

HTTP/2時代の新しいユースケース

本章ではブラウザとサーバー間での通信ではなく検索エンジンやソーシャルメディアが理解するプロトコルを中心に紹介します。

レスポンシブデザイン

小さなスクリーンのユーザーに適したレイアウト、あるいはタブレット端末の横断方向それぞれに最適なレイアウトなど、ユースケースごとに適切な表現を可能にするのが**レスポンシブデザイン**。端末ごとに、画面の解像度は違う。しかしそれぞれの端末にあった解像度に合わせた共通解像度として**論理解像度(論理上のスクリーンサイズ)**がある。これを**CSSピクセル**という。

```
モバイルシミュレーションモードでのCSSピクセル
> window.innerWidth
412
> window.innerHeight
737
> window.devicePixelRatio
3.5
```

CSSピクセルは幅412ピクセルで、高さが732ピクセルある。ピクセルデバイスレシオは3.5。乗算すると\$2562\times 1442\$ピクセルで実際の物理解像度の\$2560\times 1440\$と一致する。

デフォルトのモバイル向けブラウザは、PC専用に作られたサイトも、モバイル向けのサイトもなるべくユーザーに負担なく見せようとレンダリングしながら適切なスクリーンの拡大縮小比率を決定しようと頑張る。

```
<meta name="viewport" content="width=device-width", initial-scale=1.0">
```

画面サイズはOKだが、幅400ピクセルの画像が表示されるとジャギーが目立ってしまう。
srcset属性を使って複数の画像ソースを指定して、ブラウザが適切なファイルを選択できる。

```

```

srcset属性はピクセルデバイスレシオ以外にも、実際の表示サイズをもとにした洗濯など様々な機能を備えている。スタイルシートでも**image-set()** 関数を利用して同じことができる。

レスポンシブデザインはモバイル/デスクトップの切り替え以外にも、画面を横にしたときや、アイコンをクリックしたときなどにも柔軟に対応できる。

対応する画像ファイルの増加

近年はブラウザの対応も進んだことから新しい画像ファイルも活用できる土台が整った。
うまくファイルフォーマットを選択するだけで大きなパフォーマンスのインパクトが期待できる。

ファイルの種類	色数	可逆	アルファチャネル	アニメーション	HDR対応
JPEG	フルカラー —				
PNG	フルカラー —	✓	✓	APNGあり	
GIF	256色		1ビットのみ		
WebP	フルカラー —	可逆・非可逆両方対応	✓	✓	
AVIF	フルカラー —		✓	✓	✓
HEIF(Safariのみ)	フルカラー —		✓	✓	✓

セマンティックウェブ

セマンティックウェブはウェブで表面的な「テキスト」「文章」ではなく、「意味」を扱えるようにして、ウェブの可能性を広げようという運動のこと。

現在のセマンティックウェブが目指す世界は2つの別の方法で実現された。

1. XMLやJSONによる表現

1. 用途ごとのボキャブラリー(意味??)を解釈??
2. schema.org 803種類の型が登録されている
3. マイクロデータやJSON-LDからも参照される

2. ChatGPT

1. データ量があれば意味を人間が正規化しなくても使える
2. 大量のデータから意味を理解して応答する

マイクロデータ

マイクロデータはWHATWG、W3Cなどで議論がされてきたHTMLに埋め込み可能なセマンティック表現方法。

マイクロデータはボキャブラリーのコレクションを持っており、現在でも検索エンジンがサポートしている。

マイクロデータのサンプル(schema.orgのサンプル)

```
<div itemscope itemtype="http://schema.org/Person">
  <span itemprop="name">ジョージ・ブッシュ</span>
  <span itemprop="disambiguatingDescription">第41代アメリカ大統領</span>
  は
  <div itemprop="children" itemscope itemtype="http://schema.org/Person">
    <span itemprop="name">ジョージ・W・ブッシュ</span>
    <span itemprop="disambiguatingDescription">第43代アメリカ大統領</span>
  </div>
```

```
の父です
</div>
```

- disambiguationDescription
 - description のサブプロパティ。他の類似項目との曖昧性解消に使用される項目の短い説明。記述が曖昧性解消に役立つためには、他のプロパティ（特に名前）からの情報が必要な場合がある。

既存のHTMLの属性と衝突しないように使う属性は

- itemscope
 - itemtype
 - itemprop
- の3種類を使って表現。
- schema.orgの前身は以下をサポートしており、今の検索エンジンでも利用できるらしい
- Event
 - Organization
 - Person
 - Product
 - Review
 - Review-aggregate
 - Brenderumb
 - Offer
 - Offer-aggregate

JSON-LD

schema.orgはサンプルデータとしてマイクロデータ、JSON-LD、RDFaの3種類を掲載している。

今はGoogleはJSON-LDの使用を推奨している。

JSON-LDはHTMLではないため本文中に埋め込むことはできないが、scriptタグを使ってHTMLに記述する方法がW3Cの仕様書に書かれている。既存のコンテンツに影響なく埋め込むことができるのがでけえ。

検索エンジンはこのタグを含むページを読み込むときに、ページに付随するメタデータとして特別扱いする。

JSON-LDの例

```
<script type="application/ld+json">
  {
    "@context": "http://schema.org",
    "@type": "Organization",
    "url": "http://www.your-company-site.com",
    "contactPoint": [{
      "@type": "ContactPoint",
      "telephone": "+1-401-555-1212",
      "contactType": "customer service"
    }]
  }
</script>
```

SEO対策になるらしい。

対応しているデータフォーマット

- パンくずリスト

suumo.jp > 賃貸 > 東京都 > 新宿区

【SUUMO】新宿駅の賃貸(賃貸マンション・アパート)住宅の ...

【SUUMO(スーモ)賃貸】新宿駅(東京都)周辺でアパート・賃貸マンションを借りるならリクルートの賃貸情報サイトSUUMO(スーモ)にお任せください。全国の賃貸住宅・貸家の物件情報からさまざまな条件を指定して検索！豊富な賃貸物件 ...

-
- ウェブサイト情報
 - サイト名
 - 企業の問い合わせ窓口
 - ロゴ
 - ソーシャルメディア情報
- 創作物
 - 記事
 - 書籍
 - 教育コース
 - 音楽
 - レシピ
 - レビュー
 - テレビ番組、映画
 - ビデオ
- 商業
 - ローカルビジネス
 - イベント
 - 商品
- 科学
 - 科学データセット

RSS

セマンティックウェブの初期の取り組みではRDF(Resource Description Framework)と呼ばれるXML形式で記述する意味グラフ構造をもとに意味のデータベースを作ろうとしていた。あんま成功しなかった。

RDFの応用例として一番成功したのがRSS(RDF Site Summary)。一時期多くの人が使っていた。

ブログやウェブサイトを更新するとブログエンジンやコンテンツ管理システムが自動的に更新。

RSSファイルには新しい順に新規コンテンツの概略が格納される。

専用のリーダーが登録したRSSの更新チェックを行い未読のエントリーを集めて効率的に閲覧できるようにする。 [rdfの例](#)

RDF系以外のデータ

vCard

RFC 2425とRFC 2426で定義され、RFC 6350で統合された連絡先の交換フォーマット。

MIMEタイプはtext/vcard。HTTPを使って配信もでき、vcfの拡張子を持つファイルとして扱える。

QRコードに載せてスマートフォンに直接読み込ませることも可能。

vCardはテキストフォーマット。BEGIN:VCARDとEND:VCARDで挟まれたテキストブロックの中に1行で1項目のプロパティが並ぶ。

vCardのサンプル

BEGIN:VCARD

VERSION:3.0

ADR:;;四谷坂町 1 2 - 2 2 ;新宿区;東京都;160-0002;日本

N:渋谷;よしき;;;

TEL:(+81) 3-3356-5227

END:VCARD

iCalendar

RFC 2445で定義されRFC 5545で最新版が公開。スケジュールやカレンダー、TODOの交換用フォーマット。

MIMEタイプはtext/calendar。HTTPでの配信や.ics拡張子のファイルとしても使える。QRコードに載せてスマートフォンに直接読み込ませることも可能。

BEGIN:VCALENDARとEND:VCALENDARの中に、BEGIN:VEVENTとEND:VEVENTなどの構成要素が含まれた複合フォーマット。

iCalendarのサンプル

BEGIN:VCALENDAR

VERSION:2.0

BEGIN:VEVENT

CREATED:20190712T010442Z

LAST-MODIFIED:20190712T010442Z

DTSTART:20190712T000000Z

DTEND:20190712T010000Z

SUMMARY:Goconference'19 summer in Fukuoka

DESCRIPTION: Go and Enjoy Fukuoka!

LOCATION:Fukuoka Growth Next

URL:https://fukuoka.gocon.jp/jp/

END:VEVENT

END:VCALENDAR

コンテンツを返すサーバーがあるとき、そのURLをGoogleカレンダーに登録することで自動登録、更新ができる。

オープングラフプロトコル

ソーシャルネットワークで使われるメタデータ。オープングラフプロトコルが設定されているウェブサイトのURLをSNSに張り付けると記事の一部が引用され画像も表示される。SNS以外にも普通のウェブサイトでもオープングラフプロトコルに対応している。xmlns:og="<http://ogp.me/ns#>"で名前空間を定義。

オープングラフプロトコルの例

```
<html xmlns:og="http://ogp.me/ns#">
```

SNS向けのメタタグ

- og:title
 - タイトル
- og:type
 - 種類
- og:url
 - URL
- og:image
 - 画像
- og:description
 - 引用ページに張り付けるテキスト

og:description以外は必須。

ほかにも

- og:audio
 - 音声ファイル
- og:determiner
 - a, theなどの冠詞
- og:locale
 - ページのコンテンツが対象としている言語
- og:locale:alternate
 - このページが提供するほかの言語
- og:site_name
 - サイト名
- og:video
 - 動画ファイル

[オープングラフプロトコルの公式ページ](#)により細かく書いてある。

QRコード

- デンソーウェーブが1994年に開発。
- 数値は7000桁以上まで対応、印刷面積が小さい。高密度。
- 読み出しが高速。
- 多少の欠損があっても復元可能
- 数字最大限詰め込むモード
- 日本語詰め込むモード(1800文字)
- 英字記号モード(4296文字)
 - 英語の小文字が使えない
 - 空白、\$、%、*、+、-、.、/、:、使える
 - ?、&、#、使えない
- バイナリモード
 - 一般的

- 2953バイト
- 誤り訂正レベル4段階
- レベル4だと1273バイト

QRコードとスキーム

Googleが提供しているQRコードの作成、読み取りライブラリZxingデファクトスタンダード。いろいろなデータに対応している。

- URL
- メールアドレス
- 電話、SMS、MMS、FaceTime
- Wi-Fiネットワーク設定
- vCardとiCalendar
- TOTPの秘密鍵
 - 鍵管理アプリケーションに秘密鍵が登録される

モバイルアプリケーションにより多様化するブラウズ環境

ウェブサイトとアプリケーションをシームレスにつなげる昨日はiOSやAndroidで整備されている。URLのような女王をもとにアプリケーションの特定の機能を呼び出せる仕組みがOSに加わり、ウェブからアプリケーションへのリンクのような形で動作する。

このような仕組みを**DeepLink**と呼ぶDeepLinkを使えば、ブラウザやSNSの他のアプリをアプリケーションから直接起動できる。ウェブビューを利用するアプリケーション側で頑張ってインテグレーションしなければならない。

iOSのDeep Link

iOSではUniversal Linkで実装される。ウェブサイトのオーナー向け機能でウェブサイト側にアプリケーションの情報が書かれたJSONファイルや、署名のファイルを配置する必要がある。

それ以外にも、**カスタムスキーム**というものがあり、line://というスキームでLINEのアプリケーションを起動する方法。ウェブサイト上で単にリンクを配置するだけでよい。

AndroidのDeep Link

Androidでは**internetフィルター**という機構を使ってDeep Linkを実装する。

アプリケーションがフィルタの条件にマッチした場合にそのアプリケーションを起動する。

App indexingも同様に、サーバー側にファイルを置いていろいろ設定すると、Googleの検索結果からアプリケーションを直接起動することができる。

DeepLinkの例：X(Twitter)のスマートフォンアプリを起動

XのアプリケーションはiOS向けにはUniversal Link、Android向けにはinternetフィルタが設定されている。

```
https://twitter.com/intent/tweet?url=<URL>&text=<本文>&via=<他者のXのID>&related=<投稿後にフォローされたいXのID>&hashtags=<カンマ区切りのハッシュタグ>
```

なお、このURLは通常の投稿フォームを開くウェブページとしても動作する。

Xのアプリがインストールされてなかったら、ウェブページが表示される。

HTTPライブストリーミング(HLS)による動画のストリーミング再生

HTTPライブストリーミング(HLS)はAppleが2009年に提唱した動画のストリーミング再生の仕組み。

2017年にRFC 8216で標準化された。モバイル向けにSafari、Chrome、デスクトップのSafari、Edgeでサポートされている。

- カンファレンスのリアルタイム中継などのストリーミング
- 通信回線のビットレートに合わせて適切な解像度の動画が選択できる
- 字幕スーパー、音声の切り替えも対応
- MPEG2-TSとFragmented MPEG-4を使用
- H.264、H.265、AAC、MP3、AC-3をサポート
 - 一部でAVIのサポート

HLSのビデオタグ

HLSを使うHTMLを紹介。ブラウザごとに、使用できる動画のフォーマットが異なるため、<video>では複数のソースを記述することができる。

```
<video class="video center" controls="" autoplay data-id="1251">
  <source src="http://devstreaming.apple.com/videos/.../hls_vod_mvp.m3u8"/>
  <source src="/videos/images/ogg_bumper_no_tv.ogv" type='video/ogg'>
</video>
```

マスターの.m3u8ファイル

.m3u8ファイルはファイル情報を含むテキストファイル。UTF-8のプレイリスト(m3u)。

```
#EXTM3U
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-TARGETDURATION:10
#EXTINF:10,
file-640-00001.ts
#EXTINF:10,
file-640-00002.ts
#EXTINF:10,
file-640-00003.ts
#ZEN-TOTAL-DURATION:30.02999
#ZEN-AVERAGE-BANDWIDTH:1147001
#ZEN-MAXIMUM-BANDWIDTH:1340365
#EXT-X-ENDLIST
```

字幕の.m3u8ファイル

字幕ファイルは60秒ごとにぶつ切りされたファイル

WEBVTTという形式の字幕用スクリプトで書かれる。

WLS専用のフォーマットではなく、W3Cの定義するウェブ標準の一つ。

動画ファイル

.m3u8ファイルに書かれた順序でクライアントは動画をロードする。動画を見返せる機能や、ライブだけなどの設定もできる。

HLSのメリットデメリット

メリット

- HTTPを利用している点
- サーバー側でMIMEタイプとキャッシュ生存期間を設定するだけ

デメリット

- ストリーミングと名乗っているが、実態はプログレッシブダウンロード方式になっている
 - チャンクごとにダウンロードが完了しないと再生が始められない
 - 遅延する
- 12秒のチャンクで「12秒 \times 2+エンコード時間」だけ遅延が発生する(だいたい30秒)
- サポートされていない環境が多い
 - とくにデスクトップ版でのサポートがされていない
 - Chromeのデスクトップ版
 - サードパーティのドライバを使う必要がある
- コンテンツ保護(DRM)に制約がある

MPEG-DASHによる動画ストリーミング再生

HTTPのライブストリーミングはAppleが提唱した方式だが、ほかのブラウザベンダーはより幅広く、共通化できる仕組みとして、MPEGコンソーシアムによる標準化を進めようとした。

2013年にISO/IEC 23001-6として規格化された。

DASHはDynamic Adaptive Streaming over HTTP の頭文字をとった名前で、HTTPを使い、動画に適切なビットレートでストリーミングされた方式のこと。MPEG-DASHはオープンな規格という以外はHLSと同じ。

- コンテナフォーマットとして、ISOベースメディアファイルフォーマットMPEG4 Part 12/SBMFF、MPEG2-TS、WebMに対応
- コーデックに関しては主に、H.264、H265、VP8、VP9が使われる
- HLSも最初にビットレートを選択するか、MPEG-DASHは回線に合わせて動的に素早くビットレートを切り替える
- マルチアングルや、広告にも対応

video.jsというライブラリが提供されている。

MPEG-DASHとHLSの再生方法の違い

- HTTPを使ったプログレッシブダウンロードで動画再生というコア要素は変わらない
- HLS はHLSの.m3u8ファイルを解釈して再生するシステムをブラウザが持っていた
 - HTML上で再生するというわけではない
 - macOS、iOSのMediaPlayerフレームワークを利用することでブラウザ外で再生可能
- MPEG-DASHにもブラウザ外ビューアがある
 - JSの機能
 - JSから扱うAPI

- HTML5 Media Source Extensionsも利用
- iOSのSafari以外で機能する
- AppleはHLSに対応していないブラウザでも再生できる試みをしている
 - Media Source Extensionsを利用
 - ほぼほぼどの形式でも行ける
 - 2016年からFragment MPEG4に対応
 - どのブラウザでも使えるフォーマット
 - 今後の主流になる
- Shaka Player
 - Googleが提供
 - JSのライブラリ
 - オープンソースのビデオ再生ライブラリ

Media Presentation Description(MPD)ファイルの構造

- MPD
 - MPEG-DASHのマニフェスト
 - 拡張子 .mpd
 - 実態はXML
 - “MPDマニフェストの構造”
 - XLink昨日を使い複数のファイルに分割
 - profileにオンデマンド、ライブ配信を設定できる
 - ライブ配信の時は、動的にMPDが書き換わる
- <Period>タグ
 - HLSにはない
 - DVDのチャプタ
 - 広告の挿入時間の枠
- <AdaptionSet>メディアの構成を表現、言語、字幕、吹替情報
- <Presentation>理想の帯域ごとの実際に使うメディアへのリンク
- <Representation>帯域幅、コーデック、セグメントファイルの指定
- ほかにいろいろ、、、およそきりがない
- 実際のところ、これらのファイルを編集する機会はない、ライブラリのcommitterならありそう??
- MP4Boxを使うとMPEG-DASHのMPDファイルを生成できる
- Amazon AWSの Elastic Transcoderを使うとMPEG-DASH、HLSの両方に変換できる

CMAF(Common Media Application Format)

- Apple Microsoftが共通で策定し、ISO化されたメディアフォーマット
- MPEG-DASH、HLSと同様の Fragmented MPEG-4を使うフォーマット
 - 遅延を減らすための工夫
 - チャンク形式でメディアの取得
 - チャンクの大きさを減らすとHTTPに比べ往復時間やフィードバック送受信のオーバーヘッドが小さくなる
 - セグメントファイル内もより小さなチャンクに分かれている
 - (HLSなど)通常セグメントサイズに比例した遅延が発生する
 - セグメント内をさらに細かく区切ることで、遅延を解消

特殊機能の有効化

- プライバシー情報
 - カメラ、マイク、地理情報
 - ウェブサイトごとに、これらの機能の許可をする必要がある
 - JSのAPIで機能を追加
 - requestPermission()
 - ユーザーが人にの時に呼び出せる
 - HTTPでは、Basic認証、Digest認証以外のユーザーインタラクションは挟めない

使用例

```
Notification.requestPermission().then((permission) => {  
  if (permission === "granted") {  
    new Notification("Hi there!");  
  }  
})
```

本章まとめ

本章ではHTML5とHTTP/2時代に作られたり、実用化された様々な技術要素を紹介した。ブラウザをターゲットにしていないものも数多く取り上げた。

- レスポンシブデザインはスクリーンサイズに合わせてコンテンツを表示し、分ける仕組み
 - スクリーンサイズの大小を見るだけではなく、高解像度のスマートフォンに合わせた画像配信もできる
- セマンティックウェブから出発したJSON-LDなどの構造化データは、検索エンジンの結果を装飾するための情報として活用される
- オープングラフプロトコルではFacebook向けのメタ情報の形式が策定され、実用化されている
- XカードはX向けのメタ情報を生成する
- QRコードはHTTPのパスも内包するユニバーサルなリンクやアプリケーションに情報を伝達するインフラとして活用が広がっている
- スマートフォンのアプリケーションもまた、ブラウザと相補的にアプリケーションを組み合わせられるようにDeepLinkのような仕組みを整えた
- 動画のストリーミングはHTTPを土台にした細切れ画像をつなぎ合わせて表示するライブ配信の仕組みも作られている
- 帯域幅に応じて異なる解像度のデータを取得しに行く仕組みも実用化されている
- ウェブに限らず様々な情報の伝達手段として、QRコードが普及している
- このコード内でURLスキームのようなものを利用することにより、アプリケーションに情報を届けられる

人間がページをアクセスするだけではなく、ほかのサイトやモバイルアプリ、パスワードマネージャとの連動など、機械がページを読み取ってユーザー体験をアップさせという仕組みも増えている

次の章ではGoでHTML5、HTTP/2、HTTP/3世代の通信方式について取り上げる。