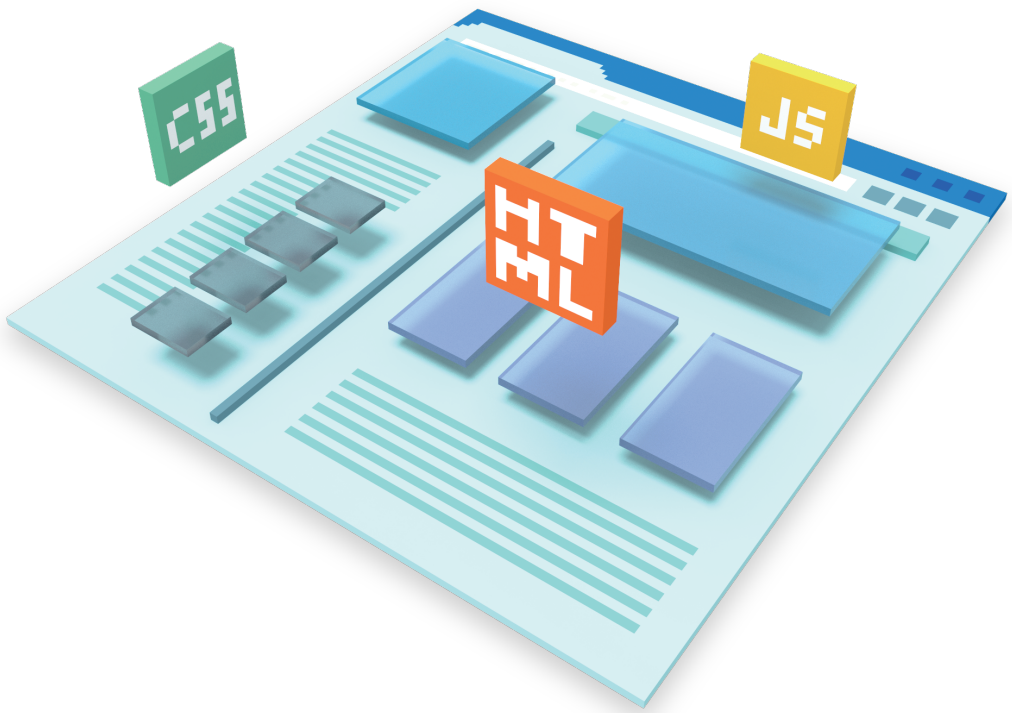
実習で身につく！

新しいWebデザインの教科書

～基礎から学べるHTML&CSSデザイン～

境 祐司 著



SCC

実習で身につく！

新しい Webデザイン の 教科書

～基礎から学べるHTML&CSSデザイン～

境 祐司 著

著者略歴

境 祐司（Yuji Sakai）

URL ■ http://design-zero.tv/　Twitter ■ http://twitter.com/commonstyle

インストラクショナルデザイナーとして講座企画、IDマネジメント、記事執筆、講演などを中心に活動。2012年5月、電子出版専門のパブリッシャーとして電子書籍のプランニング、情報設計、デリバリデザインなどを手掛ける。2014年、デジタル専門の一人出版社「Creative Edge School Books（クリエイティブエッジスクールブックス）」を立ち上げ、主にクリエイティブ系のオンライン学習コンテンツを企画・制作・販売し始める。2016年より、AI（人工知能）本格導入のためのトライアルを開始し、AIシステムやロボティクス関連の実証実験に参加。ニューラルネットワークを使った機械翻訳と画像置換処理（写真画像のイラスト表現など）を中心に小規模なテストを実施している。Adobe Museを学ぶためのエデュケーションサイト「Museアカデミー（http://design-zero.tv/muse2016/）」を運用中。Adobe Community Evangelist として主に「Adobe Sensei（Adobeの機械学習）」とクリエイティブに関する講演などをおこなっている。

著書

『Adobe Museランディングページ制作ガイド ～コード知識ゼロで作るWeb広告』（技術評論社）『Webデザイン基礎トレーニング』『Webデザインの見本帳　実例で学ぶ最新のスタイルとセオリー』（監修・執筆／MdN）など。

はじめに

1994年にインターネットが商用化され、全く新しいメディアに興味を持ったマルチメディアクリエイターやグラフィックデザイナー、プログラマーなど異業種の人たちが集まり、Web業界が形成されました。「Webデザイン」という分野も同時期に生まれ、職業「Webデザイナー」が誕生します。テクノロジーに依存した分野だけに変化のスピードが速く、この分野で仕事をしていくには常に新しいことを学習していく必要があります。

90年代は、HTMLを使ってWebサイトを構築するだけのシンプルな仕事でしたが、その後、SEOやユーザビリティ、Webアクセシビリティなど、ページ作成時に考慮しなければいけないことが増えていきます。パソコンで閲覧されることが前提だったWebページは、今やモバイルが中心となり、FacebookやTwitter、LINEなどのソーシャルメディアがコミュニケーションツールとして定着しています。今後は、AI（人工知能）などの最先端技術がWebサービスに導入され、今まで不可能だったことを次々と可能にしていきます。Webデザインのワークフローにも影響を与えることになるでしょう。

Webデザイナーやエンジニアはこれらの新しい潮流を積極的に受け入れながら、自分の仕事に活かしていくことになりますが、「木を見て森を見ず」に陥ってはいけません。部分を追いかけるのではなく常に全体を俯瞰して、必要・不必要を判断していくことが重要です。

「不易流行」という言葉があります。今も昔も変わることがない本質的なものがある一方で、絶えず変化している世界もあるわけです。変わるもの・変わらないものを同時に受け止めていくこと。そして、何より好奇心を持って変化を楽しむことです。

Webデザインは常に最新の技術に触れられる仕事であり、他の分野より一歩先の未来を見ることができます。本書をそのパスポートとして役立ていただけたら幸いです。

2018年2月

境祐司

本書の読み方

本書は全 7 章を通して、Web デザインに関するさまざまな知識を解説しています。15 時限の授業を想定して構成していますが、ご自身の学習しやすいペースで読み進めてください。

Chapter 1 Webデザインの基礎知識

1-1 インターネットとWebの歴史
1-2 Webサイトの種類
1-3 Webブラウザーの使い方

1時限

Chapter 2 HTMLの基礎

2-1 HTMLの基本文法
2-2 見出しと段落を指定
2-3 画像の種類と特性について
2-4 画像を配置する指定
2-5 他のページにリンクする指定
2-6 同じページ内のリンクを指定
2-7 表の構造と指定方法
2-8 フォームの種類と指定方法

2時限
3時限

Chapter 3 CSSの基礎

3-1 CSSの基本文法
3-2 CSSを記述する場所とCSSの検証について
3-3 CSSの「継承」と「ボックスモデル」の考え方
3-4 ブラウザーのデフォルトCSS
3-5 テキストとフォントの指定
3-6 行と行の間隔を指定
3-7 文字色と背景色の指定
3-8 背景画像の指定

4時限
5時限

Chapter 4 Webレイアウトの基礎

4-1 文書をHTMLで構造化する
4-2 HTMLのアウトライン（階層構造）を確認する
4-3 セマンティックコーディングしていこう
4-4 ページ全体のレイアウトとナビゲーションを配置する
4-5 Webレイアウトの基本「フロート処理」
4-6 ページのマージンとパディングを調整する

6時限

Chapter 5 Webレイアウトの応用

5-1 レスポンシブWeb デザインの仕組みを理解する
5-2 CSSフレームワークについて理解する
5-3 ワイヤーフレームを描く
5-4 プロトタイプの作成（1）～ページの構造化
5-5 プロトタイプの作成（2）～ページの視覚表現「固定幅レイアウト」
5-6 プロトタイプの作成（3）～コンテンツの「幅」の調整
5-7 プロトタイプの作成（4）～ブレークポイントの設定
5-8 レスポンシブWeb デザインの応用と検証

7時限
8時限
9時限
10時限

Chapter 6 拡張技術の活用

6-1 HTML5の周辺技術と活用方法
6-2 Webページに動画を配置する
6-3 WebページにGoogleマップを配置する
6-4 JavaScript について理解する
6-5 ライブラリの基礎知識と活用方法
6-6 スライドショーを組み込む
6-7 ソーシャルメディアと連携する方法

11時限
12時限
13時限

Chapter 7 Webサイトの検証と公開

7-1 FTPについて理解する
7-2 作成したWeb ページを「公開」する
7-3 Webサイトの評価・検証とは？
7-4 Web ユーザビリティとアクセシビリティ
7-5 Web サイトを総合評価する方法

14時限
15時限

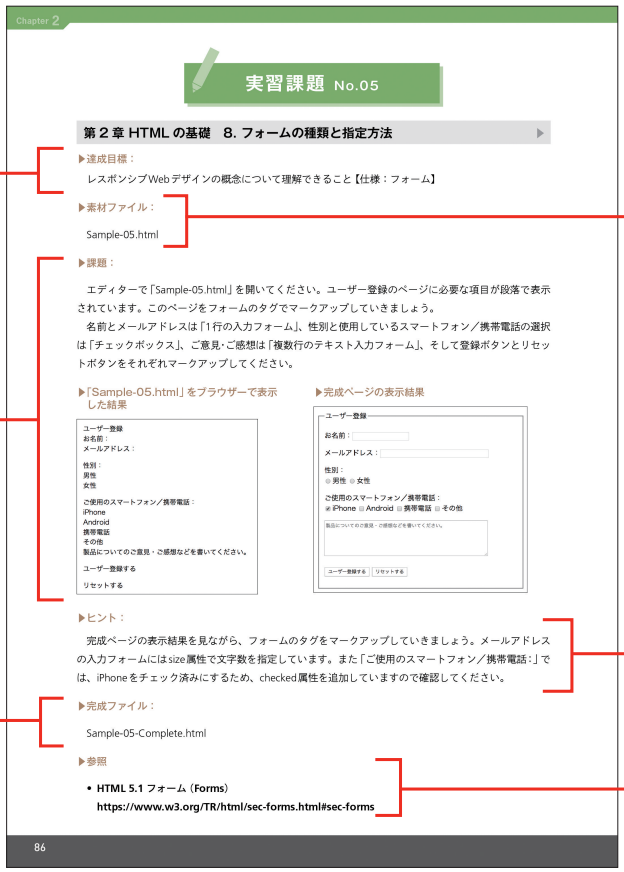
実習課題について

いくつかの節に実習課題を用意しています。ぜひ挑戦して、その節の内容を正しく理解できたか確認しましょう。

達成目標
この実習課題で習得できる内容を示します。

課題
実習課題の内容です。

完成ファイル
答え合わせに利用する完成ファイルです。自分で作成したファイルと比較してみましょう。


Chapter 2

実習課題 No.05

第 2 章 HTML の基礎　8. フォームの種類と指定方法

▶達成目標：

レスポンシブWebデザインの概念について理解できること【仕様：フォーム】

▶素材ファイル：

Sample-05.html

▶課題：

エディターで「Sample-05.html」を開いてください。ユーザー登録のページに必要な項目が段落で表示されています。このページをフォームのタグでマークアップしていきましょう。

名前とメールアドレスは「1行の入力フォーム」、性別と使用しているスマートフォン／携帯電話の選択は「チェックボックス」、ご意見・ご感想は「複数行のテキスト入力フォーム」、そして登録ボタンとリセットボタンをそれぞれマークアップしてください。

▶「Sample-05.html」をブラウザーで表示した結果

ユーザー登録
お名前：
メールアドレス：
性別：
男性
女性
ご使用のスマートフォン／携帯電話：
iPhone
Android
携帯電話
その他
製品についてのご意見・ご感想などを書いてください。
ユーザー登録する
リセットする

▶完成ページの表示結果

ユーザー登録
お名前：
メールアドレス：
性別：
男性 女性
ご使用のスマートフォン／携帯電話：
iPhone Android 携帯電話 その他
製品についてのご意見・ご感想などを書いてください。
ユーザー登録する　リセットする

▶ヒント：

完成ページの表示結果を見ながら、フォームのタグをマークアップしていきましょう。メールアドレスの入力フォームにはsize属性で文字数を指定しています。また「ご使用のスマートフォン／携帯電話：」では、iPhoneをチェック済みにするため、checked属性を追加していますので確認してください。

▶完成ファイル：

Sample-05-Complete.html

▶参照

- HTML 5.1 フォーム（Forms）
  https://www.w3.org/TR/html/sec-forms.html#sec-forms

86

素材ファイル
課題を解くために必要となるサンプルファイルです。

ヒント
課題を解くためのヒントです。

参照
課題の参考となる情報源を示します。

サンプルファイルのダウンロードについて

本書のサンプルファイルは以下のURLからダウンロードできます。

http://www.scc-kk.co.jp/scc-books/support/B-408/support.html

・サンプルファイルは本書にもとづいた学習用途のみにご利用できます。

・サンプルファイルを実行した結果については、SCC および著者は一切の責任を負いかねます。お客様の責任と判断においてご利用ください。

目次

Chapter

1

Webデザインの基礎知識

この章ではインターネットの歴史および Web デザインの基礎知識について学習していきます。

Chapter 1

01 インターネットとWebの歴史

インターネットは私たちの生活や仕事に浸透し、電気・ガス・水道のような生活者の重要なインフラになっています。この節ではインターネットが商用化された 1994 年以降の歴史を見ていきます。

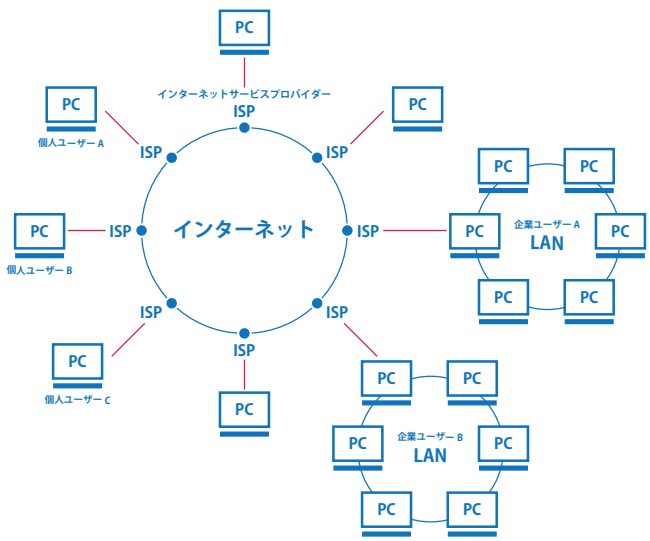
インターネットとワールド・ワイド・ウェブ

インターネットの歴史と通信の仕組み

インターネット（Internet）とは、世界中に存在する膨大な数のネットワークを「決められた通信ルール」に従って接続している「世界規模の巨大なネットワーク」のことです。私たちは、インターネットに接続した専用の業者と契約して、毎月お金を払ってインターネットを利用しています。1994年にインターネットが商用化され、接続サービス業者が登場し、一般のユーザーでも気軽にインターネットが利用できるようになりました。この接続サービスを請け負う企業を「インターネットサービスプロバイダー（Internet Service Provider）」と呼び、国内では1993年に初めて「IIJ（株式会社インターネットイニシアティブ）」が事業を開始しています。

「決められた通信ルール」とは通信プロトコルのことです。人間が共通の言葉を使ってコミュニケーションするように、通信にも共通の言葉があります。インターネットの場合は「TCP/IP」と呼ばれる通信プロトコルによってネットワークに接続し、データのやりとりなどが行われます。データの送受信には「HTTP（ハイパーテキスト転送プロトコル）」という技術が使われています。URLの頭に付く「http://」がまさにHTTPによる通信であることを表しています。

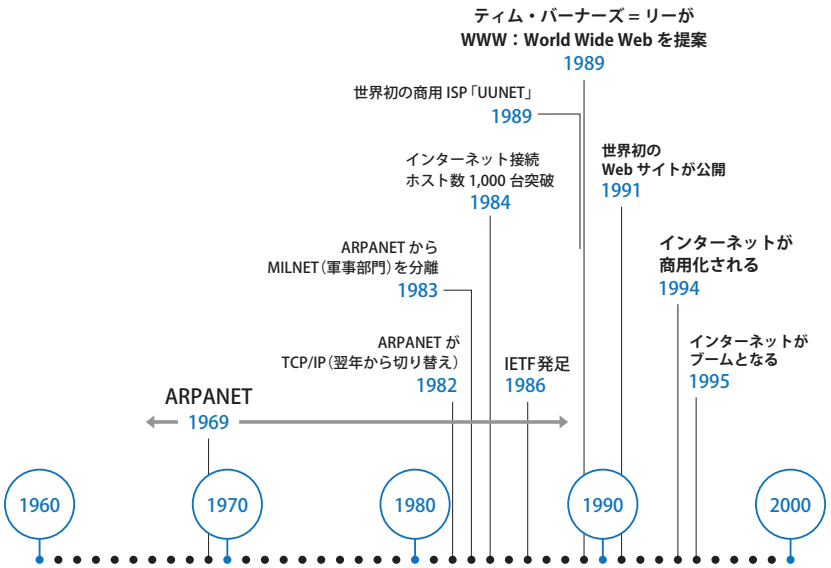
▶インターネットと「インターネットサービスプロバイダー（ISP）」



インターネットの前身は「ARPANET（アーパネット）」と呼ばれる米国の大掛かりなネットワークシステムでした。その後、学術系のネットワークとして広まり大学や研究機関で使われるようになります。現在のWebは、1989年にCERN（European Organization for Nuclear Research：欧州原子核研究機構）の計算機科学者だったティム・バーナーズ＝リーによって提案されました。1991年には世界で初めてのWebサイトが公開されています。

※現在私たちは「Web（ウェブ）」と呼んでいますが、これは「WWW（World Wide Web：ワールド・ワイド・ウェブ）」の略称です。

▶1960年代のARPANETからインターネット商用化への流れ



WWW（ワールド・ワイド・ウェブ）とHTML

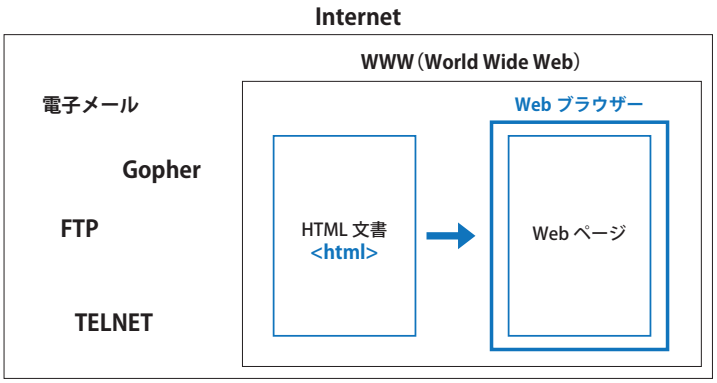
インターネットが急速に普及し始めたのは、商用化された翌年（1995年）です。MicrosoftのWindows95が発売され、国内でも「インターネット」という言葉が年末の流行語大賞にノミネートされるほど一大ムーブメントになりました。インターネットのアプリケーションの1つだったWWW（ワールド・ワイド・ウェブ）が急速に普及したことで、世界中に無数のWebサイトが誕生しました。

インターネットには電子メールやWWW、FTP、Gopher、TELNETなどのアプリケーションがあり、誰でも自由に利用できましたが、特に電子メールとWWWについては利用者が急増し、世界中に普及しました。多くの人は「インターネット＝WWW（ウェブ）」だと認識していたほどです。

WWWは「タグ付けされた文書」を「専用のビューアーソフトで閲覧する」システムです。HTML（エイチ・ティー・エム・エル）と呼ばれる言語で文章にタグを付けて「.html」という拡張子で保存をするだけで「Webページ」が出来上がります。あとは「Webブラウザー」と呼ばれるアプリケーションソフトで開けば、ページとして閲覧することができます。

ティム・バーナーズ＝リーが開発したHTMLは、シンプルなマークアップ言語で一般のユーザーでも容易に習得できるため、インターネットが商用化されてすぐに大量のWebサイトが公開されました。HTMLが簡単な仕様だったことで、Webの時代が到来したといっても過言ではないでしょう。

▶インターネットのアプリケーションの１つだったWWW（Web）



※ HTML 文書を Web ブラウザーで表示したものを「Web ページ」と呼び、Web ページの集合体を「Web サイト」と定義しています。ただし、一般的には「ホームページ」という呼称が浸透しており、同義の用語として混在していますが、本書では「Web ページ／ Web サイト」で統一しています。

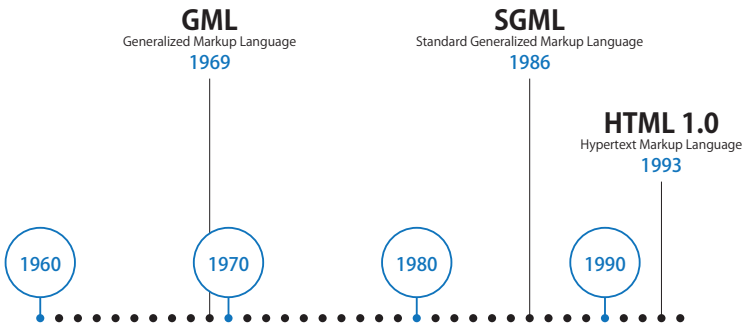
マークアップ言語の歴史

マークアップ言語の始まり

HTMLは（文書にタグを付けて構造化するための）マークアップ言語ですが、この技術は60年代に出版の分野で生まれました。1969年の「GML（Generalized Markup Language：ジェネラライズド・マークアップ・ランゲージ）」までさかのぼります。

GMLは、IBMの研究者チャールズ・ゴールドファーブ（Charles Goldfarb）、エドワード・モシャー（Edward Mosher）、レイモンド・ローリー（Raymond Lorie）らが開発したマークアップ言語です。文書の構造と表示方法を分離して、汎用的なタグ付けを可能にしたことで、複数のドキュメントを１回の処理でまとめて変更することが可能になったのです。このGMLによって膨大なドキュメントを効率よく管理できるようになりました。

▶「GML」から始まるマークアップ言語の歴史



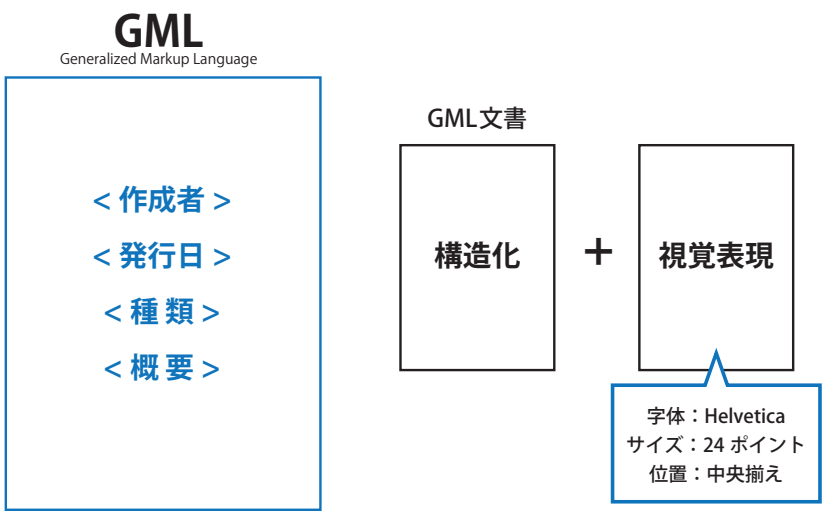
文書構造と視覚表現の分離

GMLでは文書内の見出しに対して「これは見出しである」というタグを付けて、構造上どういう意味を持っているか定義していきます。見た目のデザインについては指定しません。例えば「字体はHelvetica、サイズは24ポイント、位置は中央揃え」といった視覚的な表現は、GMLの文書から完全に分離されます。

GMLには「作成者」「発行日」「種類」「概要」などのタグがあり、数百、数千のドキュメント（技術書やマニュアルなど）がタグ付けされました。この仕組みによって、特定の分野について書かれた文書の概要を参照したり、指定した期間で一覧表示させることなどが可能になりました。

これから学習していくHTMLは、この48年前の技術がベースになっています。

▶GMLは構造と視覚表現を分離



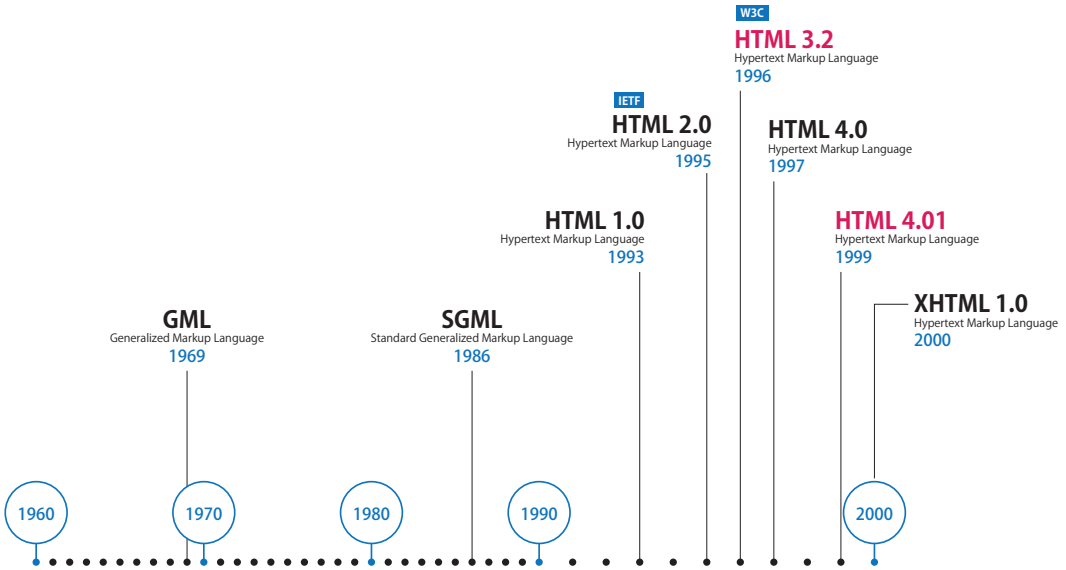
HTMLはW3Cで策定されたWebの標準技術

HTML は習得しやすいシンプルな言語

HTMLは、1986年にISO（国際標準化機構）で規格化された出版技術「SGML（Standard Generalized Markup Language）」を簡素化したマークアップ言語です。SGMLはとても複雑で難しいマークアップ言語でしたが、HTMLは必要最小限のタグに絞って徹底的にシンプルにしたことで、言語を習得するための教育コストが下がり、一般のユーザーでもちょっと勉強をすればWebページを作成することが可能になりました。HTMLが「簡単な言語だった」ことはとても重要で、もし高機能を優先して難易度の高い言語にしていたら、ここまでWebが爆発的に普及しなかったでしょう。

※ ISO（International Organization for Standardization：国際標準化機構）は、国際規格（IS：International Standard）を策定する組織で 162 の標準化団体が集まっています。

▶マークアップ言語の歴史（GMLからSGML、HTMLへ）

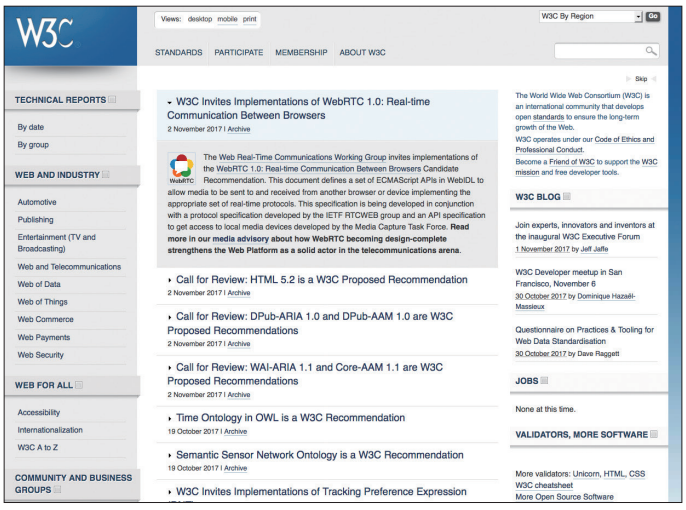


HTMLの仕様はW3Cのサイトで確認する

HTMLは、W3C（World Wide Web Consortium：ワールドワイド・ウェブ・コンソーシアム）というWebに関する標準技術を決めている非営利団体が策定しています。通称はダブリュースリーシーです。米国のMITコンピュータ科学・人工知能研究所（MIT/CSAIL）、フランスの欧州情報処理数学研究コンソーシアム（ERCIM：エルシム）、日本の慶應義塾大学SFC研究所、中国の北京航空航天大学などのW3Cホストが共同で運営しています。

HTMLの仕様書はこのW3Cのサイトですべて公開されており、誰でも自由に閲覧できるようになっています。Webブラウザーを開発している企業もこの仕様書の内容を解釈して製品に実装しています。

▶W3Cの公式サイト



1994年10月に設立された標準化団体
https://www.w3.org/

HTMLの正しい知識はW3Cの仕様書で学ぶことができますが、すべて英語で書かれており、決して読みやすい文書ではありません。現在市販されているHTML辞典やリファレンスなどは、この仕様書を日本語で分かりやすく解説したものです。1冊は購入して、いつでも参照できるようにしておきましょう。

※ HTMLのリファレンスは国語辞書などと同じように分厚く重たい本です。電子書籍でも販売されていますので、タブレットなどの大きな画面のデバイスを持っている人は電子版をお奨めします。

Web標準に準拠しよう

W3Cが策定する技術標準には「必ず従わなくてはいけない」という強制力はありません。現在のHTMLをベースにして独自のHTMLを作ったり、最先端の技術でHTMLのように動作させることも可能ですが、Web業界では「事実上の標準規格」として捉えて準拠しています。

準拠する理由は以下の通りです。

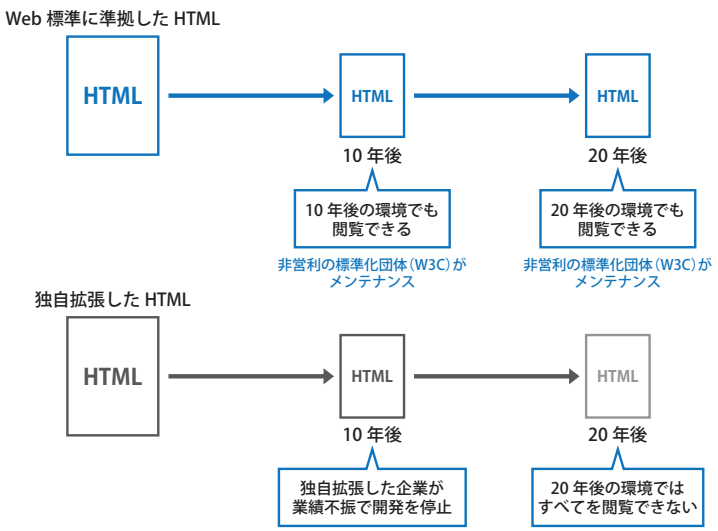
(1) 特定の環境に依存しない柔軟性の高いWebサイト構築が可能になる
(2) 技術についての知識やノウハウを多くの人たちと共有することができる
(3) 20年、30年経っても閲覧できる寿命の長いコンテンツになる

もし、HTMLを好き勝手に改変したり、独自の最先端技術などを組み込んでしまうと、互換性がない使いづらいWebサイトが増えてしまい、アプリケーションソフトのようにバージョンアップしないと、いつかは動かなくなってしまうでしょう。バージョンアップする人がいなくなったら最後です。これではWebの信頼性が低下してしまいます。

Web業界がW3Cの技術標準に準拠しているからこそ、20年前に作られた古いWebサイトも「バージョンアップすることなく」今でも閲覧することができます。現在作成しているWebサイトも標準に準拠していれば20年後も閲覧することができるでしょう。

私たちは将来の大きな利得について考えていくべきです。そうすれば、利用者にとっても、Webデザイナーにとっても利便性の高いWeb環境になっていきます。

▶Web標準に準拠したHTMLは永続性が高く、世界中の知識やノウハウを共有できる



Chapter 1
02 Webサイトの種類

Webサイトは用途や規模、デザイン、採用している技術などで分類できます。Webデザイナーにとっては「規模」によるWebサイトの特性の違いを正しく理解しておく必要があります。

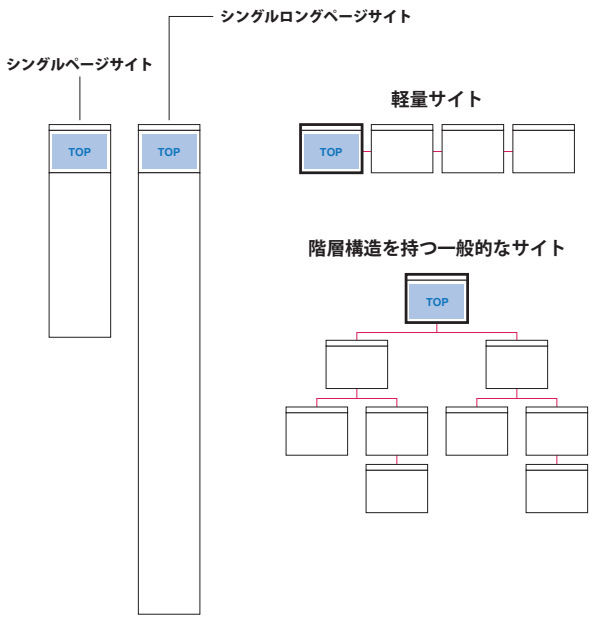
Webサイトを規模で分ける

シングルページサイト

Webサイトは規模によって複数の種類に分けることができます。1ページしかないWebサイトを「シングルページサイト」および「シングルロングページサイト」と呼びます。ここ数年で増加しており、ランディングページやプロモーションページ、キャンペーンページ、特設ページ、顧客向けの案内ページなどさまざまです。長い「巻物」のようなページを上下にスクロールさせるだけの操作しかなく、メニューが付いていないサイトもあります。

※複数のWebページの集合体を「Webサイト」と定義していますが、シングルページの場合も「ページ内リンクによって（ページ遷移するように）同様のユーザー体験が可能」なため「サイト」と呼んでいます。

▶10ページに満たない軽量サイトと一般的な階層構造型サイト



軽量サイトと一般的な階層構造を持つサイト

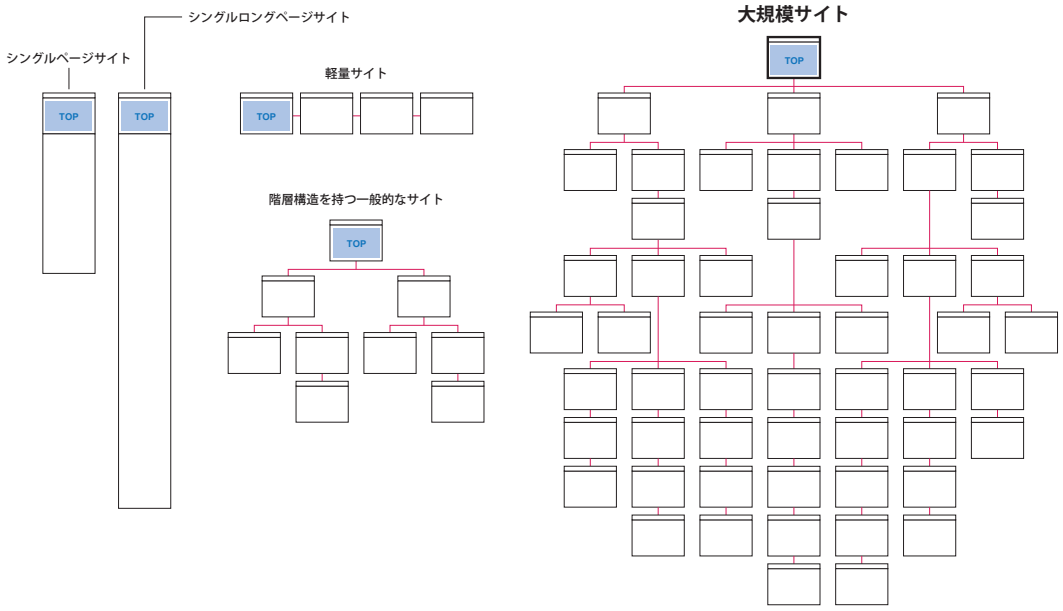
10ページに満たないWebサイトを「軽量サイト」と呼びます。中小企業のサイトや個人サイトなどは大半がこの軽量サイトです。頻繁に更新するのはブログやSNSで、Webサイトは基本情報だけのカタログのようになっています。情報発信の使い分けが進んだことで、Webサイトもその1つの媒体でしかなくなった結果だといえるでしょう。

最もスタンダードなタイプが、階層構造を持つWebサイトです。トップページがあって、第1階層に製品ページ、第2階層に各製品の専用ページ、第3階層に各製品の詳細ページといった構造で作られています。階層を持つことでサイト内が複雑になり、サイトマップと呼ばれる一覧ページを用意するなど、ユーザビリティ（使いやすさ）への配慮が重要になります。

大規模サイト

数百、数千ページ以上のWebサイトを「大規模サイト」と呼びます。大手新聞社のニュースサイトやAmazon、楽天などのショッピングサイト、大企業のサイトなどが該当します。この規模になるとWebサイト構築も大勢の人たちが携わる大きなプロジェクトとなり、長期計画の仕事として推進されます。また、Webデザイナーの役割も細分化され、ビジュアルデザイナー、インターフェイスデザイナー、UXデザイナー、コーダーなど、専門のデザイナーが担当するチーム作業になります。仕様書やスタイルガイドなどのチームで共有するドキュメントも発行されます。

▶大きなプロジェクトとして推進される大規模サイト



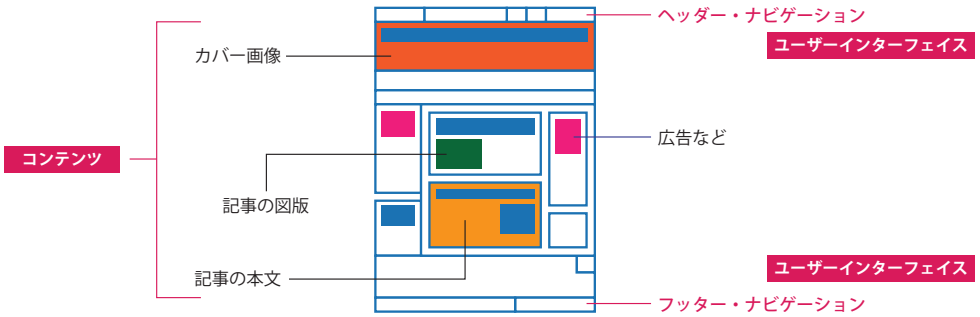
Web サイト構築と情報デザイン

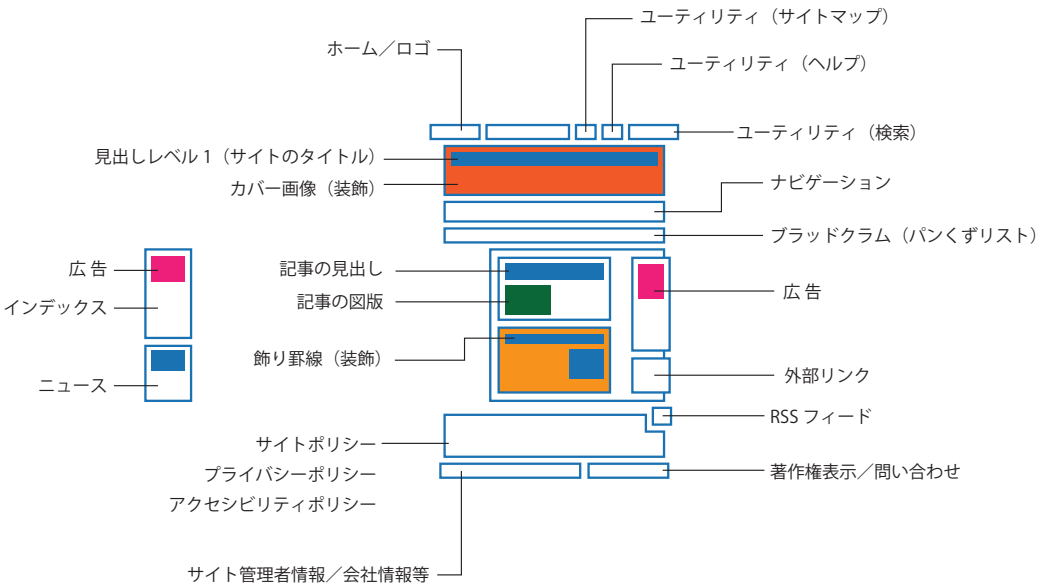
Webサイトは、情報の提供だけではなく、サイト内検索の機能でアーカイブを呼び出したり、要約をリスト表示したり、他の閲覧者と情報共有できたり、便利なユーティリティの機能も持っていますので、複数のアプリケーションの集合体として捉えることができます。紙媒体の「ページ」との大きな違いです。文字情報やイメージ情報だけではなく、ナビゲーションメニューなどのUI（ユーザーインターフェイス）も含みますので、誌面を設計するエディトリアルデザインと製品開発のプロダクトデザインを同時に扱っていくことになります。

参考にしてほしいのが「情報デザイン」という分野です。情報デザインとは、情報を分類・整理して分かりやすく伝えるための表現手段のことで、業界ではインフォメーション・アーキテクチャ（IAと略される）と呼ばれます。

Webデザイナーは「見やすく・読みやすい」魅力的なページデザインを行いつつ、ターゲットユーザーに対して「使いやすく」「アクセスしやすい」サイトを設計しなければいけません。

▶Webページにはコンテンツだけでなくインターフェイスも組み込まれるため「使いやすさ」も考慮しなくてはいけない





Webサイト制作の3つの考え方

技術やデザインを学ぶ前に理解しておこう

Webデザインは日進月歩の「テクノロジーに依存する」分野です。変化のスピードが速く、流行り廃りが短い期間で繰り返されています。Webブラウザーも頻繁にアップデートされており、最新の技術に対応していきます。このような分野で問題になるのが「古い環境をどうするか」です。世界中のすべてのユーザーが一斉に最新の環境に移行するわけではありません。古い環境を使い続けるユーザーも多いため、Webデザイナーは採用する技術を慎重に判断する必要があります。

Webデザイナーが理解しなければいけない「サイト制作に関する3つの考え方」があります。プログレッシブエンハンスメント、グレイスフル・デグラデーション、レグレッシブエンハンスメントの3つです。いきなり難しい専門用語が出てきましたが、簡潔にいうと「古い環境はどうするのか？」を判断するための指針です。「今回の仕事は、こういう制作方針で進める」と最初に決めておかないと、ユーザーからのクレームが多いWebサイトが出来上がってしまいます。

Webサイトの利用者層を見て、「このユーザー層なら新しい環境をメインにしても大丈夫」とか、「このサイトは子供からお年寄りまで幅広い層の人たちが使うので古い環境を切り捨てられない」など、サイト制作の方針を明確にしてから、作業を進めないと失敗してしまう可能性が高くなります。

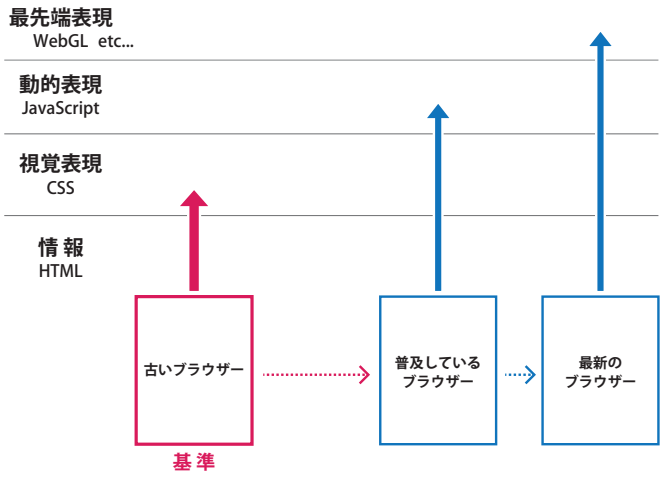
「Webサイト制作の3つの考え方」はとても重要な学習になりますので、要素技術やデザインを学ぶ前に理解しておきましょう。

プログレッシブエンハンスメント

プログレッシブエンハンスメント（Progressive Enhancement）とは、古いブラウザーでも最低限の情報を得られるように作成しておき、その後、最新のブラウザーだけを対象にして新しい技術を使っていく開発思想のことです。

漸進的（ぜんしんてき）な機能強化ということですから、古いブラウザーでも最低限の情報は提供できるように（それなりに見られるように）作成しておいて、能力の高い最新のブラウザーには新しい視覚表現をどんどん取り入れていこうという考え方です。

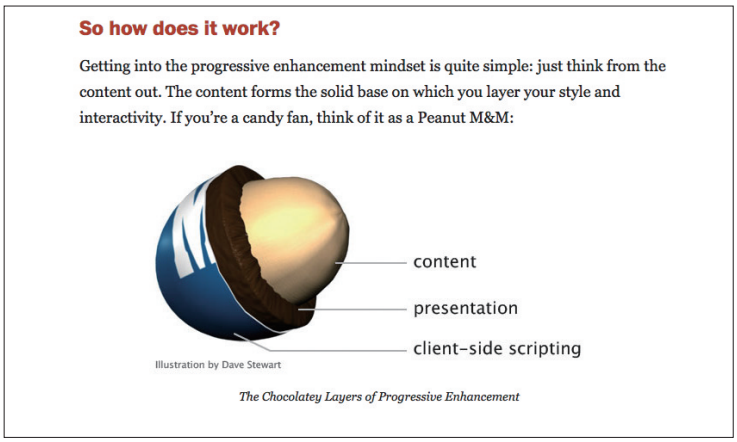
▶古い環境を切り捨てないWebサイト制作



この考え方は、2003年のSXSW（サウス・バイ・サウスウエスト）で、スティーブン・チャンピィオン（Steven Champen）とニック・フィンク（Nick Finck）によるプレゼンテーションで提唱され、その後、2008年10月7日にA List Apartで公開された「Understanding Progressive Enhancement（アンダスタンディング・プログレッシブエンハンスメント）」という記事が話題になり、世界中で実践されるようになりました。

記事を書いたのは、「Adaptive Web Design」の著者であるアーロン・グスタフソン（Aaron Gustafson）です。A List Apartの記事では、HTMLとCSS、JavaScriptの3層構造の図を掲載しています。

▶基本は情報（HTML）、次に見栄え（CSS）、最後が動的表現など（JavaScript等）。まずは古い環境に対して十分な情報（HTML）を提供できるように作り込む



A List Apart : Understanding Progressive Enhancement
http://alistapart.com/article/understandingprogressiveenhancement

グレイスフル・デグラデーション

グレイスフル・デグラデーション（Graceful Degradation）は、最新のブラウザーを対象にWebページを作成して、古いブラウザーに関しては品質を下げて対応する開発思想です。プログレッシブエンハンスメ

ントとの違いは、古いブラウザーを切り捨てないが、あくまでも現在主流の最新のブラウザーが中心ということです。

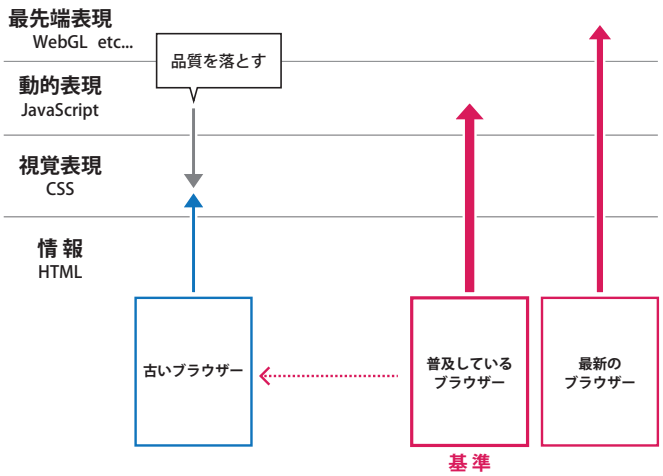
例えば、HTML5のvideo要素をサポートしていない古いブラウザーは、ビデオを再生できませんので、代替テキストや静止画を用意しておきます。ビデオは再生されないけど、静止画が表示されて意味は伝わるという最小限の対応です。

ボックスの角丸をCSSで指定した場合も、未対応の古いブラウザーは角が丸くなりませんが、情報が欠けたり、消えてしまうことはありません。角が丸くなくても情報は伝わるので、わざわざ角丸の画像を作成して貼り込むなどの対応は必要なしと判断できます。

プログレッシブエンハンスメントは、対象とするすべての閲覧環境のことを先に考え、漸進的に最新の環境向けに機能を追加していきますが、グレイスフル・デグラデーションは、最新の環境が中心となり、後から古い環境のことを考えます。

実社会にたとえるなら、健常者の利用を前提に設計された施設を、後から（段差を無くすなど）障がい者向けに対応していくバリアフリーに近いかもしれません。

▶最新の環境を対象に進めるWebサイト制作（古い環境は品質を落として最低限の対応）

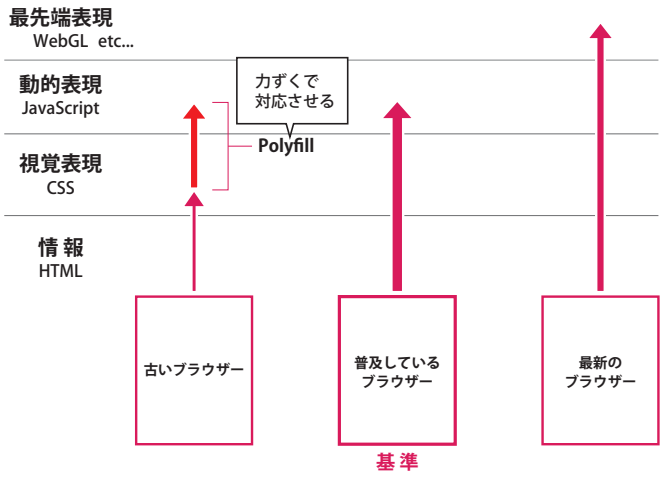


レグレッシブエンハンスメント

レグレッシブエンハンスメント（Regressive Enhancement）は、対象とする閲覧環境すべてに同等レベルの機能を実現するための開発思想、およびテクニックのことです。

最新の仕様をサポートしていない古いブラウザーに対しても、力ずくで最新ブラウザー並みの機能を持たせてしまうのです。JavaScriptのライブラリー（Polyfill＝ポリフィルと呼びます）を使って擬似的に動かします。動作が重たくなってしまいますが、最新のブラウザーと同じ機能を提供することができます。ただし、メンテナンス性が低下しますので、時間とコストがかかります。

▶古い環境に対しても「力ずくで」最新の機能を提供していくWebサイト制作

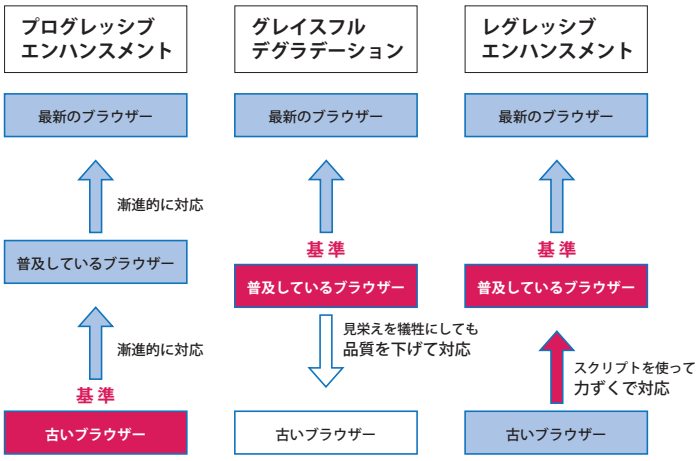


プログレッシブエンハンスメントは、閲覧環境の違いを受け入れ、その能力に応じて機能を付加していきます。グレイスフル・デグラデーションは、最新の閲覧環境を中心にしますが、古い環境には品質を落として対応します。

レグレッシブエンハンスメントは、スクリプトを駆使し、古い環境に対しても力ずくで同等の機能に引き上げ、すべての閲覧環境で同じ機能を提供します。

Webデザイナーは、Webサイトを作るときの「指針」や「方針」を決めておく必要があります。何も考えず指示されたとおりに作業をすると、「新しい表現・技術を採用してほしい」という要望と「未対応の古い環境はどうする？」という問題に悩まされることになります。Webの知識や技能を習得しても、制作の進め方を適切に判断できないとプロジェクトとしては成功しませんので、必ず理解しておきましょう。

▶Webデザイナーは（Webデザインを進めるときの）自分の方針を決めておくこと



Chapter 1
03 Webブラウザーの使い方

Webページとは、HTMLで記述された文書を「Webブラウザー」で閲覧する媒体です。Webデザインとは「Webブラウザー」で意図した通りに表示するためのデザイン。正しく理解しておきましょう。

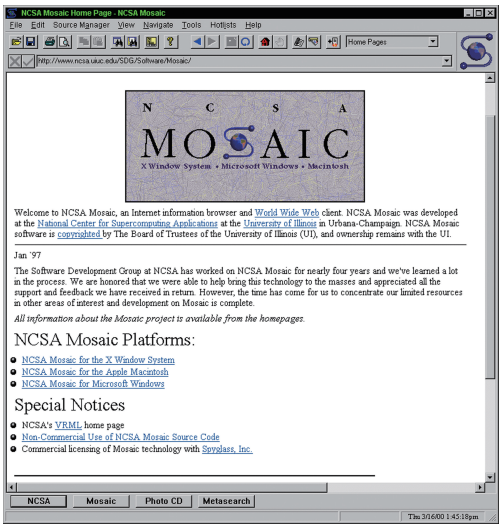
Webブラウザーの種類とレンダリングエンジン

Web ブラウザーの歴史

Webサイトを閲覧するには「Webブラウザー」が必要です。専門的にはユーザーエージェントと記されますが、たんに「ブラウザー」と呼んでもかまいません。Webページをスクロールさせながら読む閲覧用のアプリケーションソフトです。

1993年、イリノイ大学のNCSA（National Center for Supercomputing Applications：米国立スーパーコンピュータ応用研究所）にいたマーク・アンドリーセン（Marc Lowell Andreessen）らによって開発された「NCSA Mosaic（モザイク）」が、現在のWebブラウザーのルーツだといってよいでしょう。当時のブラウザーは研究論文や学術資料などを閲覧するためのビューアーとして使われており、テキストと画像を別のウィンドウでしか表示できませんでしたが、NCSA Mosaicは（雑誌のページのように）テキストと画像を一緒に表示させることができ、UNIXだけではなくWindows版やMac版も開発したことで、スタンダードなブラウザーとして世界中に広まりました。

▶現在のWebブラウザーのルーツといえる「NCSA Mosaic」



画像参照：Press Room - National Center for Supercomputing Applications at the University of Illinois
http://www.ncsa.illinois.edu/news/press#mosaic

その後、マーク・アンドリーセンは実業家のジム・クラーク（James H. Clark）に誘われ、ネットスケー

プコミュニケーションズを設立し、インターネット黎明期の代表的なブラウザーとなる「Netscape Navigator（ネットスケープ・ナビゲーター）」を開発しました。1995年にリリースされたバージョン2.0に搭載されたJavaScript（リリース時の名称はLiveScript）は現在、HTMLやCSSと同様にWebデザイナーが習得すべき重要な技術になっています。

同年8月には、Microsoftが「Internet Explorer（インターネット・エクスプローラー）」をWindowsの機能拡張パッケージに収録しました。Netscape NavigatorとInternet Explorerは利用者を獲得するために互いに機能強化をエスカレートさせ、事実上の標準として普及していたHTMLを独自に拡張してしまうなど、混乱の時代が続きます。同じHTMLで記述してもブラウザーによって動作が異なるという、Webの信頼性を低下させる大きな問題に発展していきます。

▶独自拡張されたHTMLのタグ（一例）

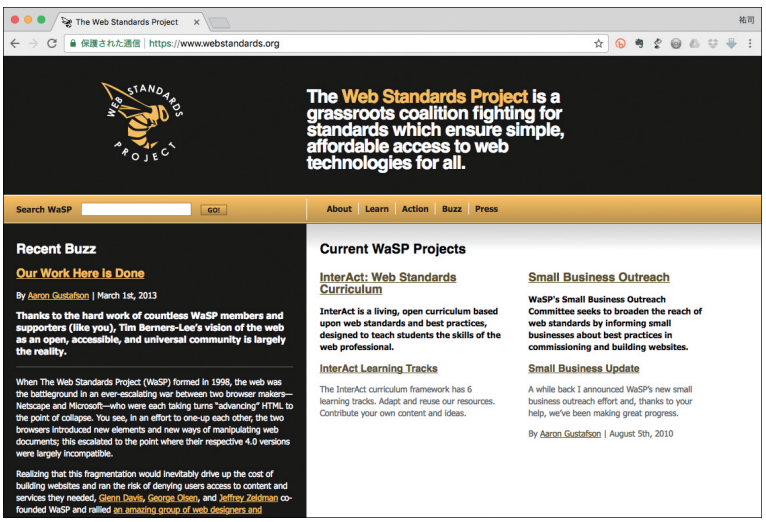
```
<blink>文字が点滅します</blink>
<marquee>文字がスクロールします</marquee>
```

HTMLを拡張して「文字が点滅（Netscape Navigatorが独自拡張）」したり「文字がスクロール（Internet Explorerが独自拡張）」するタグなども実装された。世界中に広まってしまったので、一部のWebブラウザーでは現在でも機能する

Internet Explorerが圧倒的なシェアを得る2000年頃まで続いたこの混乱期は「ブラウザー戦争」と呼ばれていましたが、Webの未来に危機感を抱いていた世界中の研究者、開発者、Webデザイナーたちが「Web標準」を掲げて大規模な啓蒙活動を開始するきっかけになりました。1998年には「The Web Standards Project（ウェブスタンダードプロジェクト：略称はWaSP）」が設立されます。

WaSPは「W3Cが策定したHTMLに準拠してWeb制作をしよう」「HTMLの独自拡張はやめよう」「正しいHTMLのマークアップを心がけよう」といったスローガンで世界規模のキャンペーンを展開し、2004年頃から正しいHTMLの使い方が浸透し始め、Web標準に準拠した（安心して作業できる）Webデザインが実現し、現在に至ります。

▶現在もアーカイブとして残存している「ウェブスタンダードプロジェクト」のサイト。彼らの活動のおかげで現在のWebデザインがある



https://www.webstandards.org/

Web ブラウザーの種類

Webブラウザーは、OSにインストール済みの標準ブラウザーと他の企業が提供するサードパーティーのブラウザーに大別することができます。Windowsが搭載されているパソコンやデバイスには「Microsoft Edge」、macOSが搭載されているパソコンやiOSが搭載されているスマートフォンやタブレットにはAppleの「Safari」、そしてAndroidが搭載されているデバイスには「Google Chrome」がインストールされています。

その他、Mozilla Foundationの「Mozilla Firefox」やOpera Softwareの「Opera」などのWebブラウザーが無償で提供されており、利用者が自由にインストールできます。

※ Microsoft Edgeは、Windows 10 以降の標準ブラウザーでそれ以前の古い OS では「Internet Explorer」が使用されています。開発はすでに終了していますが、利用者がまだいますので、しばらくの間は Microsoft Edge と混在する状況が続くと考えてよいでしょう。

Webブラウザー名	開発企業・団体	種類
Microsoft Edge （マイクロソフト・エッジ）	Microsoft	Windows 10以降の標準ブラウザー
Internet Explorer （インターネットエクスプローラー）	Microsoft	Windows標準ブラウザー
Safari（サファリ）	Apple	macOSおよびiOS標準ブラウザー
Google Chrome （グーグル・クローム）	Google	Android標準ブラウザー（他のOSにも提供）
Mozilla Firefox （モジラ・ファイアーフォックス）	Mozilla Foundation	サードパーティー製品（利用者が自分でインストールする）
Opera（オペラ）	Opera Software	サードパーティー製品（利用者が自分でインストールする）

▶Microsoft Edge

https://www.microsoft.com/ja-jp/windows/microsoft-edge

▶Apple Safari

https://www.apple.com/jp/safari/

▶Google Chrome

https://www.google.co.jp/chrome/browser/desktop/

▶Mozilla Firefox

https://www.mozilla.org/ja/firefox/new/

Web ブラウザーのレンダリングエンジン

Webブラウザーには、クルマと同じようにエンジンが搭載されています。HTMLやCSSを解釈してブラウザーのウィンドウに文字や画像を表示するための「レンダリングエンジン」のことです。同じWebページでも処理するエンジンが異なると、あるエンジンでは表示できても、他のエンジンでは表示できない、といったHTMLやCSSの実装の違いが出ています。

Microsoft Edgeのレンダリングエンジンは「EdgeHTML」、Internet Explorerは「Trident」、Google ChromeやOperaは「Blink」、Safariは「WebKit」、Mozilla Firefoxは「Gecko」を採用しています。

HTMLやCSSの実装状況などを伝えている技術サイトでは、レンダリングエンジンの名前で書かれていることが多いため、Webブラウザー名と一緒にエンジンの名前も覚えておきましょう。

▶Webブラウザー採用しているレンダリングエンジン

Webブラウザー名	レンダリングエンジン名
Microsoft Edge	EdgeHTML（エッジエイチティーエムエル）
Internet Explorer	Trident（トライデント）
Safari	WebKit（ウェブキット）
Google Chrome	Blink（ブリンク）
Mozilla Firefox	Gecko（ゲッコー）
Opera	Blink（ブリンク）

Webブラウザーの「先行実装」について

HTML や CSS の新しい機能は策定中でも利用可能になる

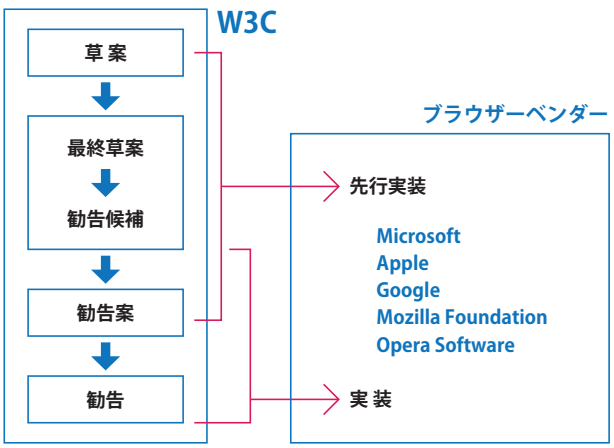
HTMLやCSSなどのWebの技術はW3Cによって策定されていますが、策定作業は草案が提出されてからワーキンググループで議論を行い、最終草案を経て、勧告候補になります。ここで実装例がないと勧告候補に上げられないため、策定途中でもWebブラウザーに「先行実装」されることになります。

実装とは「仕様に沿って機能を組み込む」ことですが、Webブラウザーが策定中の一部の仕様を「先行実装（試験的に実装）」していくことで、Webデザイナーや開発者は新しい機能を試すことが可能となり、結果的に「その機能が役に立つものかどうか」がユーザーによって検証されることになります。

つまり、完全に仕様が決定してから、Webブラウザーに実装されるのではなく、議論しながら実際にWebブラウザーで試し、本当に求められている機能かどうか検証しながら策定作業が進みます。勧告まで長い時間がかかる標準規格の策定は「何年もかけて考えて作ったのに、あまり使われなかった」ということがあり得るため、策定中にもかかわらず、このように「先行実装」されていくわけです。

Webデザイナーはこの策定作業の流れを知っておく必要があります。この知識がないと先行実装された機能を正しく利用できないからです。

▶W3Cの策定プロセス



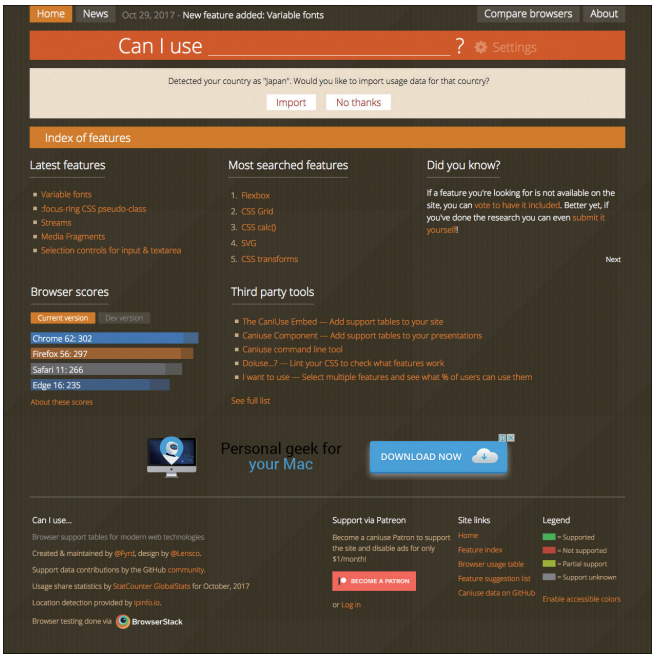
草案から最終草案、勧告候補、勧告案、勧告までのプロセスがある。最終草案を勧告候補に上げるには、ブラウザーに先行実装され、実際に使用されなければいけない。
※ブラウザーベンダーとはWebブラウザーを開発している企業や組織のこと。

ブラウザーごとの実装状況を把握すること

先行実装で注意しなければいけないのはWebブラウザーごとの「実装状況」です。すべてのWebブラウザーが足並みをそろえて実行するわけではありません。先行実装するかどうかは開発元の企業が決めますので、あるブラウザーは使えるのに、他のブラウザーはまだ使用できないという面倒なことが起こっています。このような場合は仕事で利用することはできませんが、逆にまだ策定中にもかかわらずすべてのWebブラウザーに先行実装されて「どのブラウザーでも問題なく表示される」機能であれば、すでに実績があるのですから安心して仕事でも使用することができます。

この判断は、Webデザイナーに委ねられますので、適切に判断できる知識を持っていないと、Webデザインの作業に支障が出てしまいます。実装状況が確認できる技術サイトなどは定期的にチェックして、情報収集していきましょう。

▶Webブラウザーごとの実装状況



アレクシ・デヴェリア（Alexis Deveria）氏が運営している実装状況が確認できる技術サイト。世界中のWebデザイナーが参考にしている実績のあるサイト
Can I use... Support tables for HTML5, CSS3, etc
http://caniuse.com

ベンダープレフィックスとは？

HTMLやCSSなどのWebの技術はWebブラウザーに先行実装され、いち早く試すことが可能になりますが、最終的に廃案になることもあります。例えば、HTML5には見出しをグループ化する「hgroup要素」がありました。すべてのWebブラウザーがこの要素を先行実装しており、市販されているHTML5の解説本にも掲載されているため、世界中の多くのWebデザイナーが使用していましたが、最終的に勧告された仕様書には掲載されませんでした。現在この要素は使用できません。

hgroup要素でマークアップされたWebサイトは、（すぐに修正する必要はありませんが）そのまま放置するわけにもいきませんので、どこかでタグを書き換えることになります。また、図書館などに置かれている古いHTML5の解説書には廃止されたタグも掲載されていますので注意しなければいけません。

▶hgroup要素の記述例（廃止されているため現在は使用できない）

```
<hgroup>
  <h1>大見出し</h1>
  <h2>中見出し</h2>
</hgroup>
```

Webブラウザーを開発している企業は、策定中のCSS3の仕様を先行実装する際、固有の接頭辞（ベンダープレフィックス）を付けて区別しています。Microsoft EdgeやInternet Explorerは「-ms-」、SafariやGoogle Chrome、Operaなどは「-webkit-」、Firefoxは「-moz-」です。

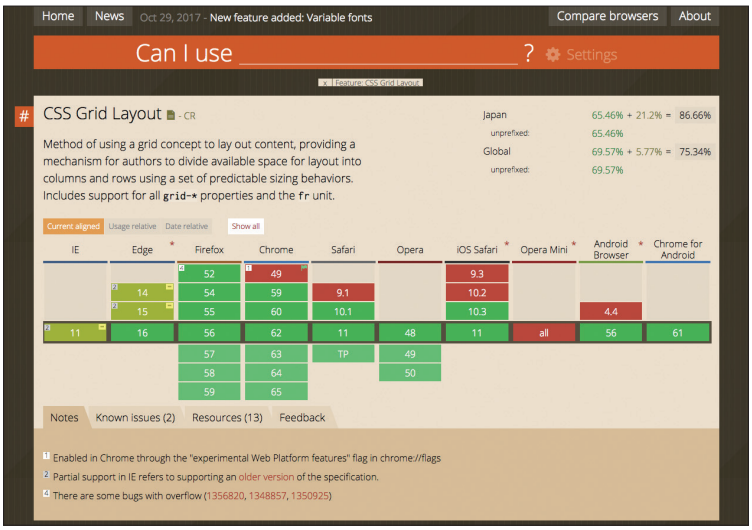
勧告候補になるとベンダープレフィックスは付きませんが、Webブラウザーによって対応が異なるため事前に確認しておく必要があります。ベンダープレフィックスを「付ける・付けない」もWebデザイナーの判断になります。

※ベンダープレフィックスは極力付けないページデザインが望ましいので、止むを得ない場合の処置だと捉えておきましょう。

▶Webブラウザーごとのベンダープレフィックス

Webブラウザー名	ベンダープレフィックス
Microsoft Edge	-ms-
Internet Explorer	-ms-
Safari	-webkit-
Google Chrome	-webkit-
Mozilla Firefox	-moz-
Opera（バージョン15以降）	-webkit-
Opera（旧バージョン）	-o-

▶CSS Grid Layoutの実装状況（例）



CSS Grid Layoutの実装状況を「Can I use...」で確認すると、最新のブラウザーであればIE（Internet Explorer）以外はサポート済みであることが分かる（2017年11月現在）。緑色はサポート済み、薄い緑は部分的なサポート、赤は未対応。
ここで「古いブラウザーは未対応なのでまだ早い」と判断するか、「最新のブラウザーは実装済みなので部分的に採用していこう」という方針を打ち出すかは、Webデザイナーが決めなくてはいけない。
Can I use... Support tables for HTML5, CSS3, etc
https://caniuse.com/#feat=css-grid

▶ベンダープレフィックスが付加された例（CSS Grid Layout）

```
header {
    -ms-grid-column: 1;
    grid-column-start: 1;
    grid-column-end: 3;
    -ms-grid-column-span: 2;
    -ms-grid-row: 1;
    grid-row: 1;
}
```

IE 11 および Edge 15 以前のためにベンダープレフィックス「-ms-」を付加している例。

Webについてこれから学習する人が「Web標準に準拠しよう」と聞くと、「完全に仕様を決定してから公開」され、その後、Webブラウザーに実装され、Webデザイナーが仕事に使っていく、という流れをイメージすると思いますが、テクノロジーに依存しているWebの世界は「実績主義」です。

実際にたくさんの人が使って、実績を得た技術が「標準」になるという考え方で進められます。策定中にWebブラウザーが「先行実装」して、私たちが実際に試して「本当に必要なもの」だと分かったら、勧告候補に挙がって、最終的に決定されます。

Webデザイナーはこのプロセスを正しく理解し、実装状況などを見極めていかなくてはいけません。