

# CSS-position

# 课程目标

- 学习CSS中的定位模型
- 使用定位模型来达到布局目的

# 课程安排

学习position之static

学习position之relative

学习position之absolute

学习position 之fixed

学习position之sticky

# 课程简介-position?

Positioned Layout Module

提供与 **元素定位**和**层叠相关** 功能 它是核心模块

```
.box{  
  position:relative;  
}
```

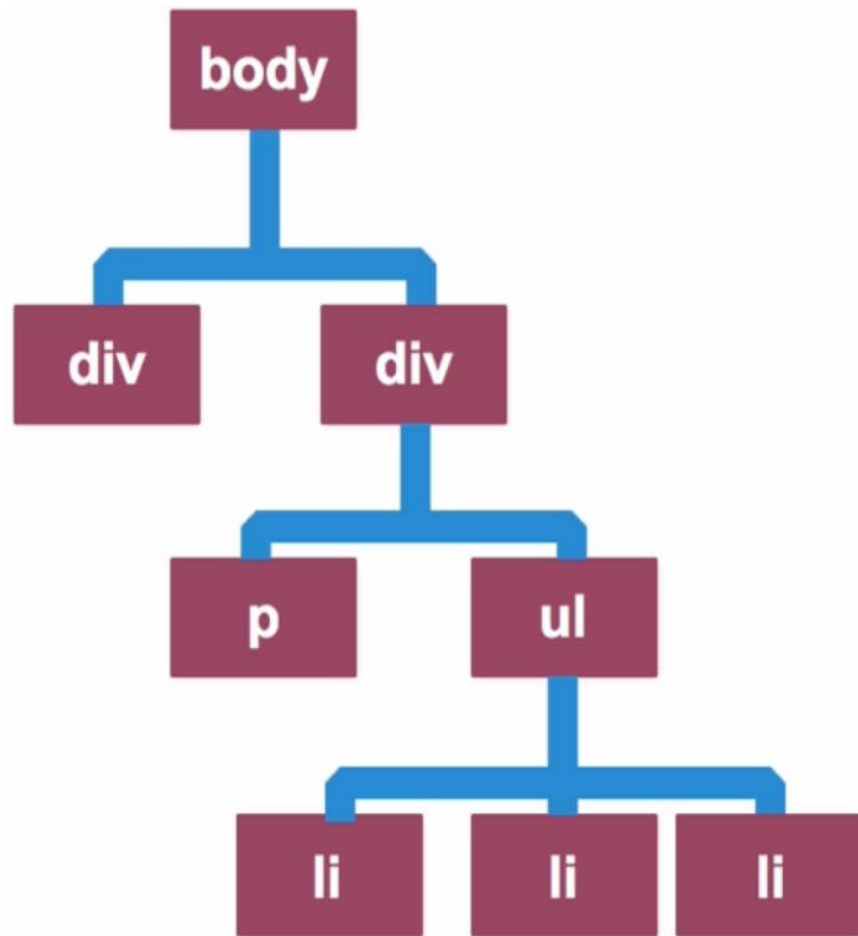
- (1)盒子模型的类型和尺寸
- (2)布局模型
- (3)元素之间的关系
- (4)视口大小、图像大小等

# 课程简介-小知识点

document tree

normal-flow

containing-block



# 课程简介- 定位模型

**static** 自然模型

**relative** 相对定位模型

**absolute** 绝对定位模型

**fixed** 固定定位模型

**sticky** 磁贴定位模型

```
.box{  
  position:relative;  
}
```



# CSS position 属性

static	relative
absolute	fixed
sticky	inherit

top	
left	right
bottom	
z-index	

# position之static

静态定位 / 常规定位 / 自然定位 —— 定位中的一股清流-回归本真

作用	使元素定位与 常规/自然流中 ( 块、行垂直排列下去、行内水平从左到右 )
特点	(1)忽略top,bottom,left,right或者z-index声明 (2)两个相邻的元素如果都设置了外边距，那么最终外边距=两者外边距中最大的 (3)具有固定width和height值的元素，如果把左右外边距设置为auto，则左右外边距会自动扩大占满剩余宽度。造成的效果就是这个块水平居中



# position之relative

## 相对定位

作用	使元素称为containing-block-官话是可定位的祖先元素
特点	<p>(1)可以使用top/right/bottom/left/z-index进行相对定位——<b>？相对的是谁</b></p> <p>(2)<b>相对定位的元素不会离开常规流</b>——心念家乡</p> <p>(3)任何元素都可以设置为relative，<b>它的绝对定位的后代都可以相对于它进行绝对定位</b>——超好用</p> <p>(4)可以使用浮动元素发生偏移，并控制它们的堆叠顺序</p>

# position之absolute

## 绝对定位

作用	使元素脱离常规流
特点	<p>(1)脱离常规流</p> <p>(2)设置尺寸要注意：<b>百分比比的是谁？——最近定位祖先元素</b></p> <p>(3)lr<b>tb</b>如果为0 <b>它将对齐到最近定位祖先元素的各边——衍生出一个居中妙计</b></p> <p>(4)lr<b>tb</b>如果设置为auto 它将被打回原形</p> <p>(5)如果没有最近定位祖先元素 会认&lt;body&gt;做爹爹（通俗的说） <b>如果元素没有已定位（除static 定位以外）的祖先元素，那么它的位置相对于最初的包含块</b></p> <p>(6)z-index可以控制堆叠顺序999999见过吧？</p>

# position之fixed

## 固定定位

作用	我跟绝对定位本是同根生相煎何太急
特点	(1)跟absolute有啥区别？ <b>相对于谁做绝对定位</b> (2) <b>固定定位元素不会随着视口滚动而滚动</b> (3)继承absolute特点

# position之sticky

磁贴定位/粘性定位/吸附定

——赛季新秀 实力布局糖

作用	relative+fixed的完美结合，制造出吸附效果
特点	(1)如果产生偏移 原位置还是会在常规流中，一亩三分地留着 (2)如果它的最近祖先有滚动 那么它的偏移标尺就是最近祖先元素 (3)如果最近祖先没有滚动 那么它的偏移标尺是视口 (4)上下左右的偏移规则

# position之总结

