

CSS方法论与高级实用技巧

讲师：夏江徽

2019年7月

目录

CONTENTS

1

CSS方法论的发展与演进

2

主流UI框架的CSS设计对比

3

滤镜与混合模式实践

4

CSS创意与视觉表现

5

练习作业

课程目标

- 了解掌握CSS方法论相关知识，包括命名方法、预处理器、常见UI框架的CSS设计等
- 掌握CSS3滤镜与混合模式的常规使用
- 通过一些非常规CSS技巧的使用，为以后在解决问题时多一个思路

1

CSS方法论的发展与演进

一、CSS方法论的发展与演进

• 1.1 CSS的诞生



一、CSS方法论的发展与演进

- 1.2 CSS命名的模块化

BEM

SUITCSS

OOCSS

SMACSS

ITCSS

.....

一、CSS方法论的发展与演进

- 1.3 CSS预处理器

SASS(SCSS)

LESS

Stylus

一、CSS方法论的发展与演进

- 1.4 CSS与工程化



一、CSS方法论的发展与演进

• 1.4 CSS与工程化

CSS in JS

```
const style = {  
  'color': 'red',  
  'fontSize': '46px'  
};  
  
const clickHandler = () => alert('hi');  
  
ReactDOM.render(  
  <h1 style={style} onclick={clickHandler}>  
    Hello, world!  
  </h1>,  
  document.getElementById('example')  
);
```

CSS Modules

```
import React from 'react';  
import style from './App.css';  
  
export default () => {  
  return (  
    <h1 className={style.title}>  
      Hello World  
    </h1>  
  );  
};
```

```
.title {  
  color: red;  
}
```

```
<h1 class="_3zyde411yATC0kgn-DBWEL">  
  Hello World  
</h1>
```

```
._3zyde411yATC0kgn-DBWEL {  
  color: red;  
}
```

一、CSS方法论的发展与演进

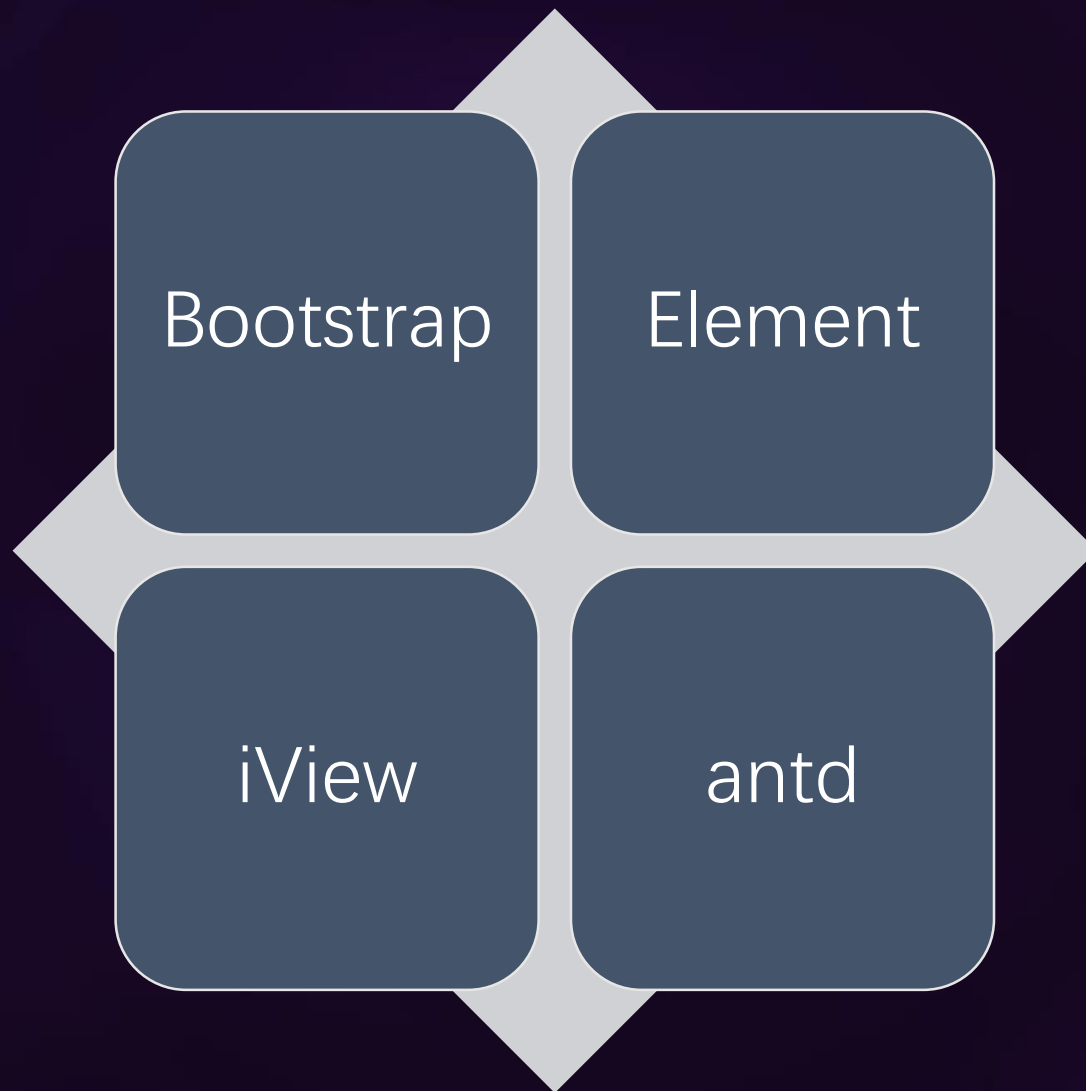
• 1.5 CSS的发展与未来



2

主流UI框架的CSS设计对比

二、主流UI框架的CSS设计对比



二、主流UI框架的CSS设计对比

• 2.1 Bootstrap4的CSS设计

<https://github.com/twbs/bootstrap/tree/master/scss>

Layout & components

```
@import "images";  
@import "code";  
@import "grid";  
@import "tables";  
@import "forms";  
@import "buttons";  
.....
```

Configuration

Helpers
Mixins
Utilities
vendor

SCSS+ITCSS

二、主流UI框架的CSS设计对比

- 2.2 Element的CSS设计 <https://github.com/ElementFE/element/tree/dev/packages/theme-chalk/src>

components

```
@import "../base.scss";  
@import "../pagination.scss";  
@import "../dialog.scss";  
@import "../autocomplete.scss";  
@import "../dropdown.scss";  
@import "../dropdown-menu.scss";  
.....
```

common
mixins

SCSS+BEM

二、主流UI框架的CSS设计对比

- 2.3 iView的CSS设计

<https://github.com/iview/iview/tree/2.0/src/styles>

components

```
@import "button";  
@import "affix";  
@import "back-top";  
@import "badge";  
@import "circle";  
@import "spin";  
.....
```

common
Animation
Color
mixins

LESS+类BEM

二、主流UI框架的CSS设计对比

- 2.4 antd的CSS设计

<https://github.com/ant-design/ant-design/tree/master/components>

components

affix/style/index.less
alert/style/index.less
anchor/style/index.less
avatar/style/index.less
back-top/style/index.less
badge/style/index.less
.....

color
core
mixins
themes

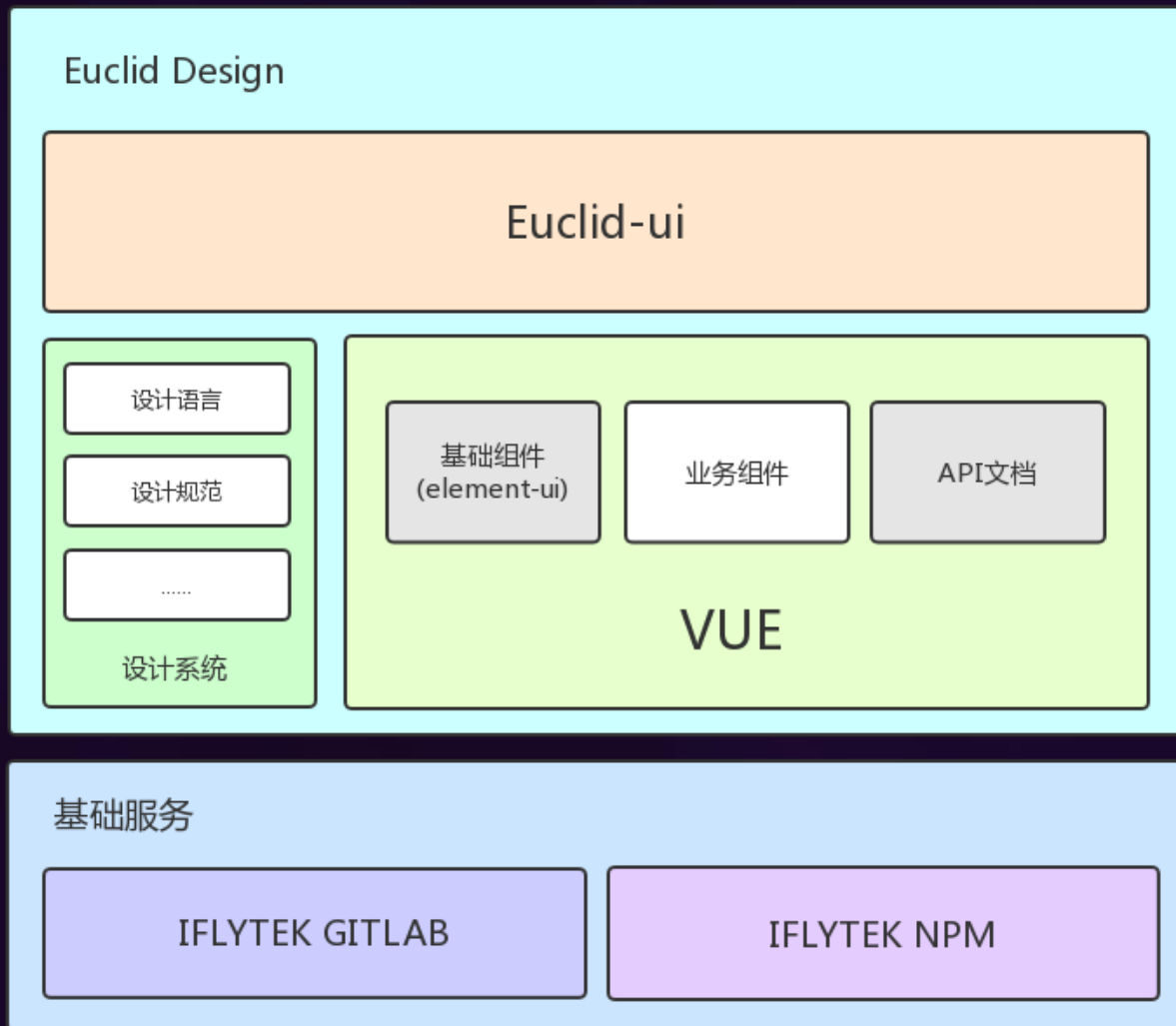
LESS+类BEM

二、主流UI框架的CSS设计对比

UI框架	预处理器	命名方法	备注
Bootstrap	SCSS	ITCSS	Bootstrap3开始由LESS编写转成由SCSS编写，基于jQuery
Element	SCSS	BEM	基于Vue
iView	LESS	类BEM	基于Vue,参考antd
antd	LESS	类BEM	基于React

CSS的设计与所使用的具体JS框架无必然关联；
CSS预处理器的使用上SCSS和LESS是主流；
命名方法的使用上BEM或类似BEM思想的命名比较流行。

二、主流UI框架的CSS设计对比



- 定制主题
- 定制业务组件规范（提供组件开发工具）

3

滤镜与混合模式实践

三、滤镜与混合模式实践

- 列表

- 3.1 兼容性与适用场景
- 3.2 CSS filter滤镜简介及应用
- 3.3 引用SVG滤镜技术
- 3.4 CSS 混合模式简介及应用
- 3.5 CSS 滤镜与混合模式同时应用

三、滤镜与混合模式实践

- 3.1 兼容性与适用场景

CSS filter滤镜兼容性

IE	Edge [*]	Firefox	Chrome	Safari	Opera	iOS Safari [*]	Opera Mini [*]	Android Browser [*]
		2-3.5						
		3.6-33	4-17	3.1-5.1	10-12.1	3.2-5.1		
	2 4 12	1 34	18-52	6-9	15-39	6-9.2		2.1-4.3
6-10	4 13-17	35-66	53-74	9.1-12	40-57	9.3-12.1		4.4-4.4.4
11	4 18	67	75	12.1	58	12.2	all	67
	76	68-69	76-78	13-TP		13		

2013年

三、滤镜与混合模式实践

- 3.1 兼容性与适用场景

CSS *-blend-mode混合模式兼容性

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini*	Android Browser *
			4-34		10-21			
			35-45	3.1-7	22-32	3.2-7.1		
		2-29	² 46	¹ 7.1-10	² 33	¹ 8-10.2		
6-10	12-17	30-66	47-74	10.1-12	34-57	10.3-12.1		2.1-4.4.4
11	18	67	75	12.1	58	12.2	all	67
	76	68-69	76-78	13-TP		13		

2014年

三、滤镜与混合模式实践

• 3.1 兼容性与适用场景

适用场景结论

➤ IE/Edge: 

➤ 中后台/内部系统: 

➤ 移动端

运营活动: 

国民级产品: 

非国民产品: 

渐进增强

三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

1.CSS filter滤镜简介

滤镜	释义
filter:blur(5px)	模糊
filter:brightness(1.4)	亮度
filter:contrast(200%)	对比度
filter:drop-shadow(4px 4px 8px blue)	投影
filter:grayscale(60%)	灰度
filter:hue-rotate(90deg)	色调变化
filter:invert(75%)	反相
filter:opacity(25%)	透明度
filter:saturate(230%)	饱和度
filter:sepia(60%)	褐色

原图



模糊 filter: blur(5px);



亮度 filter: brightness(1.4);



对比度 filter: contrast(2);



投影 filter: drop-shadow(4px 4px 8px blue); 灰度 filter: grayscale(60%);



色调变化 filter: hue-rotate(90deg);



反相 filter: invert(75%);



透明度 filter: opacity(25%);



饱和度 filter: saturate(230%);



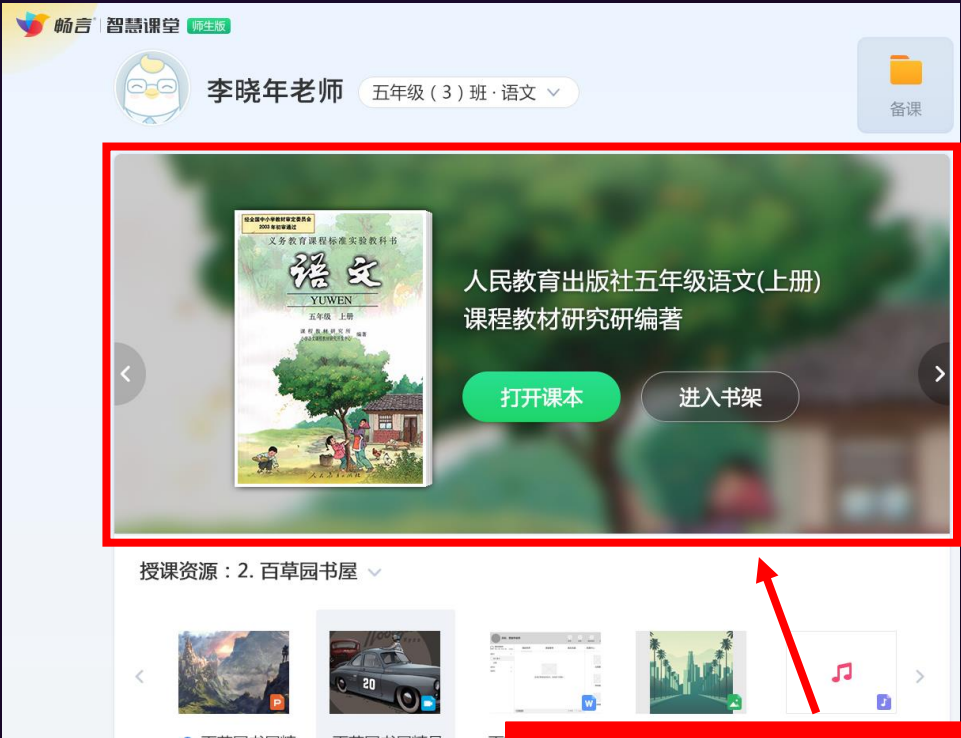
褐色 filter: sepia(60%);



三、滤镜与混合模式实践

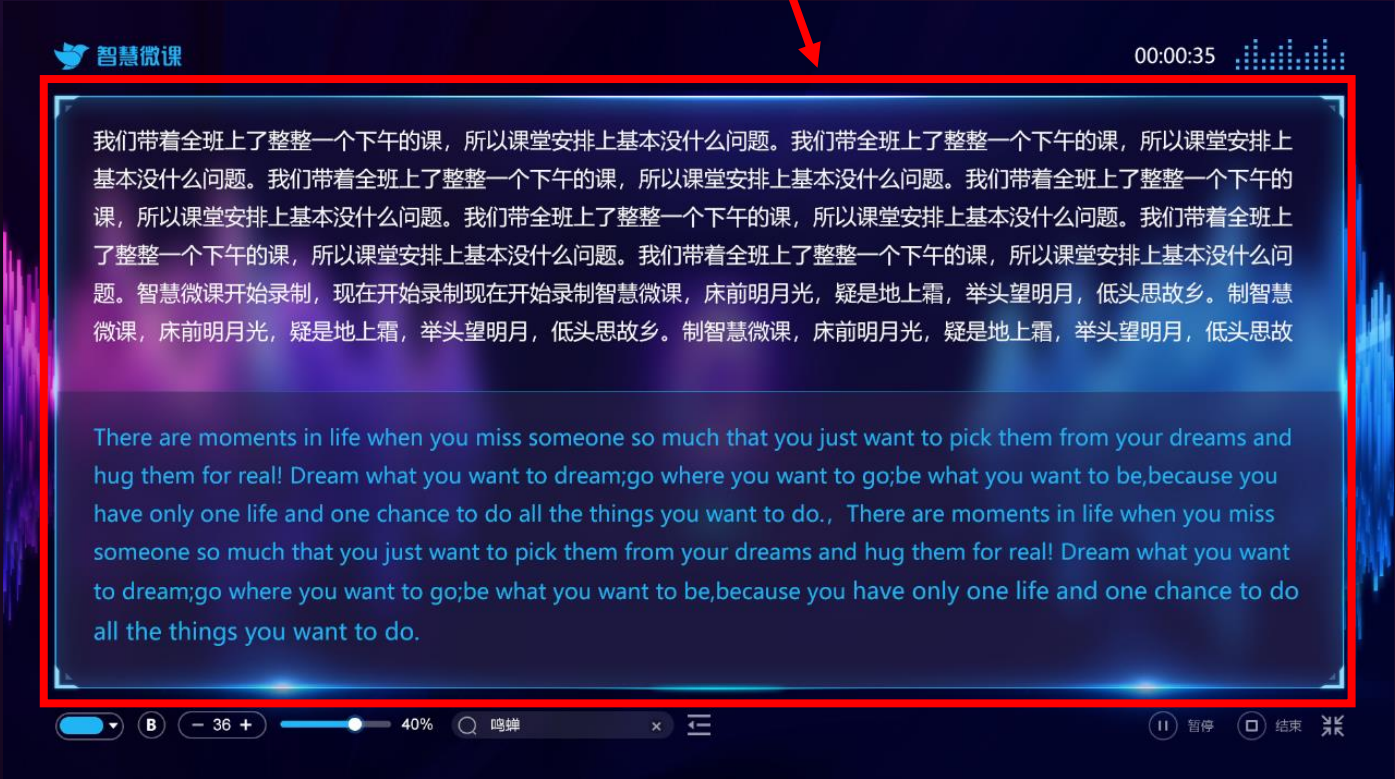
• 3.2 CSS filter滤镜简介及应用

2.filter:blur基础应用



封面图模糊做背景图

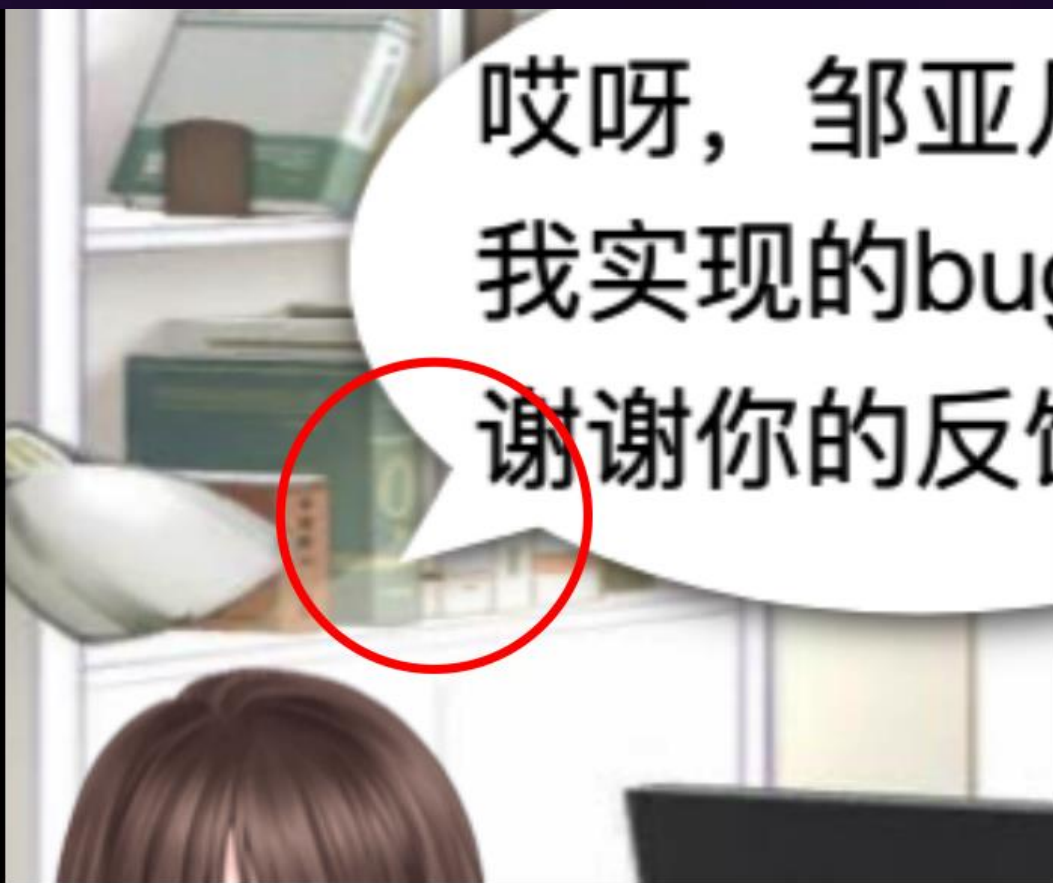
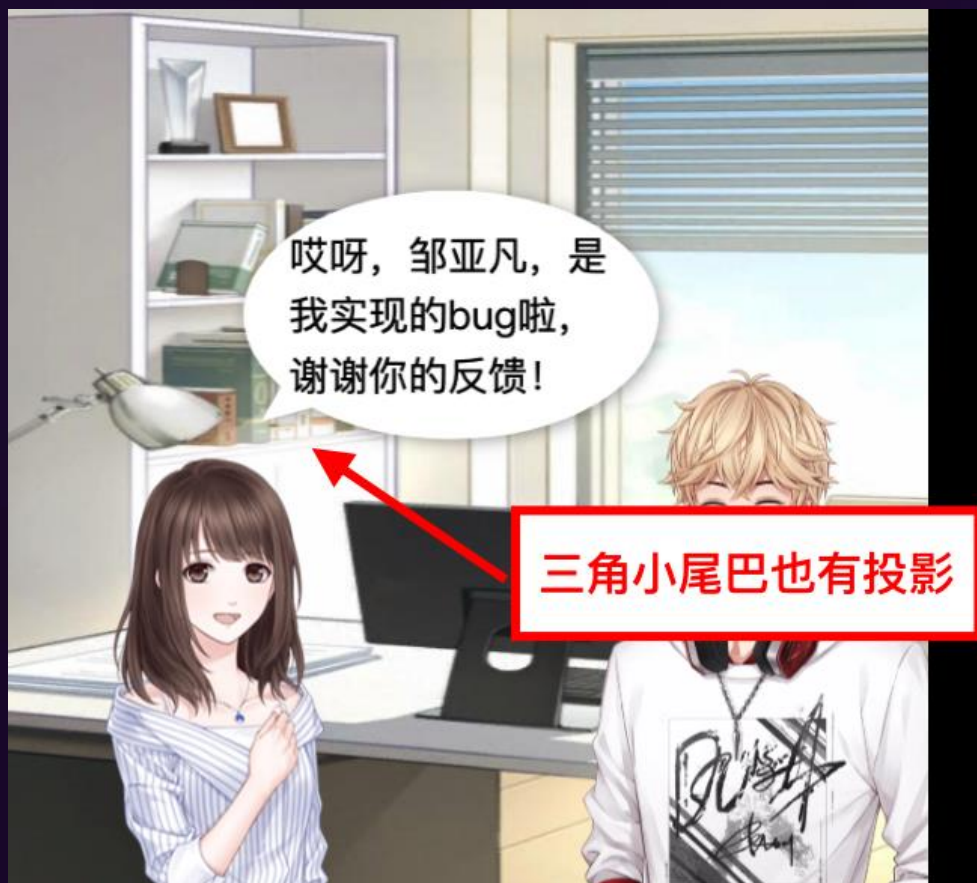
指定区域背景模糊



三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

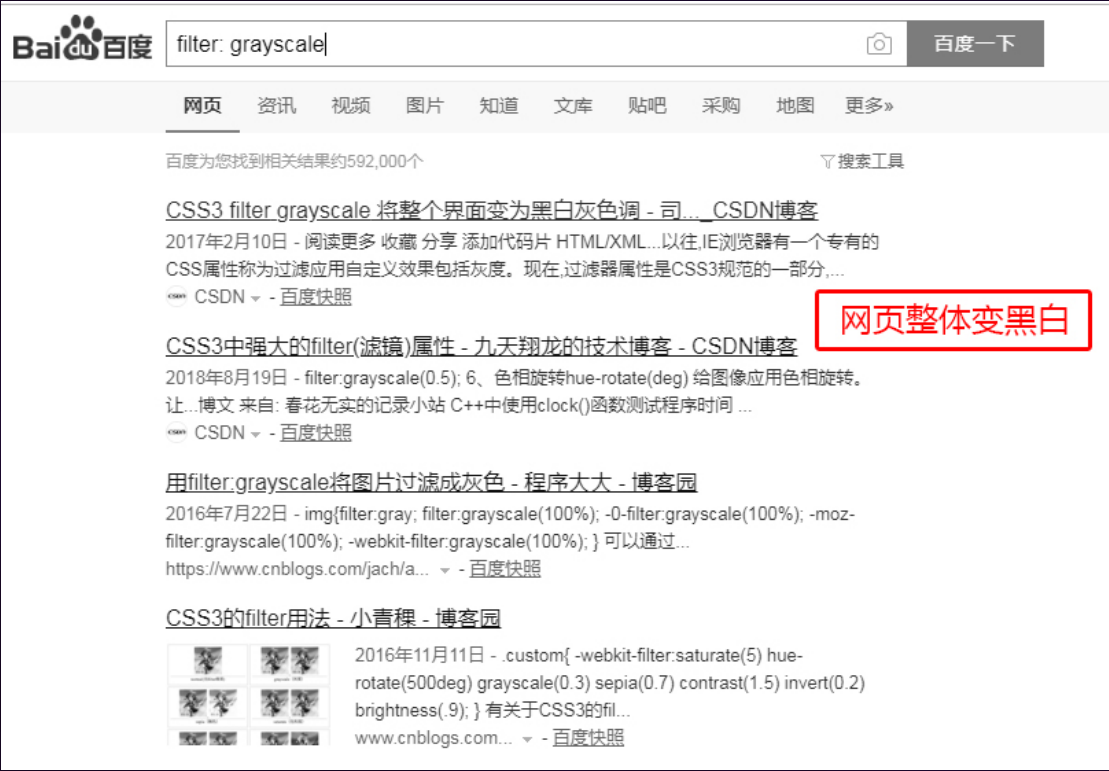
3.filter:drop-shadow基础应用



三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

4.filter:grayscale基础应用



三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

5.filter高级应用-径向模糊

原图



径向模糊



```
  

```

```
.radial-blur {  
  position: absolute;  
  filter: blur(20px);  
  mask-image: radial-gradient(transparent, transparent 10%, black 60%);  
  transform: scale(1.2);  
}
```

三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

6.filter高级应用-局部模糊



原图



局部模糊



```
  

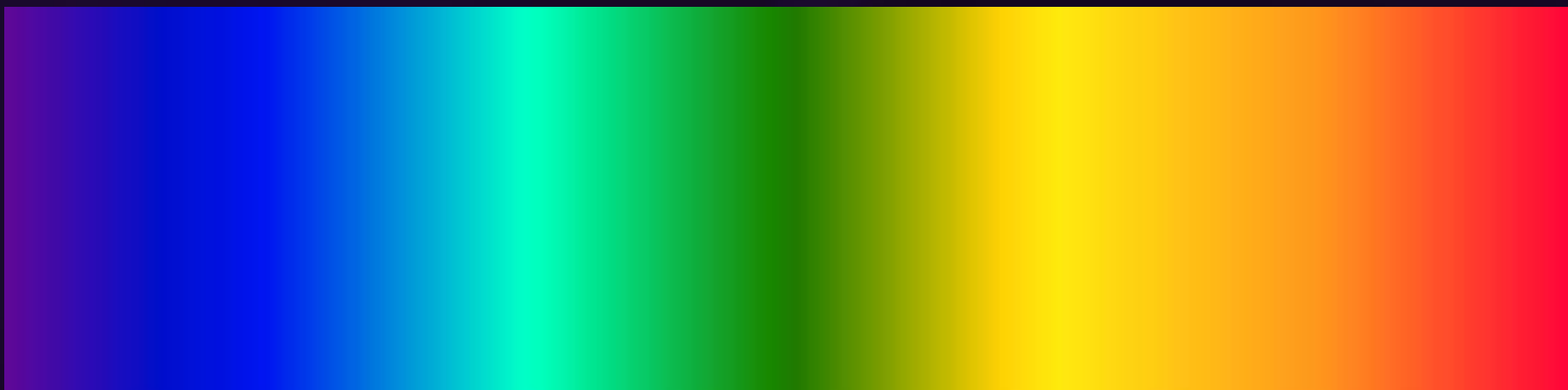
```

```
.local-blur {  
  position: absolute;  
  filter: blur(12px);  
  mask: no-repeat center;  
  mask-image: linear-gradient(□black, □black), linear-gradient(□black, □black);  
  mask-size: cover, 60px 60px;  
  mask-composite: exclude;  
  mask-composite: source-out;  
  transform: scale(1.2);  
}
```

三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

7.filter高级应用-色彩动画



看起来上面的色彩是在水平流动的，实际上是？

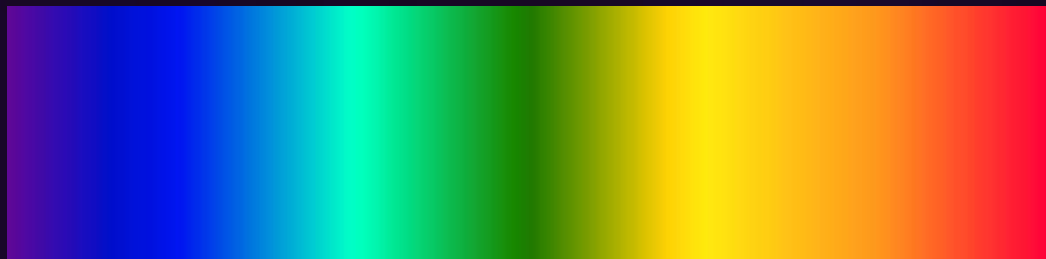
三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

7.filter高级应用-色彩动画

```
.flow-colorful-{  
  max-width: 600px;  
  height: 150px;  
  background: linear-gradient(to left, red, orange, yellow, green, cyan, blue, purple);  
  animation: hue 6s linear infinite;  
}  
  
@keyframes hue-{  
  from-{  
    filter: hue-rotate(0deg);  
  }  
  to-{  
    filter: hue-rotate(360deg);  
  }  
}
```

实际上是色相的变化



三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

7.filter高级应用-色彩动画-文字应用

```
<div class="flow-slogon">前端中级训练营系列课程</div>
```

```
.flow-slogon-{  
  font-size: 55px;  
  -webkit-background-clip: text;  
  -webkit-text-fill-color: transparent;  
  background-image: linear-gradient(to left, red, yellow, lime, aqua, blue, fuchsia);  
  animation: hue 6s linear infinite;  
}
```

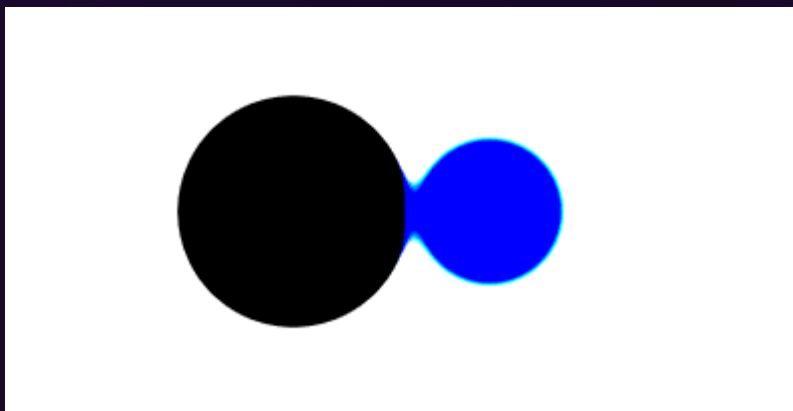


前端中级训练营系列课程

三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

8.filter高级应用-融合黏着



```
<div class="container">  
  <div class="filter-mix"></div>  
</div>
```

```
.filter-mix {  
  position: absolute; top: 50%; left: 50%;  
  transform: translate(-50%, -50%); width: 300px; height: 200px;  
  filter: contrast(20);  
  background: #fff;  
}  
  
.filter-mix::before {  
  content: ''; position: absolute; width: 120px; height: 120px; box-sizing: border-box;  
  border-radius: 50%; background: #333; top: 40px; left: 40px; z-index: 2;  
  filter: blur(6px);  
  animation: filterBallMove 4s ease-out infinite;  
}  
  
.filter-mix::after {  
  content: ''; position: absolute; width: 80px; height: 80px; border-radius: 50%;  
  background: #3f51b5; top: 60px; right: 40px; z-index: 2;  
  filter: blur(6px);  
  animation: filterBallMove2 4s ease-out infinite;  
}
```

三、滤镜与混合模式实践

- 3.2 CSS filter滤镜简介及应用

8.filter高级应用-融合黏着



<https://codepen.io/Chokcoco/pen/aMRPjR>



<https://codepen.io/Chokcoco/pen/gZVjJw>

三、滤镜与混合模式实践

- 3.3 引用SVG滤镜技术

1. 引用SVG滤镜

```
-- /* 外链SVG滤镜 */  
-- .filter {  
-- | filter: url('filtername.svg#filter-id');  
-- }  
-- /* 内联SVG滤镜 */  
-- .filter {  
-- | filter: url('#filter-id');  
-- }
```

三、滤镜与混合模式实践

• 3.3 引用SVG滤镜技术

2.SVG滤镜融合黏着效果实现1

1、页面任意位置插入

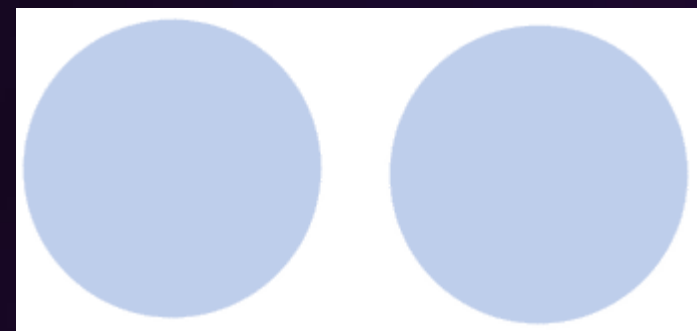
```
<svg width="0" height="0">
  <defs>
    <filter id="goo">
      <feGaussianBlur in="SourceGraphic" stdDeviation="10" result="blur" />
      <feColorMatrix in="blur" mode="matrix" values="1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 19 -9" result="goo" />
      <feComposite in="SourceGraphic" in2="goo" operator="atop" />
    </filter>
  </defs>
</svg>
```

2、假设页面上有两个球

```
<div class="target">
  <i id="ball1" class="ball"></i>
  <i id="ball2" class="ball"></i>
</div>
```

3、在需要融合的父亲元素上设置样式

```
.target {
  filter: url('#goo');
}
```



三、滤镜与混合模式实践

• 3.3 引用SVG滤镜技术

3.SVG滤镜融合黏着效果实现2



1、页面任意位置插入

```
<svg width="0" height="0">
  <defs>
    <filter id="goo">
      <feGaussianBlur in="SourceGraphic" stdDeviation="10" result="blur" />
      <feColorMatrix in="blur" mode="matrix" values="1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 19 -9" result="goo" />
      <feComposite in="SourceGraphic" in2="goo" operator="atop" />
    </filter>
  </defs>
</svg>
```

2、分享菜单html结构

```
<div class="target">
  <label class="share" for="share">分享</label>
  <span class="icon-share-weibo"></span>
  <span class="icon-share-wechat"></span>
  <span class="icon-share-qq"></span>
</div>
```

3、在需要融合的父亲元素上设置样式

```
.target {
  filter: url('#goo');
}
```

三、滤镜与混合模式实践

- 3.3 引用SVG滤镜技术

4.其它SVG滤镜动效示意-水波特效



三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

1.混合模式相关属性

- background-blend-mode:

用于混合元素背景图案、渐变和颜色

- mix-blend-mode:

用于元素与元素之间的混合

- isolation: isolate

用在祖先元素上，限制mix-blend-mode混合模式应用范围

三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

2.混合模式属性值

属性	描述
mix-blend-mode: normal	正常
mix-blend-mode: multiply	正片叠底
mix-blend-mode: screen	滤色
mix-blend-mode: overlay	叠加
mix-blend-mode: darken	变暗
mix-blend-mode: lighten	变亮
mix-blend-mode: color-dodge	颜色减淡
mix-blend-mode: color-burn	颜色加深
mix-blend-mode: hard-light	强光
mix-blend-mode: soft-light	柔光
mix-blend-mode: difference	差值
mix-blend-mode: exclusion	排除
mix-blend-mode: hue	色相
mix-blend-mode: saturation	饱和度
mix-blend-mode: color	颜色
mix-blend-mode: luminosity	亮度

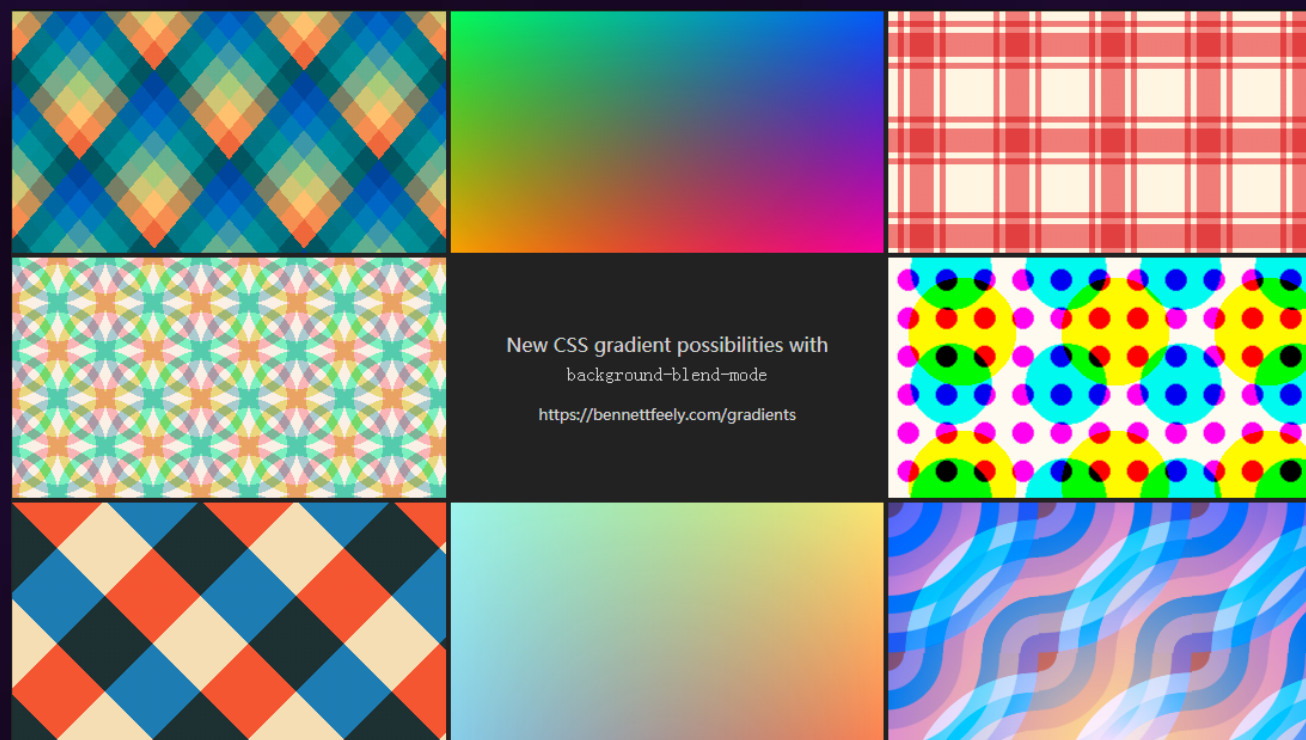
三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

3. CSS3绘制背景项目



<https://leaverou.github.io/css3patterns/>



<https://codepen.io/bennettfeely/pen/wJbtk>

三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

4.差值模式(difference)应用

前端中级训练营系列课程

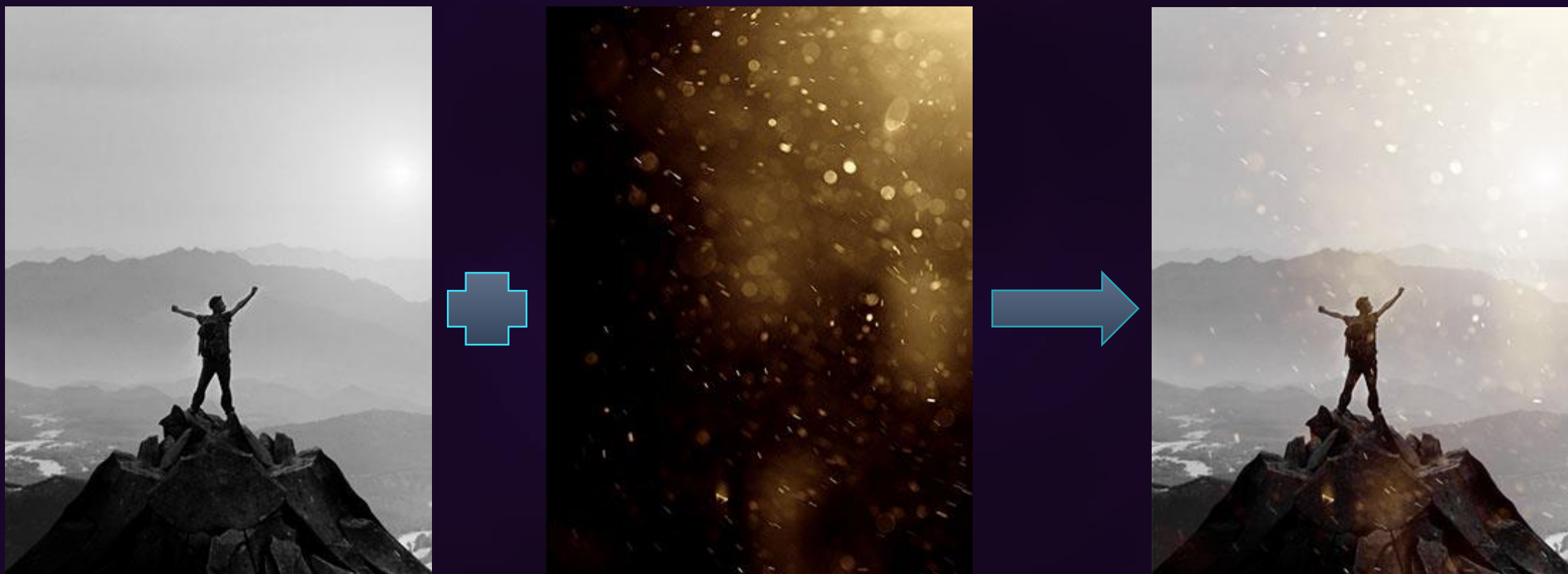
```
..text-{  
  - mix-blend-mode: difference;  
-}  
..backgrund-{  
  - background: linear-gradient(■ #fff 50%, □ #000 50%);  
  - animation: rotate 5s linear infinite;  
-}
```



三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

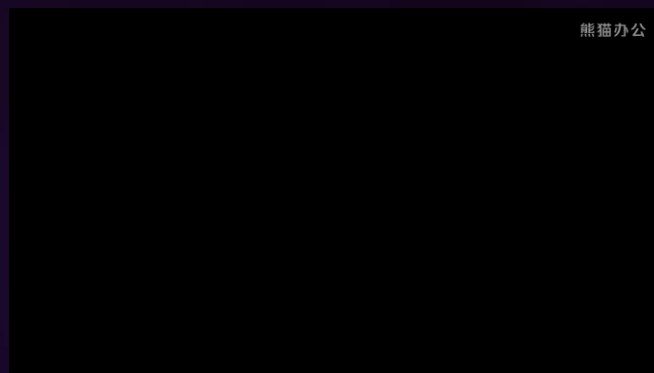
5. 滤色模式(screen)应用- 图片



三、滤镜与混合模式实践

- 3.4 CSS 混合模式简介及应用

5. 滤色模式(screen)应用-视频



三、滤镜与混合模式实践

- 3.5 CSS 滤镜与混合模式同时应用

1.静态图像处理-白天模拟夜晚



白天模拟夜晚




三、滤镜与混合模式实践

- 3.5 CSS 滤镜与混合模式同时应用

1.静态图像处理-白天模拟夜晚



```
.night {  
  width: 256px;  
  height: 256px;  
  background:  rgba(0, 40, 140, 0.6) url(images/6.jpg);  
  background-size: 100%;  
  background-blend-mode: darken;  
  filter: brightness(80%) grayscale(20%) contrast(1.2);  
}
```

三、滤镜与混合模式实践

- 3.5 CSS 滤镜与混合模式同时应用

2.静态图像处理-CSS PS照片



照片更朦胧



```
.friend-{  
  width: 256px;  
  height: 171px;  
  background: ■ rgba(214, 224, 175, 0.8) url(images/3.jpg);  
  background-size: 100%;  
  background-blend-mode: soft-light;  
  filter: contrast(1.1);  
}
```


三、滤镜与混合模式实践

- 3.5 CSS 滤镜与混合模式同时应用

3.静态图像处理-CSSgram项目

<https://github.com/una/CSSgram>



26种图片处理风格



4

CSS创意与视觉表现

四、CSS创意与视觉表现

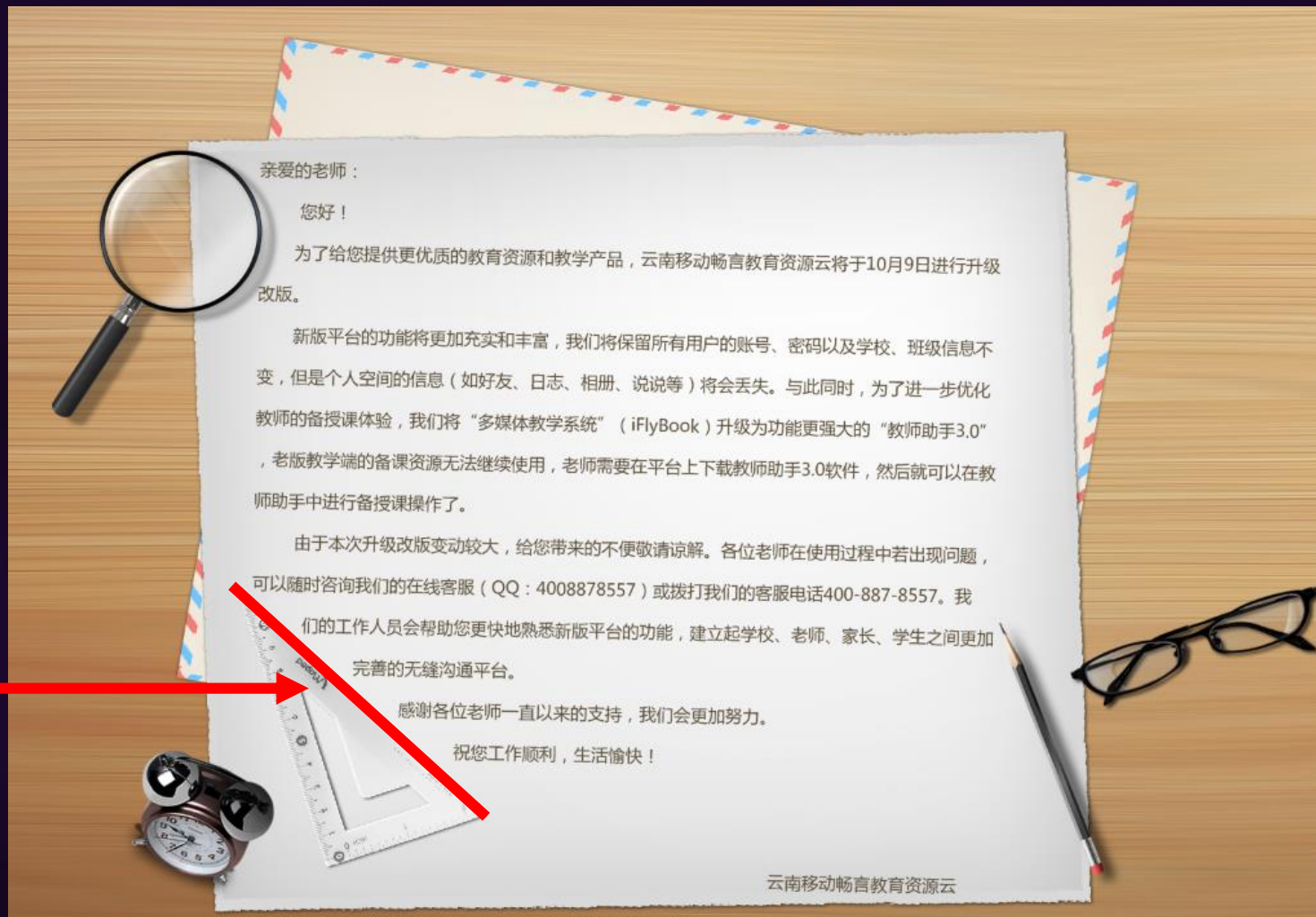
- 列表

- 4.1 布局-不规则内容排版
- 4.2 图形-镂空
- 4.3 动画-饼图
- 4.4 动画-打点
- 4.5 行为-分栏宽度拉伸

四、CSS创意与视觉表现

• 4.1 布局-不规则内容排版

某块内容具有一定的倾斜角度



四、CSS创意与视觉表现

• 4.1 布局-不规则内容排版

`shape-outside`是不规则形状环绕布局的核心，支持的属性值分为如下四大类：

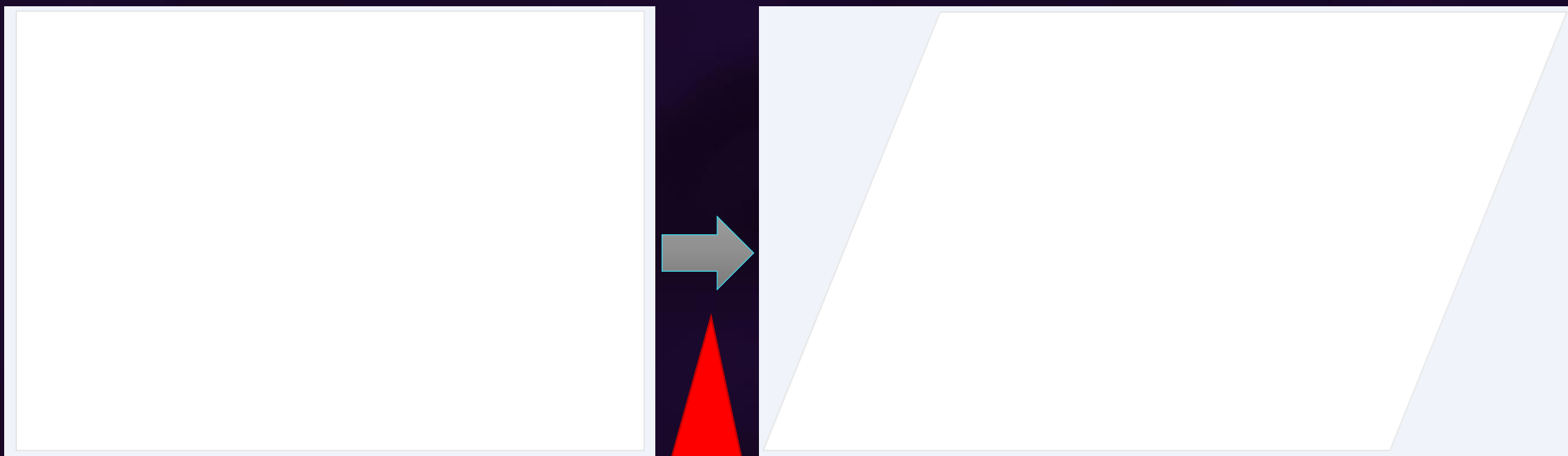
1. `none` - 默认值
2. `<shape-box>` - 图形盒子。
3. `<basic-shape>` - 基本图形函数。
4. `<image>` - 图像类。

其中：

- `none` 该浮动区域不产生影响，就是普通的矩形环绕。
- `<shape-box>`（图形盒子）是`shape`相关布局中的一个名词，根据浮动元素的边缘（通过 `CSS box model` 来定义）形状计算出浮动的区域。可能是 `margin-box`，`border-box`，`padding-box`，或者 `content-box`。这个形状包括了由 `border-radius` 属性制造出来的弧度（与 `background-clip` 的表现类似）。
- `<basic-shape>`指的是基本形状函数，基于 `inset()`，`circle()`，`ellipse()`，或者 `polygon()` 其中一个创造出来的形状计算出浮动区域。
- `<image>` 值的是图像类，包括URL链接图片，渐变图像，`cross-fade()`，`element()`等。所有这些图像类，`CSS3 mask`遮罩属性也都支持，本文只会介绍常用的URL链接和渐变图像，其他图片类不介绍。

四、CSS创意与视觉表现

- 4.1 布局-不规则内容排版



transform: skewX(-22deg);

四、CSS创意与视觉表现

• 4.1 布局-不规则内容排版

shape-outside是不规则形状环绕布局的核心，支持的属性值分为如下四大类：

1. none - 默认值
2. <shape-box> - 图形盒子。
3. <basic-shape> - 基本图形函数。
4. <image> - 图像类。

其中：

- none该浮动区域不产生影响，就是普通的矩形环绕。
- <shape-box>（图形盒子）是shape相关布局中的一个名词，根据浮动元素的边缘（通过 CSS box model 来定义）形状计算出浮动的区域。可能是 margin-box, border-box, padding-box, 或者 content-box。这个形状包括了由 border-radius 属性制造出来的弧度（与 background-clip 的表现类似）。
- <basic-shape>指的是基本形状函数，基于 inset(), circle(), ellipse(), 或者 polygon() 其中一个创造出来的形状计算出浮动区域。
- <image>值的是图像类，包括URL链接图片，渐变图像，cross-fade(), element()等。所有这些图像类，CSS3 mask遮罩属性也都支持，本文只会介绍常用的URL链接和渐变图像，其他图片类不介绍。



```
.shape-left {  
  float: left; width: 200px; height: 500px;  
  shape-outside: polygon(0% 0, 100% 0, 0% 100%);  
}
```

```
.shape-right {  
  float: right; width: 200px; height: 500px;  
  shape-outside: polygon(100% 0, 100% 100%, 0 100%);  
}
```



四、CSS创意与视觉表现

- 4.2 图形-镂空

图像剪裁



四、CSS创意与视觉表现

• 4.2 图形-镂空

```
.clip-shape-{  
  width: 150px; height: 150px;  
  outline: 9999px solid transparent; /* 这里有一个小方框图标 */  
  position: absolute; left: 0; right: 0; top: 0; bottom: 0;  
  margin: auto; cursor: move;  
}
```

图像剪裁



四、CSS创意与视觉表现

• 4.2 图形-镂空

新手引导



四、CSS创意与视觉表现

• 4.2 图形-镂空

新手引导



四、CSS创意与视觉表现

• 4.2 图形-镂空

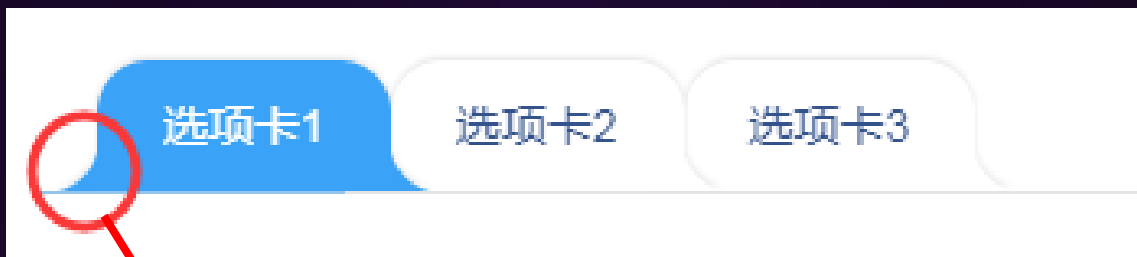
```
.guide-overlay-shape-{  
  width: 100px; height: 60px; position: absolute;  
  left: 0; top: 0; right: 0; bottom: 0; margin: auto;  
  border-radius: 50%;  
  box-shadow: 0 0 0 9999px □ rgba(0, 0, 0, 0.75);  
}
```



四、CSS创意与视觉表现

• 4.2 图形-镂空

外圆角选项卡

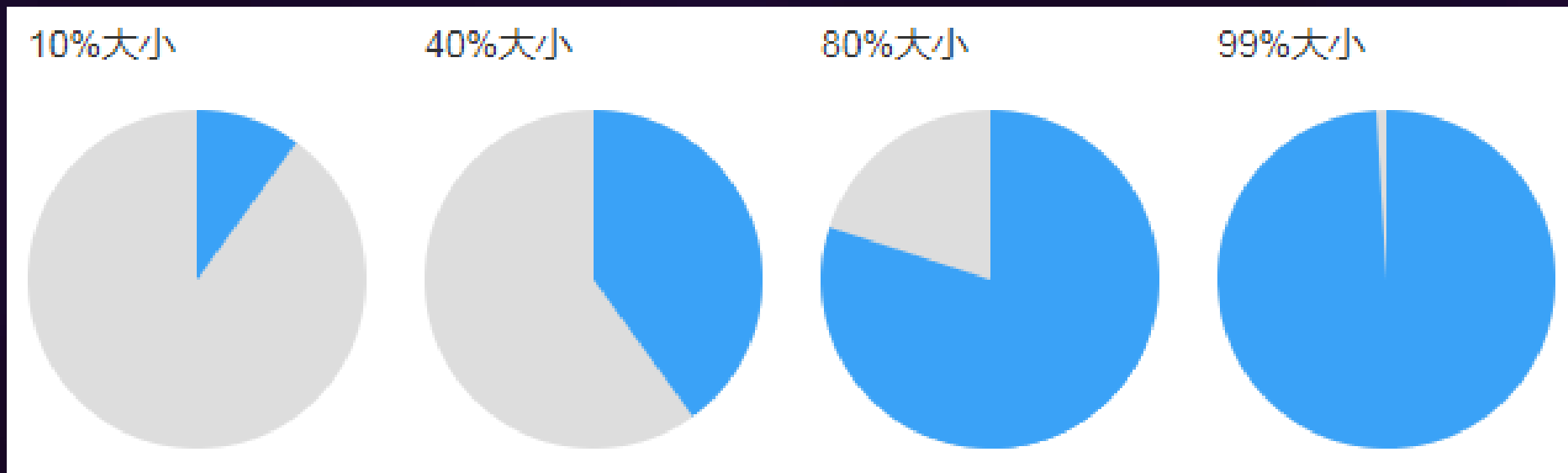


```
.quar-radius-{  
width: 96px; height: 96px;  
border-radius: 0 0 50% 0;  
clip-path: polygon(0% 0, 100% 0, 100% 100%, 0 100%);  
box-shadow: 0 0 0 100px blue rgb(58, 162, 247);  
}
```



四、CSS创意与视觉表现

• 4.3 动画-饼图



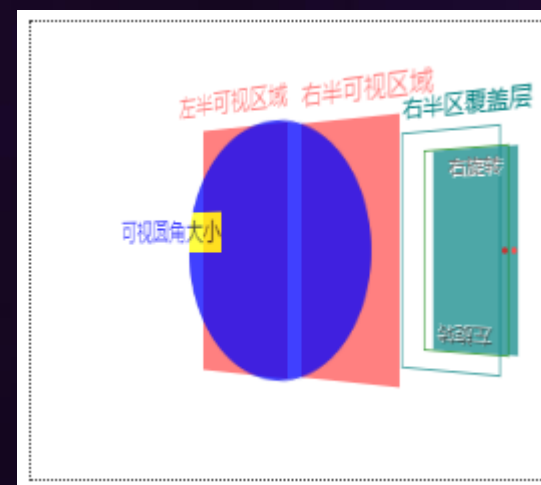
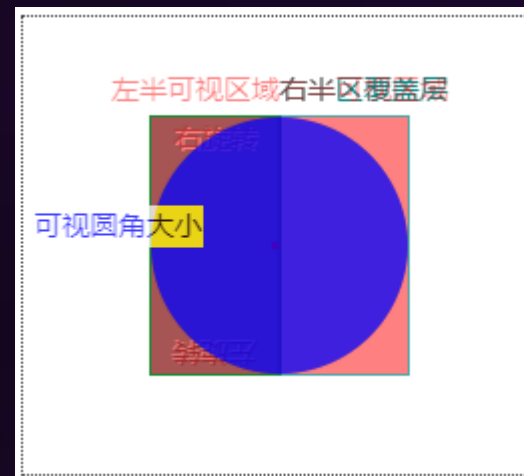
如何根据百分比属性显示对应的饼图？

四、CSS创意与视觉表现

• 4.3 动画-饼图

原理简介:

- 1.分左右两个半区，溢出隐藏；
- 2.进度前50%左半区隐藏，右半区旋转；
- 3.进度后50%左半区显示并旋转，右半区色块完整覆盖；



四、CSS创意与视觉表现

• 4.3 动画-饼图

```
.pie-left,
.pie-right {
  width: 50%; height: 100%; float: left;
  position: relative; overflow: hidden;
}
.pie-left::before,
.pie-right::before,
.pie-right::after {
  content: ''; position: absolute; width: 100%; height: 100%;
  background-color: #3498db;
}
.pie-left::before {
  left: 100%; transform-origin: left;
  transform: rotate(calc(3.6deg * (var(--percent) - 50)));
  opacity: calc(99999 * (var(--percent) - 50));
}
.pie-right::before {
  right: 100%; transform-origin: right;
  transform: rotate(calc(3.6deg * var(--percent)));
}
.pie-right::after {
  opacity: calc(99999 * (var(--percent) - 50));
}
```

左半区前50%进度不显示

右半区覆盖层后50%进度显示



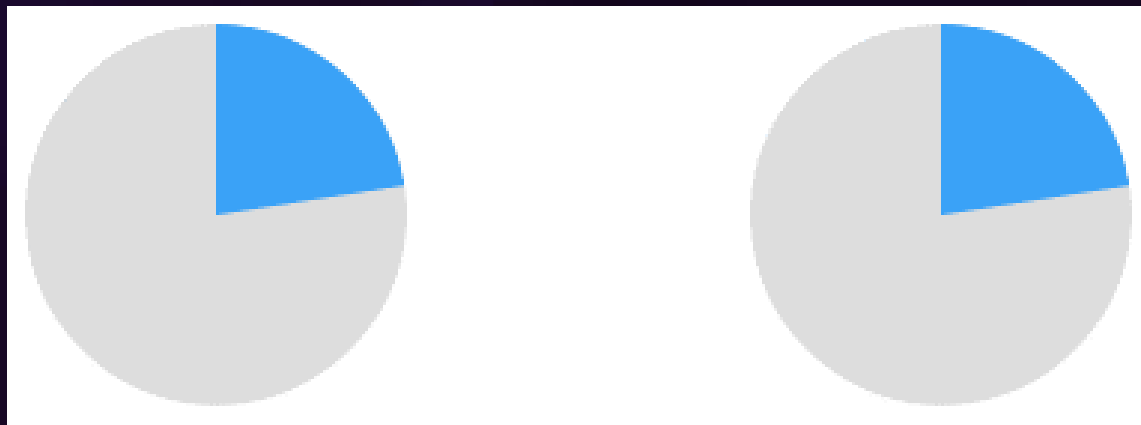
opacity负数浏览器按照0解析;
opacity大大于1, 浏览器按照1解析;

```
<div class="pie-simple" style="--percent: 80;">
  <div class="pie-left"></div>
  <div class="pie-right"></div>
</div>
```


四、CSS创意与视觉表现

- 4.3 动画-饼图

单循环与交替循环

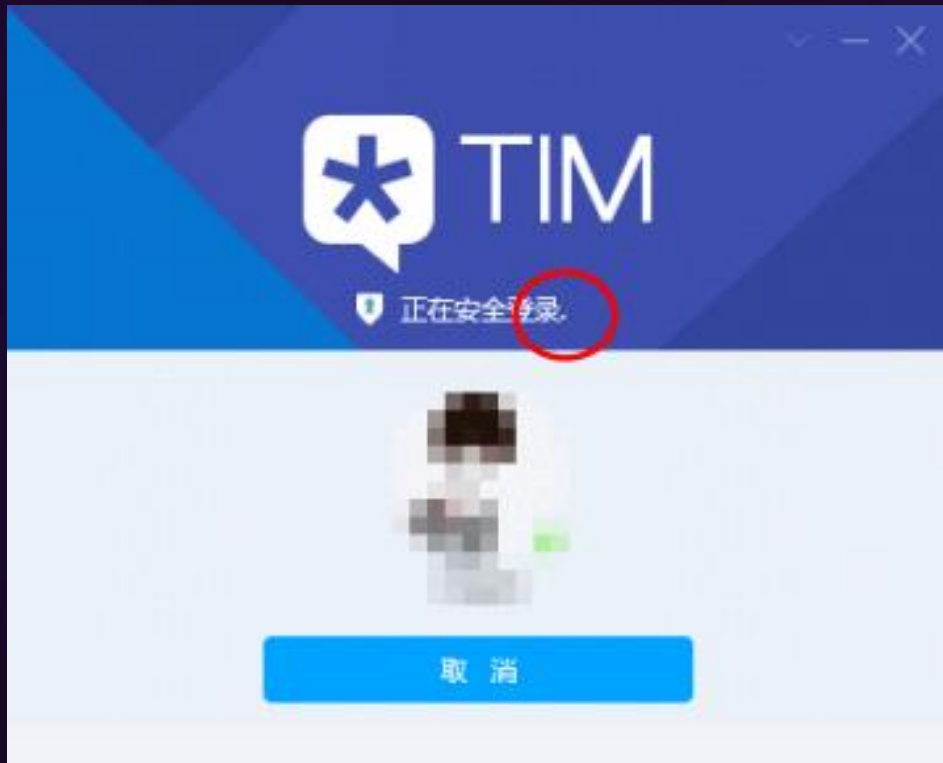


适用于进度示意

适用于loading提示

四、CSS创意与视觉表现

• 4.4 动画-打点



提交订单中.

四、CSS创意与视觉表现

• 4.4 动画-打点

订单提交中<dot></dot>

```
dot::before {  
  content: '...';  
  position: absolute;  
  animation: dot 3s infinite step-start both;  
}  
dot:after {  
  content: '...';  
  color: transparent;  
}  
@keyframes dot {  
  33% { content: '.'; }  
  66% { content: '..'; }  
}
```



- 易理解易上手
- 颜色、字号天然自适应



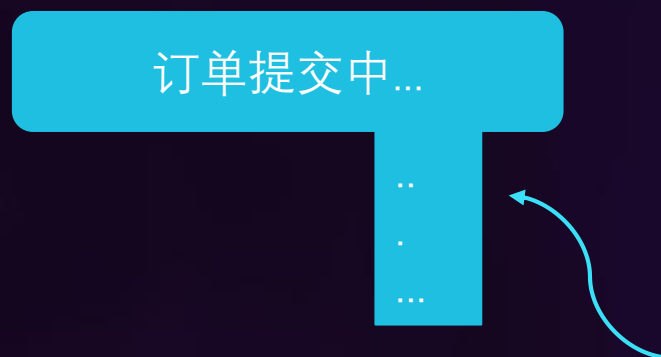
- IE不兼容
- Safari不兼容

四、CSS创意与视觉表现

• 4.4 动画-打点

订单提交中<dot></dot>

```
dot {  
  display: inline-block;  
  height: 1em; line-height: 1;  
  text-align: left;  
  vertical-align: -.25ex;  
  overflow: hidden;  
}  
dot::before {  
  display: block;  
  content: '...\A..\A.';  
  white-space: pre-wrap;  
  animation: dot1 3s infinite step-start both;  
}  
@keyframes dot1 {  
  33% { transform: translateY(-2em); }  
  66% { transform: translateY(-1em); }  
}
```

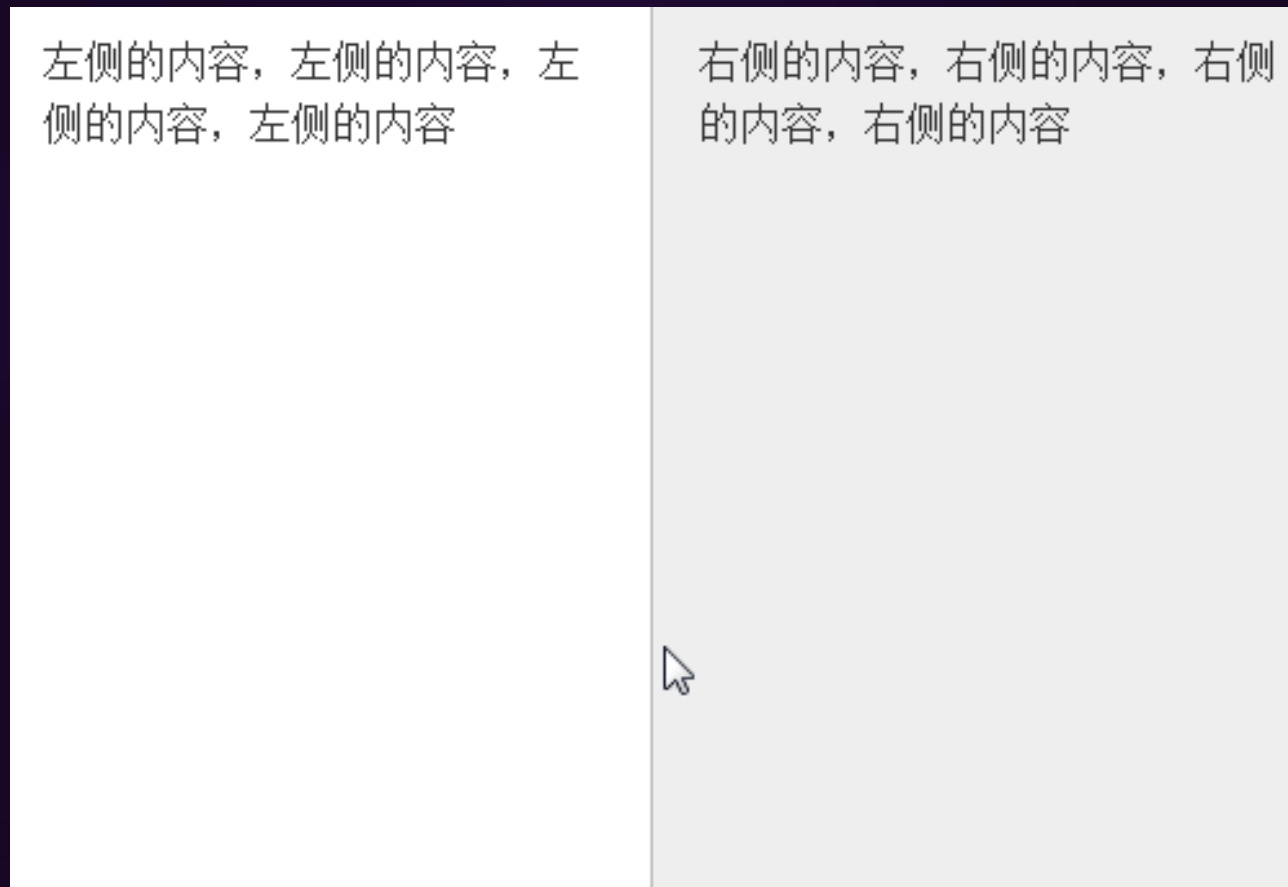


垂直位移形成动画

- \A指LF换行符
- \D指CR回车符

四、CSS创意与视觉表现

- 4.5 行为-分栏宽度拉伸



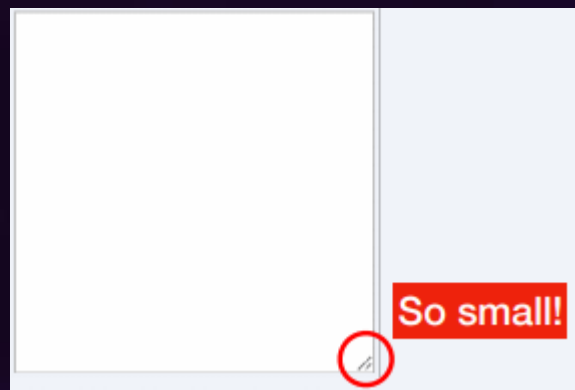
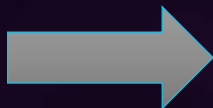
如何不使用JavaScript实现分栏宽度拖动调整？

四、CSS创意与视觉表现

• 4.5 行为-分栏宽度拉伸

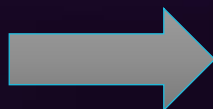
1.resize拉伸

```
.resize-bar-{  
  width: 200px; height: 200px;  
  cursor: ew-resize; opacity: 0;  
  resize: horizontal;  
  overflow: scroll;  
}
```



2.扩大拉伸的垂直范围

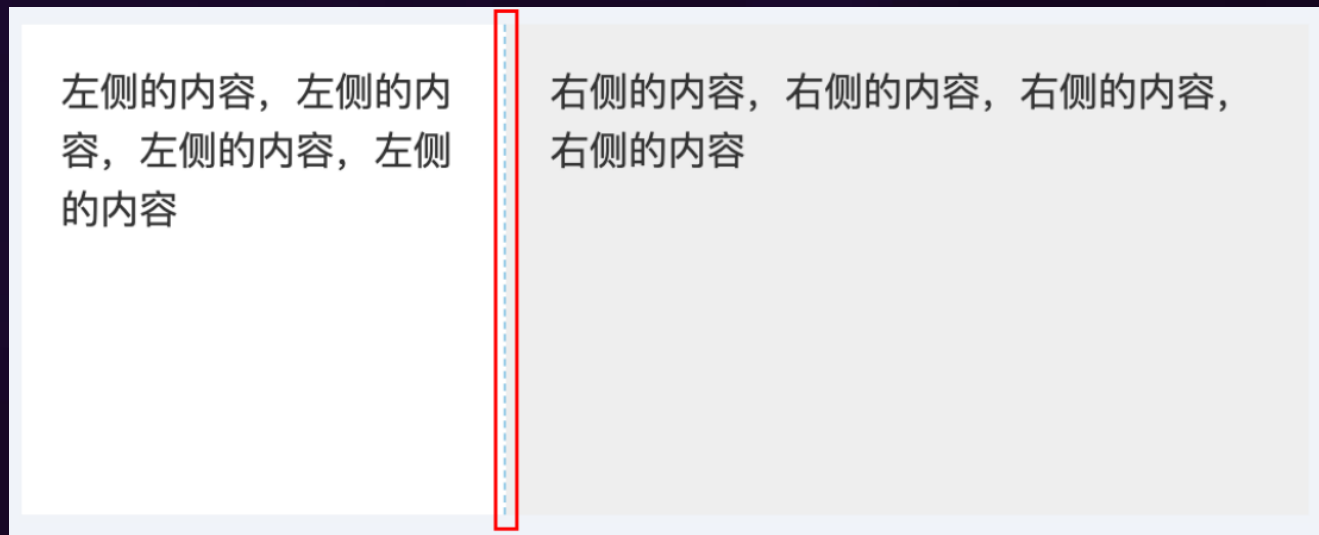
```
.resize-bar::-webkit-scrollbar-{  
  width: 200px; height: 200px;  
}
```



四、CSS创意与视觉表现

- 4.5 行为-分栏宽度拉伸

3.藏在某一栏的后面，透出部分宽度用来拖拽



4.采用宽度自适应布局

右侧内容自动填充剩余空间

5

练习作业

五、练习作业

- 实现这样一个效果：有个标题文字可能很长，标题后面会有一些标签，标签有长有短有多有少，超过一行时需要对标题进行超出打点处理。
 - 标题后面打点：智能利用除了标签以外的剩余空间，宽度自适应得2分，点点点效果得2分，兼容到IE8得1分；标题前面打点：（如下图）得1分；
 - 最终布局html语义良好、干净整洁，css无多余属性得1分；
 - 上下两块使用css的resize实现上下高度自由拉伸得3分。



提交产物：一个html文件

提交时间：2019年8月9日

THANKS

前端赋能 共创卓越