1. 什么是HTML、CSS?

超文本标记语言 (Hyper Text Markup Language)。HTML不是一种编程语言，而是一种标记语言 (markup language)，是网页制作所必备的。浏览器把代码解析后的样子就是我们看到的网站。

CSS 指层叠样式表 (Cascading Style Sheets)，样式定义如何显示HTML 元素

1. 五大浏览器？

IE、Firefox、Google Chrome、Safari、Opera

1、IE浏览器内核：Trident内核，也是俗称的IE内核；  
2、Chrome浏览器内核：统称为Chromium内核或Chrome内核，以前是Webkit内核，现在是Blink内核；  
3、Firefox浏览器内核：Gecko内核，俗称Firefox内核；  
4、Safari浏览器内核：Webkit内核；  
5、Opera浏览器内核：最初是自己的Presto内核，后来是Webkit，现在是Blink内核；  
6、360浏览器、猎豹浏览器内核：IE+Chrome双内核；  
7、搜狗、遨游、QQ浏览器内核：Trident（兼容模式）+Webkit（高速模式）；  
8、百度浏览器、世界之窗内核：IE内核；  
9、2345浏览器内核：以前是IE内核，现在也是IE+Chrome双内核；

1. web三大核心技术？

HTML、CSS、JavaScript

结构(HTML) 表示(css) 行为(js)

1. HTML属性？

写法分成两种：

单标签 <header>

双标签 <header></header>

标签的属性：来修饰标签的，设置当前标签的一些功能。

<标签 属性="值" 属性2="值2">

1. HTML初始代码？

<!DOCTYPE html>文档声明 : 告诉浏览器这是一个html文件

<html lang="en"> html文件的最外层标签：包裹着所有html标签代码 lang="en"表示是一个英文网站 lang="zh-CN"表示一个中文网站

<head>

<meta charset="UTF-8"> 元信息：是编写网页中的一些辅助信息 charset="UTF-8"国际编码，让网页不出现乱码的情况

<title>Document</title> 设置网页的标题

</head>

<body>

显示网页内容的区域

</body>

</html>

1. HTML注释？

写法：<!-- 注释的内容 --> 在浏览器中看不到，只能在代码中看到注释的内容。

意义：

1. 把暂时不用的代码注释起来，方便以后使用。

2. 对开发人员进行提示。

1. 标题与段落？

标题 -> 双标签 : <h1></h1> ... <h6></h6>

在一个网页中，h1标题最重要，并且一个.html文件中只能出现一次h1标签。

h5、h6标签在网页中不经常使用。

段落 -> 双标签 : <p></p>

1. 文本修饰标签？

强调 -> 双标签 : <strong></strong>、<em></em>

区别：

1. 写法和展示效果是有区别的，一个加粗、一个斜体

2. strong的强调性更强，em的强调性稍弱。

下标 : <sub></sub>

上标 : <sup></sup>

删除文本 : <del></del>

插入文本 : <ins></ins>

注：一般情况下，删除文本都是和插入文本配合使用的。

1. 图片标签？

<img src="图片的路径"/>

img -> 单标签

src : 引入图片的地址。

alt : 当图片出现问题的时候，可以显示一段友好的提示文字。

title : 提示信息

width、height : 图片的大小

1. 路径的引入？

相对路径：引用的文件的路径相对于你所编写的HTML文件的相对位置

绝对路径：相对于由盘符开始到你的文件位置结束的全部路径

1. 链接标签？

a -> 双标签 <a></a>

href属性 : 链接的地址

target属性 : 可以改变链接打开的方式，默认情况下：在当前页面打开 \_self 新窗口打开 \_blank

base -> 单标签 ：作用就是改变链接的默认行为的。

1. 锚点？

两种做法

1. #号 + id属性

2. #号 + name属性（注意name属性加给的是a标签）

1. 特殊字符？

1. & + 字符

2. 解决冲突啊 左右尖括号、添加多个空格的实现

&lt; &gt; &nbsp;

|  |  |  |
| --- | --- | --- |
| **特殊字符** | **含义** | **特殊字符代码** |
|  | 空格符 | &nbsp； |
| **©** | 版权 | &copy; |
| **®** | 注册商标 | &reg； |
| **<** | 小于号 | &lt； |
| **>** | 大于号 | &gt; |
| **&** | 和号 | &amp； |
| **¥** | 人民币 | &yen; |
| **°** | 摄氏度 | &deg； |

1. 列表标签？

1. 无序列表 -> ul li 符合嵌套的规范

2. 有序列表 -> ol li 一般用的比较少，可以用无序列表来实现

3. 定义列表 -> dl dt dd 列表项需要添加标题和对标题进行描述的内容

注：列表之间可以互相嵌套，形成多层级的列表。

<dl>：定义列表

<dt>：定义专业术语或名词

<dd>：对名词进行解释和描述

1. 表格标签结构？

table、tr、th、td、caption 等

<table>：表格的最外层容器

<tr> ： 定义表格行

<th>： 定义表头

<td>： 定义表格单元

<caption>：定义表格标题

border ： 表格边框

cellpadding ： 单元格内的空间

cellspacing ： 单元格之间的空间

rowspan ： 合并行

colspan ： 合并列

align ： 左右对齐方式

valign ：上下对齐方式

注：之前是有嵌套关系的，要符合嵌套规范。

语义化标签：tHead、tBody、tFood

注：在一个table中，tBody是可以出现多次的，但是tHead、tFood只能出现一次。

align : left、center、right

valign : top、middle、bottom

1. 表单标签？

form、input、textarea、select、label、 ..

<form> ： 表单的最外层容器

<input>： 标签用于搜集用户信息，根据不同的type属性值，展示不同的控件，如输入框、密码框、复选框等。

<textarea> ： 多行文本框

<select>、<option>：下拉菜单

<label>：辅助表单

input(单标签)标签有一个type属性，决定是什么控件。

text、reset、submit、radio、checkbox、password

|  |  |
| --- | --- |
| **type属性** | **含义** |
| **text** | 普通的文本输入框 |
| **password** | 密码输入框 |
| **checkbox** | 复选框 |
| **radio** | 单选框 |
| **file** | 上传文件 |
| **submit** | 提交按钮 |
| **reset** | 重置按钮 |

还有一些常见的属性：

checked、disabled、name、for ...

1. div和span？

div : 做一个区域划分的块

span : 对文字进行修饰的内联

div （块）：

div全称为division，“分割、分区”的意思，<div>标签用来划分一个区域，相当于一块区域容器，可以容纳段落、标题、表格、图像等各种网页元素。即HTML中大多数的标签都可以嵌套在<div>标签中，<div>中还可以嵌套多层<div>，用来将网页分割成独立的、不同的部分，来实现网页的规划和布局。

span（内联） :

用来修饰文字的，div与span都是没有任何默认样式的，需要配合CSS才行。

1. CSS基础语法？

选择器 { 属性1 : 值1 ; 属性2 : 值2 }

width : 宽

height : 高

background-color : 背景色

长度单位 :

1. px -> 像素

2. % -> 百分比

外容器1 -> 600px 当前容器 50% -> 300px

外容器2 -> 400px 当前容器 50% -> 200px

1. CSS样式的引入方式？

1. 内联样式

style属性

2. 内部样式

style标签

区别：内部样式的代码可以复用、复合W3C的规范标准，进行让结构和样式分开处理。

3. 外部样式

引入一个单独的CSS文件，name.css

通过link标签引入外部资源，rel属性指定资源跟页面的关系，href属性资源的地址。

通过@import方式引入外部样式 ( 这种方式有很多问题，不建议使用 )

1. JavaScript，与HTML、CSS之间的关系？

1. HTML是网页内容的载体。内容就是网页制作者放在页面上想要让用户浏览的信息，可以包含文字、图片、视频等。

2. CSS样式是表现。就像网页的外衣。比如，标题字体、颜色变化，或为标题加入背景图片、边框等。所有这些用来改变内容外观的东西称之为表现。

3. JavaScript是用来实现网页上的特效效果。如：鼠标滑过弹出下拉菜单。或鼠标滑过表格的背景颜色改变。还有焦点新闻（新闻图片）的轮换。无论是动画还是与用户进行交互的都可以通过JavaScript来实现的。

1. HTML语义化

所谓HTML语义化指的是，根据网页中内容的结构，选择适合的HTML标签进行编写。

好处：

1. 在没有CSS的情况下，页面也能呈现出很好的内容结构。

2. 有利于SEO，让搜索引擎爬虫更好的理解网页。

3. 方便其他设备解析（如屏幕阅读器、盲人阅读器等）。

4. 便于团队开发与维护。

1. CSS命名规范

常用的CSS命名规则

1.注释的写法:

/\* Footer \*/

内容区

/\* End Footer \*/

2. id的命名:

(1)页面结构

容器: container

页头：header

内容：content/container

页面主体：main

页尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

(2)导航

导航：nav

主导航：mainbav

子导航：subnav

顶导航：topnav

边导航：sidebar

左导航：leftsidebar

右导航：rightsidebar

菜单：menu

子菜单：submenu

标题: title

摘要: summary

(3)功能

标志：logo

广告：banner

登陆：login

登录条：loginbar

注册：regsiter

搜索：search

功能区：shop

标题：title

加入：joinus

状态：status

按钮：btn

滚动：scroll

标签页：tab

文章列表：list

提示信息：msg

当前的: current

小技巧：tips

图标: icon

注释：note

指南：guild

服务：service

热点：hot

新闻：news

下载：download

投票：vote

合作伙伴：partner

友情链接：link

版权：copyright

3.class的命名:

(1)颜色:使用颜色的名称或者16进制代码,如

.red { color: red; }

.f60 { color: #f60; }

.ff8600 { color: #ff8600; }

(2)字体大小,直接使用”font+字体大小”作为名称,如

.font12px { font-size: 12px; }

.font9pt {font-size: 9pt; }

(3)对齐样式,使用对齐目标的英文名称,如

.left { float:left; }

.bottom { float:bottom; }

(4)标题栏样式,使用”类别+功能”的方式命名,如

.barnews { }

.barproduct { }

主要的 master.css

模块 module.css

基本共用 base.css

布局，版面 layout.css

主题 themes.css

专栏 columns.css

文字 font.css

表单 forms.css

补丁 mend.css

打印 print.css

1. CSS背景样式？

background-color 背景色

background-image 背景图

url(背景地址)

默认：会水平垂直都铺满背景图

background-repeat 平铺方式

repeat-x x轴平铺

repeat-y y轴平铺

repeat ( x , y 都进行平铺，默认值 )

no-repeat 都不平铺

background-position : 背景位置

x y : number(px、%) | 单词

x : left、center、right

y : top、center、bottom

background-attachment : 背景图随滚动条移动的方式

scroll : 默认值 ( 背景位置是按照当前元素进行偏移的 )

fixed ( 背景位置是按照浏览器进行偏移的 )

1. CSS边框样式？

border-style : 边框样式

solid : 实线

dashed : 虚线

dotted : 点线

border-width : 边框大小

px ...

border-color : 边框颜色

red #f00 ...

边框也可以针对某一天边进行单独设置 : border-left-style : 中间是方向 left、right、top、bottom

颜色：透明颜色 transparent

三角形显示：

1. 支持border，支持宽高
2. 设置宽高为0

border-top-color: transparent;

border-top-style:solid;

border-top-width:30px;

border-right-color: red;

border-right-style:solid;

border-right-width:30px;

border-bottom-color: transparent;

border-bottom-style:solid;

border-bottom-width:30px;

border-left-color: transparent;

border-left-style:solid;

border-left-width:30px;

1. CSS文字样式？

font-family : 字体类型

英文字体：Arial , 'Times New Roman'

中文字体：微软雅黑 , 宋体

中文字体的英文名称

微软雅黑 : 'Microsoft YaHei'

宋体 : SimSun

英文、中文

衬线体、非衬线体

注意点：

1.多个字体类型的设置目的

2.引号的添加的目的

font-size : 字体大小

默认 : 16px

写法 : number(px) | 单词 ( small large ... 不推荐使用)

font-weight : 字体粗细

模式： 正常( normal ) 加粗 ( bold )

写法：单词(normal、bold) | number ( 100 200 ..... 900 , 100到500都是正常的，600都900都是加粗的 )

font-style : 字体样式

模式： 正常 ( normal ) 斜体 ( italic )

写法：单词 ( normal 、 italic )

注：oblique也是表示斜体，用的比较少，一般了解即可。

区别：

1.italic 带有倾斜属性的字体的才可以设置倾斜操作。

2.oblique 没有倾斜属性的字体也可以设置倾斜操作。

1. CSS段落样式？

text-decoration：文本装饰

下划线 : underline

删除线 :line-through

上划线 : overline

不添加任何装饰 : none

注：添加多个文本修饰：line-through underline overline

text-transform：文本大小写 （ 针对英文段落 ）

小写：lowercase

大写：uppercase

只针对首字母大写：capitalize

text-indent : 文本缩进

首行缩进

em单位：相对单位，1em永远都是跟字体大小相同

text-align : 文本对齐方式

对齐方式 : left 、right、center、justify（两端点对齐）

line-height : 定义行高

什么是行高，一行文字的高度，上边距和下边距的等价关系。

默认行高：不是固定值，而是变化的。根据当前字体的大小再不断的变化。

取值：1. number( px ) | scale ( 比例值 , 跟文字大小成比例的 )

letter-spacing : 字之间的间距

word-spacing : 词之间的间距 ( 针对英文段落的 )

英文和数字不自动折行的问题：

1. word-break : break-all; (非常强烈的折行)

2. word-wrap : break-word;(不是那么强烈的折行 ，会产生一些空白区域)

1. CSS复合样式

复合的写法，是通过空格的方式实现的。复合写法有的是不需要关心顺序，例如background、border；有的是需要关心顺序，例如font。

1. background : red url() repeat 0 0;

2. border : 1px red solid;

3. font :

注：最少要有两个值 size family

weight style size family √

style weight size family √

weight style size/line-height family √

注：如果非要混合去写的话，那么要先写复合样式，再写单一样式，这样样式才不会被覆盖掉。

1. CSS选择器？

1. ID选择器

#elem{} id="elem"

注：

<1. 在一个页面中，ID值是唯一的。

<2. 命名规范， 字母 \_ - 数字（命名的第一位不能是数字）。

<3. 命名方式，驼峰式、下划线式、短线式。

驼峰写法 : searchButton (小驼峰) SearchButton (大驼峰) searchSmallButton

短线写法：search-small-button

下划线写法：search\_small\_button

2. CLASS选择器

.elem{} class="elem"

注：

<1. class选择器是可以复用的。

<2. 可以添加多个class样式。

<3. 多个样式的时候，样式的优先级根据CSS决定，而不是class属性中的顺序。

<4. 标签+类的写法

3. 标签选择器(TAG选择器)

div{} <div></div>

使用的场景：

<1. 去掉某些标签的默认样式时

<2. 复杂的选择器中，如 层次选择器

4. 群组选择器(分组选择器)

可以通过逗号的方式，给多个不同的选择器添加统一的CSS样式，来达到代码的复用。

5. 通配选择器

\*{ } -> div,ul,li,p,h1,h2....{}

注：尽量避免使用通配选择器，因为会给所有的标签添加样式，慎用。

使用的场景：

去掉所有标签的默认样式时

1. 层次选择器？

后代 M N { }

父子 M > N { }

兄弟 M ~ N { } 当前M下面的所有兄弟N标签

相邻 M + N { } 当前M下面相邻的N标签

1. 属性选择器？

M[attr] {}

= : 完全匹配

\*= : 部分匹配

^= : 起始匹配

$= : 结束匹配

[ ][ ][ ] : 组合匹配

|  |  |
| --- | --- |
| **选择器** | **说明** |
| **M[attr]** | M元素选择指定为attr属性的集合 |
| **M[attr=value]** | M元素选择指定为attr属性和value值的集合 |
| **M[attr\*=value]** | M元素选择指定为attr属性并且包含值为value的集合 |
| **M[attr^=value]** | M元素选择指定为attr属性并且起始值为value的集合 |
| **M[attr$=value]** | M元素选择指定为attr属性并且结束值为value的集合 |
| **M[attr1][attr2]** | M元素选择满足多个属性的集合 |

1. 伪类选择器？

M:伪类{}

:link 访问前的样式 ( 只能添加给a标签 )

:visited 访问后的样式 ( 只能添加给a标签 )

:hover 鼠标移入时的样式 (可以添加给所有的标签)

:active 鼠标按下时的样式 (可以添加给所有的标签)

注：

一般的网站都只设置

a{} ( link visited active ) a:hover{}

:after、:before 通过伪类的方式给元素添加一段文本内容，使用content属性

:checked、:disabled 、:focus 都是针对表单元素的

结构性伪类选择器

nth-of-type() nth-child()

角标是从1开始的，1表示第一项，2表示第二项 | n值 表示从0到无穷大

first-of-type

last-of-type

only-of-type

nth-of-type()和nth-child()之间的区别

type : 类型

child : 孩子

1. CSS继承？

文字相关的样式可以被继承

布局相关的样式不能被继承 ( 默认是不能继承的，但是可以设置继承属性 inherit 值 )

1. CSS优先级？

1. 相同样式优先级

当设置相同样式时，后面的优先级较高，但不建议出现重复设置样式的情况。

2. 内部样式与外部样式

内部样式与外部样式优先级相同，如果都设置了相同样式，那么后写的引入方式优先级高。

3. 单一样式优先级

style行间 > id > class > tag > \* > 继承

注：style行间 权重 1000

id 权重 100

class 权重 10

tag 权重 1

4. !important

提升样式优先级，非规范方式，不建议使用。( 不能针对继承的属性进行优先级的提升 )

5. 标签+类与单类

标签+类 > 单类

6. 群组优先级

群组选择器与单一选择器的优先级相同，靠后写的优先级高。

7. 层次优先级

<1. 权重比较

ul li .box p input{} 1 + 1 + 10 + 1 + 1

.hello span #elem{} 10 + 1 + 100

2. 约分比较

ul li .box p input{} li p input{}

.hello span #elem{} #elem{}

1. CSS盒子模型？

组成 : content -> padding -> border -> margin

物品 填充物 包装盒 盒子与盒子之间的间距

content : 内容区域 width和height组成的

padding : 内边距(内填充)

只写一个值： 30px (上下左右)

写两个值 : 30px 40px ( 上下、左右 )

写四个值 : 30px 40px 50px 60px（上、右、下、左）

单一样式只能写一个值：

padding-left

padding-right

padding-top

padding-bottom

margin : 外边距(外填充)

只写一个值： 30px (上下左右)

写两个值 : 30px 40px ( 上下、左右 )

写四个值 : 30px 40px 50px 60px（上、右、下、左）

单一样式只能写一个值：

margin-left

margin-right

margin-top

margin-bottom

注：

1. 背景色填充到margin以内的区域 （不包括margin区域）

2. 文字在content区域添加。

3. padding不能为负数，而margin可以为负数。

box-sizing:

允许您以特定的方式定义匹配某个区域的特定元素。取值为content-box（默认值） | border-box。

盒尺寸，可以改变盒子模型的展示形态。

默认值： content-box : width、height -> content

border-box : width、height -> content padding border

使用的场景：

1. 不用再去计算一些值

2. 解决一些100%的问题

盒子模型的一些问题：

1. margin叠加问题，出现在上下margin同时存在的时候。会取上下中值较大的作为叠加的值。

2. margin传递问题，出现在嵌套的结构中，只是针对margin-top的问题。

扩展：

1. margin左右自适应是可以的 ，但是上下自适应是不行的。( 如果想实现上下自适应的话，需要在第二大部分来进行学习 )

2. width、height不设置的时候，对盒子模型的影响，会自动去计算容器的大小，节省代码。

1. 标签分类？

按类型

block : div、p、ul、li、h1 ...

1. 独占一行

2. 支持所有样式

3. 不写宽的时候，跟父元素的宽相同

4. 所占区域是一个矩形

inline : span 、a、em、strong、img ...

1. 挨在一起的

2. 有些样式不支持，例如：width、height、margin、padding

3. 不写宽的时候，宽度由内容决定

4. 所占的区域不一定是矩形

5. 内联标签之间会有空隙，原因：换行产生的

inline-block : input、select ...

1. 挨在一起，但是支持宽高

注：布局一般用块标签，修饰文本一般用内联标签

按内容

Flow：流内容

Metadata：元数据

Sectioning：分区

Heading：标题

Phrasing：措辞

Embedded：嵌入的

Interactive：互动的

按显示

替换元素 :浏览器根据元素的标签和属性，来决定元素的具体显示内容。

img、input ...

非替换元素 : 将内容直接告诉浏览器，将其显示出来。

div、h1、p ...

1. 显示框类型？

display: block inline inline-block none ...

区别：

display:none 不占空间的隐藏

visibility: hidden 占空间的隐藏

1. 标签嵌套规范？

ul、li

dl、dt、dd

table、tr、td

块能够嵌套内联

<div>

<span></span>

<a href="#"></a>

</div>

块嵌套块

<div>

<div></div>

</div>

特殊：

错误的写法：

<p>

<div></div>

</p>

内联是不能嵌套块

错误的写法：

<span>

<div></div>

</span>

特殊：

正确的写法：

<a href="#">

<div></div>

</a>

1. 溢出隐藏？

overflow :

visible : 默认

hidden

scroll

auto

x轴、y轴

overflow-x、overflow-y针对两个轴分别设置

1. 透明度与手势？

opacity : 0(透明) ~ 1(不透明)

0.5(半透明)

注：占空间、所有的子内容也会透明

rgba() : 0 ~ 1

注：可以让指定的样式透明，而不影响其他样式

cursor : 手势

default : 默认箭头

要实现自定义手势：

准备图片： .cur 、 .ico

cursor : url(./img/cursor.ico),auto;

1. 最大、最小宽高？

min-width、min-height

max-width、max-height

%单位：换算 -> 已父容器的大小进行换算的

一个容器怎么适应屏幕的高 : 容器加height:100%; body:100%; html:100%;

html,body{ height:100%;}

.contrainer{ height:100%;}

1. CSS默认样式？

没有默认样式的： div、span

有默认样式的：

body -> marign : 8px

h1 -> margin : 上下 21.440px

font-weight : bold

p -> margin : 上下 16px

ul -> margin : 上下 16px padding : 左 40px

默认点：list-style : disc

a -> text-decoration: underline;

css reset :

\*{ margin:0; padding:0;}

优点：不用考虑哪些标签有默认的margin和padding

缺点：稍微的影响性能

body,p,h1,ul{ margin:0; padding:0;}

ul{ list-style : none;}

a{ text-decoration: none; color:#999;}

img{ dispaly:block}

问题的现象：图片跟容器底部有一些空隙。

内联元素的对齐方式是按照文字基线对齐的，而不是文字底线对齐的。

vertical-align: baseline; 基线对齐方式，默认值

img{ vertical-align:bottom;} 解决方式是推荐的

写具体页面的时候或一个布局效果的时候：

1. 写结构

2. css重置样式

3. 写具体样式

1. float浮动？

脱离文档流，延迟父容器靠左或靠右进行排列。

left、right、none

float注意点

只会影响后面的元素。

内容默认提升半层。

默认宽根据内容决定。

换行排列。

主要给块元素添加，但也可以给内联元素添加。

如何清除浮动？

上下排列：clear属性，表示清除浮动的，left、right、both

嵌套排列：

固定宽高 : 不推荐 , 不能把高度固定死，不适合做自适应的效果。

父元素浮动 : 不推荐 , 因为父容器浮动也会影响到后面的元素。

overflow : hidden (BFC规范) , 如果有子元素想溢出，那么会受到影响。

display : inline-block (BFC规范)，不推荐，父容器会影响到后面的元素。

设置空标签 : 不推荐 , 会多添加一个标签。

after伪类 : 推荐，是空标签的加强版，目前各大公司的做法。

( clear属性只会操作块标签，对内联标签不起作用 )

1. CSS 定位？

position :

static（默认）

relative

absolute

fixed

sticky

relative：

如果没有定位偏移量，对元素本身没有任何影响

不使元素脱离文档流

不影响其他元素布局

left、top、right、bottom是相对于当前元素自身进行偏移的

absolute：

使元素完全脱离文档流

使内联元素支持宽高 （让内联具备块特性）

使块元素默认宽根据内容决定（让块具备内联的特性）

如果有定位祖先元素相对于定位祖先元素发生偏移，没有定位祖先元素相对于整个文档发生偏移（绝对、相对、固定）

fixed：

使元素完全脱离文档流

使内联元素支持宽高 （让内联具备块特性）

使块元素默认宽根据内容决定（让块具备内联的特性）

相对于整个浏览器窗口进行偏移，不受浏览器滚动条的影响

sticky黏性定位：

在指定的位置，进行黏性操作。

z-index定位层级。

1. CSS添加省略号

width

必须有一个固定的宽

white-space : nowrap

不让内容折行

overflow : hidden

隐藏溢出的内容

text-overflow : ellipsis

添加省略号

1. CSS圆角

border-radius

给标签添加圆角。

#box{ width:200px; height:200px; background: red; border-radius: 100px; }

圆形圆角

#box{ width:300px; height:300px; background: red; border-radius: 50%; }

整圆

#box{ width:300px; height:300px; background: red; border-radius: 10px 20px; }

圆角 上下、左右

#box{ width:300px; height:300px; background: red; border-radius: 10px 20px 30px 40px; }

圆角 上、右、下、左

#box{ width:300px; height:300px; background: red; border-radius: 20px / 40px; }

椭圆圆角

#box{ width:300px; height:150px; background: red; border-radius: 150px 150px 0 0; }

半圆

1. PC端的布局？

通栏 : 自适应浏览器的宽度。

版心 : 固定一个宽度，并且让容器居中。