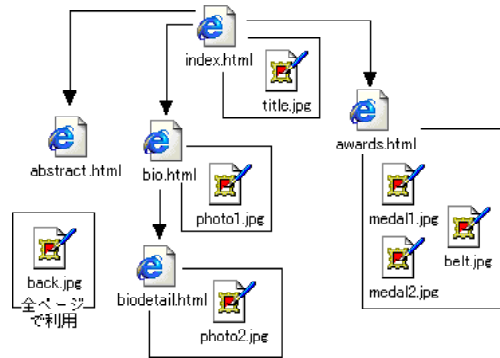


資源の分割・整理とパス

WWW の階層構造

たとえば、あるスポーツ選手の情報を載せようとして、以下のような構成の WWW ページを作るとする。

- ・タイトル
 - ・概要
 - ・生い立ち
 - ・アマチュアの頃
 - ・受賞歴など

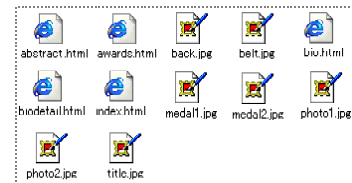


これを画像ファイルの関係も合わせて図示すると上のようになる。つまり実体は上のような HTML ファイルと絵による合計 5 ページの WWW サイトとなる。WWW ページがある程度「階層的」に構成されているのが判るだろう。

ただし、上の例では back.jpg という画像ファイルは index.html というページを除く全ページで使われるために、完全な階層構造にはなっていない。(このボタンは ◀ のような「元に戻る」ことを意味するアイコンなのだろう)

ところが Windows 上でこのファイル群を扱うときは、上のように階層構造とは関係なく、例えば右図のようにアルファベット順に整列して表示される。

このまま例えば 10 ページ、30 画像ファイルになると整理が付かなくなり、WWW 製作時に混乱することが判るだろう。

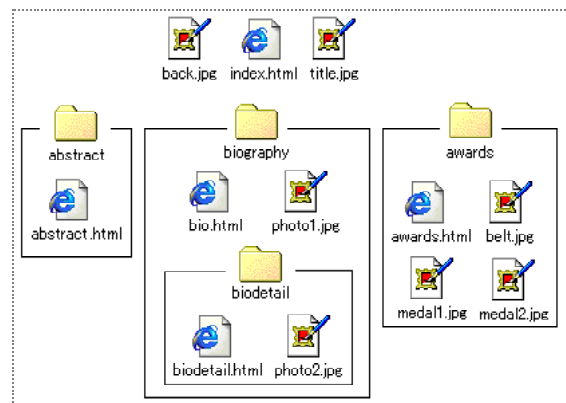
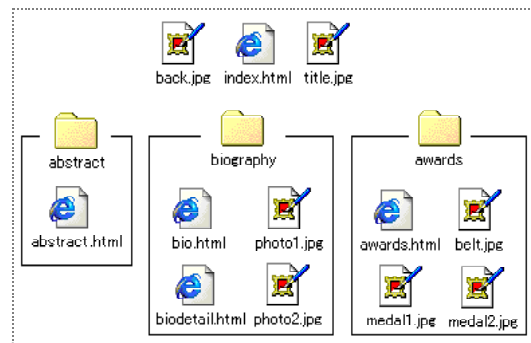


フォルダを利用したファイルの階層配置

Windows にはフォルダと呼ばれるもの (📁 のようなアイコン) があり、そのなかにフォルダやアイコンを入れておくことができる。つまりファイルを階層的に整理することができるわけで、この機能を利用してファイルを整理してみると、例えば右図のようになる。

つまり index.html, back.jpg, title.jpg という三つのファイルのある場所には他に三つのフォルダだけが見えており、フォルダをそれぞれ開くと、そこに各パートの HTML ファイルや画像ファイルがある。これならページ数が増えてもそれほど混乱しないだろう。

もちろん biography 以下を、さらに階層化して、右図のようにしてもよい。この場合は biography フォルダを開くと、そこには bio.html と photo1.jpg の二つのファイルと、biodetail というフォルダが一つ見えることになる。さらに biodetail フォルダを開くと biodetail.html と photo2.jpg ファイルが見える。



フォルダを越えるリンクの書き方

1. 「すぐとなり」へのリンク

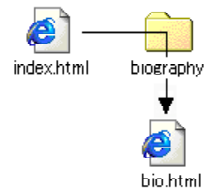
前ページの最初の例のような、index.html から「すぐとなりにある bio.html 」へのリンク（右図）は下記のように記述する。



```
<a href="bio.html">生い立ち</a>
```

2. 「あるフォルダの下」へのリンク

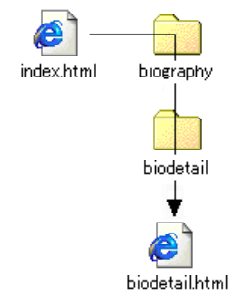
フォルダを利用して整理した後の例のような、「biography フォルダの下の bio.html」へのリンク（右図）は下記のように記述する。



```
<a href="biography/bio.html">生い立ち</a>
```

3. 二段階下へのリンク

同様に、「biography フォルダの下の biodetail フォルダの下の biodetail.html」へのリンク（右図）は下記のように記述する。



```
<a href="biography/biodetail/biodetail.html">アマチュアの頃</a>
```

4. 「すぐとなり」の絵を表示する

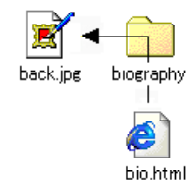
「すぐとなり」の絵をページ中に表示させるためには下記のように記述する。

```

```

5. 「ひとつ上」の絵を表示する

フォルダを利用して整理した後の例では、back.jpg アイコン画像を、abstract.html, bio.html, awards.html ファイルから利用したい。例えば bio.html ファイルに「一つ上の back.jpg」を表示させる（右図）には下記のように記述する。



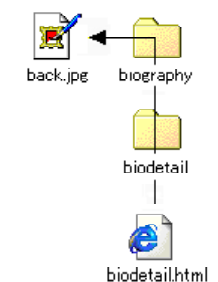
```

```

こうしなければ各フォルダに back.jpg ファイルの複製を作っておかなければならない。結果的に管理するファイル数が増えて面倒である。

6. 二段階上の絵を表示する


さらにもう一つ上のフォルダの back.jpg を利用したい場合、つまり「一つ上の、もう一つ上の back.jpg」を表示させるときは下記のように表現する。

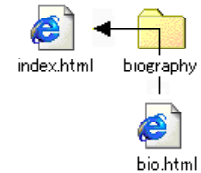


```

```

7. 応用例 1. ひとつ上の絵を利用して、一つ上にリンクを張る

実際の利用場面では、back.jpg つまり  のようなアイコン画像をクリックすると、index.html を表示するようにするだろう。つまり、



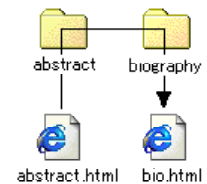
1. bio.html から「一つ上の index.html」に向けたリンクを、
2. 「一つ上にある back.jpg」画像に対して作成することになる

そのためには下記のように記述する。

```
<a href="../index.html"></a>
```

8. 応用例 2. となりのフォルダの HTML ファイルへのリンク

「概要」ページに「次のページに進む」というリンクを作りたくなった場合、つまり右図のような状態での abstract.html から bio.html へのリンクは、「一つ上の隣にある biography フォルダの下の bio.html」へのリンクとなる。そのためには下記のように記述する。



```
<a href="../biography/bio.html">[次のページ]</a>
```

パス (path, 経路)

こうしたファイルの位置を示す記述のことを「ファイルのパス(path)」と呼ぶ。ここで示したのは「あるファイルから見て(あるファイルを起点として)、目的のファイルがどこにあるか」を表現するもので、これを「相対的なパスの表記」と呼ぶ。コンピュータシステム全体から見て、そのファイルの場所を絶対的に指示する「絶対パス」表記の方法もあるが、今は説明しない。

補足:「../」とは何か?

「/」はフォルダの存在を示す区切りであるが、これは Unix という OS の慣習からきている。WWW というシステムが開発され、またインターネットに登場したときに主として利用されていた OS が Unix であったため、Unix での階層表記の流儀が引き継がれたのだろう。「..」が「ひとつ上」を示すのも Unix の慣習である。では何故 Unix では「..」をそのように使ったのか、ということについては講師は知らない。調べてみると面白いだろう。

OS とは: OS という用語には学術的な定義が正しく存在するが、ここでは単に Windows や Macintosh という名前と呼ばれる利用環境だと考えればよい。Windows と Macintosh で操作方法や流儀がかなり異なるのと同様、Unix はこの実習で利用している Windows 環境とは流儀の異なるコンピュータ利用環境である。

もう少しだけ雑学: Unix を真似て階層的なファイル管理方法を取り入れた DOS/Windows は「\」(または¥)を区切りに利用している。一つ上は同じく「..」である。このあたりも面白い。