アンケートからのピックアップ内容

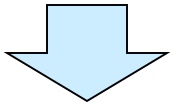
名古屋大学 情報基盤センター
情報基盤ネットワーク研究部門
嶋田 創

アンケートで出てきた内容

- プログラミング
  - 時間があれば、プログラミングの基礎を勉強したい
  - 頭の体操のようなゲーム感覚でできるものをやってみたい
- マクロについて
- その他: 小物いっぱい

小物(1/4)

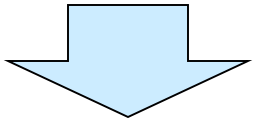
- スマホでPDFファイルがうまくダウンロードできないことが多い
  - この講義のページのPDFはうまくいく
  - 別の講義のページではエラーが発生することがある



- おそらく、ウェブサーバ側で何か制限されているのでは?
  - IPアドレス(学内かどうか)やブラウザ等を識別している?
    - 「403 Forbidden」はサーバ側で禁止している時のエラー
  - あるいは、HTMLのcontent-typeがうまく設定されていないか
- 後者ならば、長押しで出るメニューから「リンク先を保存」を利用すれば問題は緩和される可能性あり

小物(2/4)

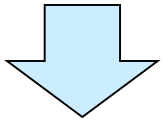
- 自分のパソコンやスマホに入っているデータを学校で印刷するには? あるいは、データを移動させるには?
- ファイル共有ソフト（Dropbox等）について



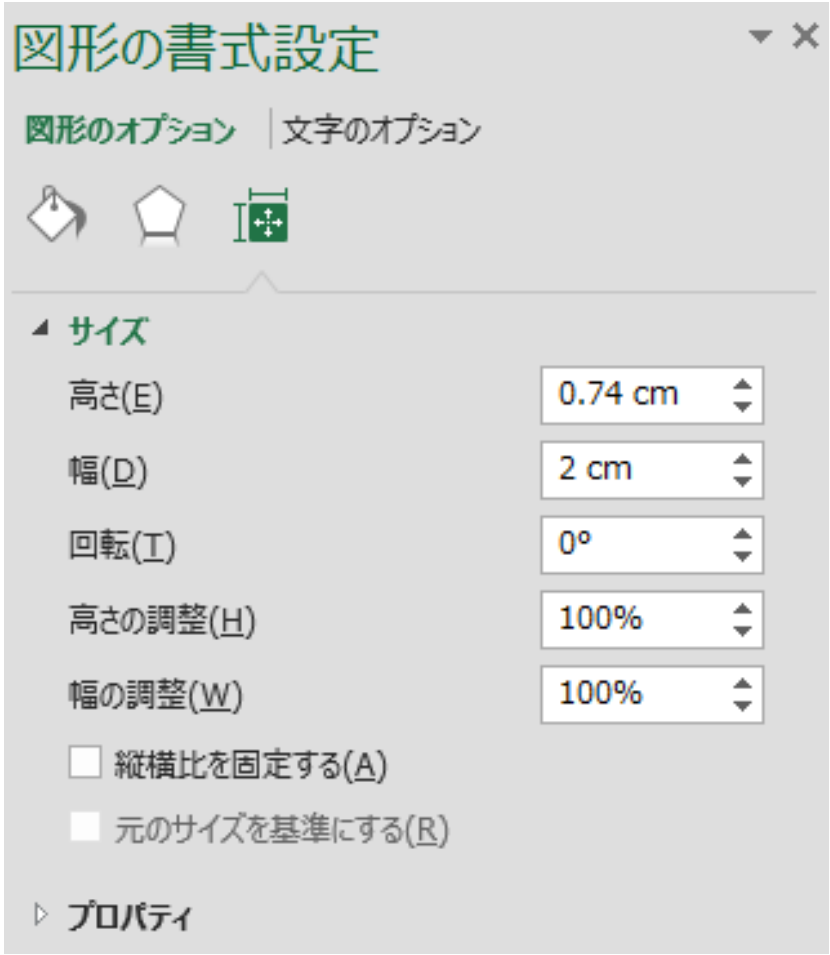
- 案1: USBメモリを利用
  - 最近では、microUSBなどのスマホ用の口と両方あるUSBメモリも
- 案2: クラウドストレージを利用
  - 例: Dropbox, Google Drive, Microsoft OneDrive
  - ただし、常に違法データが無いか検閲されているものと考えて使う
    - 違法データが存在すると判断された場合(実際に無くても)は利用停止措置
  - 名大でもNUSS(Nagoya University Storage Service)があるが教職員のみ　→生徒にも利用できるように要望を上げてみては?

小物(3/4)

- 自分のPCではOffice 2016なので一部が違うのでその説明はある?

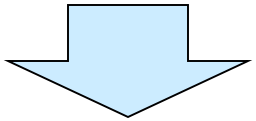


- 自分も2013までしか使っていないので何とも言えない(メディアセンターは2010)
- ただ、右のようなメニューは使いにくくていらいらしている
  - クリックの有効範囲が分かりにくい
  - マウスの移動量が多い(右端に固定されているので図形から遠い)



小物(4/4)

- おすすめのネットショッピングサイトを教えてください



- 趣味によっては、海外サイトを利用するのもあり
  - 例: amazon.com, eBay
  - クレジットカード必須なので、クレジットカードが作れないならばVisaデビットカードなどを作る必要があり
  - 日本以外に発送不可ならば、発送中継サービスを利用するのもあり
  - ちゃんと関税や消費税を払いましょう(自分で調べる)
  - ただし、日本の有名サイトよりも地雷が多いので、事前にいろいろと下調べをして使いましょう
- 逆に、おすすめできないショッピングサイトは色々ありすぎて…

その他(1/2)

- Officeの詳しい使い方等、基本を全部教えて

→無理、時間がいくらあっても足りない

- “MacBook”と“Windows”の違い

→OS: Opearationg System(全てのアプリケーションの実行基盤となる母体)自体が違う

c.f. 他のメジャーなOSとしてはLinuxがある

- Windows10自動更新問題についてどう思いますか?

→Microsoft社のやり口はモラル無さ過ぎ。個人的には偽アンチウィルスソフトウェアを売りつける会社と同レベル。

- この授業教室と図書館以外に、利用可能な印刷機は?

→学部が計算機システムを持っていればそちらにあるかも。PDF化してコンビニのコピー機がおすすめ。

その他(2/2)

- ハッキングのやり方とされた時の対処法
- パソコンのセキュリティについて
- パソコンに感染するウイルスを無効化する方法

→最初のあたりを参考に。完全な対策は無い。重要データはこまめにバックアップ。最悪、PCをリカバリディスクで初期化。

- 自分のウェブサイトの作り方
- レポートの書き方（特に図の入れ方、データからの表の作り方）を学びたいです
- 名大の授業のレポートに使えるwordやexcelの使い方
- パソコンを使う上での豆知識のようなもの

→このあたりは話せたと思う

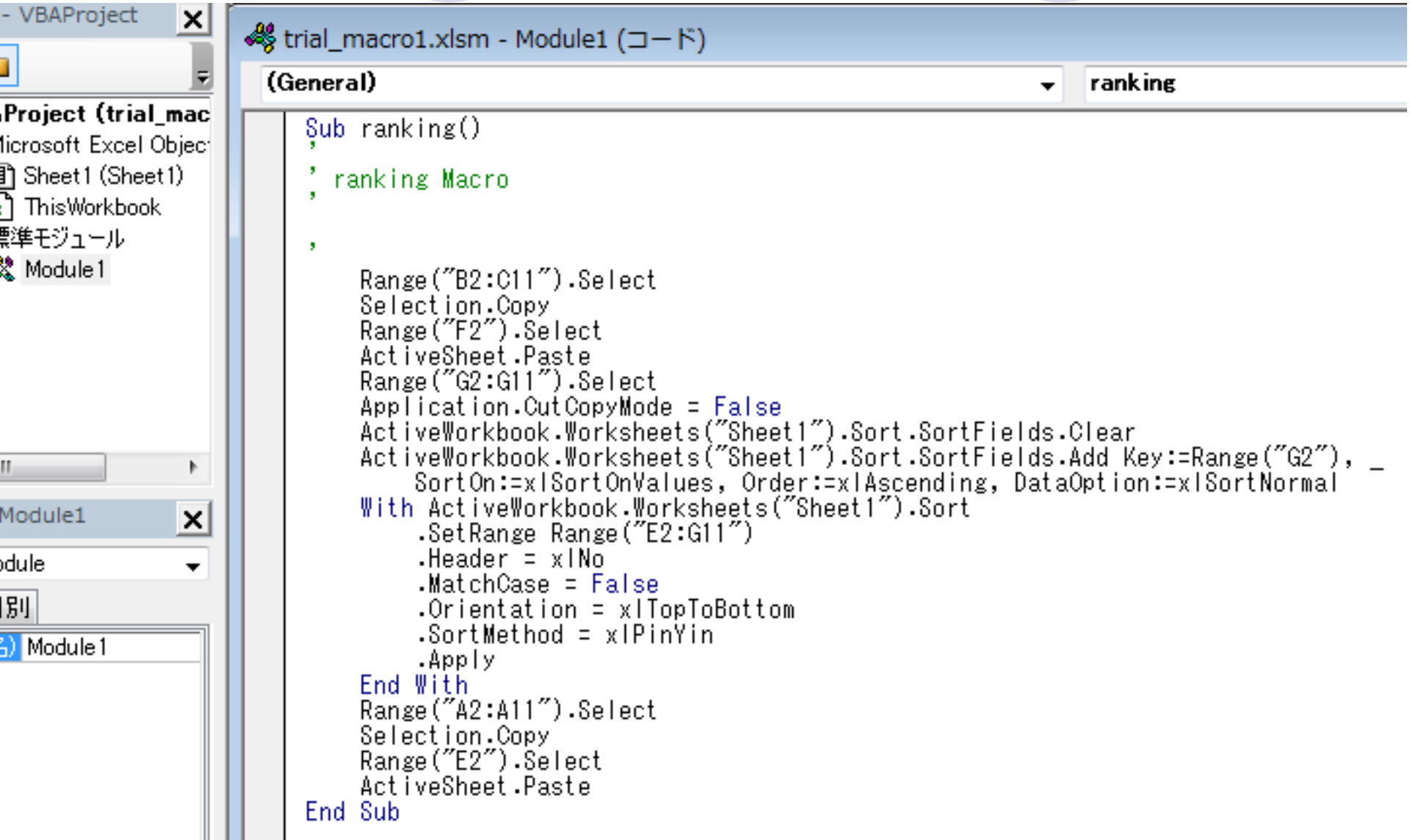
ウィルス(マルウェア)対策

- 基本を抑えておくのが大事
  - アンチウィルスソフトウェアを入れておく
  - セキュリティアップデートはちゃんとかける
  - 怪しいメールの添付ファイルは開かない
- 現在では、いの一番に狙われるとどうしようもない
- やられた後の対策を考えましょう
  - その機器から利用していたネットのサービスのパスワードの変更
  - クレジットカード等の履歴のチェック
  - PCのクリーンインストール
    - リカバリディスクは作成しておく
    - データは事前にバックアップを取っておく

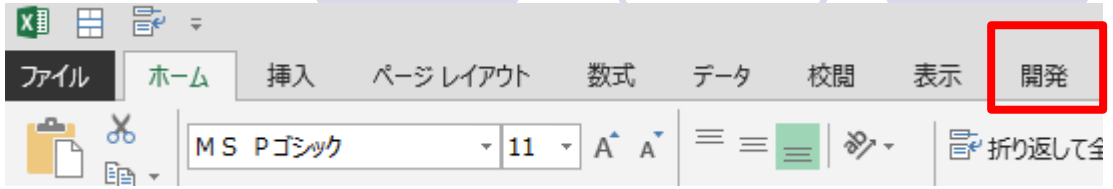
Excelマクロとは

- プログラミング言語(VBA)を用いてExcelへの独自の処理の追加を実現するもの
  - VBA: Visual Basic for Applications
- 特に何が嬉しい?
  - 複雑な順次処理の実装
    - 例: ボタンを押したら値が入っている範囲だけをコピーしてソートして表示し、表示された範囲だけ罫線を付けて印刷書式とする
  - リアルタイムな処理の適用
    - 例: 値が書き込まれたらソートをやり直す
  - いろいろなPCで動くプログラムの実装
    - 今のPCは
    - パズルゲームとかはよく実装されています

マクロの例


- VBAProject
Project (trial_mac
Microsoft Excel Objec
Sheet1 (Sheet1)
ThisWorkbook
標準モジュール
Module1
Module1
odule
別
Module1
trial_macro1.xlsm - Module1 (コード)
(General)
ranking
Sub ranking()
'
' ranking Macro
'
'
Range("B2:C11").Select
Selection.Copy
Range("F2").Select
ActiveSheet.Paste
Range("G2:G11").Select
Application.CutCopyMode = False
ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Clear
ActiveWorkbook.Worksheets("Sheet1").Sort.SortFields.Add Key:=Range("G2"), _
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
With ActiveWorkbook.Worksheets("Sheet1").Sort
.SetRange Range("E2:G11")
.Header = xlNo
.MatchCase = False
.Orientation = xlTopToBottom
.SortMethod = xlPinYin
.Apply
End With
Range("A2:A11").Select
Selection.Copy
Range("E2").Select
ActiveSheet.Paste
End Sub

Excelマクロを作る下準備

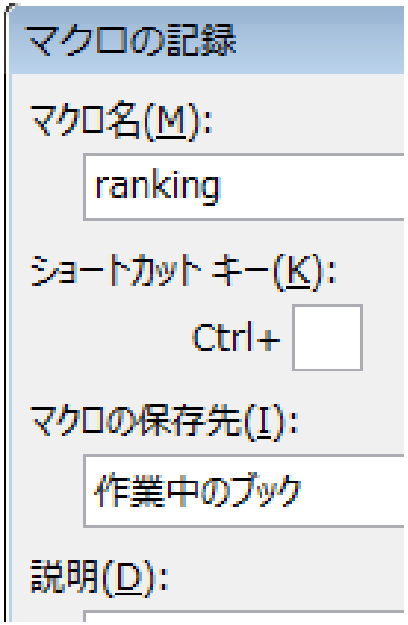


- 「開発」タブを出す
  - リボンUI→ファイル→オプション→リボンのユーザー設定→「開発」にチェック

- ファイルの形式を「マクロ有効ブック(拡張子xlsm)」に変更
  - 新しいExcelでは
- セキュリティ警告時にマクロの実行を許可

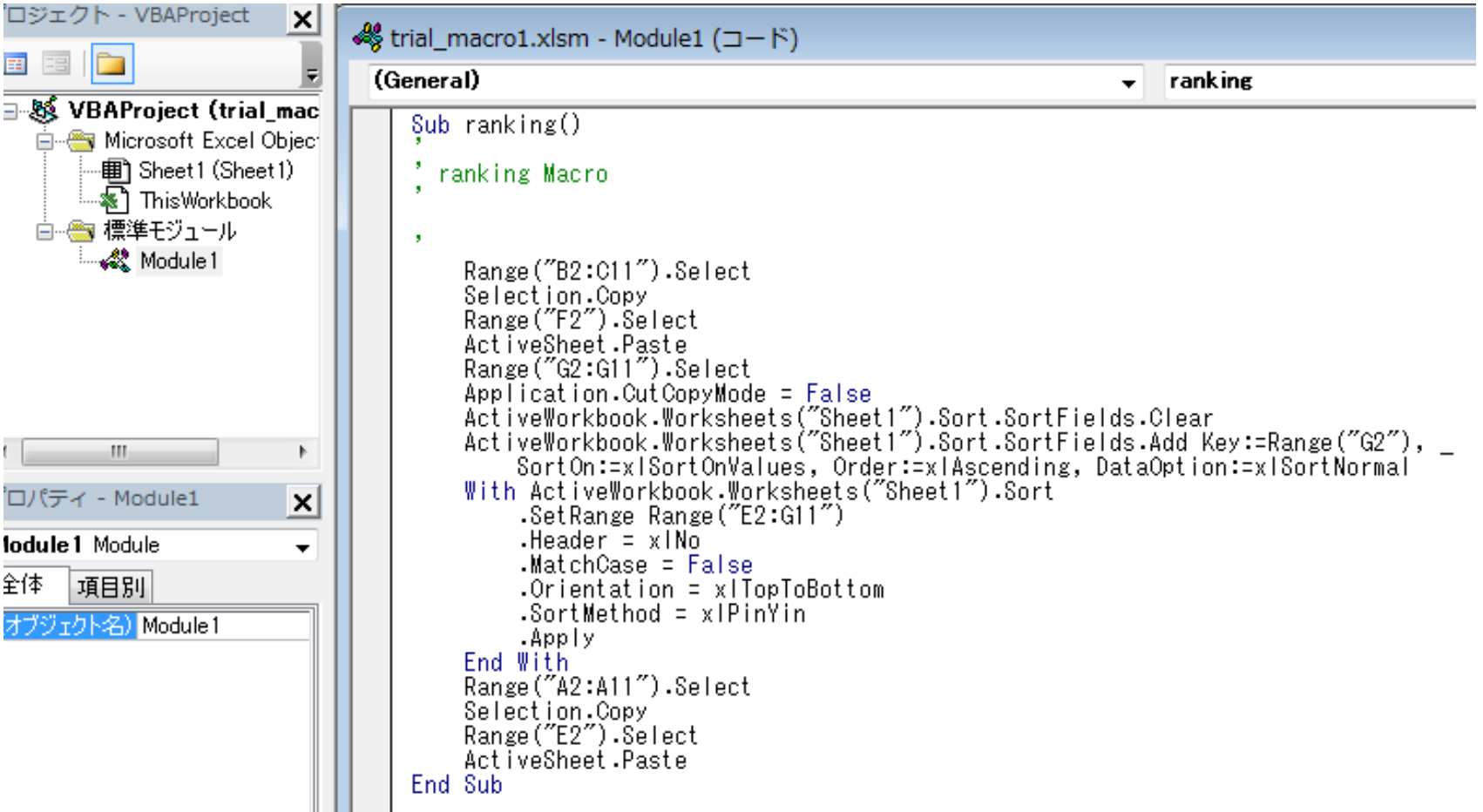
お手軽マクロ作成

- リボンUI→開発→マクロの記録
  - 適当に名前をつける
  - 必要ならばショートカットキーを定義する
    - 「Ctrl+何か」だとかぶることが多いので、「Ctrl+Shift+何か」とか「Ctrl+Alt+何か」とするのもあり
- マクロとしたい動作を実行する
  - 例: データをコピーして昇順にソート
- リボンUI→開発→記録終了



作成されたマクロの確認

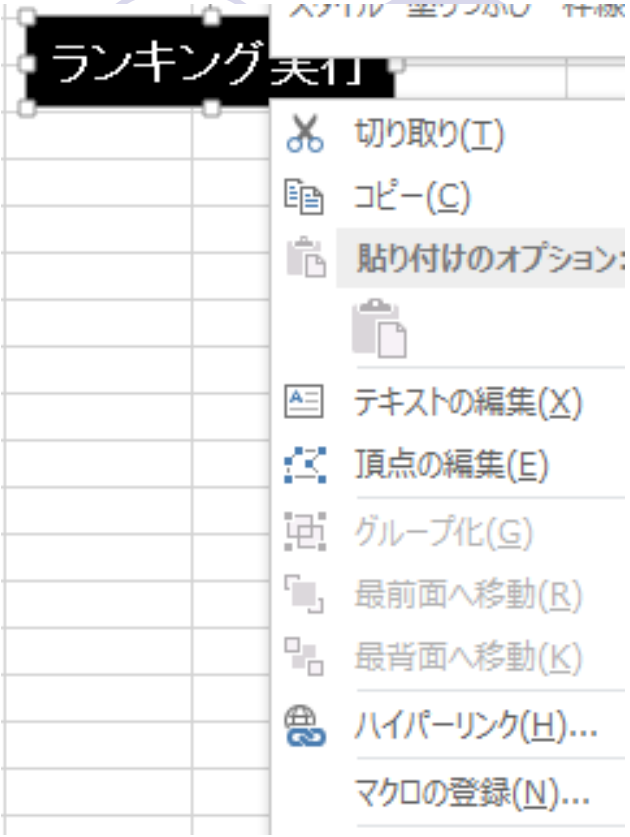
- 開発→マクロ→マクロ名選択→編集
  - VBAのウィンドウが新しく開きます



マクロの利用

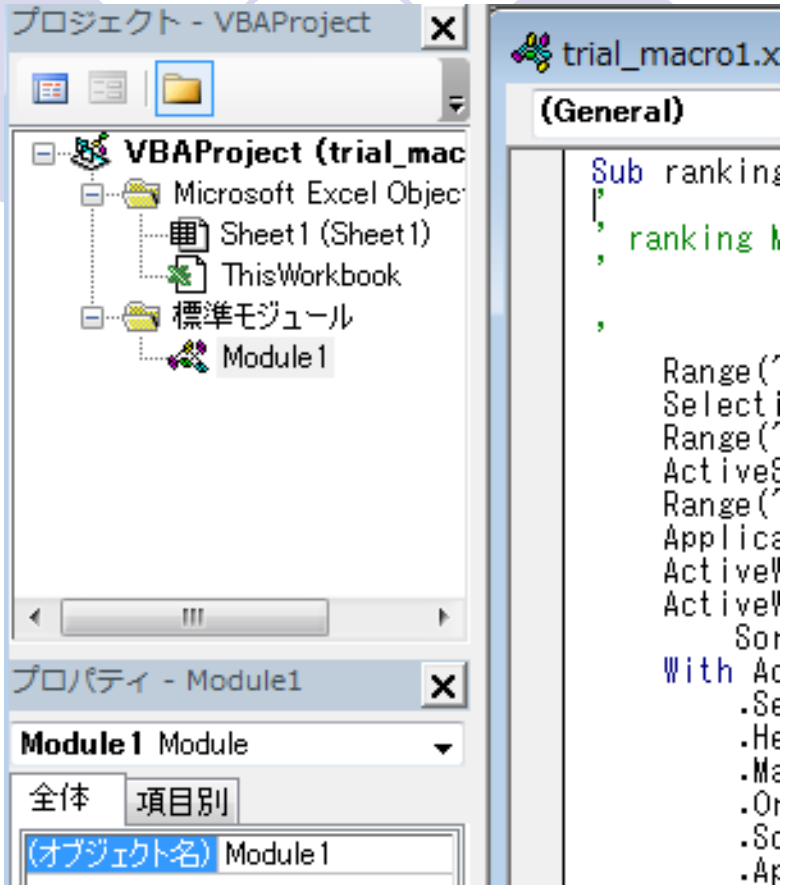
- 開発→マクロ→マクロ名選択→実行
- オブジェクトにマクロを登録
  - Excelシート上にオブジェクト(例: 四角ボックス)を設定
  - オブジェクトを選択して右クリックメニューを出してマクロの登録を選択
  - 登録するマクロを選択してOK

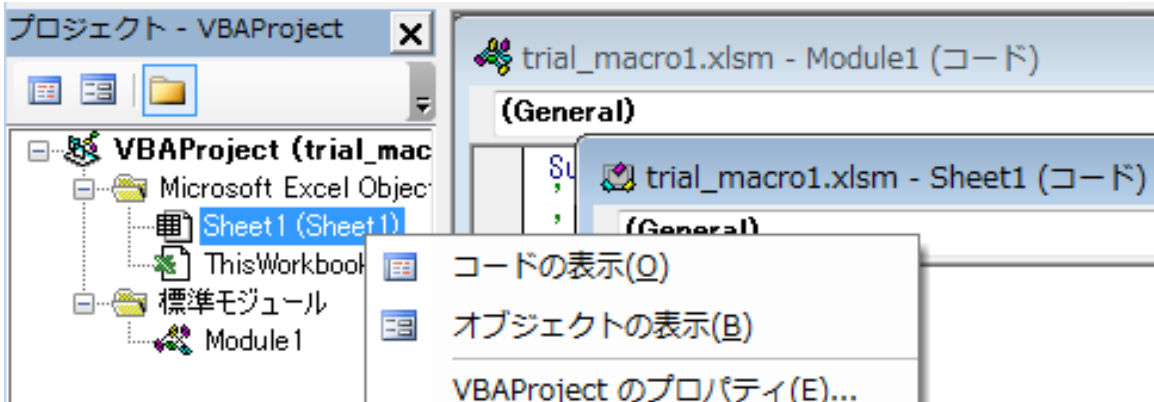
  →そのオブジェクトをクリックするとマクロが実行される
- シート自体にマクロを設定(次スライドより)



シートにマクロを設定

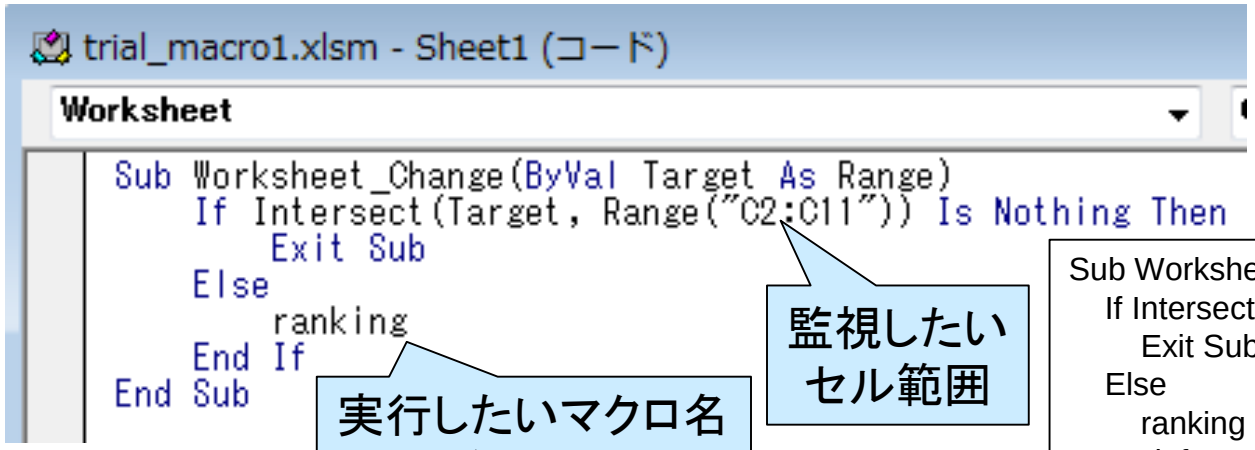
- VBAウィンドウ中のプロジェクトを見ると、Sheet1とかThisWorkBookとかの対象もある
  - Sheet1: その名前のシートにマクロを設定
  - ThisWorkBook: ブックにマクロを設定
- 右クリックメニュー→コードの表示でマクロを設定可能
  - 初期状態では空白





シートの一部セルが変更されたらマクロを実行

- シート自体に以下のマクロを設定




```
Sub Worksheet_Change(ByVal Target As Range)
    If Intersect(Target, Range("C2:C11")) Is Nothing Then
        Exit Sub
    Else
        ranking
    End If
End Sub
```

  - Worksheet_Changeという特別なモジュールを定義
  - 監視したいセル範囲と実行したいマクロ名(モジュール名)は適時設定

→監視したいセル範囲が変更されるとrankingが実行される

- VBAウィンドウで上書き保存を忘れずに

マクロで条件/入出力を設定できるものの例

- 入力
  - キー入力、マウス入力
  - セルの値
- 出力
  - キー出力
  - オブジェクトの位置、大きさ、文字の設定
  - セルの値、文字の設定、色、罫線、
  - ポップアップウィンドウ

他にもいろいろと使えるものはあるはず

プログラミング言語Python

- ここ最近のお勧め軽量プログラミング言語の1つ
- 特徴
  - インデントをベースにプログラム構造を組む
    - インデント(字下げ): 字下げの量が同じ所は同じ
  - 「1つのことを複数の手段で実現できる」ことをあまり良しとしていない
    - 誰が書いても同じような記述になる →他のプログラムを参考にしやすい
    - 学術的にとかプログラムの工夫をする点では面白くないのだが…
  - 軽量言語の中では比較的後発 →改良されている点が多い
- 前述のコマンドプロンプトから実行可能
- 簡単なデータ処理のフィルタなどを書いてさっと適用出来たりすると、実験や研究で便利かも?

Pythonプログラムの例

```
from easygui import *
import random

counter = 0
number = int(random.random() * 100)
message = 'I generated random number from 0 to 99. Estimate it and enter it!'
message_small = 'Too small! Estimate it again and enter it!'
message_large = 'Too large! Estimate it again and enter it!'
window_title = 'Large / small game'
in_key = enterbox(message, window_title)
in_key = int(in_key)

while 1:
  counter = counter + 1
  if number < in_key:
    in_key = enterbox(message_large, window_title)
    in_key = int(in_key)
  elif number > in_key:
    in_key = enterbox(message_small, window_title)
    in_key = int(in_key)
  else:
    final_message = 'True! The number is %d. Total trial is %d.' % (number, counter)
    msgbox (final_message)
    break
```

一般的なプログラムの制御構造(演算/代入)

- 変数に数値や文字列を代入し、それを演算
- 変数名にスペースを使えない点に注意
  - よくアンダースコア(_)がスペースの代わりに利用される
- 例(数値演算):
  - counter = counter + 1
- 例(文字列演算):

  small = “To small! ”

  estimate = “Estimate it again! ”

  message = small + estimate
- 表記に注意したい演算子
  - 乗算: * 、除算: / 、剰余: % 、除算(小数点以下切り捨て) : // 、累乗: ** 、論理和: and 、論理積: or 、論理否定: not

一般的なプログラムの制御構造(条件分岐)

- もし「条件が真」なら「処理1実行」、それ以外なら「処理2実行」
  - 「処理1実行」だけで終わるのもあり
  - 拡張版: もし「条件1が真」なら「処理1実行」、それ以外で「条件2が真」なら「処理2実行」、…、それ以外なら「処理nを実行」
- if ～ elif ～ else構文を使う

例1
```
if number < in_key:
    処理1
次の処理
```

例2
```
if number < in_key:
    処理1
else:
    処理2
次の処理
```

例3
```
if number < in_key:
    処理1
elif number > in_key:
    処理2
else:
    処理3
次の処理
```

- もちろん複数条件指定も可能

一般的なプログラムの制御構造(ループ)

- 「条件が真」ならば処理を繰り返す
- 例: while構文を使って1から10の合計を求めるプログラム

```
i = 1
sum = 0
while i <= 10:
   i = i + 1
   sum = sum + i
print sum
```

  - iはiterator(反復子)の意味で繰り返し回数でよく使われる変数
- for文は他の言語と挙動が異なることに注意
  - 配列(リスト)の各要素を順番に取り出す
  - リストの最後の要素の取り出しとループ内処理が終われればループ終了

一般的なプログラムの制御構造(関数呼び出し)

- 同じような処理を行う場合は関数(サブルーチン、モジュール、などという呼び方もある)を作成してその都度呼び出し
- 再帰呼び出しを使って複雑な処理を簡単に書くことも可能
  - 例: 階乗の計算をn! = n * (n-1)!としてn>=1の条件で再帰呼び出し
- 例

定義
```
def  sin_tailor5 (x)
    sin_x = x – x**3/6 + x**5/120 – x**7/5040 + x**9/362880
    return sin_x
```

呼び出し
```
output = sin_tailor5(a)
```

再帰定義
```
def  sin_tailor (x, n)
   sin_x = (-1)**n * 2**(2*n) / math.factorial(x)
   if n >= 0:
      sin_x = sin_x + sin_tailor (x, n-1)
   return sin_x
```

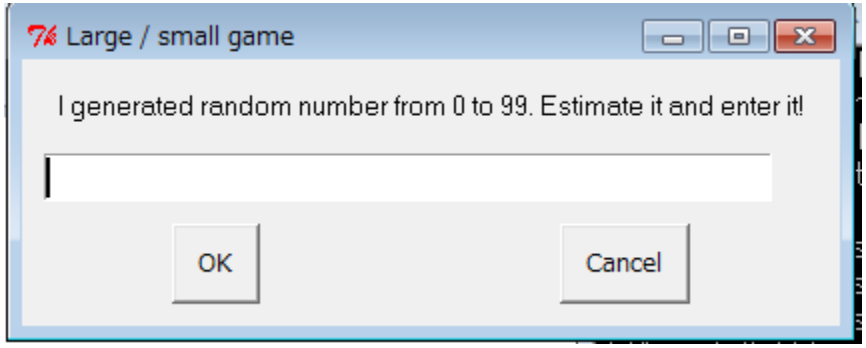
メディアセンターでのPythonプログラムの実行方法

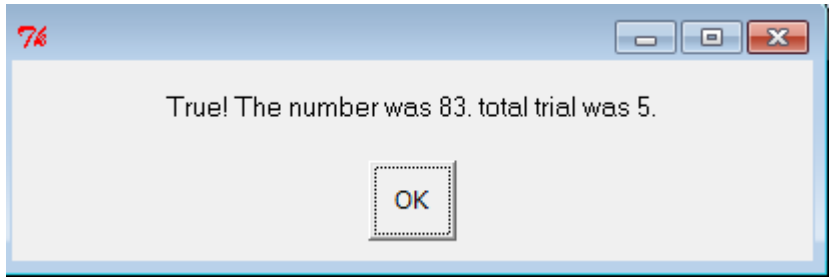
- コマンドプロンプトから実行
  - 左下のWindowsマーク→全てのプログラム→アクセサリ→コマンドプロンプト
  - 「C:¥Python27¥python.exe Pythonプログラムファイル」と打ち込んでリターンを入力
    - Pyとかpyと入力した後でTabキーを押すと自動的に残りが入力される
- 「C:¥Python27¥python.exe」だけだと対話型処理になる
  - プログラムを1行1行入力して反応を見ることができる
  - 簡易電卓みたいに使うこともできる

```
Z:¥Local Settings¥デスクトップ>
Z:¥Local Settings¥デスクトップ>
Z:¥Local Settings¥デスクトップ>
Z:¥Local Settings¥デスクトップ>C:¥Python27¥python.exe memo.py
```

グラフィカルなインタフェースを実現するには?

- GUIを実現するライブラリを利用
  - ライブラリ: 有用な関数を集めたもの
- Python用簡易GUIライブラリEasyGUI
  - メディアセンターのマシンに入っています
  - 「from easygui import *」の文でライブラリ読み込み
  - 日本語は文字コード変換(Shift JISへ)をしないと文字化けします
- EasyGUIを使ったサンプル
  - ランダムに生成された値を当てるゲーム
  - 「Pythonプログラムの例」として出したプログラム





ライブラリは徹底的に活用しましょう

- ある程度プログラミングの基礎ができたら、ライブラリを活用する方向に努力しましょう
  - 今話題のディープラーニングもPythonライブラリがあります
- ライブラリにも、初心者向けライブラリと上級者向けライブラリがあるので、ライブラリ利用もスキルに応じて工夫
  - 上級者向けライブラリはライブラリの処理を呼び出す前の準備がめんどくさい
    - データ生成とか他の関連ライブラリの設定とか
- 使いたいライブラリがあるからそのプログラミング言語に手をだすことも
  - 基礎がきちりできてれば、構文規則や予約語を確認するだけで他のプログラミング言語への以降は容易

軽量プログラミング言語とは?

- スクリプト言語の一種で、特に、プログラムを構成するコードの取り回しに優れるもの
- 特徴
  - 変数のタイプに自由度がある(動的型付け)
    - 普通の言語は事前に「整数」、「小数(浮動小数点型」、「文字列」などを決めないとだめ
    - ただし、必要に応じて変換処理は必要
  - 文字列処理(連結、切り出し、置換、など)が強力
  - 逐次機械語に変換されながら実行するので、処理速度は遅め(インタプリタ型)
  - GUI作成用ライブラリも数多く提供されている
- Perl/PHP/Pythonなど名前がPで始まる物が多いため、P言語と呼ばれることもある

メディアセンターで使える軽量言語

- システムに入っている軽量プログラミング言語
  - Perl(ActivePerl)
  - Python →メディアセンターでは比較的充実している
    - PyMOL
    - python imaging library
    - easygui
  - JavaScript(ブラウザから)
- 開発環境
  - Eclipse
  - Emacs/Meadow(右図)
  - サクラエディタ

Python(+プログラミング)に関するTips(1/2)

- Pythonプログラムを独立した実行ファイル(拡張子.exe)にしたい
  - Py2exe，Pyinstallerを使う
- Pygameライブラリ
  - ビデオゲームを製作するために設計されたPythonのモジュール集
  - http://www.pygame.org/
- print文でコマンドプロンプトに文字を出力可能
  - 例: print ‘Total count:’ , counter
- raw_input文でコマンドプロンプトから文字入力可能
  - 例: input_data = raw_input(‘Input value: ’)
- ファイルへの入出力
  - ファイルを開いて(ファイル変数生成)それを操作(詳細は割愛)

Python (+プログラミング)に関するTips(2/2)

- 覚えておくと便利なデータ構造: 配列(リスト)、ディクショナリ
  - array[10]のような形で配列の要素(=変数)を参照可能
  - 配列は2次元配列、3次元配列なども実現可能
  - ディクショナリはインデクスに文字を使える配列(連想配列、ハッシュとも呼ばれる)
- 比較時に文字列と数字を比較してうまくいかないこともある
  - 変数の内容を変換: int(変数)→整数へ、float(変数)→小数へ、str(変数)→文字列へ
- プログラムを書いてからしばらくたつと自分でも書いた内容を忘れるので、適時コメントを入れましょう
  - 「#」記号後の文字はコメントとして解釈されてプログラムから除外
- プログラミングの世界では数字が0からカウントされることが多い点に注意