

Web制作のコアカリキュラム

小松香爾

はじめに

本論文では、大学のWeb制作のカリキュラムを論じる。Web制作の業界は非常に動きが早い。まず、Webページが配信されるWorld Wide Web（以下WWW）とシステム自体が新しい技術である。OSやオフィスソフトで圧倒的なシェアを取ったマイクロソフト社のような存在もない。1990年のWWW誕生以来、その利便性によって、インターネットは一気に商用利用されるようになった。ヤフー、グーグルなどのベンチャーが登場し、Web業界にイノベーションを起こした。だが、インフラとしてはインターネット、ページ記述言語としてはHTMLが使われていることは不変である。CSSなどのように、徐々に浸透していった技術もあれば、JavaScriptのように、突然復活した技術もある。また、テーブルレイアウトのように、時代遅れといわれながらも残り続けている技術もある。Webの世界は、そのステークホルダーの多さから、常に混沌としている。このような分野においては、何を教え、どのように演習するかについての指針が必要である。第1節では、WWWとインターネットの理論面、第2節ではWeb標準に則したWeb制作、第3節ではWeb制作演習の内容、第4節では、HTML5で付け加わった新機能、第5節では、Webを制作する際の著作権について論じる。

1 WWWとインターネット

Webの世界は移り変わりが激しい。しかし3つだけ変わらない技術がある。1970年代から使われていたインターネット、インターネットにあるクライアントとサーバ、クライアントサーバモデルに実装されたWWWというアプリケーションである。HTMLという言語で書かれたWebページをサーバに置き、世界中のクライアントからアクセスする技術がWWWである。

1-1 WWWに関する誤解

WWWとインターネットは同じものがあると誤解している学生が存在する。ホームページをインターネットと誤解させるようなメディア広告にも問題があるが、それ以上にWWWの解説方法に問題がある。入門者むけの解説書において、WWWは「世界中にはりめぐらされた蜘蛛の巣である」と説明されることが多い。もちろん、この表現は間違いではない。しかし、インターネットも世界中のローカルエリアネットワークをつないだ蜘蛛の巣のようなネットワークなのである。したがって、「世界中にはりめぐらされた蜘蛛の巣」は、WWWとインターネットを混同させる表現として忌避すべきである。全てのネットワークはグラフ構造であり、ネットワークを基盤にして実装される全てのアプリケーションもまたグラフ構造になる。グラフ構造を

「蜘蛛の巣」というアナロジーを用いて説明するのは妥当である。しかし、その場合、基盤としての「蜘蛛の巣」とアプリケーションとしての「蜘蛛の巣」を混同しないように解説すべきである。なぜなら、基盤技術と基盤の利用技術はレベルが異なるからである。WWWとインターネットが同じものであるという誤解を解くためには、インターネットが、汎用性を持つ情報インフラであることから解説すべきである。

1-2 インフラとしてのインターネット

インターネットが情報をやりとりするためのインフラであることを理解させることは容易である。現在の学生は、身近な情報端末として、ケータイを使用している。第2世代以降のケータイ、すなわちデジタル化されたケータイは、もはや単なる電話機ではない。パケット使い放題プランの登場により、メールやWebといったデータ通信がメインの用途になったといえる。したがって、ケータイのデータ通信も、インターネットを利用していると解説すればよい。そして、インターネットが急激に発展し、情報社会を支えるインフラとなった理由を解説する。インターネットはありとあらゆる情報をやりとりできる。それが汎用性であり、エネルギー網や交通網などの専用インフラとの違いを決定的にする性質である。この汎用性は、プロトコルスタックにより実現されている。また、インターネットの回線網はダムネットワークである。すなわち、ネットワーク自体は単純であり、エラー処理や、通信スピードの決定はエンドノードであるサーバやクライアントマシンで行われる。ダムネットワークであるがゆえに低コストである。全体を管理する組織がないに等しいこともあり、インターネットは自己増殖的に広がった。このエンドツーエンドの原理もプロトコルスタック抜きには説明することはできない。インターネットのプロトコルスタックは四層からなる。第一層は物理層、第二層はインターネット層、第三層はトランスポート層、第四層はアプリケーション層である。しかし、この四層構造を解説する前に、電話に代表される回線交換方式とパケット交換方式の違いについて説明すべきである。回線交換方式と比較することで、パケット交換方式の特徴と、プロトコルスタックの必要性が浮かび上がるからである。

1-3 インターネットのアドレス解決

電話をかける際には、電話番号を指定して、1対1の回線を成立させる。市内電話は東西NTTが所有する回線網で完結する。長距離電話は、市内回線網に加えてNTTコミュニケーションズの回線網も使われる。しかし、いずれの通信でも、回線交換機によって、電話をかける側と受ける側の電話機とが直接接続される。回線交換機による接続では、電話番号が必要不可欠である。インターネットにおいては、電話番号に相当するものがIPアドレスである。電話番号の指定は、IPアドレスの指定に相当する。しかし、インターネットを利用するユーザがIPアドレスを指定することは稀である。それゆえ、IPアドレスの解説は難しい。世界で唯一のネット上の住所を表すURLさえわかれば、アプリケーションが自動的にIPアドレスの指定を行うという説明でも十分かと思われるが、実際には、アプリケーションがDNSに問い合わせを行い、アドレス解決を行う。アドレス解決は、インターネットの特性とは、直接的な関係がない

ので、説明を省略しても問題ない。しかし、アドレス解説も電話とのアナロジーで解説することもできる。アプリケーションが通信する際には、相手のコンピュータのIPアドレスがわかっていなければならない。IPアドレスの電話帳に相当するデータベースがインターネット上にはあり、DNSと呼ばれている。DNSには、URLとIPアドレスが対になって記録されている。コンピュータのネットワーク設定で、DNSのアドレスを登録しておけば、アプリケーションは自動的にDNSに問い合わせを行い、IPアドレスの情報が取得される。URLとIPアドレスは企業名と電話番号の関係と同じである。そして、人間が電話帳を使って電話番号を調べると同様に、アプリケーションはDNSを使って、IPアドレスを取得する。回線の終端が仕事をするエンドツーエンドの通信の特徴であるといえる。アドレス解決の処理はプロトコルスタックの最上位であるアプリケーション層で行われる。つまり、物理的なネットワークの性質からもっとも遠いアプリケーション側でアドレス解決が行われている。インターネットの仕組みの解説するためには、IPアドレスは不可欠であり、最初に解説すべきものである。なぜなら、インターネット技術の要であるパケットに、IPアドレスが記録されているからである。

1-4 パケットの送受信

パケット交換方式の比較例として、電話の回線交換方式を用いるのがよい。インターネットと電話の回線の制御に関しては大きな違いがある。その違いを通じてインターネットの利便性を強調するのである。電話は回線交換方式であり、インターネットはパケット交換方式である。インターネットおよびWWWの汎用性と利便性はパケット交換方式による、通信インフラであるインターネットが、放送メディアとしても使用できるのも、パケット交換方式が同時接続に適した方式であるからである。電話回線の場合、NTTの交換局で制御される。電話の回線交換方式は必要な時にだけ接続が成立する。また、接続は排他的である。すなわち、電話機間の接続が実現されると、他の電話機とは接続できない。この排他的な接続方式が、信頼性を保証する。しかし、同時にサービスの可能性を限定することになる。実際、電話以外に電話の回線交換方式を利用したサービスはファックスしかない。⁽¹⁾インターネットは、回線交換ではなく、パケット交換方式である。データをパケットに分割してやりとりすることで、回線を占有しない。占有しないという性質は、電話での話中という状態がなくなることを保証する。その反面、回線には、送信元も送信先も異なる多数のパケットが混在することになる。パケットの輻輳が起こるとサービスの質が低下するというデメリットがある。しかし回線の輻輳は、光ファイバ化など高速化することで改善できる。輻輳というは質ではなく量の問題である。それに対して、パケット方式には原理的に大きな利点がある。1つのサーバに多数のクライアントが接続できるという利点である。前述の「話中がない」という特性と、1対1の通信が同時に多数存在するという特性は、本来相容れない。WWWは24時間、どこからでも、何台からでもサイトにつながる。このような電話にない利便性は、インターネットの特性によるものである。したがって、Webの基本技術の解説する際に、インターネットの特性を生み出すパケット交換方式を強調すべきである。パケットを使うことで、同時多数アクセスの問題を解決しているのである。

インターネットのパケットには、必ず送信元アドレスと送信先 IP アドレスの両方が指定される。アドレス指定によって、サーバとの1対1の通信が実現される。インターネット回線は多数のパソコンによって、常時、共有されている。送信元から送信先へ、ルータによって経路が決められて、パケットが届けられる。パケットの中身のデータはいかなるものでもよい。データはホームページのものに限らず、メール、ファイル、電話、映像など、ありとあらゆるものである。大きなデータも細かく分割されて、パケットに入れられる。データの中身によって、要求される送受信の確実性や速度は異なる。しかし、プロトコルスタックの第二層であるインターネット層には、IP アドレスを調べて、パケットを送るだけの機能しかない。確実性や速度を決めるのは、アプリケーション寄りの第三層、トランスポート層である。この、インターネット層の単純さにより、インターネットが普及したと解説する。通信の確実性や速度はエンドノード、すなわちサーバやクライアントで決める。中間ノード、すなわちルータでは、IP アドレスを調べて、適切なノードにパケットを転送する。ネットワーク装置としては、シンプルなルータしか必要とされない。分割されたパケットの正しい到着順、到着したパケットをどのアプリケーション使用するのかということも第二層であるインターネット層では制御されないことを強調すべきである。あくまで、送信元ノードから送信先ノードへ IP アドレスを参照して、パケットを転送するという最低限の機能しか、IP には実装されていないからである。

1-5 パケットの制御

プロトコルスタックの第二層では、パケット送受信の確実性、速度、到着順、およびパケットを使用するアプリケーションの指定の制御が行われない。全て、第三層のトランスポート層で行われる。第三層では、アプリケーションの性質によって、TCP と UDP という2つのプロトコルが使い分けられることを、まず解説すべきである。2種類のトランスポートプロトコルの存在は、インターネットのアプリケーションに多様性をもたらすからである。TCP と UDP の違いは、パケットの制御方法の違いである。TCP は、送信側と受信側のコネクションを成立させる。そして、受信側の受け入れ状態を確認後、パケットの到着順が制御される通信を行う。受信側には、送信側にパケットが受信されたことが通知される。オーバーヘッドは大きくなるが、TCP によってパケット通信の確実性が保証される。UDP はコネクションを作らない。パケットの到着順序も保証されず、パケットの送受信の確認もない。UDP は、確実性を犠牲にして、オーバーヘッドの小ささを優先する。TCP とは正反対の性質を持つプロトコルであるといえる。TCP と UDP の共通点は、パケットを使用するアプリケーションを特定することである。送信元アプリケーションのポート番号と送信先アプリケーションのポート番号を指定することにより、クライアント / サーバ間の特定アプリケーション間の通信ができる。ポート番号がなければ、パケットがエンドノードに到着しても、どのアプリケーションに使用されるものなのかを、エンドノードは判別できない。クライアントマシンの中に、多数のアプリケーションがインストールされていることは周知の事実である。同様に、1つサーバマシンの中に、複数のサーバが存在するということが同時に解説すべきである。インターネットの接続の容易さはプロ

トコスタックの設計にあり、エンドツーエンドの通信を実現するために、第三層、第四層の役割はソフトウェアが担うからである。届いたパケットの振り分けは、第二層ではなく第三層に割り当てられている。実際には、OSに含まれるTCP/IP通信機能⁽²⁾に対して、各アプリケーションが固有のポート番号を指定する。

1-6 インターネットのアプリケーション

第四層はアプリケーション層であり、Webを含む様々なサービスが実装されている。第四層のサービスは、その目的により、第三層のTCPかUDPのどちらかを使用する。確実性が必要なサービスにはTCP、速度が必要なサービスにはUDPが採用される。第四層のアプリケーションを開発する際には、TCPかUDPに、ポート番号を指定して、データを渡す部分だけを実装することになる。開発者は、第二層以下のことは考慮しなくてもよいのである。このインターネットの階層化された設計思想により、開発者の負担は最小限で済むことになった。帯域の拡大、すなわち回線の高速化も重なり、結果的に、多数のインターネットサービスが登場した。具体的には、メール、電子掲示板、FTP、WWW、ファイル共有、IP電話、動画ストリーミングなどである。各サービスは、それぞれ異なるプロトコルを第四層で使用する。WWWが使用するプロトコルは、速度を優先する場合はHTTPで、機密性を優先する場合はHTTPSである。あくまでWWWはインターネット上のサービスの1つにすぎない。第四層では、他のサービスのプロトコルも同時に多数存在できる。このインターネットをインフラとした、各種サービスの実装を、よく認識させる必要がある。インターネットには、現在では考えもつかないサービスが登場する可能性がある。また過去では、メールとファイル転送が主なインターネット上のサービスであった。しかし、現在WWWが最も有力なインターネット上のサービスであることは間違いない。企業の業務は、特に非コアコンピタンス分野において、Webサービスが用いられるようになってきた。一般人のパソコン利用も、立ち上げた直後にホームページを見るという使われ方になった。以前はOutlookなど専用のアプリケーションを使用していたメールも、Gmailなどの登場により、WWWベースに移行しつつある。WWW登場以前のメディアは、TV放送に代表されるマスメディアか電話に代表されるパーソナルメディアに分化していた。しかし、WWWでは、1対1のパーソナルメディアの性質を基本にしているにもかかわらず、1対多、すなわち同一情報への同時かつ多数のアクセスもできる。WWWはパーソナルメディアとしてもマスメディアとしても使用できるのである。さらにセキュリティの問題や回線輻輳も、早期に解決された。通信に欠かせない「通信の秘密」は、第四層に実装されたSSLで守られる。SSLは公開鍵暗号系をWebのプロトコルであるHTTPに実装した暗号化通信の仕組みである。ネット銀行、ネット証券、オンラインショップにおける注文や決済はSSLの仕組みなしには成り立たない。一方、同時多数アクセスによる回線輻輳は、第二層のロードバランサによる負荷分散で回避される。実際、NHKオンデマンドなど動画のストリーミング放送がビジネスとして動き始めた。もはや、通信と放送の違いは法律にしかないといえる。インターネットが一般人に公開された当時は、インターネットサービスプロバイダなどの多数の第二種電気通信事業者

が生まれた。しかし、近年プロバイダは飽和状態にある。通信インフラとしてのビジネスはパイが限られる。しかし、インターネットは放送メディア、すなわちマスメディアとしての性質をもつ。マスメディアとしても使用できるということは、広告ビジネスが成立することを意味する。ネット広告は、当初は単純なバナー広告やメールマガジンに添付される広告リンクがほとんどであった。しかし、近年、インターネットのパーソナルメディアとしての性質を生かした広告方式が登場した。検索キーワードにマッチした広告が掲載される検索連動型広告である。他にも、ユーザのネットでの行動履歴を記録して広告を出す行動ターゲティング広告、自然言語処理の技術を生かして、ページの内容にマッチした広告を出す内容連動型広告などが、主流になりつつある。これらは、TVのような従来型マスメディアでは、原理的に実現できない広告である。また、インターネットでは広告の料金も様々である。ページが表示された回数、広告がクリックされた回数、広告に誘導されて実際に契約にいたった回数などによって、広告料が支払われる。このような広告の方式や、広告料の取り方も、インターネットが1対1の通信を基本としていることで実現されている。

1-7 インターネットの物理的な性質

伝送媒体を選ばないということも、インターネットが自己増殖的に広がった一因である。プロトコルスタックのもっとも底である第一層は、物理層と呼ばれる。物理層には、デジタル信号の送受信のプロトコルが実装されている。インターネットでは、電話線、光ファイバ、電波など、いかなる物理的な性質をもつ伝送媒体も使用されている。新しい伝送媒体が開発されると、その特性を生かすプロトコルが設計される。基本的には、伝送媒体の数だけプロトコルが存在する。また、同じ伝送媒体を使う場合でも、帯域改善をするために、より効率的に信号を送ることができるプロトコルが開発される。したがって、第一層のプロトコル数は多い。LAN内部とWANではプロトコルが異なることが多い。その場合、LANとWANの接続には、第二層のネットワークデバイスであるルータを使う必要がある。ルータはパケットを次のノードに転送するための経路を決定する装置である。しかし、厳密に言えば、経路を決めると共に、LAN内のフレームに載せたパケットを別のフレームに載せ替える装置である。フレームの数だけ、第一層のプロトコルが存在する。第一層のプロトコルの違いは、第二層のルータで完全に吸収されるということを理解させられればよい。例として身近なのは、携帯用ゲーム機である。WiiやDSなどがインターネット接続できるのは、無線LANルータが、無線用のフレームから優先用のフレームにパケットを載せ替えているからである。

2 Web標準

本来のWWWは、どんな環境でも同じように使えることを目的に設計されている。ページを表示するブラウザによって、ハードウェアやオペレーティングシステムの違いは吸収される。HTMLやCSSの仕様は、WWWの標準化団体であるWorld Wide Web Consortium（以下W3C）によって策定される。

2-1 Internet Explorer の扱い

W3C の勧告には強制力がなく、ブラウザベンダの利害関係を完全に調整するものではない。それゆえ、各ブラウザの実装は、W3C の Web 標準に準拠していない場合がある。特にマイクロソフト社の Internet Explorer (以下 IE) には、独自の仕様が多い。2001 年にリリースされた IE6 は、当時としては優れたブラウザであり CSS1 に対応していた。さらに、IE6 は Windows XP に標準でバンドルされていたブラウザである。それゆえ、ブラウザのデファクトスタンダードとして長らく使われてきた。IE6 は DOM1 にも対応しており、JavaScript も Jscript として搭載されていた。各種の Web サービスも、Windows XP 上の IE6 で動くことが必須条件とされてきた。Jscript の独自仕様は、JavaScript ライブラリで吸収された。IE6 の CSS の解釈は、最新のブラウザと CSS の解釈と異なる部分がある。しかし、それに対しても CSS ハックなどの対策があった。そもそも、テーブルレイアウトを使えば、CSS によるデザインと同様なページを制作できる。ただし、CSS を使わなければ、ソースコードのサイズも増え、サイトのメンテナンス性は悪化することになる。また、ブログに代表されるコンテンツマネジメントシステムでは、コンテンツとデザインが完全に分離できる。Web のプラットフォーム化と、CSS の使用という流れは、この先も止らない。過去には、優れたブラウザであった IE6 の存在が、Web の開発の効率を妨げてきた。IE は現在では、無理して使われる存在であるが、無理すれば使えてしまう存在であるともいえる。IE への対処を教えるべきかどうかは現状において非常に判断が難しい。シェアは確実に落ちてはいるが、これからも使われ続けられることは間違いない。しかし、Windows XP のサポートが切れるのが 2014 年 4 月であることを考慮すると、2010 年度からは、IE を使った Web 制作は行わないほうが良いといえる。

2-2 HTML と XHTML

Web 制作の際に、HTML と XHTML のどちらを採用するかという問題がある。いわゆる Web 標準への対応という面では、現在は XHTML である。しかし、今後主流となると思われる HTML5 は HTML4 がベースとなっている。Web 標準、すなわち WWW で使われる技術の仕様を策定する非営利機関は W3C である。W3C は 1999 年の HTML4.01 を最後に HTML の仕様の改訂を停止した。そして、HTML に変わるマークアップ言語として、XHTML の仕様を策定した。しかし、この XML をベースとした XHTML の普及の遅さが、HTML5 の登場の原因となった。WWW および HTML の考案者で、W3C の主要人物でもあるティム・バーナーズ・リーが、事業者ではなく研究者であることも XHTML の商業的な普及を妨げたといえる。XHTML1.0 の策定後に、実用の見通しが経っていない Semantic Web に重点を置いたことが重なり、W3C は影響力をなくしつつある。XHTML には HTML にはない利点がある。HTML 文書を XML 文書としても扱えるというものである。しかし、その利点は、これまでほとんど生かされなかった。ページ作成用言語として、HTML で十分であるというより、Web アプリケーションを実現できる安定したプラットフォームが要求された結果である。一般に、XHTML への移行が進まなかった原因として、後方互換性のなさが原因といわれている。しかし、IE6 以降の PC 用ブラウザ

は、XHTMLのソースを正しく表示できる。IE6は2001年8月に公開されたブラウザである。Windows XPに標準添付されていたため、2009年の現在でも15%以上の市場シェアを持つ。IE6はCSSレベル1、DOMレベル1に対応しているため、Webプログラミング対応できる。DOMは、HTMLの要素を動的に扱うためのオブジェクトモデルである。DOMは、主にJavaScriptで制御される。JavaScriptは、2000年ごろまではほとんど無視されてきた。商業的には、フォームの入力チェックに使用されるぐらいであった。Webサービスはほとんどサーバサイドで、実現されていたのである。Webサービスの分野で、JavaScriptを積極的に使用したのはGoogle社である。Asynchronous JavaScript+XML（以下 Ajax）という技術である。Ajaxは、XMLHttpRequestという命令をつかったWebアプリケーションの実装方法である。そして、あまり知られていない事実であるが、マイクロソフト社によって、IEに独自に導入された命令でもある。「Outlook Web Access」というメールソフトのWebインターフェースのためにActiveXオブジェクトとして、1999年に実装された。XMLHttpRequestによって、ブラウザは、状態遷移を起こすことなく、HTTPプロトコルを用いて、サーバとXMLデータの非同期通信を行うことができるようになった。デスクトップアプリケーション並の使用感を持つWebアプリケーションが実現された。Ajaxでは、サーバとブラウザのデータ通信の際に使われるデータのフォーマットがXMLなのである。サーバに置いておくHTMLのソースファイル自体がXMLである必要は、少なくとも商業的にはなかったといえる。Ajaxをもってしても実現できない動画再生などは、Flashのプラグインなどをブラウザにインストールすることで対処された。このように、Webの歴史は、「ページの記述言語が何であるか？」とは、別の次元で動いてきた。過去に、「HTMLの後継は作らない」と宣言したW3Cから、HTML5がリリースされることになる。しかし、XMLというデータフォーマット自体は廃れるものではない。したがって、XMLとはいかなるものかも含めて学問的に教えるならばXHTML、商業利用を重視して教えるならばHTML4.01の採用ということになる。そして、将来性を見据えれば、すでに草案がリリースされ、一部の機能がIE以外のブラウザに実装されているHTML5を採用するほうがよい。すでにブラウザは、ページ記述言語でありながら、アプリケーション提供のプラットフォームになったからである。

2-3 CSSの使用

W3Cでは、Webページのデザインを行う際には、CSSを用いるように推奨している。XHTML1.1では、Transitionalの文書型を廃止し、テーブル関係の属性を大幅に削除した。テーブルレイアウトの使用はXHTML1.1では不可能に近い。HTML5でも、テーブルレイアウトはほとんど使われなくなると予想される。次期CSSである、CSS3の能力が飛躍的に高まっているからである。特に、角丸ボックスが画像を使用することなく作成できる利点は大きい。いわゆるWeb2.0と呼ばれているサイトでは、角丸がつかわれることが多いからである。

ブラウザでレンダリングしたHTML文書のデザインを変更するのが、CSSの役割である。あくまでデザインの「変更」であり「作成」ではない。HTML文書になにも書き加えなくても、

ブラウザのデフォルトスタイルによって自動的にデザインされて、表示されるからである。この部分に関して、誤解する学生が見受けられるので、何度も強調すべきである。ブラウザのデフォルトスタイルは、あまりに貧弱である。

1990年代中盤、WWWの商用利用の黎明期には、CSSの機能自体がブラウザに実装されていなかった。したがって、Webページは、ほとんどデザインされていなかった。2000年ごろになり、インターネットはブロードバンドが当たり前になってきた。回線スピードの向上とともに、テーブルレイアウトがデザインの手段として用いられるようになった。テーブルレイアウトとは、画面をテーブルで区切るレイアウト法である。テーブルの背景に画像を用いれば、画像のデザイン次第で、いかなるデザインもできることになる。しかし、画像はサーバ側におかれる。したがって、通信データは増加。CSSでも画像を使うが、テーブルレイアウトほどではない。テーブルレイアウトでは、コンテンツの変更が難しいということも重大な問題点である。テーブルのどこのセルに何が書かれているかが分からないのである。また、デザインの変更も困難である。HTML文書の中に、デザインの情報が散りばめられるため、デザインの管理が不可能である。実際、コンテンツの変更が頻繁なブログ、SNSなどのCMSでは、CSSが積極的に使用されている。

CSSも利点ばかりではない。CSSはブラウザごとの違いが大きいというデメリットも教えるべきである。ネットスケープやMachintosh用のIEは、使用しているユーザはほとんどいない。Firefox3、Opera10、Safari3以降では、CSS2の勧告通りに表示される。問題は、IE5、IE5.5、IE6、IE7である。しかし、IEでの不具合の回避法は定番となっている。Web標準に準拠しているブラウザで意図通り表示されていれば大きな問題ではない。CSSを書いている途中では、IEを無視して、Firefoxで正常に表示されるように作るのがよい。IEのCSSの問題点は以下の通りである。

1. 背景画像が出ない
2. borderが出ない
3. 互換モード時のボックスの幅の算出
4. 横方向のmarginに指定するautoがきかない
5. floatした要素の横マージンが2倍になる
7. 垂直方向のmarginが相殺されない
6. borderのdottedがdashedになる

Web制作に慣れているプロにとっては、大きな問題ではない。しかし、初学者を惑わしやすい問題もある。IEは、Web制作、とくにCSSの演習には向いていないといえる。Mac OSでは、現状は、ほぼSafariである。Windows XPでは、IE6が、VistaにはIE7が、標準添付されているブラウザである。Windowsで、CSSの演習を行う場合は、Firefox、Opera、Safari、Google Chromeのいずれかをインストールするべきである。

3 Web制作演習

Web業界は、変化が早く、新技術が数多くでてくる。これからも確実に使われるものだけを内容にとりいれるべきである。そもそもWeb制作のやり方は、完全にHTMLを手打ちする方法、ホームページオーサリングソフトを使う方法、どちらも使用する方法など様々である。ページの種類によっても異なる。しかし、基本となるのは手打ちである。Webサービスを作る場合はWebプログラミングを行う。このときHTML要素を理解していないと実装できない。オーサリングソフトはHTML生成することに関しては問題ない。しかし、CSSやスクリプト言語に関しては、補助程度のことしかできない。本章では、エディタによる手打ちの演習の内容について論じる。

3-1 HTML演習

HTMLを手打ちで書く場合、通常の汎用エディタか、HTMLエディタを用いる。汎用エディタはメモ帳が一般的である。再設定せずに、IEでソース表示をすると、メモ帳が起動する。ただし、メモ帳はあまりに低性能である。行番号表示機能がないことはWebプログラミングで使えないことを意味する。通常、スクリプトのエラーメッセージなどでは、エラーが発生した行の番号が情報として表示されるからである。また、空白記号を見分ける機能がないのは、初学者がHTMLやCSSを書くときに、大きなストレスとなる。HTMLの“<”と“>”の中に、全角空白を入れるとエラーになるからである。HTMLのタグが色分けもされないのは初学者でなくても見にくい。したがって、指定したエディタでソースを見るときは、「ツール」—「インターネットオプション」—「プログラム」でHTMLエディタを設定する。ただし、この設定では「表示」—「ソース」ではなく、「ファイル」—「編集」でソースファイルを開くことになる。「表示」—「ソース」で、メモ帳以外のエディタを起動したい場合は、レジストリを書き換えることが必要になる。

HTML文書の作成は、以下の手順で進めると問題が起きない。原稿とHTML文書の作成を分ける理由は2つある。日本語変換モードに入ることによるタグの記述ミスを防ぐためと、HTMLが持つ「マークアップ言語」の意味を理解させるためである。

1. マイドキュメントの中に、Webページを作るルートフォルダを新規作成
2. Wikipediaから文章をコピーして原稿を制作してルートフォルダに保存
3. 画像専用のimageフォルダを、ルートフォルダの中に新規作成
4. ライセンスフリーの画像をWebで検索し、imageフォルダの中に保存
5. HTMLの文書型定義を記述
6. 構造要素をマークアップ
7. 原稿を開き、body要素の中に、原稿を流し込む
8. body要素の中を、主にh要素とp要素などでマークアップ
9. HTMLヴァリデータでHTMLの文法を検証

10. 主に p 要素の内部をインライン要素でマークアップ
 11. div 要素で、ブロックレベル要素をグループ化

FTP を用いてアップロードする場合に、通常はフォルダごと転送するので、制作用のフォルダを 1 で用意する。HTML を書くためには、伝えたい内容があることが前提である。学生に原稿を用意させるのは、Web 制作演習ではない。しかし、著作権意識は持たせておかせたいので、Web の文章をコピーするのはよくない。WikiPedia の著作権は特殊なライセンスの GNU Free Documentation License (以下 GFDL) である。GFDL の文章は、GFDL で公開しなければならない。[1] GFDL の特殊性は「公開するときは、必ず GFDL で公開する」という点にある。したがって、WikiPedia の文章を Web ページに掲載する場合も GFDL になる。画像に関しても、著作権に留意すべきである。現在の Web 制作は、画像なしで行われることは極めて稀である。CSS によるデザインでも、画像が多用される。したがって、画像専用のフォルダを作る。次の文書型定義と構造要素のマークアップに関しては、決まり切ったものである。しかし、文書型定義は複雑で間違いやすい。文書型定義に誤りがあると、CSS の互換モードになることがある。したがって、テンプレート生成サイトなどを利用したほうがよい。[2] テンプレート生成サイトでは、ダウンロード不要なので、何度も試すことができる。しかも、XML 宣言の有無、文書型を選択できる。XHTML 文書は、XML 文書でもあるため、本来なら XML 宣言をしなくてはならない。しかし、IE6 以前のブラウザで XML 宣言をすると CSS が互換モードになってしまう。互換モードとは、古いブラウザ用の CSS を使うためのモードである。互換モードでは、CSS の要であるボックスモデルの解釈が異なる。したがって、標準モードで CSS を適用するために、XML 宣言は選択させない。文書型は、テーブルレイアウトを行うなら「Transitional」、行わないなら「Strict」を選択させる。次に、テキストエディタに保存されている WikiPedia の原稿を、body 要素の中にコピーさせる。この段階で文法のチェックをさせないのは、body 要素の子要素としては、ブロックレベル要素しか許されていないからである。したがって、見出しは h 要素、段落は p 要素でマークアップさせる。body 内部をマークアップしたら、必ず HTML を検証させる。なぜならば、HTML が正しくなければ、CSS を適用できないからである。HTML ヴァリデータはいくつかあるが、慶應大学の Web サービスが日本では定番である。[3] インライン要素のマークアップは、必ずブロックレベル要素の中で行う。重要な要素は限られる。最後に、div 要素で、グループ化して扱いたいブロックレベル要素を囲む⁽³⁾。最低限、盛り込むべき要素を、表 1 に列挙する。

表 1 盛り込むべき HTML 要素

構造要素	html, head, body, title
ブロックレベル要素	h1-h6, p, ul, ol, dl, blockquote, address, div, form
インライン要素	a, img, strong, span, input
リスト項目の要素	li, dt, dd

CSS, スクリプト	link, script, style
メタ情報	meta
プラグイン	object

表1で、構造要素は必須である。これらがなければ、そもそもHTML文書にもならない。ブロックレベル要素のうち、form要素は、JavaScriptやサーバサイドのプログラミングと共に使われるものである。実際のWeb制作では、単独で仕様されることはない。これらを教える予定がなければ、カットしたほうがよい。インライン要素のinput要素も同様である。

3-2 CSS 演習

CSSを使用する場合3通りの方法がある。W3Cによると、link要素で外部スタイルシートを読み込むことが推奨されている。外部スタイルシートの使用は、コンテンツとデザインの分離が明確であり、必ず教えるべき内容である。しかし、この手法は、HTML文書とCSSファイルが別になるため、演習がやりにくい場面もある。また、実用的にも、全てのページで同じスタイルを適用することはまれである。したがって、style要素で、head要素内にスタイルシートを書く方法も必ず含まれなければならない。

CSSの演習は大きく2つに分類できる。セレクタの種類と使い方とプロパティと値の組み合わせである。前者は、HTML文書のどの部分に対して、デザインを施すかを指定するものである。それに対し、後者は、どのようなデザインを施すかを指定するものである。これらは、組み合わせで指定することにより、CSSの定義になる。HTMLとCSSのように、分離して教えることはできない。

セレクタに関しては、覚えることは少ない。基本となるセレクタは3つ、要素タイプセレクタ、IDセレクタ、クラスセレクタである。IDセレクタは、ページヘッダー部分やナビゲーション部分などページに一カ所しかない要素に対して指定する。クラスセレクタは、同じ要素に対して違うデザインを適用したいときに指定する。大規模サイトを作る場合は、これらの基本セレクタを子孫セレクタとして使用すればよい。実用を見据えると、a要素で使う疑似クラスセレクタも必要になる。ハイパーリンクに対する装飾で、a:link、a:visited、a:hover、a:activeの4種類である⁽⁴⁾。

HTML要素の属性とCSSのプロパティは同じ性質を持つ。しかし、HTMLの属性の場合、数が少なく、属性値も限られている。CSSのプロパティと値は、量が膨大である。教育現場では、扱うプロパティの数を絞るべきである。プロパティの取り得る値についても、注意が必要である。HTMLの属性値は、数値に単位が必要なかった。すべてピクセルとして解釈されるからである。CSSの値は、単位をつける必要がある⁽⁵⁾。唯一の例外が、倍率を表す場合である。1.5は150%と同様である。最小限教えるべきプロパティの数値を、表2に挙げる。

表2 数値のプロパティ値

長さ	em, px, pt
割合	%
倍率	単位なし
色	#rrggbb, #rgb

表2の他にも、CSSの値はある。しかし、それらは、プロパティに特有の値であるので、プロパティとペアで覚えるべきである。CSSで最も重要なプロパティはボックスに関するプロパティである。なぜならば、HTMLのbody要素の内部の要素は、全てボックスモデルで、ブラウザに表示されるからである。表3にボックスモデルに基づくデザインの際に、最低限教えるべきプロパティを挙げる。

表3 ボックスモデルのプロパティ

border	ボックスの外枠
margin	ボックスの外枠から他のボックスまでの余白
padding	ボックスの外枠から内容領域までの余白
width	ボックスの内容領域の幅
height	ボックスの内容領域の高さ
float	ボックスのフロート配置
clear	ボックスのフロート配置の解除

これらは、あくまで最低限である。実際にデザインする場合には、他にも様々なプロパティを用いる。CSSのプロパティと値の組み合わせは、数多い。覚えようとして、覚えきれぬ分量ではないので、辞典やオンライン辞書を見ながら制作させるべきである。

表4 主要プロパティ

line-height	行の高さ
text-align	行内の横方向の配置位置
vertical-align	行内の縦方向の配置位置
text-decoration	テキストの装飾
font-size	フォントの大きさ
font-weight	フォントの太さ
font-family	フォントの種類
color	前景色
background	背景
list-style	リスト
display	レイアウト方法
position	ボックスの配置方法
left, right	ボックスの横方向の配置位置
top, bottom	ボックスの縦方向の配置位置
overflow	ボックスからはみ出た内容の処理
z-index	ボックスの重なり

ソースファイルが短く簡潔になるため、ショートハンドプロパティを積極的に使用したほうが良い。しかし、font プロパティだけは、値の順序が決まっているなど扱いが難しく、個別のプロパティを用いたほうがよい。他に、特に複雑なのは display, position と left, right, top, bottom でのレイアウトである。position の値が relative か absolute で、挙動が異なる。しかも、該当ボックスの要素の親要素が作るボックスからの相対的な関係で位置が決まる。absolute の場合、親要素のボックスの position の値が static (デフォルト値) 以外でないと、body 要素のボックスからの相対的な位置になる。

4 HTML5 での Web 制作

2004 年に、Safari、Mozilla、Opera を作るブラウザベンダ Apple、Mozilla Foundation、Opera Software が中心となり、Web Hypertext Application Technology Working Group (以下 WHATWG) という団体が設立された。WHATWG は W3C とは異なった実用性が第一という思想のもとで新たな仕様を検討した。そして、2007 年に新たな HTML の仕様を W3C に提案した。この提案は、HTML5 という新しい HTML として、共同で策定された。正式勧告の予定は 2010 年であるが、2008 年に草案が公開された。HTML5 の新機能は、既に、いくつかのブラウザで部分的に実装されている。HTML5 でもっとも期待されている新機能は canvas 要素である。これまで画像は GIF や JPEG の画像ファイルを読み込むしかなかった。canvas 要素を使用すれば、ブラウザの機能だけでベクター画像を記述することができる。これまでは、サーバ側に画像を置いて、クライアントがそれをダウンロードしてブラウザが表示していた。canvas 要素の使用により、クライアント側の CPU やメモリなどのリソースを用いて、ブラウザで描画できるようになるのである。具体的には JavaScript を書く。以下の例は、黒い四角形を書くスクリプトである。現在のところ描画できるのは、2次元のみであるが、将来的に3次元への拡張が予定されている。以下に例を示す。

```
<html>
<head>
<script type="application/x-javascript">
  function rectdraw () {
    var canvas = document.getElementById ("canvas") ;
    var ctx = canvas.getContext ("2d") ;
    ctx.fillStyle = "rgb (0,0,0) ";
    ctx.fillRect (10, 10, 50, 50) ;
  }
</script>
</head>
<body onload="rectdraw () ">
```



図1 canvas 要素で描いた矩形

```
<canvas id="canvas" width="300" height="300"></canvas>
</body>
</html>
```

HTML5では、文書構造をマークアップするための要素の追加である。従来のHTML、およびXHTMLでは、文書構造は、h要素とp要素しかなかった。ブロックレベル要素をグループ化するためのdiv要素を構造をマークアップする要素として代用してきた。しかしdiv要素1つでは、id属性で名前をつけることによって、区別する手法に頼るしかない。id属性は、本来、CSSを適用する際に、要素を区別するためのものである。HTML5では、一般的な章をマークアップするsection、ブログのエントリーや独立した記事などをマークアップするarticle、補助的な情報をマークアップするaside、ヘッダにあたるセクションをマークアップするheader要素、フッターにあたるセクションをマークアップするfooter要素、ナビゲーションをマークアップするnav要素が導入される。

HTML5では、フォームに関しても、Webサービスへの対応が考慮されている。従来は入力データのチェックなどはJavaScriptで行っていた。HTML5では、input要素のpattern属性で指定できる。以下の例は、URLのみを許可できるフォームである。

```
<label>URL:
  name="ad" title="http address" />
</label>
```

クライアント側のパソコンの能力は、飛躍的に高まった。それに対し、サーバ側で多数アクセスがおきると処理は重くなる。ロードバランシングやクラスタシステムなどの技術はあるが、コスト的な問題がある。グーグルが巨大企業になったのは、技術力があつたからである。しかし、初期に資金豊富なスポンサーが存在したことも大きい。大規模な商用サーバの維持は莫大なコストがかかる。それゆえ、これからのネットワークアプリケーションは、極力クライアント側のリソースを使うことになる。もっとも効果的な実現方法は、HTMLの規格変更であり、HTML5がその大本命である。いまのところ、WHATWGには主要ブラウザベンダであるマイクロソフトが加入していない。しかし、IE8では、すでにHTML5の機能を一部実現している。クッキーの代わりとして期待される「sessionStorage」や「globalStorage」である。また、ブラウザを使用するユーザがウェブページの特定要素を編集できる「contentEditable」の開発元でもある。今後はIEも、Web標準に準拠していくはずである。IEの独自仕様がなくなるということは、商業的にも教育的にも意味が大きい。これまで、IEとそれ以外のブラウザで仕様が異なり、Web制作者は、本来は不要なテクニックで、それを吸収してきた。過去に起きたNetScape社とのブラウザ戦争の結果、XMLHttpRequestなど有用な技術も生まれたが、混乱を生んだだけの独自仕様も残った。その混乱もW3Cが商用利用を考慮したHTML5を正式勧告することで収まる可能性が高い。

5 Web制作と著作権

Web制作においては、コピー & ペーストが簡単にできる。厳密には、著作権侵害であるケースが頻繁に見受けられる。しかし、ネットでの著作権違反の訴訟例は少ない。それは、削除が用意であることによる。いきなり訴訟になるのではなく、通常は、著作権利権者が違反者に警告を出すからである。著作権に関する法律には例外もある。学校において、書籍などのコピーはある程度認められている。ただし、イントラネットにWebページを公開する場合は、注意が必要である。

5-1 WWWに関する著作権

Webページを作り、公開する際には、著作権に留意する必要がある。WWWは登場して20年近いメディアではあるが、現在までのところ、個別の著作権は立法されていない。従来の著作権を延長した「自動公衆送信権」がWebページの著作権に相当する。「公衆送信」は、従来の法律の著作権法では、「公衆によって直接受信されることを目的として無線通信又は有線電気通信の送信を行う」ことをと定義している。ただし、これは従来メディアを含む大きな概念である。WWWは、「公衆からの求めに応じて自動的に行うもの」である。公衆送信の中の「自動公衆送信」として定義されている。WWWはオンデマンド放送の性質を持っているからである。そして、Webページをネットに公開することは「送信可能化」と定義されている。著作者は、その著作物について、送信可能化の権利を専有する。したがって、従来メディアやWebの

著作物を、無断で、WWW上に公開することは、著作権法違反である。

5-2 教育機関における例外

著作権は、商用利用がからむ場合は、厳しく守られるべきものである。しかし、あまりに厳しく著作権者の権利を守ると、文化や教育の促進に悪影響を及ぼす。実際、著作権法でも例外が設定されている。例外は、著作権法第35条の「学校その他の教育機関における複製」に記述されている。「学校その他の教育機関において教育を担当する者は、その授業の過程における使用に供することを目的とする場合には、必要と認められる限度において、公表された著作物を複製することができる」というものである。ただし、この条文には条件があり、「著作権者の利益を不当に害することがない」というものである。この条件に加え、「授業の過程における使用」、「必要と認められる限度」の解釈にも曖昧性が多い。紙にコピーする場合は、複製の部数が一応の基準となる。しかし、Web ページの場合は、「必要と認められる限度」に関する判断が困難である。

5-3 Web ページの公開の危険性

組織として、Web ページを公開する方法は二通りある。インターネットのWWW上に公開する場合と、イントラネットに限って公開する場合である。インターネットにWeb ページを公開する場合は、著作権は厳密に守られなければならない。インターネット上のページは不特定多数への送信可能化である。教育機関であるとしても、組織外部への公開は、「授業の過程」ではなく、特例は認められない。イントラネットに限定したWeb ページの公開は、論点がいくつかある。イントラネットに公開されるページは、教材として制作されたものと学生に演習で作らせたものである。いずれの場合も、実際には、Web ページを閲覧する人間は一部に限られる。しかし、そのページが、授業を受けている学生だけがアクセスできるという特殊なページでない限り、著作権法違反の可能性がでてくる。たとえイントラネットであっても、Web ページが、授業を受講していない学生が閲覧できる状態になっている場合、著作権法違反の可能性がでてくる。しかし、アクセスできる人間が限られているため、「不特定多数」とまではいえない。実際には、グレーゾーンとして見なされている。グレーゾーンであれば、法廷に持ち込まれる可能性がある。したがって、イントラネットへのWeb ページの公開においても、著作権は十分に考慮すべきである。

6 まとめ

現在利用されているHTMLはバージョン4.01であり、XHTMLはバージョン1.0である。HTMLからXHTMLへの移行はスムーズには進まなかった。当初対応できるWebブラウザが少なかったことなど、テクニカルな原因は多数ある。しかし、根本は1つである。W3Cという組織が、WWWの考案者であるティム＝バーナーズ・リーがMIT内に設立したコンソーシアムであるからだ。W3Cの規格策定は、学問的、思想的な側面も強い。教育用としては優れているのであるが、実用面に徹していない。Webの商業利用は、W3Cとは相容れなかった。結果とし

て、HTMLの規格は10年間、仕様としての進化は止まったままとなっている。しかし、この間、Web技術そのものは格段の進歩を遂げた。そして、Webブラウザのリッチクライアント化は大きく進んだ。たとえば、リッチクライアント技術 Ajax の中核「XMLHttpRequest」は、IEの独自規格であるにも関わらず、全てのPCブラウザで採用され、デファクトスタンダードとして広まった。2004年には、W3Cの不満を持つブラウザベンダが、WHATAGを結成した。Webの仕様は、WHATAGが策定した規格が優位となった。現在、HTML5の規格を、グーグルの主要メンバーがW3Cで策定中である。HTML5は、2010年に正式勧告される。マイクロソフト社のシェアやプレゼンスが落ちていることもあり、今度はIEも準拠せざるを得なくなると予測される。特に、JavaScriptはマイクロソフトではJscriptと呼ばれ、仕様の違いが大きかった。このような仕様の違いは、JavaScriptのライブラリで吸収してきた。CSSの仕様の違いは、CSSハックと呼ばれる無駄なテクニックで回避してきた。しかし、WebデザイナーとWebプログラマの不満は限界に達しつつある。すでに、YouTubeはIE6ユーザに、ブラウザのアップグレードを促すメッセージを表示している。Ajaxがグーグルによって使用されて以降、ブラウザは本質的に変化した。ブラウザは、すでに、WebサーバからWebサーバへ渡り歩き、Webページを表示するだけのアプリケーションではない。急速にWebサービスのプラットフォームの比重が高くなってきている。グーグルがJavaScriptの実行速度に特化した「Google Chrome」をリリースし、「Google Chrome」を動かすための「Google Chrome OS」を開発中であることが、それを裏付けている。また、OSも、ブラウザを動かすためのプラットフォームと化しつつある。マイクロソフトがリリースしたVISTAの不調がこれを裏付けている。また、いまだに圧倒的なシェアを持つマイクロソフトOfficeも、次期バージョンでは、Webサービス版が登場する。本論文では、このような激しいWebの潮流、および、その中にある技術についてまとめた。また、Web制作のカリキュラムとして何を盛り込むべきか、また、盛り込むべき根拠はどこにあるかについて論じた。

(注)

- (1) 電話線を使ったダイヤルアップやADSLなどのインターネットアクセス回線がある。しかし、アクセス回線は、インターネットサービスプロバイダまでの接続を提供するためのものであり、アプリケーションとはいえない。
- (2) 以前の一般向けのクライアント用OSには、TCP/IPの通信機能は実装されていなかった。インターネットに接続するためには、UNIXマシンか専用の高価な通信ソフトを購入するしかなかった。インターネットが急速に普及したのは、1995年に発売されたWindows95にTCP/IP機能が標準添付されていたのが一因である。
- (3) 実際にはCSSを適用しなければ、ブラウザ上では変化がない。グループ化しても、むしろ、ソースが複雑になるデメリットの方が大きい。CSSを適用しないのであれば、divを導入する理由はない。
- (4) 本来a要素以外にも疑似クラスはつけられるが、IEでサポートされていないので、滅多にもちいられない。また、:focusという疑似クラスもあるが、これもIEでされていない。
- (5) 値が0のときはいかなる単位でも0は0であるので省略できる

参考 URL

- 1) [http://ja.wikipedia.org/wiki/Wikipedia: 著作権](http://ja.wikipedia.org/wiki/Wikipedia:著作権)
- 2) http://heaven7.sakura.ne.jp/reference/xhtml_ref/xhtmltmp.html
- 3) <http://htmlint.itc.keio.ac.jp/htmlint/htmlint.html>