

遅いサイトは嫌われる！

フロントエンド改善テク3選

Ok

なぜ速さが大事？

- ・ 表示が遅いとユーザー離脱が↑
- ・ コンバージョン（購入率・申し込み率）にも直結
- ・ Google検索順位（SEO）にも影響

 デザインと同じくらい「速さ」がサービスの価値になる！

① 画像最適化

- ・ 画像形式はJPEG／PNGよりWebP／AVIFのほうが軽量

WebP（ウェッピー）とは？

→Google社が開発した画像形式。JPEG／PNGより圧縮率が高く軽量。

AVIFとは？

→AV1 Image File Formatの略で非営利団体Alliance for Open Media（Google、Amazon、Appleなどの多くの企業が参加している）によって開発された画像フォーマット。WebPよりさらに高圧縮＆高品質。

② 初期表示を軽くする

- ・ lazy loadで画像読み込みを早くする

→遅延ロードを行うライブラリ。JavaScriptで記述する。

実装することでWebページの表示速度が上がり、検索エンジンからの評価が高まる。

遅延ロードとは？

通常ではimgタグの画像ファイルを一度に全て読み込む仕組みになっているのを、ユーザーに表示している範囲内の画像のみを読み込む仕組みのこと。

2 初期表示を軽くする

- ・ **defer／async**でJSファイルの読み込みを早くする

→HTMLでJavaScriptを読み込むときに使う属性。

`<script src="main.js"></script>`だとHTMLの読み込みが一時停止してJSを読み込むのでページ表示が遅くなる。

`<script src="main.js" defer></script>`や`<script src="analytics.js" async></script>`のような記述をするとJSの読み込みのタイミングが早くなる。

deferとasyncの違い

defer→HTMLのパース（解析）と並行してJSを読み込む。JSの実行はHTMLが読み込まれた後、順番に実行される。基本はdefer推奨。

async→同じようにHTMLのパース（解析）と並行してJSを読み込むが、読み込みが終わったらHTMLが途中でも実行されるので実行順序がバラバラになる可能性がある。
独立したスクリプト（例：広告タグ、解析タグ）に向いている。

2 初期表示を軽くする

- ・ **preload**でフォントの読み込みを早くする

→通常はCSSを解析してからフォントを取りに行くが、**preload**の記述でページ表示と同時にフォントを読み込み開始できる。

- ・ **font-display: swap** をCSSに記述でフォントの読み込みを早くする

→フォントが読み込まれる前でシステムフォントで代用する。

読み込み後に対象フォントに切り替える。

これによりユーザーに文字が出ない状態を避けることができる。

3 コード最適化

- ・ **Code Splitting(コード分割)**

→アプリを小さなファイルに分けて必要な時だけ読み込む

例：全部まとめた[bundle.js](#)を最初に読み込むと表示が遅い

これをページごと・機能ごとに分割することで、ユーザーが必要なコードだけ読み込むことができ
初期表示が速くなる

- ・ **Tree Shaking(不要コードの削除)**

→使われていないコードを自動的に削除して最終的なファイルを軽量化する仕組み。

JavaScriptでコードをファイルごとに分割して再利用できる仕組みであるES Modules(ECMAScript Modules)を利用してコードを整理して必要な部分だけ取り込める。

参考資料

- ・ 結局Webサイトの画像なにを使えばいいのよ2024春【WebP/AVIF】

https://qiita.com/whike_chan/items/4cbec9d30002f9212ec7

- ・ WebPはもう古い？次世代画像フォーマット「AVIF」の実力とは

<https://note.com/takahide1013/n/ncefacd939d86>

- ・ Lazy Loadとは？メリットや注意点・実装方法を解説

<https://digitalidentity.co.jp/blog/engineerblog/web-performance/lazy-load.html>

- ・ HTMLの<script>タグのasyncとdeferを覚える

<https://leapcell.io/blog/ja/html-script-tag-async-to-difa-oboe-ru>

- ・ 【React_70】 コード分割（Code Splitting）の意義

https://note.com/happy_avocet7237/n/n81797d6e58df

- ・ Tree Shaking とは？

https://zenn.dev/btc/articles/240403_tree_shaking



LINE公式アカウントにて
最新情報配信中！



無料で自由に使える
学習 & カフェスペース開放中！

変化を楽しみ、自分らしく未来へ。

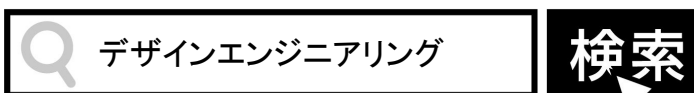
デザインエンジニアリングは、
挑戦するエンジニアの一步を応援する会社です。

“好き”や“ワクワク”をそのままキャリアに変え、
自分の可能性を信じて前へ進む人には、無限のチャンスが広がっています。

失敗も学びに変え、仲間と共に笑い、共に成長しながら、
毎日が少しずつ楽しくなる未来へ。
未経験でも大丈夫。あなたの最初の一步を、心からお待ちしています！

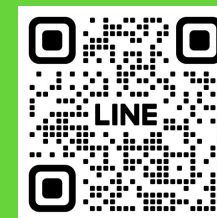


イベント・セミナー開催中！



URL: <https://design-engineering.jp/>

カジュアル面談・エントリーは
こちらから！



LINE ID: @749gaovb