Chapter 2 CSS

CSS（Cascading Style Sheets）是一种用于样式化网页元素的语言，用于控制网页的外观和布局。它与 HTML 一起使用，可以将样式应用于不同的 HTML 元素，从而实现页面的美化和排版。

1. 选择器的种类： CSS 有多种选择器，包括元素选择器、类选择器、ID 选择器(只能调用一次且唯一性)、属性选择器、伪类和伪元素等。每种选择器都允许你选择不同的 HTML 元素。

/\* 元素选择器 \*/

p {

color: blue;

}

/\* 类选择器 \*/

.my-class {

font-size: 16px;

}

/\* ID 选择器 \*/

#my-id {

background-color: gray;

}

/\* 属性选择器 \*/

[type="text"] {

border: 1px solid black;

}



1. 标签选择器：

p {

font-size: 12px;

color: red;

}

1. class 可以有多个值:

<style>

.red {

background-color: red;

}

.green {

background-color: green;

}

.box{

width: 100px;

height: 100px:

}

</style>

<body>

<div class=”red box”></div>

<div class=”green box”></div>

</body>

1. 后代选择器：

<style>

ol li {

color:pink

}

ol li a {

color: red

}

.nav a {

color: yellow

}

</style>

<body>

<ol>

<li>xxx</li>

<li>xxx</li>

<li><a href=”#”>xxx</a></li>

</ol>

<ul class=”nav”>

<li>xxx</li>

<li>xxx</li>

<li><a href=”#”>xxx</a></li>

</ul>

</body>

1. 子元素选择器:

<style>

.nav > a {

color： red;

}

</style>

<body>

<div class=”nav”>

<a href=”#”>child</a>

<p>

<a href=”#”>grandchild</a>

</p>

</div>

</body>

1. 并集选择器：

<style>

div,

p,

.pig li {

color: red;

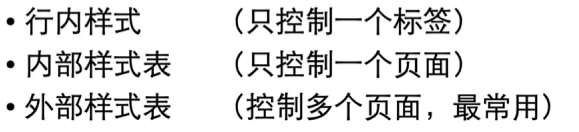
}

</style>

1. 复合属性：

font: italic bold 16px aerial;

1. 样式种类;



外部样式：

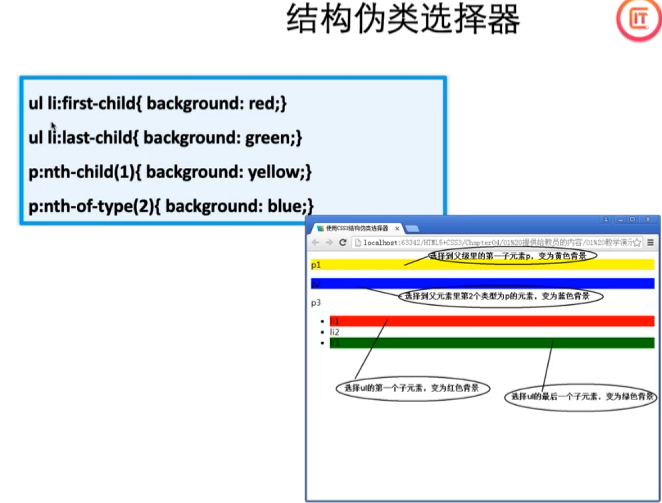
<head>

<link href=”style.css“ rel=”stylesheet” type=”text/css” />

<head>

1. 结构伪类选择器：





1. 属性选择器:



1. 链接伪类选择器：

<style>

/\* 未点击过链接为红色，且去掉下划线 \*/

a:link{

color: blue;

text-decoration: none;

}

/\* 点击过链接橙色 \*/

a:visited{

color: orange;

}

/\* 鼠标经过链接 \*/

a:hover {

color: blue;

}

/\* 按下链接未弹起 \*/

a:active {

color: green;

}

</style>

1. focus选择器：

<style>

inout:focus {

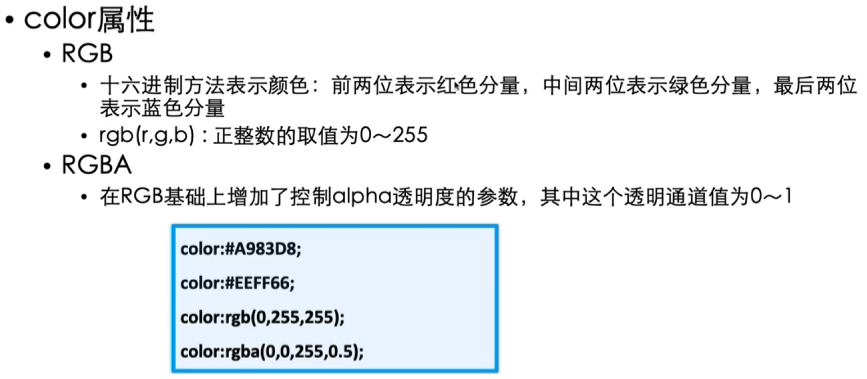
background-color: lightblue;

color: red;

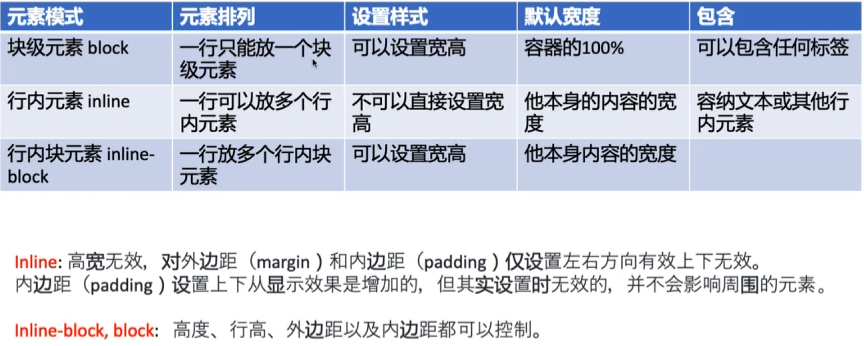
}

</style>

1. color:



1. block(review):



1. 元素显示模式转换：

<style>

/\* a 转为block设置宽高 \*/

a {

display: block;

width: 150px;

height: 50px;

}

/\* div转为inline \*/

div {

display: inline;

}

/\* span设置宽高并一行多个 \*/

span {

display: inline-block;

width: 150px;

height: 50px

}

</style>

1. 文字上下左右居中：

<style>

div {

width: 200px;

height: 40px;

background-color: purple;

color: #fff;

/\* 一种方法：text-align: center;

line-height: 40px; \*/

display: flex;

justify-content: center;

align-items: center;

}

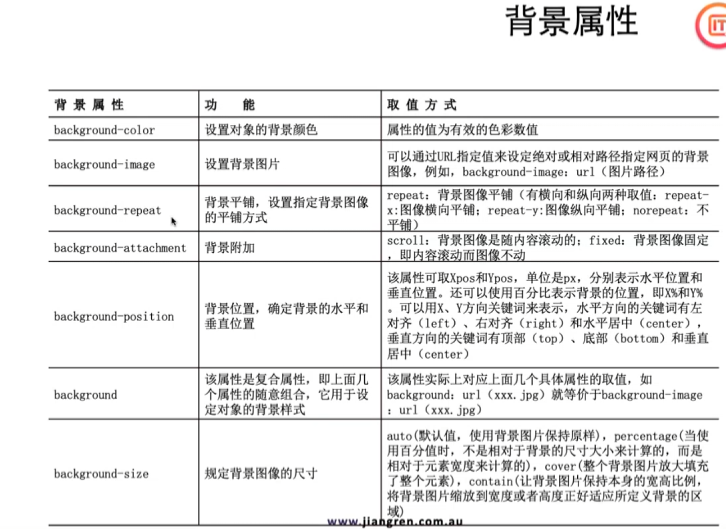
</style>

<body>

<div>center</div>

</body>

1. background attribute:



<style>

/\* 给box1设置宽高200px，背景色蓝色 \*/

.box1 {

height: 200px;

width: 200px;

background-color: lightblue;

}

/\* 给box2添加背景logo图，宽高300px， 不平铺\*/

.box2 {

width: 300px;

height: 300px;

background-image: url(images/logo.png);

background-repeat: no-repeat;

}

/\* 给box3添加背景logo图，添加背景色黄色，沿着y轴平铺 \*/

.box3 {

width: 300px;

height: 300px;

background-color: yellow;

background-image: url(images/logo.png);

background-repeat: repeat-y;

}

</style>

<body>

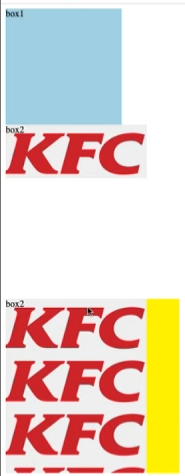
<div class="box1">box1</div>

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

<div class="box3">box2</div>

</body>

效果：



1. 背景位置：

<style>

/\* 给h3添加背景图icon.png， no repeat，定位左中部 ，

x:left center right

y: top center bottom

文字垂直居中，文字出现在icon后 \*/

/\* 默认顺序是水平，垂直。 left top, top left效果一样\*/

/\*如果只指定一个方位名称，另一个省略。省略的那个默认是center \*/

h3{

background-color: lightblue;

width: 200px;

height: 40px;

line-height: 40px;

font-weight: 400;

background-image: url(images/icon.png);

background-repeat: no-repeat;

background-position: left;

text-indent: 1.5em;

}

/\* 给box设置宽高300px，背景色lightblue，背景图logo.png，no repeat，

图片位置水平居中，垂直20px \*/

.box {

width: 300px;

height: 300px;

background-color: lightblue;

background-image: url(images/logo.png);

background-repeat: no-repeat;

background-position: center 20px;

}

</style>

<body>

<h3>Get the best deals!</h3>

<div class="box"></div>

</body>

1. 背景固定：

<style>

/\* 给body添加背景图bg.jpg，背景色黑色， no repeat, center top， 把背景图固定住 \*/

/\* background-attachment 默认值是scroll \*/

body {

color: #fff;

font-size: 20px;

\*/ background-color: #000;

background-image: url(images/bg.jpg);

background-repeat: no-repeat;

background-position: center top;

background-attachment: fixed; \*/

background: #000 url(images/bg.jpg) no-repeat center top fixed;

}

</style>

<body>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hello world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

<p>hellow world</p>

</body>

1. 背景半透明：

<style>

/\* 背景图https://placekitten.com/1200/800 \*/

/\* 背景透明，文字居中不透明 \*/

.hero {

height: 100vh;

width: 100%;

display: flex;

align-items: center;

position: relative;

}

.hero::before {

content: '';

background-image: url('https://placekitten.com/1200/800');

background-size: cover;

position: absolute;

top: 0px;

right: 0px;

bottom: 0px;

left: 0px;

opacity: 0.5;

}

.h1 {

position: relative;

color: #000;

font-size: 4rem;

}

</style>

<body>

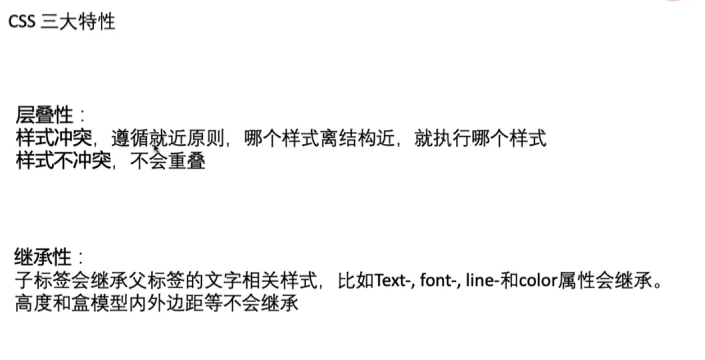
<div class="hero">

<h1>Cute kittehs everywhere!</h1>

</div>

</body>

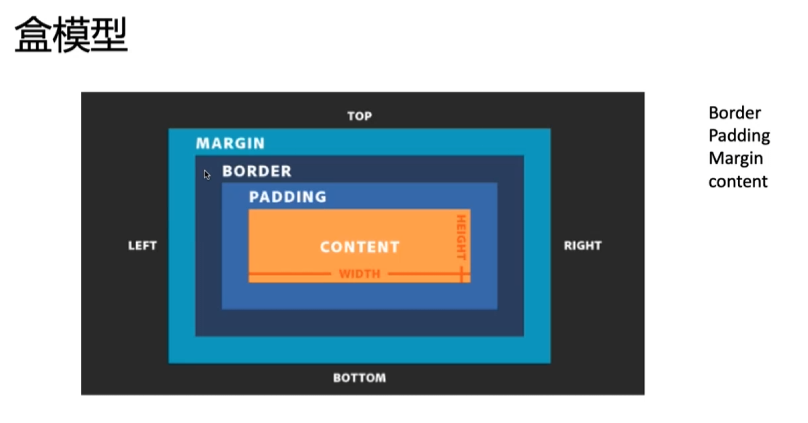
1. css层叠性：

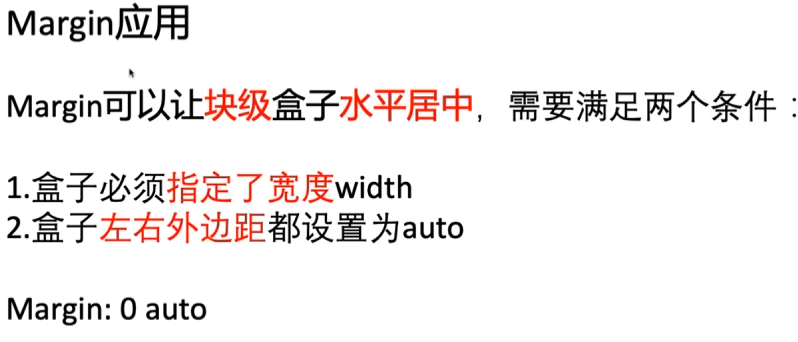


1. 权重叠加：



1. 盒模型:





/\* h1继承了body的宽度，如果h1不指定宽度，padding左右不会撑开盒子 \*/

/\* 如果指定h1宽度，padding就会撑开左右大小 \*/

<style>

div {

width: 200px;

height: 200px;

background-color: lightblue;

}

/\* 给box1加上下左右padding5px

给box2加上上下padding5px，左右padding10px

给box3加上 上padding5px，下padding 20px，左右padding 10px

给box4加上 上5px,下10px， 左20px， 右30px

思考：加border和padding后，如何不影响盒子大小

\*/

/\* box1上下左右 \*/

.box1 {

padding: 5px;

}

/\*box2 上下，左右 \*/

.box2 {

padding: 5px 10px;

}

/\*box3 上，左右，下 \*/

.box3 {

padding: 5px 10px 20px;

}

/\* 从top开始顺时针 top right bottom left\*/

.box4{

padding: 5px 30px 10px 20px;

}

</style>

<body>

<div class="box1">hello world</div>

<div class="box2">hello world</div>

<div class="box3">hello world</div>

<div class="box4">hello world</div>

</body>

1. margin：

<style>

/\* Horizontal centering of inline and inline-block elements \*/

/\*1.header设置 margin top 100px, 并且水平居中 \*/

/\*2. span水平居中 \*/

/\*3. span上下左右居中 \*/

\* {

margin: 0;

padding: 0;

}

.header {

width: 800px;

height: 200px;

background-color: blue;

margin: 100px auto;

color: #fff;

/\* text-align: center; \*/

display: flex;

justify-content: center;

align-items: center;

}

</style>

<body>

<div class="header">

<span>hello</span>

</div>

</body>

align-items： align-items 是用于在弹性容器中设置子元素的垂直对齐方式。它适用于使用弹性布局（Flexbox）的情况。这个属性控制弹性容器内部的子元素在交叉轴（与主轴垂直的轴，比如在flex-direction: row布局中，交叉轴是垂直方向）上的对齐方式。可能的值包括：flex-start（默认，顶部对齐）、flex-end（底部对齐）、center（居中对齐）、baseline（基线对齐）和 stretch（拉伸填充）等。

text-align 是用于控制文本内容在其容器中的水平对齐方式。它适用于文本内容以及行内块级元素。这个属性只影响元素内部的文本内容或内联元素的水平对齐，不影响元素本身在其父容器中的位置。

可能的值包括：left（左对齐）、right（右对齐）、center（居中对齐）、justify（两端对齐）等。

综上所述，align-items 用于弹性容器中子元素的垂直对齐，而 text-align 用于控制文本内容或行内元素的水平对齐。两者作用的对象和效果不同。

https://codepen.io/enxaneta/full/adLPwv/