

页面产生乱码的根本原因:

编码和解码的字符集不同, 编码是将汉字转换为01的二进制, 解码反之

<meta标签>:

主要是用来设置一些元数据给浏览器和seo爬取。

<meta charset="utf-8">

name的值:

1.Keywords:

<meta name="Keywords" content="网上购物, 购物">

Keywords只要是用来设置一些网站的关键字列如, 上面的关键字就是 网上购物, 购物

关键字之间要用逗号隔开。

只是京东的Keywords:

```
<meta name="keywords" content="网上购物,家电,手机,计算机,服装,国际,海外,居家,母婴,美妆,个护,食品,京东,集运,全球"> == $0
```

2.description:

用于被爬取显示的文字。

<meta name="description" content="京东全球版-专业的综合网上购物商城,为您提供正品低价的购物选择、优质便捷的服务体验。商品来自全球数十万品牌商家,囊括家电、手机、计算机、服装、居家、母婴、美妆、个护、食品等丰富品类,满足各种购物需求。支持香港自提、京东集运和国际卡支付,现时已覆盖香港、澳门、台湾、新加坡、马来西亚、美国、加拿大、澳大利亚、新西兰和日本等海外200多个国家和地区。">

京东(JD.COM)-正品低价、品质保障、配送及时、轻松购物! 官方



京东JD.COM-专业的综合网上购物商城,销售家电、数码通讯、电脑、家居百货、服装服饰、母婴、图书、食品等数万个品牌优质商品。便捷、诚信的服务,为您提供愉悦的网上...

<https://www.jd.com/> - 百度快照 - 5615条评价

3.Author :用于表明该网页的作者

4.Copyright : 用于表明该网页的版权

标题标签:

h1:建议只写一个, 越少越好, 因为SEO爬取完title后会立即爬取h1标签里的内容, 从而影响搜索引擎的排名

h2:页面的副标题, 或者一个段落的标题

h3:小节标题

h4-h6:一些重要的内容

语气强调词:

 语气强调词产生斜体<i> </i>标签的效果也是相同

强调词:

 字体加粗, 标签的效果也是相同

<i> 与 的区别: <i> 没有语义, 单纯的字体加粗与斜体建于一使用这两个, 要是你要强调什么内容建议用 , 但是<i>标签一般都是来设置一些icon图标

字体大小:

<big> </big>

<small> </small>

与<big>的区别, <small>有语义, 用来存放网页的版权信息, <big>没有语义,h5不支持了

上标与下标:

 :添加一个上表, 比如 2² 就会显示成2的2次方

:有上就有下

删除线:

预设格式:

<pre> </pre>: 不管你在里面打多少个空格与回车, HTML显示的效果多是和你在编辑器里的

格式一样

<code> </code>:没有语义, 一般用来配和<pre>来表示代码

Html5中的布局标签

header标签: 头部标签

footer标签: 底部标签

section标签：区块标签

aside标签：侧边栏标签

nav (navigator) 标签：导航标签

父类塌陷：

在标准文档流中，如果不给父类元素设置高宽，子元素则会把父类元素的内容给撑开，但是，给子元素设置了float后，子类元素脱离了文档流，这将会导致父类元素无法撑开，

就导致了父类元素的塌陷，由于父类元素塌陷了，父类的兄弟元素就会向上移动，导致布局的混乱。

解决方法：

1.给父类开启bfc：

bfc的特点：

- 1.父类的垂直外边距不会与子元素重叠
- 2.开启bfc的元素不会给float元素给覆盖
- 3.可以包含float的子元素

如何开启bfc??：

- 1.给父类设置float（不推荐）
- 2.给父类设置绝对定位
- 3.设置元素为inline-block
- 4.给要开启bfc的元素overflow已给非visible的属性

(aotu,hidden),但是这样不支持ie8以下，但是可以给ie8以下的设置 zoom:1

图片标签

alt: 图片如果显示不出，就会显示先设置好的文字，如果百度xx图片，百度也是根据这个属性来爬取相关关键字的图片。

width, height.宽和高，如果只设置了期中一个属性的大小，第二个也会等比例的缩放，当然，实际开发中建议不要设置这两个，想让图片缩小，图片是缩小了，但是占用的内存还是和没缩小前一样大的，这样子很划不来，有需求可以交给美工来

src:

相对路径:相对当前资源的路径

图片格式:

jpg:支持的颜色多, 支持压缩, 唯一的缺点就是不支持透明

gif:支持的颜色较少, 支持简单的透明(比如, 直线和横行透明, 如果是比较复杂的就不行了)

png:支持的颜色多, 支持透明。

效果一致时, 使用小的, 效果不一致时, 使用效果好的

``

target的属性: `_self` 在自身页面打开

`_blank` 打开一个新的页面

如果设置成iframe标签的name

那么就会在内连框架打开当前页面

块级元素:

a标签可以包含块级元素, 但是不能包含自己(内联元素不能包含块级元素, a标签是个特例)。

p标签只能包含内联元素, 不能包含块级元素

伪类元素：

伪类元素是一种特殊的元素，用来选择元素的特殊位置。

:first-line 用来选择元素的第一行

: first-letter: 用来选择每一行的第一个字。

属性选择器：

p[title="lpl"] : 选中p标签title属性并且等于lpl

p[title^="lpl"] 选中以lpl开头的p标签属性

p[title\$="lpl"] 选中以lpl结尾的p标签属性

p[title*="l"]选中title中包含l的p标签属性

p:first-child选中选中父类元素下的第一个p标签

p:last-child选中选中父类元素下的最后个p标签

p:nth-child(3)选中选中父类元素下的第3个p标签

p:first-of-type选中选中父类元素下的第一个p标签

p:last-of-type选中选中父类元素下的最后个p标签

-child和-of-type的区别，拿p标签来举例，-child： 如果p标签前面有其他的标签（例如span），那么就不会被选中。

-of-type： 不管你前面有多少你前面有多少个其他的标签，都会被选中。

兄弟选择器：

p+span{}: 只能选择p标签下面的第一个span（同一父类情况下），并且中间不能隔着其他什么标签

相邻兄弟选择器：

p~span{}: 选择p下面的所有span标签（同意父类情况下），中间隔着其他标签也没是

否定选择器：

:not(.asd):不选中class名为asd的p标签

长度单位：

px：像素大小。

个

em：和百分比类似，但是是根据自身的字体大小来变换的，1em=一

font-size。

rem：和em差不多，但是区别是，em是根据自身的字体大小来改变，

rem是根据

浏览器的html设置字体大小来改变的默认字体大小是16px

计算公式：

行间距：行高-字体大小=行间距。

div总宽度：border(left and right)+padding(left and right)+width.

div总高度：border(top and bottom)+padding(top and bottom)+height

外边距重叠：

指垂直相邻的兄弟级div同时设置margin-bottom和margin-top,外边距会取最大值而不是两个margin的和。

解决方法：

把两个相邻的div用文字或标签隔开（例如or）。

如果父子元素垂直外边距相邻了。子类设置的外边距会同时给父类设置外边距。

解决方法：

- 1, 不要给子类设置外边距, 而是给父类设置内边距,
- 2, 用文字或者空标签隔开 (例如or<i>).

css显示与隐藏元素:

display: inline :设置为行内元素

block : 设置为块级元素

none : 设置元素为不可见 (隐藏)

inline-block :设置为行内块级元素 (行内元素的特性, 但是可以设置长宽)

visibility: hidden :隐藏元素

visible(默认值) : 设置元素为可见

display与visibility的区别:

visibility隐藏元素后, 会保留当前元素的位置, 二display不会保留当前元素的位置

高度塌陷:

在标准文档流中, 父类的高度会自动被子元素给撑开 (父元素设置高宽除外), 但是, 如果子元素设置了float后, 子元素会脱离文档流, 导致父类元素无法撑开, 父类的兄弟元素向上移动, 此举会导致整个布局混乱。

解决高度塌陷:

- 1.开启bfc (什么是bfc? : 每一个页面的元素都有一个隐藏的属性英文名叫 block formatting context, 默认是关闭状态, 单开启bfc后, 元素会有

以下特征:

- 1.父类的外边距不会与子类重叠
- 2.元素不会被float元素覆盖
- 3.子类可以包含float

如何开启bfc? :

- 1.设置绝对定位

- 2.设置float
 - 3.设置非visible的overflow
 - 4.设置inline-block)
- 2.使用clear给受影响的元素清除浮动

```
clear: left
      right (清楚左右的浮动)
      both (清楚俩边的浮动, (实际上是清楚对它影响最大
```

的))

```
.clearFix:after{
  content:" ";
  display:block;
  clear:both
  /*给名为.clear最后面 (也就是</div>处) 添加一个空
```

格,

在因为content的默认样式是行内样式不能自动换行和

清除浮动*/

```
}
.clearFix{
  zoom:1;
  /*兼容憨憨ie6及以下*/
}
```

定位:

position: static (默认值, 不定位)

absolute: 绝对定位

relative: 相对定位

fixed: 固定定位 (绝对定位)

sticky: 粘滞定位(相对定位)

relative相对定位:

- 1.使用了relative相对定位对提升一个层级, 会盖住其他的div
- 2.偏移量是根据自己现在的位置来偏移的, 并且会保留偏移前的位置
- 3.relative相对定位不会偏离文档流

absolute绝对定位:

- 1.开启absolute绝对定位会脱离文档流
- 2.偏移量是根据最近的祖先定位元素（只要不是static就行了）来进行偏移量

移量

如果祖先元素没有设置定位，就会根据页面来进行偏移并且不会占据定位前的

位置

位置

- 3.会提升一个层级

fixed固定定位：

fixed固定定位虽然也是absolute绝对定位的一种，但是有不同点

- 1.不管你祖先元素有没有定位，fixed都是根据浏览器窗口来进行偏移的，

所以，滚动条进行滚动的时候，它也会跟着浏览器窗口一起滚动

sticky:粘滞定位：

特性和absolute相对定位一样，不过sticky粘滞定位可以让某个元素到达一定位置的时候让其固定住

背景图片：

background-color

background-image:url()

background-repeat: repeat(默认值，背景重复)

repeat-x 图片向X轴重复水平方向

repeat-y 图片向y轴重复垂直方向

no-repeat 图片不重复

background-position: top left center right bottom

比如：left bottom就是左下角，center center就是剧中，

如果只设置

了一个值，那么第二个值默认center。

第二种值：0% 0%就是左上角，100% 100%就是右下

角，第一个值

为x轴（水平方向的）第二个为y轴（垂直方向

的），

也可以为负值

第三种值：0px 0px效果同上

简写顺序：

background:#00ff00 url() no-repeat 16% 25%