

Web Vitals

谷歌的新一代 Web 性能体验和质量指标



什么是Web Vitals

Web Vitals是Google的一项重大举措，旨在为质量信号提供统一的指导，这对于在Web上提供出色的用户体验来说很重要。网站的开发者需要了解自己的网站给用户带来的体验，但不一定要成为性能优化的专家。Web Vitals旨在简化流程，并帮助网站开发者聚焦在核心性能指标上，也称为Core Web Vitals。



Core Web Vitals

(Loading)

LCP

Largest Contentful Paint



(Interactivity)

FID

First Input Delay



(Visual Stability)

CLS

Cumulative Layout Shift



最大内容绘制

首次输入延迟

累积布局偏移

LCP (Largest Contentful Paint)

加载体验 · 最大内容绘制

如何准确衡量网站的性能

思考

以用户为中心的性能指标，应该能回答以下四个问题

- **是否发生?** 导航是否成功启动？服务器是否有响应？
- **是否有用?** 是否已渲染可以与用户互动的足够内容？
- **是否有用?** 用户可以与页面交互，还是页面仍在忙于加载？
- **是否令人愉快?** 交互是否顺畅而自然，没有滞后和卡顿？

性能指标 (Performance Metrics)

- **Load** - 当整个页面加载完成时（包括所有依赖资源，如样式表和图像）触发该事件
- **DOMContentLoaded** - HTML加载完成时（无需等待样式、图片和子框架加载完成）触发该事件

```
window.addEventListener('load', (event) => {  
  console.log('page is fully loaded');  
});
```

```
window.addEventListener('DOMContentLoaded', (event) => {  
  console.log('DOM fully loaded and parsed');  
});
```

是否发生？

问题一

当用户访问一个网站的时候，关心的第一个问题永远是“是否发生”——浏览器是否成功地把我的请求发送出去，而服务器是否已经知道并开始处理我的请求？

- **TTFB (Time to First Byte)**

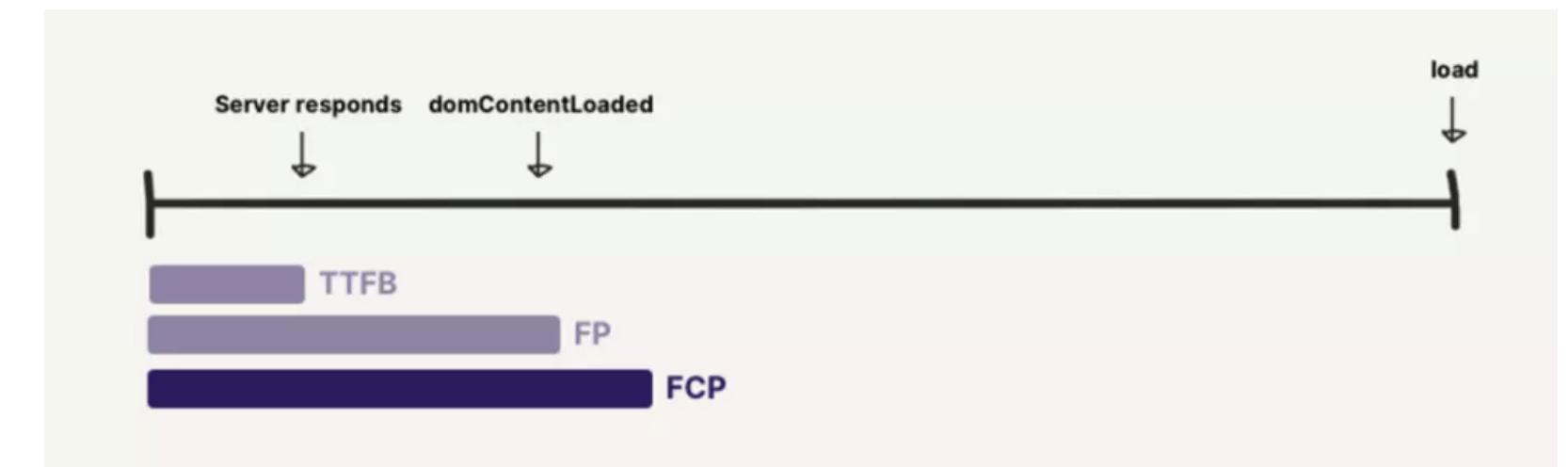
- 首字节到达的时间点

- **FP (First Paint)**

- 首次绘制，标记浏览器渲染任何在视觉上不同于导航前屏幕内容的时间点

- **FCP (First Contentful Paint)**

- 首次内容绘制，标记浏览器渲染来自DOM第一位内容的时间点，内容可能是文本、图像等元素



LCP (Largest Contentful Paint)

Core

最大内容绘制时间，计算从页面开始加载到用户与页面发生交互（点击，滚动）这段时间内，最大元素绘制的时间，该时间会随着页面渲染变化而变化，因为页面中的最大元素在渲染过程中可能会发生改变。

- **Load** - 当整个页面加载完成时（包括所有依赖资源，如样式表和图像）触发该事件。
- **DOMContentLoaded** - HTML加载完成时（无需等待样式、图片和子框架加载完成）触发该事件。

```
window.addEventListener('load', (event) => {  
  console.log('page is fully loaded');  
});
```

```
window.addEventListener('DOMContentLoaded', (event) => {  
  console.log('DOM fully loaded and parsed');  
});
```

Navigation

Hover on the bottom-left corner to see the navigation's controls panel, [learn more](#)

KEYBOARD SHORTCUTS

right / space

next animation or slide

left

previous animation or slide

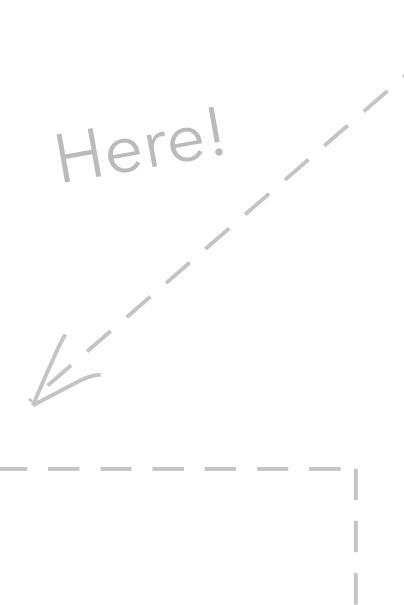
up

previous slide

down

next slide

Here!



Code

Use code snippets and get the highlighting directly!

```
interface User {  
  id: number  
  firstName: string  
  lastName: string  
  role: string  
}  
  
function updateUser(id: number, update: User) {  
  const user = getUser(id)  
  const newUser = { ... user, ... update}  
  saveUser(id, newUser)  
}
```



Components

You can use Vue components directly inside your slides.

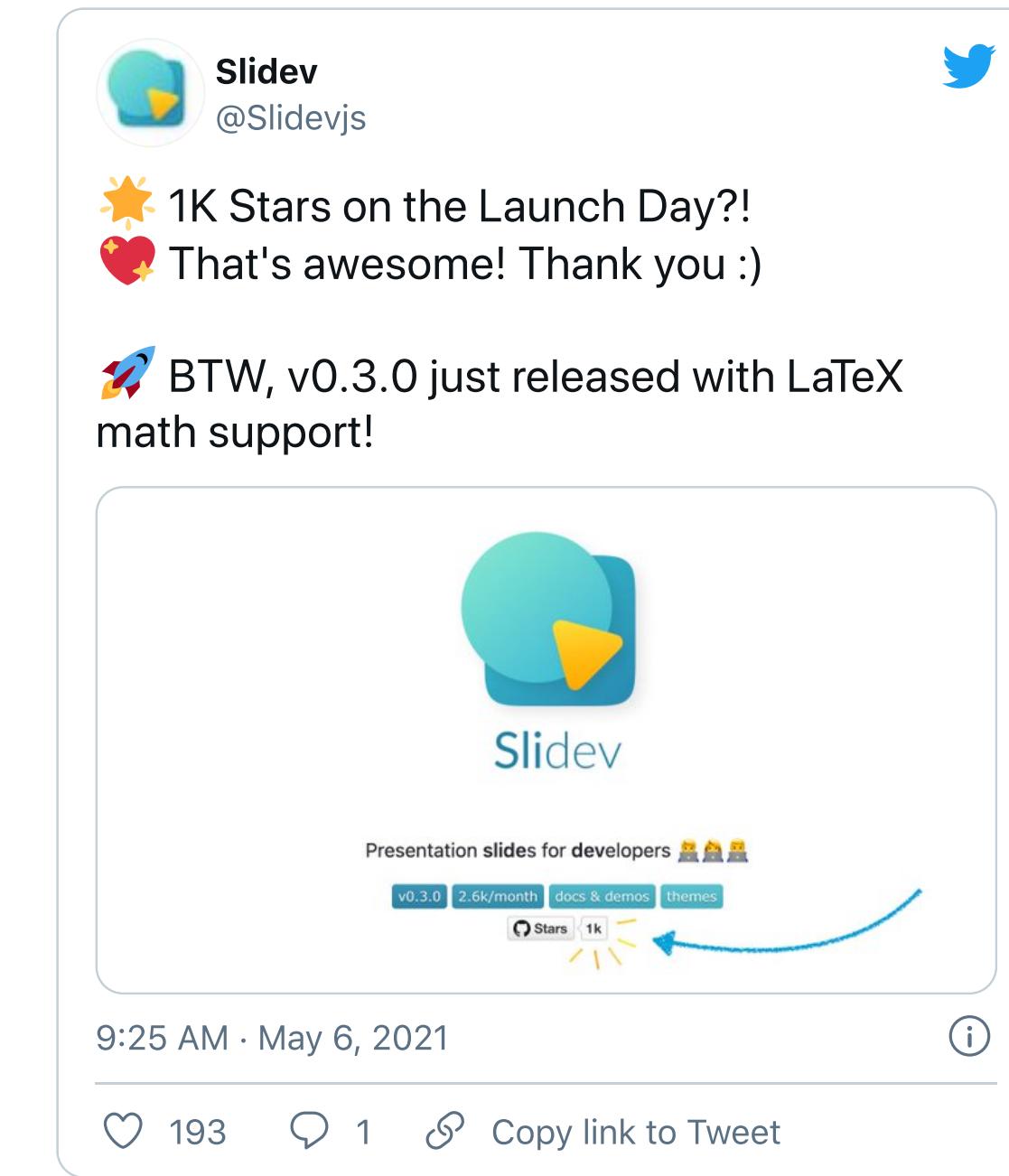
We have provided a few built-in components like `<Tweet/>` and `<Youtube/>` that you can use directly. And adding your custom components is also super easy.

```
<Counter :count="10" />
```

- 10 +

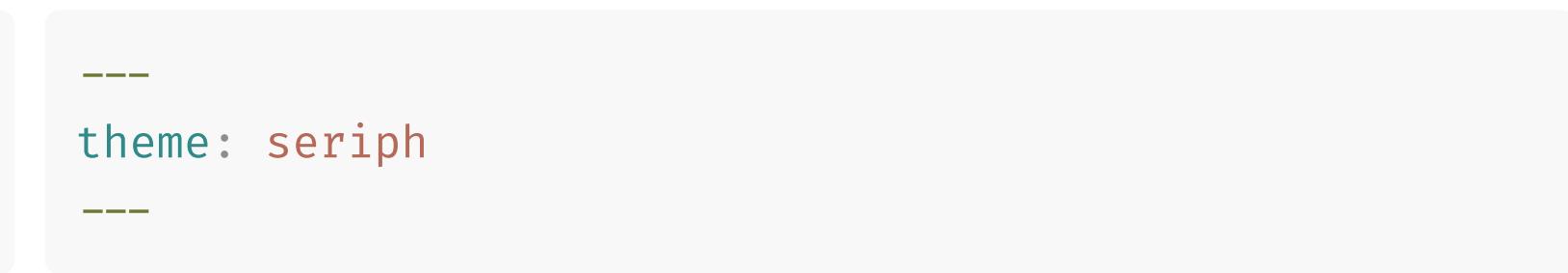
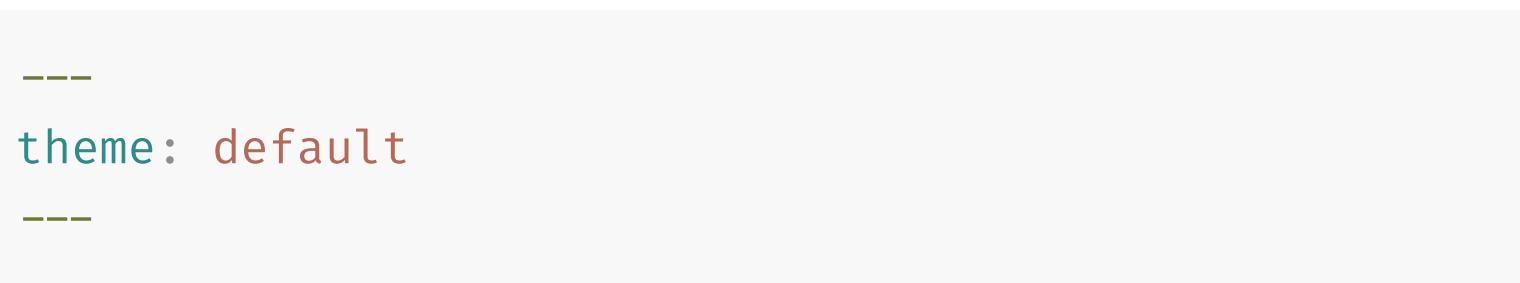
Check out the guides for more.

```
<Tweet id="1390115482657726468" />
```



Themes

Slidev comes with powerful theming support. Themes can provide styles, layouts, components, or even configurations for tools. Switching between themes by just **one edit** in your frontmatter:



Read more about [How to use a theme](#) and check out the [Awesome Themes Gallery](#).

Animations

Animations are powered by [@vueuse/motion](#).

```
<div
  v-motion
  :initial="{ x: -80 }"
  :enter="{ x: 0 }">
  Slides
</div>
```



LaTeX

LaTeX is supported out-of-box powered by KaTeX.

Inline $\sqrt{3x - 1} + (1 + x)^2$

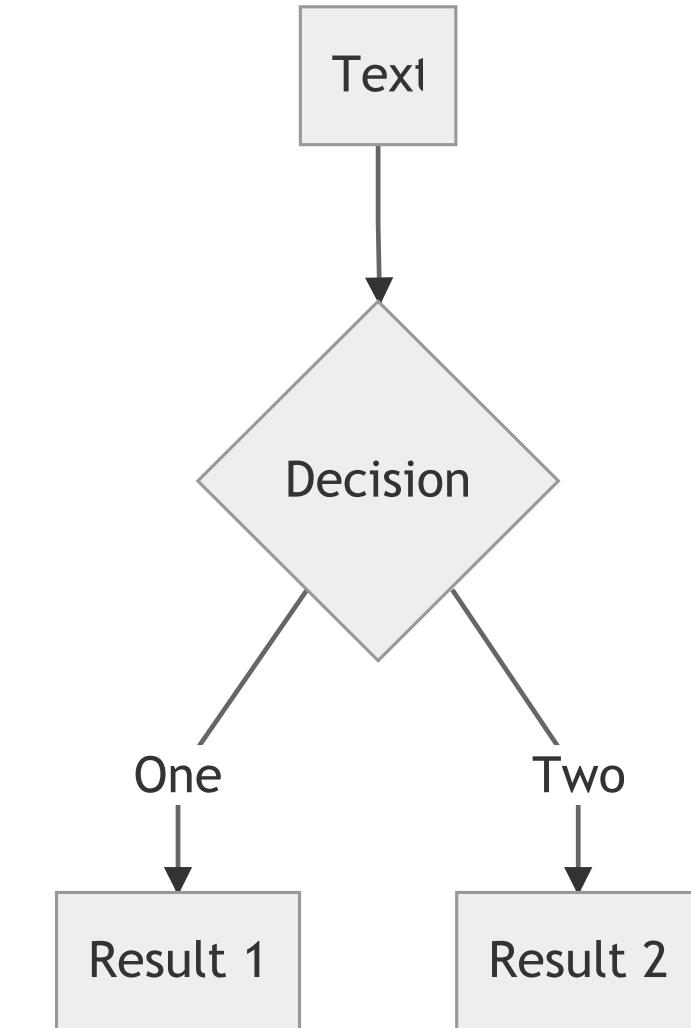
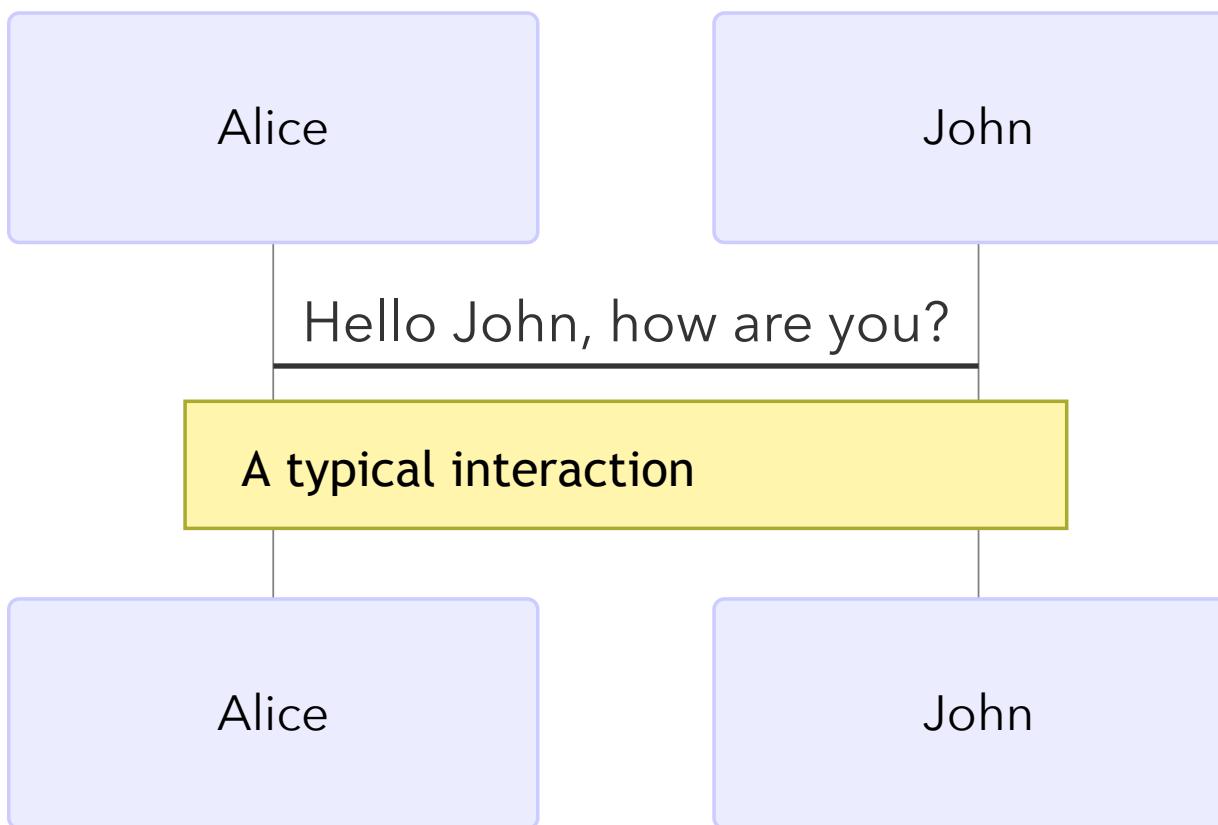
Block

$$\begin{aligned}\nabla \times \vec{\mathbf{B}} - \frac{1}{c} \frac{\partial \vec{\mathbf{E}}}{\partial t} &= \frac{4\pi}{c} \vec{\mathbf{j}} \nabla \cdot \vec{\mathbf{E}} &= 4\pi\rho \\ \nabla \times \vec{\mathbf{E}} + \frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} &= \vec{\mathbf{0}} \\ \nabla \cdot \vec{\mathbf{B}} &= 0\end{aligned}$$

Learn more

Diagrams

You can create diagrams / graphs from textual descriptions, directly in your Markdown.



[Learn More](#)

Learn More

[Documentations](#) / [GitHub Repo](#)
