

# CSS3入门到进阶 css3基本文法

- CSS概述 CSS概要
- CSS基本语法 CSS3基本文法



1

**CSS概述** [css概要](#)

2

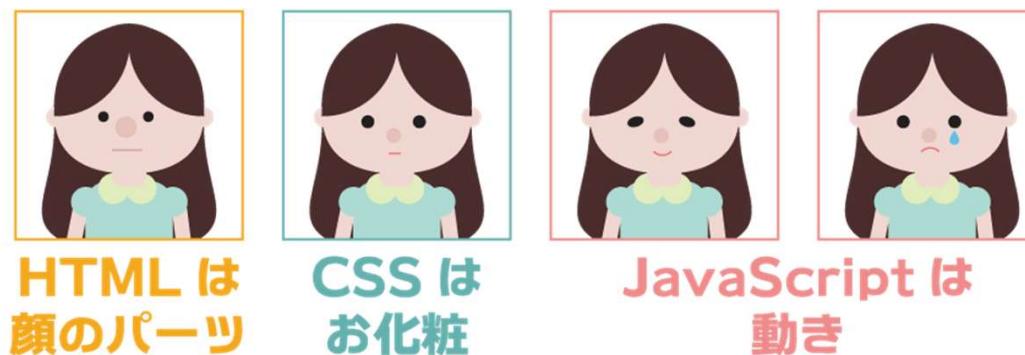
**CSS基本语法** [css3基本文法](#)

3

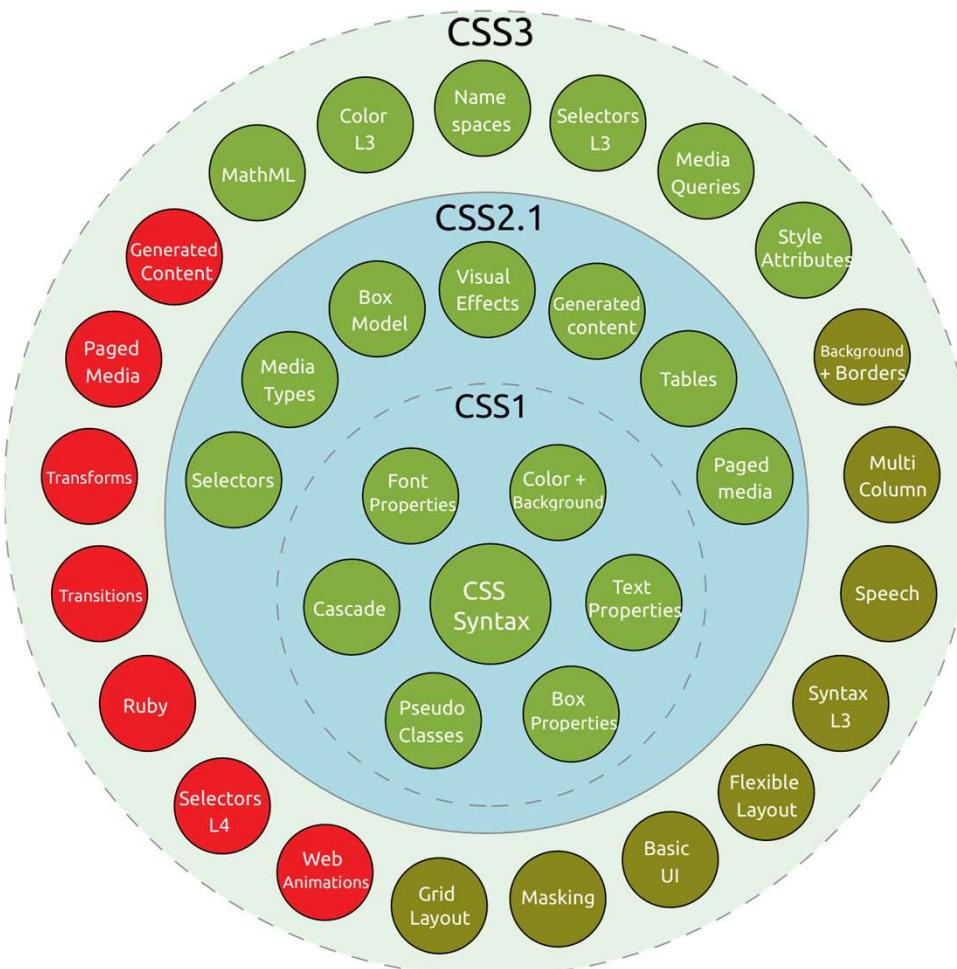
**CSS基本语法** [css3基本文法](#)

# CSS概述

- CSS全称Cascading Style Sheets（层叠样式表），是一种描述HTML文档样式的语言。
- CSS指定了在浏览器[ウェブブラウザ]上如何显示HTML元素。
- 代码可以单独放在.css文件作为link引入到html里，也可以直接在html里写css代码，还可以同时使用两种方式。
- 单独的css代码不能运行，化妆品离开了人就没有意义了。



# CSS历代版本



我们学习CSS3~

# CSS参考链接（之后可能会用到）

- [CSS属性参考](#)
- [CSS选择器参考](#)
- [CSS默认值](#)
- [CSS颜色参考](#)

# CSS节省了大量工作!

- CSS代码通常保存在外部.css文件中。
- 因为可以多个HTML文件通过link标签链接同一个外部.css文件，只需更改一个.css文件即可更改整个网站的外观。

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# Q & A

You Have Questions  
We Have Answers

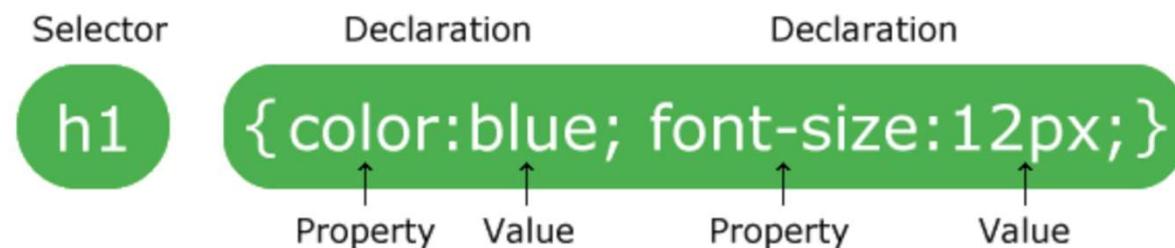


1 **CSS概述** [CSS概要](#)

2 **CSS基本语法** [CSS3基本文法](#)

# CSS语法

- CSS规则集由一个选择器[セレクタ] (selector) 和一个声明块[宣言ブロック] ({} ) 组成：



- 选择器[セレクタ]指向你要设置样式的HTML元素。
- 声明块包含一个或多个用分号（;）分隔的声明[宣言](Declaration)。
- 每个声明都包含一个CSS属性名称和一个属性值，用冒号（:）分隔。
- 多个CSS声明用分号分隔，声明块用花括号括起来。

# CSS语法范例

- 在此示例中，所有

元素都将居中对齐，并带有红色文本颜色：

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

- 示例说明：

- p是CSS中的选择器[セレクタ]（它指向要设置样式的HTML元素：

）。
- color是一个属性，red是属性值。
- text-align是一个属性，并且center是属性值。

# CSS 选择器[セレクタ]

- CSS选择器[セレクタ]用于“查找”（或选择）要设置样式的HTML元素。
- 我们可以将CSS选择器[セレクタ]分为五类：
  - 简单选择器（根据name, id和class选择元素）
  - 组合器选择器（根据它们之间的特定关系选择元素）
  - 伪类选择器（根据特定状态选择元素）
  - 伪元素选择器（选择元素的一部分并设置其样式）
  - 属性选择器（根据属性或属性值选择元素）
- 接下来将讲解最基本的CSS选择器[セレクタ]。

# CSS元素选择器

- 元素选择器根据元素名称选择HTML元素。
- 下面的CSS会让页面上的所有

元素都居中对齐，并带有红色文本颜色：

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

# CSS id选择器

- id选择器使用HTML元素的id属性来选择特定元素。
- 元素的id在页面中是唯一的，因此id选择器用于选择一个唯一的元素。
- 要选择具有特定id的元素，写一个井号（#），后面跟该元素的id。
- 下面的CSS代码将修饰id为“para1”的HTML元素：

```
#para1 {  
    text-align: center;  
    color: red;  
}
```

# CSS class选择器

- class选择器选择具有特定class属性的HTML元素。
- 要选择具有特定类的元素，写一个点（.），后跟类名。
- 在此示例中，所有带有class =“center”的HTML元素将为红色且居中对齐：

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

# CSS class选择器

- 还可以指定特定的HTML元素受class影响。
- 在此示例中，只有具有class =“center”的

元素将居中对齐：

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

- HTML元素也可以引用多个类。
- 在此示例中，

元素将根据class =“center”和class =“large”进行样式设置：

```
<p class="center large">This paragraph refers to two classes.</p>
```

注意：类名不能以数字开头！

尝试代码：[class\\_selector.html](#)

# CSS通用选择器

- 通用选择器 (\*) 选择页面上的所有HTML元素。
- 下面的CSS规则将影响页面上的每个HTML元素：

```
* {  
    text-align: center;  
    color: blue;  
}
```

# CSS分组选择器

- 分组选择器选择所有具有相同样式定义的HTML元素。
- 查看以下CSS代码（h1, h2和p元素具有相同的样式定义）：
- 尽量对选择器[セレクタ]进行分组，以最大程度地减少代码。
- 要对选择器进行分组，请用逗号分隔每个选择器。

```
h1 {  
    text-align: center;  
    color: red;  
}  
  
h2 {  
    text-align: center;  
    color: red;  
}  
  
p {  
    text-align: center;  
    color: red;  
}
```



```
h1, h2, p {  
    text-align: center;  
    color: red;  
}
```

# 如何添加CSS

- 浏览器[ウェブブラウザ]加载样式表时，将根据样式表中的信息来修饰HTML文档。
- 有三种插入样式表的方法：
  - 外部CSS[外部ファイル]
  - 内部CSS[styleタグ]
  - 内联CSS[INLINE]

# 外部CSS

- 使用外部样式表，只需更改一个文件即可更改整个网站的外观。
- 每个HTML页面的头部必须在<link>元素内包含对外部样式表文件的引用。
- 外部样式在HTML页面<head>部分内的<link>元素内定义：

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

    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>

  </body>
</html>
```

# 外部CSS

- 外部样式表可以在任何文本编辑器中编写，并且必须以.css扩展名保存。
- 外部 .css文件不应包含任何HTML标记。
- 这是“mystyle.css”文件的样子：

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: navy;  
    margin-left: 20px;  
}
```

尝试代码：external\_css

注意：请勿在属性值和单位之间添加空格（例如margin-left: 20 px;）。正确的写法是：margin-left: 20px;

# 内部CSS

- 如果一个HTML页面具有唯一样式，则可以使用内部样式表
  - 内部样式是在head部分的<style>元素内定义的：

尝试代码：[internal\\_css.html](#)

如果在不同的样式表中为同一选择器[セレクタ]（元素）定义了一些属性，则将使用最后读取的样式表中的值。

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

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

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

# 内联[インライン]CSS

- 内联[インライン]样式可用于为单个元素作为唯一样式。
- 要使用内联[インライン]样式，请将style属性添加到相关标签中。样式属性可以包含任何CSS属性。

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

- 提示：内联[インライン]样式失去了样式表的许多优点（通过将内容与表示混合在一起）。谨慎使用此方法。

## 级联顺序[スタイルの優先順位]

- 为HTML元素指定了多个样式时，将使用哪种样式？
- 页面中的所有样式都将按照以下规则“层叠”为新的“虚拟”样式表，其中第一优先级最高：
  - 1.内联[インライン]样式（在HTML元素内）
  - 2.外部和内部样式表（在头部）
  - 3.浏览器[ウェブブラウザ]默认
- 因此，内联[インライン]样式具有最高优先级，并将覆盖外部和内部样式以及浏览器[ウェブブラウザ]默认设置。

## CSS注释 [コメント]

- 注释用于解释代码，在你以后编辑源代码时可能会有所帮助
  - 浏览器[ウェブブラウザ]将忽略注释。
- CSS注释放置在<style>元素内，以/\*开头和\*/结尾：

```
/* This is a single-line comment */  
p {  
    color: red;  
}  
  
p {  
    color: red; /* Set text color to red */  
}
```

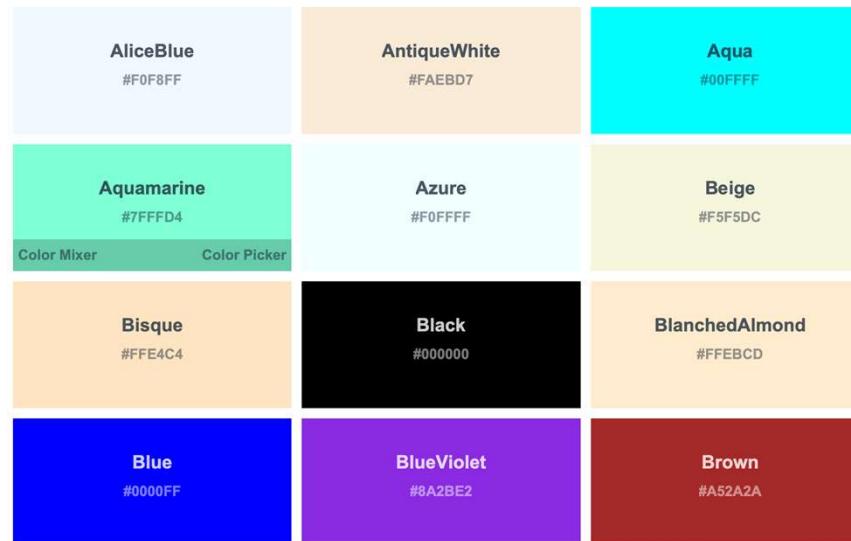
```
/* This is  
a multi-line  
comment */  
p {  
    color: red;  
}
```

# Q & A

You Have Questions  
We Have Answers

# CSS颜色名称

- 在CSS中，可以通过使用颜色名称来指定颜色：



- CSS / HTML支持140种标准颜色名称。

# CSS背景颜色

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

# CSS文字颜色

```
<h1 style="color:Tomato;">Hello World</h1>
<p style="color:DodgerBlue;">Lorem ipsum...</p>
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

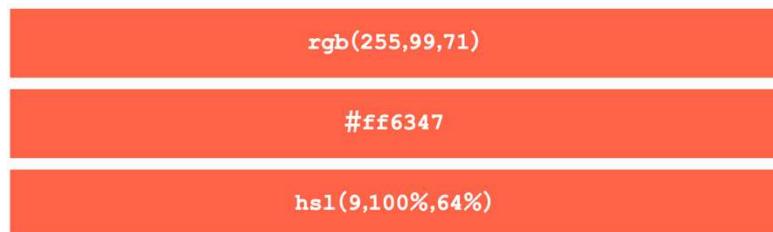
Hello World

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

## CSS颜色值[カラーコード]

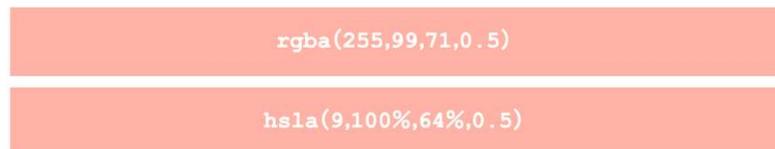
- 在CSS中，还可以使用RGB值，HEX值，HSL值，RGBA值和HSLA值来指定颜色：
- 与颜色名称“Tomato”相同：



```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
<h1 style="background-color:#ff6347;">...</h1>
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

- 与颜色名称“Tomato”相同，但透明度为50%：



# CSS 背景

- CSS背景属性用于定义元素的背景效果：
  - background-color属性指定元素的背景色。
  - opacity属性指定元素的不透明度/透明度。取值范围为0.0-1.0。值越低，越透明。
  - background-image属性指定要用作元素背景的图像。
  - background-attachment属性指定背景图像应该滚动还是固定的（不会与页面的其余部分一起滚动。
- 尝试代码：[background.html](#)

# CSS 边框

- CSS border属性指定元素边框的样式，宽度和颜色。

我四面都有边界。

我有一个红色的底边框。

---

我的边界是圆的。

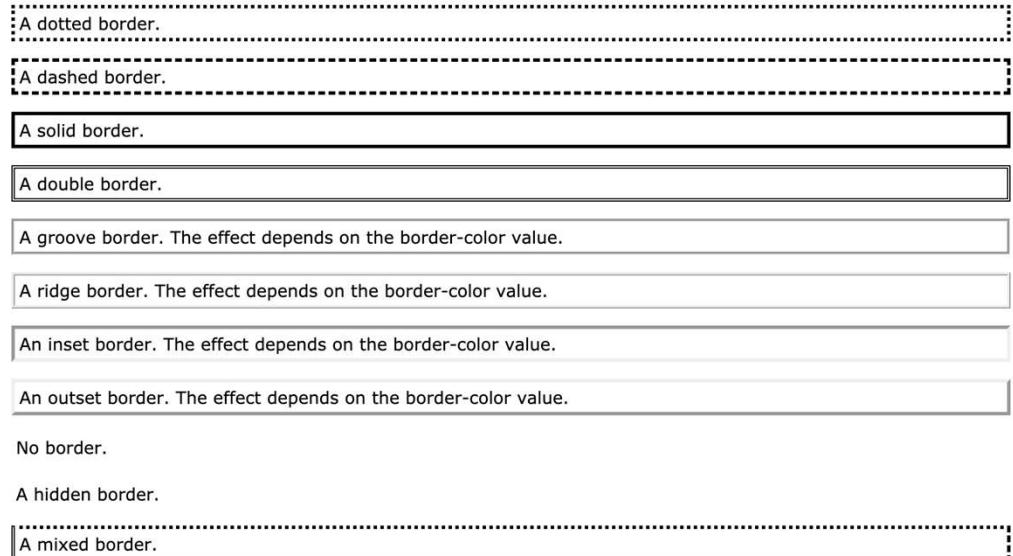
我有一个蓝色的左边框。

# CSS边框[ボーダー]样式

- border-style属性指定要显示的边框类型。
- 允许以下值：
  - dotted - 定义虚线边框
  - dashed - 定义虚线边框
  - solid - 定义实线边框
  - double - 定义一个双边框
  - groove - 定义3D沟槽边框。效果取决于边框颜色值
  - ridge - 定义3D棱形边框。效果取决于边框颜色值
  - inset - 定义3D嵌入边框。效果取决于边框颜色值
  - outset - 定义3D起始边框。效果取决于边框颜色值
  - none - 没有边界
  - hidden - 定义一个隐藏的边框
- border-style属性可以具有一到四个值（用于顶部边框，右侧边框，底部边框和左侧边框）。

# CSS边框样式

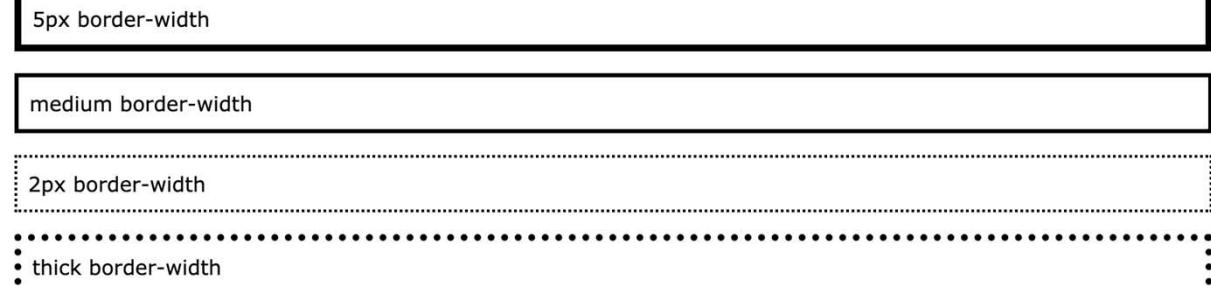
```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}
```



# CSS边框宽度

- border-width属性指定四个边框的宽度。
- 可以将宽度设置为特定大小（以px, pt, cm, em等为单位），也可以使用以下三个预定义值之一：薄，中或厚：

```
p.one {  
    border-style: solid;  
    border-width: 5px;  
}  
  
p.two {  
    border-style: solid;  
    border-width: medium;  
}  
  
p.three {  
    border-style: dotted;  
    border-width: 2px;  
}  
  
p.four {  
    border-style: dotted;  
    border-width: thick;  
}
```



# 特定边宽

border-width 属性可以有一到四个值（用于，右侧边框，底部和左侧边框）：

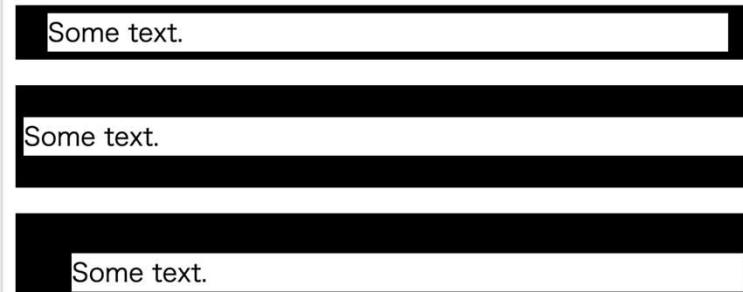
```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-width: 5px 20px; /* 5px top and bottom, 20px on
the sides */
}

p.two {
    border-style: solid;
    border-width: 20px 5px; /* 20px top and bottom, 5px on
the sides */
}

p.three {
    border-style: solid;
    border-width: 25px 10px 4px 35px; /* 25px top, 10px right,
4px bottom and 35px left */
}
</style>
</head>
<body>
```

## The border-width Property

The border-width property can have from one to four values (for the top border, right border, bottom border, and the left border):



Some text.

Some text.

Some text.

# CSS边框颜色

```
p.one {  
    border-style: solid;  
    border-color: red;  
}  
  
p.two {  
    border-style: solid;  
    border-color: green;  
}  
  
p.three {  
    border-style: dotted;  
    border-color: blue;  
}
```

Red border

Green border

Blue border

颜色也可以通过数值来指定

# CSS Box模型 [ボックス]

- 所有HTML元素都可以视为box。在CSS中，在谈论设计和布局[レイアウト]时会用到术语“box模型”。
- CSS的box模型实质上是一个包装每个HTML元素的box。它包括：margin, border, padding和实际内容。见下图：



# CSS Box模型 [ボックス]



尝试代码：box.html

- 说明：
  - content - 框的内容，其中显示文本和图像
  - padding - 清空内容周围的区域。padding是透明的
  - border - 围绕padding和内容的边框
  - margin - 清空边界外的区域。margin是透明的
- box模型允许我们在元素周围添加边框，并定义元素之间的空间。

# CSS边框-速记[一括指定]属性

- 为了缩短代码，也可以在一个属性中指定所有单个边框属性。
- border属性是以下各个边框属性的简写属性：
  - border-width
  - border-style (需要)
  - border-color

```
p {  
    border: 5px solid red;  
}
```

Some text

```
p {  
    border-bottom: 6px solid red;  
    background-color: lightgrey;  
}
```

Some text

# CSS圆形边框

- border-radius属性用于向元素添加圆角边框：

正常边框

圆形边框

圆角边框

最圆的边框

- 尝试代码：border.html

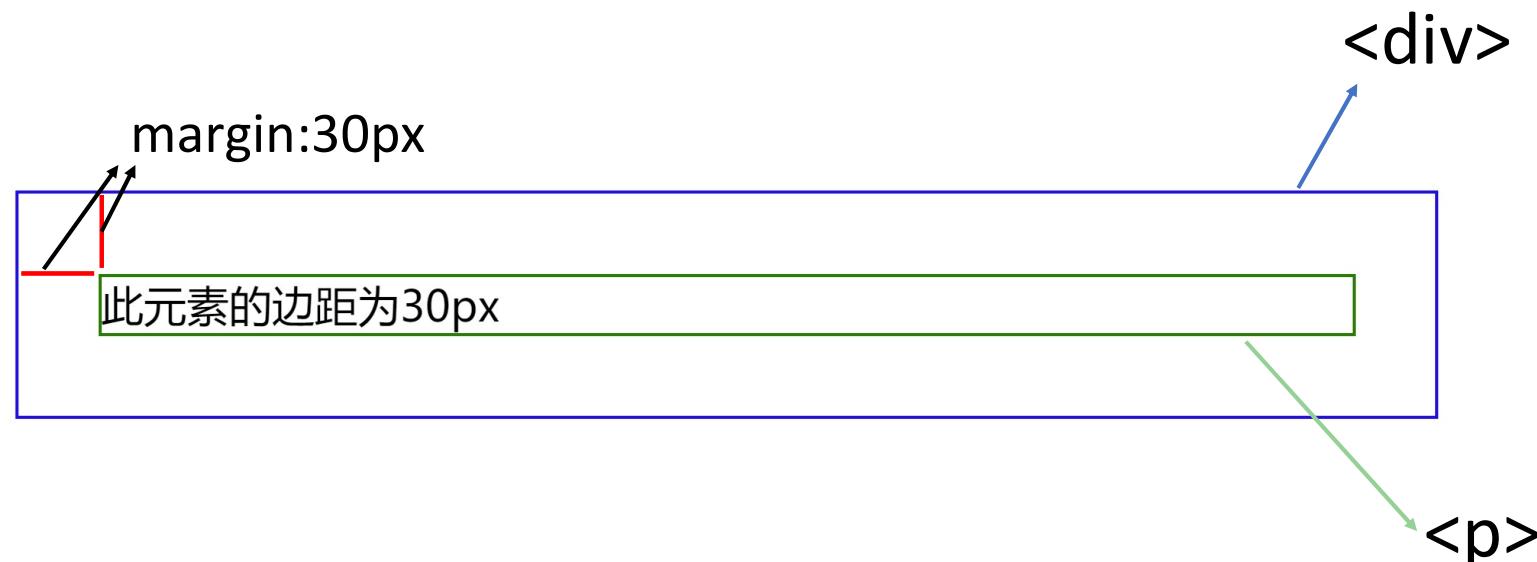
# CSS边框属性汇总

Property	Description	Property	Description
<u>border</u>	Sets all the border properties in one declaration	<u>border-left-width</u>	Sets the width of the left border
<u>border-bottom</u>	Sets all the bottom border properties in one declaration	<u>border-radius</u>	Sets all the four border-* -radius properties for rounded corners
<u>border-bottom-color</u>	Sets the color of the bottom border	<u>border-right</u>	Sets all the right border properties in one declaration
<u>border-bottom-style</u>	Sets the style of the bottom border	<u>border-right-color</u>	Sets the color of the right border
<u>border-bottom-width</u>	Sets the width of the bottom border	<u>border-right-style</u>	Sets the style of the right border
<u>border-color</u>	Sets the color of the four borders	<u>border-right-width</u>	Sets the width of the right border
<u>border-left</u>	Sets all the left border properties in one declaration	<u>border-style</u>	Sets the style of the four borders
<u>border-left-color</u>	Sets the color of the left border	<u>border-top</u>	Sets all the top border properties in one declaration
<u>border-left-style</u>	Sets the style of the left border	<u>border-top-color</u>	Sets the color of the top border
		<u>border-top-style</u>	Sets the style of the top border
		<u>border-top-width</u>	Sets the width of the top border
		<u>border-width</u>	Sets the width of the four borders

# CSS边距[マージン] (margin)

- CSS margin属性用于在任何定义的边界之外的元素周围创建空间。
- CSS可以完全控制边距。一些属性可以为元素的每一侧（上，右，下和左）设置边距。

```
<div style="border:1px solid blue">
  <p style="margin:30px; border:1px solid green">
    此元素的边距为30px
  </p>
</div>
```



# 单侧边距

- CSS具有用于为元素的每一侧指定边距[マージン]的属性：
  - margin-top
  - margin-right
  - margin-bottom
  - margin-left
- 所有边距属性都可以具有以下值：
  - 自动-浏览器[ウェブブラウザ]计算边距
  - 长度-以px, pt, cm等为单位指定边距。
  - %-指定包含元素宽度百分比的边距
  - 继承-指定此边距应从父元素继承边距

提示：允许使用负值。

# 边距速记[一括指定]

- 如果margin属性具有四个值，则分别表示上，下，左，右：

```
p {  
    margin: 25px 50px 75px 100px;  
}
```

- 具有三个值，则分别表示上，左右，下：

```
p {  
    margin: 25px 50px 75px;  
}
```

- 具有两个值，则表示上下，左右：

```
p {  
    margin: 25px 50px;  
}
```

- 具有一个值，则表示上下左右：

```
p {  
    margin: 25px;  
}
```

尝试代码：margin.html

# 自动边距

- 可以将margin属性设置为auto将元素在其容器中水平居中显示
- 元素将占据指定的宽度，剩余空间将在左右边界之间平均分配。

```
div {  
    width: 300px;  
    margin: auto;  
    border: 1px solid red;  
}
```

# 继承边距

- 此示例使

元素的左边距继承自父元素 (<div>)
- :

```
div {  
    border: 1px solid red;  
    margin-left: 100px;  
}  
  
p.ex1 {  
    margin-left: inherit;  
}
```

尝试代码：[margin\\_auto.html](#)

# CSS 填充[パディング] padding

- CSS padding属性使得在任何边界内的内容周围生成空间。
- CSS可以完全控制填充。一些属性可以为元素的每一侧（顶部，右侧，底部和左侧）设置填充。
- CSS具有用于为元素的每一侧指定填充的属性：
  - padding-top
  - padding-right
  - padding-bottom
  - padding-left
- 所有填充属性都可以具有以下值：
  - 长度 -以px, pt, cm等指定填充。
  - % -指定包含元素宽度的填充百分比
  - 继承 -指定填充应从父元素继承
- 注意：不允许使用负值。

# Padding速记[一括指定]

- 与margin相似

```
div {  
    border: 1px solid black;  
    padding: 25px 50px 75px 100px;  
    background-color: lightblue;  
}
```

This div element has a top padding of 25px, a right padding of 50px, a bottom padding of 75px, and a left padding of 100px.

- 尝试代码：padding.html

# CSS 高度和宽度

- height和width属性用来设置元件的高度和宽度。
- height和width属性不包括填充，边框或边距。它设置元素的边框，边框和边距内部区域的高度/宽度。
- height和width属性可能具有以下值：
  - auto-这是默认设置。浏览器[ウェブブラウザ]计算高度和宽度
  - length -以px, cm等为单位定义高度/宽度。
  - % -定义包含块的百分比的高度/宽度
  - initial -将高度/宽度设置为其默认值
  - inherit -高度/宽度继承其父值
- 尝试代码：[heightwidth\\_html](#)

# Box模型里元素的宽度和高度

- 重要提示：使用CSS设置元素的width和height属性时，只需设置内容区域的width和height。要计算元素的完整大小，还必须添加padding，border和margin。

- 例：

```
div {  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```

- 计算如下：

- $320\text{px} \text{ (宽度)} + 20\text{px} \text{ (左侧+右侧填充)} + 10\text{px} \text{ (左侧+右侧边框)} + 0\text{px} \text{ (左侧+右侧边距)}$   
 $= 350\text{px}$

# CSS 文字对齐

- text-align属性用于设置文本的水平对齐方式。
- 文本可以左对齐或右对齐，居中。
- 下面的示例显示居中对齐以及左右对齐的文本（如果文本方向是从左到右，则默认为左对齐；如果文本方向是从右到左，则默认为右对齐）：

```
h1 {  
    text-align: center;  
}  
  
h2 {  
    text-align: left;  
}  
  
h3 {  
    text-align: right;  
}
```

# CSS文字的一些其他属性

- `text-indent`属性用于指定文本第一行的缩进。
- `letter-spacing`属性用于指定文本中字符之间的间隔。
- `line-height`属性用于指定行之间的间隔。
- `word-spacing`属性用于指定文本中单词之间的间隔。

Property	Description
<code>color</code>	Sets the color of text
<code>direction</code>	Specifies the text direction/writing direction
<code>letter-spacing</code>	Increases or decreases the space between characters in a text
<code>line-height</code>	Sets the line height
<code>text-align</code>	Specifies the horizontal alignment of text
<code>text-decoration</code>	Specifies the decoration added to text
<code>text-indent</code>	Specifies the indentation of the first line in a text-block
<code>text-shadow</code>	Specifies the shadow effect added to text
<code>text-transform</code>	Controls the capitalization of text
<code>text-overflow</code>	Specifies how overflowed content that is not displayed should be signaled to the user
<code>unicode-bidi</code>	Used together with the <code>direction</code> property to set or return whether the text should be overridden to support multiple languages in the same document
<code>vertical-align</code>	Sets the vertical alignment of an element
<code>white-space</code>	Specifies how white-space inside an element is handled
<code>word-spacing</code>	Increases or decreases the space between words in a text

# CSS 字体[フォント]

- CSS字体属性定义了字体系列，粗体，大小和文本样式。

Serif和Sans-serif字体之间的区别



# CSS 字体家族[フォントファミリ]

- 在CSS中，有两种类型的字体系列名称：
  - 通用族(generic) - 外观相似的一组字体族 (例如“Serif”或“Monospace”)
  - 字体家族(font family) - 特定的字体家族 (例如“Times New Roman”或“Arial”)

Generic family	Font family	Description
Serif	Times New Roman Georgia	Serif fonts have small lines at the ends on some characters
Sans-serif	Arial Verdana	"Sans" means without - these fonts do not have the lines at the ends of characters
Monospace	Courier New Lucida Console	All monospace characters have the same width

# 字体系列

- 文本的字体系列是使用font-family属性设置的。
- font-family属性应包含多个字体名称作为“备用”。如果浏览器[ウェブブラウザ]不支持第一个字体，它将尝试下一个字体，依此类推。
- 如果没有其他可用字体，请从所需字体开始，然后以通用系列结束，以使浏览器[ウェブブラウザ]选择通用系列中的相似字体。
- 注意：如果字体系列的名称超过一个单词，则它必须用引号引起，例如：“Times New Roman”。

```
.serif {  
    font-family: "Times New Roman", Times, serif;  
}  
  
.sansserif {  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
.monospace {  
    font-family: "Lucida Console", Courier, monospace;  
}
```

# 字体样式

- font-style属性主要用于指定斜体文本。
- 此属性具有三个值：
  - 正常-文本正常显示
  - 斜体-文字以斜体显示
  - 斜线-文字为“倾斜”（斜线与斜体非常相似，但支持较少）

```
p.normal {  
    font-style: normal;  
}  
  
p.italic {  
    font-style: italic;  
}  
  
p.oblique {  
    font-style: oblique;  
}
```

# CSS 字体大小

- font-size属性设置文本的大小。
- 在网页设计中，管理文本大小是很重要的。但是，不要同通过调整字体大小来使段落看起来像标题，或者标题看起来像段落。
- 始终使用正确的HTML标签，例如<h1>-<h6>用于标题，<p>用于段落。
- 字体大小值可以是绝对或相对大小。
- 绝对尺寸：
  - 将文本设置为指定大小
  - 在所有浏览器[\[ウェブブラウザ\]](#)中用户不可以更改文本大小
- 相对尺寸：
  - 设置相对于周围元素的大小
  - 允许用户在浏览器[\[ウェブブラウザ\]](#)中更改文本大小
- 注意：如果未指定字体大小，则普通文本（如段落）的默认大小为16px（ $16\text{px} = 1\text{em}$ ）。

# 用px设置字体大小

- 使用像素[画素]设置文本大小可完全控制文本大小：

```
h1 {  
    font-size: 40px;  
}
```

```
h2 {  
    font-size: 30px;  
}
```

```
p {  
    font-size: 14px;  
}
```

# 响应式[レスポンシブ]字体大小

- 文本大小可以用vw单位设置，即“视口[ビューポート]宽度”。
- 这样，文本大小将遵循浏览器[ウェブブラウザ]窗口的大小：

```
<h1 style="font-size:10vw">Hello World</h1>
```

- 查看代码：[responsive\\_font\\_size.html](#)

# Google字体

- 也可以使用[Google字体API](#)向页面添加数百种其他字体。
- 只需添加一个样式表链接并引用你选择的字体系列：

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Sofia">
<style>
body {
    font-family: "Sofia";
    font-size: 22px;
}
</style>
</head>
<body>

<h1>Sofia Font</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>

</body>
</html>
```

# 所有CSS字体属性

Property	Description
<u>font</u>	Sets all the font properties in one declaration
<u>font-family</u>	Specifies the font family for text
<u>font-size</u>	Specifies the font size of text
<u>font-style</u>	Specifies the font style for text
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font
<u>font-weight</u>	Specifies the weight of a font

# CSS 图标[アイコン]

- 如何添加图标：
  - 向HTML页面添加图标的最简单方法是使用图标库，例如Font Awesome。
  - 将指定图标类的名称添加到任何内联[インライン]HTML元素（如*<i>*或<span>）。
  - 之后的图标库中的所有图标都是可缩放矢量，可以使用css（大小，颜色，阴影等）进行自定义。



# Font Awesome (图标库)

- 要使用Font Awesome图标，请访问[fontawesome.com](https://fontawesome.com)，登录并获取代码添加到<head>HTML页面的部分中：
- <script src="https://kit.fontawesome.com/yourcode.js"></script>
- 在[Font Awesome 5教程](#)中阅读有关如何开始使用Font Awesome的更多信息。
- 注意：无需下载或安装！

```
<!DOCTYPE html>
<html>
<head>
<script src="https://kit.fontawesome.com/a076d05399.js"></script>
</head>
<body>

<i class="fas fa-cloud"></i>
<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>

</body>
</html>
```



# Google图标

- 要使用Google图标，<head>请在HTML页面的部分内添加以下行：
  - <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
- 注意：无需下载或安装！

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
      href="https://fonts.googleapis.com/icon?
      family=Material+Icons">
</head>
<body>

<i class="material-icons">cloud</i>
<i class="material-icons">favorite</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">traffic</i>

</body>
</html>
```



# Q & A

You Have Questions  
We Have Answers

# CSS 链接

- 使用CSS，可以用不同的方式设置链接的样式。

文字链接

文字链接

链接按钮

链接按钮

# 样式链接

- 链接可以有任何CSS属性（如设置样式color, font-family, background等）。
- 此外，可以根据链接处于什么状态来设置不同样式的链接。
- 四个链接状态为：
  - a:link -正常，未访问的链接
  - a:visited -用户访问过的链接
  - a:hover -用户将鼠标悬停在链接上时的链接
  - a:active -单击后的链接
- 为多个链接状态设置样式时，有一些顺序规则：
  - a: hover必须在a: link和a: visited之后
  - a: 活动必须在a: hover之后
- 尝试代码：style\_links.html

# CSS 布局[レイアウト]- position属性

- position属性指定用于元素的定位方法的类型（静态，相对，固定，绝对或粘性）。
- 有五个不同的位置值：
  - static
  - relative
  - fixed
  - absolute
  - sticky
- 然后使用top, bottom, left和right属性定位元素。但是，除非position先设置该属性，否则这些属性将不起作用。根据位置值，它们的工作方式也不同。

尝试代码： position.html

# Position: static

- HTML元素默认情况下处于静态位置。
- 静态定位的元素不受top, bottom, left和right属性的影响。
- 元素的position: static;定位没有任何特殊方式。它始终根据页面的正常流程定位：

```
div.static {  
    position: static;  
    border: 3px solid #73AD21;  
}
```

# Position: relative

- 元素position: relative; 的位置相对于其正常位置。
- 设置相对放置的元素的top, right, bottom和left属性将导致其偏离其正常位置进行调整。其他内容将不进行调整以适合元素留下的任何间隙。

```
div.relative {  
    position: relative;  
    left: 30px;  
    border: 3px solid #73AD21;  
}
```

# Position: fixed

- 具有的元素position: fixed; 相对于视口[ビューポート]放置，这意味着即使滚动页面，它也始终位于同一位置。top, right, bottom和left属性用于定位元素。
- 固定元素不会在页面上通常留有空隙的地方留下空隙。
- 请注意页面右下角的固定元素。这是使用的CSS：

```
div.fixed {  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    width: 300px;  
    border: 3px solid #73AD21;  
}
```

此<div>元素具有 position: fixed;

# Position: absolute

- position: absolute; 的位置相对于最近的祖先（而不是相对于视口[ビューポート]的位置，如固定）。
- 但是，如果绝对定位的元素没有祖先，它将使用文档body，并随页面滚动一起移动。
- 注意：“position”元素是位置除以外的任何元素 static。
- 这是一个简单的示例：

```
div.relative {  
    position: relative;  
    width: 400px;  
    height: 200px;  
    border: 3px solid #73AD21;  
}  
  
div.absolute {  
    position: absolute;  
    top: 80px;  
    right: 0;  
    width: 200px;  
    height: 100px;  
    border: 3px solid #73AD21;  
}
```

这个

元素的位置是：relative；

该

元素的位置为：绝对；

## Position: sticky

- 元素position: sticky;的位置基于用户的滚动位置。
- sticky元素根据滚动位置在relative和之间切换fixed。它将相对定位，直到在视口[ビューポート]中遇到给定的偏移位置为止-然后将其“粘贴”在适当的位置（例如position： fixed）。
- 注意：有的浏览器[ウェブブラウザ]如Internet Explorer， Edge 15和更早版本不支持sticky定位。

# CSS 布局[レイアウト]-float

- CSS float属性指定元素应如何浮动。
- float属性用于定位和格式化内容，例如，使图像向左浮动到容器中的文本。
- float属性可以具有以下值之一：
  - left-元素浮动到其容器的左侧
  - right-元素浮动到其容器的右侧
  - none-元素不会浮动（将显示在文本中刚出现的位置）。这是默认值
  - inherit-元素继承其父元素的float值
- 在最简单的用法中，float属性可用于在图像周围包裹文本。

尝试代码：float.html

# CSS 布局[レイアウト]-clear

- clear属性指定哪些元素可以在clear的元素旁边以及在哪一侧浮动。
- 该clear属性可以具有以下值之一：
  - none-允许两面都浮动。这是默认值
  - left-左侧不允许有浮动元素
  - right-右侧不允许有浮动元素
  - both-左侧或右侧均不允许使用浮动元素
  - inherit-元素继承其父元素的clear值
- 使用clear属性的最常见方法是在float元素使用后。
- 尝试代码：[clear.html](#)

# CSS 布局[レイアウト]-float示例

- 网站布局示例
- 使用float属性进行整个Web布局也很常见：

The Flight

The City

The Island

The Food

# The City

Chania is the capital of the Chania region on the island of Crete. The city can be divided in two parts, the old town and the modern city.

You will learn more about web layout and responsive web pages in a later chapter.

Footer Text

尝试代码： float\_example.html

# CSS 组合器[结合子]

- 组合器可以解释选择器[セレクタ]之间的关系。
- 在选择器[セレクタ]之间，我们可以包含一个组合器。
- CSS中有四种不同的组合器：
  - 后代选择器（空格）
  - 子选择器（>）
  - 相邻的兄弟选择器（+）
  - 一般同级选择器（~）

尝试代码：combinators.html

# CSS 组合器[结合子]

- 后代选择器匹配作为指定元素后代的所有元素。
- 子选择器选择所有属于指定元素的子元素。
- 相邻的同级选择器选择所有作为指定元素的相邻同级的元素。同级元素必须具有相同的父元素，“相邻”表示“紧随其后”。
- 一般的同级选择器选择所有属于指定元素的同级元素。

# CSS 查询方式

- 进入[CSS官方文档](#)，默认为英语，可以更改语言为中文
- 在右上角搜索想要搜索的内容即可。

The screenshot shows a search interface with a search bar containing 'css border color'. Below the search bar, it says 'About 6,960 results (0.29 seconds)'. Three search results are listed:

- CSS border-color property**  
www.w3schools.com › cssref › pr\_border-color  
Definition and Usage. The **border-color** property sets the color of an element's four borders. This property can have from one to four values. If the **border-color** ...
- CSS Border Color**  
www.w3schools.com › css › css\_border\_color  
The **border-color** property is used to set the color of the four borders. The color can be set by: name - specify a color name, like "red"; HEX - specify a HEX value, ...
- CSS Borders**  
www.w3schools.com › css › css\_border  
The CSS **border** properties allow you to specify the style, width, and **color** of an element's **border**. I have **borders** on all sides. I have a red bottom **border**.

# Bootstrap模版

- Bootstrap是最流行的HTML, CSS和JavaScript网站框架。
- Bootstrap有各式各样的写好的css样式的类，可以直接拿来用
- Bootstrap完全免费提供下载和使用！
- [BootStrap4.1中文官方文档](#)
- 强烈建议直接读BS的官方文档，甚至不需要自己写css就可以做出很漂亮的网站。

# Q & A

You Have Questions  
We Have Answers

THANK YOU