Engineer

遅延読み込み

初めて必要になるまで、読み込みを待つこと

どのようなものか

遅延読み込みは、最初にすべてを読み込むのでなく、必要になるまで取得を待つことです。ウェブページでは、通常、下の方にある画像や埋め込みフレームを指し、読み手がその方向へスクロールして初めて取得します。同じ考えは、クリックした後にだけ動くコードや、アプリケーションがデータベースから読む、関連レコードにも当てはまります。

HTML標準は、マークアップに、その機能を組み込んでいます。画像とiframeは、二つの値を持つ `loading` 属性を使えます。既定の `eager` は、リソースをすぐ取得します。 `lazy` は、標準の言葉では、「何らかの条件が満たされるまで、リソースの取得を延期するために使う」ものです。標準は、 `audio` と `video` にも、同じ属性を定めています。値がない場合や、つづりが間違っている場合は、 `eager` として扱うため、誤記しても、ページは従来どおり読み込みます。

条件は、要素が表示領域に近づくことです。文書ごとに、この処理専用の交差オブザーバーがあります。遅延要素が見える位置へ近づいたと報告すると、ブラウザーは読み込みを再開します。オブザーバーは、見える領域より少し外まで見るため、読み手が画像へ向かってスクロールしている間に、画像を届けられます。標準は、その余白の大きさを、ブラウザーに任せています。

ページから、同じ仕組みを直接使うこともできます。W3CのIntersection Observer仕様は、要素が表示領域や別の要素と重なり始めるときと、重ならなくなるときに、スクリプトへ非同期に通知する仕組みです。想定する用途に、コンテンツやデータを独自の方法で遅延読み込みすることも挙げています。この仕組みは、属性の対象外にも使えます。CSSで指定した背景画像、自身のスクリプトが必要な地図、読み手が末尾に来ると次のページを取得する一覧などです。

コードの場合の単位は、モジュールです。Next.jsの文書は、ルートを描画するためのJavaScriptを減らす方法として、遅延読み込みを示します。クライアントコンポーネントやライブラリーを、最初のバンドルから外し、必要なときに取得します。例は、開こうとクリックしたときに初めて読み込むダイアログです。`next/dynamic`は、Reactの`lazy()`と`Suspense`を組み合わせ、通常の`import()`呼び出しは、ライブラリーに対して同じことをします。

言葉自体は、それらより古くからあります。Martin Fowlerの業務アプリケーションパターン集にあるLazy Loadは、関連データの場所に印を残し、何かが使うときだけ、データを取得します。一つのオブジェクトをデータベースから読むときに、誰も見ない大量の関連オブジェクトまで引き出さないためです。コードが関連レコードに初めてアクセスしたときに取得するORMも、この方式です。Djangoは、既定でそうします。最初のクエリーに含まれなかったフィールドは、初めてアクセスしたときに、データベースから読みます。文書は、その名前を使わず、この動きを説明します。

遅延読み込みは、取得済みのものをコピーして、再利用のために残すキャッシュとは違います。先読みとは逆です。`<link rel="preload">`は、ページが使う可能性が高いため、前もってリソースを取得するよう、ブラウザーに伝えます。

概念を判断に生かす

「遅延読み込みとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

ページは、誰かが見るかどうかにかかわらず、要求した全バイトの代償を払います。四十枚の写真がある長い記事、地図を埋め込んだフッター、下の方に動画がある商品ページは、読み手の一部が一度もたどり着かない内容に、帯域幅、バッテリー、接続時間を使います。遅延読み込みなら、そこまで来た人だけが要求を出すため、最初の画面に必要なものが、それらと接続を取り合わずに済みます。

どれだけ早ければ十分かは、ブラウザーが決めます。標準は、読み手のスクロールの速さと、回線の状態を余白に反映し、低速や高遅延の接続では広くして、画像が見える位置に来るまでに用意することを提案します。同時に、読み手を識

別するフィンガープリントに使える情報を、余白から漏らさないよう求めています。

プライバシーの規則も組み込まれています。ブラウザーが遅延読み込みを行うのは、スクリプトが有効な場合だけです。そうしないと、サイトは、マークアップ内に画像を散らし、サーバーにどの画像がいつ要求されたかから、各読み手がどこまでスクロールしたかを、おおよそ推定できると、標準は説明します。スクリプトが無効なら、`loading="lazy"` は、そのまま通常の読み込みになります。

コードについて、Next.jsは、初回読み込みの軽減を利点に挙げます。ルートの描画前に必要なJavaScriptを減らせます。一つのタブだけで使うグラフライブラリーや、編集ボタンの先にあるリッチテキストエディターが、ほかのすべてのページを遅くする必要はありません。

ORMでの利点は、まったく読まないものにあります。Fowlerは、オブジェクトと、関連するすべてを一緒に読むと便利でも、少数しか要らないのに、大量のオブジェクトを読むことになると問題だと指摘します。Lazy Loadは、その連鎖を切ります。

実際の活用例

画像とフレームなら、この属性で処理できます。次の画像と地図は、ページの下の方にあるため、どちらも待てます。

```
<img
  src="/photos/harbour-800.jpg"
  srcset="/photos/harbour-800.jpg 800w, /photos/harbour-1600.jpg 1600w"
  sizes="auto, 100vw"
  width="1600"
  height="900"
  alt="夜明けの港に浮かぶ漁船"
  loading="lazy"
/>

<iframe
  src="/embeds/route-map.html"
  title="徒歩ルートの地図"
  width="640"
  height="360"
  loading="lazy"
></iframe>
```

`width` と `height` は、画像のデータが一切届く前に、比率を与えます。HTMLの描画規則が、二つの数値をアスペクト比にするため、ブラウザーは、正しい形の枠を確保できます。 `sizes="auto"` を使えるのは、遅延読み込みする画像だけです。実際に配置した幅に合わせて、 `srcset` から選ぶよう伝えます。遅延読み込みする画像は、取得時には、その幅を持つためです。その後の `100vw` は、幅を決められない場合の代替です。

スクリプトが必要な遅延読み込みもあります。この例は、コンテナーが、スクロールで見える位置へ近づいて初めて、地図ライブラリーを取得します。

```
const box = document.querySelector("#route-map");

const observer = new IntersectionObserver(
  async (entries) => {
    if (!entries.some((entry) => entry.isIntersecting)) return;
    observer.disconnect();
    const { drawMap } = await import("./route-map.js");
    drawMap(box);
  },
  { rootMargin: "200px 0px" },
);

observer.observe(box);
```

`rootMargin` は、この検査の表示領域を、上下それぞれ200ピクセル広げます。そのため、枠が見える少し前に、モジュールの読み込みを始めます。コールバックは、非同期で動き、一度に複数の項目を報告できます。領域の外にある要素も報告するため、 `isIntersecting` を調べます。先に監視を解除することで、読み手が何度通り過ぎても、コールバックの仕事は一度です。それまで、 `route-map.js` は、ページの一部にすらなりません。

フォントは、スクリプトなしでも、ページが使う部分だけを取得するよう分割できます。Safariの下線の不具合についての、このサイトのノートは、当時、サイトの書体Ubuntuを、ラテン文字の三つの太さと、キリル文字の三つの太さの、計六ファイルで配信していた方法を記録しています。各ファイルは、含む文字の範囲を示します。ブラウザーは、ラテン文字の本文にはラテン文字用のファイルを、キリル文字の本文にはキリル文字用のファイルを使います。そのため、ページのほかの部分が英語やデンマーク語で書かれていても、キリル文字で書いた言語名の表示に、キリル文字用のファイルが必要になることがあります。ノート自身は、遅延読み込みとは呼びませんが、必要に応じて選ぶ同じ仕組みにより、明示的に先読みしない限り、使わないファイルは、ページのフォント要求に含まれません。

どのページも、すぐ必要とするファイルには、反対の判断をします。スタイルシートのバンドルを縮めた実績は、スタイルシートにbase64で埋めたフォントを外した理由を説明します。埋め込んだのは、最初の描画を速くするためでしたが、先読みする独立したファイルでも、同じ利点を得られます。一方、埋め込んだコピーは、必要かどうかに関係なく、どのページにも重さを加えました。その段階では、代わりに三つの太さを先読みしていました。三つ目を加えたのは、全部そろえたいという感覚ではなく、実際の利用環境で得たデータだったと、実績は述べます。遅延読み込みと先読みは、一つの判断の裏表です。最初の画面が必要とするものは早く届け、残りは順番を待ちます。

問題が起きるところ

最初の画像が、最後に届きます。 遅延読み込みする画像が待つのは、ネットワークだけではありません。取得を再開するのは、ブラウザーのオブザーバーが、表示領域に近いと報告してからです。オブザーバーの更新は、フレームを描画する処理の中で、スタイルと配置を計算した後に行います。ページの下の方の写真なら、その待ち時間に害はありません。一番上の大きな画像では、すべての訪問者が見る写真を遅らせます。携帯電話でもデスクトップでも、最初の画面に出る可能性があるものは、`loading` を省くか、`eager` にします。`lazy` は、その下のものに使います。

画像が届くたびに、ページが動きます。 `width` も `height` もない画像は、ファイルが届くまで形が分からず、届くと下にあるものを押し下げます。遅延読み込みでは、スクロール中、しばしば文の途中で起きます。すべての画像に、マークアップかCSSで寸法を与えます。`sizes="auto"` について、HTML標準は明確です。寸法がなければ、300×150ピクセルで描画される可能性が高いとし、寸法を与えるよう強く勧めます。

存在するために、スクリプトが必要な画像です。 手作りの方式は、実際のURLを `data-src` 属性へ置き、画像が近づくと、スクリプトで `src` へコピーします。そのスクリプトを読み込めない場合、エラーになる場合、実行を阻止される場合は、画像が一度も現れません。スクリプトを動かさずにマークアップを読むものには、取得元のない画像に見えます。組み込みの属性は、この場合、逆の動作になります。スクリプトが無効なら、標準は、ブラウザーに、画像をすぐ読み込ませます。画像とフレームには `loading="lazy"` を使い、属性が扱えないものだけに、スクリプトを使います。

どの訪問でも必要なものを、遅らせます。 遅延読み込みが有効なのは、遅らせたものの多くを、一度も使わない場合です。全部使うなら、一回の取得を、多数に分けるだけです。一覧の各項目に関連レコードを読むループは、一件ごとに独

立したクエリーを送ります。Djangoの文書は、既定では、一覧に一つと、各項目に一つのクエリーになり、N+1クエリー問題と呼ぶと説明します。各部品が現れるたび、自身のデータを要求するページも、ネットワーク越しに同じことをします。必ず必要なものは、一覧と一緒に、一つのクエリーか要求で読みます。遅延読み込みは、使わずに済む可能性が高いものに使います。

参考資料

- HTML標準：遅延読み込み属性 — `lazy` と `eager` のキーワード、ブラウザー自身の交差オブザーバー、スクリプト無効時に遅延読み込みを止める規則です。
- Intersection Observer — オブザーバーの基になる、W3Cの作業草案です。ルート、余白、しきい値、コールバックを待ち行列に入れる方法です。
- クライアントコンポーネントとライブラリーを、遅延読み込みする方法 — `next/dynamic` 、 `React.lazy()` と `Suspense` の組み合わせ、サーバー描画の省略、 `import()` でのライブラリーの読み込みです。
- Lazy Load — Patterns of Enterprise Application Architectureの、Martin Fowlerのパターンです。四つの種類を説明します。
- 取得モード — 最初のクエリーで読まなかったフィールドを、Djangoが取得する方法と、その取得をまとめるか止める、各モードです。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「遅延読み込みとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/lazy-loading/