

## エディタとワープロソフト

- **エディタ (editor):**  
純粋にテキストファイルを作る
  
- **ワープロ (word processor):**  
レイアウト・文字飾り等の情報を含んだ  
データを作る

→ 目的によって使い分けよ

## エディタとワープロソフト

ワープロソフトの固有の形式は  
レイアウト・文字飾り等の情報を含む

その分、ファイルサイズも大きく、  
(基本的に) そのソフトウェアでしか読めない

そのような情報が必要ならワープロを用いて  
ワープロソフト固有の形式で作れ

そうでなければエディタを用いて  
汎用のプレインテキストで作れ

## ワープロソフト

ここでは“**MS-Word**”を使う

始めからこの上で文章を打ち込んでもいいが、

この実習では専ら、

エディタで予め作った文書(文字データ)に、

**MS-Word** でレイアウト・文字飾り等を  
施していくことにしよう

→ “**内容と書式との分離**”

## MS-Word の使い方

- **起動**: [スタート] → [プログラム]  
→ [Applications] → [Word2007]
- **終了**: [Office ボタン] → [Word の終了]  
または [ウィンドウ右上の×]
- **新規作成**: [Ctrl+n]  
または [Office ボタン] → [新規作成]
- **開く**: [Ctrl+o]  
または [Office ボタン] → [開く]  
(複数のファイルを同時に開いて編集できる)

## MS-Word の使い方

- **上書き保存: [Ctrl+s]**  
または [Office ボタン] → [上書き保存]
- **名前を付けて保存: [F12]**  
または  
[Office ボタン] → [名前を付けて保存]
- 変更前の版は残して、  
変更後を別ファイル名で保存したい時:  
開いて即「新ファイル名で保存」が賢明

## 文書の書式

文書の内容 (データ) を入れる枠組

(想定すべき) どんな内容でも

対応できることが重要

「内容」を入力して「表示」を出力する

一種のプログラムと言っても良い

→ 動作確認が必要

書類作成を生業 (の一部) にする人にとっての

基本的な素養 (リテラシー)

## 文書の書式

前回は、予め用意したテキスト形式の文書

absent.txt

に様々な書式を設定してみた

- 日付 → 右揃え (右詰め)
- 宛名 → 左揃え (左詰め)
- 差出人 → 右揃え (右詰め)
- 表題 → 中央揃え・文字を大きく (太く)
- 本文段落 → 先頭行のみ字下げ
- 記 → 中央揃え
- 記載内容 → 番号付き箇条書
- 結語 → 右揃え (右詰め)

ところで、もっと踏み込んで考えると、

「教育実習 …… について (お願い)」に  
中央揃え・大きい文字を指定したのは、

本当に我々がやりたかったことなのか？

本当は

「教育実習 …… について (お願い)」は  
表題なんだから、表題らしく表示してくれ

ということなのではないか？



ところで、もっと踏み込んで考えると、

「教育実習 …… について (お願い)」に  
中央揃え・大きい文字を指定したのは、

本当に我々がやりたかったことなのか？

本当は

「教育実習 …… について (お願い)」は  
表題なんだから、表題らしく表示してくれ

ということなのではないか？

## 論理指定と表示形式

- 「**内容**と**書式**との分離」を  
もう一段突き詰めると、  
「**書式**」を更に分析して、  
「**論理指定**と**表示形式**との分離」に進む

- 「教育実習 …… について (お願い)」に  
「表題」という意味を与える (論理指定)
- 「表題」は中央揃え・大きい文字・…  
で表示する、と定める (表示形式)

論理指定による方法 …… スタイルの指定

## 論理指定と表示形式

- 「**内容と書式との分離**」を  
もう一段突き詰めると、
- 「書式」を更に分析して、  
「**論理指定と表示形式との分離**」に進む
- 「教育実習 … について (お願い)」に  
「表題」という意味を与える (**論理指定**)
  - 「表題」は中央揃え・大きい文字・…  
で表示する、と定める (**表示形式**)

論理指定による方法 … スタイルの指定

## 論理指定と表示形式

- 「**内容と書式との分離**」を  
もう一段突き詰めると、
- 「書式」を更に分析して、  
「**論理指定と表示形式との分離**」に進む
- 「教育実習 … について (お願い)」に  
「表題」という意味を与える (**論理指定**)
  - 「表題」は中央揃え・大きい文字・…  
で表示する、と定める (**表示形式**)

論理指定による方法 … **スタイル**の指定

## スタイル

**スタイル:** 論理指定に対する表示方法の指定

- 出来合いのスタイルを使う
- 自分でスタイルを作る

ここでは、出来合いのスタイルを  
(必要ならすこし修正して) 使う実習を行なう

- 内容に対して直接行なうのは論理指定
- 表示形式はスタイルで与える

## スタイル

**スタイル:** 論理指定に対する表示方法の指定

- 出来合いのスタイルを使う
- 自分でスタイルを作る

ここでは、出来合いのスタイルを  
(必要ならすこし修正して) 使う実習を行なう

- 内容に対して直接行なうのは論理指定
- 表示形式はスタイルで与える

## 実習 (今日の提出課題)

本授業のウェブサイト

`http://pweb.cc.sophia.ac.jp`  
`/tsunogai/kougi/09/jolite.html`

に用意したサンプルテキストデータ  
`stylesample.txt`

を使おう

[右クリック] → [Save Link As...]

→ 0: ドライブの授業用フォルダに保存

→ 指示に従ってスタイル指定を行なう

## 実習 (Word 文書として保存)

- MS-Word を起動
- [Ctrl+o] または [Office ボタン] → [開く]  
→ 保存した stylesample.txt を開く  
(見付からないときは  
[すべてのファイル (\*.\*)] で)
- 即、[名前を付けて保存] → [Word 文書]  
→ ファイル名: A0nxxyyy-1123(.docx)  
(自分の学生番号-今日の日付)
- 次からは編集したら適度に保存  
([上書き保存] または [Ctrl+s])



## 実習 (例: スタイル “表題” の指定)

「スタイル指定のサンプル」に「表題」を指定

- 範囲指定 → [スタイル] → [表題] を選択

表示形式を変更したいときは、

「表題」の表示形式で変更

- [表題] の所で [右クリック] → [変更]

適切なスタイルが見付からない時、

[スタイル] の右端の (一覽) → [オプション]  
→ [表示するスタイル]: すべてのスタイル

## 実習 (例: スタイル “表題” の指定)

「スタイル指定のサンプル」に「表題」を指定

- 範囲指定 → [スタイル] → [表題] を選択

表示形式を変更したいときは、

「表題」の表示形式で変更

- [表題] の所で [右クリック] → [変更]

適切なスタイルが見付からない時、

[スタイル] の右端の (一覧) → [オプション]  
→ [表示するスタイル]: すべてのスタイル

## スタイルを用いる利点

文書内容 (データ) → 論理指定 → 表示形式

- 同じ論理指定の部分の表示形式を一斉に (統一的に) 変更できる
- 見た目は同じような表示の箇所でも、論理的な意味付けの違いを区別できる

## 箇条書

段落を範囲指定 →

- 番号なし → **[箇条書]**
  - ★ 最初の項目
  - ★ 次の項目
- 番号付き → **[段落番号]**
  1. 最初の項目
  2. 次の項目
- **[段落番号]** を設定すると、  
項目が増えても番号を自動管理してくれる

## テーマ

各論理指定に対する表示形式 (スタイル) は、  
文書中で独立ではなく、関連性がある

例えば、

- 色・フォント選択に一貫性があるべき
- 強制度合は  
[表題]>[見出し 1]>[見出し 2] であるべき

テーマ: 一連のスタイル設定の組  
→ 表示形式の選択の統一感のあるモデル

## テーマ

各論理指定に対する表示形式 (スタイル) は、  
文書中で独立ではなく、関連性がある

例えば、

- 色・フォント選択に一貫性があるべき
- 強制度合は  
[表題]>[見出し 1]>[見出し 2] であるべき

**テーマ**: 一連のスタイル設定の組  
→ 表示形式の選択の統一感のあるモデル

## 今日の課題の提出法

### 電子メールで提出

- 件名: 1123
  - 本文 1 行目に学生番号・氏名を書く
  - 自分の学生番号・氏名に変更
  - 作成した MS-Word 文書は、ファイル名  
A0nxxyyy-1123.docx  
(自分の学生番号-今日の日付) で保存  
(半角英数字で!!)
- 添付ファイルで提出

## 書式的设计について

書類 (データ) を提出する側としては、  
指定の書式で作成する能力があれば良いが、

書類を提出させる (データを集める) 側には、

「どんな書式で集めるか  
(= どんな書式を選択するか)」

を適切に決める能力が必要である



## 書式的设计について

どんな書式を選択するか:

- 作成コストが低い
- 再利用能力が高い(集計など)

← 集めたデータ(提出書類)を  
どのように使うつもりか

← なぜデータを集めるのか  
(書類を提出させるのか)

目的を考えた運用が必要

## 書式的设计について

どんな書式を選択するか:

- 作成コストが低い
- 再利用能力が高い(集計など)

← 集めたデータ(提出書類)を  
どのように使うつもりか

← なぜデータを集めるのか  
(書類を提出させるのか)

目的を考えた運用が必要

## 書式的设计について

どんな書式を選択するか:

- 作成コストが低い
- 再利用能力が高い(集計など)

← 集めたデータ(提出書類)を  
どのように使うつもりか

← なぜデータを集めるのか  
(書類を提出させるのか)

目的を考えた運用が必要

## 書式的设计について

どんな書式を選択するか:

- 作成コストが低い
- 再利用能力が高い(集計など)

← 集めたデータ(提出書類)を  
どのように使うつもりか

← なぜデータを集めるのか  
(書類を提出させるのか)

目的を考えた運用が必要

## 書式の設計について

現状の書類の書式は、  
ワープロなどで電子的に作成しても、  
紙(手書き)しかなかった頃の書式の  
模倣でしかない場合が多い

電子データが利用できる状況でも、  
従来の形式が最善であるのか

こういうことを考えて改善していくことこそが

“情報リテラシ”

## 書式の設計について

現状の書類の書式は、  
ワープロなどで電子的に作成しても、  
紙(手書き)しかなかった頃の書式の  
模倣でしかない場合が多い

電子データが利用できる状況でも、  
従来の形式が最善であるのか

こういうことを考えて改善していくことこそが

“情報リテラシ”

## 書式の設計について

現状の書類の書式は、  
ワープロなどで電子的に作成しても、  
紙(手書き)しかなかった頃の書式の  
模倣でしかない場合が多い

電子データが利用できる状況でも、  
従来の形式が最善であるのか

こういうことを考えて改善していくことこそが

“情報リテラシ”

## テキスト形式で保存

[名前を付けて保存] → [その他の形式]  
→ [ファイルの種類]：書式なし (\*.txt)

- 中のテキストデータだけが必要な時
- 他のソフトウェアで再利用したい時

などに便利

但し勿論、

**MS-Word 固有の書式情報は失われる**



他にも様々な機能はあるが、きりがないので、

これでワープロ (MS-Word) は一段落

次は

## 表ソフト (作表・表計算)

を取り扱う → “MS-Excel” を使う

## MS-Excel の使い方

- **起動**: [スタート] → [プログラム]  
→ [Applications] → [Excel2007]
- **終了**: [Office ボタン] → [Excel の終了]  
または [ウィンドウ右上の×]
- **新規作成**: [Ctrl+n]  
または [Office ボタン] → [新規作成]
- **開く**: [Ctrl+o]  
または [Office ボタン] → [開く]  
(複数のファイルを同時に開いて編集できる)

## MS-Excel の使い方

- **上書き保存: [Ctrl+s]**  
または **[Office ボタン] → [上書き保存]**
- **名前を付けて保存: [F12]**  
または  
**[Office ボタン] → [名前を付けて保存]**
- 変更前の版は残して、  
変更後を別ファイル名で保存したい時:  
開いて即「新ファイル名で保存」が賢明

## ユーザインタフェースについて

Word も Excel も

基本的な操作法が同じように作ってある

“ユーザインタフェース (User Interface)”

の統一

→ アプリケーションソフトの設計で重要

コンピュータソフトウェアに限らず

“仕事の流れ” の設計では常に重要なこと

## ユーザインタフェースについて

Word も Excel も

基本的な操作法が同じように作ってある

“ユーザインタフェース (User Interface)”

の統一

→ アプリケーションソフトの設計で重要

コンピュータソフトウェアに限らず

“仕事の流れ” の設計では常に重要なこと

## 実習 (既存の定型テキストデータを読み込む)

以前作った kencho.txt を  
MS-Excel で読んでみよう

MS-Excel の起動 → [開く]  
→ 0: ドライブ・授業用フォルダ  
→ ファイル名指定  
または [すべてのファイル (\*.\*)]

## 実習 (既存の定型テキストデータを読み込む)

以前作った `kencho.txt` は、

- 1 行 1 対象 (都道府県)
- 各対象の項目区切りは :

であった

→ [区切り文字] に : を指定

このように、

定型データは簡単に取り込むことが出来る  
表ソフトでのデータ形式の基本は

- 1 行 1 対象 (item)
- 1 セル (桁目) 1 項目

## 実習 (既存の定型テキストデータを読み込む)

以前作った `kencho.txt` は、

- 1 行 1 対象 (都道府県)
- 各対象の項目区切りは :

であった

→ [区切り文字] に : を指定

このように、

定型データは簡単に取り込むことが出来る

表ソフトでのデータ形式の基本は

- 1 行 1 対象 (item)
- 1 セル (桁目) 1 項目



## 実習 (既存の定型テキストデータを読み込む)

以前作った `kencho.txt` は、

- 1 行 1 対象 (都道府県)
- 各対象の項目区切りは :

であった

→ [区切り文字] に : を指定

このように、

定型データは簡単に取り込むことが出来る  
表ソフトでのデータ形式の基本は

- 1 行 1 対象 (item)
- 1 セル (桁目) 1 項目

後は、適当に

- 外枠 (罫線)
- 文字飾り (フォント・大きさ)
- 背景色

等を指定すれば、見栄えの良い表の出来上がり

→ [セルの書式指定]

またはリボンメニューの [フォント][配置] 等

- 表全体に指定
- 行に指定・列に指定
- セルに指定
- 文字に指定

(論理的に正しく指定せよ)

## CSV 形式について

先程のデータは： で項目を区切っていたが、  
実は、標準的には ， で区切るのが一般的

### “Comma Separated Values (CSV)”

- この形式 (拡張子) のファイルは、  
Excel が自動認識する (拡張子の対応付け)
- テキストファイルでもあるので、  
エディタでも編集可能

## CSV 形式について

先程のデータは： で項目を区切っていたが、  
実は、標準的には ， で区切るのが一般的

### “Comma Separated Values (CSV)”

- この形式 (拡張子) のファイルは、  
Excel が自動認識する (拡張子の対応付け)
- テキストファイルでもあるので、  
エディタでも編集可能

## CSV 形式について

先程のデータは： で項目を区切っていたが、  
実は、標準的には ， で区切るのが一般的

### “Comma Separated Values (CSV)”

- この形式 (拡張子) のファイルは、  
Excel が自動認識する (拡張子の対応付け)
- テキストファイルでもあるので、  
エディタでも編集可能

## 実習 (CSV データの作成と読込)

- **EmEditor** で、kencho.txt の  
: を , に置換  
→ ファイル名 kencho.csv で保存
- kencho.csv を
  - ★ [ダブルクリック] で開く
  - ★ **Excel** の中から [開く] で読込むと、拡張子の対応付けにより、  
Excel に表の形で自動的に読込まれる