# 1.HTML&CSS

## HTML中元素塌陷问题解决方案

### 1.问题描述：

如果父元素只包含了浮动元素 ，那么父元素在未设置高度的同时，则父元素高度塌缩为零;

### 解决方案:

#### 使用css方式;

1. 创建一个用来清除浮动的css样式类（.clearfix）；
2. 针对包裹的全是浮动元素的父级容器使用;

代码实现:

.clearfix{zoom:1}

.clearfix:after{

content:".";

display:block; visibility:hidden;

height:0;

clear:both;

}

/\*

\* zoom:1;IE浏览器的专用属性（针对IE）

\* :after{} 伪对象选择符 （在这个对象被浏览器渲染后添加一定的内容）

\* content:"."; 添加内容写在这里的属性值 这个属性是专门配合伪类对象,必须写;

\* display:block;将添加的内容转换为块级元素

\* visibility:hidden;可视化属性；控制元素是否可见;无论是否可见，保留物理空间；和display属性不同;

\* height:0;将添加进去的内容的高度设置为0；消除其占位；

\* 将添加去的内容作为清除浮动的对象，实现外围对象中有内容存在，因此可以自动判断高度;（告诉浮动元素说不要出现在我的两边）

\* \*/

#### 2.使用css+html

在父元素的最后单独放一个块级元素来清除浮动

.clear{

clear: both;

}

<div class="cotainer">

<div class="fl col1">left</div>

<div class="fl col2">right</div>

<div class="clear"></div>

</div>

## 2.边距的设置

### 1.padding

### 2.margin

## 3.元素的居中

#### 1.通过width和margin属性

a.设置margin必须指定宽，否则无效

.wrapper{

width:1200px;

margin:0 auto;

}

#### 2.通过line-height和height

b.使元素的line-height和height的值相同;

input{

height:50px;

line-height:50px;

}

#### 3.使用padding属性

c.使用padding

适合高度固定的元素;

.inner{

height: 52px;

box-shadow: 0 1px 3px rgba(26,26,26,0.1);

}

.link{

display: inline-block;

padding: 14px 0;

}

<div class="inner">

<a href="#" class="link">首页</a>

<a href="#" class="link">发现</a>

</div>

## CSS盒子模型

1. 3.相关博客：
2. <https://blog.csdn.net/qq_29393273/article/details/83690676>

## Inherit关键字

**1. 指定一个属性应从父元素继承它的值。**

**inherit 关键字可用于任何 HTML 元素上的任何 CSS 属性;**

color:inherit; 规定应该从父元素继承颜色

## 7.overflow属性

规定当内容溢出元素框时发生的事情；

|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

## 8.text-overflow

规定当文本溢出包含元素时发生的事情。

|  |  |  |
| --- | --- | --- |
| **值** | **描述** | **测试** |
| clip | 修剪文本。 | [测试](https://www.w3school.com.cn/tiy/c.asp?f=css_text-overflow) |
| ellipsis | 显示省略符号来代表被修剪的文本。 | [测试](https://www.w3school.com.cn/tiy/c.asp?f=css_text-overflow&p=2) |
| *string* | 使用给定的字符串来代表被修剪的文本。 |  |

## 9.white-space 属性

设置如何处理元素内的空白。

|  |  |
| --- | --- |
| normal | 默认。空白会被浏览器忽略。 |
| pre | 空白会被浏览器保留。其行为方式类似 HTML 中的 <pre> 标签。 |
| nowrap | 文本不会换行，文本会在在同一行上继续，直到遇到 <br> 标签为止。 |
| pre-wrap | 保留空白符序列，但是正常地进行换行。 |
| pre-line | 合并空白符序列，但是保留换行符。 |
| inherit | 规定应该从父元素继承 white-space 属性的值。 |

## 5.图片的尺寸指定方法:

### a.通过img标签的width和height属性指定;

<img src="#" width="200" height="200" alt="">

1.可使用css改变图片的大小;

img{

width:210px;

height:210px;

}

### b.通过css指定;

b1.直接指定图片的大小

.image img{

widht:100px;

height:100px;

}

b2.通过父元素的宽和高

1.指定父元素的的宽和高

2.指定img的宽和高为100%

.image{

width:120px;

height:120px;

}

.image img{

widht:100%;

height:100%;

}

<div class="image">

<a href="#"><img src="#" alt=""></a>

</div>

## 3.box-sizing解决div高度被内边距（padding）撑开问题;

## 4.Display属性

display 属性规定元素应该生成的框的类型。

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 此元素不会被显示。 |
| block | 此元素将显示为块级元素，此元素前后会带有换行符。 |
| inline | 默认。此元素会被显示为内联元素，元素前后没有换行符。 |
| inline-block | 行内块元素。（CSS2.1 新增的值） |
| inherit | 规定应该从父元素继承 display 属性的值。 |

<https://www.cnblogs.com/ZheOneAndOnly/p/10847591.html>

## 5. flex-shrink

<https://www.jianshu.com/p/d9373a86b748/>

知乎有使用;

## 6.CSS中的关键字

### 1.Inherit

## CSS基本的文本属性

## 垂直居中的方法

1. 高度固定的情况：

使用padding

## vertical-align: 属性的使用;

## css绘制三角形

## 7.CSS3中可能会用到的一些属性

### -webkit-tap-highlight-color：

<https://www.cnblogs.com/libin-1/p/5903350.html>

知乎有使用;

## 怎么用css去掉chrome浏览器中input获得焦点时的带颜色边框呢

<https://blog.csdn.net/u010874036/article/details/51224793>

<https://www.jianshu.com/p/9a463d50e441>

## line-height属性

# 布局

## 网站参考

### 京东[https://www.jd.com](https://www.jd.com/)

红色和白色为主色调；

### 同城旅游<https://www.ly.com/>

颜色较丰富；（绿色和白色为主色调）

### 知乎[https://www.zhihu.com](https://www.zhihu.com/)

蓝色和白色为主色调

### CSDN博客<https://www.csdn.net/>

红色和白色为主色调

### 开源中国 <https://www.oschina.net/>

绿色和白色为主色调

## 字体问题

## 备选网站名字

1. 旅行密码
2. 旅行攻略
3. 灵魂旅行
4. 享受旅行
5. 爱旅行
6. 旅行
7. 去旅行

## 网站logo

# JavaScript

## 1.JavaScript的数据类型

## 2.JavaScript对象（迭代，取值，赋值）

## 3.JavaScript Array对象

## 4.JSON及JSON和JavaScript对象的区别;

## 5.JavaScript 三元表达式

## 6.JavaScript 正则表达式

## 7.JavaScript函数的声明方式和回调函数

## 8.JavaScript 小数运算处理方法

# 3.Jquery

## 1.$.extend()方法的使用和实际开发中的的使用场景;

## 2.对Ajax的封装

## 3.Jquery 全局函数

# EasyUI

## 1.EasyUI中datagrid数据网格的使用

## 2.EasyUI中Tree树的使用

## 3.EasyUI中 Layout布局的使用

## 4.EasyUI中 Tabs选项卡的使用

## 5.EasyUI中的消息框

## 6.EasyUI中的对话框

# 5.Springboot

## Springboot的配置文件

相关资料: <https://blog.csdn.net/qq_29393273/article/details/87956267>

## Springboot处理静态资源

相关资料: <https://blog.csdn.net/qq_29393273/article/details/88076194>

## 3.Springboot整合数据库和mybatis-plus

### 1.Springboot中整合数据库的步骤

1.导入相关数据库的驱动包和mybatis-plus的jar包

2.在springboot配置文件中配置数据库和mybatis-plus相关信息;

3.在启动类上加入 @MapperScan("com.itour.presist") 将mappper注入Spring;

### 2.代码:

### 3.相关资料:

<https://blog.csdn.net/f45056231p/article/details/83817740>

## 4.Springboot 整合beetl

#### 1.Springboot整合beetl的步骤;

1.导入beetl的jar

2.编写配置类(beetlConfig.java)

3.编写beetl.properties配置文件

#### 2.代码实现;

## 5.Springboot 整合SpringCloud Eureka

#### 1.基本步骤

1.搭建Eureka server

1.创建eureka server 模板;

2. 需添加依赖spring-cloud-starter-netflix-eureka-server

3. 在springboot中配置eureka server

4. 在启动类中使用@EnableEurekaServer

2.搭建服务提供者

1.创建服务提供者模块;

2.需添加依赖: spring-cloud-starter-netflix-eureka-client

3.在springboot配置文件中配置eureka,将服务注册到到eureka server服务注册中心;

4.在启动类中使用@EnableEurekaClient

3.搭建服务消费者

1.创建消费模块;

2.在springboot配置文件中配置eureka,来订阅eureka server服务注册中心的服务；

#### 2.代码

## 6.Springboot整合Spring Cloud OpenFeign

### 1.Mave多模块整合步骤;

1.需spring-cloud-dependencies和spring-cloud-starter-openfeign 的依赖;

2.公共接口端(itour-account-api)

1.使用@FeignClient(name = "itour-account-service")标记接口;name是服务提供者(itour-account-service)模块中在Springboot的配置文件中配置的spring.application.name；

2.服务提供者模块

实现公共接口;

3.服务消费者模块(itour-online-web)

1.在启动类上使用@EnableFeignClients启用feign客户端;

2.像传统的方式访问;

### 2.遇到的问题:

#### 1.公共接口对象注入失败;

**原因**：公共接口模板中的公共接口对象的包名和必须是消费者模块启动项目的包名相同或是启动类包的子包 ; com.itour🡪com.itour.api;当执行启动类时,springboot会将该包及子包下的对象注入给spring;

**解决办法**:将公共接口模块的对象所在包改为启动类中的包名相同或改为启动类的子包;

## 7.Springboot对日志的支持

## Springboot AOP统一管理web日志

## Springboot整合redis

### 1．基本步骤

1. 添加依赖spring-boot-starter-redis;
2. 在Springboot配置文件中配置redis相关信息;
3. 编写配置类

相关博客：<https://blog.csdn.net/weixin_40623736/article/details/98097708>

# 7.Java

## 1.Java中的异常

## 2.Java 面向对象

## 3.Java注解和自定义注解

## 4.Java的反射

# 8.Web开发中遇到到问题及解决方案

## 1.防重复提交

# 9.Spring

## 1.什么是Spring？

## 2.Spring的IOC操作

## 3.Spring对Bean的管理

## 4.Spring 的AOP的概念、基本原理、AOP操作；

# 10.数据库

## 1.MySQL数据库表设计

### 1.基本的数据类型

### 2.日期类型使用什么类型合适？

INT类型(占用资源少，查询速度快,使用较灵活)；INT(13)

DATETIME：固定格式YYYY-MM-DD HH:MM:SS，当格式比较固定时使用DATETIME;

## 2.公司数据库test

<spring.datasource.url>jdbc:oracle:thin:@10.103.32.24:1521:onlinetest</spring.datasource.url>

<spring.datasource.driverClassName>oracle.jdbc.driver.OracleDriver</spring.datasource.driverClassName>

<spring.datasource.username>onlinetest</spring.datasource.username>

<spring.datasource.password>onlinetest438</spring.datasource.password>

# 11.网页设计

## 1.版权

### 1.图片版权问题

## 2.字体

### 免费字体

### 收费字体

## 3.网站色调

**蓝色和白色为主色调;**