



CSS 第二天

复合选择器 + 三大特性 + 背景属性 + 显示模式



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例



目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例



第一章

复合选择器

- 后代选择器
- 子代选择器
- 并集选择器
- 交集选择器
- 链接伪类选择器
- Emmet语法（拓展）



后代选择器

- 语法

一、复合选择器

1、后代选择器

- 作用：选择父元素 后代中 满足条件的元素
- 选择器语法： **选择器1 选择器2 { css }**
- 结果：
 - 在选择器1找到标签的后代（**儿子、孙子、重孙子...**）中，找到满足选择器2的标签，设置样式



总结

1. 后代选择器中选择器之间以什么隔开?
空格



子代选择器

- 语法

一、复合选择器

2、子代选择器

- 作用：选择父元素 **子代中** 满足条件的元素
- 选择器语法：**选择器1 > 选择器2 { css }**
- 结果：
 - 在选择器1所找到标签的 **子代（儿子）** 中，找到满足选择器2的标签，设置样式



总结

1. 子代选择器中选择器之间以什么隔开?

>



并集选择器

- 语法

一、复合选择器

3、并集选择器

- 作用：同时选择 **多组** 标签，设置相同的样式
- 选择器语法：**选择器1 , 选择器2 { css }**
- 结果：
 - 找到 选择器1 和 选择器2 选中的标签，设置样式



总结

1. 并代选择器中选择器之间以什么隔开?

,



交集选择器

- 语法

一、复合选择器

4、交集选择器

- 作用：选中页面中 **同时满足** 多个选择器的标签
- 选择器语法：**选择器1选择器2 { css }**
- 结果：
 - （既又原则）找到页面中 **既** 能被选择器1选中，**又** 能被选择器2选中的标签，设置样式
- 注意点：
 - 交集选择器中如果有标签选择器，标签选择器必须写在最前面



总结

1. 交集选择器中选择器之间以什么隔开?
紧挨着



链接伪类选择器

- 语法

一、复合选择器

5、链接伪类选择器

➤ 作用：选中超链接的 **不同状态**，设置样式

➤ 选择器：

选择器	作用
:link	访问前
:visited	访问后
:hover	鼠标悬停
:active	点击时（激活）

➤ 注意点：

1. 如果要同时设置超链接多个状态，需要按照 **LVHA** 顺序书写
2. **:hover** 可以用于其他标签，使用频率最多



总结

1. 链接伪类选择器有哪些？

- :link 链接未访问状态
- :visited 链接访问过之后
- :hover 鼠标悬停状态
- :active 鼠标点击时状态

emmet语法（拓展）

➤ 作用：代码的简写，输入简写在 VScode会自动生成对应代码

➤ HTML：

说明	标签结构	Emmet
类选择器	<code><div class="box"></div></code>	标签名.类名
id 选择器	<code><div id="box"></div></code>	标签名#id名
同级标签	<code><div></div><p></p></code>	div+p
父子级标签	<code><div><p></p></div></code>	div>p
多个相同标签	<code>123</code>	span*3
有内容的标签	<code><div>内容</div></code>	div{内容}

➤ CSS：

说明	CSS属性	Emmet
宽度	<code>width</code>	w
宽度500px	<code>width: 500px;</code>	w500
背景色	<code>background-color</code>	bgc
多个属性	<code>width: 200px;height: 100px;background-color: #fff;</code>	w200+h100+bgc



目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例



第二章

三大特性

- 继承性
- 层叠性
- 优先级
- 权重叠加计算



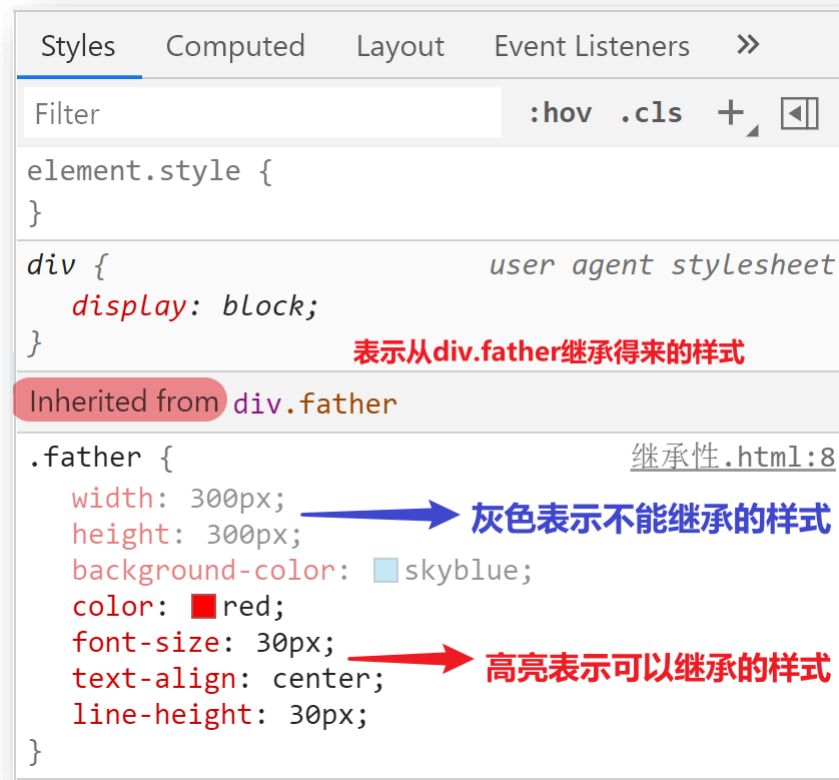
继承性

- 特性

二、三大特性

1、继承性

- 继承性：CSS中子元素有默认继承父元素样式的特点（子承父业）
- 可以继承的常见属性：
 1. color
 2. font-style、font-weight、font-size、font-family
 3. text-indent、text-align
 4. line-height
 5.
- 拓展：
 - 可以通过调试工具判断样式是否可以继承





总结

1. CSS的继承性可以继承哪些样式？

- color
- font相关
- text相关
- line-height

继承失效的特殊情况（拓展）

- 如果元素有浏览器默认样式，此时继承性依然存在，但是优先显示浏览器的默认样式
- 1. a标签的color会继承失效
 - 其实color属性继承下来了，但是被浏览器默认设置的样式给覆盖掉了
- 2. h系列标签的font-size会继承失效
 - 其实font-size属性继承下来了，但是被浏览器默认设置的样式给覆盖掉了
- 3. div的高度不能继承，但是宽度有类似于继承的效果
 - 宽度属性不能继承，但是div有独占一行的特性



层叠性

- 特性

2、层叠性的介绍

➤ 层叠性：

1. 给同一个标签设置不同的样式 → 此时样式会层叠叠加 → 会共同作用在标签上
2. 给同一个标签设置相同的样式 → 此时样式会层叠覆盖 → 写在下面的样式会覆盖上面的样式

➤ 注意点：

1. 只有当选择器相同时，才能通过层叠性判断结果



总结

1. 给同一个标签设置相同样式时，谁会生效？
相同选择器情况下，写在最下面的样式生效



优先级

- 特性

3、优先级

- 优先级（选择器权重）：不同选择器具有不同的优先级，优先级高的选择器样式会覆盖优先级低选择器样式
- 优先级公式：
 - 继承 < 通配符选择器 < 标签选择器 < 类选择器 < id选择器 < 行内样式 < !important
- 注意点：
 1. !important写在属性值的后面，分号的前面！
 2. !important不能提升继承的优先级，只要是继承优先级最低！
 3. 实际开发中不建议使用 !important。



总结

1. 选择器优先级公式是什么样？

继承 < 通配符选择器 < 标签选择器 < 类选择器 < id选择器 < 行内样式 < !important



权重叠加计算

- 特性

4、权重叠加计算

- 场景：如果是复合选择器，此时需要通过权重叠加计算方法，判断最终哪个选择器优先级最高会生效
- 权重叠加计算公式：（每一级之间不存在进位）



- 比较规则：
 1. 先比较第一级数字，如果比较出来了，之后的统统不看
 2. 如果第一级数字相同，此时再去比较第二级数字，如果比较出来了，之后的统统不看
 3.
 4. 如果最终所有数字都相同，表示优先级相同，则比较层叠性（谁写在下面，谁说了算!）
- 注意点：!important如果不是继承，则权重最高，天下第一！

总结

1. 权重叠加计算公式是什么样？





目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例



第三章

背景属性

- 背景图片
- 背景平铺
- 背景位置
- 背景大小
- 背景固定（了解）
- background连写



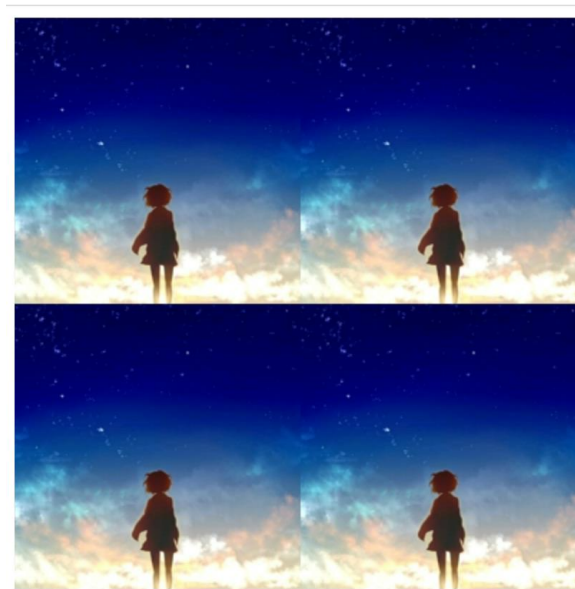
背景图片

- 属性

三、背景属性

1、背景图片

- 属性名: **background-image** (bgi)
- 属性值: `background-image: url('图片的路径');`
- 注意点:
 - 背景图片中url中可以省略引号
 - 背景图片默认是在水平和垂直方向平铺的
 - 背景图片仅仅是指给盒子起到装饰效果, 类似于背景颜色, 是不能撑开盒子的





总结

1. 背景图片属性是什么？

`background-image:url(路径)`



背景平铺

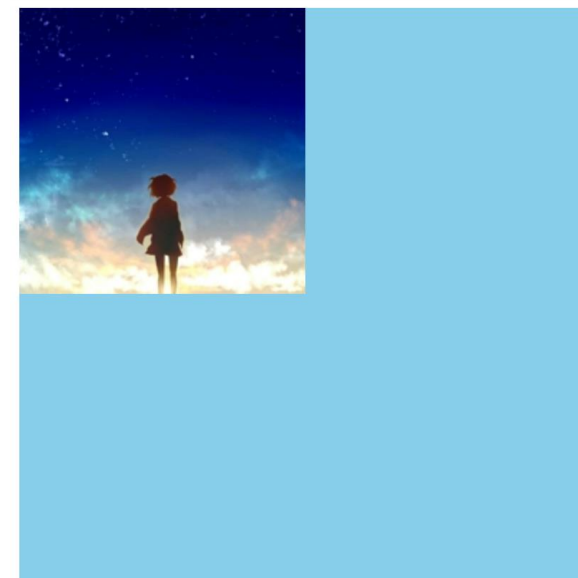
- 属性

三、背景属性

2、背景平铺

- 属性名: **background-repeat** (bgr)
- 属性值:

取值	效果
repeat	(默认值) 水平和垂直方向都平铺
no-repeat	不平铺
repeat-x	沿着水平方向 (x轴) 平铺
repeat-y	沿着垂直方向 (y轴) 平铺





总结

1. 背景图片不平铺如何设置?

`background-repeat:no-repeat`

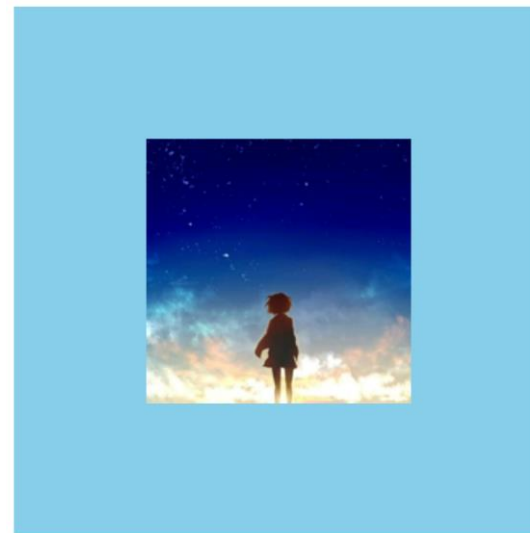
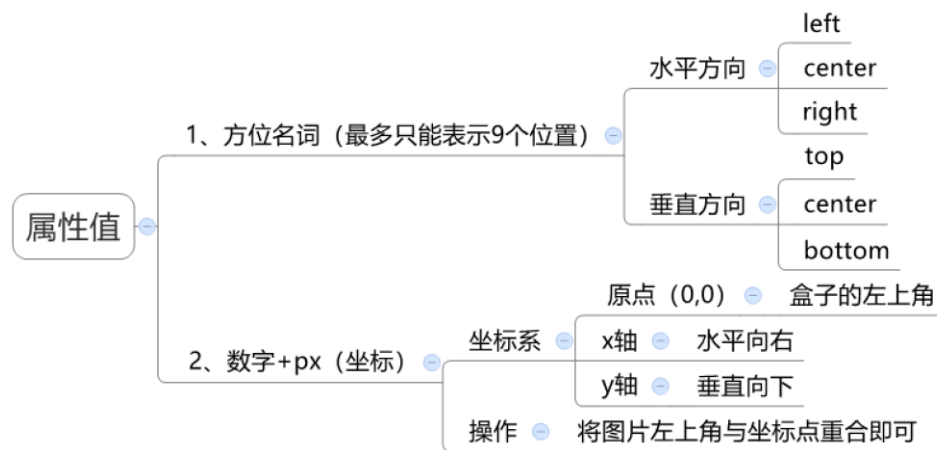


背景位置

- 属性

3、背景位置

- 属性名: **background-position** (bgp)
- 属性值: **background-position**: 水平方向位置 垂直方向位置;



- 注意点:
 - 方位名词取值和坐标取值可以混使用, 第一个取值表示水平, 第二个取值表示垂直



总结

1. 设置背景图片居中如何设置?

`background-position:center center`



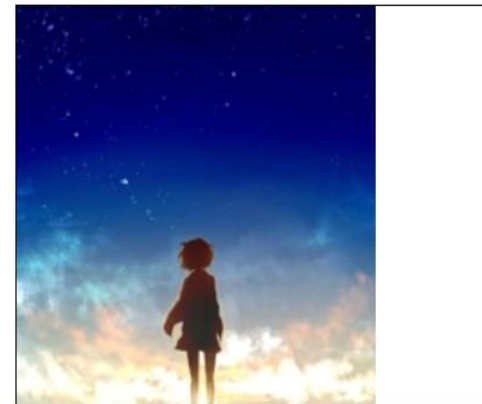
背景大小

- 属性

4、背景大小

- 作用：设置背景图片的大小，
- 语法：background-size: 宽度 高度;
- 取值：

取值	场景
数字+px	简单方便，常用
百分比	相对于当前盒子自身的宽高百分比
contain	包含，将背景图片等比例缩放，直到不会超出盒子的最大
cover	覆盖，将背景图片等比例缩放，直到刚好填满整个盒子没有空白



contain



cover



总结

1. 设置背景图片大小和当前盒子大小一致如何设置?
`background-size:100% 100%`



背景固定（了解）

- 属性

5、背景固定（了解）

- 作用：设置背景图片不会跟随元素的内容滚动
- 语法：background-attachment: fixed;

```
body {  
    background-image: url(./images/bg.jpg);  
    background-repeat: no-repeat;  
  
    background-attachment: fixed;  
}
```



总结

1. 设置背景图片固定不跟随盒子滚动的代码是什么?
`background-attachment:fixed`

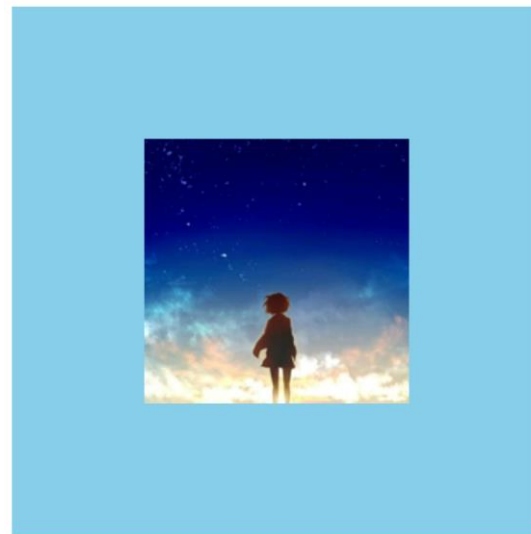


background连写

- 属性

6、background连写

- 属性名: **background** (bg)
- 属性值 (推荐顺序): color image repeat position/size attachment
- 省略问题:
 - 可以按照需求省略
 - 在pc端, 如果盒子大小和背景图片大小一样, 此时可以直接写 background: url()
- 注意点
 - 如果需要设置单独的样式和连写
 - ① 要么把单独的样式写在连写的下面
 - ② 要么把单独的样式写在连写的里面





总结

1. 背景图片的连写如何书写？

`background:color image repeat position/size attachment`



目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例



第四章

显示模式

- 块级元素
- 行内元素
- 行内块元素
- 显示模式转换



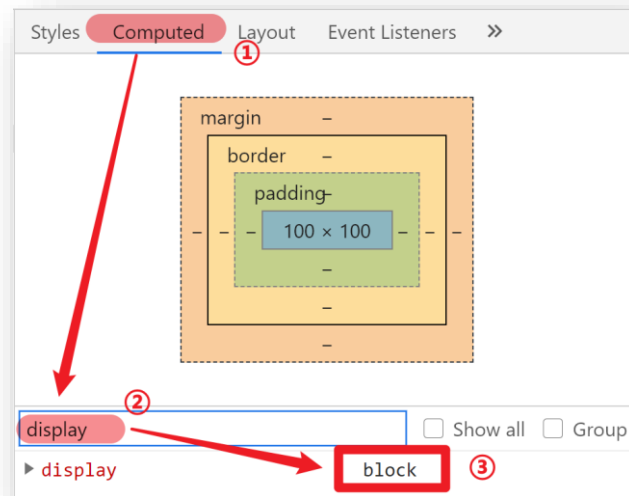
块级元素

- 显示特点

四、显示模式

1、块级元素

- 属性: `display: block`
- 显示特点:
 1. 独占一行 (一行只能显示一个)
 2. 宽度默认占满父元素一行, 高度默认由内容撑开
 3. 可以设置宽高
- 代表标签:
 - **div、p、h系列**、ul、li、dl、dt、dd、form、header、nav、footer.....





总结

1. 块级元素的display取值是什么？

block

2. 块级元素的显示特点是什么？

独占一行

宽度默认占满一行，高度默认由内容撑开

可以设置宽高

3. 常见的块级元素有哪些？

div、p、h系列

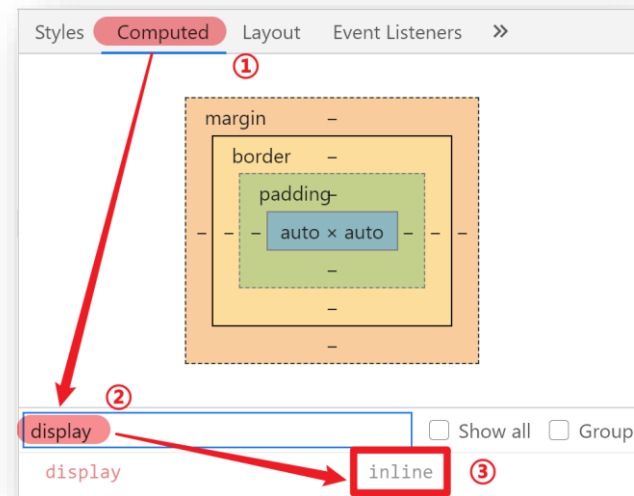


行内元素

- 显示特点

2、行内元素

- 属性: `display: inline`
- 显示特点:
 1. 一行可以显示多个
 2. 宽度和高度默认由内容撑开
 3. 不可以设置宽高
- 代表标签:
 - **a**、**span**、b、u、i、s、strong、ins、em、del.....





总结

1. 行内元素的display取值是什么？

inline

2. 行内元素的显示特点是什么？

一行可以显示多个

宽度和高度默认由内容撑开

不可以设置宽高

3. 常见的行内元素有哪些？

a、span



行内块元素

- 显示特点

3、行内块元素

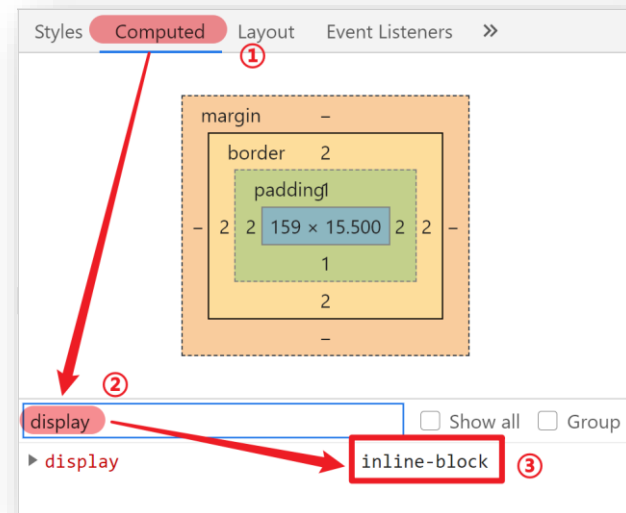
➤ 属性: display: **inline-block**

➤ 显示特点:

1. 一行可以显示多个
2. 可以设置宽高

➤ 代表标签:

- **input**、**textarea**、**button**、**select**.....
- 特殊情况: **img**标签有行内块元素特点, 但是Chrome调试工具中显示结果是**inline**





总结

1. 行内块元素的display取值是什么?
inline-block
2. 行内块元素的显示特点是什么?
一行可以显示多个
可以设置宽高
3. 常见的行内块元素有哪些?
input、textarea



显示模式转换

- 转换代码

4、显示模式转换

- 目的：通过改变元素显示模式，让元素获取对应布局特点，从而符合布局要求
- 语法：

属性	效果	使用频率
<code>display: block</code>	转换成块级元素	较多
<code>display: inline-block</code>	转换成行内块元素	较多
<code>display: inline</code>	转换成行内元素	极少



总结

1. 把元素转换成块级元素如何实现?
`display:block`
2. 把元素转换成行内块元素如何实现?
`display:inline-block`



目录

Contents

- ◆ 复合选择器
- ◆ 三大特性
- ◆ 背景属性
- ◆ 显示模式
- ◆ 综合案例

案例

综合案例1 – 热词案例

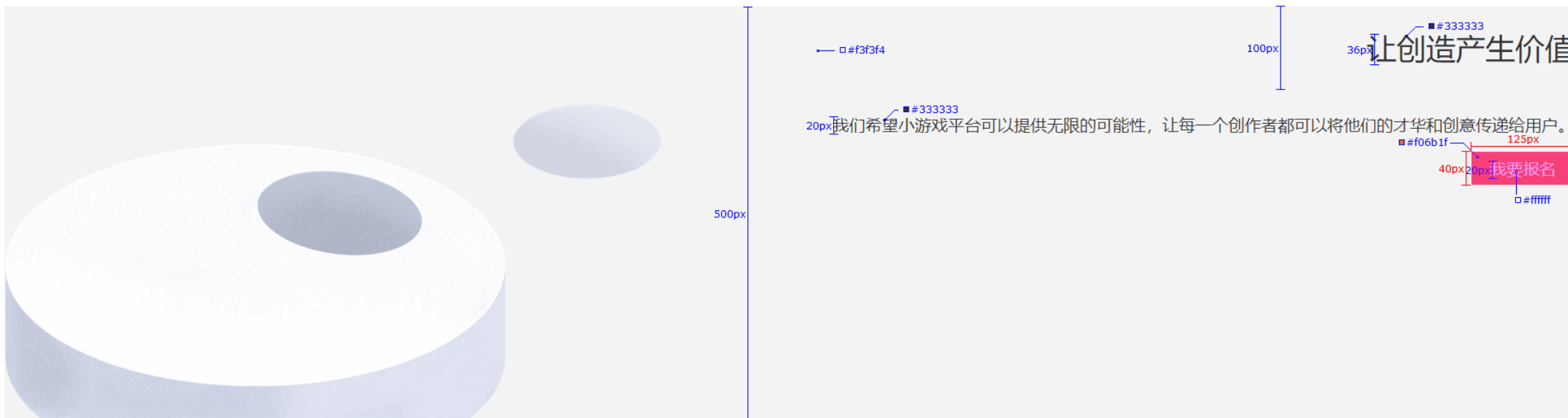
案例实现步骤：

1. 先分析：从上到下，从外往内
2. 再写代码：写一点，浏览器运行看一点



案例

综合案例2 – banner效果





传智教育旗下高端IT教育品牌