

## 第五篇 前沿技术拓展

# Contents

## 01.SVG 动画

svg

## 02. clip-path 裁剪

clip-path

## 03. 背景滤镜

backdrop-filter

## 04. 时间线动画

animation-timeline

## 05. 变量和计算函数

var calc()

## 06. 综合案例

# svg图片

SVG (Scalable Vector Graphics) 是一种基于 XML 的**矢量图形**标准, 由 W3C 制定, 支持无损缩放、交互性和动态效果。其核心特点包括:

## 矢量特性

无论放大或缩小均保持清晰。  
文件体积小, 适合网络传输,  
尤其适用于高分辨率设备。

## 可编辑性

直接通过文本编辑器修改  
SVG 代码, 支持颜色、形状、  
动画参数的实时调整。

## 交互性

支持 JavaScript 和 CSS 控制,  
可实现点击、悬停等动态响  
应。

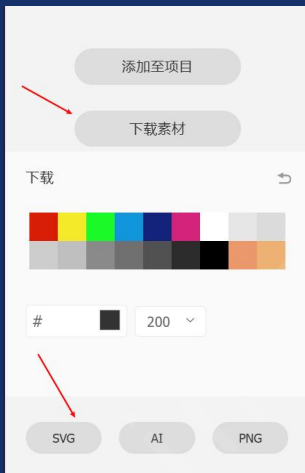
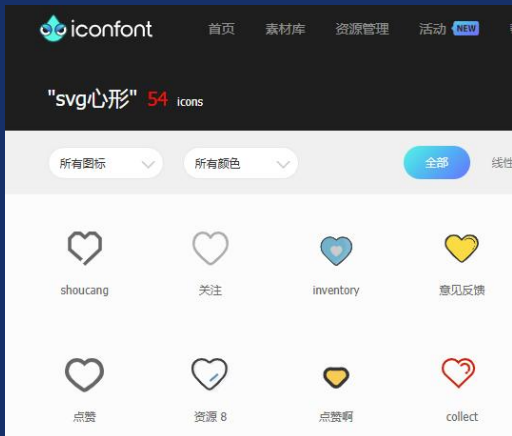
## 兼容性

主流浏览器 (Chrome、  
Firefox、Safari) 均原生支  
持 SVG, 移动端适配性强



# svg图片

下载svg图标:



```
<svg t="174721141484" class="icon" viewBox="0 0 1178 1024" version="1.1" xmlns="http://www.w3.org/2000/svg"
p-id="5266" width="200" height="200">
<path
d="M344.565805 0a344.956717 344.956717 0 0 1 244.816723 102.245039 344.295132 344.295132 0 0 1 589.111855 242.080165 509.
029932 509.029932 0 0 1-43.421996 237.569354 852.573262 852.573262 0 0 1-150.360351 199.979267c-107.387163 108.
469957-249.177174 202.595537-361.225707 239.824759a45.108105 45.108105 0 0 1-28.448178 0c462.989617 984.449362 321.
199807 890.31371 213.782372 781.873825a852.54319 852.54319 0 0 1-150.360351-199.979267 509.029932 509.029932 0 0 1-63.
421996-237.569354a344.295132 344.295132 0 0 1 344.325229 0.030072z m244.816723 217.330851a45.108105 45.108105 0 0 1-36.
89843-19.125836 254.139065 254.139065 0 0-462.057358 146.090117c0 152.555612 102.084462 287.579207 187.559502 373.
976264a957.073706 957.073706 0 0 311.366214 212.639609 957.073706 957.073706 0 0 311.336143-212.639609c85.55504-86.
397058 187.559502-221.420653 187.559502-373.976264a254.108993 254.108993 0 0 626.250887 198.174943a45.108105 45.108105
0 0 1-36.868359 19.155908z"
p-id="5267">
</path>
</svg>
```

<https://www.iconfont.cn/>

## svg组成

- SVG 根元素。SVG 图标必须包裹在 <svg> 标签内。
- <path> 路径。通过 d 属性定义路径指令。M: 移动画笔到坐标点, Z: 闭合路径。

# svg图标

svg图标常见CSS属性：

svg是行内块元素类似，可以设置大小、移动位置、动画等。但是有自己特殊的样式属性。



属性	描述	取值示例
fill	填充颜色（支持颜色值、渐变、图案）不需要改为 none	fill: #f00;
stroke	定义描边颜色（支持颜色值、渐变、图案）	stroke:#f00;
stroke-width	描边宽度（支持像素、百分比、px、em 等单位）	stroke-width: 2px;
stroke-dasharray	虚线模式（实线长度+间隔长度）	stroke-dasharray: 10 ;
stroke-dashoffset	调整虚线与间隔的起始位置	stroke-dashoffset: 100;

## svg图片

svg图标常见CSS属性:

stroke-dasharray: 100; 默认px 实线和间隔都是 100



stroke-dasharray: 100 50; 第一个值是实线, 第二个是间隔



# svg动画

svg图标常见CSS属性：

## 参观·导览



参观预约



建筑一览



校园地图



校园设施

# svg图标

svg图标常见CSS属性：

svg是行内块元素类似，可以设置大小、移动位置、动画等。但是有自己特殊的样式属性。



属性	描述	取值示例
fill	填充颜色（支持颜色值、渐变、图案）不需要改为 none	fill: #f00;
stroke	定义描边颜色（支持颜色值、渐变、图案）	stroke:#f00;
stroke-width	描边宽度（支持像素、百分比、px、em 等单位）	stroke-width: 2px;
stroke-dasharray	虚线模式（实线长度+间隔长度）	stroke-dasharray: 10 ;
stroke-dashoffset	调整虚线与间隔的起始位置	stroke-dashoffset: 100;



# svg动画

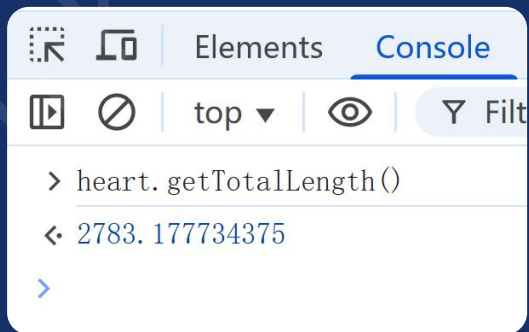
svg图标常见CSS属性：

stroke-dashoffset 用于调整虚线模式的起始偏移量。

增大 stroke-dashoffset，让虚线模式向路径起点移动，视觉上像线条被“擦除”。

获得svg 描边的长度：

```
<svg class="heart" t="17545550"
  xmlns="http://www.w3.org/2001/svg"
  <path id="heart"
    d="M32 407.584a279.584 279.584 0 0 1 279.584 279.584
    p-id="6245"></path>
</svg>
```



# svg动画

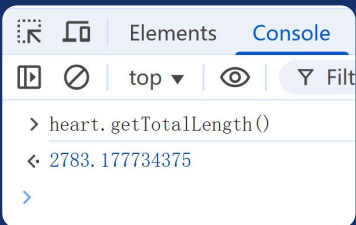
绘制动画（“画笔”效果）总结：

## 最经典的应用：

通过动画逐渐减小 stroke-dashoffset，让虚线模式“逆向移动”，视觉上像画笔沿着路径绘制线条。

## 实现步骤：

1. 计算路径总长度 L（通过 getTotalLength()）。
2. 设置 stroke-dasharray: X;
3. 初始 stroke-dashoffset: X;（实线段完全隐藏）。
4. 动画中逐渐将 offset减小到 0，实线段逐渐覆盖整个路径。



```
.box svg {  
  width: 3520px;  
  stroke-dasharray: 3852;  
  stroke-dashoffset: 3852;  
}
```

```
@keyframes move {  
  0% {  
    stroke-dashoffset: 3852;  
  }  
  100% {  
    stroke-dashoffset: 0;  
  }  
}
```



# svg动画

svg图标常见CSS属性:

## OPPO AI 里程碑



2024

### OPPO AI 手机战略

OPPO 承诺生成式 AI 功能将进入全产品系列，今年要让近 5 千万 OPPO 用户的手机搭载 GenAI 功能。



2024

### AI 手机白皮书

OPPO 联合 IDC 发布首个《AI手机白皮书》，宣布 AI 手机时代已经到来。

OPPO 推出首款 AI 手机，引领行业迈入 AI 手机新时代。

### 首款 OPPO AI 手机

2024



OPPO 于 2024 年成立了 OPPO AI 中心，将资源向 AI 集中，更加系统化地提升 AI 能力。

### OPPO AI 中心

2024

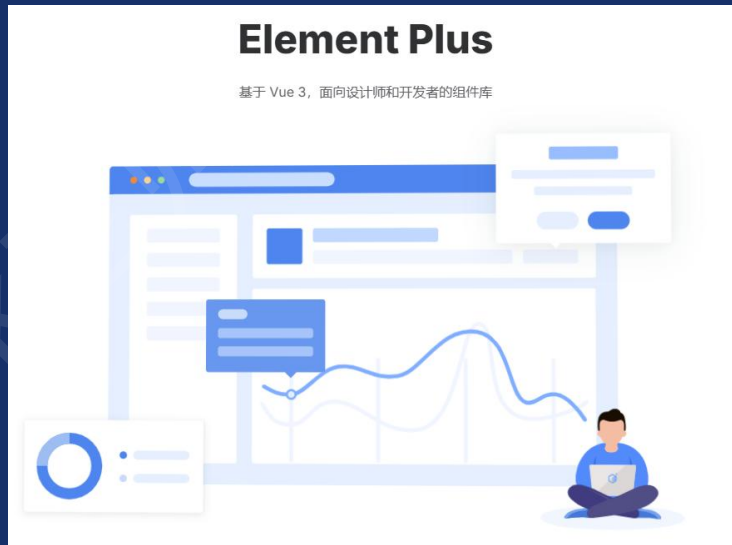


<https://www.oppo.com/cn/discover/technology/oppo-ai/>

## svg动画



<https://undraw.co/>



<https://element-plus.org/zh-CN/#/zh-CN>

# clip-path

clip-path 创建复杂的裁剪形状，使元素仅显示被裁剪区域内的部分。



## 语法

clip-path: 内置几何形。

□ 内置几何形。状如 circle()、polygon()

```
.circle {  
  clip-path: circle(40% at 50% 50%);  
}
```

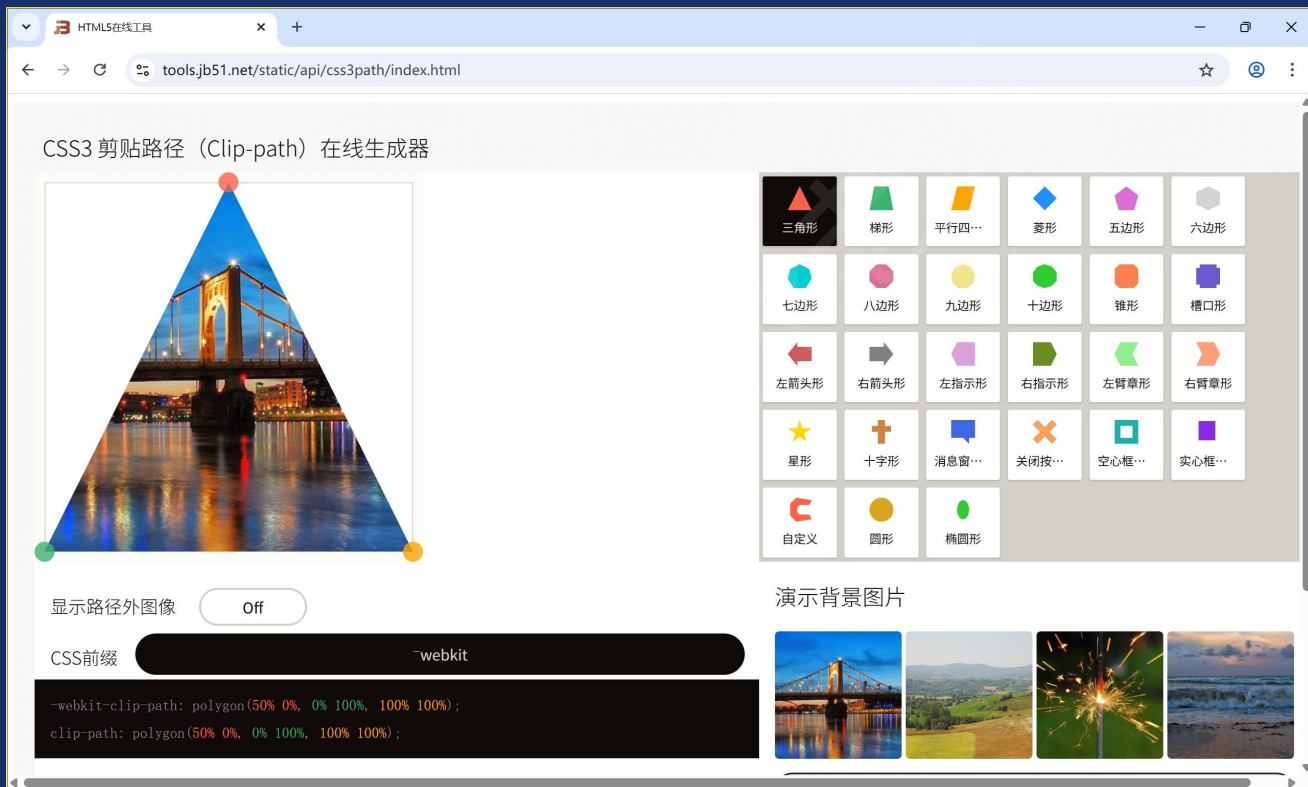
可视化工具：

<https://tools.jb51.net/static/api/css3path/index.html>



# clip-path

clip-path 创建复杂的裁剪形状，使元素仅显示被裁剪区域内的部分。



# clip-path

clip-path 创建复杂的裁剪形状，使元素仅显示被裁剪区域内的部分。

可视化工具：

<https://tools.jb51.net/static/api/css3path/index.html>



# clip-path

clip-path 创建复杂的裁剪形状，使元素仅显示被裁剪区域内的部分。

可视化工具：

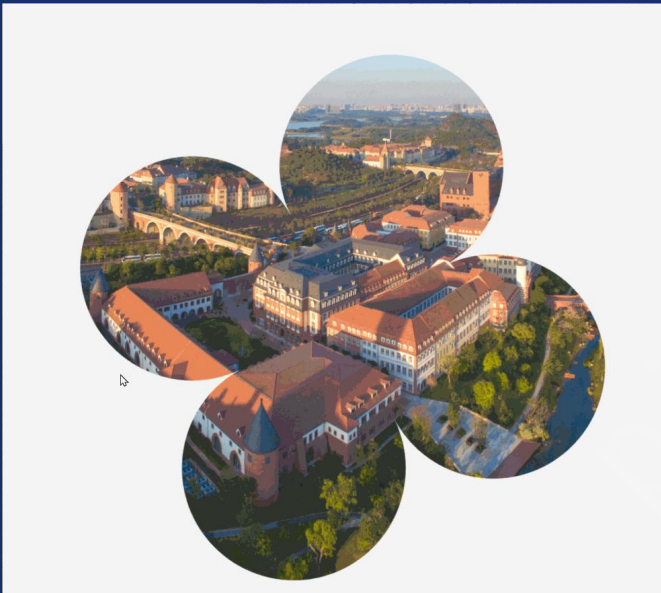
<https://tools.jb51.net/static/api/css3path/index.html>





# 旋转 turn

请看效果：



新知识：

1turn 就是旋转一周的意思 等价于 360deg

```
@keyframes rotate {  
  0% {  
    transform: rotate(0deg);  
  }  
  100% {  
    transform: rotate(1turn);  
  }  
}
```

<https://developer.huawei.com/home/hdc/hdc2025.html>

# 滤镜 filter

CSS滤镜通过filter属性实现，可对元素及其子元素进行实时图像处理，无需修改原始素材。

滤镜函数	语法示例
blur()	高斯模糊 blur(5px)
brightness()	调整元素亮度 brightness(150%) 100%（或 1）为原始亮度； - <100%（或 <1）变暗； - >100%（或 >1）变亮。
contrast()	调整元素对比度 contrast(200%) 同上
saturate()	调整背景饱和度 取值 0%（无色彩）~ 100%（原饱和度）~ >100%（更高饱和度）。
grayscale()	将元素转换为灰度图 grayscale(100%) - 0%（或 0）为原始色彩；- 100%（或 1）为完全灰度。
hue-rotate()	调整元素色相（改变颜色倾向） hue-rotate(90deg)
sepia()	将元素转换为深褐色（复古效果） sepia(70%) - 0%（或 0）为原始色彩； - 100%（或 1）为完全深褐色。
drop-shadow()	为元素添加投影 类似 box-shadow，但支持非矩形元素。drop-shadow(5px 5px 5px #669)

## 滤镜 filter



filter: brightness(150%);  
亮度



filter: blur(5px);  
高斯模糊



正常图片



filter: contrast(150%);  
对比度



filter: saturate(150%);  
饱和度

## 滤镜 filter



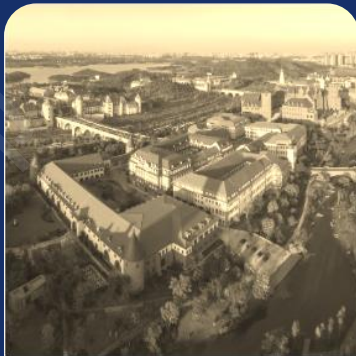
filter: hue-rotate(90deg);  
色相旋转



filter: grayscale(100%);  
灰度



正常图片



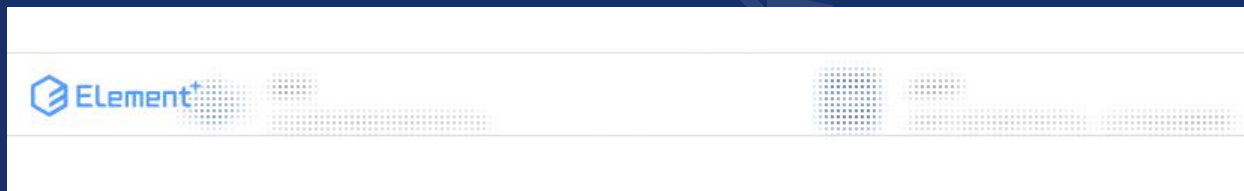
filter: sepia(100%);  
褐色复古效果



filter: drop-shadow(5px  
5px 5px red);  
阴影

# 背景滤镜

backdrop-filter是 CSS 中的一个图形滤镜属性，主要用于对元素背后的区域（即“背景”）应用滤镜效果。它通常与半透明背景配合使用，能实现类似“毛玻璃”的视觉效果



## 背景滤镜

backdrop-filter是 CSS 中的一个图形滤镜属性，主要用于对元素背后的区域（即“背景”）应用滤镜效果。它通常与半透明背景配合使用，能实现类似“毛玻璃”的视觉效果

滤镜函数	语法示例
blur()	高斯模糊 blur(5px)
brightness()	调整元素亮度 brightness(150%) 100%（或 1）为原始亮度； - <100%（或 <1）变暗； - >100%（或 >1）变亮。
contrast()	调整元素对比度 contrast(200%) 同上
saturate()	调整背景饱和度 取值 0%（无色彩）~ 100%（原饱和度）~ >100%（更高饱和）。
grayscale()	将元素转换为灰度图 grayscale(100%) - 0%（或 0）为原始色彩；- 100%（或 1）为完全灰度。
hue-rotate()	调整元素色相（改变颜色倾向） hue-rotate(90deg)
sepia()	将元素转换为深褐色（复古效果） sepia(70%) - 0%（或 0）为原始色彩； - 100%（或 1）为完全深褐色。
drop-shadow()	为元素添加投影 类似 box-shadow，但支持非矩形元素。drop-shadow(5px 5px 5px #669)

# 动画时间线

`animation-timeline` 是 CSS 中用于控制动画时间线的核心属性，它允许开发者将动画进度与特定事件（如滚动、视口可见性）绑定，从而实现更复杂的交互效果。

[全球支付](#)[平台支付和 BaaS](#)[收入和财务自动化](#)[授权优化](#)[与 Stripe 携手合作](#)[用例](#)[资源](#)[联系销售 >](#)

<https://www.porsche.cn/china/zh/models/cayenne/cayenne-models/cayenne-passion/>

<https://stripe.com/zh-us/enterprise>



## 动画时间线

`animation-timeline` 是 CSS 中用于控制动画时间线的核心属性，它允许开发者将动画进度与特定事件（如滚动、视口可见性）绑定，从而实现更复杂的交互效果。

```
.scrollbar {  
  width: 0%;  
  height: 3px;  
  background: linear-gradient(90deg,  
  animation: move 2s;  
  /* 绑定时间线 */  
  animation-timeline: scroll();  
}
```

```
@keyframes move {  
  0% {  
    width: 0;  
  }  
  
  100% {  
    width: 100%;  
  }  
}
```

滚动时间线：动画进度与页面或容器的滚动位置关联



## 动画时间线

`animation-timeline` 是 CSS 中用于控制动画时间线的核心属性，它允许开发者将动画进度与特定事件（如滚动、视口可见性）绑定，从而实现更复杂的交互效果。

```
.box img {  
  opacity: 0.2;  
  animation: scale 1s linear forwards;  
  animation-timeline: view();  
}  
  
@keyframes scale {  
  0% {  
    transform: scale(1);  
    opacity: 0.2;  
  }  
  
  100% {  
    transform: scale(1.5);  
    opacity: 1;  
  }  
}
```

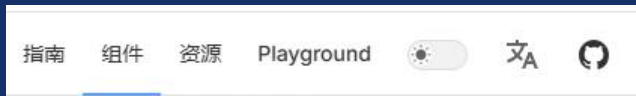
视图时间线：动画进度与元素进入/离开视口的可见性关联

# 变量和函数

CSS增强了逻辑性和动态计算能力，比如CSS变量和一些动态函数(calc()、clamp()等)

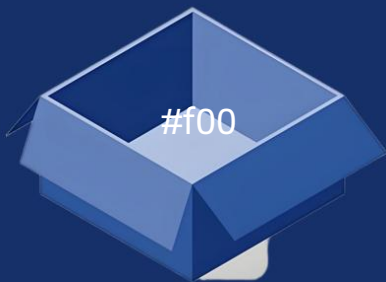
使用场景：

- ❑ 主题切换。
- ❑ 响应式设计。
- ❑ 交互动画等



# 变量

CSS 变量(也称为自定义属性)是一种动态存储和复用值的机制。它让样式管理更灵活、可维护。  
变量可以理解为一个容器，里面可以存放数据。



定义变量:

通过 --变量名 定义变量。

```
--color: #000;
```

使用变量:

通过 var(--变量名) 使用定义的变量。

```
color: var(--color);  
background-color: var(--bgcolor);
```

# 变量

作用域：变量的作用范围，哪个区域有效。

## 定义变量：

```
:root { /* 全局变量 */  
  --color: #000;  
}  
  
.box { /* 局部变量 */  
  --bgcolor: pink;  
}
```

```
:root {  
  --brand_pink: var(--Pi5);  
  --brand_pink_thin: var(--Pi1);  
  --brand_blue: var(--Lb5);  
  --brand_blue_thin: var(--Lb1);  
  --stress_red: var(--Re5);  
  --stress_red_thin: var(--Re1);  
}
```

- 全局作用域：通过 `:root` 定义，作用于整个网页。
- 局部作用域：在特定选择器中定义，仅影响该元素及其子元素。

## 使用变量：

```
.nav {  
  color: var(--color);  
  background-color: var(--bgcolor);  
}
```

- 通过 `var(--变量名)` 使用定义的变量。

```
#i_cecream {  
  margin: 0 auto;  
  max-width: 2560px;  
  background-color: var(--bg1);  
}
```

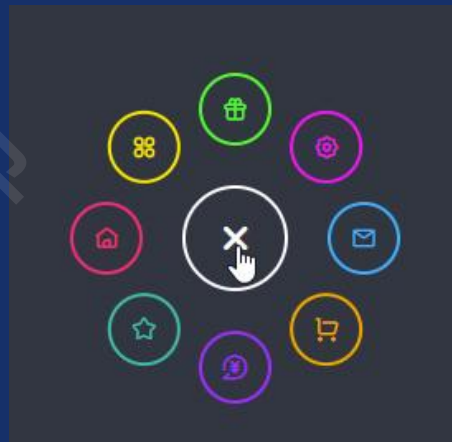
# CSS3 计算能力

calc() 函数可以执行数学运算（加减乘除），支持混合单位

语法：

```
.box {  
  width: calc(100% - 20px); /* 父宽度减固定值 */  
  height: calc(var(--base-size) * 1.5); /* 变量与运算结合 */  
}
```

□ 加减乘除运算写法: + - \* / 注意符号左右两侧保留空格



```
.carousel .carousel-container .vui_carousel .carousel-footer-text[data-v-9bad986e]  
  position: absolute;  
  z-index: 2;  
  bottom: 42px;  
  left: 15px;  
  width: calc(100% - 100px)  
}
```

<https://www.iqiyi.com/>

# 综合案例1

## 动感菜单

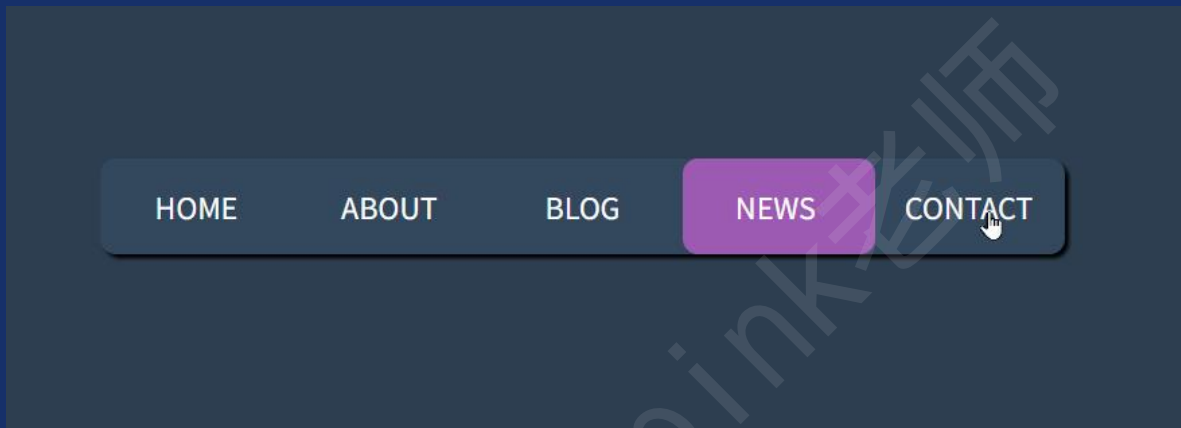


```
transition: .5s;  
opacity: 0;  
transform: translateX(-250px);  
transition-delay: calc(0.1s * var(--i));
```

```
<li style="--i:0">主页</li>  
<li style="--i:1">关于我们</li>  
<li style="--i:2">售后服务</li>  
<li style="--i:3">产品介绍</li>  
<li style="--i:4">联系我们</li>
```

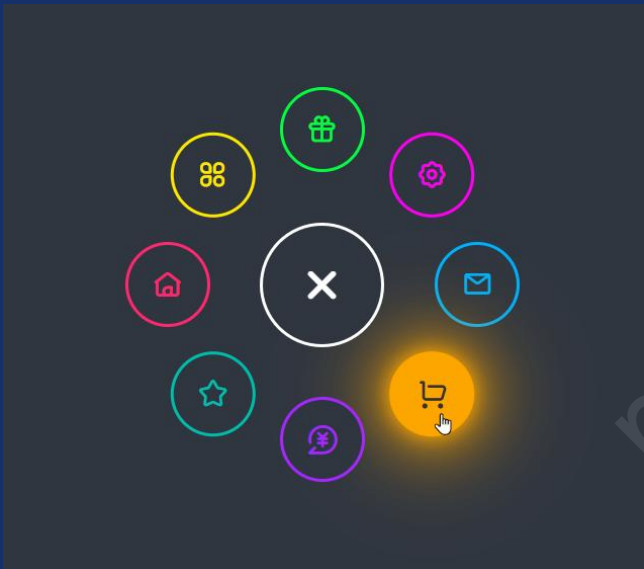
## 综合案例2

滑动导航栏



## 综合案例3

炫酷导航条





## 综合案例4

### 风景卡片滚动效果



# vw/vh

vw和vh是 CSS 中的视口单位 (Viewport Units)，用于基于浏览器视口 (Viewport) 的尺寸来定义元素的尺寸或位置。

vw 和 vh始终基于浏览器可视区域的实时尺寸计算 (不包含浏览器工具栏、标签栏等界面元素)，，它们的值会自动更新。特别适合响应式设计、全屏布局、移动适配等场景

**vw** (Viewport Width)：表示视口宽度的 1%。

例如，若当前浏览器视口的宽度为 1920px，则  $1\text{vw} = 1920\text{px} \times 1\% = 19.2\text{px}$ 。

**vh** (Viewport Height)：表示视口高度的 1%。

例如，若当前视口的高度为 1080px，则  $1\text{vh} = 1080\text{px} \times 1\% = 10.8\text{px}$ 。

T H A N K Y O U

行无止境