

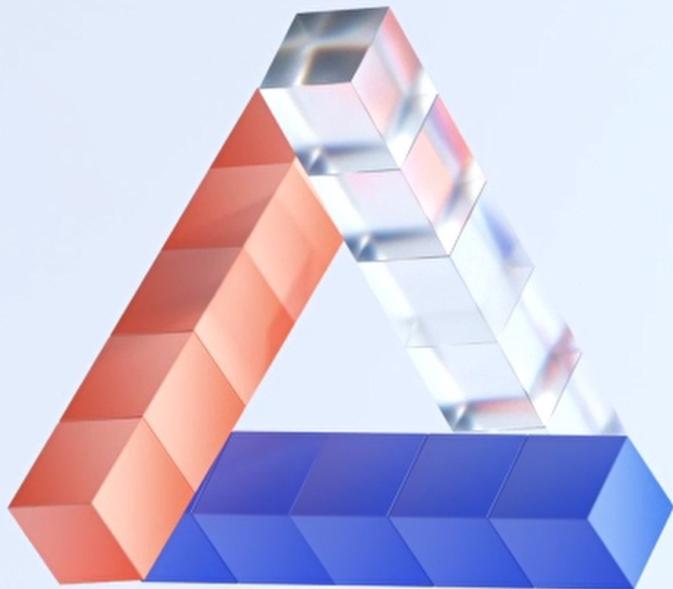


Web Frontend Performance Optimization

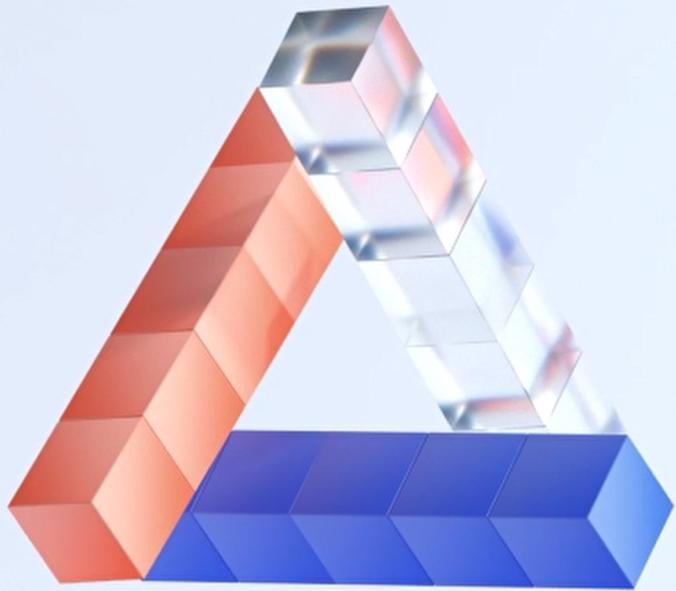
Jinsik Rho & Jeongmin Shin



Agenda



1. Introduction
2. Why should Web Performance be optimized?
3. How to measure Web Performance?
 - Tools & Audit
4. What is ifland studio?
 - Web Performance Optimization target
5. Case : How to optimize web performance of ifland studio
6. Lessons learned & tips
7. Conclusion



1. Introduction

이번 세미나에서 **알아 가셨으면 하는 것**

웹 프론트엔드 성능 최적화 하는 이유

for 개발자, 디자이너, PM, 세미나 참석자

웹 프론트엔드 성능 최적화 방법

for FE개발자

Introduction

Model: GPT-4



웹 프론트엔드 개발자의 역량이란?



웹 프론트엔드 개발자는 사용자의 웹 브라우징 경험을 개선하기 위해 사용자 인터페이스를 디자인하고 개발하는 역할을 합니다. 웹 프론트엔드 개발자가 되기 위해 다음과 같은 기술 및 역량이 필요합니다:

- HTML/CSS:** 웹 프론트엔드 개발자는 HTML과 CSS를 깊게 이해하고 있어야 합니다. HTML은 웹페이지의 구조를 만드는데 사용되며, CSS는 웹페이지를 스타일링하고 레이아웃을 구성하는데 사용됩니다.
- JavaScript:** JavaScript는 웹페이지를 동적으로 만들고 사용자 인터랙션을 처리하는 데 사용됩니다. JavaScript 프레임워크나 라이브러리에 대한 이해도 중요하며, 이를 중 일부는 React, Angular, Vue.js 등입니다.
- 반응형 및 모바일 디자인:** 사용자가 다양한 장치를 사용해 웹사이트를 방문하기 때문에, 웹사이트는 다양한 화면 크기에 대응할 수 있어야 합니다. 프론트엔드 개발자는 반응형 웹 디자인 및 모바일-퍼스트 디자인에 익숙해야 합니다.
- 퍼포먼스 최적화:** 빠른 로딩 시간과 원활한 실행은 좋은 사용자 경험을 만드는 핵심적인 요소입니다. 프론트엔드 개발자는 성능을 최적화하는 방법을 알아야 합니다.
- 버전 관리/Git:** 버전 관리 시스템은 팀에서 코드 변경을 추적하고 이전 버전으로 되돌리는 데 사용됩니다. Git은 가장 널리 사용되는 버전 관리 시스템 중 하나입니다.
- 테스팅/디버깅:** 프론트엔드 개발자는 코드의 문제점을 찾고 수정하기 위한 디버깅 기술을 알아야 하며, 단위 테스트, 통합 테스트 등을 통해 코드의 안정성을 보장할 수 있어야 합니다.
- 웹접근성 / 사용성:** 웹사이트는 모든 사용자, 예를 들어, 장애를 가진 사용자들에게도 접근 가능하도록 디자인되어야 합니다. 웹 접근성 가이드라인에 익숙해져야 합니다.
- 크로스브라우저 호환성:** 다양한 웹 브라우저가 웹사이트를 올바르게 해석하고 표시할 수 있도록, 크로스 브라우저 호환성을 보장하는 것도 중요합니다.
- APIs / AJAX:** 웹 프론트엔드 개발자는 백엔드 서비스와 통신하기 위해 AJAX와 APIs를 사용해야 합니다.
- CSS 및 JavaScript 프레임워크:** 프레임워크는 기본적인 웹 디자인 요소와 기능을 쉽게 구현할 수 있도록 돋습니다. Bootstrap이나 Material-UI와 같은 CSS 프레임워크, React나 Vue.js 와 같은 JavaScript 프레임워크에 대한 이해가 중요합니다.

이외에도 소프트 스킬(협업 능력, 커뮤니케이션, 문제 해결 능력 등)이 중요합니다. 프론트엔드 개발자는 디자이너, 백엔드 개발자, 프로젝트 매니저 등 다양한 이해관계자와 협업하게 됩니다.



웹 프론트엔드 개발자의 가장 중요한 5가지 역량이 뭘까?



웹 프론트엔드 개발자의 역량은 매우 다양하지만, 가장 중요한 5가지 역량은 아래와 같을 수 있습니다:

- HTML/CSS/JavaScript 기술:** 이 세 가지 기술은 웹 프론트엔드 개발의 핵심입니다. HTML과 CSS는 웹 페이지의 구조와 스타일을 담당하며, JavaScript는 페이지에 동적인 요소와 인터랙션을 추가합니다.
- 반응형 및 모바일 디자인:** 오늘날 사용자들은 다양한 장치에서 웹사이트를 방문합니다. 따라서 프론트엔드 개발자는 모든 장치에서 웹사이트가 잘 작동하도록 반응형 웹 디자인과 모바일-퍼스트 디자인에 대한 이해가 필요합니다.
- 프레임워크 및 라이브러리 이해:** 프론트엔드 개발에는 다양한 자바스크립트 프레임워크와 라이브러리가 쓰입니다. React, Vue.js, Angular 등의 프레임워크와 라이브러리에 대한 이해는 중요한 역량입니다.
- 테스팅/디버깅:** 웹사이트가 예상대로 작동하고 있고, 잠재적인 문제를 신속하게 해결할 수 있어야 합니다. 이를 위해 단위 테스트, 통합 테스트 등의 테스팅 방법과 디버깅 기술에 대한 이해가 중요합니다.
- 퍼포먼스 최적화:** 사용자 경험을 개선하기 위해 웹 페이지의 로딩 시간을 최소화하고 성능을 최적화하는 능력이 필요합니다.

이러한 기술적 역량 외에도 문제 해결 능력, 협업 능력, 커뮤니케이션 능력과 같은 소프트 스킬도 매우 중요합니다. 웹 프론트엔드 개발자는 종종 다른 개발자, 디자이너, 프로젝트 매니저 등과 협업해야 하므로, 팀에서 원활하게 작업할 수 있는 능력이 필요합니다.

Introduction

서비스 관점에서도 웹 성능 최적화는 중요합니다.



As page load time goes from:

1s to 3s the probability of bounce **increases 32%**

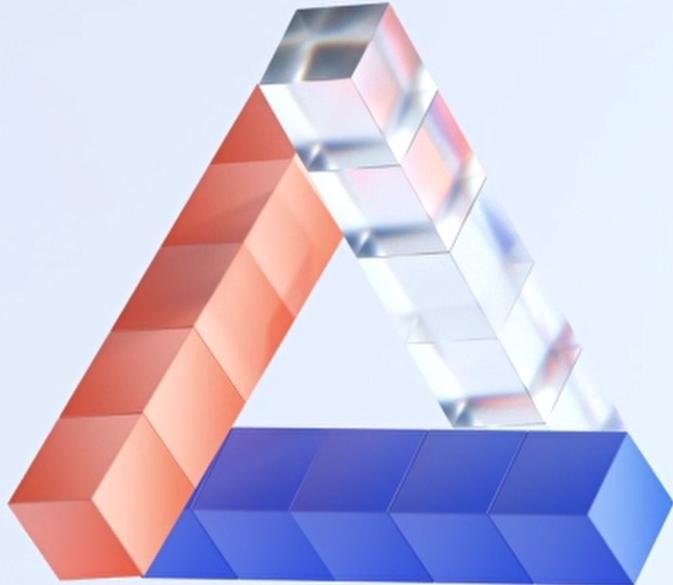
1s to 5s the probability of bounce **increases 90%**

1s to 6s the probability of bounce **increases 106%**

1s to 10s the probability of bounce **increases 123%**

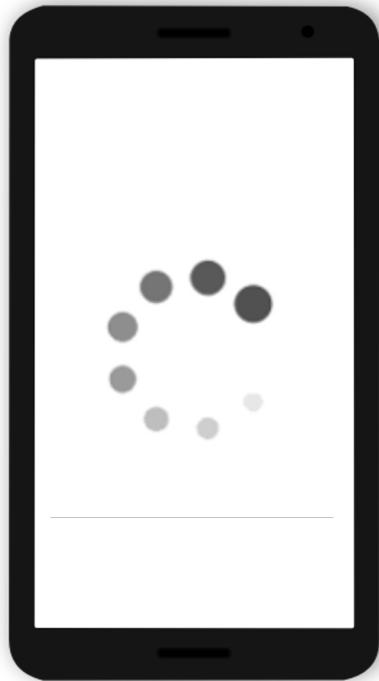
Think with Google, 2017(Google 모바일 부문 제품 리드 Daniel An)

백엔드 개발자, PM, 디자이너도 웹 성능 최적화를 알아야 합니다.

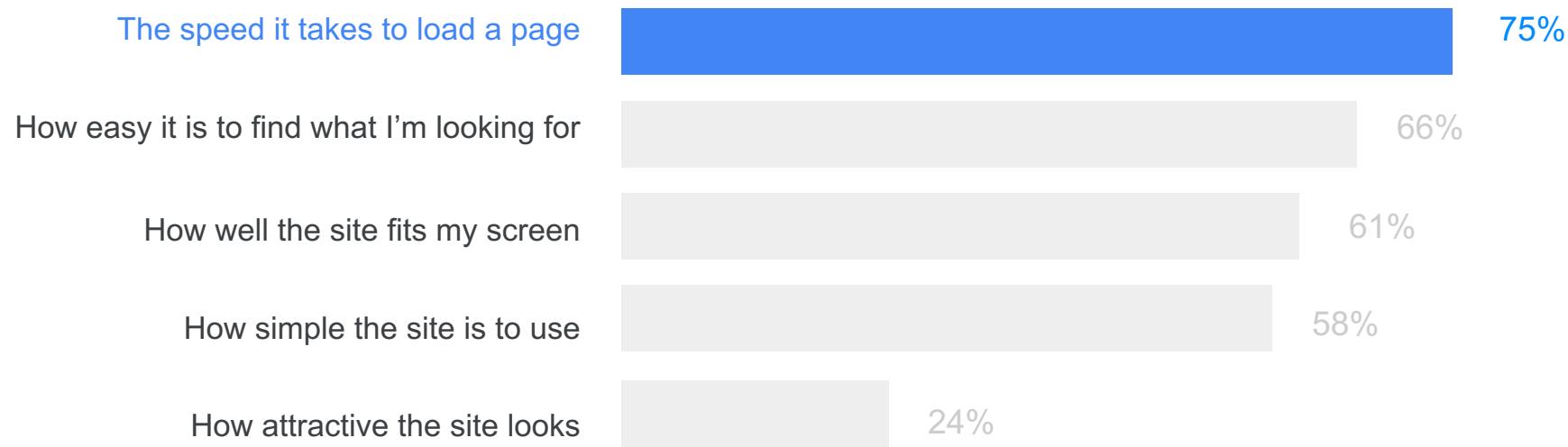


2. Why should Web Performance be optimized?

How long can you wait?



The impact on user experience



Source: SPEED MATTERS: Designing for Mobile Performance, Awwwards

Milliseconds makes Millions

Site speed improvement of just 0.1s
leads to an [increase in conversion
rates & average order value](#)

Google/55/Deloitte, [Speed Impact study](#), EMEA and US, Oct-Nov 2019, n=37 brand sites analysed
hourly over a 30 day period totalling 30.5m sessions.



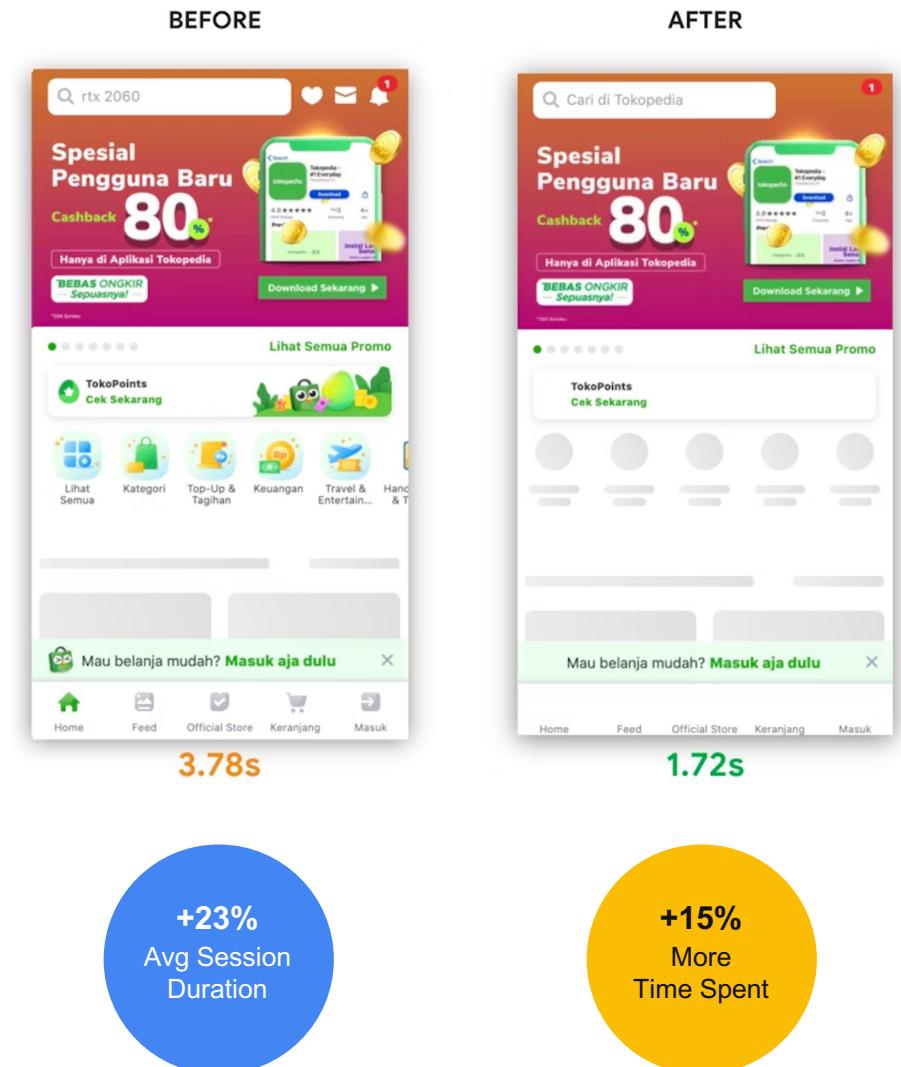
Web performance affects business impact

“At Tokopedia, our web platform is an integral part of our ecosystem. To democratize commerce through technology, we strive to give our customers and merchants a highly performant, comprehensive, stable, and innovative experience.

- Herman Widjaja (SVP & CTO, Tokopedia)

Tokopedia @ I/O 2021

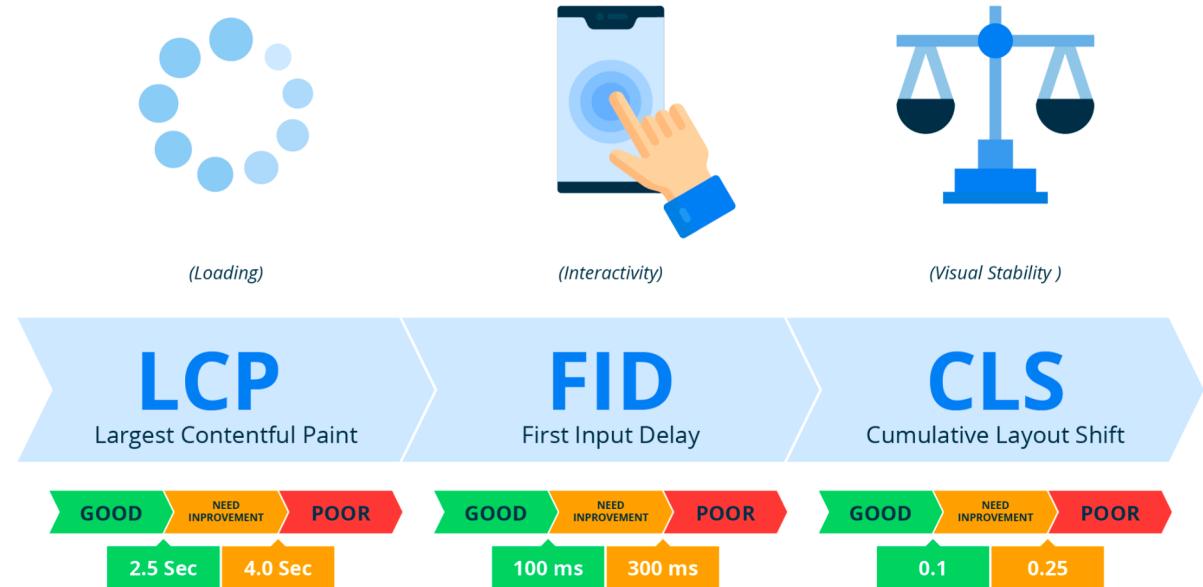
<https://web.dev/vitals-business-impact/#tokopedia>



How Google measure web performance?

- Largest Contentful Paint (LCP)
for measuring the loading speed
- First Input Delay (FID)
for scoring interactivity
- Cumulative Layout Shift (CLS)
for calculating the visual stability

Core Web Vitals



INP - what next? INP will replace FID

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/inp-cvv/>

gShoe product Q&A:

- What is gShoe?
- What technology does gShoe use?
- How much does gShoe cost?

Poor responsiveness

gShoe product Q&A:

- What is gShoe?
- What technology does gShoe use?
- How much does gShoe cost?

Good responsiveness

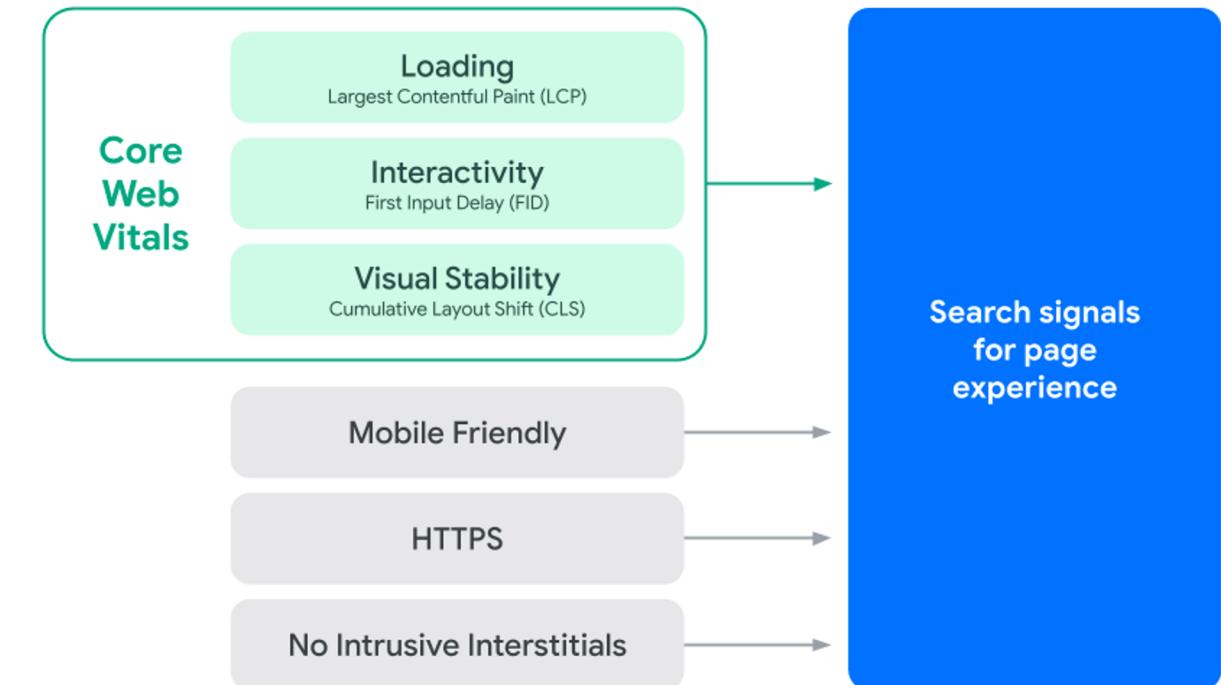
Mobile website experience affects All marketing channels

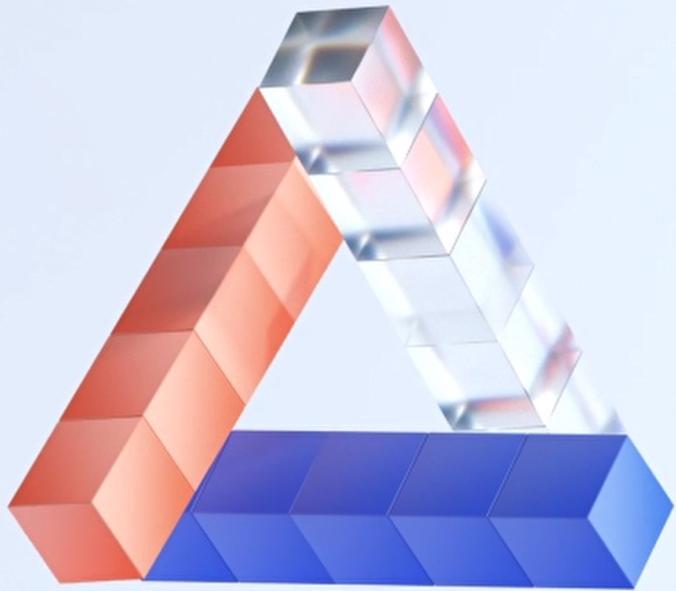
CHROME ([blog post here](#))

Chrome announced "Core Web Vitals," a set of essential metrics that measure quality of user experience, which correlate to loading, interactivity, and visual stability

GOOGLE SEARCH ([blog post here](#))

Search announced an evolution of the page experience factors considered for ranking content. It combines existing signals (such as mobile-friendly, HTTPS) with new signals - the Core Web Vitals.





3. How to measure Web Performance?

Core Web Vitals

Now in your favorite developer tools

	LCP	FID	CLS
 PageSpeed Insights	✓	✓	✓
 Chrome UX Report Brand new API, BigQuery and Dashboard	✓	✓	✓
 Search Console	✓	✓	✓
 Chrome DevTools	✓	TBT	✓
 Lighthouse	✓	TBT	✓
 Web Vitals Extension	✓	✓	✓

LCP = Largest Contentful Paint, FID = First Input Delay, CLS = Cumulative Layout Shift, TBT = Total Blocking Time

Measuring Real User Monitoring (RUM) tools

Google Tools



Search
Console



PageSpeed
insights



web-vitals
JS



Chrome User
Experience Report

Third party Tools



New Relic.



Calibre



SpeedCurve



Blue Triangle



Sentry

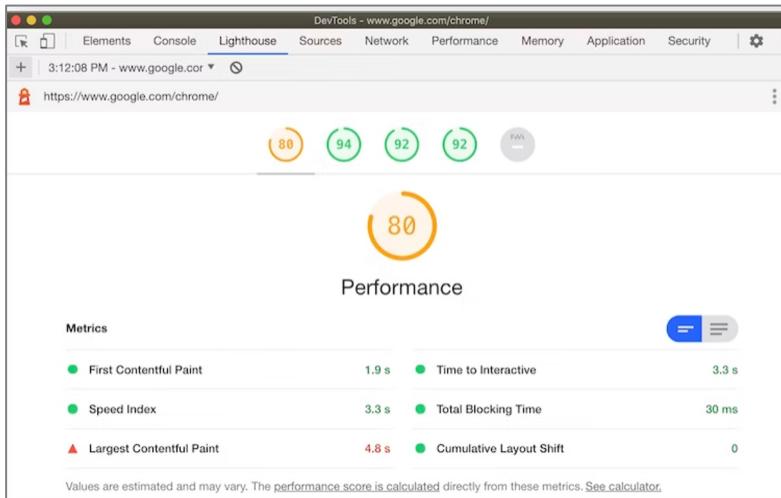


RAYGUN

Page speed audit reports

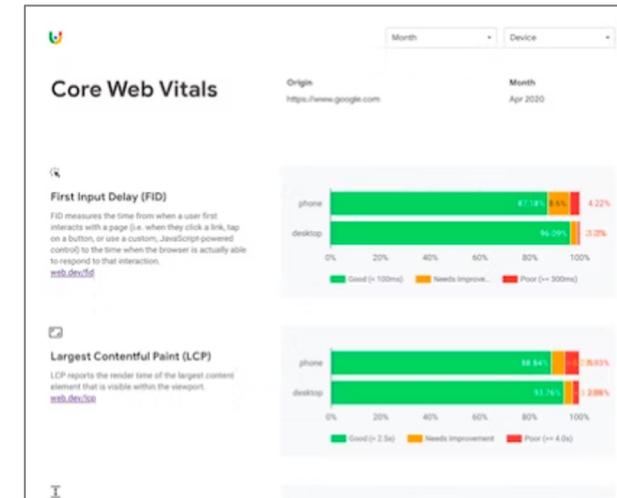
Lighthouse

Lighthouse is a lab tool that provides specific opportunities for improving page performance.



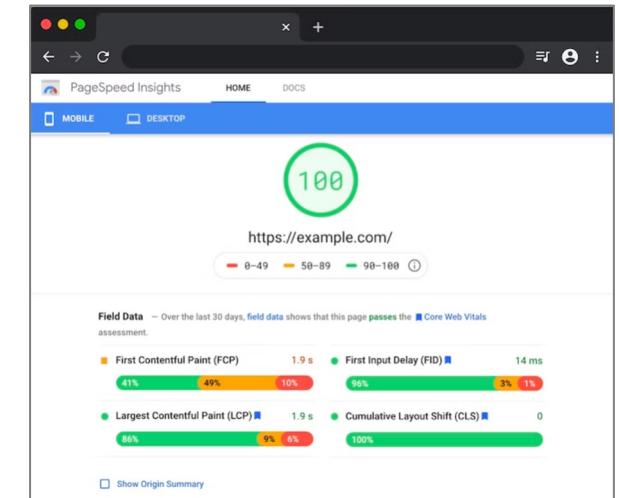
Chrome User Experience Report

CrUX is a BigQuery dataset of field data gathered from a segment of real Google Chrome users from millions of websites.



PageSpeed Insights (PSI)

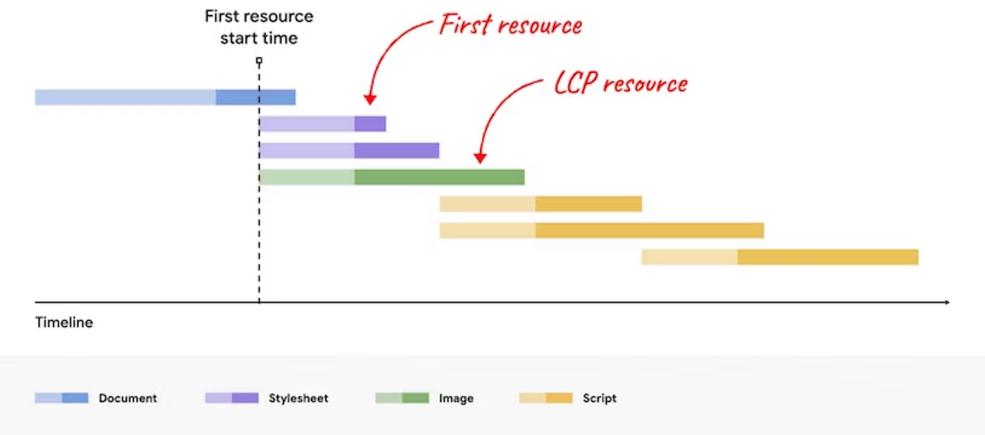
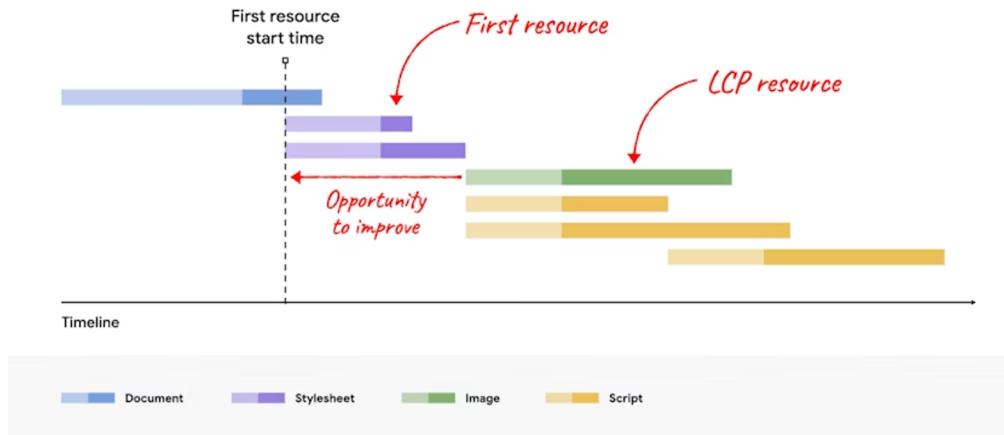
PSI is a tool that reports field data from CrUX and field lab from Lighthouse for a given page.



LCP optimization

- Ensure the LCP resource is discoverable from the HTML source
- Ensure the LCP resource is prioritized
- Use a CDN to optimize document and resource TTFB

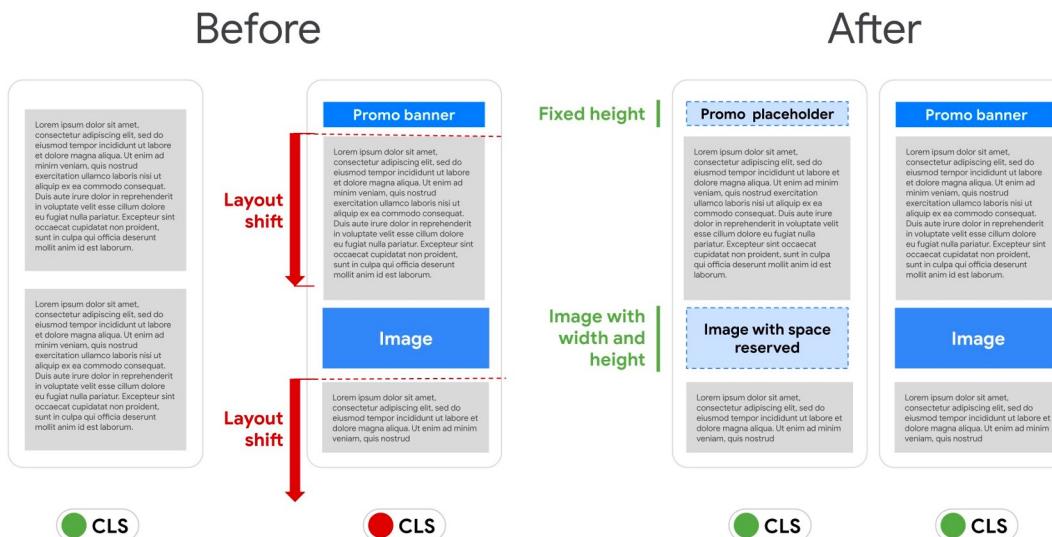
- Load the image using an `` element with the `src` or `srcset` attribute.
- Prefer **server-side rendering (SSR)** over client-side rendering (CSR)
- If your image needs to be referenced from an external CSS or JS file, use a `<link rel="preload">` tag.
- Add `fetchpriority="high"` to the `` tag of your LCP image.
- Never set `loading="lazy"` on the `` tag of your LCP image. Defer non-critical resources when possible.



CLS Optimization

- Set explicit sizes on any content loaded from the page
- Avoid animations/transitions that use layout-inducing CSS
- Be eligible for bfcache

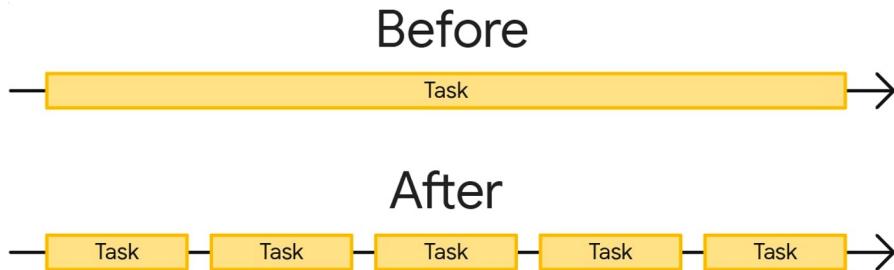
- The most straightforward way to fix layout shifts is to explicitly set **width** and **height** attributes (or equivalent CSS properties).
- The [**aspect-ratio**](#) property can help
- Setting a sensible **min-height** is almost always better than allowing the browser to use the default height

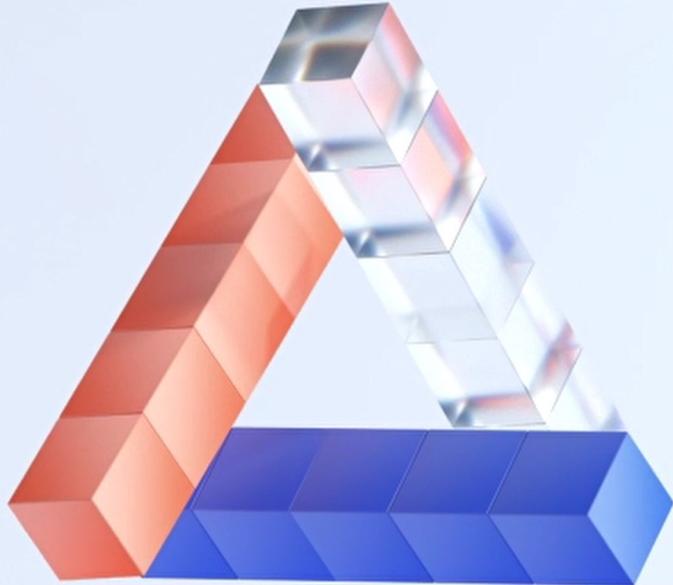


@addysosmani

FID Optimization

- Avoid or break up long tasks
- Avoid unnecessary JavaScript
- Avoid large rendering updates
 - breaking up long tasks into smaller ones
 - yielding to the main thread
 - Use the coverage tool in Chrome DevTools
 - code splitting
 - Be sure to periodically check your tags to make sure they are optimized
 - Avoid using requestAnimationFrame() for doing any non-visual work.
 - Keep your DOM size small





4. What is ifland studio (Web Performance Optimization Target)

What is ifland?

- **ifland**
 - Social Metaverse Service (공동 공간 ifsquare + 개인 공간 ifhome)



https://www.youtube.com/watch?v=GnkVNmo-n_4

Brand Identity

매일매일 메타버스 집들이 ifland

누구든 될 수 있고, 무엇이든 할 수 있고,
언제든 만날 수 있고, 어디든 갈 수 있는 곳!
ifland (이프랜드)는 모든 **if가 현실이 되는**
새로운 세상이라는 의미에서 탄생하였습니다.

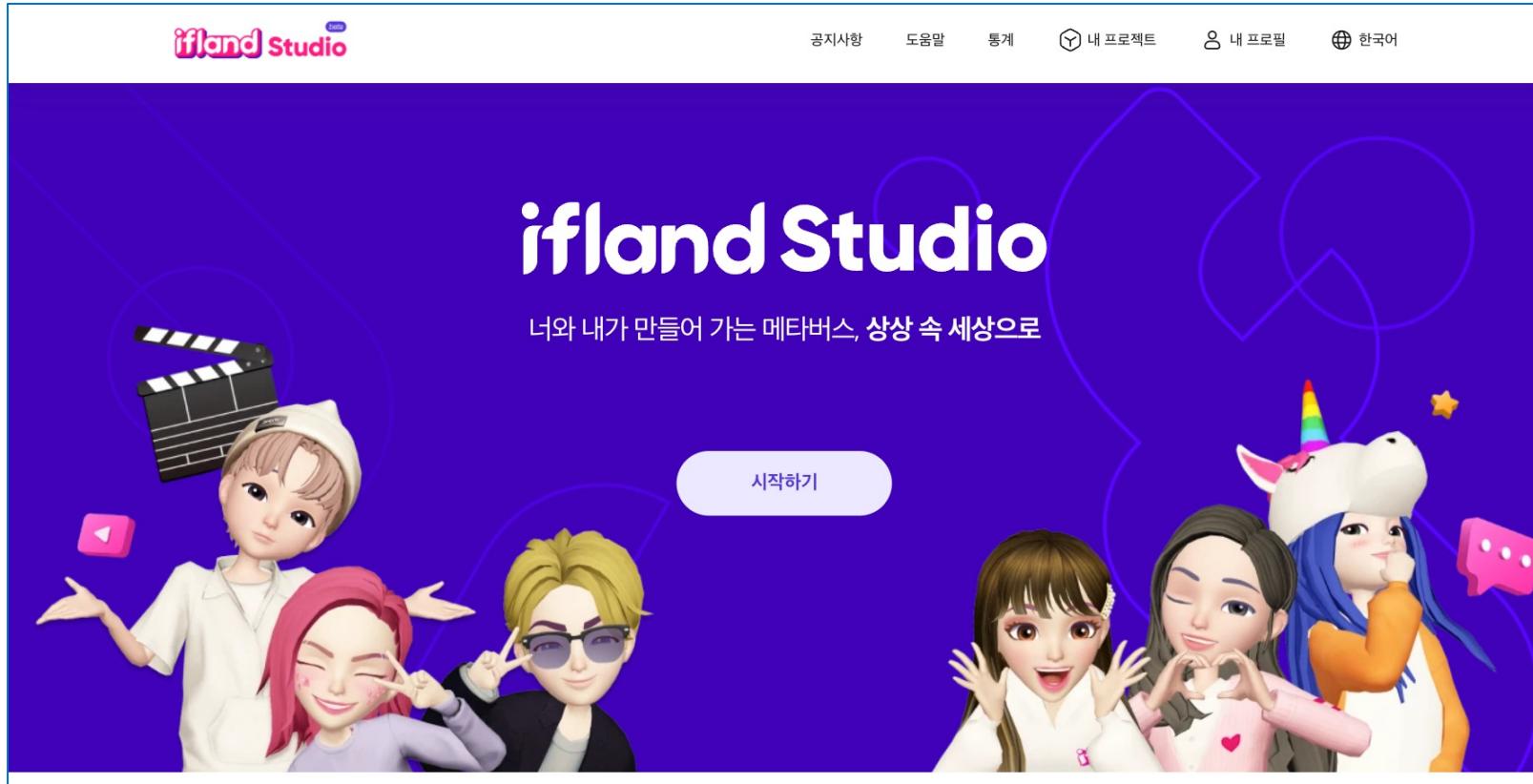
[AI 다운로드 >](#) [PNG 다운로드 >](#)

<https://ifland.io/>

What is ifland studio?

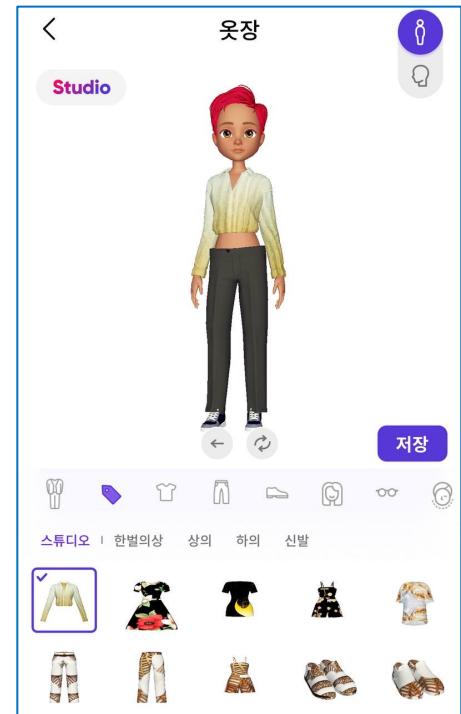
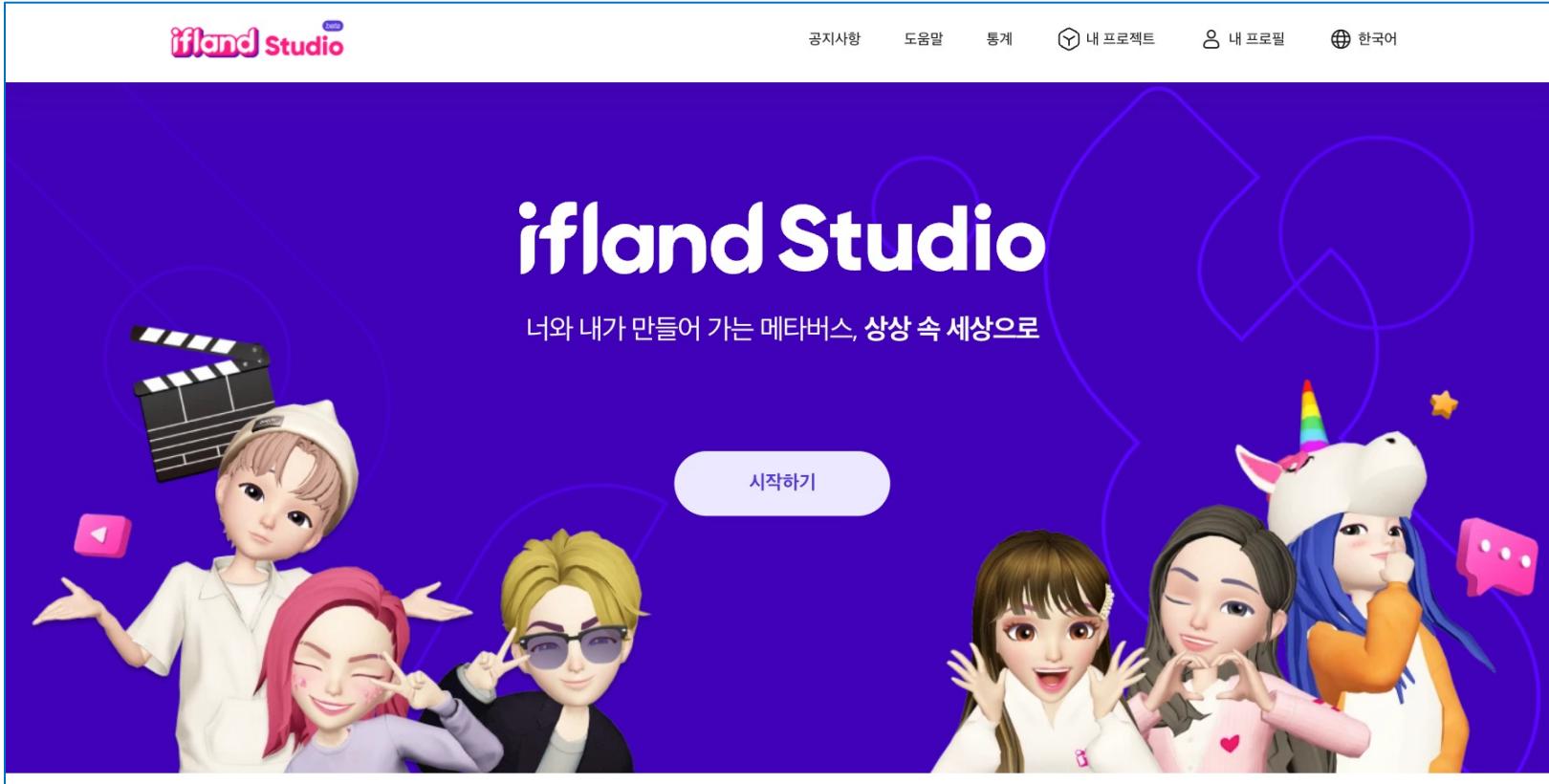
- **ifland studio**

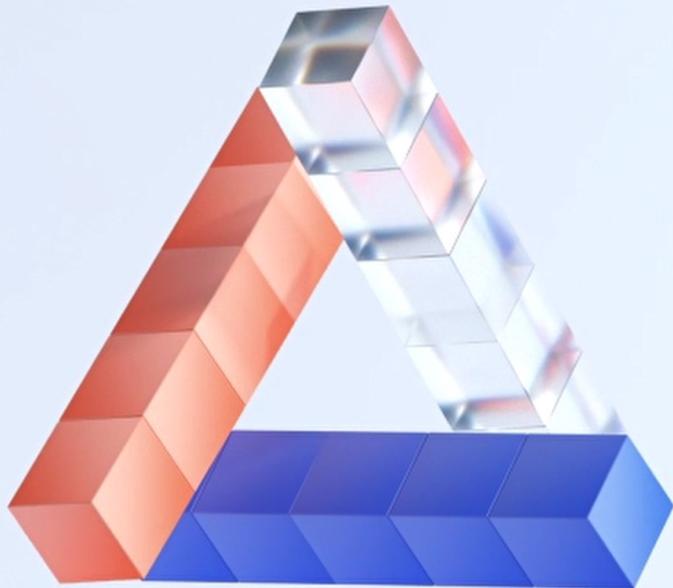
- ifland에서 **아바타(ifme) 의상을 만들 수 있는 곳**



What is ifland studio?

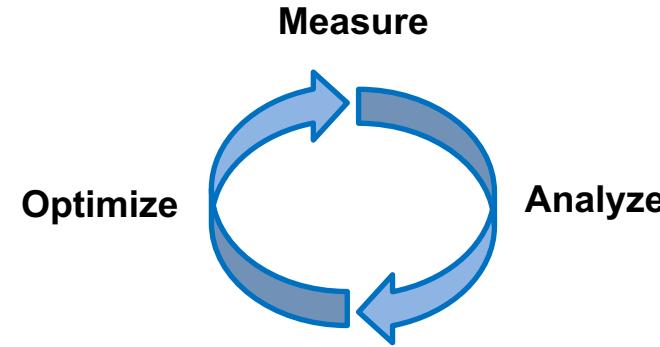
- 웹 성능 최적화 대상 페이지는?
 - 사용자들이 가장 많이 사용하고, 서비스의 진입점인 **home** (메인 페이지)





5. How to optimize web performance at ifland studio

Case : How to optimize web performance of ifland studio

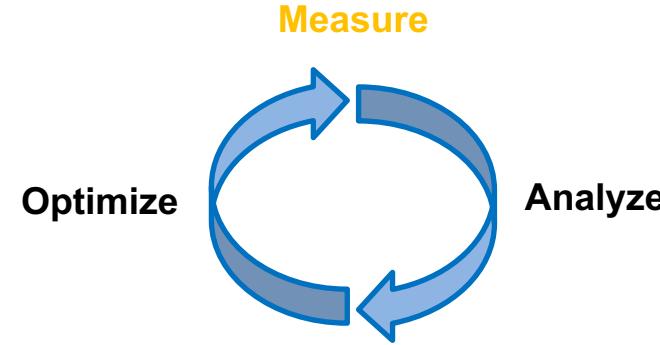


Measure : 현재 상황을 측정, 인지

Analyze : 현상 분석 및 개선점 도출

Optimize : 최적화 (개선) 적용

Case : How to optimize web performance of ifland studio



Measure : 현재 상황을 측정, 인지

Analyze : 현상 분석 및 개선점 도출

Optimize : 최적화 (개선) 적용

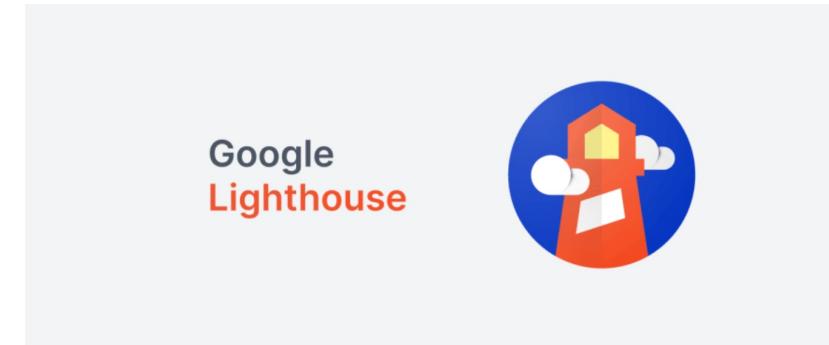
Case : How to optimize web performance of ifland studio

Say hello to **WebPageTest Pro**
All of the WebPageTest features you already love,
plus **API Access & No-Code Experiments!**

[View Plans & Learn More >>](#)

The screenshot shows the Catchpoint WebPageTest platform. At the top, there's a navigation bar with links for Platform, Start Test, Test History, Products, Pricing, Resources, About, Login, and Sign-up. Below the navigation, there's a large call-to-action button for "WebPageTest Pro". To the right of this button is a preview window showing a mobile device screen with a play button overlay. Below the preview is a section titled "Experiment Results" with several performance metrics listed:

Metric	Value	Status
First Contentful Paint	0.43s	FASTER
Speed Index	0.94s	FASTER
Largest Contentful Paint	1.42s	FASTER
First Meaningful Paint	0.05s	FASTER
Mean Complete	7.1s	FASTER
Header Blocking	4s	FASTER



Enter a web page URL.

Analyse

Make your web pages fast on all devices

CHECK OUT

[What's new](#)

[Documentation](#)

[Learn about web performance](#)

The screenshot shows the Google PageSpeed Insights tool. It has a simple form with a text input field for entering a URL and a blue "Analyse" button. Below the form, there's a large headline "Make your web pages fast on all devices". Underneath this, there's a section titled "CHECK OUT" with three links: "What's new", "Documentation", and "Learn about web performance". To the right of the text area is a small illustration of a smartphone, a laptop, and a tablet connected by arrows, symbolizing responsive design and multi-device optimization.

Case : How to optimize web performance of ifland studio

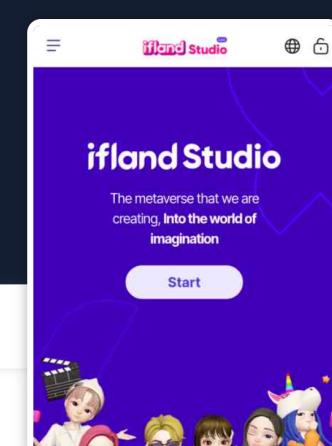
웹 성능 최적화 적용 전 측정 (Webpage test)

URL: <https://studio.ifland.io/> DATE: 2023. 3. 8. 오후 5:36:30

Webpage Performance Test Result

SETTINGS: MOTO G4 v111 4G Virginia USA More Share

View: Performance Summary Tools: Export Re-Run Test



Observed Metrics (Based on Median Run by: Speed Index)

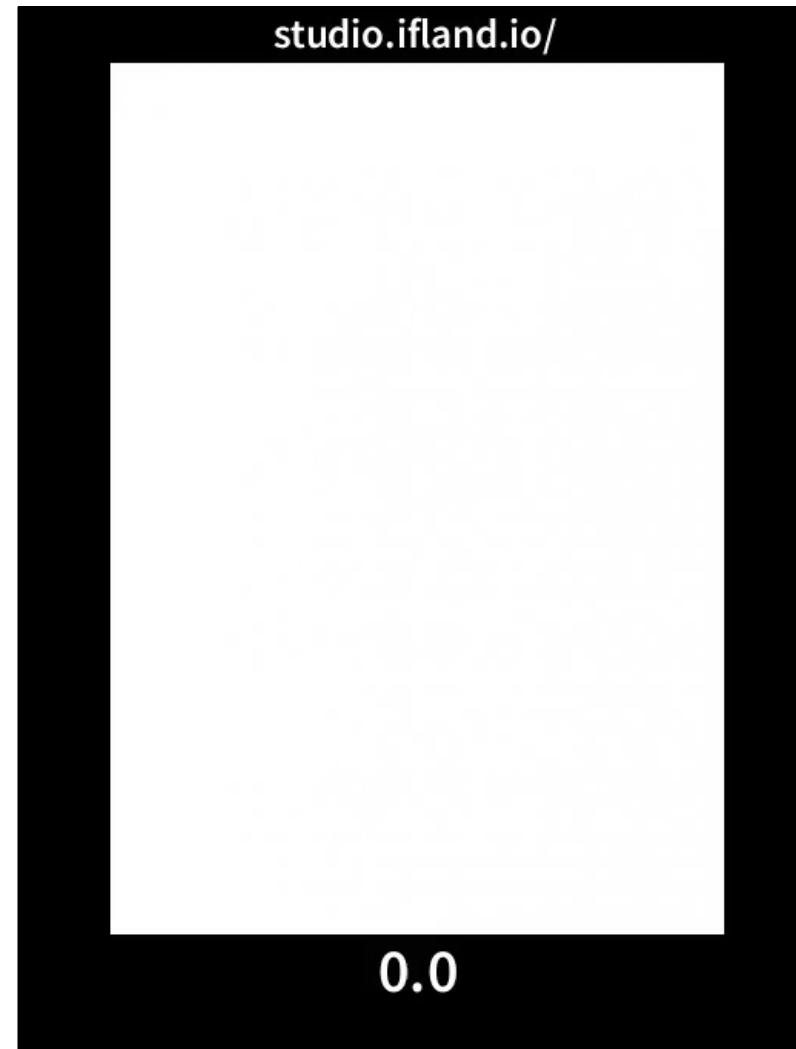
Note: Metrics offered will vary

FIRST VIEW (RUN 3)

First Byte	Start Render	FCP	Speed Index	LCP	CLS	TBT	Total Bytes
1.879 s	4.700 s	4.636 s	4.869 s	5.456 s	.006	≥ .480 s	2,843 KB

Case : How to optimize web performance of ifland studio

웹 성능 최적화 적용 전 측정 (Video)



Case : How to optimize web performance of ifland studio

M1 : Render Blocking Resource 다수 존재

M2 : 이미지 용량 문제

M3 : 해상도에 맞지 않는 이미지 사용

M4 : 네트워크 요청 낭비

M5 : 메인 이미지가 늦게 로드

M6 : 모바일 해상도에서 Header 아이콘 및 로고 이미지 늦게 로드

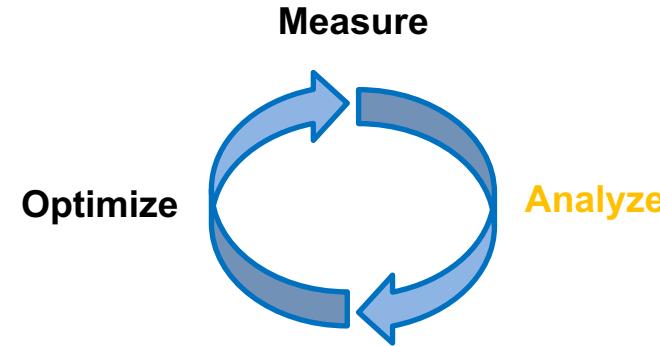
M7 : CSS, JS 용량 감소 여지 존재

M8 : 화면 영역에 불필요한 움직임 존재

M9 : html, JS 중복 코드 발견

M10 : 재방문자 페이지 접속시간 단축 개선점 발견

Case : How to optimize web performance of ifland studio

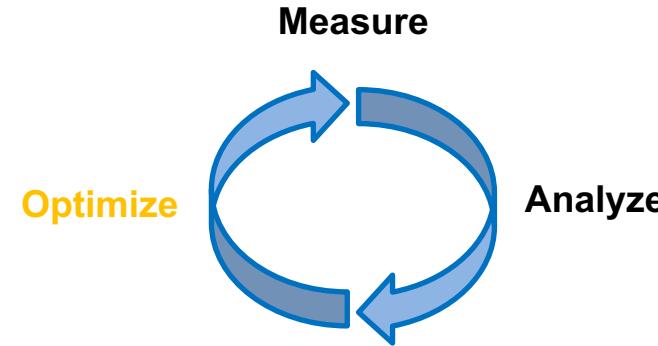


Measure : 현재 상황을 측정, 인지

Analyze : 현상 분석 및 개선점 도출

Optimize : 최적화 (개선) 적용

Case : How to optimize web performance of ifland studio



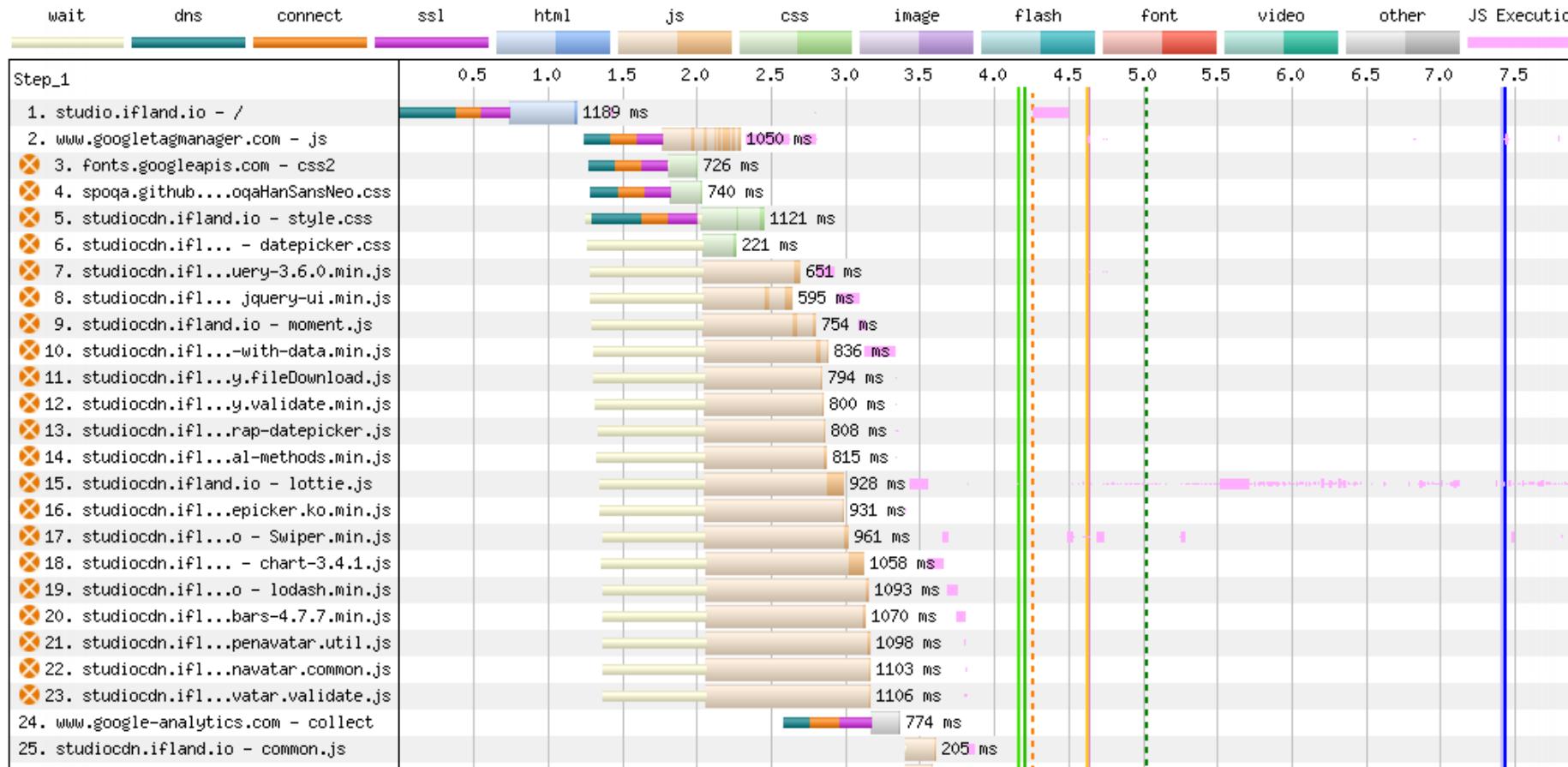
Measure : 현재 상황을 측정, 인지

Analyze : 현상 분석 및 개선점 도출

Optimize : 최적화 (개선) 적용

Case : How to optimize web performance of ifland studio

M1 : Render Blocking Resource 다수 존재

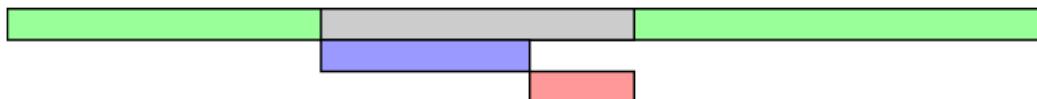


Case : How to optimize web performance of ifland studio

A1 : 스크립트 로딩 옵션이 존재

<script>

Let's start by defining what <script> without any attributes does. The HTML file will be parsed until the script file is hit, at that point parsing will stop and a request will be made to fetch the file (if it's external). The script will then be executed before parsing is resumed.



Legend

- HTML parsing
- HTML parsing paused
- Script download
- Script execution

<script async>

async downloads the file during HTML parsing and will pause the HTML parser to execute it when it has finished downloading.



<script defer>

defer downloads the file during HTML parsing and will only execute it after the parser has completed. defer scripts are also guaranteed to execute in the order that they appear in the document.



<head>

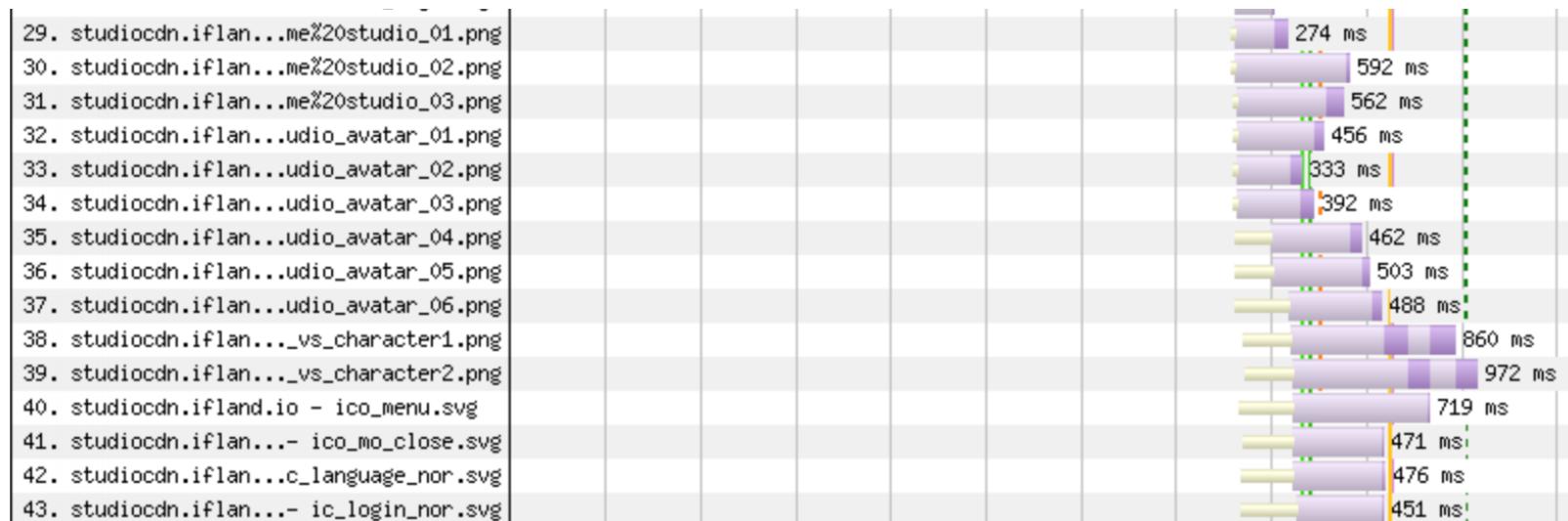
```
...
<script src="https://test-defer.js" defer></script>
...
</head>
```

<https://www.growingwiththeweb.com/2014/02/async-vs-defer-attributes.html>

O1 : 불필요 스크립트 제거 및
스크립트 로딩 옵션 지정을 통한 render blocking 요소 제거

Case : How to optimize web performance of ifland studio

M2 : 이미지 용량 문제

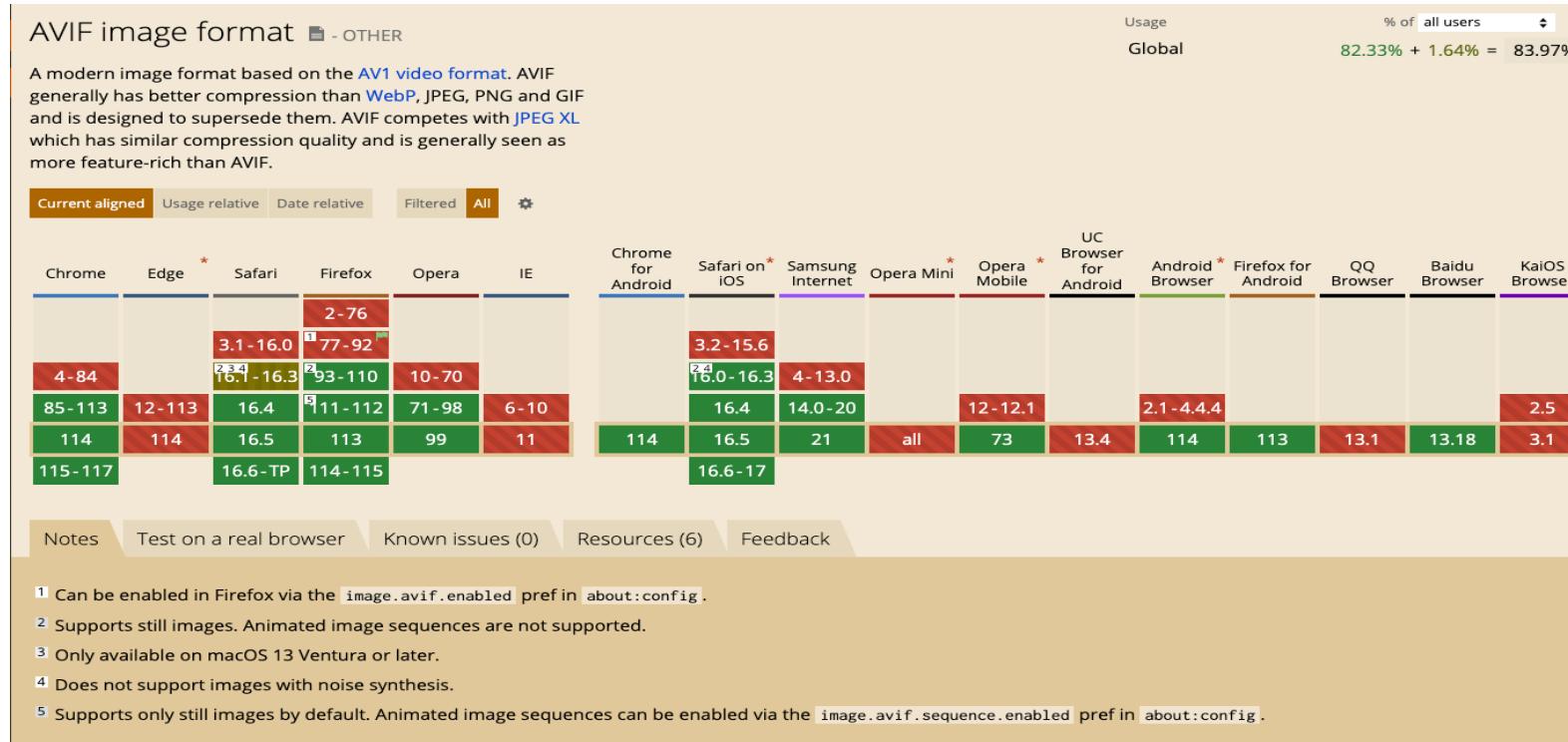


Case : How to optimize web performance of ifland studio

A2 : WebP, AVIF 가 PNG, JPEG에 비해 압축률 우수

⚠ Serve images in next-gen formats 1.09s ^

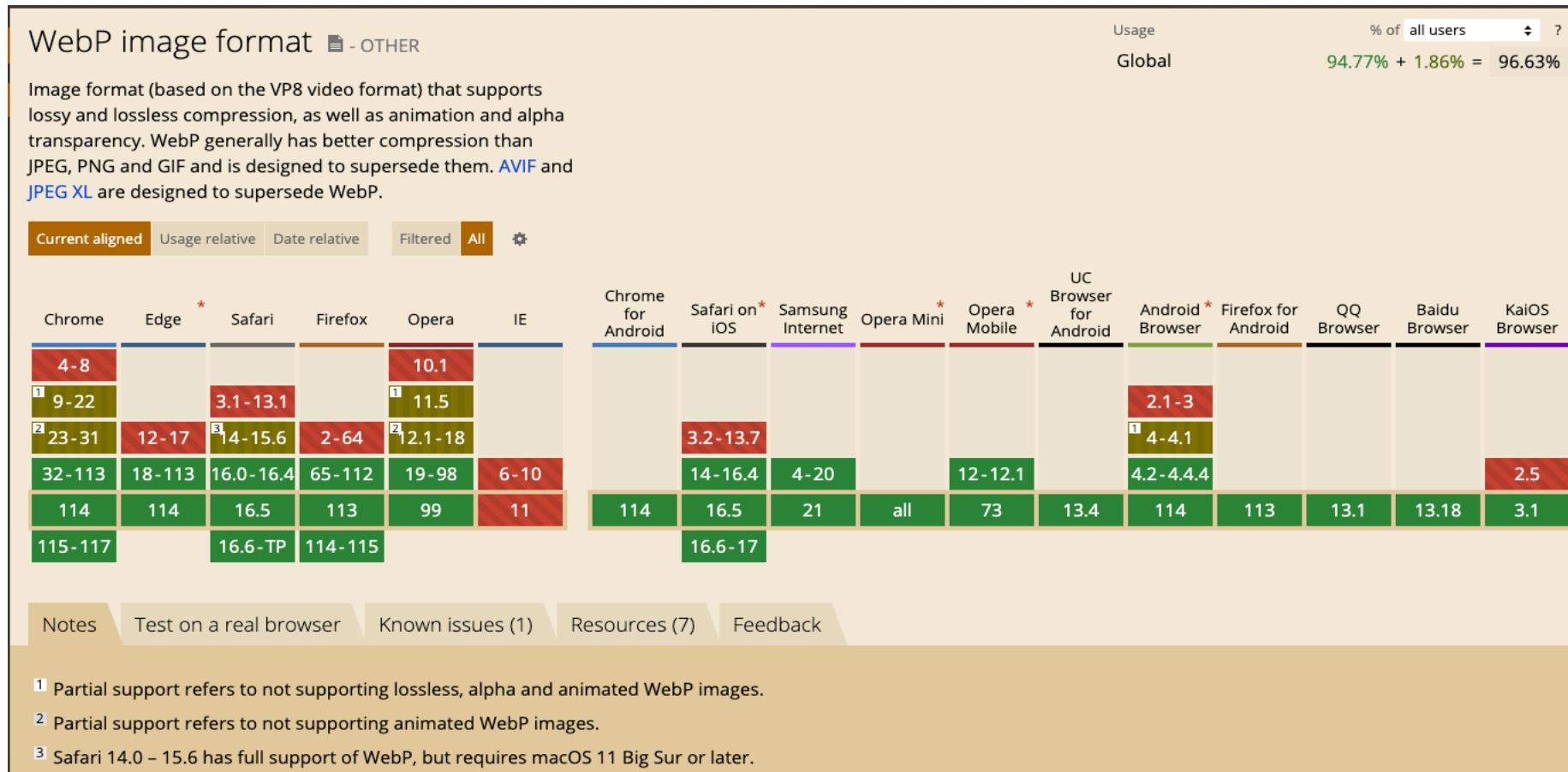
Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption. [Learn more about modern image formats.](#)



<https://caniuse.com/>

Case : How to optimize web performance of ifland studio

A2 : Image 포맷 PNG에서 WebP로 변경



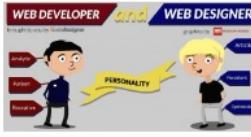
<https://caniuse.com/>

O2 : 기존 대비 이미지 사이즈 크게 감소

Case : How to optimize web performance of ifland studio

M3 : 해상도에 맞지 않는 이미지 사용, 즉 Desktop과 Mobile 공통 이미지 사용

s_article_rep -->



Rendered size: 162 × 81 px

Rendered aspect ratio: 2: 1

Intrinsic size: 1252 × 627 px

Intrinsic aspect ratio: 1252: 627

File size: 691 kB

Current source: <https://img1.daumcdn.net/thumb/R1280x0/?scode=mtis...%2FbtqMBvzfnxY%2FjnNcfM5Y8ZSNedHs14UWGk%2Fimg.png>

```
 ... </div>

> <div class="character1"> ... </div>

> <d...



Rendered size: 222 × 180 px

Rendered aspect ratio: 37: 30

Intrinsic size: 444 × 360 px

Intrinsic aspect ratio: 37: 30

File size:

Current source: [https://studiodcdn.ifland.io/images/layout/img\\_main\\_02\\_m.webp](https://studiodcdn.ifland.io/images/layout/img_main_02_m.webp)

```

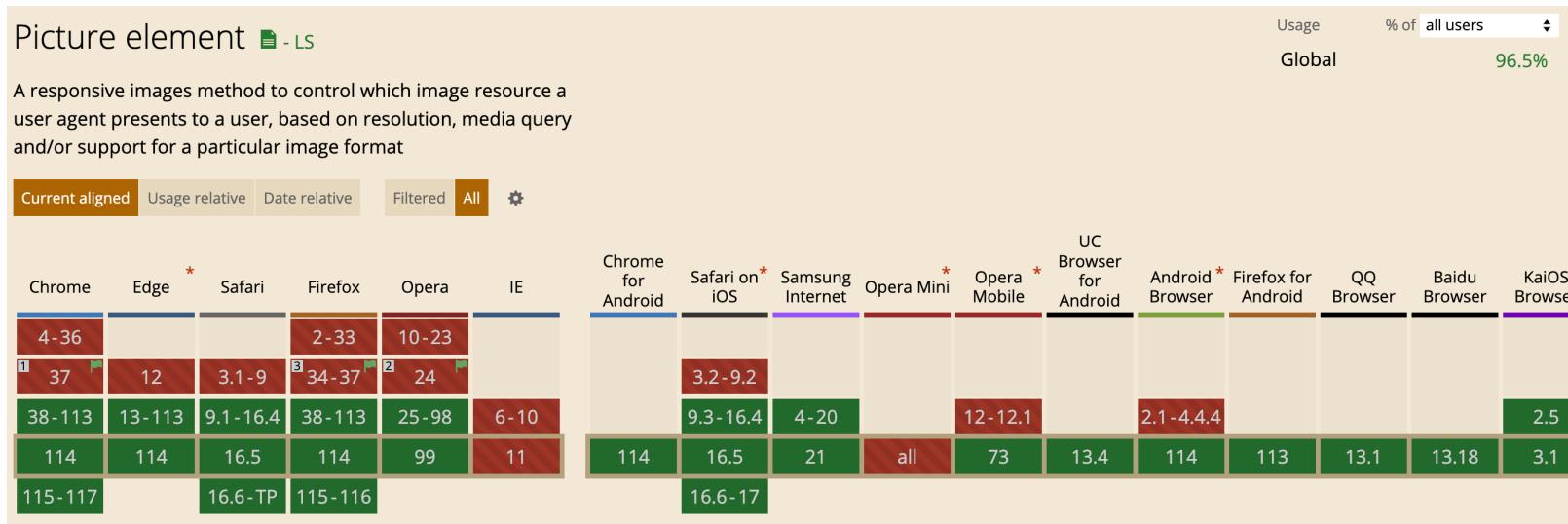
```

적절한 해상도 이미지 사용

A3 : (디자인) Desktop, Mobile용 별도 이미지 제작 필요

# Case : How to optimize web performance of ifland studio

A3 : picture source 태그 사용하여 Desktop, Mobile 별도 이미지 적용



<https://caniuse.com/>

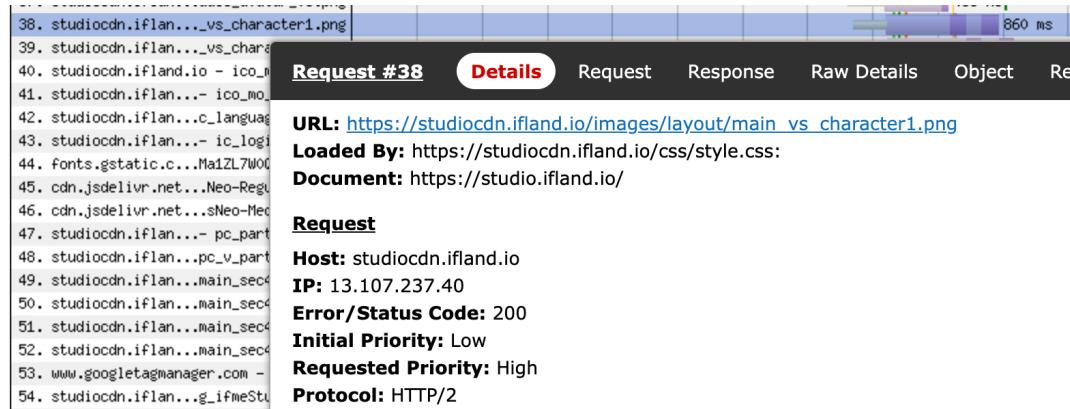
```
<picture>
 <source media="(max-width:1200px)" srcset="https://studiocdn.ifland.io/images/layout/img_main_01_m.webp" type="image/webp">
 <source media="(max-width:1200px)" srcset="https://studiocdn.ifland.io/images/layout/img_main_01_m.png" type="image/png">
 <source media="(min-width:1201px)" srcset="https://studiocdn.ifland.io/images/layout/img_main_01.webp" type="image/webp">
 <source media="(min-width:1201px)" srcset="https://studiocdn.ifland.io/images/layout/img_main_01.png" type="image/png">

</picture>
```

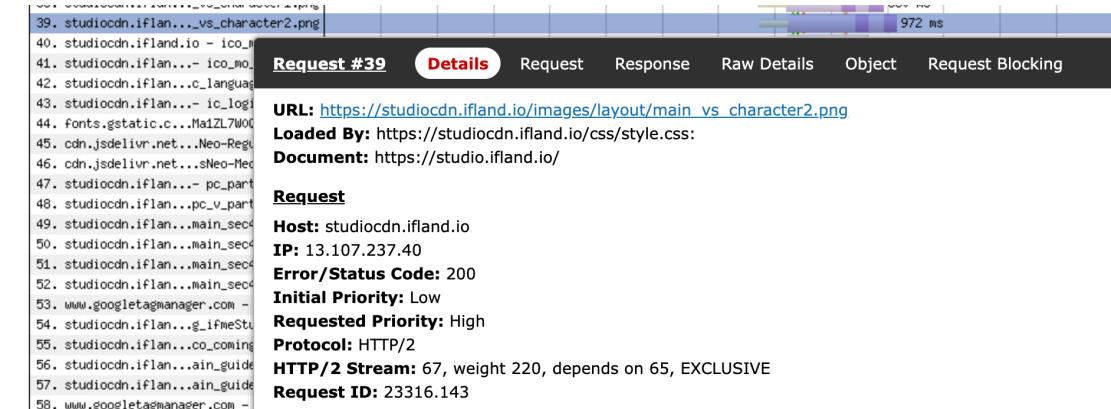
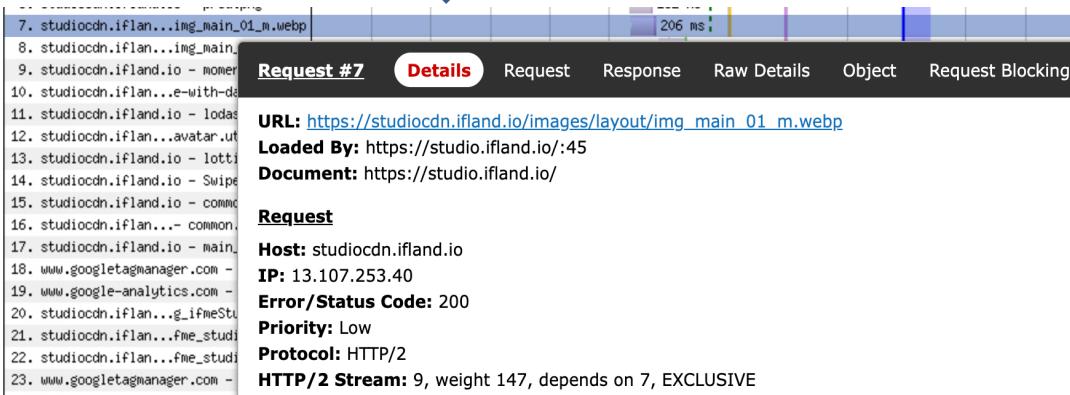
# Case : How to optimize web performance of ifland studio

A2 : Image 포맷 PNG에서 WebP로 변경

A3 : 기기, 해상도에 맞는 이미지 사용



이미지 용량 76% 감소



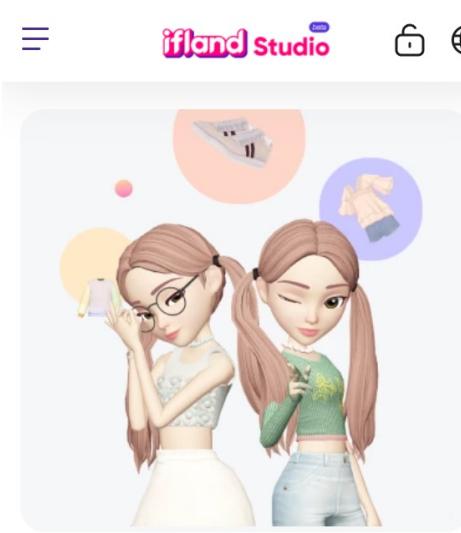
이미지 용량 78% 감소



O2,O3 : 기존 대비 이미지 하나당 사이즈 60~80%정도 감소

# Case : How to optimize web performance of ifland studio

M4 : 네트워크 요청 낭비 (페이지 로딩시 불필요한 이미지 한번에 로드)

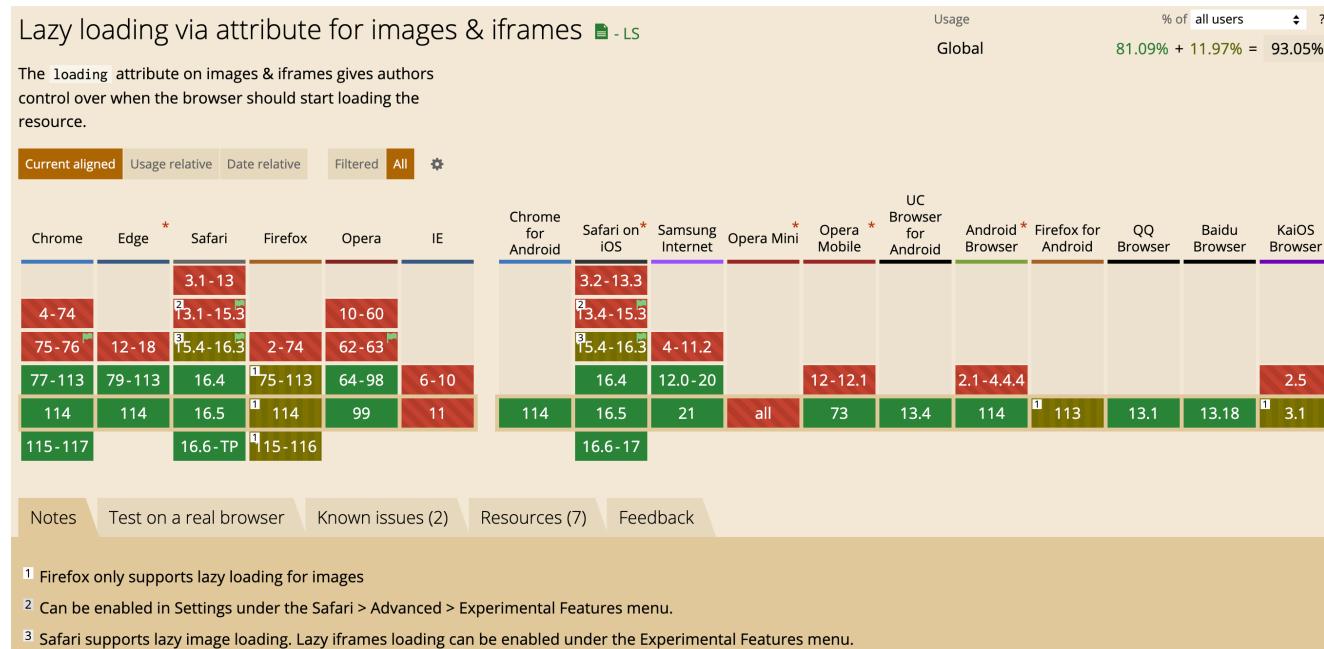


M4 : 페이지 로드시 이미지 네트워크 요청 수 20개

# Case : How to optimize web performance of ifland studio

## A4 : 이미지 lazy loading

메인 이미지 제외한 이미지에 대해 lazy loading 적용



```
<picture>
 <source srcset="https://studiodcdn.ifland.io/images/layout/img_ifmestudio_avatar_01.webp" type="image/webp">
 <source srcset="https://studiodcdn.ifland.io/images/layout/img_ifmestudio_avatar_01.webp" type="image/png">

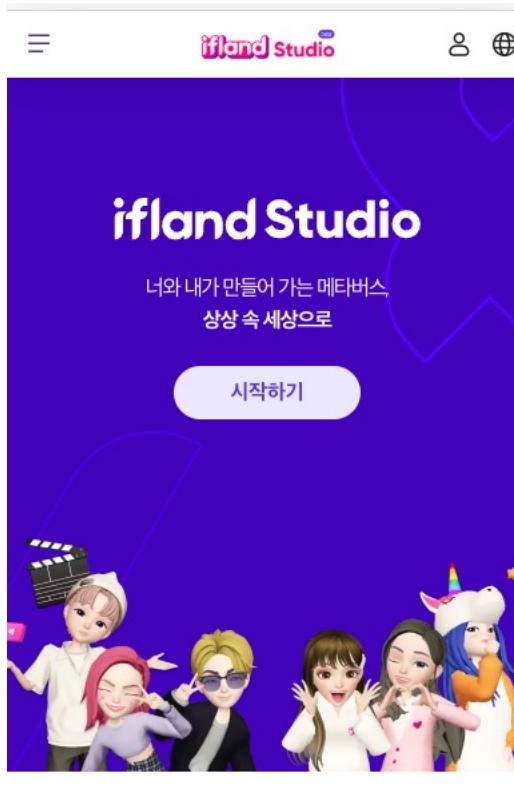
</picture>
```

O4 : 불필요한 네트워크량 감소  
페이지 초기 로드시 이미지 요청 수 20개에서 2개로 감소

# Case : How to optimize web performance of ifland studio

M5 : 메인 이미지가 늦게 로드 (LCP, UX 영향)

A5 : 메인 이미지 Preload 적용

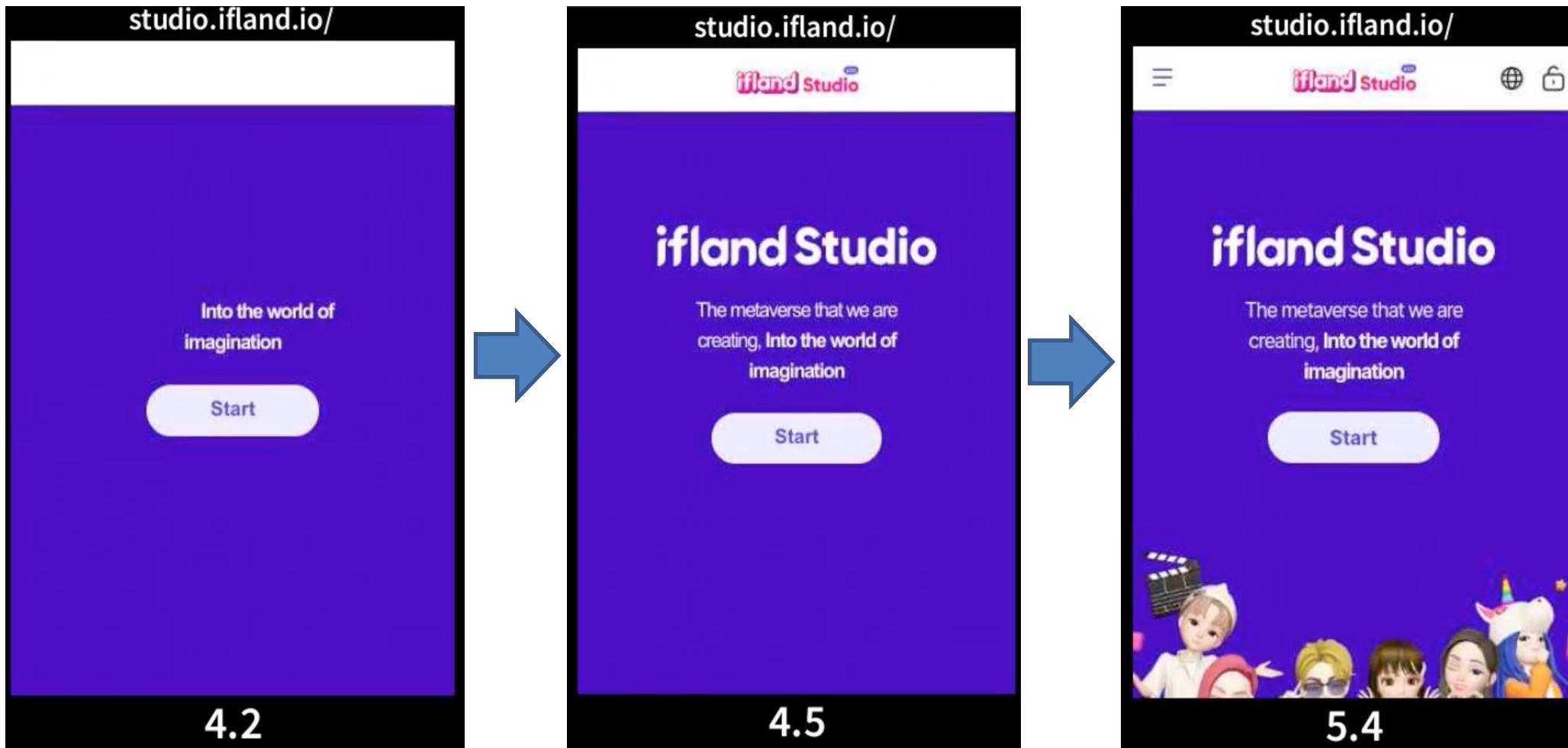


```
<link rel="preload" as="image" media="(max-width:1200px)" href="https://studiocdn.ifland.io/images/layout/img_main_01_m.webp" type="image/webp">
<link rel="preload" as="image" media="(min-width:1201px)" href="https://studiocdn.ifland.io/images/layout/img_main_01.webp" type="image/webp">
```

O5 : 메인 이미지 preload 적용하여 로딩속도 개선

# Case : How to optimize web performance of ifland studio

M6 : 모바일 해상도에서 Header 아이콘 및 로고 이미지 늦게 로드 (UX)



# Case : How to optimize web performance of ifland studio

A6 : Header 아이콘을 html 내 inline svg로 추가, 로고 이미지 preload 추가



O6 : 자연스러운 웹 사이트 로딩으로 UX 개선

# Case : How to optimize web performance of ifland studio

M7 : CSS, JS 용량 감소 여지 존재

A7 : CSS, JS gzip, minify 작업

▼ General

**Request URL:** <https://studiodcdn.ifland.io/js/moment.min.js>

**Request Method:** GET

**Status Code:** 200

**Remote Address:** 150.2.127.249:9090

**Referrer Policy:** strict-origin-when-cross-origin

▼ Response Headers

**access-control-allow-methods:** GET

**access-control-allow-origin:** <https://studio.ifland.io>

**access-control-expose-headers:** ETag

**content-encoding:** gzip

**content-type:** application/javascript

**date:** Wed, 21 Jun 2023 11:58:04 GMT

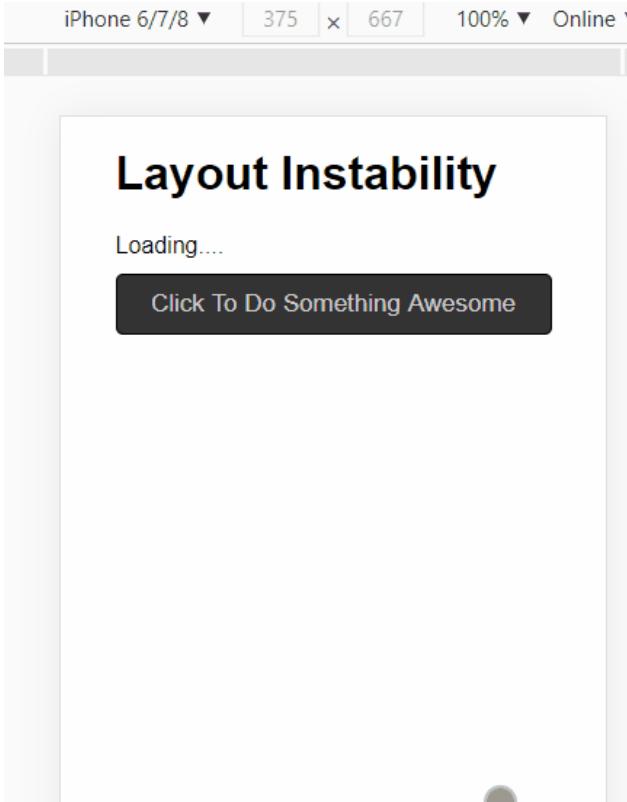
**last-modified:** Mon, 24 Apr 2023 23:02:08 GMT

**vary:** Origin,Accept-Encoding,Access-Control-Request-Method,Access-Control-Request-Headers

O7 : 서버 네트워크 요청 용량 감소

# Case : How to optimize web performance of ifland studio

M8 : 화면 영역에 불필요한 움직임 존재



# CLS

Cumulative Layout Shift

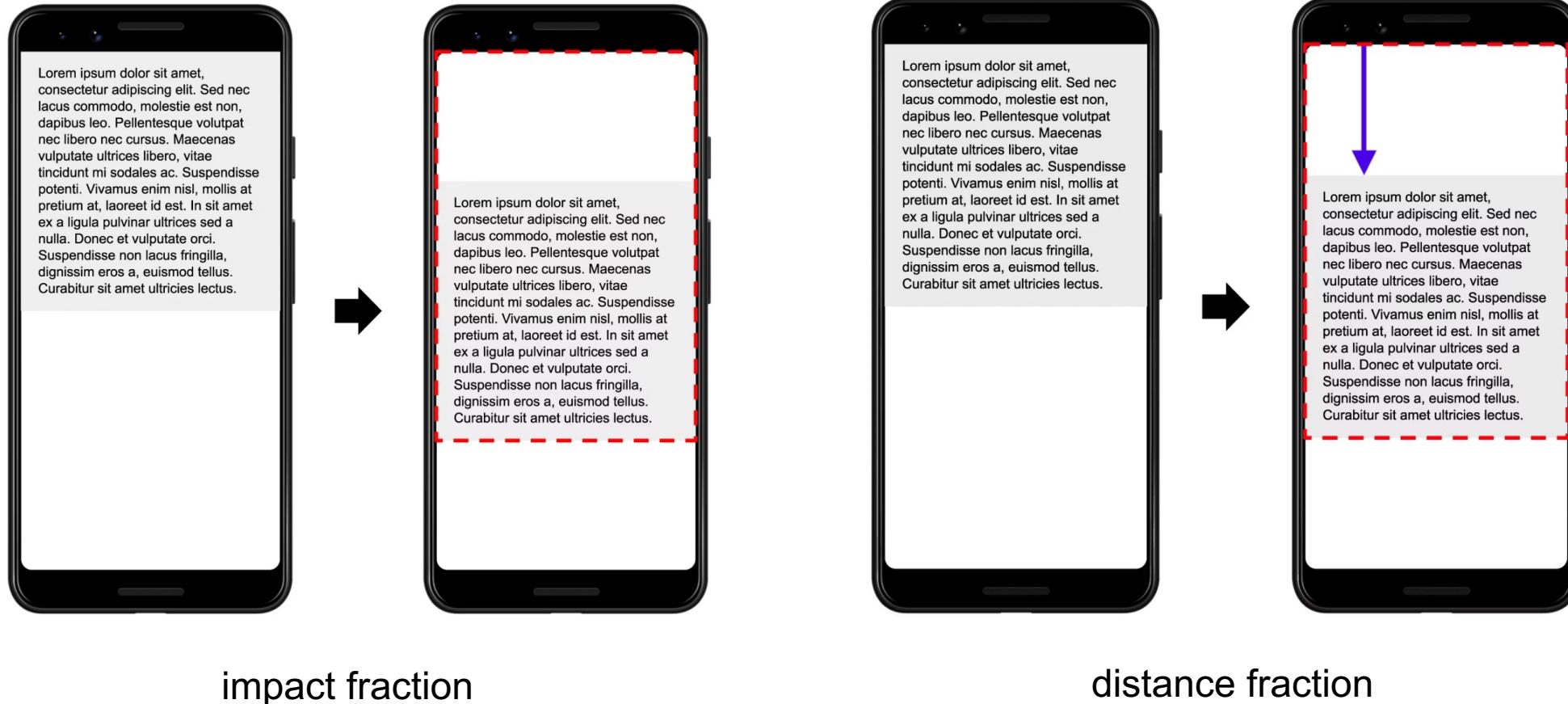


<https://web.dev/cls/>

<https://requestmetrics.com/web-performance/cumulative-layout-shift>

# Case : How to optimize web performance of ifland studio

Layout shift score = impact fraction \* distance fraction



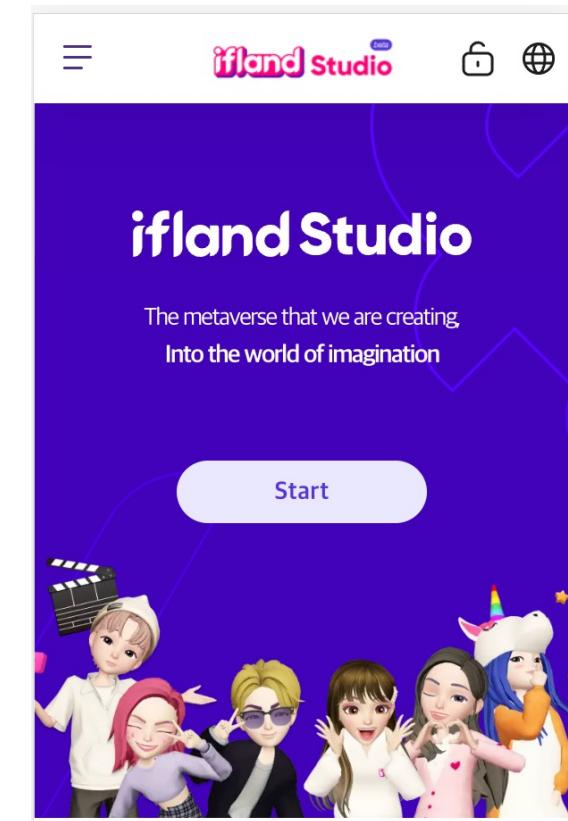
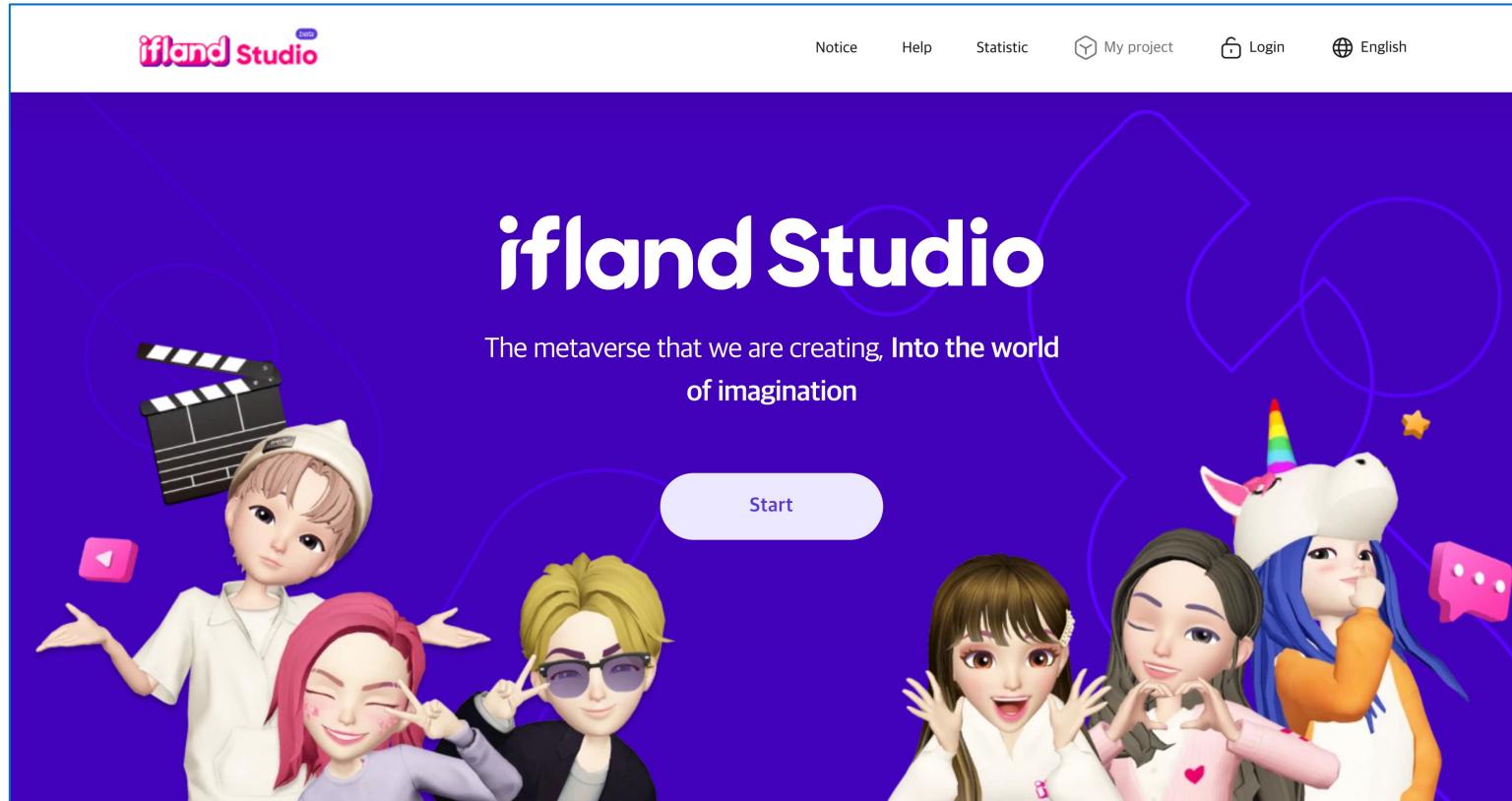
<https://web.dev/cls/>

A8 : 콘텐츠 로드 부분 영역 확보  
O8 : CSS 설정을 통해 CLS 수치 0으로 개선

# Case : How to optimize web performance of ifland studio

M9 : html, JS 중복 코드 발견(Desktop, Mobile)

A9 : 코드 개선을 통한 소스 경량화

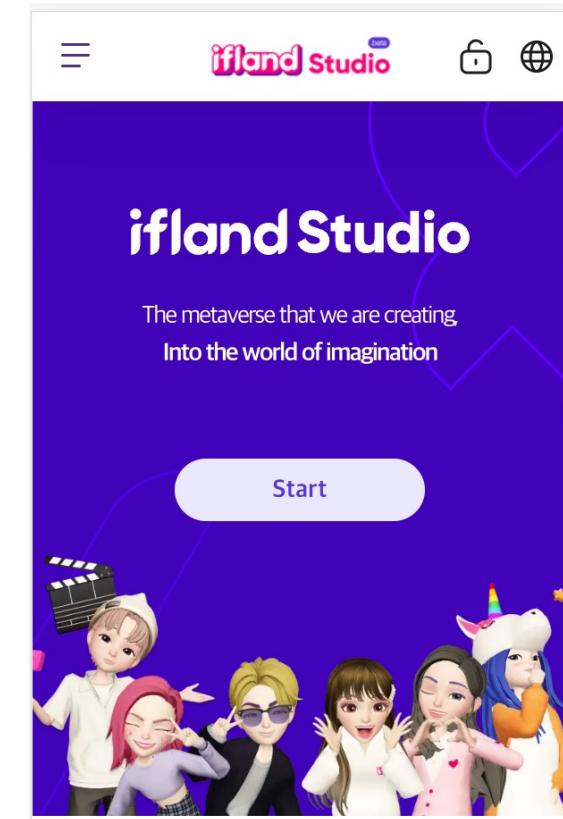
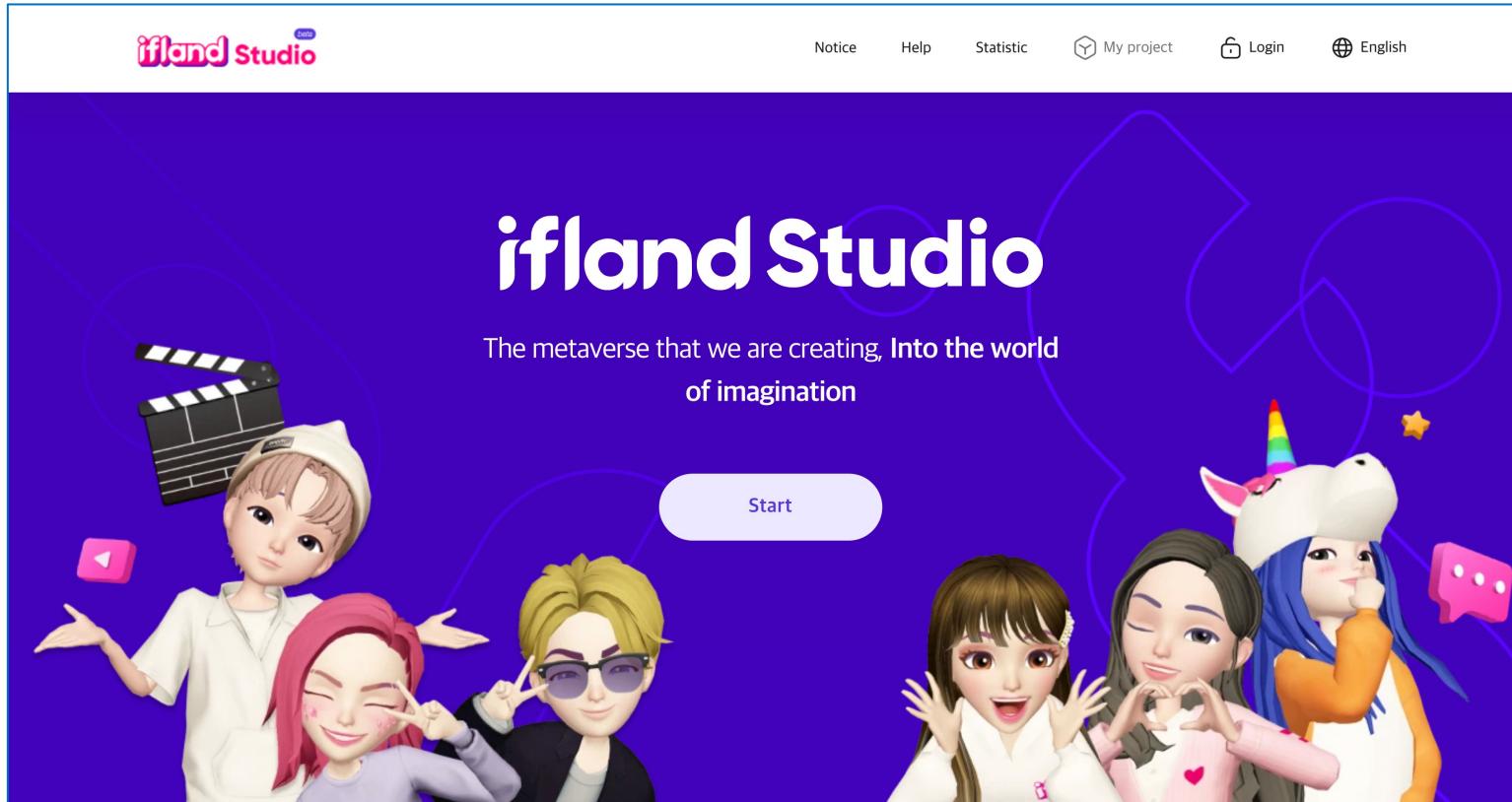


O9 : 서버 네트워크 요청 용량 감소

# Case : How to optimize web performance of ifland studio

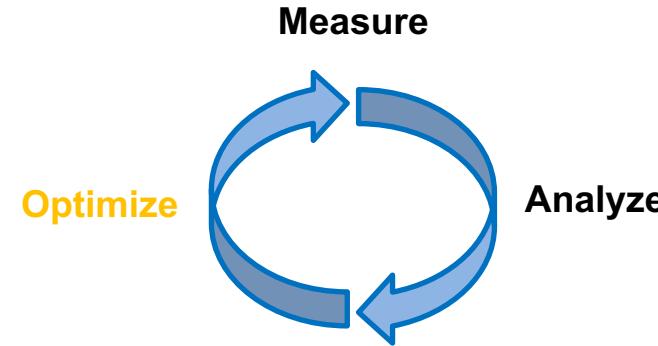
M10 : 재방문자 페이지 접속시간 단축 개선점 발견

A10 : 백엔드에서 브라우저 캐싱 설정 (Request header 옵션 추가)



O10 : 일부 콘텐츠 브라우저 캐시 적용 (예정)

# Case : How to optimize web performance of ifland studio



**Measure** : 현재 상황을 측정

**Analyze** : 현상 분석 및 개선점 도출

**Optimize** : 최적화 (개선)

# Case : How to optimize web performance of ifland studio

## (종합) 웹 성능 최적화 방법

M1 : Render Blocking Resource 다수 존재

- 스크립트 로딩 옵션 변경 : defer, async 옵션

M2 : 이미지 용량 문제

- webp 이미지로 전환

M3 : 해상도에 맞지 않는 이미지 사용

- img 태그를 picture 방식으로 변경

M4 : 네트워크 요청 낭비

- 이미지 lazy load

M5 : 메인 이미지가 늦게 로드

- 메인 이미지 preload 적용

M6 : (모바일) Header 아이콘 및 로고 이미지 늦게 로드

- Header 이미지 : png에서 html inline svg로 변경

M7 : CSS, JS 용량 감소 여지 존재

- 사용하는 css, js 최적화 (minify, gzip)

M8 : 화면 영역에 불필요한 움직임 존재

- HTML, JS 효율화 (Desktop, Mobile 중복코드 제거)

M9 : HTML, JS 중복 코드 발견

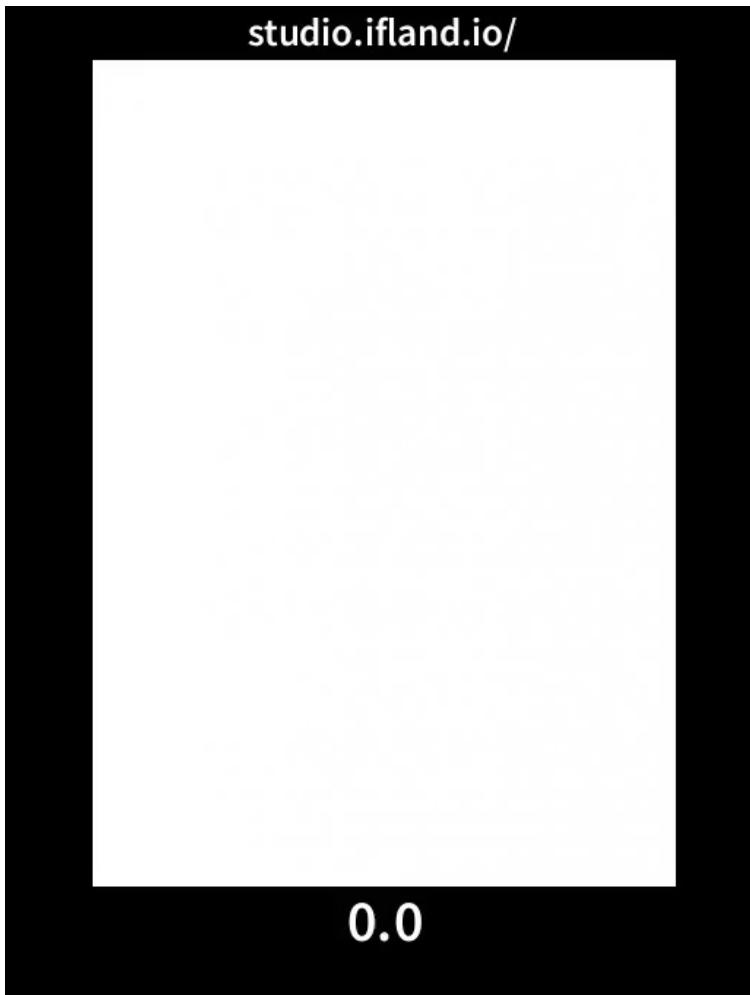
- CLS 개선 (UX, DOM 영역 늘어지는 부분 확인)

M10 : 재방문자 페이지 접속시간 단축 개선점 발견

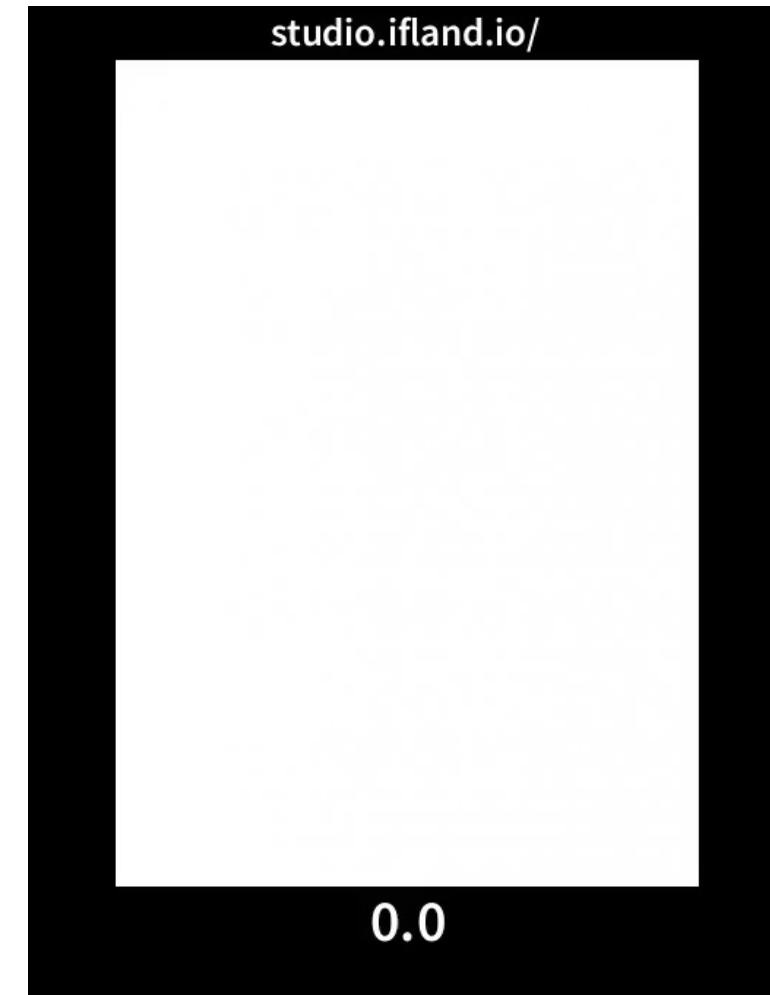
- 리소스 브라우저 캐싱

# Case : How to optimize web performance of ifland studio

- 웹 성능 최적화 적용 후 측정 (Video)



성능 개선 전



성능 개선 후

# Case : How to optimize web performance of ifland studio

- 웹 성능 최적화 적용 후 측정 (Webpage test)

URL: <https://studio.ifland.io/> DATE: 2023. 6. 21. 오후 4:38:05

### Webpage Performance Test Result

SETTINGS: MOTO G4 v114 4G Virginia USA More Share

View: Performance Summary Tools: Export Re-Run Test

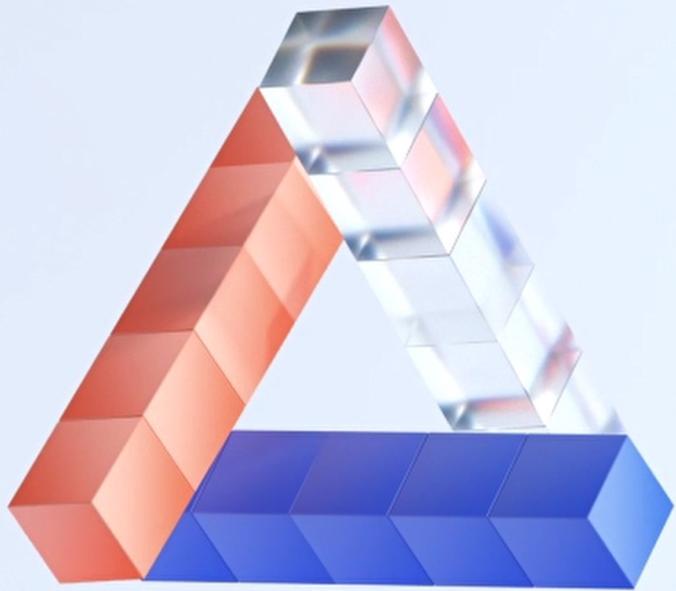
**Page Performance Metrics** (Based on Median Run by: Speed Index)

**Note:** Metric availability will vary.

First View (Run 2)	Time to First Byte	Start Render	First Contentful Paint	Speed Index	Largest Contentful Paint	Cumulative Layout Shift	Total Blocking Time	Page Weight
When did the content start downloading?	1.210s	3.300s	3.287s	3.406s	3.319s	0	.526s	637 KB
When did pixels first start to appear?	When did the content start downloading?	When did the pixels first start to appear?	How soon did text and images start to appear?	How soon did the page look usable?	When did the largest visible content finish loading?	How much did the design shift while loading?	Was the main thread blocked?	How many bytes downloaded?

- 개선 결과 \* 측정환경마다 차이 존재

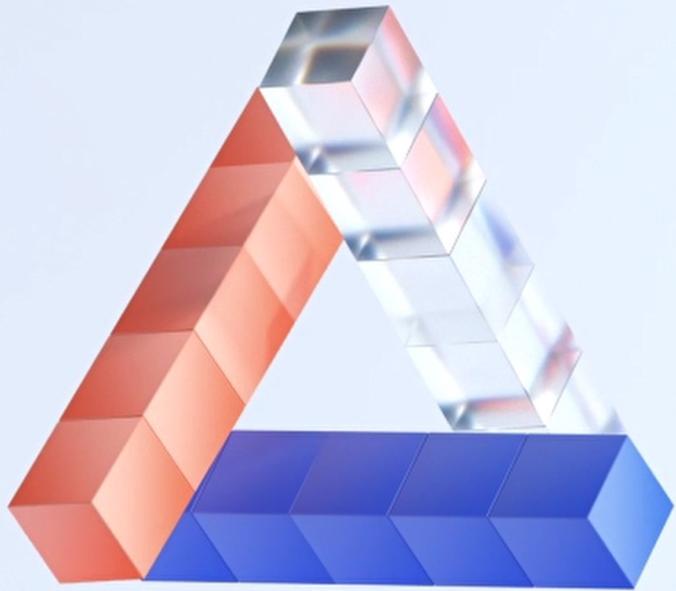
	FCP	Speed Index	LCP	CLS	TBT	Page Weight
개선 전	4.646s	4.869s	5.456s	0.006	0.480s	2843KB
개선 후	3.287s	3.406s	3.319s	0	0.526s	637KB
결과 (차이)	31% 감소	30% 감소	39% 감소	100% 감소	비슷 (약간 증가)	78% 감소



## 6. Lessons learned & tips

# Lessons learned & tips

1. Script에 옵션 추가 (defer, async)
2. WebP, AVIF 이미지 포맷 적용
3. 이미지 lazy loading 적용
4. 리소스 Preload 옵션 적용
5. 개발코드는 작품을 만들 수 있지만, 부채가 될 수도 있음 (항상 경량화를 염두)
6. CWV 개선은 지속적인 노력이 요구되는 작업
7. 웹과 그를 위한 기술은 지속적으로 변화하고 발전함



## 7. Conclusion

# Conclusion

1. 웹 프론트엔드 개발자는 웹 성능 개선을 통해 서비스 향상에 크게 기여할 수 있다.
2. 웹 프론트엔드 성능 개선은 비즈니스 관점에서 상당히 중요하다.  
(로딩 시간이 길면 사용자는 이탈한다)
3. 사용자 중심으로 사이트를 지속적으로 개선하는 문화를 만들 필요가 있다.
4. 측정 가능한 지표를 정의하고 목표를 설정하고 달성하는 일은 재미있다.  
(인정 받기까지 한다면 더욱 좋다)

# 감사합니다!