情報活用演習（応用編 1）

担当：坪田・嶋村・小塚；TA：川﨑

2018 年 5 月 16，23 日

日程

5/16　第 9 回　応用編 1　Web ページを作ろう(1) HTML 言語の成り立ちと Web ページの作成
5/16　第 10 回　応用編 1　Web ページを作ろう(2) Perl を使った動的 Web ページの作成
5/23　第 11 回　応用編 1　Web ページを作ろう(3) シェルとコマンド
5/23　第 12 回　応用編 1　Web ページを作ろう(4) 簡単なプログラミング

はじめに

1. 基礎編で学んだ内容は習熟しておくこと．
2. 友達同士で教えあうことで，理解が深まる．
3. 公開され，実際に動作している他の Web ページを参考にすること．

応用編 1 の内容

1．HTML の説明
1-1．歴史と概念
1-2．タグ
2．HTML 文書の作成
2-1．テキストファイルの HTML 化
2-2．見出しタグ
2-3．他のページへのリンク
2-4．画像へのリンク
3．ホームページの作成
3-1．public_html サブディレクトリの作成
3-2．index.html の作成
3-3．ホームページ公開試験と公開の確認

WWW（World Wide Web（ワールドワイドウェブ），Web（ウェブ））

- 現在，インターネットで広く用いられているドキュメントシステム．文章の内容に加えて，構造や見え方を HTML 言語で記述する．また，文中に画像などのデータや他の文章の位置を埋め込む（ハイパーリンク）ことができる．
- 1989 年，欧州原子核研究機構（CERN，セルン，サーン）の Tim Berners-Lee により，論文閲覧システムとして考案された．当初，コンテンツはテキストのみ．1991 年，一般に公開．1990 年代に，インターネット上でのドキュメントシステムとして普及し，現在一般化している．W3C（World Wide Web Consortium（WWW コンソーシアム））が標準化している．
- 現在は，PC だけでなく，携帯端末（スマートフォンやスマートテレビなど）でも閲覧できる．

用語

- サーバ～情報を渡す仕組みを持ったコンピュータ．さまざまなサービスを提供する．Web サーバや FTP サーバなど．
- クライアント～Web サーバからデータを受けとる端末．「依頼人・顧客」という意味．PC や携帯電話，PDA などをクライアントとして利用できる．
- サーバクライアントシステム～ネットワーク上で，サーバとクライアントの二者からなる関係．
- プロトコル～サーバとクライアントが通信を行う際の約束事・取り決め．Web については，http というプロトコルを利用する．
- インターネット（Internet）～全世界のネットワークを相互接続したコンピュータネットワーク．通信プロトコルに TCP/IP を用いる．

Web サーバと Web ブラウザの関係
　Web ブラウザからの請求に対して，Web サーバが情報を送り返す．送り返された情報を Web ブラウザが解釈し，解釈した結果を表示する．

Web サーバ（web server）～WWW において，HTML の文章や画像などを蓄積し，要求に応じて情報を渡すコンピュータやソフトウェアなど含めた全体の仕組み．Web サービスを提供する．
1. Web サーバ上にあるファイルを，クライアントに渡す．
2. Web サーバ上で実行されたプログラムの結果をクライアントに渡す．

Web ブラウザ～情報閲覧ソフト．Web コンテンツの解析・プログラム実行・閲覧．「Web ページを見る」ためのソフト．
- Internet Explorer（IE）
- Firefox
- Opera
- Safari
- Chrome　etc.

Web ブラウザの役割
1. Web ページの場所を指定する～URL アドレス欄にアドレスを入力することで，特定の Web ページを指定する．
2. Web ページを解析・表示する～HTML ファイルを解析して，人間に理解しやすい表現に変換する．
3. Web ページ閲覧のサポート～閲覧を便利な機能で補助する．「戻る」ボタンによる直前にみた Web ページへ戻る．ブックマークによる記録．「更新」ボタンによる情報の更新．

Web サーバと Web ブラウザのやりとり
1. Web ブラウザを使って，Web サーバにアクセス．
2. Web サーバからデータを送信．データには，テキストや画像などのファイルとプログラムを実行させた結果の 2 種類がある．
3. Web ブラウザが受信，情報の解釈，表示．

Web ページが表示されるまで
解析（パーシング）→整理・プログラム実行→表示（レンダリング）
- **解析**（パーシング）～Web ブラウザが，HTML ファイルの情報を解析する．ファイルが，HTML の文法に則って要素を解析する（HTML パーサー）．
  - ◆ parse：文法的に説明する，語法を解説する，構文分析する，文法・構文を解剖・解析する．
- **整理**～HTML を解析したデータを整理し，リンクされたファイルなどを取り込む．
- **プログラム実行**～Web サーバ上でプログラムを動かし，その結果をサーバから得る．動的な Web サイトでは，Perl や PHP，JavaScript などのプログラム言語を用いたプログラムを利用する（CGI（Common Getaway Interface））．
- **表示**～見出しや本文などの大きさや位置，リンクされた画像などを，取り込まれたデータを解析結果に従って画面に表示する（rendering レンダリング）．

Web ページ
　ページ記述言語の HTML（HyperText Markup Language（ハイパーテキストマークアップランゲエージ））で書かれたファイル．拡張子 html/htm をもつテキストファイル．

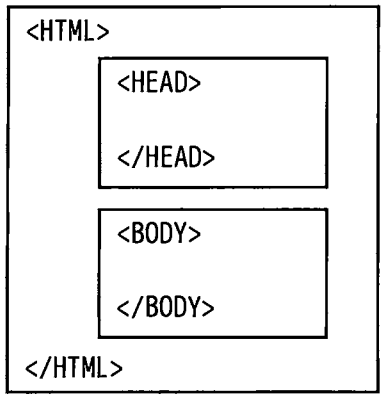
- ハイパーテキスト（HypnerText）～文章同士の結びつき，リンクを含むテキスト．
- マークアップ（Markup）～組版を指示するという印刷用語．
- マークアアップ言語（Markup Language）～文章構造を指定する言語．

注．タグがなくても，拡張子を html とすることで，最低限の内容は表示される．

Web ページの構造

テキストで，本分などの内容に，それぞれの内容の属性（見出し，大きさなど）をタグで指示する．

HTML ファイルの基本構造



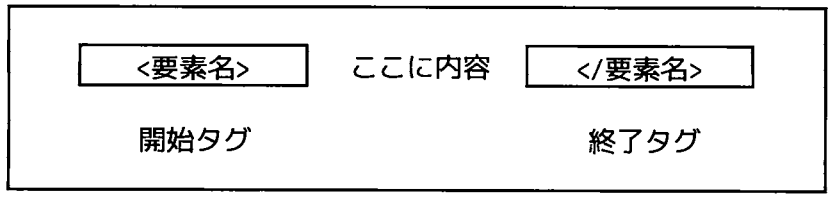
一般的な Web ページの必要最小限 HTML

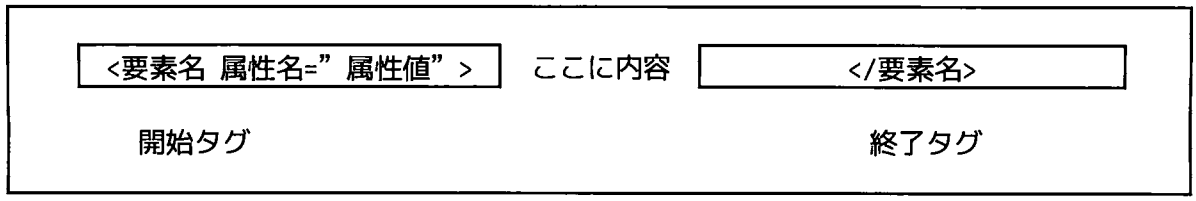
```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
        <meta http-equiv="content-type" content="text/html; charset=shift_jis">
        <title>広島大学理学部生物科学科</title>
</head>
<body>
<p align=center>広島大学理学部生物科学科</p>
<br>
<p>これが最低限の Web サイトです．</p>
</body>
</html>
```

- 下線部分（3 カ所）は変更することができる．
- <body>～</body>の部分が，本文に該当する．ここを自分のページに変更する．
- 保存の際のエンコーディングは Shift-JIS にする．

タグ

- <と>で囲まれた記号．内容の属性を示す印．
- 属性を与える場所で，本文中に直接埋め込んでいく．
- 両カッコ（<と>）で囲んで表現される．本来は<で始まり，>で終わる符号を表す用語である．
- 基本的に，タグは半角文字で書く．





ページの作り方
- エディタなど，テキストを直接編集できるプログラムを開く．
- 内容を記述し，タグで飾り付ける．タグで内容を囲む．
- タグは半角文字で表記する．
- 基本的に，タグは開始タグで始まり，終了タグで終わる．（ただし，終了タグを省略できる要素もある．
や<hr>など．）
- 半角スペースや改行，タブは Web ブラウザで解釈される際，無視される．

入力の際の注意点
- エディタで入力する際，間違いがなく入力されていることを十分に注意を払って確認する．
- 半角文字と全角文字を区別して入力する．
- エディタで入力後，ファイル名をつけて保存する．保存の際，エンコードを「Shift-JIS」にする．

タグの種類

A．本文や見出しにかかわるもの

p　本文の段落
例． これがはじめの段落 次の段落

h1, h2, h3, h4, h5, h6　　見出しの大きさ
例．<h1>レベル１の見出しです</h1>

br　改行
例．このあと改行
が入ってますか？

hr　罫線（水平線，区切り線）
例．水平線↓<hr>

B．文字の見た目

b　太字
例．太字です．

i　斜体
例．斜体です．

u　下線
例．<u>下線</u>です．

font　　フォントの大きさ
例１．フォントのサイズ１です．
例２．フォントのサイズはプラス・マイナスで相対表記もできます．

C．レイアウト

center　　中よせ
例． 確かに中によっています．

D．リンク

a href="アドレスなど"　　アドレスへのリンク．外部でも，ローカルでも可能．
例．広島大学理学部生物科学同窓会

img src="アドレスなど"　　画像ファイルのリンク．外部でも，ローカルでも可能．
例．<img src="http://www.geocities.jp/preaxis/dou10.gif">

E．色

font color="色"　　文字に色を付けます．色の名前または番号が指定できます．

例. 赤

body color="色"　　背景に色を付けます．色の名前または番号が指定できます．

例. <body bgcolor="silver">内容</body>

これらのほかに，テーブルやリストなどがあります．

色の指定

#rrggbb

rr: 赤; gg: 緑 ; bb: 青

それぞれ，16 進数で 00〜ff（10 進数で 0〜255）までの値をとる．

標準の 16 色

#000000	black	#00ffff	aqua
#808080	gray	#00ff00	lime
#c0c0c0	silver	#ffff00	yellow
#ffffff	white	#ff00ff	fuchsia（ピンク）
#0000ff	blue	#808000	olive
#000080	navy（紺）	#800080	purple
#008080	teal（緑青）	#800000	maroon（茶）
#008000	green	#ff0000	red

ソースを見てみよう

適当な Web サイト（標準的でない場合もある）を開いて，Web ブラウザの「表示」→「ソース/ページのソース」あるいは「右クリック」→「ページのソースを表示」などを選択すると HTML ソースが見える．

ページの例

以下のソースをエディタを使って入力して，動作確認すること．

参考文献

1. アンク． 2007． HTML タグ辞典，第 6 版，XHTML 対応． Pp. 384． 翔泳社，東京． [ISBN-10: 4798113522; ISBN-13: 978-4798113524]
2. 大藤 幹． 2007． 詳解 HTML&XHTML&CSS 辞典，第 3 版． Pp. 561． 秀和システム，東京． [ISBN-10: 4798016020; ISBN-13: 978-4798016023]
3. 水津弘幸・石井 歩・C&R 研究所． 2008． HTML+CSS Handbook, 3rd Edition, Web 系ハンドブック． Pp. 640． ソフトバンククリエイティブ，東京． [ISBN-10: 4797344490; ISBN-13: 978-4797344493]

ここにあげたものは，比較的安価で，現在市販されている書籍のうち代表的なものです．これら以外にも多くの書籍や参考書があります．書店や図書館などで自分に合ったものを探してください．

補足 1. 画像の種類

基本的にどのようなファイルでも Web サイトで公開できる．ただし，ページの中の画像として利用されるのは，ファイルサイズや汎用性の高い JPEG や GIF，PNG などの形式（画像フォーマット）が一般的．

- JPEG〜色数が多い画像の圧縮に適した画像圧縮形式のひとつ．写真などに向く．
- GIF〜色数が少ない画像（256 色まで）に適した画像圧縮形式のひとつ．アイコンやイラストなどに向く．また，パラパラ漫画の原理でアニメーションも作成できる．
- PNG〜GIF や JPEG に代わる画像圧縮形式のひとつ．GIF の特許の関係で開発された．WWW に関する標準化団体 W3C の推奨形式．

```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
        <meta http-equiv="content-type" content="text/html; charset=shift_jis">
        <title>広島大学理学部生物科学科</title>
</head>
<body bgcolor="silver">
<p align=center>広島大学理学部生物科学科</p>
<br>
<p>これが最低限の Web サイトです．</p>

<p>
<p>これがはじめの段落</p><p>次の段落</p>
<h1>レベル 1 の見出しです</h1>
このあと改行<br>が入ってますか？
水平線<hr>
</p>

<p>
<b>太字</b>です．<br>
<i>斜体</i>です．<br>
<u>下線</u>です．<br>
<font size="1">フォントのサイズ 1 です．</font><br>
<font size="3">標準はサイズ 3 です．</font><br>
<font size="+1">フォントのサイズはプラス・マイナスで相対表記もできます．直前までの
基本のフォントサイズ 3 に 1 加えるのでサイズ 4 になります．</font><br>
<font size="-1">フォントのサイズはプラス・マイナスで相対表記もできます．直前までの
基本のフォントサイズが 3 なのでサイズ 2 になります．</font><br>
</p>

<p>
<center>確かに中によっています．</center><br>
</p>

<p>
色付けます．<font color="red">赤・あか・red</font>
<font color="blue">青・あお・blue</font>
</p>

<p>
こちらは，<a href=" http://home.hiroshima-u.ac.jp/biodoso/">広島大学理学部生物科学
同窓会の Web サイト</a>です．
</p>

<p>
このあたりに画像が現れます．<br>
png ファイルの例．<br>
<img src="http://www.hiroshima-u.ac.jp/fix-images/ja//header.png"><br>
gif アニメーションの例．<br>
<img src="http://www.geocities.jp/preaxis/dou10.gif">
</p>

</body>
</html>
```

まとめ
- ページファイルはテキストエディタで加工する．
- タグがなくても，拡張子を html とすることで，最低限の内容が表示される．
- 実際には，タグを使って，わかりやすいページを作成する．
- WWW サーバ（またはローカルでも可）に置かれた html ファイルを，ブラウザで開くことで，ページが表示される．

一般的な Web ページの必要最小限 HTML

```
<!doctype html public "-//W3C//DTD HTML 4.0 Transitional//EN">
<html lang="ja">
<head>
        <meta http-equiv="content-type" content="text/html; charset=shift_jis">
        <title>広島大学理学部生物科学科</title>
</head>
<body>
<p align=center>広島大学理学部生物科学科</p>
<br>
<p>これが最低限の Web サイトです．</p>
</body>
</html>
```

- 下線部分（3 カ所）は変更することができる．
- <body>～</body>の部分が，本文に該当する．ここを自分のページに変更する．

具体的にページを作ってみよう
1. エディタを開く．
2. テキストを入力する．例えば，自分の名前など．
3. ファイルを保存する．保存の際に拡張子を.html にする．
4. ブラウザで保存したファイルを開いてみる．正常に見える場合と，文字化けする場合がある．

4.1. 文字化けする場合は，保存の際 Shift JIS で保存する．
4.2. それでも文字化けする場合は，ブラウザの文字コードを Shift JIS に指定する．
4.3. さらに文字化けする場合は，上の枠内をすべて入力してみる．

5. 最終的に，枠内のテキストをすべて入力する．
6. 下線部を変更してみる．例えば，前回のテキストを参考にしてタグを入れてみる．

ページの作成
- さまざまなタグを使って，個人のページを作成する．
- 前回のプリントの課題の内容を，ページの中に反映させて，保存する．
- 保存の際に，ファイル名を index.html とする．

ページの公開
- 前提として，Web サイト公開のための試験に合格しておくこと．
- 「http://home.hiroshima-u.ac.jp/アカウント/」が個人の Web サイトのアドレスになる．
- ディレクトリ public_html の中に置いたファイルが公開される．
- index.html が標準で表示される．

公開までの流れ
1. 試験を受けて，合格する．
2. 個人ディレクトリのルートに，「public_html」のいう名前のディレクトリを作成する．
3. 作ったディレクトリの中に，index.html という名前のファイルを置く．

4. ファイルのパーミッションを適宜変更する．
5. Web ブラウザで，自分のサイトにアクセスし，表示を確認する．

発展．XHTML（Extensible HyperText Markup Language）について
HTML をマークアップ言語のひとつ XML で定義し直したもの．さまざまなブラウザが作られ，ブラウザごとの違いから現在の HTML が，今後利用範囲が広がることが想定されている XML と一部整合性が取れていない部分がある．そのため，XML に準拠して HTML を定義し直して XHTML が作られている．現在過渡期ではあるが，XHTML では構造を記述することに重点を置き，レイアウトは CSS を使って表現することになっている．タグが小文字で表記されることや，修了タグ
が
と表記されるなどいくつか違いがある．

発展．CSS（Cascading Style Sheets）について
HTML や XHTML で作られた Web ページのレイアウトを定義する規格，またそれを定義したファイル．ブラウザごとに違いの多い見た目に関するタグなど肥大化しすぎた HTML の反省から，レイアウト（見た目）に関する内容を切り離して，自由度を大きくした．ページの記述が構造だけになり，利用者が自由に選択できるという利点がある．

課題
次の時間までに，以下の内容を行うこと．課題については，演習時間が始まるまでに，必ず終わらせておくこと．

1. Web 公開チェック（情報メディアセンターの利用登録システム内）を受け，試験にパスしておくこと．
2. 個人のホームページのもととなる内容を作成しておくこと．氏名だけでなく，趣味，現在取り組んでいること，将来の目標，生物学に関する項目の解説（1 項目，200-400 文字で）の 5 点は必ず含めること．また，画像ファイル（風景写真など，256 x 256 pix 以上）も用意しておくこと．解説などは著作権に十分留意し，自分で考えたオリジナルな内容とすること．
3. 個人ページを公開しておくこと．

■講義の目的

生物学の実験を行う場合，さまざまな場面でコンピュータが必要となる．今回は，コンピュータの基本操作に慣れ，基本的な使い方を習得することを目的とする．また，研究を行う上で有用なプログラム作成の基礎を PHP または Perl を使って習得する．

■注意事項

- 基礎編の内容を十分に理解し，習得しておくこと．
- 十分に理解している場合は Windows 環境・Unix（Linux）環境のいずれを用いてもよいが，説明中は指示には従うこと．
- 理想的にはいずれの環境も利用できる技能を身につけることが望ましい．
- 説明は時間の関係でいずれか一方しかできない場合がある．この場合，基本的に Unix 環境による説明を優先・基本とする．時間が許せば MS-DOS についても概略を述べる．
- ひとりで悩まず，分からなくなったら近くにいる人にたずねてみよう．
- コンピュータの操作は，他人の操作をいくら見ても上達しない．必ず自分で操作してみること．
- メールでレポートを提出する場合は，必ずメールの Subject（題名）と本文のはじめに学生番号および名前を明記すること．（不明な場合は，採点できないこともある．）
- メールが必ずしも届くとは限らないので，メールの内容を印刷したものを理 A504 生物系事務室に提出すること．左上をホッチキスでとめて，名前と学生番号を明記すること．

■資料

- 配布資料
- 情報活用演習のページ（広島大学理学部生物科学科のサイトの中）
  <http://home.hiroshima-u.ac.jp/biology/riseijok/>

演習に関する情報は上記サイト（およびそのリンク先）に掲載するので，よく見ておくこと．必要ならば，ブックマークに入れておくとよい．

■目標

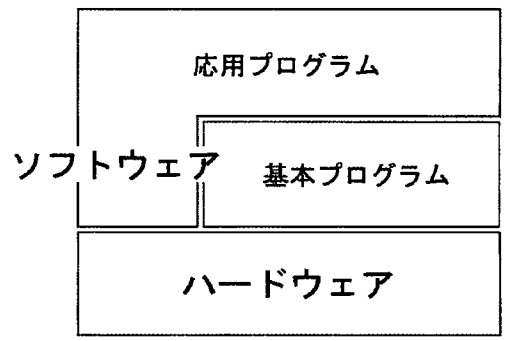
- エディタ・HTML・ブラウザの復習．
- UNIX のシェルコマンドになれる．
- PHP や Perl を使って簡単なプログラミングを行う．（今年度は Perl）
- 自分の Web サイトにカウンタをつける．

■コンピュータの構成

コンピュータはハードウェアとソフトウェアより構成される．

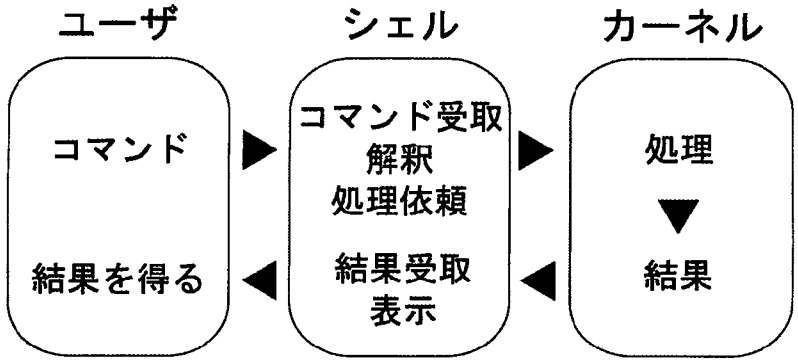
- ハードウェア
- ソフトウェア
  - 基本プログラム（OS オペレーティングシステム（カーネル，ライブラリ，シェル，基本的なコマンド，ウインドウシステムなど））
  - 応用プログラム（ブラウザ，ワードプロセッサ，表計算プログラムなど）

UNIX，Windows，MacOS などでは，基本プログラムと一部の応用プログラムまでがまとめられて一つのシステムになっている．



■カーネルとシェル

- カーネル ～ ハードウェアやメモリの管理などを実際に行う基本的な部分．



- シェル ～ カーネルとユーザとのやり取りを実際に行う部分.
  1. プロンプトを表示して，コマンド入力を待つ.
  2. 入力されたコマンドを読み取り，解釈した後，必要な処理をカーネルに依頼.
  3. カーネルから帰って来た結果を受け取り，表示などする.

■GUI と CUI

- CUI（Character-based User Interface）～ インターフェースのうち，表示を文字のみ，操作をキーボードからの入力だけを用いて行うもの．基本的にコマンドを入力することで操作する．GUI に比べて，作業の自動化などが比較的簡単にできる．キャラクタユーザインターフェース，コマンドラインインターフェース，コンソールも同じ意味．Unix や MS-DOS で利用できる．シーユーアイと読む.
- GUI（Graphical User Interface）～ インターフェースのうち，表示にアイコンなどの画像，操作の基本的な部分をキーボードではなく，マウスやトラックボール，タブレットなどのポインティングデバイスによって行うもの．コマンドを記憶していなくても，直感的に操作しやすい．グラフィカルユーザインターフェースも同じ意味．ジーユーアイまたはグーイと読む．MS-Windows や Mac OS などでは OS に GUI が実装されている．また，OS 本体に GUI が組み込まれていない Unix 系の OS でも，X Window System というソフトにより，GUI 機能を追加して利用できる.

■シェルとその実際

- コマンド行でコマンドを受け取る（コマンドインタプリタ）.
- いくつか種類がある．標準的なシェルとしては，sh（Bourne shell [B shell]），bash（Bourne Again shell），csh（C shell），ksh（Korn Shell [K shell]），tcsh，zsh などがある.

1. コンソールを起動する．ディスクトップで右クリックしてメニューを表示
2. コンソールの窓が開く．$（または%）が表示されて，カーソルが点滅し，コマンドが入力されるのを待った状態になる.

コンソールを終了したいとき.
% exit

ログアウトしたいとき（ダイヤルアップなどで接続した場合）.
% logout

■基本的なコマンド
以下の内容について確認する．以下本文中の%はこの待機状態を意味するので，入力の必要はない．太い文字で書かれた部分だけを入力する.

% date（現在の日付と時間を表示する.）
2005 年 06 月 11 日 17 時 19 分 55 秒

% **cal**（今月のカレンダーを表示する.）
```
   June 2005
 S  M Tu  W Th  F  S
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

% **!!**（直前のコマンドの繰り返し）
```
cal
   June 2005
 S  M Tu  W Th  F  S
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

% **!da**（もっとも最近使った da ではじまるコマンドを繰り返す.）
```
date
2005 年 06 月 11 日 17 時 19 分 55 秒
```

% **ls**（カレントディレクトリの内容を表示する.）

<ディレクトリについて>
~（チルダ）　ホームディレクトリ
.　　カレントディレクトリ（現在いるディレクトリ）
..　　一つ上のディレクトリ

% **ls**（カレントディレクトリの内容表示．環境によって表示される内容は異なる.）
```
-                   public_html         Desktop             GNUstep
Mail                Mailboxes
```

% **mkdir jisshu**（ディレクトリ jisshu を新たに作成）

% **ls**（現在いるディレクトリの中にディレクトリ jisshu が作成されたことを確認）
```
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu
```

% **rmdir jisshu**（ディレクトリ jisshu を削除）

% **ls**（ディレクトリ jisshu が削除されたことを確認）
```
-                   public_html         Desktop             GNUstep
Mail                Mailboxes
```

% **mkdir jisshu**（再びディレクトリ jisshu を作成）

% **ls**（ディレクトリ jisshu が作成されたことを確認）
```
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu
```

% man ls（ls のマニュアルを表示）
```
ls(1)                                                                ls(1)
名称
      ls, lc, l, ll, lsf, lsr, lsx - ディレクトリの内容のリスト

構文
      ls [-abcdefgilmnopqrstuxACFLR1] [names]
        .
        .
        .
```

<man コマンド名>
コマンドのマニュアルを表示させる．コマンドの使い方を知りたい場合に利用できる．man は q で終了，スペースで次画面，b で全画面への移動．

% ps（現在動いているプロセスの表示）
```
   PID TTY      TIME COMMAND
 23078 pts/30   0:00 csh
 23105 pts/30   0:00 ps
```

<プロセスについて>
UNIX はマルチプロセス（マルチタスク）な OS である．マルチプロセスとは，1 台のコンピュータで，並行して複数の処理を行なうことである．最近のコンピュータでは一般的な機能になったが，以前は一度にひとつの処理を行うのが普通であった．実際には，CPU が処理する時間を細かく分割して（時分割機能），複数の処理に割り当て，一見同時に処理が行われているように見せている．このような複数の処理のひとつひとつをプロセスと呼ぶ．

例えば，ひとつのコマンドが実行されると，プログラムが動き，ひとつまたは複数のプロセスができる．プログラムが終了すると，そのプロセスは消える．このような管理は普通 OS レベルで自動的に行われている．

それぞれのプロセスは PID（プロセス ID）と呼ばれる番号で管理されている．PID を調べたい場合，ps コマンドで見ることができる．

プロセスは正常に終了した場合，自動的に消滅する．しかし，プログラムを実行している最中に ^Z（CTRL+Z）で抜けるとプロセスはそのまま残ってしまう．自分のプロセスは終了前に責任をもって消しておく必要がある．ps コマンドで PID を調べ，kill コマンドで消すことができる．

% ps（現在のプロセスの確認.）
```
   PID TTY      TIME COMMAND
 28780 pts/3    0:00 ps
 28696 pts/3    0:00 csh
```

% cat（cat コマンドから^Z で抜ける.）
^Z（実際には表示されない）
停止

% ps（プロセスの確認.）
```
   PID TTY      TIME COMMAND
 28808 pts/3    0:00 ps
 28799 pts/3    0:00 cat
 28696 pts/3    0:00 csh
```

% kill 28799（PID が 28799 のプロセスを殺す.）

```
% ps（プロセスの確認．cat が終了したことを確認．）
   PID TTY        TIME COMMAND
 28834 pts/3      0:00 ps
 28696 pts/3      0:00 csh
[1]  +終了                  cat
```

どうしても消えない場合は，

% kill -9 PID（PID は実際の番号．）

でプロセスを殺す．

シェルや ps 以外のプロセスが残った状態で logout しようとすると警告が出る．ps で PID を調べてプロセスを kill コマンドで消した上で logout する．

```
% cat
abc123（標準入力[キーボード]から入力の後，エンターキーを押す．）
abc123（入力したものが標準出力[ディスプレイ]へ出力される．）
^D（入力を終了したい場合は，CTRL+D を押す．実際には見えない．）

% ls
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu

% cat > test.txt（標準入力の内容をリダイレクトでファイルへ出力する．）
This is a test file.
^D（実際には見えない．）
```

<リダイレクトについて>
UNIX は 3 つの標準入出力を持つ．

- 標準入力　　stdin
- 標準出力　　stdout
- 標準エラー出力　stderr

標準入出力の切り替え機能をリダイレクトと呼び，<と>，|で表される．
コマンド < 入力ファイル名 > 出力ファイル名
の形式が標準的．
指定しない場合，標準入力はキーボード，標準出力はディスプレイである．それぞれ指定することで，入力ファイルや出力ファイルに切り替えることができる．一方だけを指定することも可能である．

```
コマンド > ファイル名      出力をファイルに上書きする．
コマンド >> ファイル名     出力をファイルの最後に追加する．
コマンド < ファイル名      ファイルからデータを入力する．
コマンド 1 | コマンド 2    コマンド 1 の結果をコマンド 2 に入れる．

% ls（ファイルが作られたことを確認．）
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu              test.txt

% cat test.txt（ファイルの内容を表示．）
This is a test file.

% cd jisshu（ディレクトリ jisshu の中へ移動．）
```

% ls（ファイルがないので何も表示されない.）

% cd ..（ひとつ上のディレクトリへ移動．ディレクトリ jisshu からホームディレクトリへ.）

```
% ls
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu              test.txt
```

% mv test.txt jisshu（test.txt をディレクトリ jisshu の中へ移動.）

% ls（test.txt が消えたことを確認.）

```
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu
```

% ls jisshu（test.txt がディレクトリ jisshu の中にあることを確認.）

```
test.txt
```

% cd jisshu

% ls

```
test.txt
```

% mv test.txt test2.txt（mv コマンドには名前を変える機能もある.）

% ls

```
test2.txt
```

% rm test2.txt（ファイルを消す.）

% ls（何もないことを確認.）

ここまでで，UNIX のシェルコマンドの基本的なものが大体理解できたはずである.

■Perl を使った簡単なプログラム（スクリプト）の作成
perl を起動する．KTerm を使って以下の操作を行う.
% perl -v（バージョンなどを表示する．perl が動作すると以下の内容が表示される.）

```
This is perl, version 5.005_03 built for PA-RISC2.0
(with 1 registered patch, see perl -V for more detail)
        .
        .
        .
```

% perl -e "print 'Hello';"（Hello と表示させる.）

```
Hello%
```

エディタで以下のスクリプトを入力し，保存する．入力の際，基本的にすべて半角文字で入力すること．入力の際には左の行番号とコロン（:）は入力しないこと.

スクリプト 1．Hello, World!!!と表示する．

```
1:      #!/usr/bin/perl
2:
3:      print ( “Hello, World!!!” );
```

入力後，「ファイル」-「保存」でファイル名を指定して保存する．perl のスクリプトの場合，拡張子は pl とするのが一般的．例として，test.pl として保存する．保存先はホームディレクトリとする．

```
% ls（test.pl が作成されたことを確認）
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu              test.pl
```

早速，スクリプトが動くかどうか確認する．

```
% ./test.pl（作成したスクリプトを実行）
./test.pl: パーミッションがありません。
```

<パーミッションについて>
パーミッションとは，ファイルやディレクトリに対するアクセス権（触ったり見たりすることができるかどうか）のこと．UNIX では，所有者である Owner と，ユーザ全体を意味する Group，その他全ての Other について，読み込み，書き込み，実行の権限が設定できる．

パーミッションの設定には,3 桁の数字が用いられる.一番左の数字が Owner,中央の数字が Group,右の数字が Other を意味する．それぞれの桁は，4（読み込み許可），2（書き込み許可），1（実行許可・ディレクトリー覧許可）の数字を足したものになる．例えば，perl スクリプトを実行ファイルにする場合は，755 の数字を指定する．

```
% chmod 755 test.pl（実行するためのパーミッションを与える）

% ./test.pl（再度スクリプトの実行）
Hello, World!!!

% chmod 644 test.pl（もとのパーミッションに戻す）

% ./test.pl（再度スクリプトの実行）
./test.pl: パーミッションがありません。

% chmod 755 test.pl（再度実行するためのパーミッションを与える）

% ./test.pl（スクリプトの実行）
Hello, World!!!
```

<パーミッションの確認>
ls コマンドなどでのパーミッションの表示は，文字で表される．例えば 755 の場合，-rwxr-xr-x と表示される．一番左の桁はディレクトリであるかどうかを意味し，d であればディレクトリである．その次からはパーミッションの表示である．

左から 2-4 桁が Owner，左から 5-7 桁が Group，右の 8-10 桁は Other を意味する．それぞれ，左から r（読み込み権限あり），w（書き込み権限あり），x（実行権限・ディレクトリー覧権限あり）を意味し，-は権限がないことを意味する．

```
% ll（または，ls -l）
-rwxr-xr-x   1 chubo      user1           46  6月 15日  20:24 test.pl

% chmod 644 testperl.pl

% ll
-rw-r--r--   1 chubo      user1           46  6月 15日  20:24 test.pl

% chmod 755 testperl.pl

% ll
-rwxr-xr-x   1 chubo      user1           46  6月 15日  20:24 test.pl

% ls
GNUstep      Mailboxes    Mail         public_html  jisshu       test.pl

% cd jisshu

% ls（何も表示されない.）

% mv ../test.pl .

% ls
test.pl

% cd ..

% ls
-                   public_html         Desktop             GNUstep
Mail                Mailboxes           jisshu              test.pl

% cd jisshu

% ls
test.pl

% cat test.pl
#!/usr/bin/perl
print ("Hello, World!!!\n");
```

次のスクリプトを入力する.
スクリプト 2. 入力した内容をそのまま出力する.

```
1:      #!/usr/bin/perl
2:
3:      while(<>) {
4:              print;
5:      }
```

スクリプト 3．入力した内容のうち，一致したものだけを出力する．

```
1:      #!/usr/bin/perl
2:
3:      while(<>) {
4:              if(/print/) {
5:                      print;
6:              }
7:      }
```

スクリプト 4．同じディレクトリ内にある pl の拡張子をもつファイルの中身を out.txt というファイルに出力する．

```
1:      #!/usr/bin/perl
2:
3:      open(MSG,” >out.txt” );
4:      while($file=<*.pl>) {
5:              open(IN,” $file” );
6:              while(<IN>) {
7:                      print MSG;
8:              }
9:              close(IN);
10:     }
11:     close(MSG);
```

スクリプト 5．数字当て．

```
1:      #!/usr/bin/perl
2:
3:      $max=10;
4:      $number=int(rand()*($max+1));
5:      print "Input your lucky number (0 to $max)!!!";
6:      while ($guess=<STDIN>) {
7:              chomp $guess;
8:              if($guess==$number) {
9:                      print "You are lucky!!!\n";
10:                     exit();
11:             } else {
12:                     print "Try again!!!\n";
13:             }
14:     }
15:     print "\nBye!!!\n";
```

chmod でパーミッションを実行に変えてからスクリプトを実行させる．途中でやめたい場合は，CTRL+D でプログラムを終了できる．

<プログラムの終了について>

CTRL+C	プログラムの強制終了
CTRL+D	ファイルの終端記号（EOF）
CTRL+Z	プログラムから抜ける（プロセスは残る）

注意事項

1．ファイルやディレクトリの属性の確認
CGI の属性（counter.cgi など）は 755　-rwxr-xr-x
CGI のおいてあるディレクトリ（public_html など）の属性は 755　drwxr-xr-x
CGI によって書き込まれるファイル（count.dat）の最終的な属性は 600　-rw------

2．改行コード
CGI プログラムの改行コードは UNIX 用（LF）で保存する．

3．拡張子
CGI を使うものは　.cgi
SSI を使うものは　.shtml

4．htaccess
htaccess は必ず chmod 644 .htaccess でパーミッション属性を 644　-rw-r--r--

課題．以下の内容を，次週までに済ませておくこと．来週の演習中で使用するため，必ず演習開始までに済ませておくこと．入力したファイルはすべてディレクトリ jisshu に入れておくこと．

1. 以下のキーワードから 3 つ選んで，300 文字以内で簡単に説明せよ．なお，説明は必ず自分でまとめたものとし，説明の最後に文字数を括弧書きで示しておくこと．

   キーワード：生物学，DNA，タンパク質，分類，生態，生化学，遺伝子，発生，陸上植物，緑色植物，光合成，細胞骨格，細胞分裂，バクテリア，生物多様性．

2. スクリプト 2 と同じ機能をもつコマンドを示し，スクリプト 2 の実行結果とそのコマンドの実行結果をそれぞれ別のファイルに保存せよ．

3. エディタを用いて，自分のホームページを書き換える．ホームページには，タイトル，氏名，趣味・特技などの内容を含めて自己紹介を作成すること．ただし，過度な個人情報は控えた方が無難．public_html 内にある index.shtml または index.html ファイルを書き換える．また，それぞれの項目についてはタイトルや氏名についてはフォントサイズを大きくしたり色をつけて目立つようにする．他の項目については自由とする．友達同士でリンクを張っても良い．

4. 資料中のスクリプトをすべて入力しておくこと．入力したファイルはすべてディレクトリ jisshu に入れておくこと．

前回の復習

% cd jisshu（ディレクトリ jisshu の中へ移動.）
% ls
% cat > script1.pl（標準入力の内容をリダイレクトでファイルへ出力する.）
#!/usr/bin/perl

print (“Hello, World!!!”);
^D（実際には見えない.）
% ls（script1.pl が作成されたことを確認）
% chmod 755 script1.pl（実行するためのパーミッションを与える）
% ./script1.pl（スクリプトの実行）
Hello, World!!!

ホームページ作成のながれ
1. ホームディレクトリに公開用のディレクトリを作る．広島大学の Web サーバーでは，public_html というディレクトリを作成すると，その中身がホームページとして公開されるしくみになっている．このためホームページ用のディレクトリの名前は public_html とする必要がある．
2. ディレクトリ public_html のパーミッションを 755 に変更して外部から見えるようにする．
3. ディレクトリ public_html の中に index.html というファイルを作る．このファイルがホームページのトップページになる．広島大学の Web サーバーでは，アドレスのファイル名の指定を省略した場合に，index.html が標準で開くページになる．
4. index.html のパーミッションを 644 に変更する．
5. 作成されたページをブラウザで確認する．

具体的にはコマンドライン上で以下の操作を行う．
% cd（ホームディレクトリに移動）
% mkdir public_html（ディレクトリ public_html 作製）
% chmod 755 public_html（パーミッション変更）
% cd public_html（public_html へ移動）
% cat > index.html（簡易エディタとして cat コマンドを利用して以下の html を作製する．キーボードからの入力がファイルに書き出される．エディタなどを用いてもよい．ただし，ファイルの名前は index.html とし，ディレクトリ public_html に保存すること.）

ホームページ.
```
1:   <html>
2:       <head>
3:           <title>My Home Page</title>
4:       </head>
5:
6:       <body>
7:           <center>
8:           <p><font size="+2"><b>Hello!!</b></font></p>
9:           <p>Welcome to my home page!!</p>
10:          <p>Enjoy your time!!</p>
11:          </center>
12:      </body>
13:  </html>
```

% chmod 644 index.html（自分以外でも閲覧できるようにパーミッションを 644 にする.）
最後にブラウザで確認する．ホームページアドレスは，
http://home.hiroshima-u.ac.jp/アカウント名/
となる.アカウント名とは端末を利用する場合に利用するもので,b+学生番号が標準.index.html

の内容を必要に応じて書き換える．

アクセスカウンタの作製
ホームページにアクセスカウンタをつける．これで自分のホームページへの訪問回数が分かる．このようなしくみを CGI（Common Getaway Interface）と呼ぶ．

1. ホームディレクトリにホームページ公開用ディレクトリ public_html がなければ作成する．
2. アクセスカウンタのスクリプトをエディタで入力して，ディレクトリ public_html に名前を counter.cgi として保存する．

スクリプト 6．ホームページのアクセスウンタ．

```
1:   #!/usr/bin/perl
2:
3:   $cfile = "./count.dat";
4:   open(FILE, "+<$cfile");
5:   flock(FILE,2);
6:
7:   $count = <FILE>;
8:   chomp $count;
9:   $count++;
10:
11:  seek(FILE, 0, 0);
12:  print FILE "$count\n";
13:  printf("%5d", $count);
14:
15:  flock(FILE, 8);
16:  close(FILE);
17:
18:  exit;
```

3. スクリプト counter.pl のパーミッションを 755 に変更する．
   % chmod 755 counter.cgi（CGI として実行できるようにパーミッションを 755 とする．）
4. カウントした値を記録しておくファイルを作る．
   % touch count.dat（保存先の空ファイル count.dat を新たに作る．）
5. 記録ファイルのパーミッションを 644 に変更する．
   % chmod 644 count.dat（ファイル count.dat に書き込みができるようにパーミッションを 644 とする．）
6. index.html をエディタで書き換える．アクセスカウンタが動作するようにする．ホームページのファイルの 10 行目と 11 行目の間に次の内容を追加する．
   ** あなたは<!--#exec cmd="./counter.cgi"-->人目です． **
7. index.html の名前を index.shtml に変更する．
   % mv index.html index.shtml（CGI を含んだページの拡張子 shtml に変更する．index.html を残しておくと，そちらが優先されてしまうので，index.html は消去するか別のディレクトリに移動しておく．）
8. アクセスカウンタが動作することを確認する．
9. 確認後，記録ファイルのパーミッションを 600 に変更する．
   % chmod 600 count.dat

補足２．アクセスカウンタが動かない場合

1．ファイル名の確認
カウンタのスクリプトを script6.cgi にしている場合，「... <!--#exec cmd="./counter.cgi--> ...」を「... <!--#exec cmd="./script6.cgi--> ...」に変更すること．

2. .htaccess の設定
エディタを開いて，以下の内容を入力する．

```
Options +Includes
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
```

public_html の中の index.shtml と同じ場所に，.htaccess という名前で保存する．改行コードは UNIX（LF）にする．念のため，文字コードは EUC-JP にする（Shift-JIS でも動作するはず）．http://home.hiroshima-u.ac.jp/bxxxxxx/index.shtml にアクセスして，カウンタが動いていることを確認する．

発展

ページの様々な作り方
- エディタで直接タグを書いていく．
- ホームページビルダなどの HTML 編集ソフトを利用する．
- LibreOffice Writer などのワープロソフトを使って作成し，html 形式で保存する．
- Wiki や blog などの HTML 自動生成ソフトを利用する．

ブログ Blog やウィキ Wiki―新しい形の Web サイト
ホームページの作成の際には html を記述する必要があるが，最近ではホームページ作成ソフトやワープロソフトの機能を利用することで，簡単に作成できるようになった．しかし，ホームページを公開する際に，ファイルを転送する必要がある点は，依然面倒な状態であった．

最近，よく見かけるホームページに，ブログ Blog やウィキ Wiki というものがある．これらは掲載される内容の性質がやや異なるが，基本的に更新が比較的簡単になるようなものであることが共通している．ちなみに，情報活用演習のホームページで使われているのは Wiki である．

Wiki などは，Perl や PHP などの言語で作られたプログラムが，アクセスの度にページをコンバートして出力している．結果的に，ブラウザでホームページとして閲覧できる．

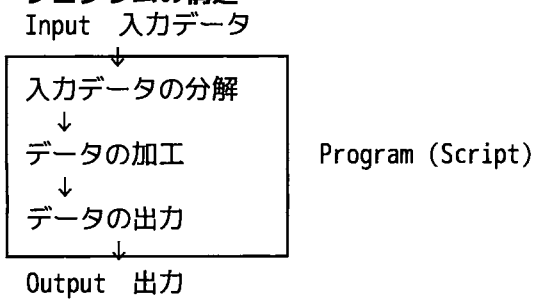
その他，WordPress や Joomla!，XOOPS（ズープス），Drupal なども利用される．これらのウェブサイトの管理や構築，管理に利用されるソフトは，コンテンツマネージメントシステム（Content Management System，CMS，コンテンツ管理システム）と総称される．最近よく利用される Twitter や Facebook のような SNS（ソーシャルネットワーキングサービス，Social Networking Service，SNS）も広い意味で CMS の一種とも言える．

コンピュータの操作の応用
a. プログラミングの基本
分子系統学的な研究を行うとき，コンピュータを用いて様々な情報を取り扱う場面に出くわす場合が多い．このようなとき，処理をさせるための簡単なプログラムを作成できると大変便利であり，研究の効率化につながる．コンピュータの取り扱いになれるために，今回はスクリプト言語 Perl を用いて簡単なプログラムを作成し，データ処理を行う．

計算の方法
- 手（+足）
- 石・棒
- 暗算
- そろばん（東洋とくに日本で発達）
- 筆算
- 数表
- 計算尺（1/100 まで読み取れる）
- 電卓（電子卓上計算機）
- 関数電卓
- ポケコン
- Excel など表計算ソフト
- プログラム

プログラム
現在もちいられているコンピュータの多くは，ノイマン型コンピュータと呼ばれ，計算手続きを記したプログラムと計算に用いるデータをメモリ上において実行する形式をとる．このため，コンピュータに何らかの仕事をさせ，その仕事を自動化する場合，プログラムが必要となる．

プログラムの構造



用語
プログラム program～コンピュータ上で動作するソフトウェア．コンピュータに行わせる計算などの手続きを順に示したもの．現在ではプログラム言語やスクリプト言語を用いて手続きを記述するのが一般的．
プログラミング programming～プログラムを作成すること．コーディング coding ともいう．
プログラム言語 programming languages～コンピュータ上で動作するプログラム(ソフトウェア)を作成するための人工的な言語体系．いくつかの命令語と文法をあわせたもの．

プログラム言語の分類～プログラム言語は，大きく分けて 2 つに分類できる．
低水準言語（低級言語）～コンピュータが理解できる 0/1 の 2 進数であらわされる機械語やそれを記号化したアセンブラ言語．
高水準言語（高級言語）～低水準言語に比べ，人間に理解しやすい人工的な言語．コンパイラを用いてアセンブラ言語に直し，さらにアセンブラを用いて機械語に変換する．現在用いられる代表的な高水準言語と歴史的な言語を以下に挙げる．

FORTRAN～科学技術計算向け．
COBOL～事務処理向け．

ALGOL～数値計算や論理演算向け．
Smalltalk～オブジェクト指向言語の元祖．
PL/I～科学技術計算および事務処理向け．FORTRAN や COBOL を参考に作成．
BASIC～初心者向け．FORTRAN を参考に作成．
LISP，Prolog，LOGO～人工知能向け．
Pascal，Ada～構造化されたプログラムの作成に適する．
C，C++～OS などシステムの記述に適する汎用言語．Pascal に類似するが，より簡潔な表記が可能で，かつ機能が強力．UNIX で標準．
awk～テキスト処理用．スクリプト言語の元祖の一つ．
Java，JavaScript，Python など～最近よく用いられる言語．
Ruby～日本産のスクリプト言語の一つ．最近用いられる機会が多い．Perl と同じような機能をもつが，言語体系が単純で使いやすい．
Perl～スクリプト言語の一つ．高機能で，さまざまな言語の寄せ集め的な言語体系をもつ．CGI などでよく用いられている．

UNIX 文化
　プログラムは小さく作り，組み合わせて用いることが多い．

b．Perl について
Perl とは
　Perl は，1986 年 Larry Wall によって UNIX 用に開発された手続き型のインタプリタ言語．行単位の文字列の処理，ファイル処理，プロセス処理などに向く．Practical Extraction and Report Language（実用データ取得レポート作成言語）または Pathologically Eclectic Rubbish Lister（病的折衷主義のがらくた出力機）の頭文字をとって Perl と呼ばれ，“パール”と読む．現在の最新バージョンは Perl5.8．フリーソフトとして配布されており，UNIX およびそのクローン，Windows（ActivePerl），Machintosh（MacPerl）などさまざまな環境で利用可能．UNIX の文化を色濃く残すスクリプト言語．

Perl の特徴
長所
1．コンパイルは必要ない．インタプリタ型言語のため．
2．メモリの許す限りファイルサイズ，再帰呼出し，ハッシュテーブルなどの制限がない．手軽に書ける．
3．勉強した分だけ実際に利用できる．
4．対開発効率のコストパフォーマンスに優れる．例えば，C 言語などの他の言語で 10 行程度の処理が Perl では 1 行で書けるものもある．
5．強力な正規表現を利用できる．
6．さまざまな言語の長所をあわせ持つ．
7．さまざまな環境で利用可能．
8．UNIX で用いられる C や sed などに類似した書式であるため，UNIX に慣れている場合は違和感が少ない．

短所
1．コンパイル言語よりは処理速度は劣る．
2．パズル的なプログラムでは，処理内容がわかりにくくなる．

c．Perl を使ったプログラムの作成
Perl の基礎
文法・書式
　Perl スクリプトを構成し，実行指示を行うコマンドの基本単位は文．文には単純文と複合文がある．文は，関数や式，コマンドが組み合わされて構成される．文の末尾には；（セミコロン）を置かなければならない．単純文とは，代入や関数の評価など副作用を得るための式のこと．複合文とは，文のならびを{ }で囲み，ひとまとまりにしたブロックから構成される．if 文や while 文などの制御構造で用いられることが多い．ブロックの最後の文に限り;を省略できる．慣例で

{の後と，}の前は改行することが多い．

変数
スカラー～Perl の基本的なデータ型の一つ．
スカラー変数～$で始まる．$識別子で変数となる．識別子は英文字または下線（_）で始まり，英文字，数字，下線の組み合わせで構成される．

$A　　スカラー変数（値が一つしか代入されない変数）
例．足し算
```
1:      $A = 1; #スカラー変数 A に 1 を代入．
2:      $B = 2; #スカラー変数 B に 2 を代入．
3:      print $A + $B;   #A と B の和を標準出力へ表示する．
```

演算子
算術演算子

記号	細分類	機能
+	加法演算子	加算
-	加法演算子	減算
*	乗法演算子	乗算
/	乗法演算子	除算

代入演算子 =
　左辺値に式の値を代入する．

代表的な関数（命令）
print　出力
例．入力したファイルをそのまま出力
```
1:      while (<>) {     #行がなくなるまで行の読み込みを繰り返す．
2:              print;   #読み込んだ行を標準出力へ表示する．
3:      }
```

while　繰り返し制御
例．3 までカウントする．
```
1:      $a = 1; #スカラー変数 a に 1 を代入する．
2:      while ($a < 4) { #変数$a が 4 より小さい間括弧内を繰り返す．
3:              print $a, “\n”;          #変数$a の値と改行コードを標準出力へ出力する．
4:              $a = $a + 1;
5:      }
```

<>　　ファイルから行を読み込む
例．print の例を参照．

reverse 逆順に並び替える
例．逆さに読む
```
1:      $_ = <>; #標準入力から 1 行入力し，変数$_に代入する．
2:      $a = reverse $_; #逆順にしたものを，変数$a 代入する．
3:      print $a;        #標準出力に表示する．
```

length　スカラー（文字列）の長さを調べ，代入する
例．文字列の長さを調べる
```
1:      $_ = <>;         #標準入力から 1 行入力し，代入する．
2:      $a = length $_;  #長さを変数に代入する．
3:      print “length = “, $a; #標準出力に表示する．
```

s///g　置換演算子
例．配列内のすべての a を A に置き換える．

```
1:      $_ = <>;          #標準入力から 1 行入力し，代入する．
2:      s/a/A/g; #含まれるすべての a を A に置き換える．
3:      print;  #標準出力に表示する．
```

正規表現～文字列のパターンを表現する方法．通常の文字および特殊な意味をもつ文字（メタ文字）で構成される．

メタ文字

文字	意味
.	任意の 1 文字（改行文字は除く）
*	パターンの 0 回以上の繰り返し
?	パターンが 0 回または 1 回現れる
\N	右隣のメタ文字を意味するエスケープ文字
[ABC]	括弧内に含まれる文字のいずれか（文字クラス）（A か B か C のいずれかと一致する）

Perl の利用

Perl を利用する場合，Perl がインストールされている必要がある．Unix 環境では多くの場合，標準で利用できることが多い．Windows では ActivePerl などをインストールする必要がある．Windows ではコマンドプロンプト上で動作する．このため，MS-DOS の知識が多少必要となる．また，スクリプトはエディタで作成するが，エディタプログラムか MS Word などの文章入力のためのプログラムも必要となる．できあがったスクリプトは，コマンドラインから実行する．

準備

まずはじめに，Perl をインストールする．正式にインストールした場合，パスを通しておくとよい．インストール終了後，コマンドプロンプトを起動させ，以下のコマンドを入力し，

```
> perl -e "print 'Hello, world\n'"
```

コマンドラインから Hello, world!と返ってくればインストールされている．または，

```
> perl -v
```

でバージョンが返ってくればインストールされている．

Perl を使ったプログラム実行の流れ

1. フローチャートの作成
2. プログラムの作成
3. デバッグ
4. プログラムの完成
5. 実行

フローチャート（流れ図）の作成

作成するプログラムの大まかな流れを図の形であらわしたもの．プログラム言語によって処理が異なり，そのままの状態ではプログラムに変換できない場合もあるが，プログラムを作る際参考になるので，コーディング前に一度作るとよい．

プログラムの作成

エディタを用いて，スクリプトを作成する．MS Word などでも代用できる．ただし，テキスト形式で保存し，保存したファイルの拡張子を*.txt から*.pl に変更しておいてもよい．また，入力の際半角文字で入力すること．

デバッグ

プログラムが正常に作動するように，誤りを修正していく作業．まず，簡単なデータファイルを作り，それを用いて正常に動作するかテストする．各行の後に，print 文を入れて変数の値を表示すると，理解が容易になる．

プログラム（スクリプト）の実行
ソース
　↓
インタプリタまたはコンパイラ
　↓
翻　訳
　↓
実　行

Perl スクリプトの実行
スクリプト名が *chotto.pl*, データファイル名が *tatoeba.txt* の場合．perl.exe にパスが通っている状態で，以下のコマンドを実行する．
> perl *chotto.pl* < *tatoeba.txt*
結果をファイル *deta.txt* に出力したい場合は，
> perl *chotto.pl* < *tatoeba.txt* > *deta.txt*

Perl スクリプトの終了
実行中の Perl スクリプトは，^C（コントロール + C）[1]で終了できる．

スクリプトの例
UNIX の場合は各スクリプトのはじめに，
```
1:      #!/usr/bin/perl
```
を入力すること．Windows の場合は不要．

スクリプト 1．入力ファイルをそのまま出力するスクリプト
```
1:      while(<>){
2:              print $_;
3:      }
```

スクリプト 2．入力ファイルの文字を置き換えるスクリプト
```
1:      while(<>){
2:              s/[nN\?]/N/g;
3:              print $_;
4:      }
```

スクリプト 3．入力ファイルから ATGCN 以外の文字を除去するスクリプト
```
1:      #!/usr/bin/perl
2:      while(<>){
3:              s/[nN\?]/N/g;
4:              s/[^aAtTgGcCN]//g;
5:              print $_;
6:      }
```

練習
1. 入力した塩基配列から ATGCN 以外の文字を除去し，その長さを出力するスクリプトを作成せよ．
2. 塩基配列を読み込み，その相補鎖を 5’ から 3’ の向きで出力するスクリプトを作成せよ．入力ファイルから ATGCN 以外の文字を除去する機能も付加すること．

参考文献

[1] コントロールキー（CTRL，Control などと表記）を押さえたまま，C のキーを押さえる．途中で離さないこと．

1. 伊藤和人．1997．入門 Perl -Perl for beginner’s-．271 pp．秀和システム，東京．(ISBN 4-87966-751-X)
2. Schwartz, R. L.（近藤嘉雪訳）1995．初めての Perl．xxxii + 305 pp．ソフトバンク，東京．(ISBN 4-89052-678-1)
3. Wall, L., T. Christiansen & J. Orwant．2000．Programming Perl, 3rd Ed．O’Reilly & Associates, Sebastopol．(ISBN 0-596-00027-8)
4. 藤田郁・三島俊司．1999．CGI & Perl ポケットリファレンス．463 pp．技術評論社．(ISBN 4-7741-0755-7)

課題.
以下の内容について結果をまとめてメールで送ること．ただし，内容は添付ファイルとせず，メールの本文に表示すること．ターミナルエミュレータ（端末プログラムなど）のコピー・ペースト機能や，ファイルへのリダイレクトを用いると便利．また，メールが必ずしも届くとは限らないので，メールの内容を印刷したものも理 A504 生物系事務室に提出すること．

1. 個人のホームページを作製し，そのアドレスを表記せよ．(例. http://home.hiroshima-u.ac.jp/chubo/) カウンタも動作させること．また，課題で解説したキーワードの内容をホームページに加えること．
2. コマンドを使って現在（課題開始時間）の日付と時間を表示させよ．
3. 現在動いているプログラムを調べよ．(ヒント．プロセスを表示させる.)
4. 授業中に出てきたコマンドの一覧を作成せよ．
5. 授業中に用いたコマンドから 5 つ選び，その機能と具体的な用例，自分の環境で使用した結果を示せ．
6. 自分のホームページを開き，現在のアクセスカウンタの値を調べよ．
7. 実習の課題で入力したスクリプトの実行結果を示せ．
8. 生物学に関係のある事柄について，計算をさせるスクリプト（プログラム）を作成し，スクリプトとその実行結果をリダイレクトでファイルに出力させよ．(例．ある塩基を入れた場合，それに相補的な塩基を表示する．例えば，A と入力した場合，T と表示する.) プログラムで用いる命令などについては，実習でふれなかったものを用いてもよいこととする．
9. もう一度現在（課題終了時間）の日付と時間を表示させよ．
10. 課題で回答した内容を担当者に送ること．ただし，メールの Subject 題目は学生番号，氏名，「情報活用演習」とし，本文のはじめに学生番号と氏名を明記すること．

出席状況，聴講態度，課題の内容，ホームページの動作確認，メールの送信などを加味して評価を行う．

提出先

坪田博美

1．メールは chubo@hiroshima-u.ac.jp まで送信すること．メールについては，必ず Subject と本文のはじめに学生番号および名前を明記すること（不明な場合は，採点できないこともある）．

2．さらにメールで送った内容を印刷したものを理 A504 生物系事務室に提出．演習終了後 2 週間，17:00 締切．期限および時間厳守．左上をホッチキスでとめて，名前と学生番号を明記すること．