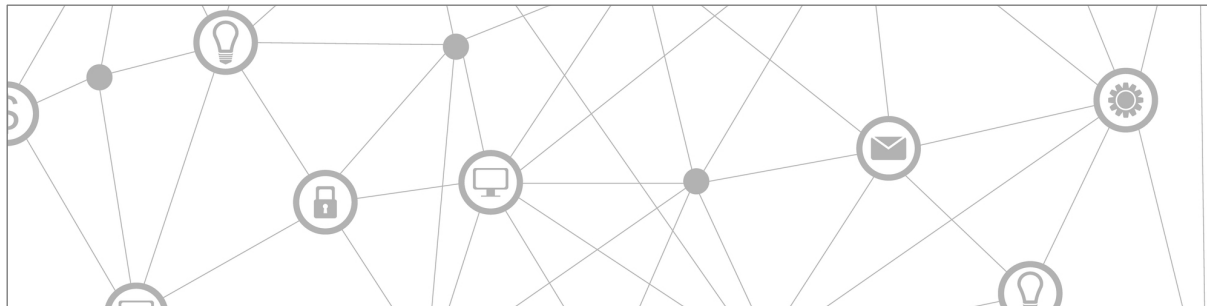


勝手に**タグ**を作っていいんです

～ W3C 標準技術によるコンテンツの自動生成 ～



用語

- **XML** …… 拡張できるマークアップ言語
- **XSLT** …… 文書変換のマークアップ言語
- **パイプライン** …… 各種媒体との入力／出力、処理
- **CMS** …… コンテンツ管理システム

目次

W3C標準技術によるコンテンツの自動生成

1. **背景** : ウェブの文書は、HTMLで記述される
2. **問題** : 不自由で不安定なHTMLコーディング
3. **対応** : より柔軟で安定したやり方はないか
4. **構成** : XML / XSLT / パイプラインの検討
5. **流れ** : W3C標準のコンテンツ生成フロー
6. **効果** : W3C標準による、ウェブサイトの統合
7. **補足** :

背景

ウェブの文書は、HTMLで記述される

- **HTML** …… ウェブ文書の**内容（意味）**を記述するマークアップ言語 [※1] [※2] [※3]
- **CSS** …… ウェブ文書の**体裁**を記述するスタイル記述言語 [※1] [※2]

HTML
<pre><html> <body> <p>ようこそ</p> </body> </html></pre>

CSS
<pre>* { font-size: 10pt; ... } p { border: 1px solid #CCCCCC; ... }</pre>

※1] HTML/CSSは、ともにW3Cにより策定された。

※2] すべてのウェブブラウザが、標準で対応している。

※3] HTMLの<意味>は、文書構造を示すものにかぎられる（表題や段落など）。

問題

不自由で不安定なHTMLコーディング

- 汎用であることの不自由さ [※1]
 - ・ ページ単位の構造（ヘッダなどを共有できない）
 - ・ 意味なしタグの散在、必要なのに提供されないタグ
- 安定しないHTMLコード
 - ・ HTML仕様のバージョンアップ
 - ・ スタイルの流行り廃り（ソリッド→リキッド、……）
 - ・ 媒体／端末ごとに違う、適切なコード
 - ・ CMS／WAFごとに違う、テンプレート記述

※1] 汎用のもの=みんなのもの=だれのものでもない

対応

より柔軟で安定したやり方はないか

- 固有の事情に合わせられる [※1] [※2]
 - ・ ページ単位、サイト単位、要素単位、管理単位、……
 - ・ 必要なタグを、必要なだけ、自在に使える
- より恒久的に使えるコード
 - ・ HTMLのバージョン／スタイルの流行、さまざまな媒体／端末／CMS／WAFなどの変遷に、影響されないコード

※1] 固有のもの=あなたのもの

※2] 特定の領域を<ドメイン>という。その領域に特化したものに、たとえばプログラミングでは<ドメイン固有言語>、知識表現では<ドメインオントロジー>などがある。

構成

自在なタグと簡潔な文書変換：XML/XSLT

- **XML** …… 拡張できるマークアップ言語。自在にタグを作成できる [※1] [※2] [※3]
- **XSLT** …… 文書変換のマークアップ言語。変換を簡潔に記述できる [※1] [※2] [※4]

```
XML
<phrase>ようこそ</phrase>
```

```
XSLT
<html>
  <body>
    <p><xsl:value-of select="phrase"/></p>
  </body>
</html>
```

- ※1] XML/XSLTは、ともにW3Cにより策定された。
- ※2] すべてのウェブブラウザ/ほとんどのパソコンやサーバが、標準で対応している。
- ※3] マークアップ言語そのものを作り出すことができる（マークアップのマークアップ）。
- ※4] テンプレート文書そのものを作り出すことができる（テンプレートのテンプレート）。

さまざまな媒体へのアクセスと処理：パイプライン

- **パイプライン** …… 各種媒体（ファイルやネットなど）との**入力／出力**、および**処理**を行う……単純な処理を**つなげる**ことで、複雑なことにも対応できる [※1] [※2]

pipeline

```
getcon x.xml | appxsl x.xsl | setcon x.htm
```

※1] パイプラインは、ほとんどのパソコンやサーバが、標準で対応している。

※2] パイプラインは、ドライバ、フィルタ、パイプからなる：

ドライバ……各種媒体との入力／出力

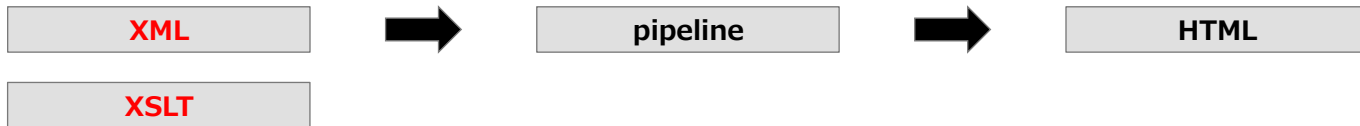
フィルタ……中間の処理

パイプ ……ドライバやフィルタ間の受け渡し

流れ

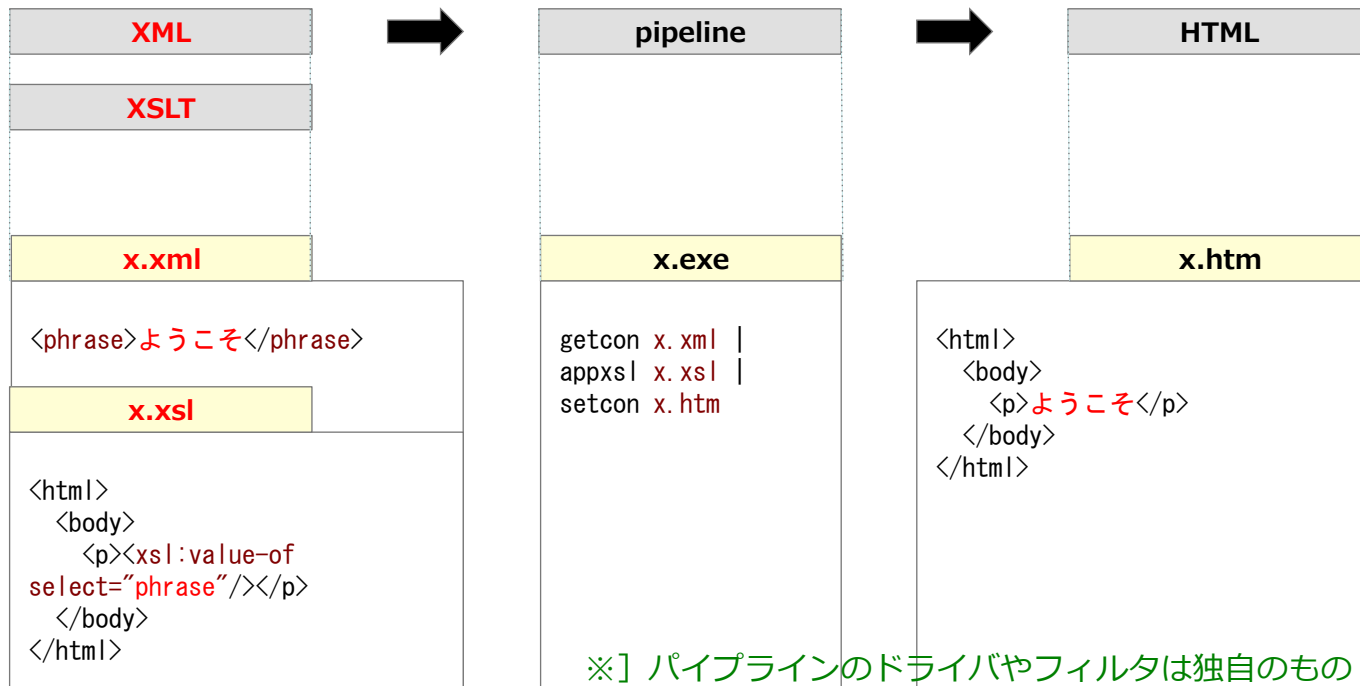
W3C標準のコンテンツ生成フロー

- XML / XSLT から、HTML を自動生成 :



W3C標準のコンテンツ生成フロー

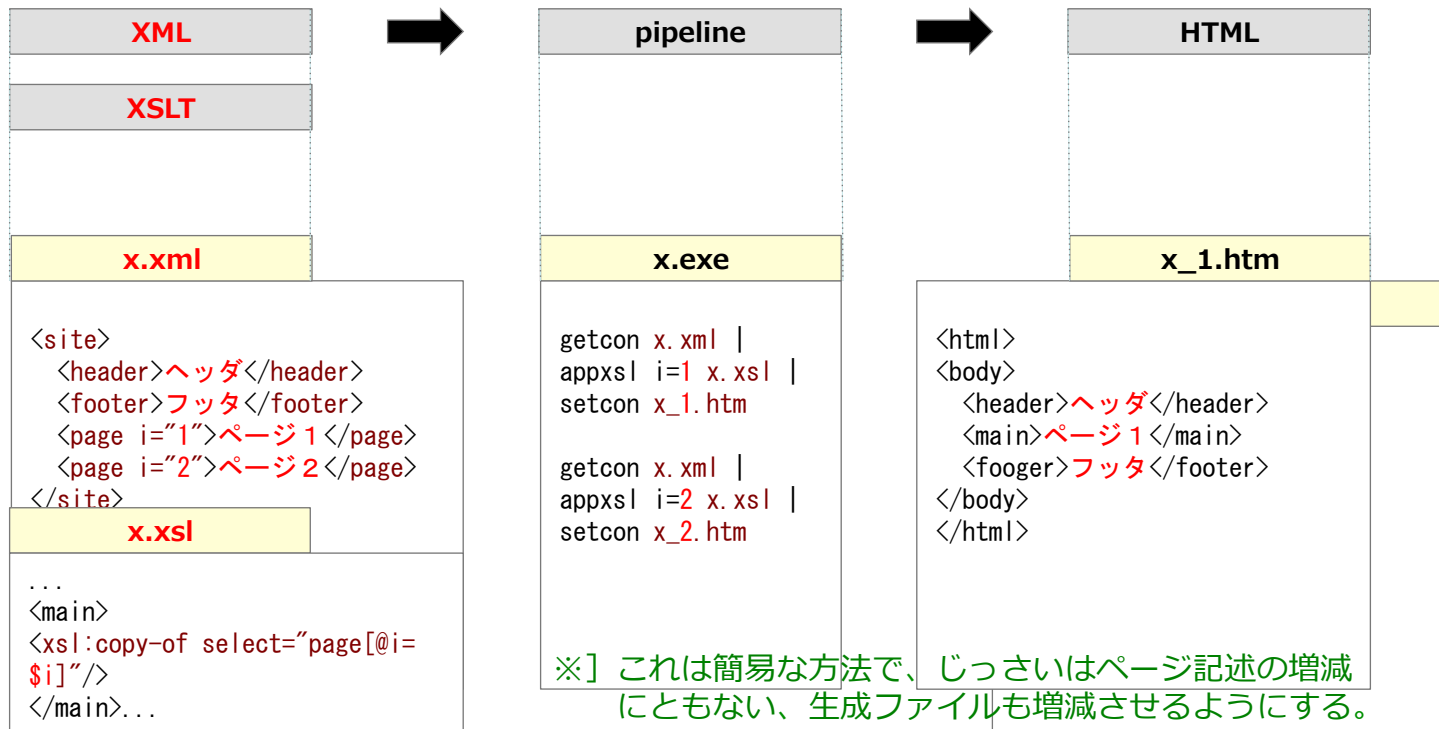
○ 例 # 1 : HTMLの生成 (フレーズの貼り付け)



※] パイプラインのドライバやフィルタは独自のものだが、ウィンドウズ/マック両OSで利用できる。

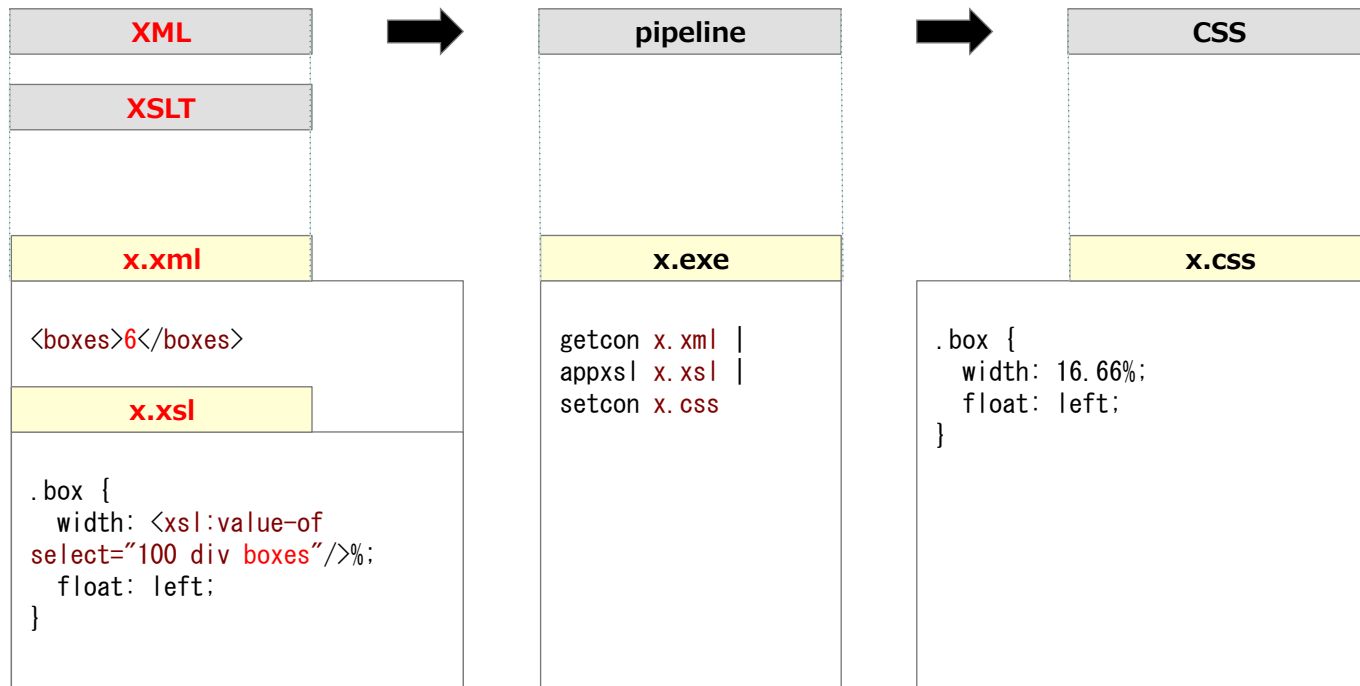
W3C標準のコンテンツ生成フロー

○ 例 # 2 : HTMLの生成 (ヘッダ/フッタ共有)



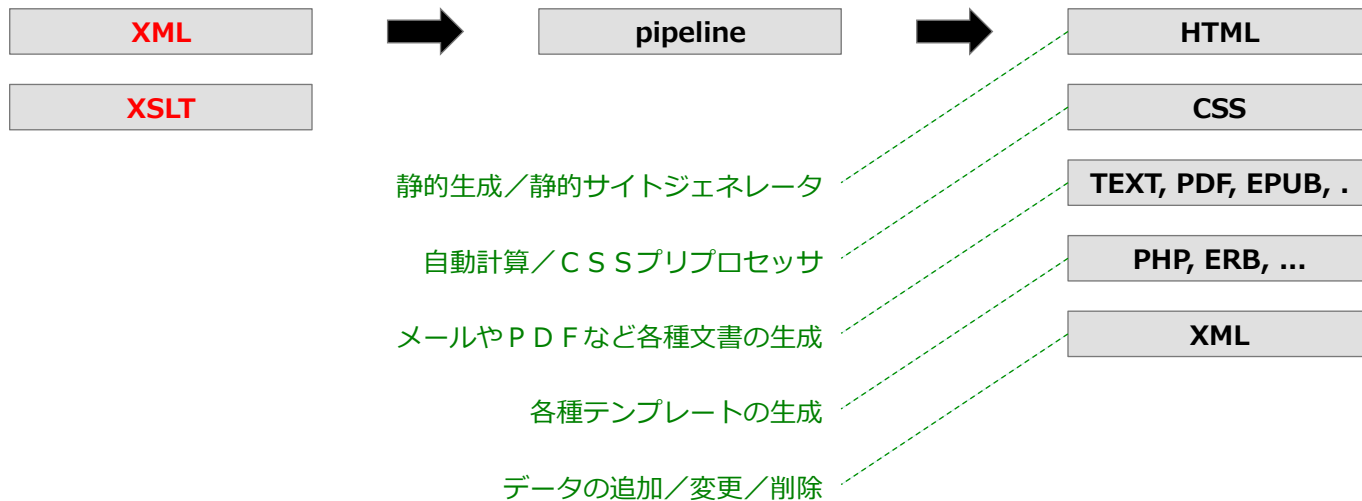
W3C標準のコンテンツ生成フロー

○ 例 # 3 : CSSの生成 (自動計算)



W3C標準のコンテンツ生成フロー

- さまざまなテキスト文書を生成：

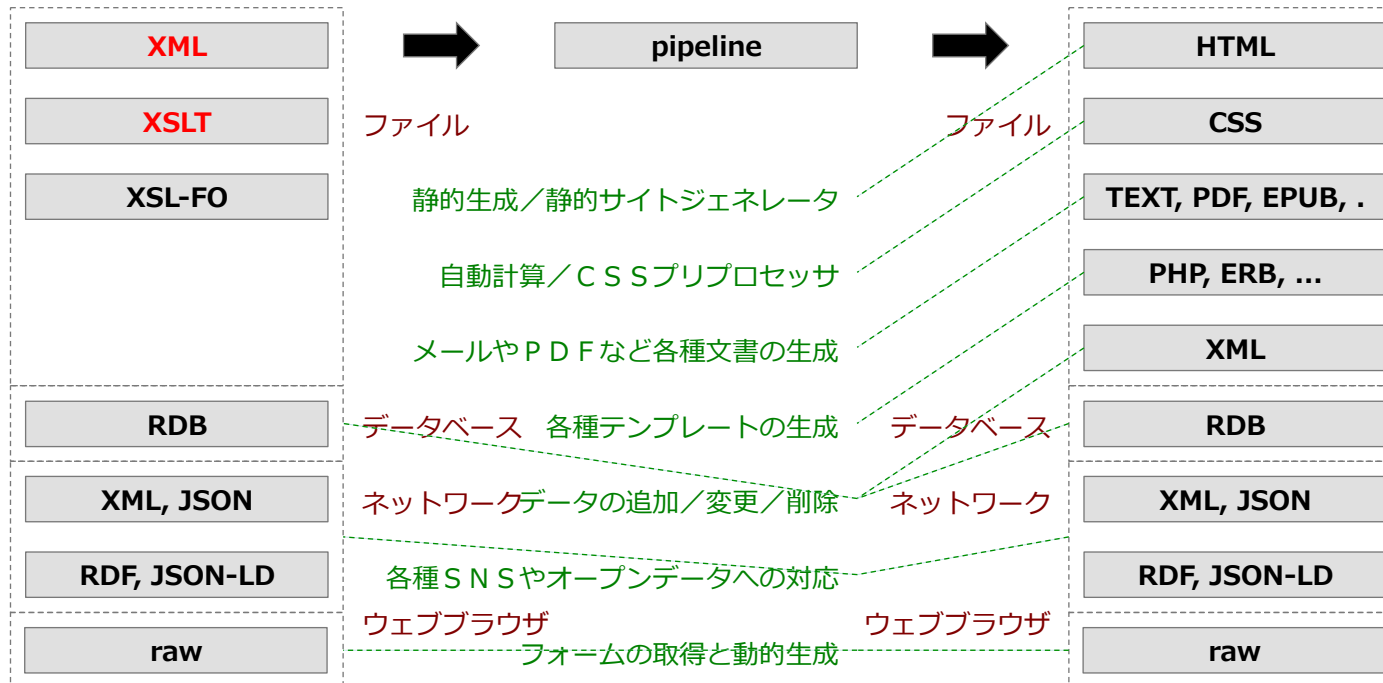


※] 静的サイトジェネレータ……パソコン側でHTMLを生成するツール

※] CSSプリプロセッサ……CSSのサイズの計算などを自動で実行するツール

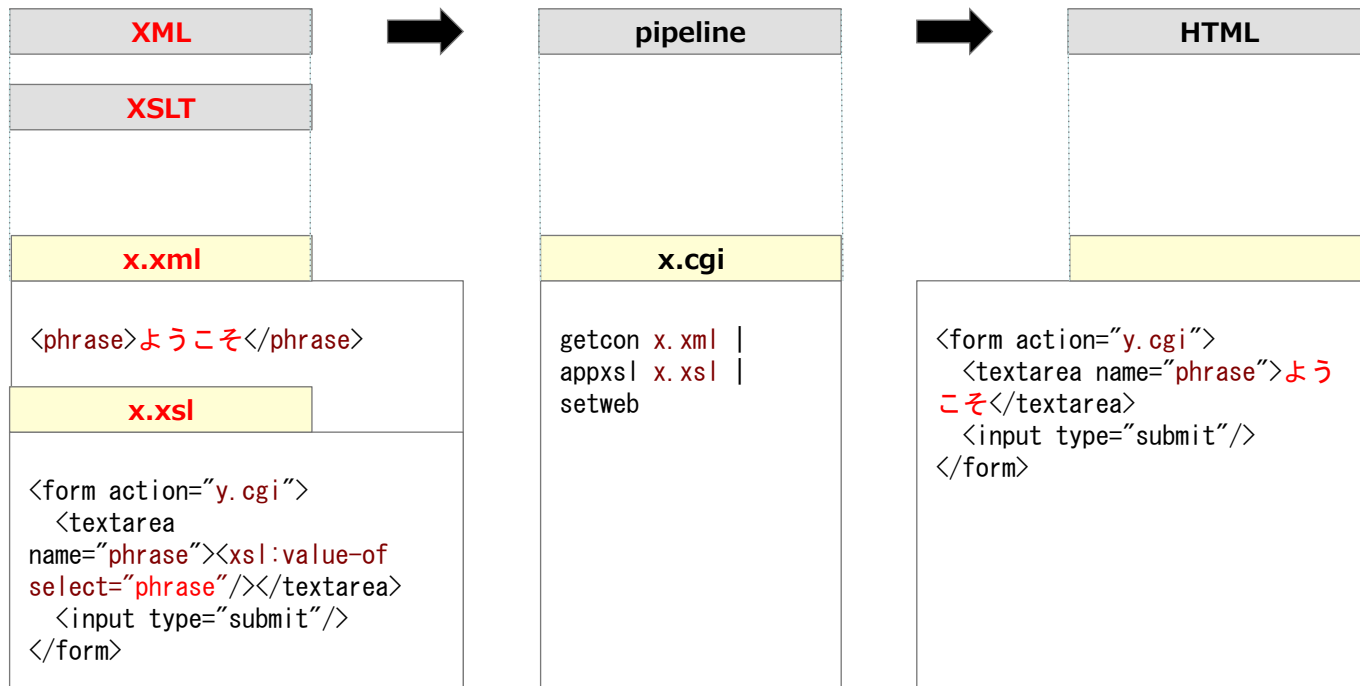
さまざまな媒体とやり取りするCMS

- やり取りできる媒体を増やせば、汎用のCMSに：



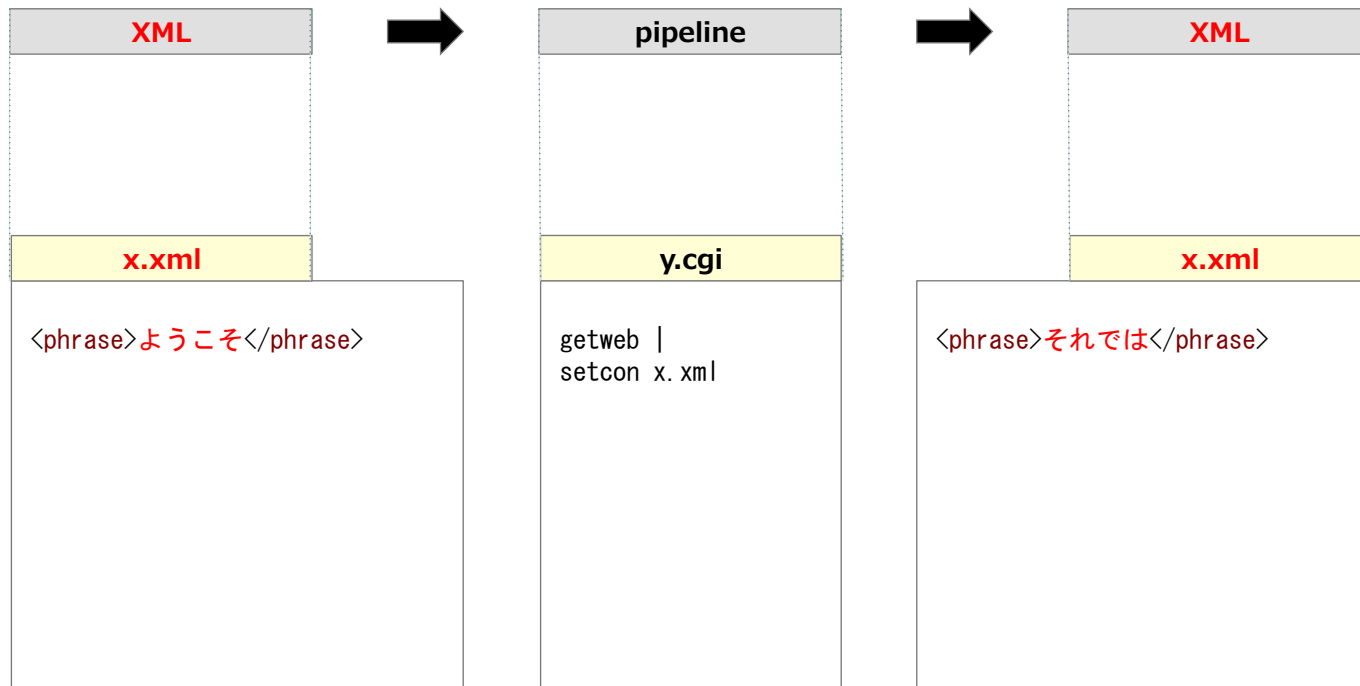
W3C標準のコンテンツ生成フロー

○ 例 # 4 : XMLファイルの取得～入力フォームの表示



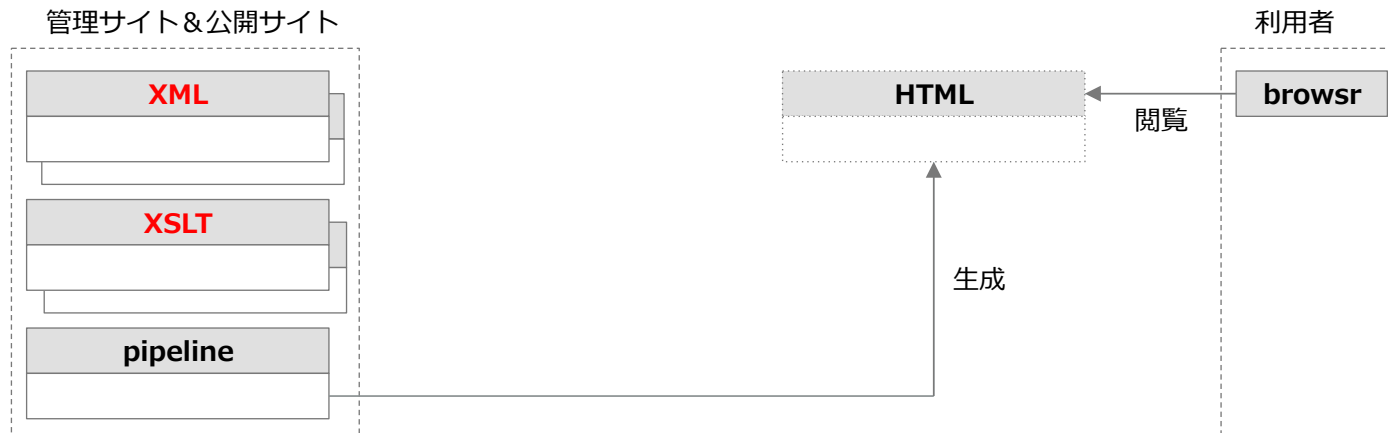
W3C標準のコンテンツ生成フロー

- 例 # 5 : 入力フォームの取得～XMLファイルの更新



コンテンツの出力先：動的生成

- **動的生成**は、生成したHTMLなどを、利用者のウェブブラウザに直接表示する： [※1] [※2]

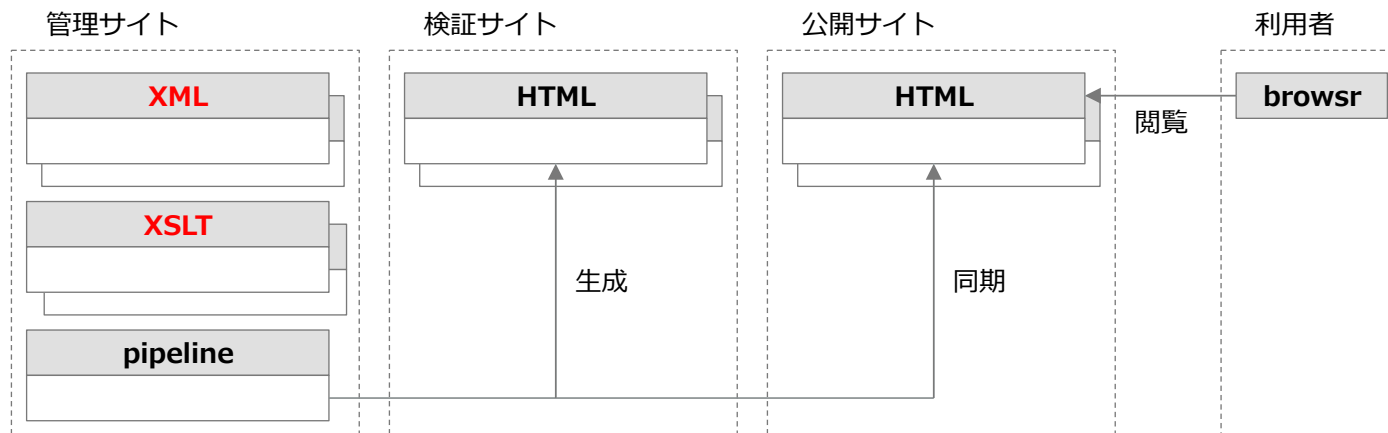


※1] XMLやXSLTが修正されたら、リアルタイムに閲覧側に反映される。

※2] これは、利用者が直接CMSのプログラムにアクセスしている、ということ。

コンテンツの出力先：静的生成

- **静的生成**は、生成したHTMLなどを、ファイルとしてサイト上に格納する： [※1] [※2] [※3]



- ※1] 通常は、いったん検証サイトに生成し、確認ののち、公開サイトに転送（同期）する。
- ※2] 公開サイトでは、かならずしもCMSを動かさなくてよい。
- ※3] 自分のパソコンにCMSを置く場合、パソコンが管理サイトと検証サイトを兼ねる。

効果

XMLによる文書の構造化

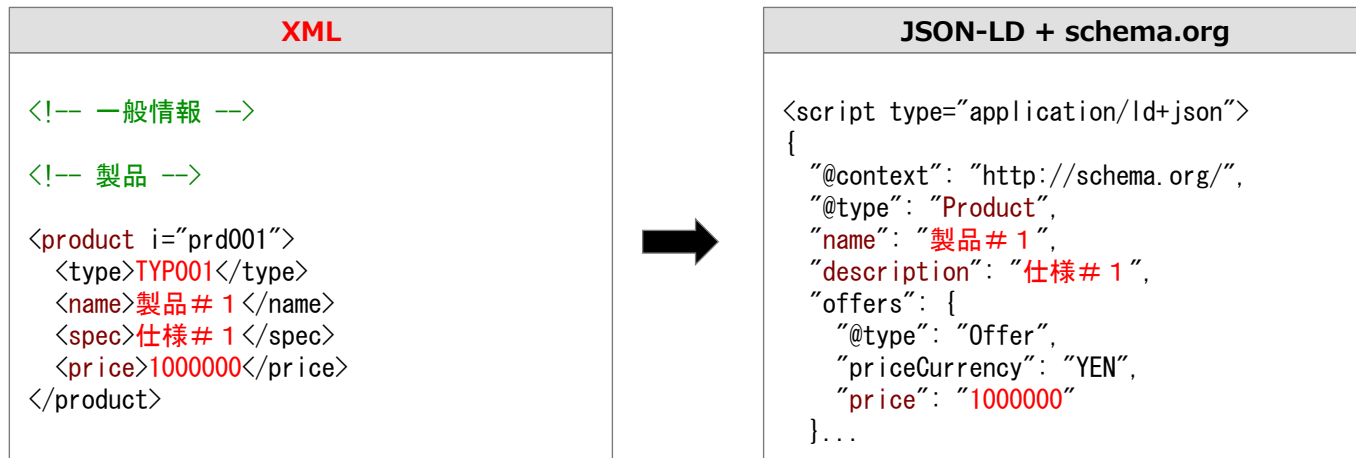
○ 文書のあらゆる要素を、さまざまなレベルで構造化：

※] これにより、コーディングを極限まで効率化できる。またサイトのデザインガイドラインを、〈単純なタギング〉により遵守させることにもつながる。

XML			
<pre><!-- 文書情報 --> <!-- サイト --> <site> <!-- 共有ヘッダ --> <header> <!-- 強調の独自タグ (z) --> <h1>株式会社<z>〇〇〇</z></h1> <!-- ページ一覧 (メニュー) --> <page_list/> </header> <!-- 共有フッタ --> <footer> <p> 著作権 / 個人情報保護 / </p> </footer></pre>	<pre><!-- 個別ページ --> <page i="doc001"> <title>ホーム</title> <section> <p>ニュース : </p> <!-- 新着一覧 (指定件数) --> <news_list m="3"/> </section> <section> <p>おすすめ : </p> <!-- 製品一覧 (推奨) --> <product_push/> </section> </page> ... <page i="doc004"> <title>新着一覧</title> <section></pre>	<pre><!-- 新着一覧 --> <news_list/> </page> </site> <!-- 一般情報 --> <!-- 新着 --> <news> <date>2020-10-20</date> <title>新着 # 1</title> <comment>内容 # 1</comment> </news> ... <!-- 支社 --> <branch i="brn001"> <name>支社 # 1</name></pre>	<pre><phone>3-0000-0000</phone> <mail>brh001@noname.local</mail> <address>東京都 〇〇〇</ address> </branch> ... <!-- 製品 --> <product i="prd001"> <type>TYP001</type> <name>製品 # 1</name> <spec>仕様 # 1</spec> <price>1000000</price> </product> ... <!-- 推奨する製品 (o=r) --> <product i="prd002" o="r"> <type>TYP002</type> <name>製品 # 2</name> <spec>仕様 # 2</spec> ...</pre>

セマンティックSEO / オープンデータへの対応

- 構造化された文書は、セマンティックSEOやオープンデータへの対応を容易にする： [※1] [※2] [※3]

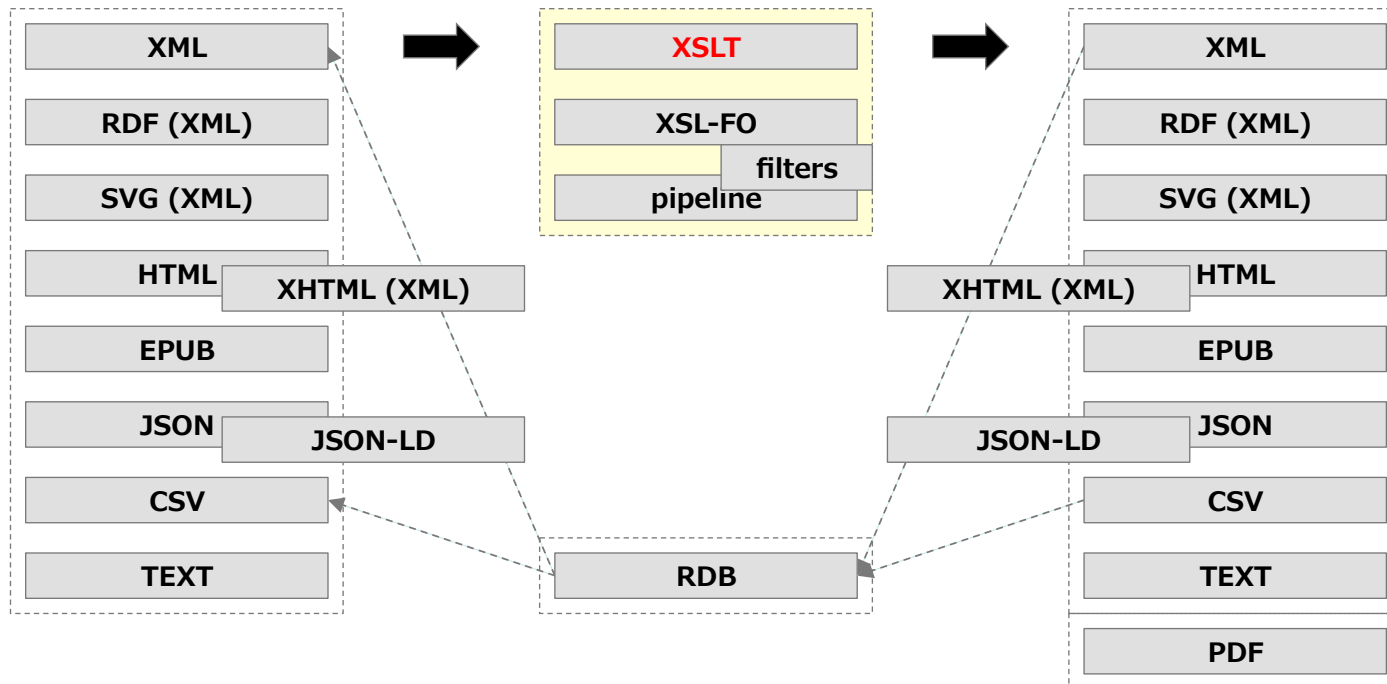


- ※1] セマンティックSEO……形式（構造化データ） + 語彙：JSON-LD + schema.org
- ※2] オープンデータ……形式（リンクトデータ） + 語彙：RDF + Dublin Core
- ※3] これらの形式や語彙は、HTML以上に変遷が大きい

効果

XSLT/フィルタによるデータ&文書変換

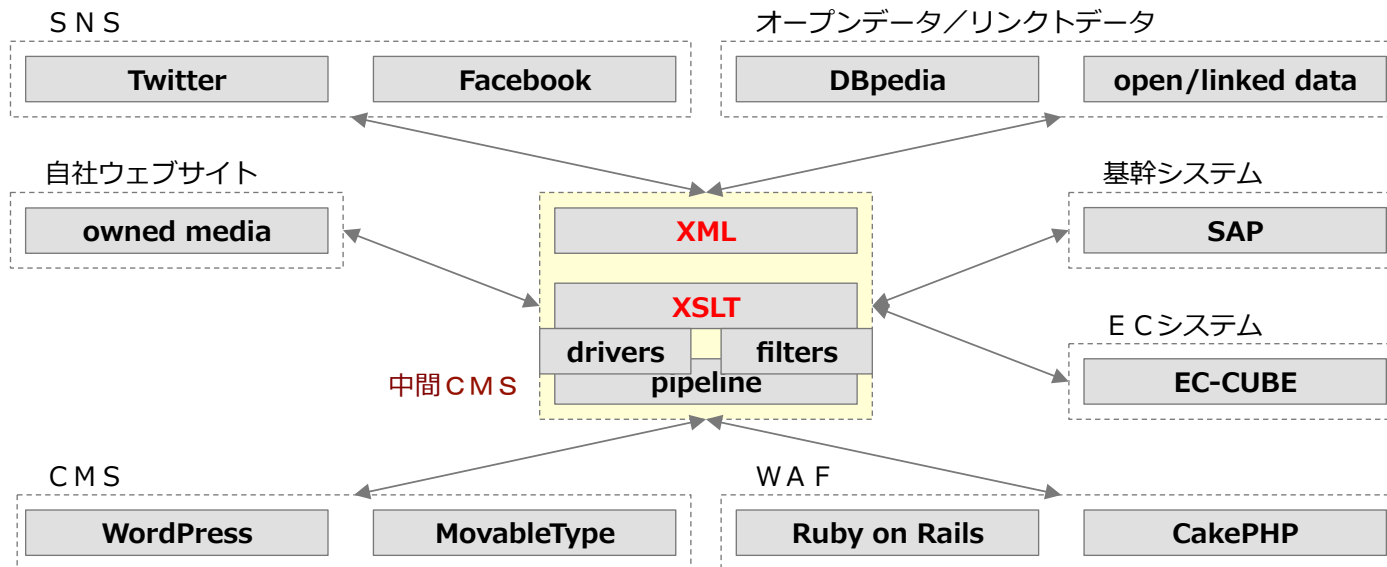
- さまざまな形式のデータと文書を、自在に変換： [※1]



※1] XSL-FOは、印刷品質の体裁（スタイル）を提供する。W3C標準。

ドライバによる各種媒体への接続

- さまざまな媒体とやり取りできるドライバにより、コンテンツ管理を仲介する（中間CMS）： [※1]



※1] 現在のコンテンツ管理は、自組織所有のウェブサイト（オウンドメディア）だけでなく、SNSなど（アードメディア）も取り込むことが重要になっている。

ウェブアプリの開発（CMS、SNS、……）

- 各種ライブラリを簡潔に開発～整理できる： [※1] [※2]

xexcms: CMSアプリ	xexsns: SNSアプリ	……
xexlng: 多言語機能	xexsec: 認証機能	……
xexedt: XMLエディタ機能: 項目の追加/削除/移動、ファイルアップロード、リッチエディタ、……		
xexweb: ウェブ出力	xexmsg: メール送信	……
xexdoc: 中間文書の定義		
docmgr: 基本ライブラリ (変換処理) : ファイル (XML)、データベース (RDB)、ウェブ、メール、…		
common: 基本ライブラリ (汎用向け) : ファイル (XML)、データベース (RDB)、ウェブ、メール、…		

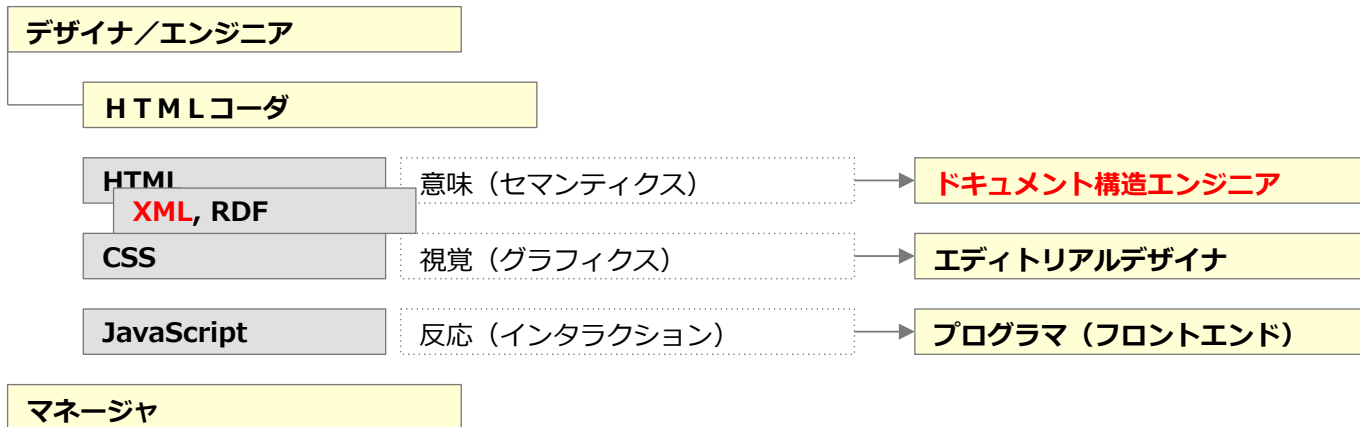
- ※1] パソコン向けの簡易なCMSから、サーバ向けの高度なCMSやSNSまで、はばひろく対応できる。
- ※2] XMLの名前空間と、XSLTのテンプレート・モジュールにより、処理とともに内容（データとタギングされた文書）や体裁（スタイル）も、同時にライブラリ化される。

W3C標準による、ウェブサイトの統合

- 独自タグと再利用（部品化）
 - ・ 大きな要素の部品化（ヘッダ、フッタ、……）
 - ・ 小さな要素の部品化（リスト、テーブル、段落、単語、文字、……）
 - ・ 文書構造と一般構造の分離（組織情報、支社情報、製品情報、サービス、イベント、……）
 - ・ ページ単位の制約からの脱却（サイト全体を1ファイルに／管理部署ごとにページを分割）
 - ・ 数値計算による効率化（CSSプリプロセッサ、等）
- さまざまな入出力
 - ・ ページの静的生成（静的サイトジェネレータふくむ）／動的生成（ワードプレス風）
 - ・ メール送信（メールフォーム、等）
 - ・ データベースの参照～更新（RDB連携、等）
 - ・ テンプレートのテンプレート（異種CMS／WAFのテンプレートを生成）
- 各種メディア統合
 - ・ セマンティックSEO（構造化データ）／オープンデータ（リンクトデータ）への対応
 - ・ シングルソース・パブリッシング（SNS、メルマガ、PDF、印刷（組版）、……）
 - ・ データ変換と移行（異種CMS／WAF／ウェブAPI／基幹システム、……）
 - ・ 独自ウェブアプリ開発（CMS、SNS、……）

補足

HTMLコードからのステップアップ



ドキュメント構造エンジニア

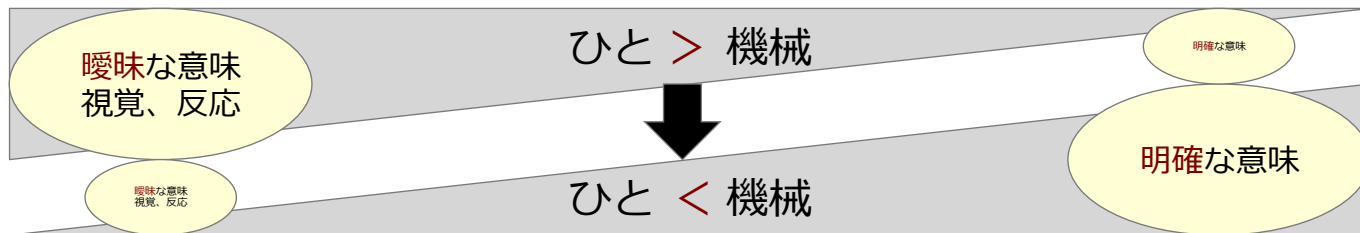
ドキュメントの意味 (セマンティクス) 構造を設計し、組織のコンテンツ管理を統合～効率化する。また機械に対して、適切な情報を提供する。

HTML +
XML, XPath, XSLT, RDB, SQL, RDF, JSON-LD, SPARQL, Dublin Core, schema.org, ...

※ 1] 建築物の設計に、外観の意匠デザイナーと内部の構造エンジニアがいるように、ウェブ文書の設計にも、体裁の情報アーキテクチャと内部の構造エンジニアを必要とする。

エージェント/AIとセマンティックSEO

- エージェントがネットに浸透するほど、ウェブ文書の意味構造の設計が重要になる：



ひとが機械（エージェント/AI）を使いこなす時代は、コンテンツの提供側も、機械に対して適切な情報を与えることが必要になる。【※1】 【※2】 【※3】

※1] 曖昧な意味 = 自然言語

※2] 明確な意味 = 人工言語（構造化データ/リンクトデータ）

※3] すでに、検索サイトで必要な情報が得られるので、個別のサイトを訪れないという状況が出始めている（店舗の基本情報（営業時間、電話番号）、……）

勝手にタグを作っているの？

いいんです