INP
A new metric for interactivity
Adam Silverstein • Developer Relations Engineer @ Google
What is INP?

“INP is a metric that assesses a page's overall responsiveness to user interactions by observing the latency of all click, tap, and keyboard interactions that occur throughout the lifespan of a user's visit to a page. The final INP value is the longest interaction observed, ignoring outliers.”

https://web.dev/articles/inp • https://web.dev/articles/vitals
The importance of responsiveness

- Not about loading - 90% of a user's time on a page is spent after it loads
- Good responsiveness means a webpage reacts promptly to user actions.
- Poor vs. good responsiveness
Why replace FID?

- ~97% / 100% of mobile / desktop views already have good FID
- FID only measures the first interaction
INP reveals potential improvements

- Only 71% of mobile views have good INP
- INP measures all interactions
Measuring interactions

- Blocking tasks
- Input delay
- Processing time
- Render
- Presentation delay
- Compositing, GPU, and raster
First Input Delay (FID)

FID measures only the input delay
Only for the first interaction
Interaction to next paint (INP)

INP measures the entire presentation delay Across all interactions
INP parts
What is a good INP score?

- Good INP score = 200 milliseconds or less
- Threshold to measure is the 75th percentile of page loads
- Focus on mobile (desktop already passes)
Measuring INP

- PageSpeed Insights
- Google Search Console
- CrUX Dashboard
- GoogleChrome/web-vitals
- WebPageTest

Lab measurement - TBT + INP
- Using the web-vitals.js JavaScript library (includes attribution)
- DevTools Performance panel Interactions track
- Lighthouse timespan mode to record interactions

https://github.com/GoogleChrome/web-vitals
What causes slow INP?

➔ Too much work happening on the page!
  ◆ Too much JavaScript
  ◆ Too large a DOM
  ◆ Too complex CSS selectors
  ◆ Not leveraging the web platform

➔ Poorly written code

➔ Code that does not yield to the main thread
Fixing slow INP

→ Identify issues from field data, identify the interaction that is causing issues
→ Use lab testing to reproduce issue
→ Address specific issue
→ Rinse and repeat

→ Prefetch/prerender or speculative prefetching
→ Reduce, reduce, reduce
→ Simplify
Thank you
bit.ly/inp-new

Adam Silverstein • Developer Relations Engineer @ Google • @roundearth