

动画 / Animations

开发高性能动画的技术。

Yurica Xu

2022/12/20

Yurica Xu

随时可能出现的地球生命体，闪烁技能✨点满。

白天是个前端，晚上兴趣广泛。

积极参与开源社区。

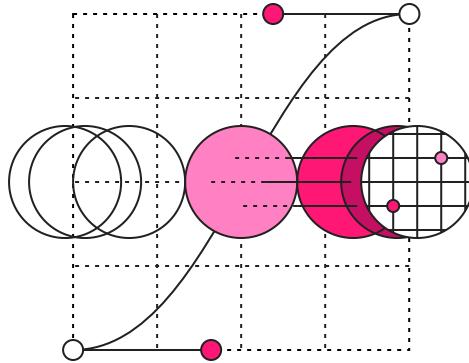
曾在 @Ctrip, Inc. + @轻流 实习。



👤 [etes.dev](#)

🐱 [etes-dev](#)

✉️ iherewithmyheart@gmail.com



1. 浏览器如何渲染动画?
 1. 渲染管线
 2. 合成属性
2. 如何创建高性能的CSS动画
 1. 避免触发布局或绘画的属性
 2. 强制创建图层
 3. 两个例子
 4. 如何调试发现缓慢动画
 5. 检查动画是否丢帧
 6. 检查动画是否触发绘画
 7. 检查动画是否创建图层
3. 实例

浏览器如何渲染动画？

为什么有些动画很慢？

现代浏览器只有两个 CSS 属性可以低成本地设置动画：`transform` 和 `opacity`。

渲染管线

要在网页上显示某些内容，浏览器必须执行以下顺序步骤

1. 样式：计算应用于元素的样式
2. 布局：为每个元素生成几何图形和位置
3. 绘画：将每个元素的像素填充到图层中
4. 合成：将层绘制到屏幕上

更多

- [From Braces to Pixels](#)
- [Inside look at modern browser](#)

合成属性

合成是将页面分成若干层，将有关页面外观的信息转换为像素（光栅化），并将各层放在一起以创建页面（合成）。

这就是为什么 `opacity` 属性包含在动画成本低的属性列表中。

只要此属性在其自己的层中，GPU 就可以在合成步骤中处理对其的更改。

基于 Chromium 的浏览器和 WebKit 为任何具有 CSS `transform` 或 `opacity` 动画的元素创建一个新层。

更多

- GPU Animation: Doing It Right

哪个更好？

你可能会问：从性能的角度来看，使用 CSS 还是 JAVASCRIPT 做动画更好？

基于 CSS 的动画，以及 Web 动画（在支持API的浏览器中），通常在一个被称为**合成器线程**的线程中处理。

这与浏览器的主线程不同，在主线程中，样式、布局、绘画和 JavaScript 被执行。

这意味着，如果浏览器正在主线程上运行一些昂贵的任务，这些动画可以继续进行而不被打断。

在许多情况下，对 `transform` 和 `opacity` 的其它变化也可以由**合成器线程**处理。

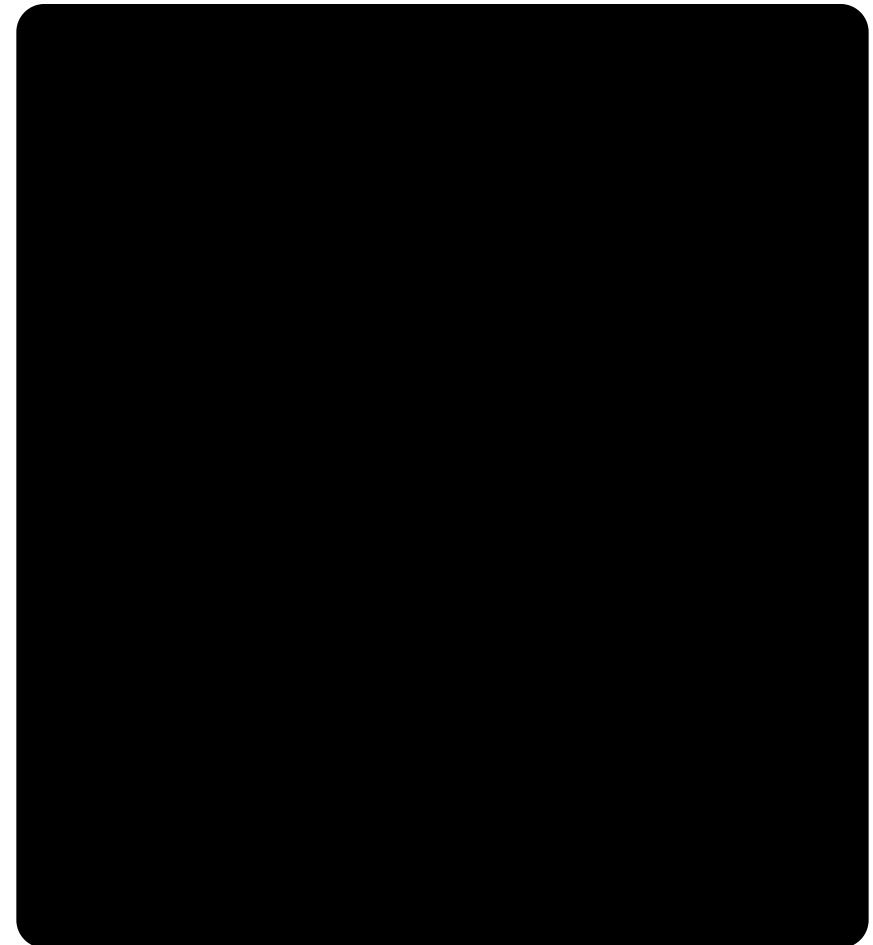
如何创建高性能的CSS动画

- `transform`
- `opacity`
- `will-change`

移动元素

要移动元素，请使用 `transform` 属性的 `translate` 或 `rotation` 关键字。

```
.animate {  
  animation: slide-in 0.7s both;  
}  
  
@keyframes slide-in {  
  0% {  
    transform: translateY(-1000px);  
  }  
  100% {  
    transform: translateY(0);  
  }  
}
```

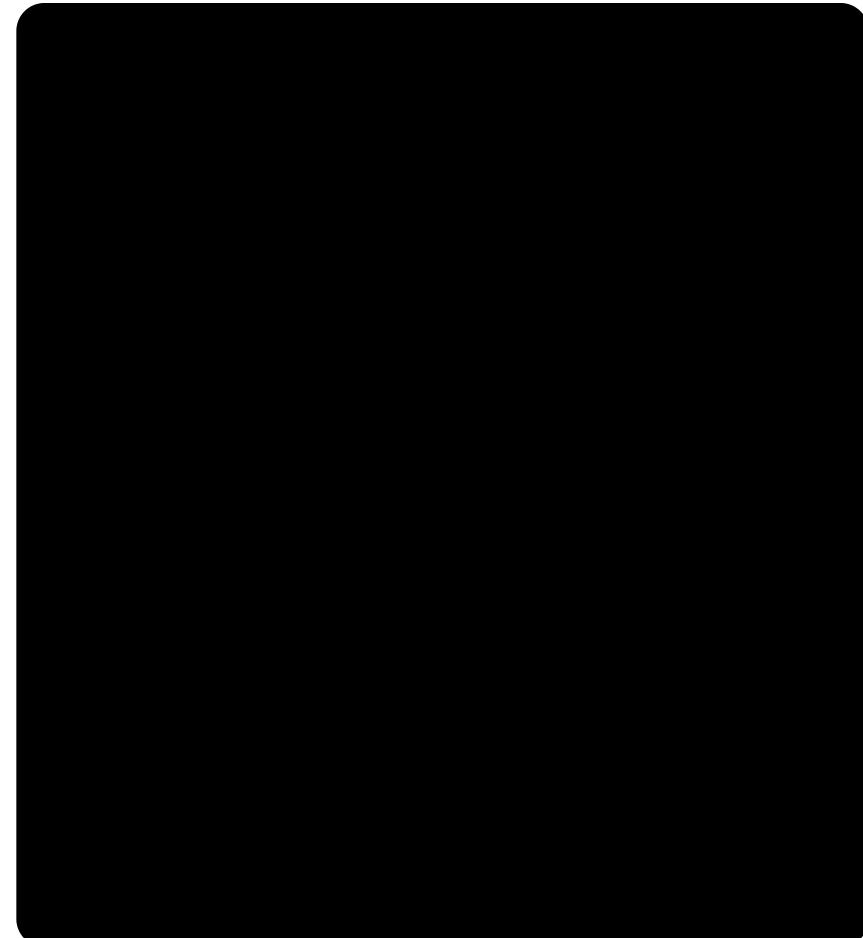


更改元素的可见性

要显示或隐藏元素，请使用 `opacity`。

```
.animate {
  animation: opacity 2.5s both;
}

@keyframes opacity {
  0% {
    opacity: 1;
  }
  50% {
    opacity: 0;
  }
  100% {
    opacity: 1;
  }
}
```



避免触发布局或绘画的属性

在将任何 CSS 属性用于动画（`transform` 和 `opacity` 除外）之前，确定该属性对渲染管道的影响。

除非绝对必要，否则避免使用任何会触发布局或绘制的属性。

强制创建图层

正如为什么有些动画很慢中所解释的那样，通过将元素放置在新的图层上，可以重新绘制它们，而无需重新绘制布局的其余部分。

浏览器通常会很好地决定哪些项目应该放在一个新的图层上，但你可以用 `will-change` 属性手动强制创建图层。顾名思义，这个属性告诉浏览器，这个元素将以某种方式被改变。



由于图层创建可能会导致其他性能问题，因此不应将此属性用作过早的优化。

相反，你应该只在看到卡顿并认为将元素提升到新层可能有帮助时才使用它。

`transform: translateZ(0)`

更多

- [Everything You Need To Know About The CSS will-change Property](#)

两个例子

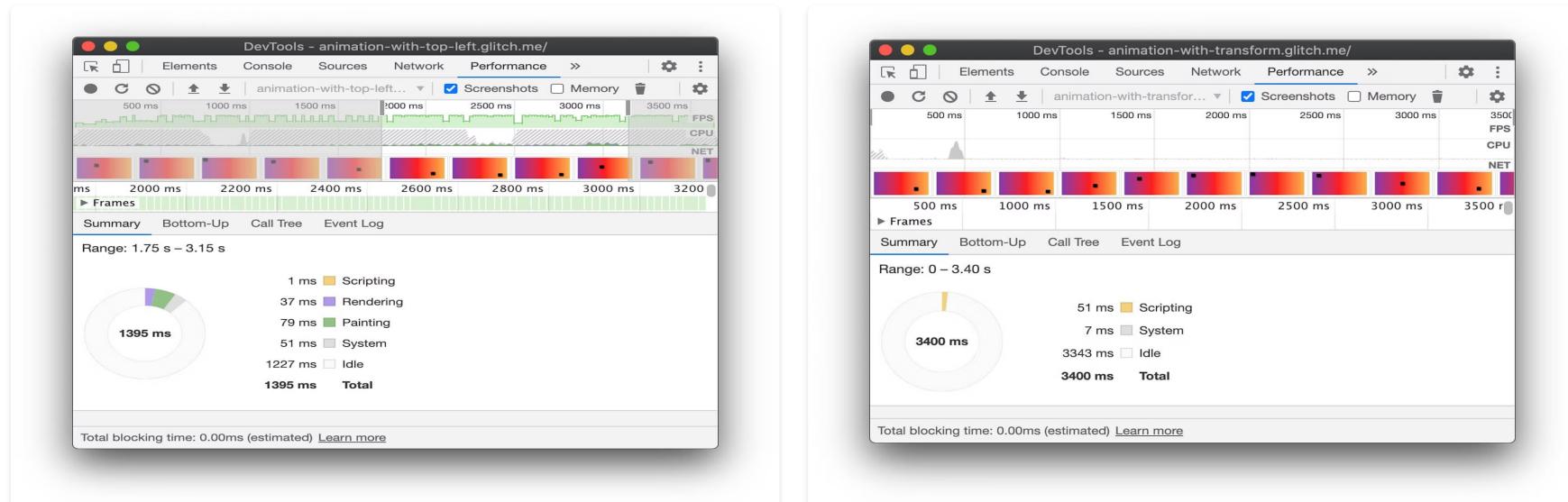
```
.box {  
  .box {  
    position: absolute;  
    top: 10px;  
    left: 10px;  
    animation: move 3s ease infinite;  
  }  
  
  @keyframes move {  
    50% {  
      transform: translate(calc(90vw - 200px),  
        calc(90vh - 160px));  
    }  
  }  
}
```

如何调试发现缓慢动画

检查动画是否触发布局

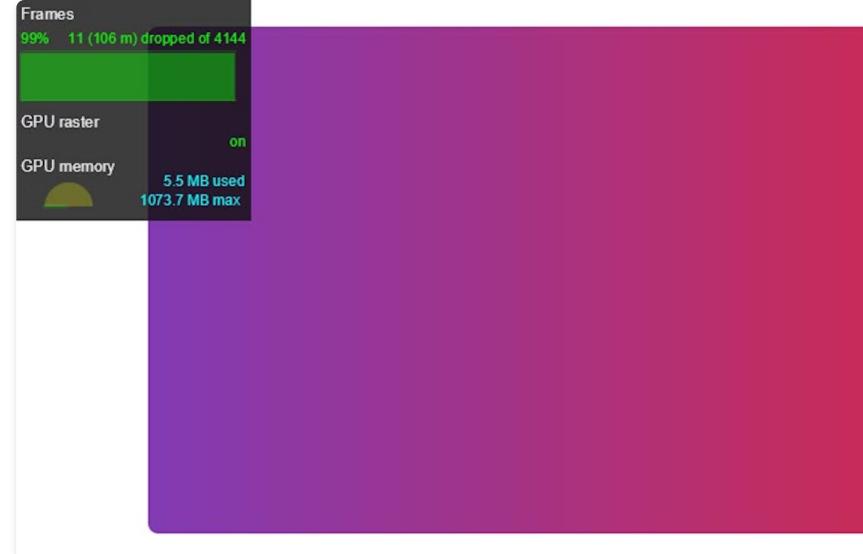
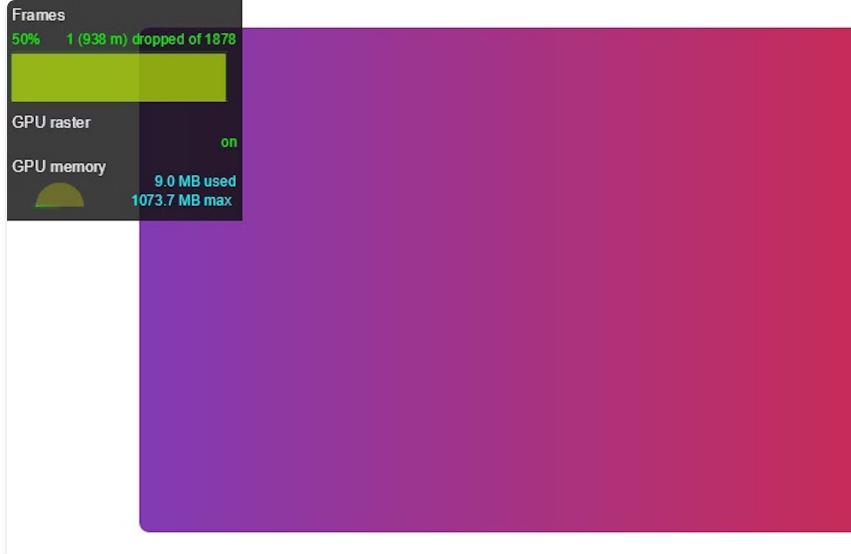
1. 打开 Performance 面板
2. 在动画发生时记录运行时性能
3. 检查 Summary 选项卡

如果你在 Summary 选项卡中看到 Rendering 的数值非零，这可能意味着你的动画会导致浏览器进行布局工作。



检查动画是否丢帧

1. 打开 Chrome 开发工具的 Rendering 选项卡
2. 勾选 FPS 仪表盘
3. 在动画运行时观察这些值



现在已经可以直接看到帧率了！

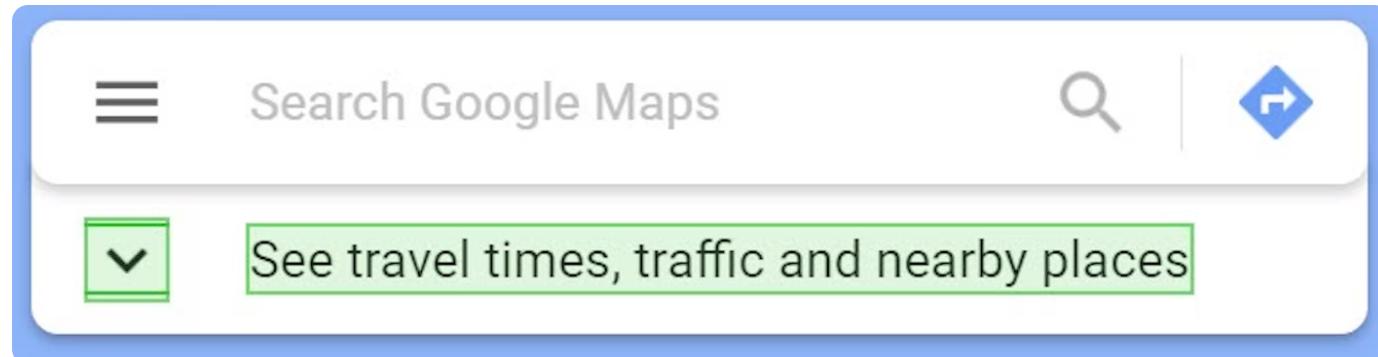
检查动画是否触发绘画

谈到绘画，有些属性比其它属性更消耗性能

例如，任何涉及模糊的东西（比如说阴影）都要比画一个红色方框花费更多的时间。然而，在CSS方面，这并不总是显而易见的：`background: red` 和 `box-shadow: 0, 4px, 4px, rgba(0,0,0,0.5)` 不一定看起来有巨大的性能差别，但它们确实如此。

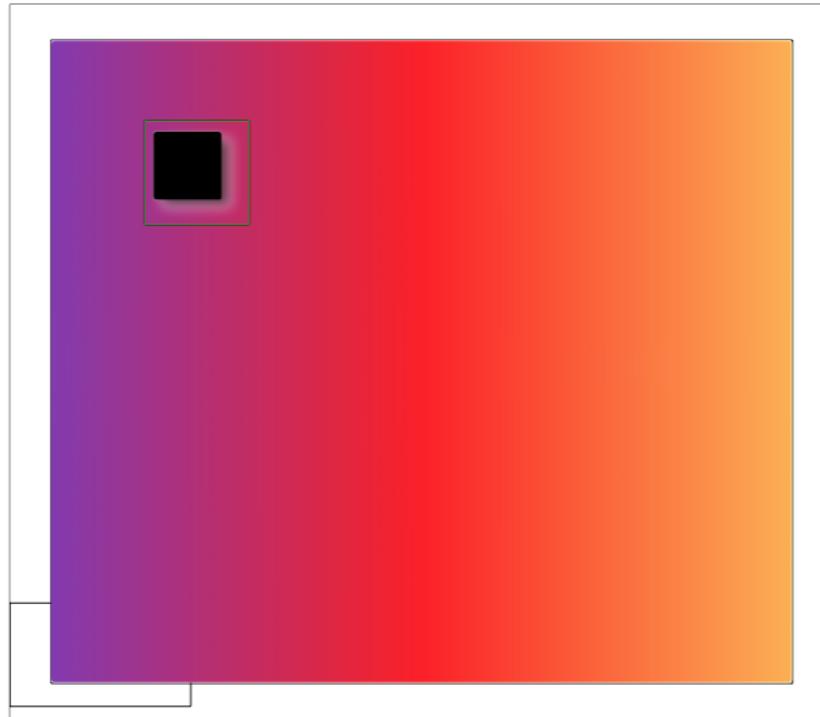
浏览器开发工具可以帮助你确定哪些区域需要重新绘画，以及与绘画有关的性能问题。

1. 打开 Chrome 开发工具的 **Rendering** 选项卡
2. 勾选 **Paint Flashing**
3. 触发动画并观察



检查动画是否创建图层

1. 打开 Chrome 开发工具的 **Layers** 选项卡
2. 在左侧侧边栏中点击图层并观察



实例

 pr

```
@keyframes right-to-left {
  from {
-   margin-left: -22px;
+   transform: translateX(22px);
    opacity: 0;
  }
  to {
-   margin-left: -54px;
+   transform: translateX(0);
    opacity: 1;
  }
}
```

```
.app-list-card-body {
+ position: relative;
  width: 240px;
  height: 108px;
  margin: 0 16px 16px 0;
  border-radius: style.$border-radius-medium;
  box-shadow: style.$shadow-base;
  cursor: pointer;
  background: style.$background-color-white;
- transition: box-shadow 0.15s ease-out 0s;
- &:hover {
  }
```

```
.bottom-line {
- animation: bottom-line-grow 0.25s cubic-bezier(0.25, 0
+ width: 97px;
+ transform-origin: left;
+ animation: bottom-line-grow 0.6s cubic-bezier(0.25, 0.1
}
@keyframes bottom-line-grow {
  from {
-   width: 0;
+   transform: scaleX(0);
+   transform: translateZ(0);
  }
```

请查阅 web.dev / developer.chrome.com 了解更多

谢谢!

幻灯片可在 etes.dev 查看与下载