目录

[一、平面转换 1](#_Toc18198)

[1. 字体图标-Iconfont 1](#_Toc5075)

[2.平面转换概念 1](#_Toc8668)

[3.位移 2](#_Toc24306)

[4.位移-绝对定位居中 2](#_Toc26195)

[5.旋转 2](#_Toc30649)

[6. 转换原点 3](#_Toc25241)

[7. 多重转换 3](#_Toc17328)

[8. 缩放 3](#_Toc32374)

[9.渐变背景 3](#_Toc10867)

[二、 空间转换 4](#_Toc28221)

[1.空间位移 4](#_Toc30090)

[2. 透视 4](#_Toc10709)

[3. 空间旋转 4](#_Toc10197)

[4. 立体呈现 5](#_Toc12513)

[5.空间缩放 5](#_Toc20870)

[三、动画 5](#_Toc11623)

[1.动画的实现步骤 6](#_Toc19651)

[2.动画属性 6](#_Toc20579)

[3.逐帧动画 7](#_Toc28763)

[4.多组动画 7](#_Toc24607)

[四、视口 7](#_Toc28988)

[五、二倍图 8](#_Toc28250)

[六、 百分比布局 8](#_Toc12430)

[七、Flex布局 8](#_Toc7288)

[1. 主轴对齐方式 9](#_Toc25496)

[2.侧轴对齐方式 9](#_Toc17999)

[3.伸缩比 10](#_Toc11227)

[4.主轴方向 10](#_Toc28754)

[5. 弹性盒子换行 10](#_Toc10509)

[八． 移动适配 11](#_Toc31786)

[1. rem 11](#_Toc7126)

[2. 移动适配—媒体查询 11](#_Toc32508)

[3.rem适配原理 12](#_Toc10104)

[4.flexible移动适配 12](#_Toc12010)

[九、 Less语法 12](#_Toc29138)

[十一、 响应式网页 15](#_Toc18834)

[1. 媒体查询 15](#_Toc7258)

[2. 书写顺序 16](#_Toc9113)

[3. 关键词 16](#_Toc9730)

[4. 媒体类型 16](#_Toc20873)

[5. 媒体特性 16](#_Toc1313)

[6. 媒体查询-link写法 16](#_Toc7295)

[十二、 BootStrap 17](#_Toc478)

**一、平面转换**

1. **字体图标-Iconfont**

①目标:使用字体图标技巧实现网页中简洁的图标效果

案例:淘宝购物车

布局标签

·li > span \* 3

·字体图标

引入字体图标样式表

购物车和箭头span调用字体图标类名

9_AUBP[OCJ~T5~]}AS69]M8

②上传矢量图:

思考:如果图标库没有项目所需的图标怎么办?

答: IconFont网站上传矢量图生成字体图标

1.与设计师沟通，得到SVG矢量图

2. lconFont网站上传图标，下载使用

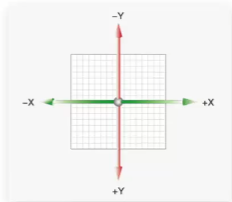
**2.平面转换概念**

目标:使用transform属性实现元素的位移、旋转、缩放等效果

·平面转换

改变盒子在平面内的形态（位移、旋转、缩放)

2D（D:direction）转换



·平面转换属性

Transform

**3.位移**

目标:使用translate实现元素位移效果

·语法：

transform: translate(水平移动距离,垂直移动距离);

·取值(正负均可)

像素单位数值

百分比(参照物为盒子自身尺寸)

注意:x轴正向为右，Y轴正向为下

·技巧：

Translate()如果只给出一个值表示x轴方向移动距离

单独设置某个方向的移动距离:translateX() & translateY()

**4.位移-绝对定位居中**

目标:使用translate快速实现绝对定位的元素居中效果



**5.旋转**

目标:使用rotate实现元素旋转效果

·语法：transform: rotate(角度);

注意:角度单位是deg

·技巧:取值正负均可

取值为正，则顺时针旋转

取值为负，则逆时针旋转

1. **转换原点**

目标∶使用transform-origin属性改变转换原点

·语法：

默认原点是盒子中心点

transform-origin:原点水平位置 原点垂直位置;

·取值：

方位名词( left. top.right、 bottom. center )

像素单位数值

百分比(参照盒子自身尺寸计算)

1. **多重转换**

目标:使用transform复合属性实现多形态转换

·多重转换技巧

transform: translate() rotate();

rotate必须放后面，因为旋转可以改变坐标轴向

1. **缩放**

目标:使用scale改变元素的尺寸

·语法

transform: scale(x轴缩放倍数, y轴缩放倍数);

·技巧

—般情况下,只为scale设置一个值,表示x轴和y轴等比例缩放

常用：transform: scale(缩放倍数);

数字大于1表示放大，小于1表示缩小。

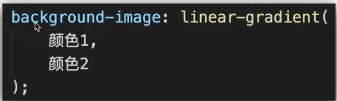
**9.渐变背景**

目标:使用background-image属性实现渐变背景效果

·渐变是多个颜色逐渐变化的视觉效果

·一般用于设置盒子的背景

·用法



1. **空间转换**

目标:使用transform属性实现元素在空间内的位移、旋转、缩放等效果

·空间:是从坐标轴角度定义的。 x、y 和z三条坐标轴构成了一个立体空间，z轴位置与视线方向相同。

·空间转换也叫3D转换

·属性: transform



**1.空间位移**

目标:使用translate实现元素空间位移效果

·语法： transform: translate3d(x. y,z);

transform: translateX(值);

transform: translateY(值);

transform: translateZ(值);

·取值（正负均可)

像素单位

数值百分比

1. **透视**

目标:使用perspective属性实现透视（近大远小，近实远虚）效果

·属性（添加给父级元素)

perspective:值;

·取值∶像素单位数值，数值一般在800 - 1200.

·透视距离也称为视距，所谓的视距就是人的眼睛到屏幕的距离。

1. **空间旋转**

目标:使用rotate实现元素空间旋转效果

·语法：transform:rotateZ(值);

transform:rotateX(值);

transform:rotateY(值);

·左手法则：判断旋转方向:左手握住旋转轴,拇指指向正值方向,手指弯曲方向为旋转正值方向

·拓展

rotate3d(x, y.z.角度度数)︰用来设置自定义旋转轴的位置及旋转的角度

x，y，z取值为0-1之间的数字

1. **立体呈现**

目标:使用transform-style: preserve-3d呈现立体图形

·思考:使用perspective透视属性能否呈现立体图形?

答:不能,perspective只增加近大远小、近实远虚的视觉效果。

·实现方法：

添加transform-style preserve-3d;

使子元素处于真正的3d空间

默认值flat,表示子元素处于2D平面内呈现

**5.空间缩放**

目标:使用scale实现空间缩放效果

·语法：

transform: scaleX(倍数);

transform: scaleY(倍数);

transform: scaleZ(倍数);

transform: scale3d(x, y, Z);

**三、动画**

目标:使用animation添加动画效果

**·思考:过渡可以实现什么效果?**

**答:实现2个状态间的变化过程**

·**动画效果∶实现多个状态间的变化过程**，动画过程可控（重复播放、最终画面、是否暂停)

**·**动画的本质是快速切换大量图片时在人脑中形成的具有连续性的画面

**·**构成动画的最小单元:帧或动画帧

**1.动画的实现步骤**

目标:使用animation添加动画效果

实现步骤：

①定义动画

两个状态的动画 多个状态的动画



②使用动画：

5`%G~EL202AE9([)I~FFRK6

**2.动画属性**

目标:使用animation相关属性控制动画执行过程

LTTJLK9L9)5GVTTS3R%{P25

**·**注意︰

动画名称和动画时长必须赋值

取值不分先后顺序

如果有2个时间值，第一个时间表示动画时长，第二个时间表示延迟时间

速度曲线：linear steps（分为几等分就直接写数字）

延迟时间：直接秒数+s

重复次数：直接写数字 无限循环：infinite

动画方向：alternate

执行完毕时状态：结束的状态：forwards 最初的状态：backwards



**3.逐帧动画**

目标:使用steps实现逐帧动画

**·**逐帧动画:帧动画。开发中，一般配合精灵图实现动画效果。

**·**animation-timing-function : steps(N) ;

将动画过程等分成N份

**·**精灵动画制作步骤：

**·**准备显示区域

设置盒子尺寸是一张小图的尺寸，背景图为当前精灵图

**·**定义动画

改变背景图的位置（移动的距离就是精灵图的宽度)

**·**使用动画

添加速度曲线steps(N)，N与精灵图上小图个数相同

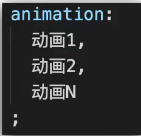
添加无限重复效果

**4.多组动画**

目标∶能够使用animation属性给一个元素添加多个动画效果

**·**思考∶如果想让小人跑远一些，该如何实现?

答∶精灵动画的同时添加盒子位移动画。



**四、视口**

目标:使用meta标签设置视口宽度，制作适配不同设备宽度的网页

·目标:网页宽度和设备宽度(分辨率)相同。

解决办法:添加视口标签。

**五、二倍图**

目标:能够使用像素大厨软件测量二倍图中元素的尺寸

·二倍图的作用：让图片分辨率更高，为了高分辨率下图片不会模糊失真。

·要基于逻辑分辨率去书写代码，在使用像素大厨测量二倍图的时候，要在最顶部设计图调整倍数。



1. **百分比布局**

目标:能够使用百分比布局开发网页

·百分比布局，也叫流式布局

·效果:宽度自适应，高度固定。

**七、Flex布局**

目标:能够使用Flex布局模型灵活、快速的开发网页

·思考

·多个盒子横向排列使用什么属性?

浮动

·设置盒子间的间距使用什么属性?

margin

·需要注意什么问题?

浮动的盒子脱标

·Flex布局/弹性布局:

是一种浏览器提倡的布局模型

布局网页更简单、灵活

避免浮动脱标的问题

·作用

①基于Flex精确灵活控制块级盒子的布局方式，避免浮动布局中脱离文档流现象发生.

②Flex布局非常适合结构化布局

·设置方式

父元素添加display: flex，子元素可以自动的挤压或拉伸

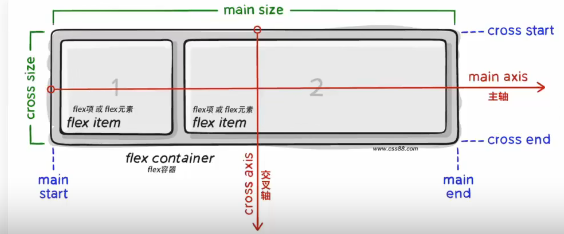
·组成部分：

弹性容器

弹性盒子

主轴

侧轴/交叉轴



1. **主轴对齐方式**

目标:使用justify-content调节元素在主轴的对齐方式

在Flex布局模型中﹐调节主轴或侧轴的对齐方式来设置盒子之间的间距**。**

·修改主轴对齐方式属性:justify-content

·先确定主轴方向再选择对应的属性实现主轴或侧轴的对齐方式。



**2.侧轴对齐方式**

目标:使用align-items调节元素在侧轴的对齐方式

·修改侧轴对齐方式属性:

align-items（添加到弹性容器)

align-self :控制某个弹性盒子在侧轴的对齐方式（添加到弹性盒子)

·先确定主轴方向再选择对应的属性实现主轴或侧轴的对齐方式。



**3.伸缩比**

目标:使用flex属性修改弹性盒子伸缩比

·属性

flex:值;

·取值分类

数值(整数)

·注意:只占用父盒子剩余尺寸

**4.主轴方向**

目标:使用flex-direction改变元素排列方向

·主轴默认是水平方向，侧轴默认是垂直方向

·修改主轴方向属性:flex-direction



1. **弹性盒子换行**

目标:使用flex-wrap实现弹性盒子多行排列效果

·思考:默认情况下，多个弹性盒子如何显示?

·弹性盒子换行显示:flex-wrap: wrap;

·调整行对齐方式: align-content

取值与justify-content基本相同

1. **移动适配**
2. **rem**

目标:能够使用rem单位设置网页元素的尺寸

·rem单位

相对单位

rem单位是相对于HTML标签的字号计算结果

1rem = 1HTML字号大小

1. **移动适配—媒体查询**

目标：能够使用媒体查询设置差异化CSS样式

·媒体查询能够检测视口的宽度，然后编写差异化的CSS样式

·当某个条件成立,执行对应的CSS样式

·写法：



·目前rem布局方案中，将网页等分成10份，HTML标签的字号为视口宽度的1/10



3.rem适配原理

·目标:实现在不同宽度的设备中，网页元素尺寸等比缩放效果

·目标:计算68px是多少个rem?(假定设计稿适配375px视口)

N\* 37.5 = 68→N= 68/ 37.5

·rem单位尺寸

1．确定设计稿对应的设备的HTML标签字号

查看设计稿宽度→确定参考设备宽度(视口宽度)→确定基准根字号(（1/10视口宽度）

2. rem单位的尺寸

rem单位的尺寸= px单位数值/基准根字号

4.flexible移动适配

目标:使用flexible js配合rem实现在不同宽度的设备中，网页元素尺寸等比缩放效果

·flexible.,js是手淘开发出的一个用来适配移动端的js框架。

·核心原理就是根据不同的视口宽度给网页中html根节点设置不同的font-size。

1. **Less语法**

目标:使用Less运算写法完成px单位到rem单位的转换

目标∶使用Less语法快速编译生成CSS代码

·思考︰在px单位转换到rem单位过程中，哪项工作是最麻烦的?

·答︰除法运算。CSS不支持计算写法。

·解决方案:可以通过Less实现。

·Less是一个CSS预处理器,Less文件后缀是.less

注意:浏览器不识别Less代码，目前阶段，网页要引入对应的CSS文件。

**1.编译插件**

目标∶使用Less语法快速编译生成CSS代码

Easy Less :

vscode插件

作用:less文件保存自动生成css文件

**2.Less语法注释**

目标∶使用Less语法快速编译生成CSS代码

注释:

·单行注释

语法://注释内容

快捷键:ctrl + /

·块（多行）注释

语法:\*注释内容\*/

快捷键: shift + alt + A

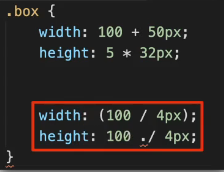
1. **Less语法计算**

目标:使用Less运算写法完成px单位到rem单位的转换

运算:

·加、减、乘直接书写计算表达式

·除法需要添加小括号或.



·注意:除法需要添加小括号或.

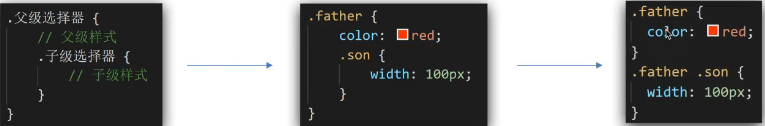
1. **Less语法嵌套**

目标∶能够使用Less嵌套写法生成后代选择器

·嵌套:

作用：快速生成后代选择器。

语法：

·注意:&不生成后代选择器，表示当前选择器，通常配合伪类或伪元素使用

**5.Less语法变量**

目标:能够使用Less变量设置属性值

变量

·方法二:把颜色提前存储到一个容器，设置属性值为这个容器名

·变量:存储数据，方便使用和修改。

·语法:

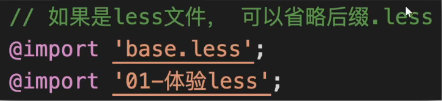
定义变量:@变量名:值;

使用变量:CSS属性:@变量名;

**6.Less导入**

目标︰能够使用Less导入写法引用其他Less文件

·导入: @import “文件路径";



**7.Less导出**

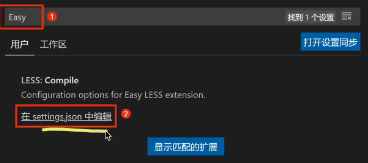
目标:使用Less语法导出CSS文件

·方法一∶

配置EasyLess插件，实现所有Less有相同的导出路径

·配置插件︰设置→搜索EasyLess→在setting.json中编辑→添加代码（注意，必须是双引号)

·语法：”out”: “../css/”



MXI{Z@6R]{()9BHD0EL{`TO

**8.Less单独导出**

在第一行加

语法：①导入到文件夹：// out: ./css/

②导入到具体文件：// out: ./css/zjd.css

A891G{PY5$G5)LO)N6]76RT

1. **Less禁止导出**

·禁止导出

在less文件第一行添加: // out: false

]3{7R[}E4343LMG%5PT79EQ

1. **VW / VH**

目标:能够使用w单位设置网页元素的尺寸

·相对单位

·相对视口的尺寸计算结果

·vw : viewport width

·vh : viewport height

VW = 1/100视口宽度

VH =1/100视口高度

1. **VW适配原理**

目标:实现在不同宽度的设备中，网页元素尺寸等比缩放效果

·vw单位尺寸

1. 确定设计稿对应的vw尺寸( 1/100视口宽度)

查看设计稿宽度→确定参考设备宽度(视口宽度)→确定vw尺寸(1/100视口宽度）

2. vw单位的尺寸= px单位数值/(1/100视口宽度)

1. **响应式网页**
2. **媒体查询**

·基本语法

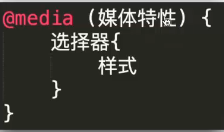
目标:能够根据设备宽度的变化，设置差异化样式

·开发常用写法

媒体特性常用写法

max-width

min-width



·完整写法：

_N%GW]IEM{30TR6XM}BLW7R

1. **书写顺序**

目标︰能够根据设备宽度的变化，设置差异化样式

min-width (从小到大)

max-width (从大到小)

1. **关键词**

目标:能够根据设备宽度的变化，设置差异化样式

and

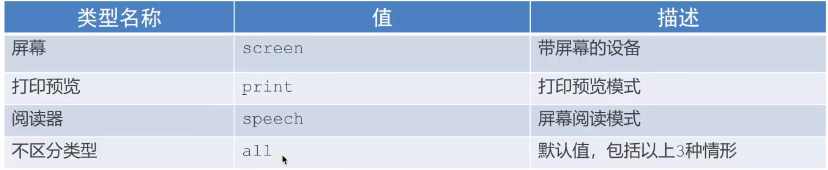
only

not

1. **媒体类型**

目标:能够根据设备宽度的变化，设置差异化样式

·媒体是用来区分设备类型的，如屏幕设备、打印设备等，其中手机、电脑、平板都属于屏幕设备。



1. **媒体特性**

目标:能够根据设备宽度的变化，设置差异化样式

·媒体特性主要用来描述媒体类型的具体特征，如当前屏幕的宽高、分辨率、横屏或竖屏等。



1. **媒体查询-link写法**

目标:能够根据设备宽度的变化，设置差异化样式

·外模式CSS引入

FA5{`@@Z_3IN4K2`L9X(GDG

1. BootStrap

目标∶使用BootStrap框架快速开发响应式网页

·Bootstrap是由Twitter公司开发维护的前端U框架，它提供了大量编写好的CSS样式，允许开发者结合一定HTML结构及JavaScript，快速编写功能完善的网页及常见交互效果。

·中文官网: https://www.bootcss.com/

**1.BootStrap使用步骤**

目标:使用BootStrap框架快速布局响应式网页

1.引入: BootStrap提供的CSS代码

<link rel="stylesheet" href="./bootstrap-3.3.7/css/bootstrap.css">

2．调用类:使用Bootstrap提供的样式

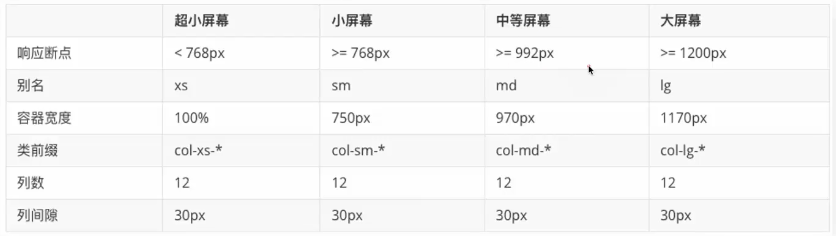
container:响应式布局版心类

**2.BootStrap栅格系统**

目标:使用BootStrap栅格系统布局响应式网页

·栅格化是指将整个网页的宽度分成若干等份

·BootStrap3默认将网页分成12等份



· .container是Bootstrap中专门提供的类名，所有应用该类名的盒子，默认已被指定宽度且居中。

·.container-fluid也是Bootstrap 中专门提供的类名，所有应用该类名的盒子，宽度均为100%。

·分别使用.row类名和.col类名定义栅格布局的行和列。

注意:

1.container类自带间距15px;

2.row类自带间距-15px

**3.全局样式**

目标:掌握BootStrap手册用法，使用全局CSS样式美化标签

手册用法︰

·BootStrap预定义了大量类用来美化页面，掌握手册的查找方法是学习全局样式的重点。

·网站首页→Bootstrap3中文文档-全局css样式-按分类导航查找目标类

**4.组件**

目标:使用BootStrap组件快速布局网页

·组件

BootStrap提供的常见功能，包含了HTML结构和CSS样式。

·使用方法

引入BootStrap样式

复制结构﹐修改内容



**5.字体图标**

目标:使用Glyphicons字体图标实现网页中的图标效果

·Glyphicons字体图标的使用步骤

HTML页面引入BootStrap样式文件

空标签调用对应类名

Glyphicon

图标类

**6.插件**

目标:使用BootStrap插件实现常见的交互效果

·插件

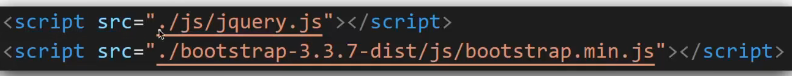
BootStrap提供的常见效果,包含了HTML结构，CSS样式与JavaScript



·插件的使用步骤：

引入BootStrap样式

引入js文件: jQuery.js + BootStrap.min.js



复制HTMIL结构,并适当调整结构或内容