

# 移动web-day02

空间转换 + 动画



# 目录

Contents

- ◆ 空间转换
- ◆ 动画
- ◆ 综合案例



# 目录

Contents

- ◆ 空间转换
- ◆ 动画
- ◆ 综合案例

# 一、空间转换

目标：使用 transform 属性 实现元素在空间内的 旋转、平移 等效果

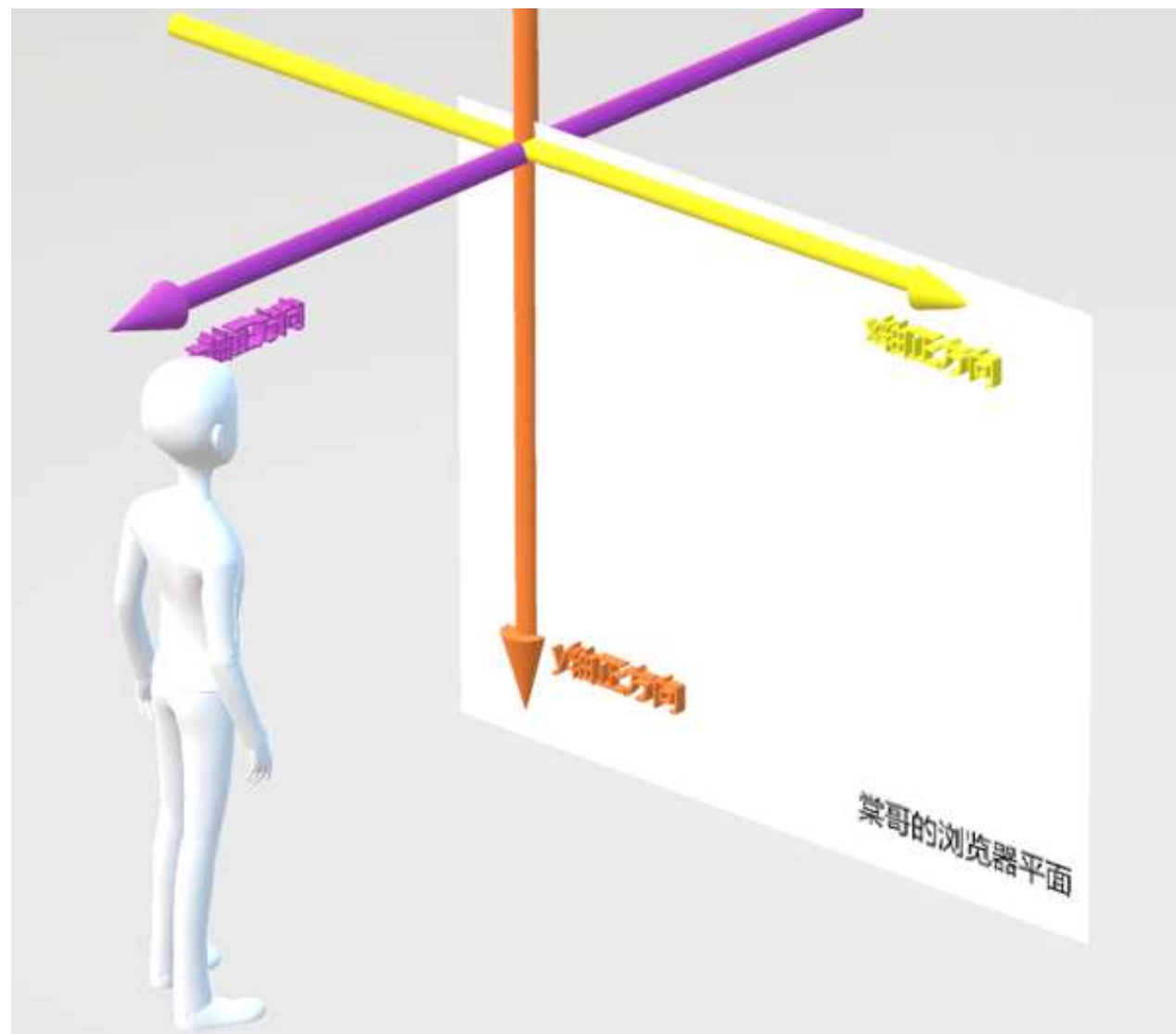
学习路径：

1. 空间坐标系
2. 视距
3. 3D旋转
4. 3D平移
5. transform-style

# 一、空间转换

## 1.1 3D坐标系

- 浏览器中存在立体3D坐标系
- 坐标轴：
  - X轴：水平向右是x轴正方向
  - Y轴：垂直向下是y轴正方向
  - Z轴：垂直于浏览器的平面，指向你脸的方向



# 一、空间转换

## 1.2 小结

- 浏览器中的3D坐标系有那几根轴，默认分别是什么方向？
  - X轴：水平向右
  - Y轴：垂直向下
  - Z轴：垂直于浏览器的平面，指向你脸的方向

# 一、空间转换

目标：使用 transform 属性 实现元素在空间内的 旋转、平移 等效果

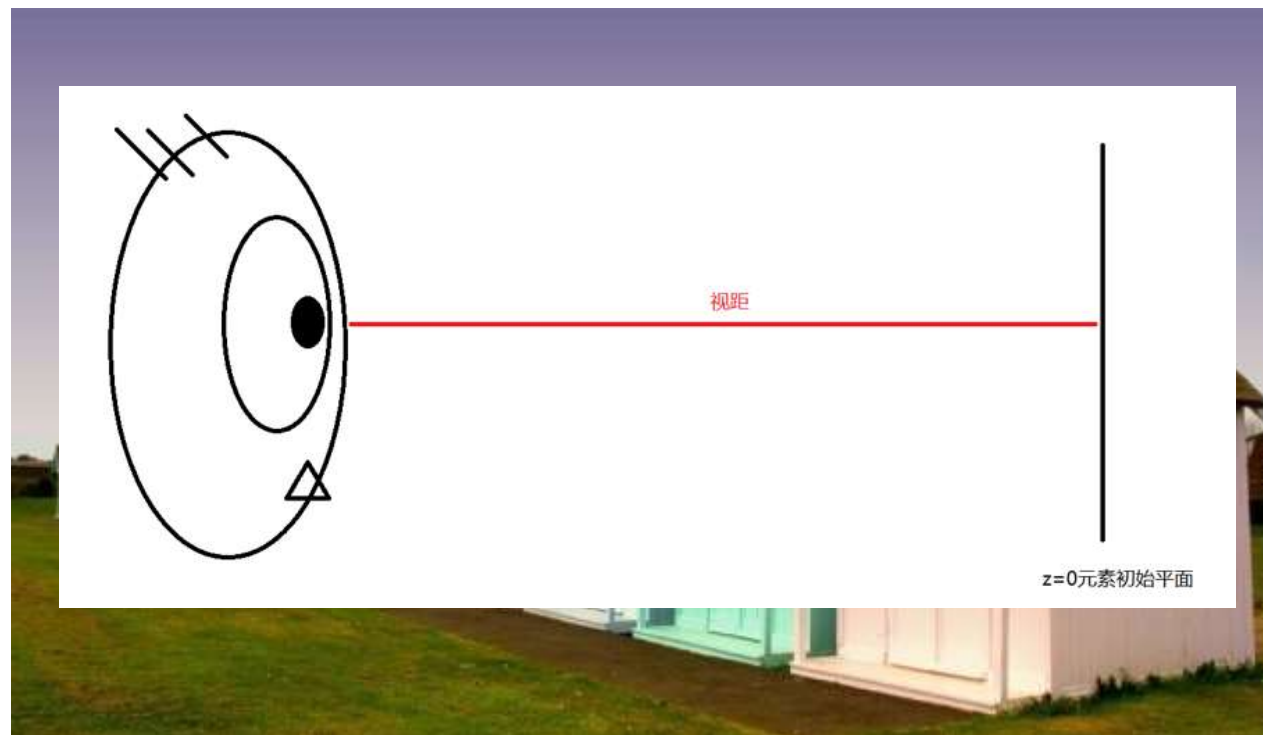
学习路径：

1. 空间坐标系
2. 视距
3. 3D旋转
4. 3D平移
5. transform-style

# 一、空间转换

## 2.1 视距、透视

- 需求：让图片沿着x轴进行旋转，此时会是空间3D的旋转，看一下此时图片是往里还是往外旋转的？
- 属性名：perspective
- 属性值：数字+px
- 注意点：
  - 视距给转换元素的父元素设置
- 作用：让子元素有近大远小的视觉效果
- 取值：观察者的眼睛距离元素初始平面的距离
  - 如果取值越小，表示距离越小，近大远小的视觉效果会？
  - 如果取值越大，表示距离越大，近大远小的视觉效果会？
  - 一般给1000px左右即可





## 2.2 小结

- 视距的属性名和属性值分别是什么？
  - 属性名：perspective
  - 属性值：1000px左右
- 视距的作用是什么？
  - 让子元素有近大远小的视觉效果
- 视距属性给哪个元素设置？
  - 给需要近大远小效果的父元素设置

# 一、空间转换

目标：使用 **transform** 属性 实现元素在空间内的 **旋转**、**平移** 等效果

学习路径：

1. 空间坐标系
2. 视距
3. **3D旋转**
4. 3D平移
5. transform-style



# 一、空间转换

## 3.1 3D旋转

- 作用：让元素进行空间上的旋转，提升网页效果
- 属性名：转换 transform
- 属性值：沿着x轴和y轴旋转，此时是3D空间旋转

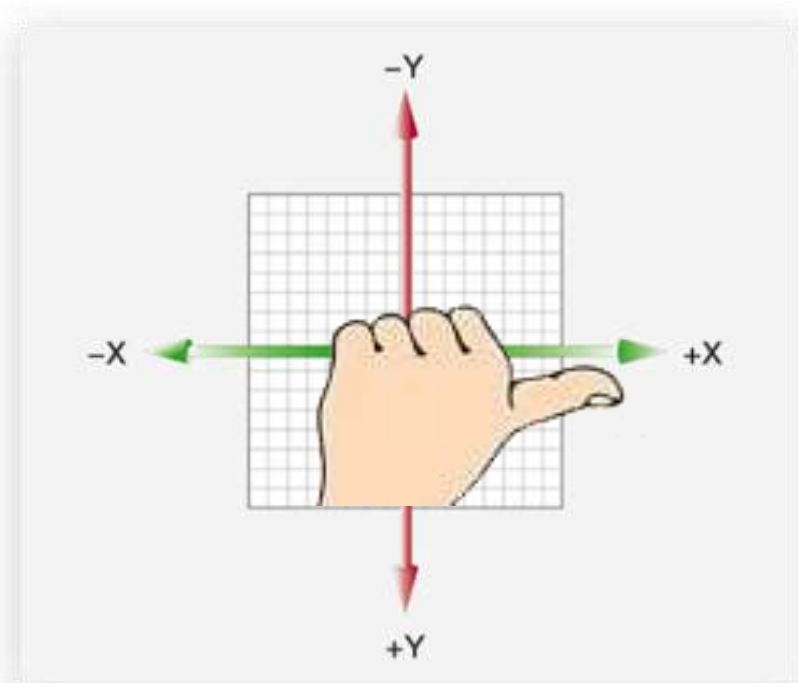
取值	效果
<code>rotateX(度deg)</code>	沿着x轴旋转
<code>rotateY(度deg)</code>	沿着y轴旋转

- 注意点：
  - `rotate(度deg)` 其实就默认在平面旋转的效果，和 `rotate(度deg)` 效果一致



## 3.2 左手法则

- 作用：通过左手法则可以判断元素旋转的方向
- 步骤：
  1. 掏出你的左手，左手握拳，伸出大拇指
  2. 让大拇指指向轴的正方向
  3. 此时四指弯曲的方向就是图片旋转的方向



## 3.3 小结

- 沿着哪根轴旋转时，会是3D旋转？
  - X轴 和 Y轴
- 3D旋转的属性名和属性值分别是什么？
  - 属性名：transform
  - 属性值：rotateX (度deg) 、 rotateY (度deg)

# 一、空间转换

目标：使用 transform 属性 实现元素在空间内的 旋转、平移 等效果

学习路径：

1. 空间坐标系
2. 视距
3. 3D旋转
4. 3D平移
5. transform-style

# 一、空间转换

## 4.1 3D平移

- 作用：让元素进行空间上的平移，提升网页效果
- 属性名：转换 transform
- 属性值：沿着z轴进行平移，此时是3D空间平移
  - transform : translateZ( 45px );

# 一、空间转换

## 4.2 3D平移的合写

- 需求：如果需要一个盒子同时往3个轴方向平移100px，如何实现呢？
- 方法一：多个取值的合写
  - `transform: translateX (100px) translateY (100px) translateZ (100px)`
- 方法二：3D平移的简写形式
  - `transform: translate3d (x, y, z)`



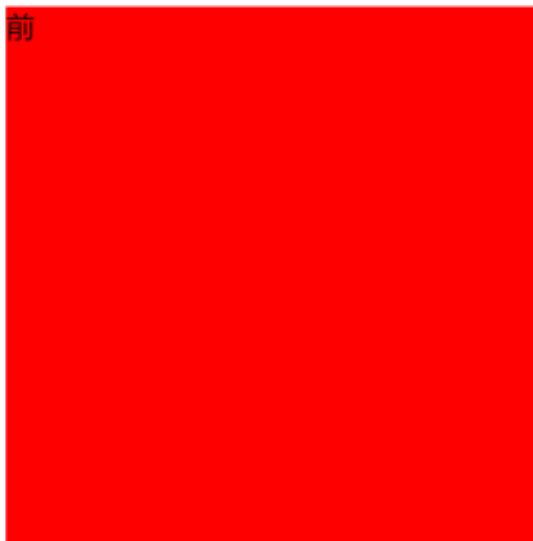
## 4.3 小结

- 沿着哪根轴平移时，会是3D平移？
  - Z轴
- 3D平移的属性名和属性值分别是什么？
  - 属性名：transform
  - 属性值：translateZ (平移的距离)

# 一、空间转换

## 综合案例1：立方体案例

- 需求：利用学习的3D旋转和3D平移属性，完成立方体案例



## 综合案例1：立方体案例-思路铺垫

- 问题：
  1. 当一个元素 **rotateY(45deg)** 是什么效果？z轴指向哪里？
  2. 当一个元素 **rotateY(90deg)** 是什么效果？z轴指向哪里？
- 结论：朝向的方向就是z轴，旋转时，面朝的方向永远是z轴

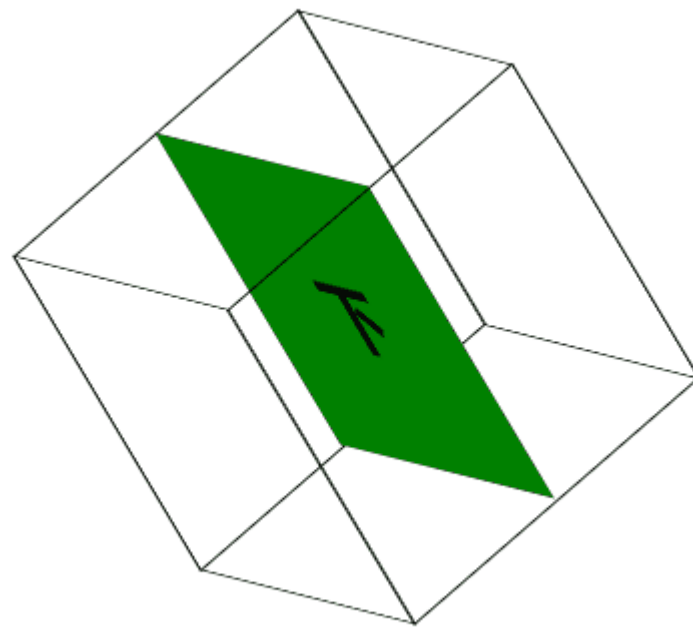
## 综合案例1：立方体案例-实现步骤

➤ 实现六个面的思路：

1. 先旋转到面朝对应的方向
2. 往面朝的方向（z轴）平移对应的距离即可

➤ 结构：

1. 整体是ul，每个面是li标签，然后每个面沿着不同方向旋转和平移，组成立方体



# 一、空间转换

目标：使用 **transform** 属性 实现元素在空间内的 **旋转、平移** 等效果

学习路径：

1. 空间坐标系
2. 视距
3. 3D旋转
4. 3D平移
5. **transform-style**

# 一、空间转换

## 5.1 transform-style

- 问题：使用perspective属性是否能实现真正的立体图形？
  - 答：不能，perspective只是给元素设置近大远小的视觉效果，并不是真正的3D立体元素
  - 如果需要对元素进行3D变换，此时需要使用transform-style
- 属性名：transform-style
- 属性值：
  - flat：子元素在2d平面中显示（默认值）
  - preserve-3d：子元素在3D空间中显示

# 一、空间转换

## 5.2 perspective和transform-style的区别

➤ 区别：

1. perspective：只是在平面中模拟出近大远小的效果，并不是真正的3D立体元素
2. transform-style：可以让其子元素在一个3D空间中布局，本身就是真正的3D元素

➤ 注意点：transform-style属性给谁设置？

- 需要变成3D元素的本身设置的。比如：立方体 ul 需要是一个真正的3D元素，所以直接给 ul 设置
- 有人会看做给面的父元素设置。比如：给每个面 li 的 父元素 设置，其实也是 ul 标签
- 说法不同，但是结果一致

## 5.3 小结

- 如果需要近大远小的效果，使用什么属性完成？给谁设置？
  - 视距：perspective
  - 给需要近大远小的父元素设置
- 如果需要让元素变成一个真正的3D元素，使用什么属性完成？给谁设置？
  - transform-style:preserve-3d
  - 给需要变成3d元素本身设置



# 一、空间转换

## 综合案例2：3D导航

- 需求：利用学习的3D旋转和3D平移属性，完成3D导航案例

首页

登录

注册

## 综合案例2：3D导航-思路分析

➤ 问题：绿色和橙色盒子是如何摆放的？

1. 绿色盒子是立方体的前面
2. 橙色盒子是立方体的上面

➤ 思路：

1. 整体是ul，每个li标签相当于立方体，li中有两个a标签（100\*40），两个a标签表示上面和前面
2. 通过定位先让两个a叠加，然后通过转换改变a标签面的位置（前面 + 上面）
3. 给li设置真3D：transform-style: preserve-3d
4. 当li被hover之后，让li标签整体往下旋转即可



# 一、空间转换

## 综合案例2：3D导航-思路分析



定位



橙色旋转 (X)



橙色位移 (Z)



绿色位移 (Z)



# 目录

Contents

- ◆ 空间转换
- ◆ 动画
- ◆ 综合案例

## 二、动画

目标：使用 **animation** 属性 实现元素的动画效果

学习路径：

1. 动画的初体验
2. 定义动画 @keyframes
3. 调用动画 animation

### 1.1 动画的初体验

- 需求：让一个div被鼠标悬停之后，旋转一圈
  - 如果需要一直不停的旋转呢？
- 过渡：
  - 不会主动执行，需要触发条件（如：hover）
  - 不能无限执行，最后肯定会停下来
  - 只能设置两个状态的样式不同
- 动画：
  - 可以主动执行（一刷新就动起来）
  - 可以无限执行，一直不停的运行下去
  - 可以设置多个状态的样式不同



### 1.2 实现动画的步骤-把旋转的盒子改写成动画的写法

#### 1. 定义动画：@keyframes 动画名 {}

```
@keyframes 自定义动画名 {  
  from {  
    开始状态的样式  
  }  
  to {  
    结束状态的样式  
  }  
}
```



#### 2. 调用动画：给对应的元素设置animation属性

```
.box {  
  animation: 动画名 动画执行的时间 ...  
}
```

### 1.3 小结

- 动画和过渡的区别有哪些?
  - 动画：不需要触发条件，一刷新就执行；可以永远执行下去；动画可以设置多个状态的不同
  - 过渡：需要有触发条件，比如：hover；最终肯定会停下来；只能设置两个状态
- 实现动画的步骤分为几步?
  - 定义动画：@keyframes
  - 调用动画：animation



## 二、动画

目标：使用 **animation** 属性 实现元素的动画效果

学习路径：

1. 动画的初体验
2. 定义动画 @keyframes
3. 调用动画 animation

### 2.1 定义动画 @keyframes

- 定义动画：需要分析清楚动画有几个状态，每个状态的样式是什么，按照顺序书写每个状态的样式即可
- 需求：

1. 默认盒子宽度200，让盒子宽度变为800的动画



```
/* 定义动画方式一：from-to两个状态 */  
@keyframes dong {  
  /* 动画的开始状态 */  
  from {  
    width: 200px;  
  }  
  /* 动画的结束状态 */  
  to {  
    width: 600px;  
  }  
}
```

2. 默认盒子宽高都是200，先让盒子变宽为800，在让盒子变高为800的动画



```
/* 定义动画方式二，可以设置多个状态 */  
@keyframes bian {  
  0% {  
    width: 200px;  
    height: 200px;  
  }  
  50% {  
    width: 600px;  
    height: 200px;  
  }  
  100% {  
    width: 600px;  
    height: 600px;  
  }  
}
```

### 2.2 小结

- 如果定义动画中只有两个状态，一般使用什么方式？

```
@keyframes 动画名称 {  
  from {}  
  to {}  
}
```

- 如果定义动画中有多个状态，一般使用什么方式？

```
@keyframes 动画名称 {  
  0% {}  
  10% {}  
  15% {}  
  100% {}  
}
```

## 二、动画

目标：使用 **animation** 属性 实现元素的动画效果

学习路径：

1. 动画的初体验
2. 定义动画 @keyframes
3. 调用动画 **animation**

### 3.1 调用动画 animation

- 调用动画：animation属性是一个复合属性，是8个属性的连写
- 需求：设置一个div变宽的动画



### 3.2 调用动画 - 单个属性

属性	作用	取值
animation-name	动画的名称	
animation-duration	动画的持续时间	数字+s
animation-timing-function	动画的速度变化效果	ease: 默认值, 先慢再快再慢 linear: 匀速 steps(步数): 分步进行
animation-delay	动画的延时	数字+s
animation-iteration-count	动画的执行次数	次数 infinite: 无限次
animation-direction	动画的执行方向	normal: 正常 reverse: 反转 alternate: 交替执行 (from->to->from->to.....)
animation-fill-mode	动画的结束状态	forwards: 最后一帧状态 backwards: 第一帧状态
animation-play-state	动画的播放状态	running: 播放 paused: 暂停

### 3.3 调用动画 - 连写

- 属性名: animation
- 属性值: 单个取值的合写, 取值之间以空格隔开
- 注意点:
  1. 省略问题: 动画名字: animation-name + 动画的持续时间: animation-duration 不能省略
  2. 顺序问题: 第一个时间表示: 动画的持续时间 animation-duration, 如果有第二个时间才表示: 动画的延时: animation-delay

### 3.4 小结

- 调用动画连写的属性名是什么？有哪些取值是必填的？
  - 属性名：animation
  - 属性值：动画名字 animation-name + 动画的持续时间 animation-duration 必填



## 二、动画

### 综合案例3：精灵图动画

- 需求：利用精灵图，完成逐帧动画



### 综合案例3：精灵图动画 - 实现步骤

➤ 步骤：

1. 设置盒子和精灵图中的小图片大小一致
2. 定义动画：让盒子的background-position 从第一个小图片的位置 切换到 最后一个小图片的位置 （类似于翻书动画）
3. 切换的过程需要分段进行 steps(N)
4. 如果需要一直动，需要设置动画无限次执行



### 综合案例3：精灵图动画 - 多组动画

- 需求：如果需要让小人不仅能动，还能同时进行位移？
- 多组动画：
  - animation属性可以同时调用多组动画，每组动画之间通过逗号隔开

```
animation:  
  动画1,  
  动画2,  
  动画N  
;
```

```
animation:  
  run 1s steps(12) infinite,  
  move 3s linear forwards  
;
```



# 目录

Contents

- ◆ 空间转换
- ◆ 动画
- ◆ 综合案例

## 综合案例4：走马灯案例

- 需求：使用animation实现逐帧图片位移效果



## 综合案例5：全民出游

➤ 需求：



## 晚自习安排

1. 参考老师的PPT内容，**梳理今日上课的xmind**（可以跟着老师写好的xmind写一遍，加深印象）
2. 在梳理每日内容时，如果发现模糊的地方，可以在单独快速的看一遍本节视频（切记：只看遗漏的，不要全都看）
3. 把**上课的案例多敲几遍**，直到能不看老师代码和视频，**能独立把案例敲出来为止**（忘记了也有上课录制的视频兜底，不怕做不出，就怕懒的做）
4. 把**每日综合案例独立做完**，第一次可以参考老师的视频/代码，但是要多做几次，直到同学们能做到不看视频和代码能独立做出了，才算是学扎实了。
5. 学习过程中遇到问题先独立思考5~10min左右，能自己解决的bug印象最深。如果超出时间可以求助同学或者助教或者百度或者先记在本子上，第二天直接问题即可。（切记：讨论声音不要影响其他同学）
6. 到了晚上 **9:00 记得在博学谷填写反馈** 网址为：
  - 教室内局域网：<http://ntlias-stu.boxuegu.com/>
  - 外网：<https://tlias-stu.boxuegu.com/login>



传智教育旗下高端IT教育品牌