double` 表示双实线，`none` 表示无边框。
2. `border-width`：边框宽度。默认值为3px。
3. `border-color`：边框的颜色。默认值为黑色。
4. `border`：简写属性，可以同时设置边框的样式、宽度和颜色。
5. `border-top`：上边框，可以同时设置边框的样式、宽度和颜色。
6. `border-bottom`：下边框，可以同时设置边框的样式、宽度和颜色。
7. `border-left`：左边框，可以同时设置边框的样式、宽度和颜色。
8. `border-right`：右边框，可以同时设置边框的样式、宽度和颜色。

利用边框之间的连接处，可以绘制三角形和梯形。例如：

```
<style>
/* 向下的三角形 */
.box1 {
    width: 0;
    height: 0;
    border-top: solid 10px red;
    border-left: solid 10px transparent;
    border-right: solid i0px transparent;
```

```
}

/* 向右的梯形 */
.box2 {
    width: 0;
    height: 10px;
    border-left: solid 10px green;
    border-top: solid 10px transparent;
    border-bottom: solid 10px transparent;
}
</style>

<div class="box1"></div>
<div class="box2"></div>
```

5.1.3 padding

`padding` 属性可以设置 1-4 个值：取 1 个值表示四周；取 2 个值，从左到右分别表示上下、左右；取 3 个值，从左到右分别表示上、左右、下；取 4 个值，从左到右分别表示上、右、下、左。

除此之外，还可以分别设置 4 个方向的内边距：`padding-top`、`padding-bottom`、`padding-left`、`padding-right`。

`padding` 属性不能取负值或 `auto`。

5.1.4 margin

`margin` 属性可以设置 1-4 个值：取 1 个值表示四周；取 2 个值，从左到右分别表示上下、左右；取 3 个值，从左到右分别表示上、左右、下；取 4 个值，从左到右分别表示上、右、下、左。

除此之外，还可以分别设置 4 个方向的外边距：`margin-top`、`margin-bottom`、`margin-left`、`margin-right`。

`margin` 属性可正可负，也可以取 `auto`。

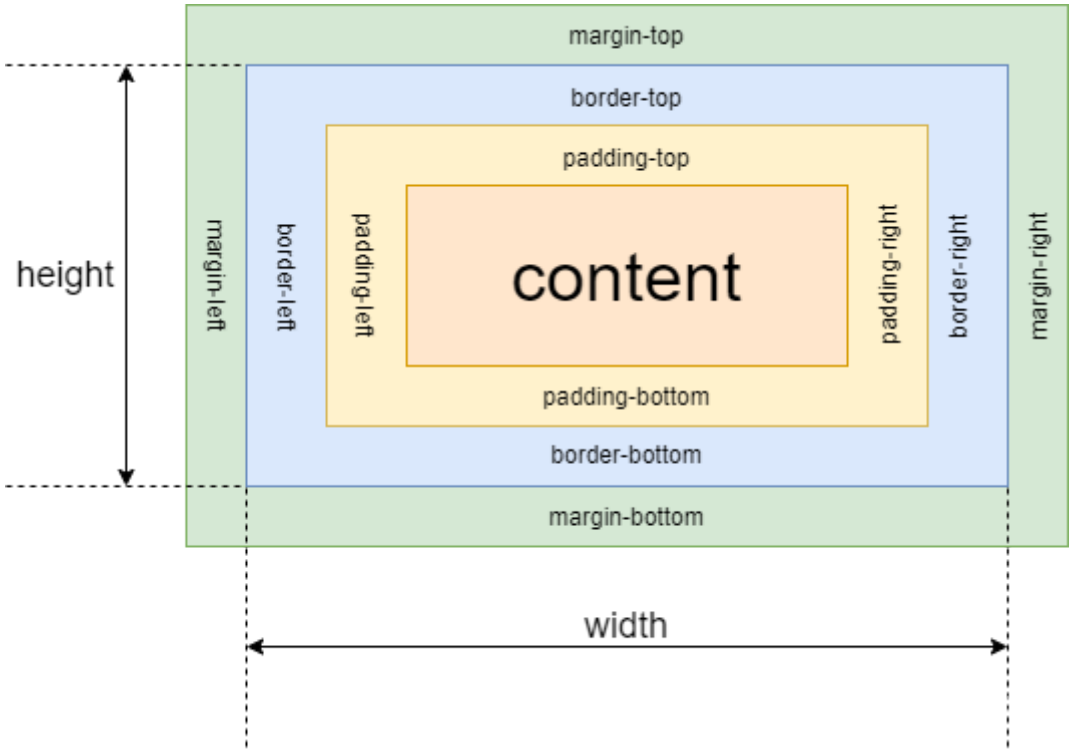
块级元素水平居中：`margin: 0 auto;`

注意：

1. 垂直方向上外边距合并。当垂直方向上外边距相撞时，取较大值。（浮动元素不合并）
2. 第一个子元素设置 `margin-top` 后，父元素会跟着一起向下移动。
   解决方法：
   （1）父元素设置 `overflow: hidden;`。
   （2）父元素或子元素加浮动。
   （3）父元素设置 `border: 1px solid transparent;`。
   （4）父元素设置 `padding-top: 1px;`。
3. margin 区域是透明的。元素的背景颜色覆盖 content、padding、border 三部分，不能覆盖到 margin。
4. 元素的实际大小并不等于 `width` 和 `height` 的值。
   实际宽度= `width` + `padding-left` + `padding-right` + `border-left` + `border-right` + `margin-left` + `margin-right`
   实际高度= `height` + `padding-top` + `padding-bottom` + `border-top` + `border-bottom` + `margin-top` + `margin-bottom`

5.2 怪异盒模型

怪异盒模型也包括 content、padding、border、margin 四部分。与标准盒模型不同的是，怪异盒模型中 content 部分包含了 padding 和 border。也就是说，`width` 属性不是 content 的宽度，而是 content、padding、border 宽度的总和，`height` 属性是 content、padding、border 高度的总和。



怪异盒模型元素的实际宽度=`width`+`margin-left`+`margin-right`
怪异盒模型元素的实际高度=`height`+`margin-top`+`margin-bottom`

`box-sizing` 属性用于设置盒模型的类型。`content-box` 表示标准盒模型，`border-box` 表示怪异盒模型。默认值为 `content-box`。

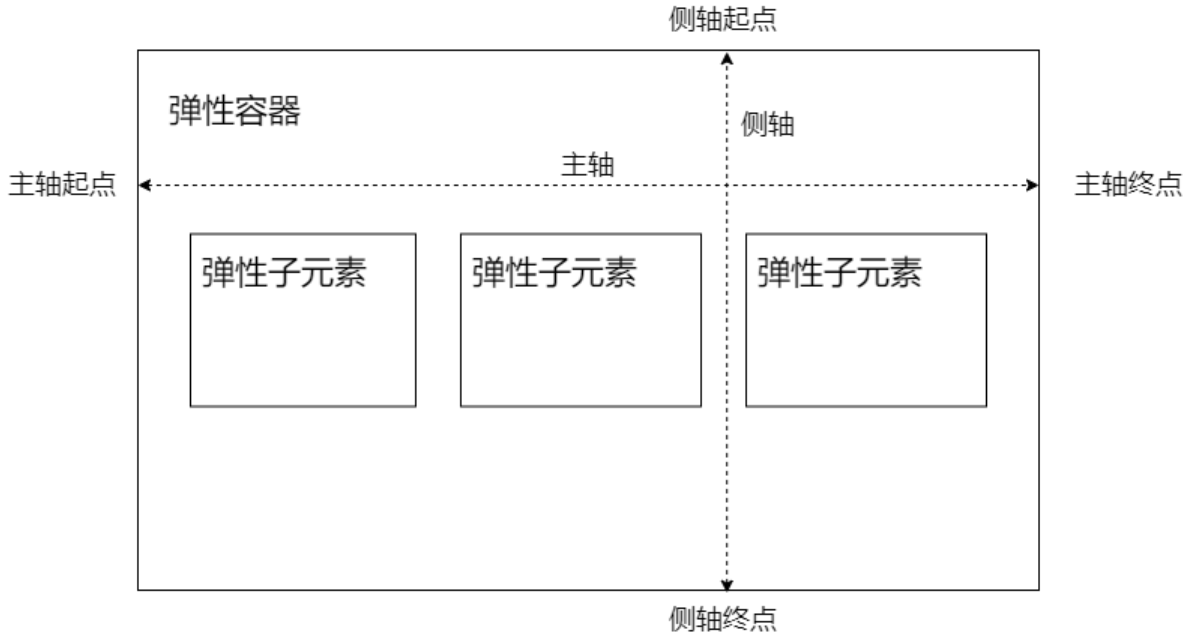
5.3 弹性盒模型

弹性盒模型是 CSS3 新增的布局模式。弹性盒模型是一种当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。引入弹性盒模型的目的是提供一种更加有效的方式来对一个容器中的子元素进行排列、对齐和分配空白空间。

弹性盒由弹性容器和弹性子元素组成。

弹性容器内包含一个或多个弹性子元素。弹性子元素默认在弹性容器内水平显示，默认每个弹性容器只有一行。

父元素开启弹性盒模型后，子元素的宽度默认由内容撑开。



弹性容器外和弹性子元素内是正常渲染的，弹性盒只定义了弹性子元素如何在弹性容器内布局。

父元素上的属性：

属性	描述	取值
`display`	设置元素框的类型	取值为 `flex` 时开启弹性盒模型
`flex-direction`	设置子元素排列方向	`row`：默认值。子元素水平排列 `column`：子元素垂直排列 `row-reverse`：子元素水平倒序排列 `column-reverse`：子元素垂直倒序排列
`justify-content`	子元素在主轴方向的对齐方式	`flex-start`：默认值。位于主轴起点 `flex-end`：位于主轴终点 `center`：居中 `space-around`：在子元素四周分配父元素剩余的空间，两端的距离是中间的一半 `space-between`：在子元素中间分配父元素剩余的空间
`align-items`	子元素在侧轴方向的对齐方式	`flex-start`：默认值。位于侧轴起点 `flex-end`：位于侧轴终点 `center`：居中

子元素上的属性：

属性	描述	取值
`flex-grow`	子元素分配父元素剩余空间的比例	数字

6 兼容性问题

6.1 css hack

css hack 解决 IE 低版本的兼容性问题。

6.1.1 条件注释法

语法：

```
<!--[if 条件]>
    HTML代码
<![endif]-->
```

在 HTML 文档中加入上述结构，满足条件时中间的内容生效，不满足条件则不生效。

条件	描述
`IE 版本号`	等于给定的 IE 版本号
`gt IE 版本号`	大于给定的 IE 版本号
`gte IE 版本号`	大于等于给定的 IE 版本号
`lt IE 版本号`	小于给定的 IE 版本号
`lte IE 版本号`	小于等于给定的 IE 版本号

6.1.2 属性前缀和后缀

前缀：`+`、`-`、`*`、`#`、`_`

后缀：`\0`、`\9`、`\9\0`、`!important`

6.2 厂商前缀

厂商前缀解决浏览器对 CSS3 新特性的支持问题。

浏览器	厂商前缀	内核
谷歌	`-webkit-`	blink
火狐	`-moz-`	gecko
IE	`-ms-`	trident
Safari	`-webkit-`	webkit
Opera	`-o-`	blink

在 CSS3 新增内容之前加厂商前缀，获得对特定浏览器的支持。

7 响应式布局

响应式布局（responsive design），意在实现不同屏幕分辨率的终端上浏览网页的不同展示方式。通过响应式布局能使网站在手机和平板电脑上有更好的浏览阅读体验。

优点：

1. 面对不同分辨率设备灵活性强。

2. 能够快捷解决多设备显示适应问题。

缺点：

1. 兼容各种设备工作量大，效率低下。
2. 代码累赘，会出现隐藏无用的元素，加载时间加长。

响应式布局的设计步骤：

1. 设置`meta`标签：

```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" />
```

这样设置的作用是：使用设备的宽度作为视图宽度，并禁止初始的缩放，以兼容移动设备。

2. 使用媒体查询或 bootstrap 框架实现响应式布局。

媒体查询有两种形式：内部引入、外部引入。

内部引入语法：在 CSS 文件中分别写不同终端的样式。

```
/* 移动端 */
@media screen and (max-width:768px) {}
/* pad端 */
@media screen and (min-width:768px) and (max-width:992px) {}
/* pc端 */
@media screen and (min-width:992px) {}
```

外部引入语法：在`link`标签中指定引入的 CSS 文件所对应的终端。

```
<!-- 移动端 -->
<link rel="stylessheet" href="" media="screen and (max-width:768px)" />
<!-- pad端 -->
<link rel="stylessheet" href="" media="screen and (min-width:768px) and (max-width:992px)" />
<!-- pc端 -->
<link rel="stylessheet" href="" media="screen and (min-width:992px)" />
```

8 雪碧图

CSS Sprite 又称 CSS 精灵、CSS 雪碧图，是一种网页图片应用处理方式。它允许你将一个页面涉及的所有零星图片都包含到一张大图中，这样当访问该页面时，由多次请求变为一次请求。

雪碧图的优点：

1. 减少图片的大小。
2. 减少网页的 http 请求，从而大大提高页面的性能。
3. 解决了图片命名的困扰，只需对一张大图命名即可，不需要对每一个小元素进行命名。

雪碧图的使用方法：

1. 需要一个设置好宽高的容器。
2. 通过`background-image`属性引入雪碧图。
3. 通过`background-position`属性调整位置。