

周五面试考试

2023年4月13日 14:08

面试官：说说你对盒子模型的理解？

盒子模型

对一个文档进行布局的时候、浏览器的渲染引擎会根据标准之一的css基础框盒模型，将所有的元素表示为一个个的矩形盒子。

一个盒子由四个部分组成：内容、padding、border、margin

盒模型分类

分为w3c标准盒模型和怪异盒模型

标准盒模型

总宽度=width+padding+margin

总高度=height+padding+margin

怪异盒模型

总宽度=width+margin

总高度=height+margin

width/height 包含了padding 和border值

box-sizing

content-box 标准盒模型

border-box 怪异和模型

inherit 从父元素继承

面试官：css选择器有哪些？优先级？哪些属性可以继承？

css属性选择器常有：

id选择器 (#box) ， 选择id为box的元素

类选择器 (.one) ， 选择类名为one的所有元素

标签选择器 (div) ， 选择标签为div的所有元素

后代选择器 (#box div) ， 选择id为box元素内部所有的div元素

子选择器 (.one>one_1) ， 选择父元素为.one的所有.one_1的元素

相邻同胞选择器 (.one+.two) ， 选择紧接在.one之后的所有.two元素

群组选择器 (div,p) ， 选择div、p的所有元素

伪类选择器

:link 未被访问

:hover 鼠标经过

:visited 访问过

:active 选中连接

:focus 选择具有焦点的

:first-child 父元素的首个子元素

伪元素选择器

:first-letter :于选取指定选择器的首字母

:first-line :选取指定选择器的首行

:before :选择器在被选元素的内容前面插入内容

:after :选择器在被选元素的内容后面插入内容

属性选择器

[attribute]选择带有attribute属性的元素

[attribute=value]选择所有使用attribute=value的元素

[attribute~value]选择attribute属性包含value的元素

[attribute Fvalue]:选择attribute属性以value开头的元素

CSS3中新增的选择器

普通兄弟选择器 (p~ul) , 选择前面有p元素的每个ul元素

伪类选择器

:first-of-type父元素的首个元素

:last-of-type父元素的最后一个元素

:only-of-type 父元素的特定类型的唯一子元素

:only-child父元素中唯一子元素

:nth-child(n)选择父元素中第N个子元素

:nth-last-of-type(n)选择父元素中第N个子元素, 从后往前

:last-child父元素的最后一个元素

:root设置HTML文档

: empty指定空的元素

:enabLed选择被禁用元素

:disabled选择被禁用元素

:checked 选择选中的元素

:not (selector)选择非<selector> 元素的所有元素

属性选择器

[attribute*=value]:选择attribute属性包含value的所有元素

[attribute ^ =value]:选择attribute属性开头为value的所有元素

[attribute\$ =value]: 选择attribute属性结尾为value的所有元素

优先级

!important 权重10000

!important 提高属性的权重

行内样式表 1000
id选择器 100
class选择器 10
标签选择器 1
伪类选择器 10
继承的权重 0
通配符 0
后代和子代权重 相加
群组选择器 看自身 各自计算各自的

继承属性

字体系列属性

font:组合字体
font-family:规定元素的字体系列
font-weight:设置字体的粗细
font-size:设置字体的尺寸
font-style:定义字体的风格
font-variant:偏大或偏小的字体

文本系列属性

text-indent:文本缩进
text-align:文本水平对齐
line-height:行高
word-spacing:增加或减少单词间的空白
letter-spacing:增加或减少字符间的空白
text-transform:控制文本大小写
direction:规定文本的书写方向
color:文本颜色

元素可见性

visibility

表格布局属性

caption-side:定位表格标题位置
border-collapse:合并表格边框
border-spacing:设置相单元格的边框间的距离
empty-cells:单元格的边框的出现与消失
table-layout:表格的宽度由什么决定

列表属性

list-style-type:文字前面的小点点样式

list-style-position:小点点位置

list-style:以上的属性可通过这属性集合

引用

quotes:设置嵌套引用的引号类型

光标属性

cursor:箭头可以变成需要的形状

继承中比较特殊的几点:

a标签的字体颜色不能被继承

h1-h6标签字体的大小也是不能被继承的

无继承属性

display

文本属性: vertical-align、 text-decoration

盒子模型的属性:宽度、高度、内外边距、边框等

背景属性:背景图片、颜色、位置等

定位属性:浮动、清除浮动、定位position等

生成内容属性: content、 counter-reset、 counter-increment

轮廓样式属性: outline-style、 outline-width、 outline-color、 outline

页面样式属性: size、 page-break-before、 page-break-after

面试官: css中, 有哪些方式可以隐藏页面元素? 区别?

通过css实现隐藏元素方法有如下:

display:none

特点: 元素不可见, 不占据空间, 无法响应点击事件

visibility:hidden

特点: 元素不可见, 占据页面空间, 无法响应点击事件

opacity:0

特点: 改变元素透明度, 元素不可见, 占据页面空间, 可以响应点击事件

设置height、width模型属性为0

特点: 元素不可见, 不占据页面空间, 无法响应点击事件

position:absolute 将元素移出可视区域

特点: 元素不可见, 不影响页面布局

clip-path 通过裁剪的形式

特点: 元素不可见, 占据页面空间, 无法响应点击事件

display: none、visibility: hidden、opacity: 0的区别

	display: none	visibility: hidden	opacity: 0
页面中	不存在	存在	存在
回流	会	不会	不会
重绘	会	会	不一定
自身绑定事件	不触发	不触发	可触发
transition	不支持	支持	支持
子元素可复原	不能	能	不能
被遮挡的元素可触发事件	能	能	不能

面试官：谈谈你对BFC的理解？

BFC (Block Formatting Context)

即块级格式化上下文,它是页面中的一块渲染区域，并且有一套属于自己的渲染规则:

内部的盒子会在垂直方向上一个接一个的放置

对于同一个BFC的两个相邻的盒子的margin会发生重叠，与与向无关。

每个元素的左外边距与包含块的左边界相接触(从左到右)，即使浮动元素也是如此

BFC的区域不会与float的元素区域重叠

计算BFC的高度时，浮动子元素也参与计算

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素,反之亦然

BFC目的是形成一个相对于外界完全独立的空间，让内部的子元素不会影响到外部的元素

触发条件

根元素，即HTML元素

浮动元素: float值为left、right

overflow值不为visible,为auto、scroll、hidden

display的值为inline-block、inltable-cell、table-caption、table、inline-table、flex、

inline-flex、grid、inline-grid

position的值为absolute或fixed

BFC能解决的问题

防止margin重叠（塌陷）

清除内部浮动

自适应多栏布局

面试官：元素水平垂直居中的方法有哪些？如果元素不定宽高呢？

元素水平垂直居中的方式

利定位+ margin:auto

父相子绝

```
top: 0;  
left: 0;  
right: 0;  
bottom: 0;  
margin:auto;
```

利用定位+ margin:负值

父相子绝

```
top:50%;  
left:50%;  
margin-top: - 子元素高度一半;  
margin-left: - 子元素宽度一半;
```

利用定位+ transform

父相子绝

```
top:50%;  
left:50%;  
transform:translate(-50%,-50%);
```

table布局

设置父元素为display:table-cell，子元素设置 display: inline-block。利用vertical和text-align可以让所有的行内块级元素水平垂直居中

flex布局

```
display:flex;  
justify-content:center;  
align-items:center;
```

grid布局

```
display:grid;  
justify-content:center;  
align-items:center;
```

不知道元素宽高大小仍能实现水平垂直居中的方法有：

- 利用定位+ margin:auto
- 利用定位+transform
- 利用定位+ margin:负值
- flex布局
- grid布局

内联元素居中布局

水平居中

行内元素可设置: text-align: center

flex布局设置父元素: display: flex; justify-content: center

垂直居中

行文本元素确认高度: height = line-height

多行文本父元素确认高度: display: table-cell; vertical-align: middle

块级元素居中布局

水平居中

定宽: margin: 0 auto

绝对定位+left:50%+ margin:负自身一半

垂直居中

position: absolute设置left、top、margin-left、margin-top(定高)

display: table-cell

transform: translate(x, y)

flex(不定高, 不定宽)

grid(不定高, 不定宽),兼容性相对较差

面试官：如何实现两栏布局，右侧自适应？三栏布局中间自适应呢

两栏布局

使用float左浮左边栏

右边模块使用margin-left撑出内容块做内容展示

为块级元素添加BFC,防止下方元素飞到上方内容

flex弹性布局

父元素 display: flex;

左边：设置宽度，高度自适应

右边flex: 1 占据剩余空间

三栏布局

两边使用float,中间使用margin

两边固定宽度，中间宽度自适应。

两边使用absolute,中间使用margin

左右两边使用绝对定位，固定在两侧。

两边使用float和负margin

display: table实现

flex实现

grid网格布局

面试官：说说flexbox（弹性盒布局模型），以及适用场景？

flex, 意为“弹性布局”, 可以简便、完整、响应式地实现各种页面布局

容器的属性

控制所有的弹性项目

01.项目在主轴方向上空间不够时是否换行

flex-wrap:

nowrap 默认 不换行

wrap 换行

02.主轴方向

flex-direction: row; 默认 水平从左到右

flex-direction: row-reverse; 水平从右到左

flex-direction: column; 垂直从上到下

flex-direction: column-reverse; 垂直从下到上

03.换行和主轴方向的简写

flex-flow: wrap column;

04.设置项目在主轴上的排列方式

justify-content: flex-start; 默认 主轴起点对齐

justify-content: flex-end; 主轴终点对齐

justify-content: center; 主轴中心对齐

justify-content: space-around; 容器两边有间隙

justify-content: space-between; 容器两边没有间隙

05.设置项目在交叉轴上的排列方式

align-items: stretch; 默认 拉伸以适应交叉轴的空间

align-items: flex-start; 交叉轴起点对齐

align-items: flex-end; 交叉轴终点对齐

align-items: center; 交叉轴中心对齐

align-items: baseline; 交叉轴基线对齐 了解

06.多行项目在交叉轴上的排列方式

align-content: stretch; 默认 拉伸以适应交叉轴的空间

align-content: flex-start; 交叉轴起点对齐

align-content: flex-end; 交叉轴终点对齐

align-content: center; 交叉轴中心对齐

align-content: space-around; 容器两边有间隙

align-content: space-between; 容器两边没有间隙

项目的属性

控制某一个弹性项目

01.设置单个项目在交叉轴上的排列方式

align-self:

auto 默认 继承父元素的align-items的属性值;

如果父元素没有设置align-items, 就会找自己的align-self的默认值.

stretch 默认值 拉伸以适应 项目在交叉轴方向上不设置尺寸的时候

flex-start 交叉轴起点对齐

flex-end 交叉轴终点对齐

center 交叉轴中心对齐

baseline 交叉轴基线对齐

02.控制项目如何增长

如果主轴方向上空间盈余, 项目们按照取值的占比, 自动增长对应的尺寸

flex-grow:

0 默认值 不空配空间 不扩展

数字 1 2 3 4 ...

03.控制项目如何收缩

如果主轴方向上空间不足, 项目们按照取值的占比, 自动减少对应的尺寸

flex-shrink:

1 默认值 分配空间 收缩

0 不分配空间 不收缩

数字 1 2 3 4...

04.设置项目在主轴方向上的尺寸

flex-basis:

auto 默认

项目在主轴上如果没有设置flex-basis, 会参考width;

如果width也没有设置, 靠内容撑开.

当width和flex-basis冲突, 以flex-basis为准

数值

百分比 参考父元素的宽高

05.项目属性的简写

flex: flex-grow flex-shrink flex-basis

-顺序不可以换

-平分空间: flex:1

项目的排列顺序

order:

auto 默认值

数值 + - 数值越大，排列顺序越靠后

应用场景

我们能够通过flex简单粗暴的实现元素水平垂直方向的居中，以及在两栏三栏自适应布局中通过flex完成，包括现在在移动端、小程序这边的开发，都建议使用flex进行布局。

面试官：CSS3新增了哪些新特性？

css，即层叠样式表（Cascading Style Sheets）的简称，是一种标记语言，由浏览器解释执行用来使页面变得更美观

css3是css的最新标准，是向后兼容的，CSS1/2的特性在CSS3 里都是可以使用的

选择器

选择器	例子	例子描述
<u>element1~element2</u>	p~ul	选择前面有 <p> 元素的每个 元素。
<u>[attribute^=value]</u>	a[src^="https"]	选择其 src 属性值以 "https" 开头的每个 <a> 元素。
<u>[attribute\$=value]</u>	a[src\$=".pdf"]	选择其 src 属性以 ".pdf" 结尾的所有 <a> 元素。
<u>[attribute*=value]</u>	a[src*="abc"]	选择其 src 属性中包含 "abc" 子串的每个 <a> 元素。
<u>:first-of-type</u>	p:first-of-type	选择属于其父元素的首个 <p> 元素的每个 <p> 元素。
<u>:last-of-type</u>	p:last-of-type	选择属于其父元素的最后 <p> 元素的每个 <p> 元素。
<u>:only-of-type</u>	p:only-of-type	选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。
<u>:only-child</u>	p:only-child	选择属于其父元素的唯一子元素的每个 <p> 元素。
<u>:nth-child(n)</u>	p:nth-child(2)	选择属于其父元素的第二个子元素的每个 <p> 元素。
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	同上，从最后一个子元素开始计数。
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	选择属于其父元素第二个 <p> 元素的每个 <p> 元素。
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	同上，但是从最后一个子元素开始计数。
<u>:last-child</u>	p:last-child	选择属于其父元素最后一个子元素每个 <p> 元素。

新样式

边框

css3新增了三个边框属性，分别是：

- border-radius: 创建圆角边框
- box-shadow: 元素添加阴影
- border-image: 使用图片来绘制边框

背景

新增了几个关于背景的属性,分别是background-clip、 background-origin、 background-size和background-break

background-clip

用于确定背景画区,有以下几种可能的属性:

- background-clip: border-box;背景从border开始显示
- background-clip: padding-box;背景从padding开始显示
- background-clip: content-box;背景从content区域开始显示
- background-clip: no-clip;默认属性,等同于border-box

通常情况,背景都是覆盖整个元素的,利用这个属性可以设定背景颜色或图片的覆盖范围

background-origin

当我们设置背景图片时,图片是会以左上角对齐,但是是以border的左上角对齐还是以padding的左上角或者content的左上角对齐?

background-origin正是用来设置这个的

- background-origin: border-box;从border开始计算background-position
- background-origin: padding-box;从padding开始计算background-position
- background-origin: content-box;从content开始计算background-position

默认情况是padding-box,即以padding的左上角为原点

background-size

background-size属性常用来调整背景图片的大小,要用于设定图片本身。有以下可能的属性:

background-size: contain; 缩小图片以适合元素(维持像素长宽比)

- background-size: cover;扩展元素以填补元素(维持像素长宽比)
- background-size: 100px 100px;缩小图片至指定的大小
- background-size: 50% 100%;缩小图片至指定的大小,百分比是相对包裹元素的尺寸

background-break

元素可以被分成几个独立的盒子(如使用内联元素span跨越多行), background-break 属性用来控制背景怎样在这些不同的盒子中显示

- background-break: continuous;默认值。忽略盒子之间的距离(也就是像元素没有分成多个盒子,依然是一个整体一样)
- background-break: bounding-box;把盒子之间的距离计算在内;
- background-break: each-box;为每个盒子单独重绘背景

文字

word-wrap

语法: word-wrap: normal|break-word

- normal: 使用浏览器默认的换行
- breakall: 允许在单词内换行

text-overflow

text-overflow设置或检索当当前行超过指定容器的边界时如何显示，属性有两个值选择:

- clip: 修剪文本.
- ellipsis: 显示省略符号来代表被修剪的文本

text-shadow

text-shadow可向文本应用阴影。能够规定水平阴影、直阴影、模糊距离，以及阴影的颜色

text-decoration

CSS3里开始支持对文字的更深层次的渲染，具体有三个属性可供设置:

- text-fill-color:设文字内部填充颜色
- text-stroke-color: 设置文字边界填充颜色
- text-stroke-width: 设文字边界宽度

颜色

css3新增了新的颜色表示方式rgba与hsla

- rgba分为两部分, rgb为颜色值, a为透明度
- hala分为四部分, h为色相, s为饱和度,为亮度, a为透明度

transition 过渡

transition-property: width;
transition-duration: 1s;
transition-timing- function: linear;
transition-delay: 2s;

transform 转换

transform属性允许你旋转，缩放，倾斜或平移给定元素

transform-origin:转换元素的位置(围绕那个点进行转换)，默认值为(x,y,z):(50%, 50%,0)

使用方式:

- transform: translate(120px, 50%):位移
- transform: scale(2, 0.5):缩放
- transform: rotate(0.5turn):旋转
- transform: skew(30deg, 20deg):倾斜

animation 动画

animation-name:动画名称

animation-duration:动画持续时间

animation-timing-function:动画时间函数

animation-delay:动画延迟时间

animation-iteration-count:动画执行次数, 可以设置为一个整数,也可以设置为infinite,意思是无限循环

animation-direction:动画执行方向

animation-play-state:动画播放状态

animation-fill-mode:动画填充模式

渐变

颜色渐变是指在两个颜色之间平滑的过渡, css3渐变包括

- linear-gradient: 线性渐变

background-image: linear-gradient(direction, color-stop1, color-stop2, .;

- radial-gradient: 径向渐变

linear-gradient(Odeg, red, green);

其他

关于css3其他的新特性还包括flex弹性布局、Grid栅格布局, 这两个布局在以前就已经讲过, 这里就不再展示

除此之外, 还包括多列布局、媒体查询、混合模式等等....

面试官: 如果要做优化, CSS提高性能的方法有哪些?

实现方式有很多种, 主要有如下:

内联首屏关键CSS

异步加载CSS

资源压缩

合理使用选择器

减少使用昂贵的属性

不要使用@import

其他

减少重排操作, 以及减少不必要的重绘

了解哪些属性可以继承而来,避免对这些属性重复编写

cssSprite,合成所有icon图片,用喷高加上background-position的背景图方式显现出我们要的icon图,减少了http请求

把小的icon图片转成base64编码

CSS3动画或者过渡尽量使用transform和opacity来实现动画,不要使用left和top属性

css实现性能的方式可以从选择器嵌套、属性特性、减少http这三面考虑, 同时还要注意css代码的加载顺序

面试官: 让Chrome支持小于12px 的文字方式有哪些? 区别?

常见的解决方案有：

Zoom

zoom 的字面意思是“变焦”，可以改变页面上元素的尺寸，属于真实尺寸

其支持的值类型有：

zoom:50%，表示缩小到原来的一半

zoom:0.5，表示缩小到原来的一半

使用 zoom 来“支持”12px 以下的字体

-webkit-transform:scale()

针对chrome浏览器,加webkit前缀，用transform:scale()这个属性进行放缩

注意的是，使用scale属性只对可以定义宽高的元素生效，所以，下面代码中将span元素转为行内块元素

-webkit-text-size-adjust:none

该属性用来设定文字大小是否根据设备(浏览器)来自动调整显示大小

属性值：

percentage：字体显示的大小；

auto：默认，字体大小会根据设备/浏览器来自动调整；

none:字体大小不会自动调整

Zoom非标属性,有兼容问题，缩放会改变了元素占据的空间大小,触发重排

-webkit-transform:scale (0)大部分现代浏览器支持,并且对英文数字、蚊也能够生效,缩放不会改变了元素占据的空间大小，页面布

局不会发生变化

-webkit-text-size adjust对谷歌浏览器有版本要求,在27之后，就取消了该属性的支持,粗只对英文、数字生效