



# Come e perché (\$) migliorare le prestazioni web

Consigli pratici per il 2023

Dev Romagna, 22 marzo 2023







**1 secondo**  
di tempo di  
caricamento

**-14%**

utenti che  
lasciano il sito  
all'atterraggio  
(frequenza di rimbalzo)

**+13%**

utenti che raggiungono  
gli obiettivi del sito  
(tasso di conversione)



- Comprensione di base di cosa sono le prestazioni web
- Come le prestazioni web impattano sul business
- Suggerimenti per il miglioramento delle prestazioni web



# Andrea Verlicchi

- Tonnellate di siti
- Sviluppo front-end
- Consulente per le prestazioni web





Google

Google Hackathon, London



1. Prestazioni web, Search Engine Optimisation, impatto business
2. Misurazione web performance
- Gioco a premi!** 🎉
3. Suggerimenti per il 2023

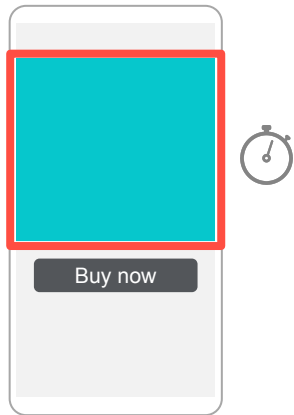
# Prestazioni web, SEO, e impatto sul business



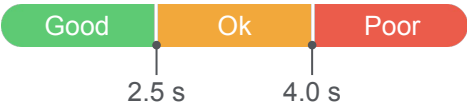


# Google Core Web Vitals

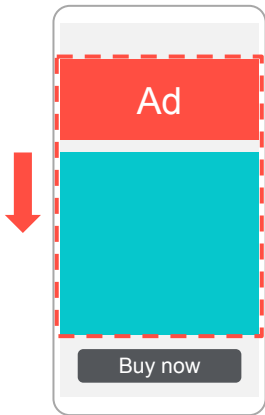
## Caricamento



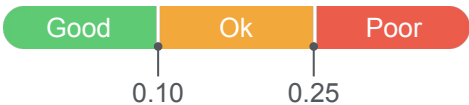
Largest Contentful Paint (LCP)



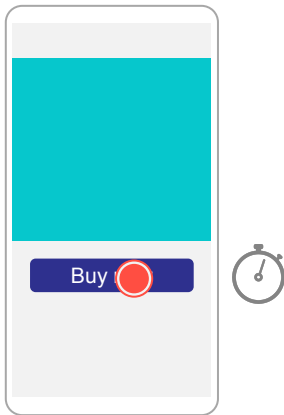
## Stabilità Visiva



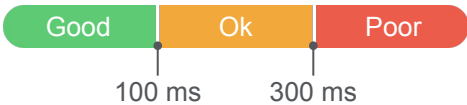
Cumulative Layout Shift (CLS)



## Interattività



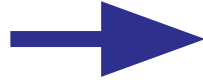
First Input Delay (FID)



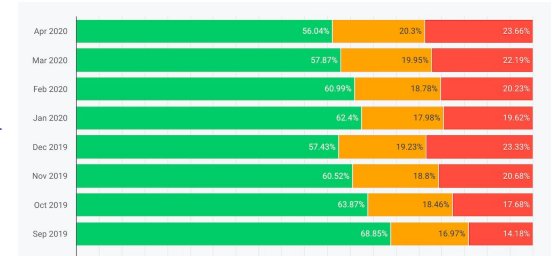
# Come fa Google a saperlo?



Google Chrome

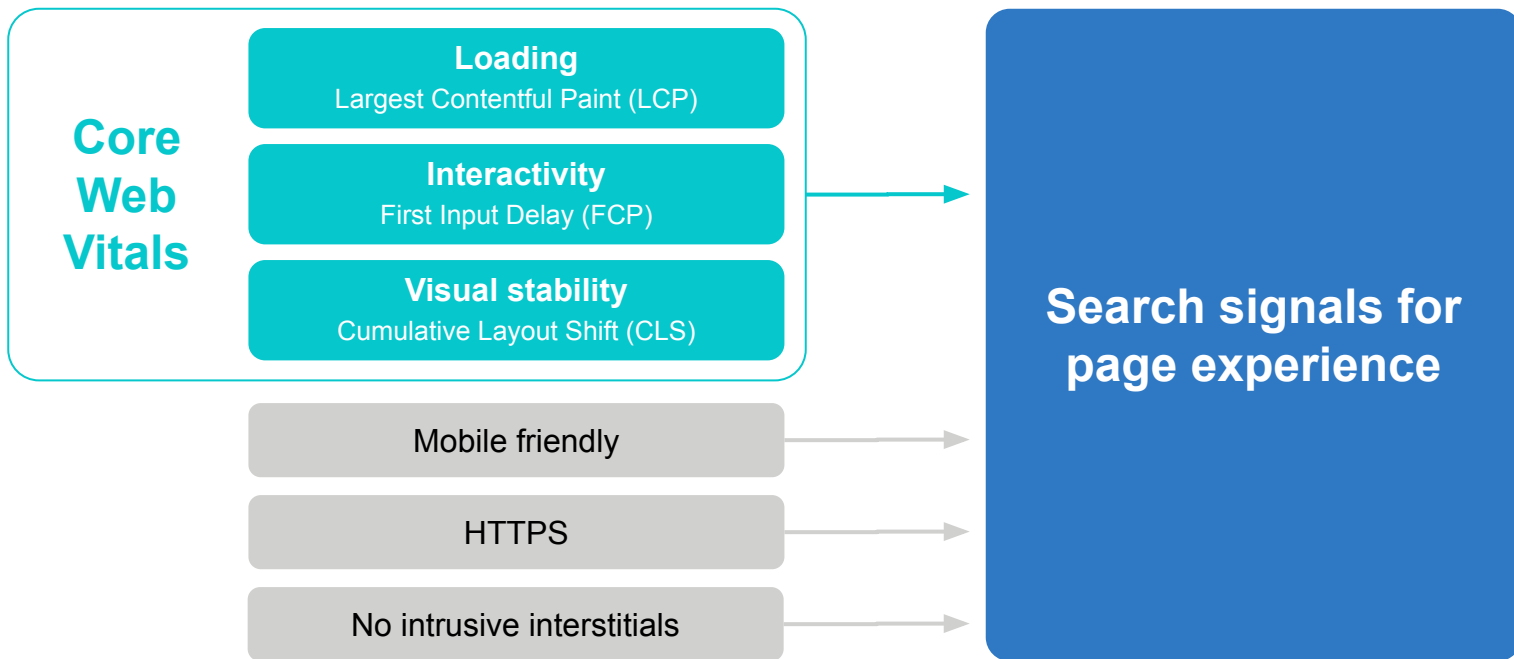


Chrome User Experience  
(CrUX)



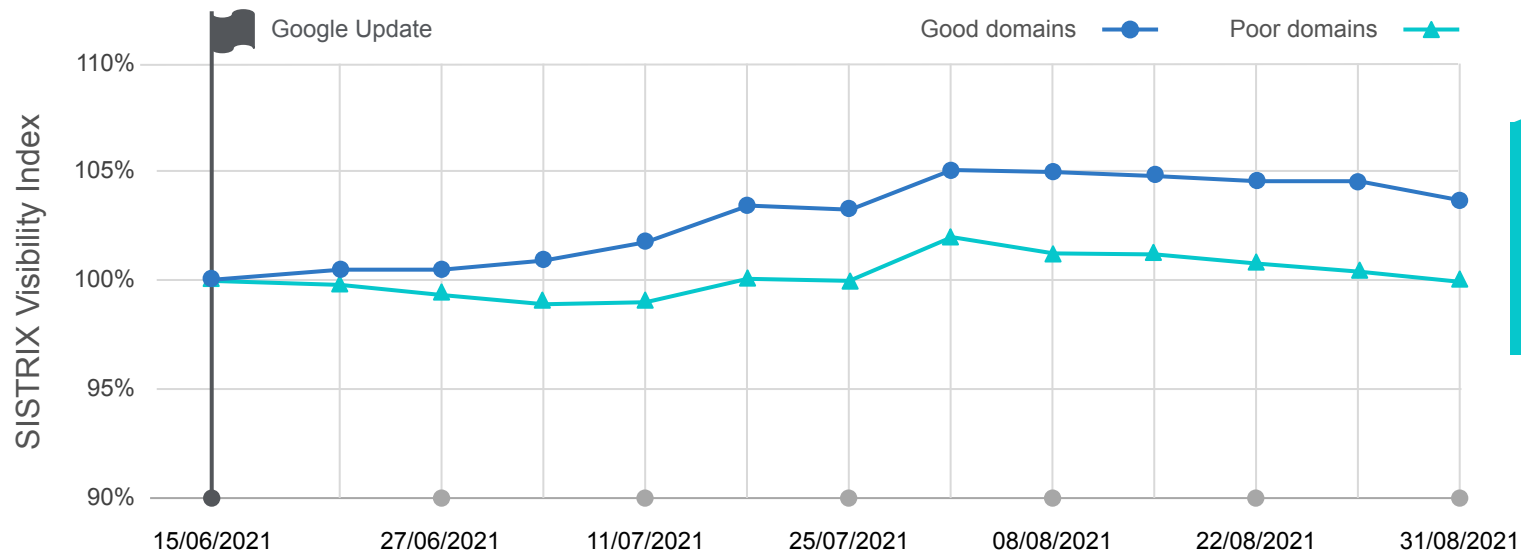
CrUX Report

# I Core Web Vitals impattano il posizionamento in Google Search





# Quanto impattano i Core Web Vitals nella classificazione



**4%**  
Increase in  
ranking

$$\text{Successo} = \text{Traffico} \times \text{Conversioni}$$

Basse conversioni +  
Alto costo del traffico

**Prestazioni scarse**



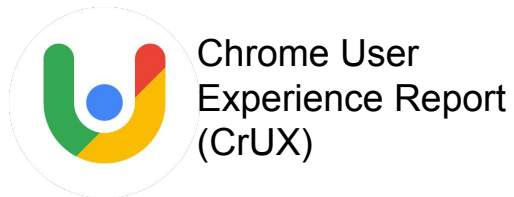
**Buone prestazioni**

Alte conversioni +  
Basso costo del  
traffico

# Misurare le prestazioni web

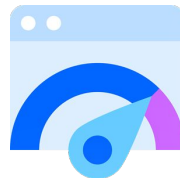


## Sul sito - Real User Monitoring



cognizant **web vitals monitoring**

## In laboratorio - Synthetic



Pagespeed  
insights



Lighthouse



## Sul sito - Real User Monitoring

- Su tutte le pagine, mentre gli utenti navigano
- Utenti reali, dispositivi reali, connessioni, posizioni reali
- Dati in una dashboard

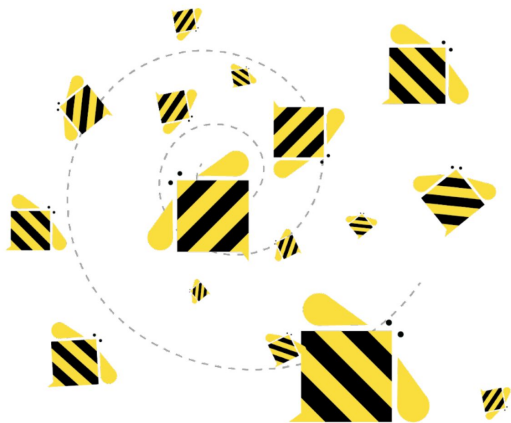
**Trovare i problemi**

## In laboratorio - Synthetic

- Un determinato URL
- Emulazione, singola località
- Risultati immediati
- Dettagli! Fotogrammi, video, waterfall di rete, CPU, etc.

**Investigarne le cause,  
testare soluzioni**

## Sul sito - Real User Monitoring



Il quadro generale

## In laboratorio - Synthetic



Lo zoom

# Gioco a premi!



Here is when I opened Kahoot :)

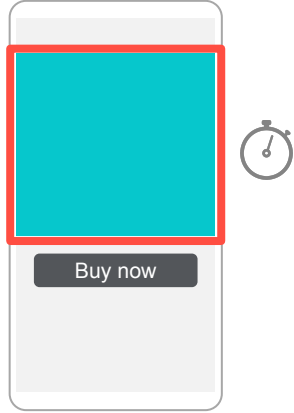


# Suggerimenti per il miglioramento nel 2023

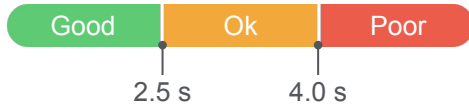
- Maggiore impatto nel mondo reale
- Pertinenti e applicabili a tutti i siti
- Realistici da implementare

# Google Core Web Vitals

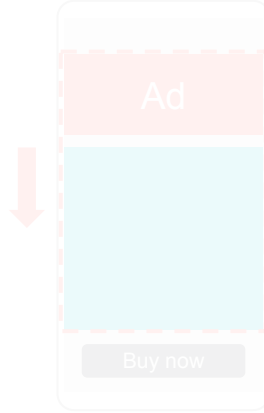
## Loading



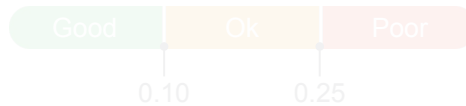
Largest Contentful Paint (LCP)



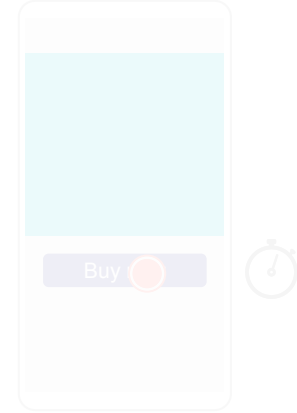
## Visual Stability



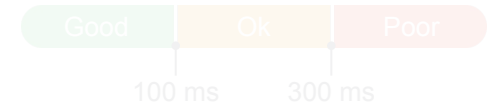
Cumulative Layout Shift (CLS)




## Interactivity



First Input Delay (FID)



Assicurati che la risorsa  
LCP sia rilevabile  dal  
sorgente HTML

**72%** delle pagine, su smartphone,  
elemento LCP = un'immagine


**39%** di queste immagini  
non è rilevabile dalla sorgente HTML



- Carica l'immagine usando un elemento `<img>` con l'attributo `src` or `srcset`
- Preferisci il rendering lato server (SSR) rispetto al rendering lato client (CSR)
- Se la tua immagine deve essere referenziata da un file CSS o JS esterno, includila nel sorgente HTML tramite tag `<link rel="preload">` tag.

Assicuratevi che la ▲  
chiamata alla risorsa LCP  
abbia la massima priorità

- Aggiungi `fetchpriority="high"` al tag `<img>` della tua immagine LCP
- Non impostare mai `loading="lazy"` sul tag `<img>` della tua immagine LCP
- Rimandare le risorse non critiche, in ogni caso in cui è possibile farlo

Utilizza una CDN   
per ottimizzare il tempo di  
risposta del server (TTFB)  
di HTML e risorse

**Servire** i tuoi contenuti dal punto più geograficamente vicino ai tuoi utenti.

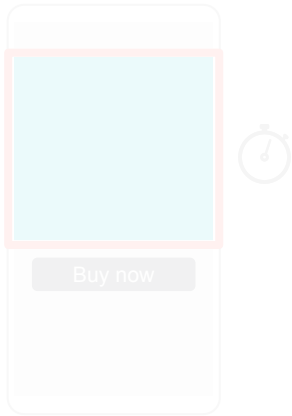
**Memorizzare** nella cache i contenuti in modo che, quando richiesti nuovamente, possano essere serviti rapidamente.



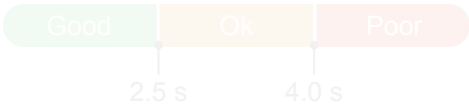
- Aumenta il tempo per cui i contenuti vengono memorizzati nella cache.
- Memorizza nella cache il contenuto a tempo indeterminato e svuota la cache quando fai gli aggiornamenti.
- Sposta la logica dinamica, dal server di origine (origin) all'edge server

# Google Core Web Vitals

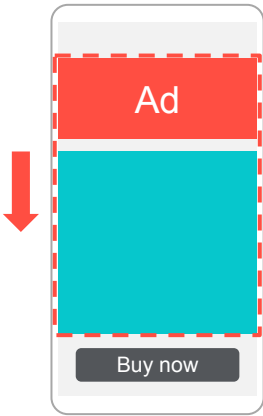
## Loading



Largest Contentful Paint (LCP)



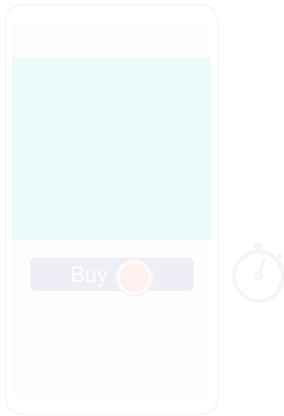
## Visual Stability



Cumulative Layout Shift (CLS)




## Interactivity



First Input Delay (FID)




Imposta dimensioni  
esplicite  su ogni  
contenuto caricato dalla  
pagina

**0px** altezza predefinita iniziale  
delle immagini non dimensionate

**72%** delle pagine ha  
almeno un'immagine senza dimensioni

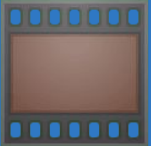
- Setta in modo esplicito gli attributi `width` e `height` (or proprietà CSS equivalenti) nelle immagini
- Usa la proprietà CSS `aspect-ratio` per riservare spazio per contenuti caricati lazy (annunci, video incorporati, ecc.)
- Se aspetto sconosciuto, usa `min-height`



Assicurati che le pagine  
siano idonee per la cache  
back/forward   
(bfcache)

**35%** delle pagine  
non sono idonee per bfcache

- Verifica se le tue pagine sono idonee utilizzando lo strumento *bfcache tester* (in developer tools)
- Lavora sui motivi per cui non lo sono 😊

Evita animazioni/transizioni  
 che utilizzano proprietà  
CSS che inducono il  
ricalcolo del layout

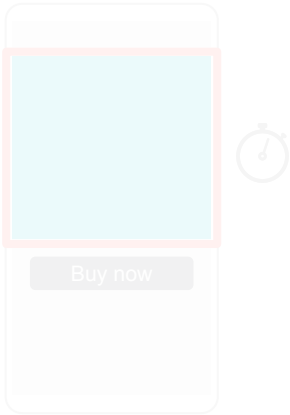
**15%** in meno di probabilità di avere un buon CLS se animi qualsiasi proprietà CSS che possa influenzare il layout

Gli elementi posizionati con **absolute**, di cui si animano le proprietà top o left, causano layout shift 🤯

- Mai animare o transizionare le proprietà CSS che richiedono ai browser di aggiornare il layout della pagina
- Invece di animare le proprietà top o left, animare `transform:translateX()` o `transform:translateY()`

# Google Core Web Vitals

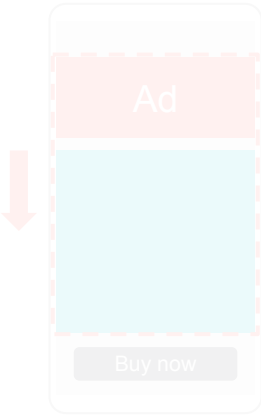
## Loading



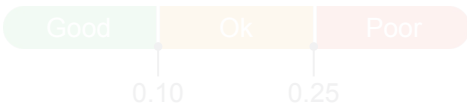
Largest Contentful Paint (LCP)



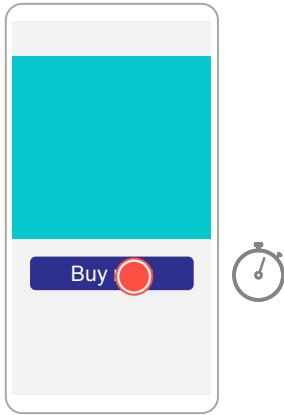
## Visual Stability



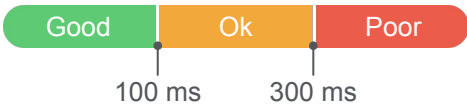
Cumulative Layout Shift (CLS)



## Interactivity

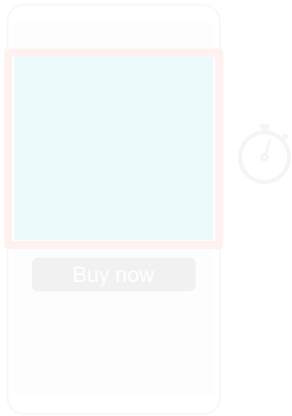


First Input Delay (FID)



# Google Core Web Vitals

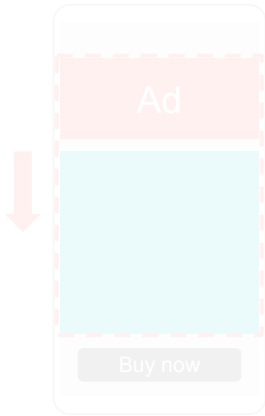
## Loading



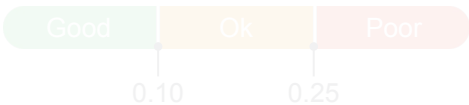
Largest Contentful Paint (LCP)



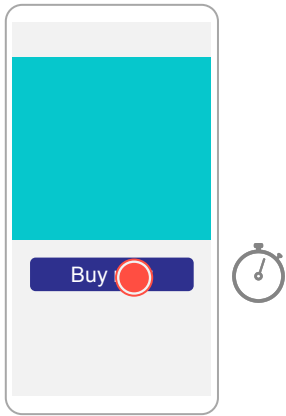
## Visual Stability



Cumulative Layout Shift (CLS)



## Interactivity



Interaction to Next Paint (INP)





Evita o spezzetta  
i long task




**Task** = rendering e layouting, oltre ad analisi, compilazione, esecuzione di script.

**50 ms** di blocco nel thread principale  
soglia per cui un task è considerato “lungo”.

**19** è il numero mediano  
di long task su mobile

- Suddividi le attività lunghe (long task) in attività più piccole, trasferendo spesso l'esecuzione al thread principale
- Prendi in considerazione l'utilizzo di API come `isInputPending` e `Scheduler`

Evita codice JavaScript  
 non necessario

**460** kb/pagina

mediana del peso del codice JS servito

**Questo** Javascript crea  
un ambiente in cui le le attività competono  
per l'attenzione del thread principale

- Utilizza lo strumento Coverage in DevTools per trovare il codice inutilizzato
- Se c'è codice inutilizzato perché verrà utilizzato in seguito, spostalo in un pacchetto separato (code splitting)
- Utilizzi un gestore di tag?  
Controlla periodicamente i tuoi tag.

Evita i grandi   
aggiornamenti di  
rendering

**Javascript** non è l'unica cosa che può influenzare la reattività del tuo sito web.

**Rendering** può essere costoso e può interferire con la capacità del tuo sito web di rispondere agli input dell'utente.



- Evita di usare `requestAnimationFrame()` per eseguire lavoro non visivo.
- Mantieni piccole le dimensioni del tuo DOM.
- Usa il contenimento CSS (proprietà `contain`)

# Per concludere













# Feedback, please?



Google Form