

# 対象

- 一 明日から会社のホームページを作れと言われた人
- 一 部下の作ったページをチェックする人
- 一 官公庁、学校、公共施設のページを作る人
- 一 個人のページからコミュニティを広げたい人
- 一 中身はいいのにアクセス数がイマイチという人





# 正しいHTMLの書き方



篠原 稔和

神崎 正英

インターネットマガジン編集部

Calligraphy: Hiroko Shiokara

会社や役所がウェブサイトを持つのは当たり前になった現在、ウェブは公的なメディアだと言える。

こうした時代には、ウェブサイトの作成を任せられた人は、個人のホームページのように好き勝手に作るわけにはいかない。

「ユーザビリティ」と「アクセシビリティ」という考えを取り入れた適切なHTMLの書き方が必要なのだ。

この特集では、「入門書やソフトは揃えたけど、実際にどう書けばいいのかわからない」という人のために、ウェブサイトを作成するときに気を付けなければならないポイントを総ざらいしよう。

正しいHTMLの作法を身に付ければ、あらゆる人に喜ばれるサイトができる。

# ユーザビリティ & アクセシビリティの発想を取り入れよ!

私たちの生活や仕事の玄関としての役割を担おうとしているウェブサイト。その姿を根底からくつがえすようなムーブメントが、ウェブ先進国の米国を中心に世界に広がろうとしている。今まさに日本にもその波が押し寄せようとしている「ユーザビリティ」と「アクセシビリティ」の発想。これをベースにしたウェブサイトのデザインや構築方法について考えてみよう。(篠原稔和)

## ! ユーザビリティ ~ 利用者の反応を取り入れる

「ユーザビリティ」(Usability)とは、日本語に訳すと「使い勝手、使いやすさ、わかりやすさ」となる。ユーザビリティは、さまざまな製品を利用者にとって使い勝手のよいものにするために、利用者の反応や意見をデザインワークに取り入れていく活動そのものだ。そもそも、ハードウェアやソフトウェアの製品開発や評価の現場で長年にわたって培われてきたもので、たとえば切符の自動発券機や銀行のATMなどの公共的な機械を少しでも利用者にとって使いやすいものにする

ために欠かすことのできない活動なのだ。ユーザビリティの権威で、全米でベストセラーとなり先頃日本でも出版された『ウェ

ブ・ユーザビリティ』の著者であるヤコブ・ニールセンは、ユーザビリティの特性を5つにまとめている。

ユーザビリティの特性

- 「学習しやすさ」 すぐに使って簡単に学習できるかどうか。
- 「効率性」 一度使うと高い生産性を上げられるようになるかどうか。
- 「記憶しやすさ」 しばらく使わなくても再び使うときに覚え直さなくても済むかどうか。
- 「エラー」 エラーが少なく、たとえエラーが起こっても簡単に回復できるかどうか。
- 「主観的満足度」 個人的に満足できて、使うことが楽しくなるようになっているかどうか。

## ! ユーザビリティを実践する2つの方法

ウェブサイトに対しても使い勝手を向上させるために「ユーザビリティ」の考え方が積極的に応用され始めている。ユーザビリティの発想を取り入れたウェブサイトは、情報を発信する側の考えを優先するのではなく、使い手の立場に立ってデザインし、構築していくことになる。たとえば、使い手の操作環境に配慮した表現方法を選択したり、利用者の立場にたった情報の並べ方や見せ方などを考えたりしていくのだ。これを実践するには、おもに2つの方法がある。

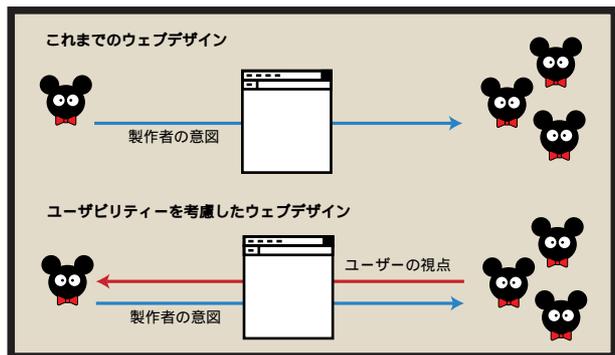
① さまざまな製品で試みられてきたテストの結果や黄金原則、ウェブサイトのユーザビリティテストで発見された数多くの法則や方法論、それらをまとめたガイドラインなどに従って、自分のデザインしたウェブサイト

をチェックしていく方法。ニールセンの書籍や、ウェブサイト上で公開されているデザインガイドライン(たとえばイール大学のもの<sup>Jump</sup>)を参考に、ポイントを確認することが有効だろう。

② 実際に利用者の声を引き出す「ユーザビリティテスト」を実施すること。これは、さまざまな角度から検討された質問を利用者に投げたり、利用者が発する声を聞いたりしながら、その内容や分析の結果をデザインに反映させていくことだ。たとえば、自分のデザインしたサイト

を友達に声を出してもらいながら操作してもらっただけで、改善のためのヒントを得られることもある。

Web Style Guide  
<sup>Jump</sup> info.med.yale.edu/caim/manual/



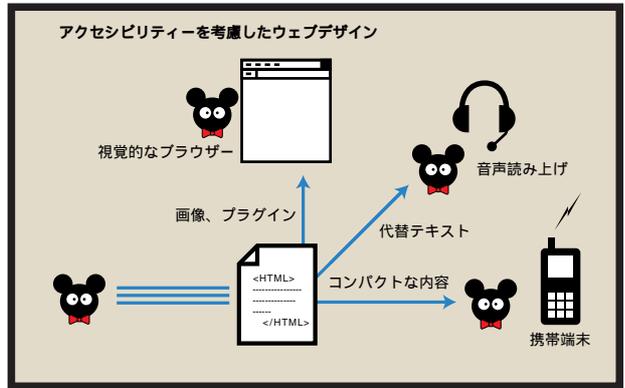
# アクセシビリティ ~すべての人が利用できるページ作り

ユーザビリティに加えて、最近では「アクセシビリティ」(Accessibility)もウェブのデザインと構築の際に欠かすことのできない視点となっている。アクセシビリティとは、ウェブのコンテンツが障害を持つ人たちにとってもアクセスしやすいものになるように配慮することを意味し、そのガイドラインがW3Cによって策定されている(215ページ参照)。

W3Cのガイドラインには、グラフィカルなブラウザを対象としたコンテンツには代替手段を必ず用意することや、色、言語、ナビゲーションなどの問題、文章表現にわたるまで、コンテンツをアクセスしやすくするためのルールが記載されている。世界中のウェブサイトがこれを準拠すべきルールとして採用する動きにある。この動きは、製品における「ユニバーサルデザイン」(身体障害者や高齢者などすべての人にやさしい

デザインを目指す運動)と連動しており、これからのデザイナーや制作者にとって欠かすことのできない視点なのだ。

アクセシビリティは障害を持つ人たちだけを対象にしているわけではない。さまざまなバージョンのブラウザや異なる環境下での操作性にも関連する問題だ。新たなインターネットのプラットフォームとして爆発的に普及しているモードなどの携帯端末に向けたサイト作りにも役立つだろう。



# 顧客を逃がすな!

今欧米では、ユーザビリティやアクセシビリティに配慮したウェブサイトでなければ、顧客が逃げてしまい、時間を費やす価値のないものとして葬られ、これを理解できないデザイナーは職を失ってしまうとまで言われている。数多くのウェブサイトが誕生し、仕事や生活の一部として使われるようになってきた日本も、すでに同じ状況を迎えようとしているに違いない。ユーザビリティやアクセシビリティのマインドを持つことができれば、ウェブサイトの未来をしっかりと捉えることができるだろう。

**篠原穂和 (しのはら・としかず)**  
ソシオメディア 代表。情報デザインやユーザビリティを実践するコンサルティング、講演、執筆に従事。訳書・監訳書に『情報アーキテクチャ入門』(オライリー・ジャパン刊)、『ウェブ・ユーザビリティ』(MdN刊)ほか多数。  
 [www.sociomedia.com](http://www.sociomedia.com)

# 「Alert box」でユーザビリティのトレンドを知る

「Alert box」 は、『ウェブ・ユーザビリティ』 の著者でニールセン・ノーマン・グループ(NNG)の代表を務めるヤコブ・ニールセン氏が隔週で連載するウェブサイト上のコラムだ。毎回インターネットやウェブサイトに関するさまざまなトピックを取り上げ、利用者中心のインターフェイスの実現に向けて、まさに鋭い「警鐘」(Alertbox)を鳴らし続けている。

このコラムは、1995年5月以来5年間にわたって100回を超える連載となっており、最近では1コラムにつき約21万ページビューを獲得するという、世界が注目するウェブの金言集となっている。これまでで一番人気のあったコラムは、「Top ten mistakes of Web design」(ウェブサイトの間違いトップ10)。現在でも月間4万ページビューを獲得し続けている。この10原則は、ウェブにおけるユーザビリティのガイドラインとして利用されており、ウェブデザイナー必見のルールとなっている。

ニールセン氏は全米を中心に数多くのイベントでのキーノートスピーチを務めている。来年2月には、NNGのワールドツアーの一環で、ドナルド・ノーマン氏をはじめとした同グループのメンバーとともに来日する予定だ。



Alertbox  
 [www.useit.com/alertbox/](http://www.useit.com/alertbox/)

ウェブ・ユーザビリティ  
(MdN刊、定価2,800円)  
 [www.mdn.co.jp/Books/](http://www.mdn.co.jp/Books/)



ユーザビリティのチェックポイントはこれだ

# 使いやすさで訪問者の心をつかめ!

ウェブサイトの基本は利用者の視点に立ったユーザビリティ。せっかく苦労していい内容のものを作っても、使いづらいページではユーザーは逃げていく。ロゴや背景画像ばかりに凝ったりしないで、まず「上手に伝える」ことに気を使おう。(神崎正英)

## ユーザビリティは全体の設計から

ユーザビリティに優れたサイトを作るには、まずサイト全体の設計から考えていくことが重要だ。ウェブサイトを作る方法には、どんな材料があるかを考えてそれを積み上げて構築していく「ボトムアップ」の手法と、全体の設計図に基づいて制作していく「トップダウン」の手法がある。利用者にわかりやすく、制作者にもメンテナンスしやすいサイトを作るためには、右ページの図のようなトップダウンのアプローチが欠かせない。

### 1. コンセプト

サイト構築の第一歩は、コンセプトの設定、つまり「誰に向かって」というターゲットと「何を何のために」という目的の確認だ。

### 2. コンテンツ

コンセプトが明確になったら、コンテンツ案を練り上げていく。手元の材料でできる内容もあるだろうし、新たに取材したり調査したりする必要も出てくるだろう。常に目的とターゲットを念頭に置いてコンテンツを考えることで、場当たりのでない整合性の取れたサイトができる。

### 3. サイト設計

コンテンツ案がそろったら、想定する利用者にメッセージがうまく伝わるようにコンテンツの配置を設計しよう。このとき重要なのは、利用者の視点で構造を考えることだ。たとえば、オンラインショップを

作ろうとしているのに、会社の組織に合わせて事業部ごとに別々の入り口を設けていては、利用者はどこで買い物ができるのかわからない。また、初心者を対象にした情報サイトならできるだけシンプルな階層構造が望ましいが、業界通向けならトップページから目的地へ近道を豊富に用意するほうが使いやすいだろう。利用者はサイトを見るときにどんな予備知識を持っているのか、何を期待しているかを意識しながら設計することが大切だ。

全体設計をしたうえで、ページの間接性をわかりやすくするナビゲーションや、サイトのアイデンティティのあり方を検討していく。ナビゲーションとアイデンティティのデザインを一体化させると、一貫した移動手段を提供することにもなり、安心感を与えられる。アイデンティティはロゴマークだけではなく、各ページのマージンの取り方や見出しの表現方法、コラムの背景色などを統一すれば、サイトの個性が生まれる。こうした要素を常に同じ手法で表現すれば、情報の意味もいっそうわかりやすくなるはずだ。一貫性のあるデザインには、スタイルシートを活用しよう。デザイン設定を1つのスタイルシートにまとめ、すべてのページから

```
<link rel="stylesheet" href="mystyle.css" type="text/css">
```

という形でスタイルシートのファイルにリンクしておく。どのページでもマージンや見

出しなどの要素が同じスタイルで表現され、統一感が生まれる。デザインを修正する場合でも、スタイルシートのファイルだけを変更すれば済むので、メンテナンスの効率も大幅に向上する。

### 4. URL 設計

最後にファイルを配置するディレクトリー(フォルダー)の構造を決める。メンテナンスのしやすさと拡張性に注意しよう。ファイル名のルールをきちんと決めておくことが大切だ。共通する画像などを置くためのディレクトリーも用意して、ファイルが散逸しないように設計する。

ディレクトリーの分け方は、将来コンテンツが増えたり担当セクションが変わったりしても移動の必要がないような方法を採用すべきだ。たとえば輸入品のページで「/imports」というディレクトリーを用意すると、その商品が国内生産に移行したらURLも変更せざるを得ない。扱う商品が増えた結果、サブディレクトリーに再分類するときにも同様の変更が生じる。

URLを変えると、リンクやブックマーク、検索サイトの登録が無効になり、せっかくの訪問ルートを開ざしてしまうことになって、マーケティング上も大きなマイナスだ。URLは商品のブランドと同様の財産であるという認識を持つようにしたい。できるだけ状況に左右されないカテゴリーを設け、その中では日付を基準にしたサブディレクトリーを設けて分類するのが比較的安全な方法だろう。

## 2 どこからアクセスしてもわかるサイトに

書物と違ってウェブサイトは現在の位置が全体のどのあたりなのかを把握するのが難しい。初めての利用者は、適切なサポートがないと手探りで使うことになってしまふ。利用者がサイトで迷子にならないようにするには、ナビゲーションをきちんとしてあげることが大切だ。すべてのページに一貫した形でサイトのトップページ（もしくはサブコンテンツのトップページ）へのリンクを用意しよう。ナビゲーションをデザインに組み込んでおくと、検索サイトなどからの来訪者はどこにたどり着いてもそのサイトに来たことが一目瞭然で、使いやすい上に統一感を与える効果もある。

一つながりのページには、前後のページや目次へのリンクもわかりやすく示しておく。リンクは「前」や「次」ではなく「第2章：論理タグについて」のように内容が具体的にわかるように書くと、利用者は安心してサイトを利用できる。最初のページから順番に読まれるとは限らないので、「前ページと同じなので繰り返さない」などという書き方を避ける配慮も大切だ。

link要素（注1）でrel属性とhref属性を使ってリスト1のようにページの間を記述しておく、Lynx（注2）やiCab（注3）はブラウザ自身ナビゲーションバーを用意してくれる。もっと多くのブラウザが対応するよう期待を込めて、積極的に利用しよう。

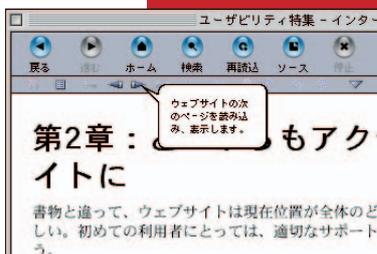
注① HTMLでは「<開始タグ>内容<終了タグ>」全体を「要素」と呼ぶ。たとえば「<a>～</a>」は「a要素」。

注② UNIXやウィンドウズなどに対応したテキストブラウザ。Jump lynx.browser.org

注③ HTML 4.0に完全対応したマッキントッシュ用のブラウザ。Jump www.icab.de

### リスト1

```
<head>
<link rel="prev" href="ch01.html" title="第1章">
<link rel="next" href="ch03.html" title="第3章">
<link rel="toc" href="toc.html" title="目次">
```



iCabには、link要素に対応したナビゲーションバーの機能がある。

## 3 「導入ページ」は作らない

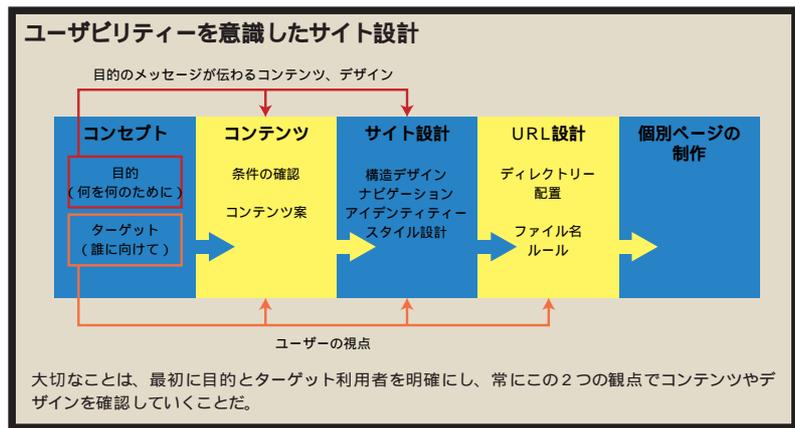
ユーザーは早く情報にたどり着きたい。雰囲気のあるデザインは悪くないが、デザインの結果として情報へのアクセスが悪くなるページに対しては、利用者の評価は厳しい。

とりわけ評判が悪いのが、目次の前に置かれる導入ページだ。ブランディングをねらってか、それともプレゼンテーション受けが目的か、最初にフラッシュなどを使ったおまけを押しつけるサイトが非常に多い。この通過儀礼はまったく無意味、もしくは逆効果だ。ユーザーは「Skip」のリンクがあればさっさとそれをクリックし、そうでなければ別のサイトに行ってしまう。自己満足のために多数のユーザーを失っていることにそろそろ気付くべきだ。

**神崎正英（かんざき・まさひで）**  
 ウェブコンサルタント、法政大学講師。著書に『プロフェッショナル電子メール』（ハルアンドアーク刊）、『ユニバーサルHTML/XHTML』（10月刊行予定）、The Web KANZAKI（Jump）にて、ごく簡単なHTMLの説明から音楽の話までを発信中。Jump www.kanzaki.com



サイトは一貫したナビゲーションを用意する。上はアップル、下はAmazon.comのナビゲーション。



## 4 リンクはわかりやすく

ウェブページは書物と違って全体像が見えない。そこで、次の行き先の内容を把握できるように、アンカー(注4)を具体的に記述することが重要だ。作者はサイトの構造が頭に入っているから簡単なアンカーでもリンク先の内容はわかると思いがちだが、初めての利用者は次の行き先が理解できず、迷ってしまうことになる。

ユーザビリティの観点から言えば、リンクは本文に埋め込むよりも、右の②のように段落や節の単位でまとめて列挙するほうが使いやすい。文章中に設定したリンクは見落としやすいし、文章の流れからリンク先を詳しく説明することが難しいからだ。

本文中にアンカーを設定する場合は、title属性を活用してリンク先についての付加説明を提供するとよい。title属性を付けたアンカーにマウスを載せると、③のよう

にツールチップなどで説明が表示されるので、利用者は納得してクリックできる。

ウェブ文書の場合、書物と違って全体像が見えないので、次の内容を把握できるようにするために、アンカーを具体的に記述することが重要だ。作者はサイトの構造が頭に入っているから、簡単なアンカーでもリンク先の内容は分かると思いがちだが、初めての利用者は、それでは次の行き先が分からず、迷ってしまうことになる。

使いやすさの観点から言えば、リンクは本文に埋め込むよりも、パラグラフや節の単位でまとめて「参考リンク」という形で列挙する方が使いやすい。本文のフレーズに設定したリンクは見落としやすいし、文章の流れからアンカーとしてリンク先を詳しく説明することが難しいからだ。

① 本文に埋め込んだリンクは、精読しないと見落としやすく、リンク先についても説明できない。

ウェブ文書の場合、書物と違って全体像が見えないので、次の内容を把握できるようにするために、アンカーを具体的に記述することが重要だ。作者はサイトの構造が頭に入っているから、簡単なアンカーでもリンク先の内容は分かると思いがちだが、初めての利用者は、それでは次の行き先が分からず、迷ってしまうことになる。

使いやすさの観点から言えば、リンクは本文に埋め込むよりも、パラグラフや節の単位でまとめて「参考リンク」という形で列挙する方が使いやすい。本文のフレーズに設定したリンクは見落としやすいし、文章の流れからアンカーとしてリンク先を詳しく説明することが難しいからだ。

② 必要に応じて、節ごとに参照リンクをまとめて使いやすいになる。

ウェブ文書の場合、書物と違って全体像が見えないので、次の内容を把握できるようにするために、アンカーを具体的に記述することが重要だ。作者はサイトの構造が頭に入っているから、簡単なアンカーでもリンク先の内容は分かると思いがちだが、初めての利用者は、それでは次の行き先が分からず、迷ってしまうことになる。

使いやすさの観点から言えば、リンクは本文に埋め込むよりも、パラグラフや節の単位でまとめて「参考リンク」という形で列挙する方が使いやすい。本文のフレーズに設定したリンクは見落としやすいし、文章の流れからアンカーとしてリンク先を詳しく説明することが難しいからだ。

③ 本文に埋め込んだアンカーにtitle属性を与えると、ブラウザによってツールチップが表示され、利用者のヒントになる。

## 5 「戻る」ボタンのじゃまをしない

利用者は、サイトの中で迷ったときはまずブラウザの「戻る」ボタンを頼りにする。自分なりに把握できていた地点まで戻ってやり直そうとするのだ。ところが、自動的に新しいウィンドウが開くリンクの場合は、しばらく進んでから戻ろうとすると「戻る」ボタンが途中で動作しなくなる。利用者が新ウィンドウに気付かないでいると混乱するだろう。

また、meta要素を使った自動リフレッシュも「戻る」ボタンの機能を麻痺させてしまう。リフレッシュ時間を0に設定している場合は、何度「戻る」ボタンを押しても、また次のページに逆戻りしてしまうからだ。

ブラウザのウィンドウ管理や「戻る」ボタンの機能の利用は、ウェブのもっとも基本的なナビゲーションだ。これらを利用者の意図しない方法で操作すると、著しくユーザビリティを悪化させることを理解したい。

## 6 論理マークアップで読みやすく

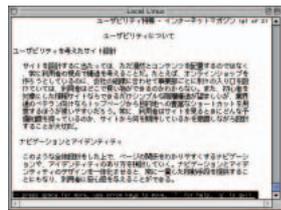
HTMLの目的は、環境に依存しない形で情報を表現することだ。大画面モニターからPDAまでさまざまな環境に合わせて異なるHTMLを作成するのは現実的ではない。そこで、HTMLでは文書の「タイトル」、「見出し」、「段落」などの意味的な役割だけをマークアップし、どんなふうに表示するかはブラウザに任せるのだ。このような書き方を「論理マークアップ」という。

論理マークアップの要点は、見出しと段落を正しく示すことだ。見出しはh1~h6の6段階の要素が用意されているので、文書の大見出しから順番に用いていく。アウトラインプロセッサで文書の構成を練り上げれば、その項目がそのまま見出し要素になる。見出し要素が適切に使われていれば、見出しに目を通すだけで文書の骨格が頭に入るはずだ。ところがh1を「文字を大きくするタグ」と勘違いして使うと、アウトライン構造が不明になってしまう。

本文は、1つのトピックについて述べた段落ごとにp要素としてマークアップする。

段落中で重要なポイントはem要素やstrong要素で強調しておく。こうした要素によって、本文の内容が明確になり読みやすいページになる。p要素は改行や空行を加えて表示されるが、br要素で改行するのは違うことに注意しよう。画面表示の指示ではなく、意味の単位を示すことが目的なのだ。

ただし、論理マークアップはそもそも文章自体が論理的でないとうまく活かせないだろう。結局は構成を考えたよい文章を書くのが先決ということになる。



h1~h6の見出し要素はフォントの大きさを指定するものではない。Lynxのように、h1要素はセンターリング、h2以降は左からのインデントで見出しのレベルを示すブラウザもある。

## 7 読めない色やフォントは困りもの

もしウェブページの文章を読んでもらいたいなら、その文字は読みやすくなければならない。あたりまえのことなのだが、文字が読みにくいサイトはとても多いのだ。

画面全体の配色は同じトーンや色相でまとめると上品で落ち着くが、文字に関してはルールが異なる。下の画面のように、文字は背景色と十分な明度のコントラストがないと読みづらいことに注意しよう。青は純色でも明度が低いので、青と黒や濃紺の組み合わせはとも視認性が低くなる。

フォントサイズも利用者によって異なるので、自分の画面に合わせてフォントサイズを落とすと、標準フォントが小さい環境では判読できなくなることも頭に入れておきたい。

画面全体の配色は同じトーンや色相でまとめると上品で落ち着くが、文字に関してはルールが異なる。文字は背景色と十分な明度のコントラストがないと読み辛いことに注意しよう。青は純色でも明度が低いので、青と黒や濃紺の組み合わせはとも視認性が低くなる。

注④ アンカーとは、リンクの始点と終点のこと。HTMLではアンカーの始点はa要素で表し、a要素のhref属性で指定するHTMLファイルや画像などが終点側アンカーになる。

注⑤ 適正なHTMLでは、HTMLの先頭に文書型宣言を置いて、どの文書型定義(DTD)を使うかを宣言する。HTML 4.01の厳密型DTDを使うなら、<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">のように記述する。

## 9 フレームは避ける、使うなら正しく

フレームはメニューを固定的に表示しながらコンテンツを切り替えていくのに便利な手法だが、問題点も多い。まず右のようにモニターのサイズが小さい場合、画面を分割されると使い勝手がとても悪くなる。ブックマークや履歴でページを区別しにくく、個別ページへのリンクも難しい。もちろん、フレーム未対応のブラウザでは表示できないという問題もある。さらに、安易に外部のコンテンツをフレーム内に表示すると、著作権を侵すことにもなりかねない。

こうした問題点から、次期のHTMLであるXHTML 1.1ではフレームは標準規格から外されることになっている。メニューの固定表示などは、スタイルシートを使って実現すべしということなのだ(P.211「CSS 2を使った次世代レイアウト」参照)。

現状でどうしてもフレームが必要な場合は、次のような点に注意して正しいフレームを提供しよう(リスト2)。

・正しい文書型宣言(注5)：通常のHTMLのDTDにはフレームは含まれないので、フレーム用のFramesetによる宣言が必要。

・フレーム非対応ブラウザのためにnoframes要素を用意する：frameset要素の中に、frame要素と併せてnoframes要素を記述する。noframes要素の中は、body要素から始まる通常のHTMLだ。

・フレームから外部サイトへのリンクは、<a href="http://..." target="\_top">としてフレームを解除する。

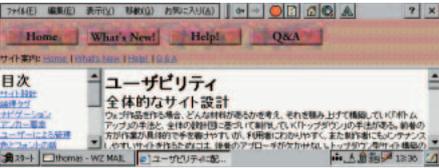
## 正しいタイトルの付け方

ページのタイトル(title要素)は、ブラウザの画面で目立たない(ウィンドウのタイトルバーに表示されるだけ)ために軽視されがちだが、この役割は実はとても重要だ。タイトルは検索サイトやブックマークの登録に使われるので、ユーザーがそのページを開くかどうかはタイトルを見て判断することになるからだ。

タイトルの使われ方を見れば、タイトルは「目次」ではなく「ユーザビリティ解説の目次」のように具体的に付けるべきだということがわかる。検索サイトの検索結果リストでは、その前後のページのことはわからないのだから、サイトの文脈を離れてもタイトルだけで内容の見当が付くようにする必要があるということだ。

ただし、「インターネットマガジン11月号特集……」のようなシリーズ名を頭に置いたタイトルは、履歴やブックマークで見るときにわかりにくい。「HTMLの正しい書き方-インターネットマガジン11月号特集」という具合に、最初に要点を書くほうが望ましい。

パソコンのモニターなら気にならないフレームでも、携帯端末の小さな画面では情報の表示範囲が極端に狭くなる。(ウィンドウズCEの画面コピーには伊藤栄一郎氏のCatCEを使用しました。)



リスト2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
<html>
<head><title>ユーザビリティに配慮したフレーム</title></head>
<frameset cols="20%, 80%">
  <frame src="toc.html" title="目次エリア">
  <frame src="section1.html" name="sections" title="本文表示エリア">
</frameset>
<noframes>
<body>
<p><a href="toc.html">目次</a>をご覧ください</p>
</body>
</noframes>
</html>
```

## 10 テーブルの使いすぎに注意

フォトショップなどで作成した画像をガイドによって切り分け、それをテーブルで固定的にレイアウトするというデザイン手法が紹介されることがある。一見ねらいどおりの美しいページが再現できそうだが、それは同じ環境で見た場合だけのこと。異なる環境では困ったデザインとなってしまふ。

この方法の問題は、テーブルのサイズを絶対値で固定してしまうことだ。本来のテーブルはウィンドウの大きさに合わせて自動的に伸び縮みするはずだが、固定レイアウトは利用者の都合などお構いなしだ。これでは、小さな画面しかない携帯端末では、横スクロールをしないと内容を読むことができないし、逆に大きなモニターを使っても、そのスペースを生かすことができない。ウェブデザイナーは、テーブルのテクニックを弄するのではなく、もっとウェブの特徴を活かした新しいデザインを考えるべきだろう。



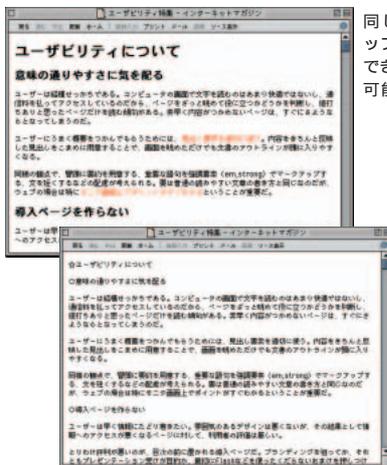
画面全体をテーブルにして横幅を固定すると、小さな画面には収まらずに横スクロールが必要になる。パソコンユーザーでも、ウィンドウの大きさは自分で管理したいもの。

## 12 ぱっと見てわかるページにする

利用者は結構せっかちだ。コンピュータの画面で文字を読むのはあまり快適ではないし、通信料を払ってアクセスしているのだから、ページをざっと眺めて役に立つかどうかを判断し、値打ちありと思ったページだけを読む傾向がある。素早く内容がつかめないページは、すぐにさようならになってしまうのだ。

ユーザーにうまく概要をつかんでもらうためには、見出し要素を適切に使う。内容

をきちんと反映した見出しをこまめに用意することで、画面を眺めただけでも文書のアウトラインが頭に入りやすくなる。同様の観点で、冒頭に要約を用意する、重要な語句を強調要素 (em, strong) でマークアップする、文を短くするなどの配慮が考えられる。要は普通の読みやすい文章の書き方と同じなのだが、ウェブの場合はモニター上でポイントがすぐわかるということが特に重要だ。



同じ文章でも見出しや強調要素を適切にマークアップすると、画面をざっと見ただけで要点を把握できる。利用者には、このように「流し読み」が可能な文書が親切。

## 13 長いコンテンツをテーブルで囲まない

通常ブラウザは、HTMLを読み込みながら届いたデータを順次画面に表示している。利用者は先頭部分を読みながら待つことができるので、長いデータでもそれほど苦痛を感じなくて済む。

ところがテーブルの場合は、そのテーブルが何列あって幅がどれだけ必要なかがわからないと、ブラウザは表示を始めることができない。つまり、テーブル部分の

## 11 1ページの適切なサイズは8秒

最近のウェブ制作者の世界では、「8秒ルール」というものが話題だ。昨年米ゾナ・リサーチが発表した「14.4kbps モデムを使っている利用者が8秒以内にページを表示できないと、1か月あたり7,300万ドルの機会損失があり得る」というレポートがその震源になっている。

14.4kbps は今やそれほど現実的ではないにせよ、この8秒という値はリアリティーがあり、ページ制作の目安となる。現在の主流が56kbpsのモデムもしくは64kbpsのISDNとするなら、通信環境がよいときで50~60Kバイトが約8秒でロードできるデータ量だ。そうすると、1ページの許容サイズは、画像も合わせてせいぜい150Kバイト程度までということになる。週刊誌1ページが約2000字として4Kバイト。テキスト中心なら50Kバイトは余裕だから、あとは画像をどう扱うかという問題だ。厳しいと感じるかもしれないが、ビジネスチャンスを逃さないためにも、基本的にはこの程度のサイズに抑えておくべきだろう。

危険なのは、最初のページはサイトの「顔」だから、グラフィックを駆使して見栄えをよくしようという発想。ユーザーは、トップページで使い勝手も含めたサイトの品定めをしていることを思い出そう。同じ会社の調査によれば、70Kバイトをロードする必要があるサイトは、訪問者の半数を取り逃がしているということだ。

テキストや画像などのデータすべてを読み込まないと、画面には何も表示できないということだ。

長いページ全体をテーブルで囲むと、利用者はずっと空白状態の画面で待たされることになる。これはかなり苦痛だ。コンテンツをテーブルで囲む場合も、リード文はテーブルの外に出すなどして、利用者を待たせない工夫が重要だ。

## 14 印刷に配慮する

ウェブページは常にモニター画面で見るとは限らない。ある程度まとまったコンテンツは、印刷してゆっくり読むという利用者は少なくないはずだ。印刷を考えた場合には、いくつか気を付ける点がある。たとえば「ここをクリック」などといったアンカーは紙の上では滑稽だし、「赤い文字の部分を読んで」という表現は白黒印刷では理解できない。画面用フォントは印刷に不向きなことが多いので、印刷用スタイルシートで別途フォントを指定すると、画面でも印刷しても読みやすいページとなる(リスト3)。ページを細かく分割しているときは、圧縮ファイルやPDF版などを用意すると親切だ。

```
@media print{
  body {
    font-family: "平成明朝",
      "MS P明朝", serif;
    font-size: 10pt;
    color: black;
    background-color: white;
  }
}
```

リスト3

## 15 チェッカーソフトを使おう

文法的な正しさがすべてではないが、間違いのないHTMLを書けば、知らない環境でもきちんと情報が伝わるはずだし、将来のデータ活用の際にもメリットがある。変なテクニックを弄しないHTMLはシンプルなのでそんなに間違えることもないはずだが、ときどきはチェッカーソフトを使って文法の正しさを確認してみるといいだろう。W3CにはHTML Validator (Jump01) とCSS Validator (Jump02) が用意されていて、

宣言している文書型に従って文法をチェックしてくれる。日本語で解説をしてくれるAnother HTML-lint (Jump03) もおすすめだ。エラーが報告されたら、何が間違いだったのかをよく確認して今後活かすようにしよう。

- Jump01 validator.w3.org
- Jump02 jigsaw.w3.org/css-validator/
- Jump03 openlab.ring.gr.jp/ki6/htmlint/

## 16 チェック体制を作れ!

制作者の考えることとユーザーの考えることは往々にして微妙なずれがあり、このギャップが使いにくさの原因となる。作っている側ではこの点はなかなかわからないので、利用者の声を聞くしかない。できればサイトの公開前に、こうした問題の可能性をテストするべきだろう。大がかりなユーザビリティテストでなくても、隣の部署の数人に「ちょっと使ってみて」と声をかけ、その様子を観察するだけでも

いろいろな発見がある。作者の思いもよらぬ方法でページを選択したり、意外なところで操作に迷ったりするといったことから、サイトの改善点が見えてくるのだ。使い勝手以外に、異なる環境でのチェックも欠かせない。開発はローカルの快適な環境で進めても、モデム接続でも8秒ルールが守れるかなどは随時確認すべきだ。こうしたテストやチェックは、思いつきでなく確実に実行するシステム作りが大切だ。

## topics CSS 2を使った次世代型レイアウト

CSS 2には一般のウェブ制作者には手に余るほどの豊富なプロパティが定義されている。ブラウザの実装はこれからのものが多いが、CSS 2を使うと従来はフレームやテーブルを駆使しなければならなかったデザインや機能をスマートに実現できる。たとえば、フレームに依存していたスクロールしないメニューをCSS 2で構成することを考えてみよう。方法は至って簡単。リスト4のようなスタイルを設定するだけで、このスタイルでは、まずbody要素の左マージンを取って、メニューを置くためのスペースを確保する。次に、メニューとなるdiv要素の幅 (width) をbody要素のマージン以内に設定し、positionプロパティをfixedに、topとleftプロパティを0にして、メニューを画面の左上に固定するのだ。

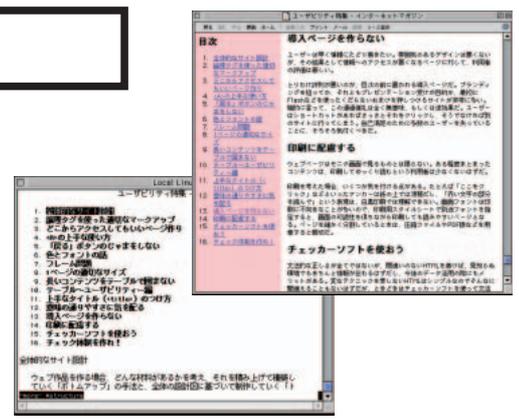
これで、本文をスクロールしても移動しない、フレーム風のメニューが実現する。スタイルシート未対応のブラウザではメニューと本文が続けて表示されるだけだ。さらに、このメニュー部分を別ファイルに保存して、

```
<object data="menu.html"
type="text/html">
```

のようにobject要素として読み込んで利用すると、フレームの場合と同様に、一度読み込んだメニューを複数文書で繰り返し使えて効率が良い。ただし、object要素を完全に実装したブラウザはほとんどないので、まだこの方法はきちんと機能しないようだ。

```
body { margin-left: 15em }
div.menu { width: 14em; position: fixed; top: 0; left: 0;
margin: 0; padding: 0.5em; background-color: #fcc }
```

リスト4



マック版IE 5 (上) では画面左に固定メニューが表示され、Lynx (左) では単純にこのメニューが画面の始めに表示される。CSS 1 のみに対応しているブラウザでは中途半端な結果になるので、今すぐには使いにくい。



アクセシビリティは代替テキストから

# 誰もが使えるページを目指せ

意義はわかっていながらも、どのように実践したらいいのかわからないのがアクセシビリティ。具体的な例さえ見れば、「ああなるほど」と納得できるのではないだろうか。まずは画像に代替テキストを用意するところから始めてみよう。(神崎正英)

## I これがaltの正しい書き方

アクセシビリティとは、どんな利用者にも同等の内容が伝わるように配慮することだ。文字ブラウザ、音声合成ブラウザ、携帯端末など、ウェブを利用する形態はさまざま。環境が違って、同じ情報が伝わるように配慮する必要がある。

簡単なのはコンテンツを文字だけで表現することだが、あまり現実的ではない。利用者を惹きつける魅力的なコンテンツのためには、画像や動画などマルチメディアを駆使してページを作ることも多いだろう。

アクセシビリティの基本は、こうした技術を用いる場合でも、その技術が使えない利用者のために「同等の内容」を表現する代替手段(alternative)を提供することだ。代表的な代替手段であるimg要素のalt属性を例に詳しく見ていこう。

### 1. 写真

たとえば写真をページに組み込んだとする。この場合alt属性をどのように記述すれば「同等の内容」になるだろうか。

`alt="楽器の写真"`

これはよく見かける例だが、わかるのは、そこに写真があるということだけだ。写真からどんな情報が得られるかということとは伝わらない。写真で伝えたいことを簡単に描写して、

`alt="人間より大きな楽器がある!"`

と書くほうが適切だ。alt属性がきちんと「同等の内容」を示しているかどうかは、Lynxなどの文字ブラウザで表現してみればよくわかる。

### 2. ロゴマーク

会社のロゴマークにalt属性を付けるときには、

`alt="ロゴマーク"`

と書いてしまいがちだが、これは

`alt="株式会社インプレス"`

でなければ意味がないことはわかるだろう。

### 3. 飾りの画像

ラインや飾りのグラフィックの場合ほとんどな代替情報を提供すればよいのだろうか。文字ブラウザでの表現を考えると、次のような記述がよさそうだ。

`alt="-----"`  
`alt="**"`

要するに、画像を使って伝えようとすることを、文字で置き換えたらどうなるかを考えてみるということになる。

### 4. 意味のない画像

純粋なアイキャッチャーのような画像が使われることも少なくないが、無理に文字

で置き換えてもあまり意味のある情報にはならない。この場合は、

`alt=""`

とすれば、テキストブラウザではすっきりし、余計な混乱を与えずにすむ。「alt属性の値が空」ということと、「alt属性がない」ということはまったく違うので要注意だ。alt属性がないと、文字ブラウザや音声ブラウザは[image]のような表現をしたり、ファイル名を示したりしてしまう。どんな場合でもalt属性は必須なのだ。

### 5. title属性を使う

文字ブラウザや音声ブラウザは数が少ないからaltを書くメリットはない、と思ったら大間違い。画像を扱えるブラウザでも、利用者が画像を読み込まない設定にしていることがあるからだ。その場合でもalt属性があれば文字が画像の場所に表示される。

「alt="装飾用イメージ"」のような画像そのものの説明は代替テキストには適さない。しかし、グラフィカルなブラウザを使っていて画像を読まない設定の人には、代替テキストが画像を手動で読み込むかどうかの判断材料になるという意見もある。こうした場合には、alt属性ではなくtitle属性で説明を加えるのがいいだろう。最新のブラウザはtitle属性の内容をツールチップに表示するので、これをヒントにして利用者は判断できる。併せて「alt=""」を設

## 2 略語には正式名称を付ける

略語は専門外の読者には理解しにくいので、abbr要素を使って説明を加え、すべての利用者に親切なページにしよう。たとえば、

```
<abbr
title="Uniform Resource Identifier">
URI</abbr>
```

といった具合だ。グラフィカルなブラウザならこの部分にマウスを持っていくと、ツールチップにtitle属性の内容が表示され、音声ブラウザならその説明を読み上げることもできる。

よく似たものにacronym（頭字語）要素がある。NATO（ナトー）のように、

WWWにおいて世界の情報にアドレスをつけるためにバーナーズ＝リーが設計したのが、URIです。WWWの3つの技術のうち、HTTPとHTMLはほかの手段で置き換えることも可能ですが（たとえばFTPによる転送や、XMLによる文書）、アドレス指定の方法はURI以外にUniform Resource Identifier（WWWを実現するための最も基本的な技術）なのです。

定しておけば、文字ブラウザや音声ブラウザの利用者には、無意味な説明を与えなくても済む。

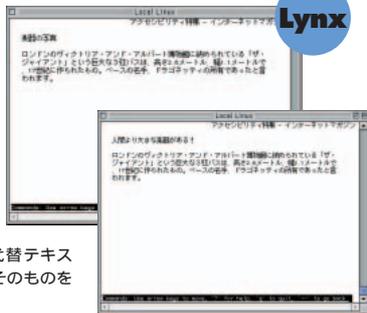
alt属性の意味は、頭でわかっていても、実際にその内容が表示されないとピンと来ない。ウェブ開発者は、Lynxなどを用意

して、ページがテキストだけのときはどんなふうに表示されるのかを随時確認してみるべきだろう。

### 1. 写真



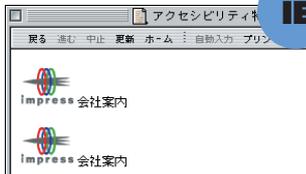
IE



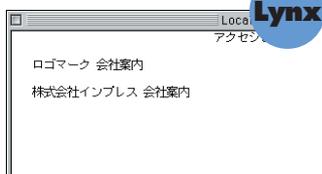
Lynx

この写真に対して、単に「写真がある」という代替テキストでは意味がない。画像で伝えようとした内容そのものをテキストに置き換えて伝える。

### 2. ロゴマーク



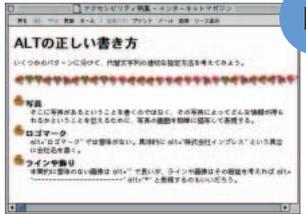
IE



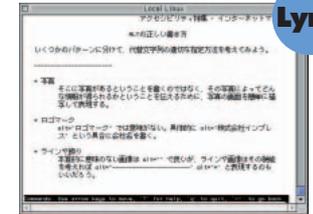
Lynx

「ロゴマーク」と説明されても、情報としては無意味だ。ここで伝えようとする内容は、具体的には会社の名前だろう。

### 3. 飾りの画像



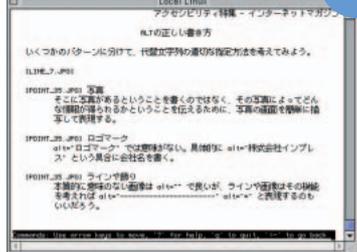
IE



Lynx

情報として必要のない画像は「alt=""」でもいいが、簡単な記号に置き換えておけば、テキストブラウザでもそれなりのアクセントを与えられる。

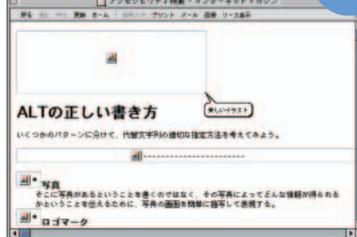
### 4. 意味のない画像



Lynx

飾りの画像の例で、altを指定しなかった場合のLynxの画面。alt属性がないと、何も表示されないのではなくファイル名が代わりに表示される。

### 5. title属性を使う



IE

「alt=""」としてtitle属性で説明を用意しておく、マウスを載せたときにヒントが表示されるので、画像をオフにしたブラウザではファイルを表示させるかどうかを判断できるし、テキストブラウザでは邪魔にならない。

## 3 イメージマップにも alt を付ける

代替テキストが必要なのは、通常の画像だけではなく、映像や音声、アプレットなど、テキストでない内容にはすべて等価な内容の代替テキストを与える。イメージマップも同様だ。img要素、map要素、area要素でマップを構成するとき、img要素とarea要素には、画像を表示しないブラウザのためにalt属性を指定する必要がある。

img要素とarea要素にalt属性を記述した場合としない場合の表示をテキストブラウザで比較してみよう。代替テキストがあれば、Lynxもイメージマップをきちんと扱える。ただし、画像をオフにしたグラフィカルなブラウザでは、こうしたイメージマップは使いにくいので、テキストリンクも別に用意した方がよい。

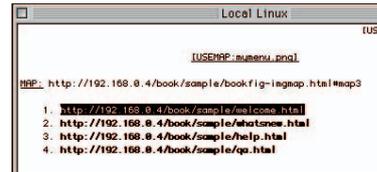
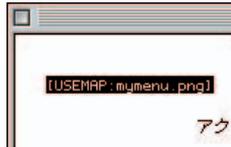
リスト5のようにHTML 4のobject要素とa要素によるイメージマップを使えば、イメージマップとテキストリンクを別々に用意しなくても一度に両方の手段を提供できるが、対応しているブラウザはiCabなどまだ少ない。

リスト5

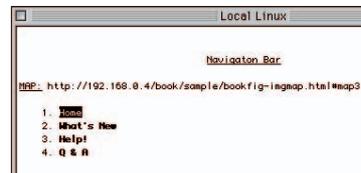
```
<object data="mymenu.png" type="image/png" usemap="#map1" width="480"
height="40"></object>
<map name="map1" id="map1">
<p>
<a href="welcome.html" shape="rect" coords="10,10,115,34">Home</a> |
<a href="whatsnew.html" shape="rect" coords="127,10,232,34">What's New</a> |
<a href="help.html" shape="rect" coords="247,10,352,34">Help!</a> |
<a href="qa.html" shape="rect" coords="364,10,470,34">Q &amp; A</a>
</p>
</map>
```

object、map、a要素を組み合わせたイメージマップ。map要素をobject要素の内容にすれば、画像オブジェクトが扱えないときだけテキストリンクを表示する。object要素の外に出すと、マップと併せてテキストリンクも示される。リンク先のメンテナンスは、a要素を修正するだけでよい。

altなし



altあり



area要素とimg要素に適切なalt属性を与えると、文字ベースのブラウザ(Lynx)でもきちんとイメージマップを利用できる。

## 4 スクリプトとプラグインにも代替手段を

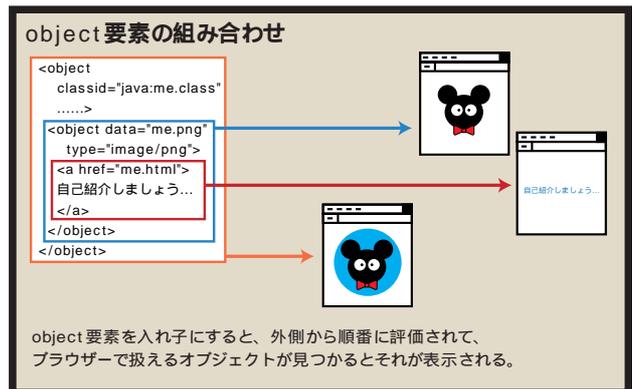
ウェブの応用範囲はどんどん広がっている。対話性を持たせたり多様な表現を行ったりする技術として、クライアントサイドスクリプトや埋め込みオブジェクトの重要性は増していくことになるだろう。その技術を活用するのは結構だが、利用者はさまざまな機器やブラウザでページを閲覧していることを忘れてはならない。さらに、セキュリティのためにこうした機能を無効にしている利用者も多いことに留意すべきだ。

画像の場合と同様に、新しい技術を使うときにも代替手段の用意が欠かせない。script要素でダイナミックにHTMLを生成するのなら、スクリプトを理解しないブラウザのためにnoscript要素でもHTMLを記述する必要がある。また、リンクを<a

href="javascript:.....>という形にして、スクリプトで仕掛けを施してからジャンプさせるページがあるが、これは避けるべきだ。スクリプトが無効になっていると、機能しないリンクになってしまう。

埋め込みオブジェクトの場合には、図のようにobject要素を入れ子にすることでプラグインがだめなら静止画、それもだめならテキストと、段階的に代替コンテンツを提供できる。ただし、object要素で

正しくプラグインを扱えるブラウザは少ないから、embed要素を使いつつ、別のページで静止画像版を用意するほうが現状では適切かもしれない。



W3Cでは、障害者や限定的な環境でもウェブを十分活用できるような技術や表現方法を推進するために「The Web Accessibility Initiative」(WAI)という活動を行っている。なかでもウェブ制作者にとって重要なものが、1999年5月に勧告された「アクセシビリティの高いコンテンツのための指針」(Jump01)だ(日本語訳には大藤幹氏のもの(Jump02)などがある)。

この勧告では、基本原則として(1) Graceful Transformation (環境に応じて適切に表現される優しいコンテンツ)、(2) Understandable and Navigatable (理解しやすく移動しやすいコンテンツ)の2

点を挙げ、14の指針によって注意点を説明している。その多くはこの特集でも取り上げたが、ほかにも

- ・古いブラウザでもコンテンツを利用できるように暫定措置を設ける。
- ・マウスなど特定の機器を前提としない
- ・プラグインなどの内部のメニューもアクセシビリティに配慮する。

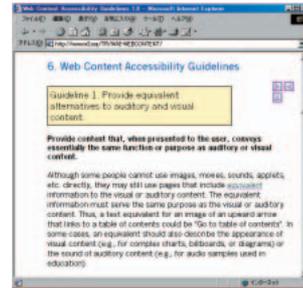
などがある。それぞれの指針には具体的なチェックポイントが示され、おのおの1~3の優先度が与えられている。たとえばalt属性の使用などが優先度1の必須事項だ。

すべてのポイントを完全にクリアするのはハードルが高いが、ウェブ制作者ならば

少なくとも指針に目を通し、優先度1のものから順次取り入れていこうにしたい。

Jump01 [www.w3.org/TR/WAI-WEBCONTENT](http://www.w3.org/TR/WAI-WEBCONTENT)

Jump02 [www.zspc.com/doc/accessibility/](http://www.zspc.com/doc/accessibility/)



## 3 本当の配色テクニックとは

赤と緑を区別しにくいような色覚を持つ人は、日本に約300万人いると言われる。男性はその割合が高く、20人に1人はこうした色の感じ方が異なるそうだ。インターネットなどで公開されている体験談を読むと、多くの人が地下鉄の路線図のような、微妙な色使いによる情報を判別するのが難しいと述べている。

区別しにくい色の代表的な組み合わせは赤と緑とされるので、少なくとも赤と緑の色差だけに頼ったデザインは避けるべきだろう。ただし、灰色と水色、ピンクと灰色など人によってほかにも混同しやすい配色があり、見え方はさまざまだという。1つの方法は、色相(赤、緑など)だ

けでコントラストを付けるのではなく、それぞれの明度も違えて配色するという工夫だ。また、グラフなら色だけでなく網掛けなども加えて領域を区別するなど、色だけに頼らない表現を心がけることが重要だ。



チャートを示す場合、色分けだけでなく塗りパターンも加えると、色による区別が困難でも違いが明確になる。

## 6 単語内にスペースを入れない

日本語を簡単に整形するためによく使われる手段が、全角スペースを挿入して文字の位置をそろえるというものだ。便利に使っている人も多いだろうが、

<li>経 済</li>  
<li>ス ポー ツ</li>

という具合にスペースを入れた場合、音声読み上げソフトは最初の項目を「けい、すみ」と発音してしまう。単語が単語として認識されなくなってしまうのだ。この場合はスタイルシートで「text-align: justify」や「letter-spacing: 2em」などのプロパティを使って調整できる。

## 7 動くコンテンツは慎重に

以前に人気アニメで画面が激しくフラッシュする場面で、多くの視聴者が卒倒して問題になった。これは極端な例だが、人間にとって点滅したり動いたりするものは特殊な刺激となる。多くの人にはこれらを不快に感じるし、障害のある人にとっては動く

テキストは読むことができない場合がある。意味もなく画面にそうした効果を持ち込まないほうがいい。blinkとかmarqueeといったタグは、そもそも標準のHTML仕様には含まれない。

アニメーションGIFも最初のうちは面白

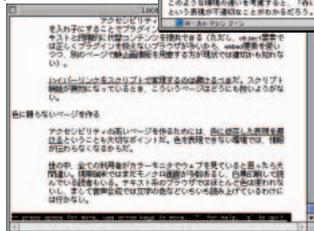
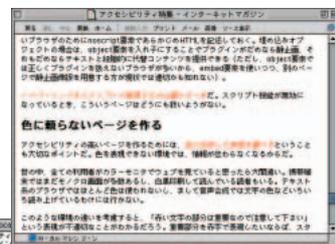
いが、すぐに飽きられてわずらわしくなる。画面にちらちら動くものがあると、本当に重要な内容に集中してもらえないことも認識しておこう。

## 色に頼らないページ作り

世の中すべての利用者がカラーモニターでウェブを見ていると思ったら大間違い。携帯端末ではまだモノクロ画面が多数あるし、白黒印刷して読んでいる読者もいる。テキストブラウザではほとんど色は使われないし、まして音声合成では文字の色などいちいち読み上げているわけにはいかない。このような環境の違いを考慮すると、

「赤い文字の部分は重要なので注意して下さい」という表現が不適切なことがわかるだろう。重要部分を赤字で表現したいなら、スタイルシートを使って「em { color: red; }」と設定し、em要素でマークアップする。そうすれば、色が表現できなくてもその要素が強調されていることは伝わる。

少し視点は違うが、濃色の背景画像に淡色の文字を指定すると、画像をロードしない設定の人に文字が読めなくなることに注意。背景画像を指定した場合は背景色も指定する必要がある。



スタイルシートでem要素の文字色を赤に設定すると、スタイルシート対応ブラウザでは強調部分が赤文字で示され、非対応ブラウザでも、ちゃんと強調要素として扱われるので、同等の情報を伝えることができる。

## 言語指定の使い方

国際化の時代にはHTML文書でも多言語を混在させるケースが出てくる。ブラウザが言語を判断できるように、lang属性で言語コードを指定しておく、多言語の文書がうまく扱えるようになる。

通常は、html要素で

```
<html lang="ja">
```

という具合に基本言語を指定しておき（jaは日本語）、異なる言語を使う部分だけ

```
<p lang="fr">
```

などと言語指定を上書きする（frはフランス語）。これによって、音声ブラウザはその部分を適切に発音できるし、視覚ブラウザは適切なフォントや引用符、単位記号を使って画面表示できるようになる。

## キーボードで便利なページに

マウスではなくキーボードを使ってブラウザを操作する環境も少なくない。こうした場合に、フォームの項目などの選択を容易にするために導入されたのがtabindex属性とaccesskey属性だ。

tabindexはリンクやフォームの選択順序を数字で設定する。特殊な要素を飛ばして必須入力項目だけをTABキーで選択できるようにするなどの使い方ができる。

accesskeyはキーボードショートカットを実現する機能だ。

```
<a href="help.html" accesskey="h">
```

としておくと、Hキーでヘルプページを表示できる。ショートカットのためのキーの組み合わせがOSやブラウザによって異なるなどまだ不統一な段階だが、これが普及すると、テキストブラウザだけでなくグラフィカルなブラウザの利用者にとっても便利なものになるだろう。

## テーブルも優しく

テーブルを使うと複雑な情報を一目で把握できるようになるが、これは視覚的にテーブルを認識できる場合の話だ。音声合成でテーブルを読み上げる場合は、全体の構造が頭に入っていないとわかりにくい。

テーブルのアクセシビリティを高めるためには、(1) テーブルの構造や全体像についての説明を提供すること、(2) 項目名とデータのセルを明確に区別してその関係

がわかるようにすること、の2点が大切だ。テーブルの説明をするには、table要素のsummary属性を使う。

```
<table summary="WWWの利用の変化を示す表。1列目に年度、2列目にホスト数、3列目にサーバー数を示している">
```

と記述すると、テーブル本体が読み上げら

れる前に構造をイメージできるので、理解が容易になる。レイアウト目的の場合は

```
<table summary="レイアウト用テーブル">
```

と説明すれば、混乱を招かずに済む。

項目名とデータの区別は、th要素とtd要素を正しく使うということだ。これがきちんと使い分けられているだけで、テーブルの意味はずっと明確になる。

## topics

### リンクに許可はいらぬ

ウェブサイトがよく見かけなのが「無断リンクは禁止します」「このサイトはリンクフリーです」という注意書きだ。はたしてこうした言葉には根拠があるのだろうか。

WWWの発明者であり、現在W3Cのディレクターであるティム・バーナーズ＝リーは、リンクに関する文章をいくつか公開している。「リンクの神話」<sup>Jump01</sup>では、「リンクは著作権を侵害する」「リンクには対価を求めることができる」「リンクはプライバシーの侵害になる」という3つの意見を誤解であると語り、右の囲みのような原理に立たなければハイパーテキストの世界は成り立たないと主張している。

一方、同じバーナーズ＝リーの「リンクと法」<sup>Jump02</sup>では、リンクをa要素による

「通常のリンク」とimg要素などによる「埋め込みリンク」に分けている。通常のリンクそれ自体は単なる参照だ(作者に責任があるのは「このページをご覧ください」などの言葉)。しかし、埋め込みリンクでは、埋め込まれた画像はページの一部と見なされるため、作者にはその画像に対する責任が発生する。また、フレームについては、文書の持ち主について混乱が起きるようなデザインは避けるべきだと述べている。

HTMLの機能によって権利の問題が発生する場合はあるが、a要素による通常のリンク自体は言論の自由に基づいて自由に行われるべきだというのが、WWWを作った人物の意見だ。リンクに関して細かい注意などは書かないのが、正統派のサイト作者の態度だと言えるだろう。

<sup>Jump01</sup> [www.w3.org/DesignIssues/LinkMyths.html](http://www.w3.org/DesignIssues/LinkMyths.html)

<sup>Jump02</sup> [www.w3.org/DesignIssues/LinkLaw](http://www.w3.org/DesignIssues/LinkLaw)

#### ティム・バーナーズ＝リーの主張

- ・ほかのサイトにリンクを張る前に許可を求めなければならない理由はない。
- ・言論の自由がなくなる限り、「参照されない権利」は認められない。
- ・公開された文書にリンクを張ることは、プライバシーの侵害ではない。

## topics

### IBM ホームページ・リーダーを使ってみた

「アクセシビリティ」の意義はわかった、alt属性の使い方もわかったという人でも、IEやナビゲーターで毎日ウェブを見ている現状では、いまいち実感がわからないのではないだろうか。「音声読み上げブラウザ」なんて遠い世界のこのように感じるかもしれない。そういう人には、IBM ホームページ・リーダーを使ってみることをおすすめする。購入するには高めの値段だが、お試し版はIBMのサイトからだれでもダウンロードしてインストールできる。対応しているOSは、ウィンドウズのみ。実行にはネットスケープナビゲーターがインストールされていることが必要だ。

インストールして実行すると、案内のページが合成音声で朗読される。テンキーの「+」と「.」を順番に押し、URLを入力してみよう。そのページにアクセスして内容が読み上げられる。通常のテキストは男性の声だが、リンクのテキストは女性の声に切り替わる。テキストの読み取りはかなり正確で、英単語が混じっていてもきちんと発音してくれる。ただし、苦手な部分も

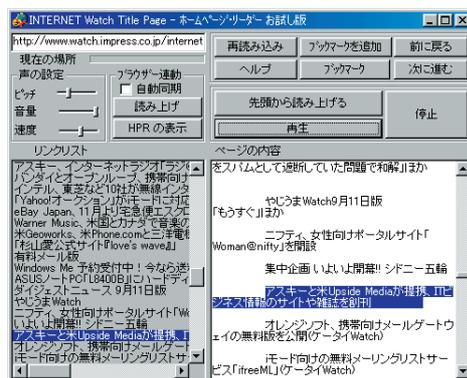
あるようで、日付の「2000/9/8」は「きゅうぶんのにせん、はち」となった。

ホームページ・リーダーは、基本的にキーボードのテンキーだけで操作できるようになっている。たとえば読み上げを止めたければ、「Enter」を押し、再開するには「0」を押す。リンクだけを順番に読みたければ「3」を押し、現在のリンク先に飛びたければ「2」を2回押す、といった具合だ。操作をマスターすれば、すばやいブラウジングも可能になるだろう。

ホームページ・リーダーでウェブを読んでみると、いかに多くのサイトが使いがらみかを実感できたり、意外にどんなサイトでもそれなりに読めてしまうものかということがわかったりする。自分のサイトの作成にも参考になることが多いに違いない。最先端の技術でウェブの未来を体感するために試してみるのがいいだろう。

ホームページ・リーダー Windows版 V2.5  
価格：15,000円

<sup>Jump</sup> [www.ibm.co.jp/accessibility/soft/hpr.html](http://www.ibm.co.jp/accessibility/soft/hpr.html)



Internet Watchを読んだところ。左側のメニューを抜けて右側のニュース一覧にたどり着くまで結構時間がかかることがわかった。



あの企業は正しく書いているか

# 有名サイトHTMLチェック!

ウェブが公的なメディアとなっている現在、企業や官公庁にとって、ユーザビリティやアクセシビリティを考慮したサイトを作ることは義務とも言える。著名な企業7社と首相官邸、それにインプレスのサイトをチェックしてみた。(ソシオメディア+編集部)

## Yahoo! Japan

91点



Jump www.yahoo.co.jp

採点：9つのサイトの中ではもっとも軽い。alt属性もほとんどの画像に付いている。文法にはやや妙なところも。

総評：見るためではなく使うためのページを意識しているようで、テキストベースで構成され、シンプルで表示も速い。また、カテゴリの分類法から自分の探したいジャンルを見つけやすい構造になっている。プロモーション、メニュー、ディレクトリ

リーという大きなエリア分けも明解さを増している。

ただし、メニュー(ニュース、掲示板、オークション)の数が多すぎることや、クリック可能なエリアが多すぎるために、クリックできない部分とのコントラストに欠ける感がある。また、ほとんどがリンクテキストであることから、すべてに下線がついていて見づらい側面も。

利用者の求める利用方法を明確化しているの、その意味では利用者の満足度が高いサイトであることがわかる。

## 楽天市場

88点



Jump www.rakuten.co.jp

採点：文法チェックの点数がもっともよかった。スクリプト未対応のブラウザでもすべてのメニューが使えるのがよい。

総評：テキスト中心でダウンロード時間も短く、ストレスはない。検索フィールドがわかりやすいのもよい。ページデザインでは、ウィンドウサイズによって横幅が可変になっているが、崩れることなく表示できる。

しかし、メニューの切り口の多さ、クリックできそうでできないもの多さ、似たようなリンクの多さなどから、サイトの全体構成が予測できない。デパートというよりは、さながら雑居ビルのような。これを解消するためにも、機能別のエリア分けを明確にしたり、タブ型の要素をもっと効果的に利用したりするなどの工夫が必要ではないか。

運営側が提示したい情報と、ユーザーが自由に利用できる分類メニューとが混在しており、情報に対する信頼性が低く感じられるのが残念。

## asahi.com

40点



Jump www.asahi.com

採点：サイズが大きいのは、サイトの特徴からしてやむを得ないか。イメージマップにalt属性が用意されていないのが難点。

総評：最新ニュースが上部に配置されていることから、ページ全体を読み込まなくても最新ニュースを確認できる。また、更新時間が表示されているので速報性がはっきりしている。トップページで焦点を当てる記事のサマリーを1つにしている

のも、ウェブサイト上での読みやすさへの配慮が感じられる。しかし、画像のクオリティーが低いのが残念。減色スキルの向上による改善に期待。また、検索機能がないこともユーザーの使い勝手からみると不便である。

全体のボリュームを把握したり、自分の求める情報が存在するかしないかを判断したりするための基準や仕組みがあると便利ではないか。1つのインターフェイスでターゲットを広げすぎている感がある。

## マイクロソフト

83点



Jump www.microsoft.com/japan/

採点：軽いために高得点。IE用のスクリプトを使ったメニューがあるが、それ以外のブラウザでも一応は使える。

総評：ヘッダー部分や左エリアにサイト全体のメニューや検索エリアが置かれ、ユーザーにとって注目度が高い情報が目立つ位置に配置されているので、わかりやすく、うまくまとまっていると言える。また、少ない画像を効果的に使っており、それ

なりの見栄えを保つことにつながっている。

しかし、リンクテキストのスタイル、マウスオーバーのフィードバックなどが統一されていないところ、横幅の広さ、最終更新日の提示位置、検索についてのヘルプなど、改良すべき点も散見される。上部2つの水平メニューの使い分けも意味不明。

このサイトの問題は、トップページより中に入ってからの部分で、画面構成が統一できていないためにすぐ迷子になることがある。

## NTT東日本

74  
点



Jump [www.ntt-east.co.jp](http://www.ntt-east.co.jp)

探点：総サイズの大きさが点数にひびいている。  
総評：「～したい方はこちら」といった表現を使って、目的指向のナビゲーションを重要視しており、探したい情報にアクセスしやすくなっている。また、各リンクの重要度が、面積によって表現されており、初めて利用する人に直感的に情報の重要性を伝える効果を出している。検索エリアやサイトマップが下方に配置されすぎていること、サイトナビゲーションが明示されていないこと、インターフェイス要素のデザインが不統一であることは、今後の改善点。ターゲットを広く想定しているにも関わらず、画像のみで構成されたトップページであることは、アクセシビリティの観点からは課題となる。現サイトは、女性週刊誌の釣り広告のような印象を受ける。

## 首相官邸

80  
点

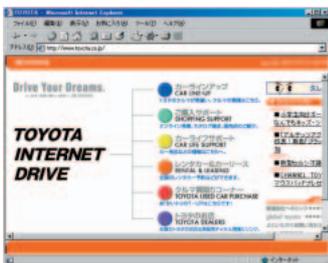


Jump [www.kantei.go.jp](http://www.kantei.go.jp)

探点：官庁らしく、alt属性の使用は完璧だった。  
総評：デザイン自体のテストは5年前のものであるが、カラーリングなどから落ち着いた雰囲気が出されている。タイトルや建物の写真は目立っているが、一瞥して何のためのサイトであるかが不明瞭。エリアごとの情報の枠組みもわかりづらい。各リンクの重要度や分類基準がまったくわからない。写真をクリックするとどこへ行くかすら予想できない。また、「English」がグレーで表示されていることから、まるでクリックを拒絶しているかの印象を受ける。インターフェイス要素の使用方法に一貫性がなく、あらゆる意味でデザインがなされていないが、その無骨なところがえって「首相官邸」らしさを出しており、情報の信頼性に通じるとも取れる。

## トヨタ自動車

69  
点



Jump [www.toyota.co.jp](http://www.toyota.co.jp)

探点：alt属性がまったくないため、点数は低くなった。  
総評：ビビッドな色使いで元気が伝わってくる。各メニューがボタンの色で区別されており、わかりやすい。各メニューに説明が付いているので求める情報を見つけやすくなっている。画面エリアの分割とその意味が明確となっていること、エリア面積や位置関係などから重要性を表していることなどがサイト全体の使いやすさ、わかりやすさを支えている。また、速報、ユーザーの求める情報、企業が発信したい情報、客観的なナビゲーションなど、トップページに求められるナビゲーションの切り口がすべて揃っていて、使い勝手のお手本のようなサイト。全体のカラーリングからは、実際の「トヨタらしさ」と異なる印象を受けるが、そこにウェブ上での戦略を感じる。

## 東京三菱銀行

32  
点

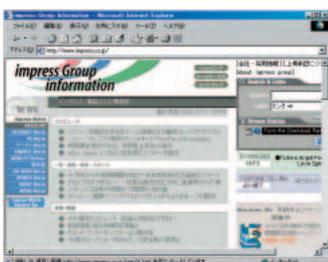


Jump [www.btm.co.jp](http://www.btm.co.jp)

探点：同じくalt属性が少ないため、点数が低い。  
総評：ナビゲーションの点で、左エリアのメニューからロールオーバーでサブカテゴリーが出るのは流行りでもあるがとても便利。また、サイトマップなどはヘッダー部分、メニューは左エリアというように、エリア分けが明確化されているので、初めてのユーザー、リピーターにもすぐにアクションが起らせてわかりやすい。しかし、画像ボタンのフィードバック（行き先など）がバラバラであったり、「New」が連続するなど、情報の重要性やコントラストに欠け、全体にインターフェイスが不統一。検索機能が無いのも疑問。ヘッドラインニュースがコンテンツエリアの中心になっていることで、一般ユーザーが訪れた際の目的意識まで奪ってしまわないか心配。

## インプレス

85  
点



Jump [www.impress.co.jp](http://www.impress.co.jp)

探点：サイズはやや大きめ。画像数が多いのが気になる。  
総評：情報量が多いにもかかわらず、エリア分けによる情報分布ができており、わかりやすいサイトになっている。また、検索機能が目立つ位置にあって、使いやすい。全体的にトーンを抑えながら、使うためのサイトである雰囲気を出している。しかし、不要な凹凸や画像要素が多いこと、ボタンの形状の種類が多すぎることで、リンクテキストの色が不統一であることなどは今後の改善点。また、下半分の情報はトップページで一覧させるには多すぎる印象を受けることから、別ページにするなどの改善が必要。数度にわたる見直しを経て、構成要素を考えてきたことが伝わり、ブランディングに配慮している様子もわかるが、まだ余計なディテールが多く見受けられる。

## 採点基準

各サイトのトップページ(9月18日現在)を評価して、以下の5項目に編集部が0~20点の点数を付けて合計した点数を出した。

HTMLと画像を合わせたサイズ。50KB以内は20点、10KB増えるごとに-1点。

alt属性の有無。alt属性の数 ÷ img要素とarea要素の数 × 20点。

JavaScriptがないと使えない部分があるかどうか。ない場合は20点、1つあるごとに-2点。

文法の正確さ。Another HTML lint(211ページ参照)の点数が50点以上は20点、50点から10点減るごとに-1点。

トップページに検索機能がある場合は20点、サブページに検索機能がある場合は10点。



これまでの常識を見直そう

# こんなページはかっこ悪い。

年月が経つにつれてページ制作も洗練されてくるはずだが、ウェブが流行し始めたころの「ノリ」をそのまま引きずっている部分もまだあるのではないだろうか。「ださい」と言われないために、習慣に従って漠然と作っていないか点検してみよう。(編集部)

## 1 Sorry, Japanese Only ?

「Sorry, Japanese Only」は、日本でウェブが普及し始めたころから受け継がれている代表的な決まり文句だが、いろいろな意味で変なフレーズだ。まず、日本語で書かれているからといって謝る必要があるだろうか。「Sorry, Chinese Only」や「Korean Only」と書いてある中国語や韓国語のページは見かけない。また、英語圏のユーザーは、「Sorry, ...」という英語が書けるくらいなら、なぜコンテンツも少しは英語にしてくれないのか、と思うのではないだろうか。さらに「Japanese Only」のあとに「What's New」という見出しがあると、ますます奇妙なページになる。

外国のユーザーのことも気になるなら、英語のページをきちんと用意したほうがいいだろう。また、日本語でページを直接表示させて情報をつかみたい外国人もいるだろう。ヤフー([Jump www.yahoo.co.jp](http://www.yahoo.co.jp))のトップページから「韓国」や「中国」を選んだときのように、日本語フォントの表示方法を紹介してあげると親切だ。



日本語の情報を直接求めている人のために、マイクロソフトのGlobal IME([Jump www.microsoft.com/windows/ie/features/ime.asp](http://www.microsoft.com/windows/ie/features/ime.asp))などを紹介するのも手だ。

## 2 「～でご覧下さい」はもう古い

「Netscape 何々以上でご覧下さい」「IE 何々以上でご覧下さい」と書いてあるページは多い。さらに、ブラウザのロゴ画像を貼り付けているページもある。これは、1995年ごろにネットスケープ社が次々と新機能を取り入れたブラウザを発表し、それが熱狂的に支持されていた時代の流行だ。現在では、ブラウザの最新機能をやみくもに取り入れることは避けられる傾向にある。また、クライアント側よりもサーバー側に機能を持たせて、どんな環境でも活用できるサイトを作るのがトレンドだ。

いろんな環境でテストしていないことが気になるかもしれないが、どんなブラウザでもテキストだけは読めるものだ。CSS

を慎重に使えば、古いブラウザでも便利に使えて新しいブラウザなら見栄えのいいページは作れる。もし自慢のデザインが崩れるから自分が使っているブラウザだけで見てほしいというなら、それは利用者に失礼な話だ。

携帯電話に代表されるように、これからの時代はウェブを閲覧する環境が多様化する。できるだけ多くの人をターゲットにしたサイトを目指すなら、ブラウザの独自機能よりも標準的な文法に気を使うべきだ。特定の層をねらったサイトならブラウザを限定する意味もあるだろうが、公共性の高い官公庁のサイトでブラウザの指定を見かけるのは困ったものだ。

## 3 「工事中」と書かない

「Sorry, Japanese Only」のように、深く考えずについつい使ってしまう決まり文句がいくつかあるが、「Under Construction」や「工事中」もその1つ。サイトは必ず未完成な状態で公開するという決まりがあるわけではなし、この習慣もそろそろなくなってもいいのではないだろうか。工事中だらけのページはみっともないし、期待してクリックしたリンク先が「工事中」となっていたら、利用者はユーザービリティに欠けたサイトだと評価するだろう。コンテンツをすべて完成させてからサイトを公開するか、十分な内容を用意してから「さらに以下のようなコンテンツを今後お届けする予定です」のように書けばいいだけの話だ。

## 4 ユーザーへの指図は悪印象

「800x600以上でご覧下さい」のように書かれているページがよくあるが、これは非現実的なお願いだ。特定のサイトを見るためにブラウザを新しくインストールしたり、パソコンを買い換えたりする人はいない。また、現状では大半の利用者が解像度の高いモニターでウェブを利用しているからと言って、指定のサイズでページを見

てくれるわけではない。利用者はウィンドウのサイズを自由に変更できる。

もちろん特定のサイズに合わせてサイトのデザインを統一するのは悪くないが、利用者に押し付けるようなお願いを表記するとサイトの印象は悪くなる。サイズに限らずブラウザやプラグインの指定などにも言えることだ。

## 5 古いテクニックは使わない

JavaScriptが初めて登場したころは、できることが少なかったので、限られた機能を活用しようとして流行したのがステータスバーを使った文字のアニメーションだ。ダイナミックHTMLをサポートしたブラウザのシェアが高くなった今となっては、このテクニックは時代遅れに見える。そもそもステータスバーは、リンク先のURLなど

の重要な情報が表示される場所であり、表示を勝手に書き換えるとページの使いやすさが低下する。

テキストボックス（input要素）を使ったアニメーションも同様に時代遅れのテクニックだ。テキストボックスはフォームの入力用部品であり、アニメーション効果に使うものではない。

## 6 スタイルシート のやりすぎに注意

スタイルシートを使えば、柔軟でアクセシビリティに優れたページが作れるという認識は広まってきたようだが、スタイルシートさえ使えば何をやってもいいと勘違いしてはならない。スタイルシートではサイズを細かい単位で指定できるので、つい凝ってしまいがちだが、フォントサイズ大きくしすぎたり小さくしすぎたりすると読みにくいページになる。長い文章にはサイズを指定せず、利用者の設定に任せようがよいだろう。また、むやみにたくさん色を指定したり、書体を変更したりすると、品のないデザインになる。テクニックに走らずに、スタイルシートを使うときにもユーザビリティを念頭に置こう。

## topics ウェブサイト成功のためのポイント

ウェブ制作には、HTMLの書き方からマーケティングまで幅広い知識が必要だ。成功するウェブサイトを作るためには、これから重点的に何に取り組んでいけばいいのかまとめてみよう。

### 1 ページのデザイン ◎ サイト全体のデザイン

ロゴやトップページをデザインしただけで満足してはいないだろうか。利用者はサイトの表紙ではなく、そのサイトで何ができるか、何が得られるかを評価する。個々のページだけでなく、サイト全体をどのように機能させるかをよく考えて設計しよう。ヤフーのように、サイト紹介のカテゴリーのページからそのジャンルの掲示板やショッピング、オークションのページにも飛べる作りは大きなヒントになるだろう。

### ブラウザ上の技術 ◎ サーバー上の技術

検索機能のないニュースサイトを積極的に使おうとする人はいるだろうか。また、商品を並べるだけで、ブラウザからは注文できないサイトをショッピングサイトと呼べるだろうか。これからはますますサーバー上の技術が重要になる。サイト構築を仕事にしようとするなら、HTMLや画像の作成技術だけではやっていけないだろう。サーバー上のプログラム作成まで含めたトータルなコーディネート能力が求められる。

### クールなサイト ◎ ユーザーの立場に立ったサイト

ウェブサイトでは、リアルなお店と違ってお客は並んで待ってくれたりしないし、テレビのように座って見てくれるわけではない。満足のいく結果が得られないと、すぐにほかのサイトに逃げてしまうだろう。いくらすばらしいと自信を持って提示しても、利用者に受け入れられるかどうかはわからない。制作者の主観だけに頼らずに、ユーザビリティやアクセシビリティを客観的にチェックできる制作スタイルが必要だ。



## [インターネットマガジン バックナンバーアーカイブ] ご利用上の注意

このPDFファイルは、株式会社インプレスR&D(株式会社インプレスから分割)が1994年～2006年まで発行した月刊誌『インターネットマガジン』の誌面をPDF化し、「インターネットマガジン バックナンバーアーカイブ」として以下のウェブサイト「All-in-One INTERNET magazine 2.0」で公開しているものです。

<http://i.impressRD.jp/bn>

このファイルをご利用いただくにあたり、下記の注意事項を必ずお読みください。

- 記載されている内容(技術解説、URL、団体・企業名、商品名、価格、プレゼント募集、アンケートなど)は発行当時のものです。
- 収録されている内容は著作権法上の保護を受けています。著作権はそれぞれの記事の著作者(執筆者、写真の撮影者、イラストの作成者、編集部など)が保持しています。
- 著作者から許諾が得られなかった著作物は収録されていない場合があります。
- このファイルやその内容を改変したり、商用を目的として再利用することはできません。あくまで個人や企業の非商用利用での閲覧、複製、送信に限られます。
- 収録されている内容を何らかの媒体に引用としてご利用する際は、出典として媒体名および月号、該当ページ番号、発行元(株式会社インプレス R&D)、コピーライトなどの情報をご明記ください。
- オリジナルの雑誌の発行時点では、株式会社インプレス R&D(当時は株式会社インプレス)と著作権者は内容が正確なものであるように最大限に努めましたが、すべての情報が完全に正確であることは保証できません。このファイルの内容に起因する直接のおよび間接的な損害に対して、一切の責任を負いません。お客様個人の責任においてご利用ください。

このファイルに関するお問い合わせ先

**株式会社インプレスR&D**

All-in-One INTERNET magazine 編集部

[im-info@impress.co.jp](mailto:im-info@impress.co.jp)