Matlab Habits

Vincent Tam

25 Feb, 2014

My Remarks

- Good habits
- Conceptual errors

Good habits (Yevick, 2012, sec. 2.2)

- *Don't* use `i` and/or `j` as loop variables.
  - This practice is common in other programming languages.
  - `i` and `j` are the imaginary units.
  - Use `loop`, `innerLoop`, etc, instead.
- Surround binary operator with spaces.

```
a = 1 ;   % Correct
a=1 ;     % Wrong
```

- *No* spaces after a unary operator.

```
b = -a ;    % Correct
b = - a ;   % Wrong
```

Good habits (cont.) (Yevick, 2012, sec. 2.2)

- Append a ‘C’ and/or an ‘R’ after vectors and/or matrices.
  - If $v \in \mathbb{R}^3$, then use `vR3` or `vC3`.
  - If $m \in M_{4\times 5}(\mathbb{C})$[1], then use `mR4C5`.

[1] The set of matrices with 4 rows, 5 columns and entries in $\mathbb{C}$.

Good habits (cont.) (Yevick, 2012, sec. 2.2)

- Proper indentation for readability

```
myVar = 0;  % An example of improper identation
for loop = 1:100
a = 1; b = 2; c = 3;
for outerLoop = 1:200
    disp(’Hello World!’)
for innerLoop = 1:300
myVar = c * a + b;
end  % Finding syntax error is hard
myVar = myVar / innerLoop
disp(’Which loop am I in?’)
end
myVar
end
```

Good habits (cont.) (Yevick, 2012, sec. 2.2)

- Proper indentation for readability

```
myVar = 0;  % Know the structure by preceding whitespac
for loop = 1:100
    a = 1; b = 2; c = 3;
    for outerLoop = 1:200
        disp('Hello World!')
        for innerLoop = 1:300
            myVar = c * a + b;
        end  % Finding syntax error is easier
        myVar = myVar / innerLoop
        disp('Which loop am I in?')
    end
    myVar
end
```

Good Habits (cont.)

- Use scripts, instead of interactive consoles, for a list of commands.
  - Examples: loops, `if-else` statements, etc
  - For loops and `if-else` statements, *indentation* is important.
- Avoid infinite loops due to logic error.
  - Running infinite loops is *time-wasting*.
  - With debugger, we can spot out errors by running *a few* steps.

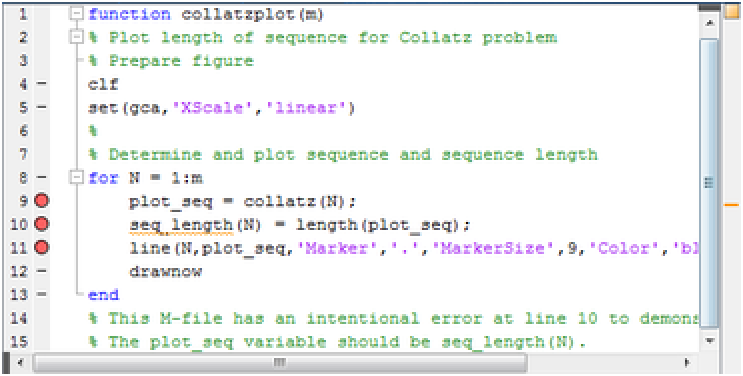
Good Habits (cont.)

- An example of infinite loops copied from Wikipedia. (“Infinite Loop,” n.d.)

```
a = 0;
while a < 10
    sprintf('%d\n', a);
    if a = 5
        sprintf('a equals 5!\n');
    end
    a++;
end
```

Debugging

1. Write some code.
2. Click the *hyphen at the right of a line number* to *set breakpoints*.

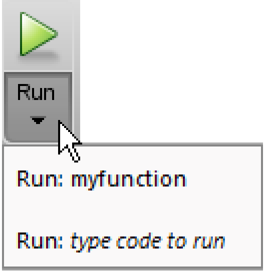


```
1      function collatzplot(m)
2      % Plot length of sequence for Collatz problem
3      % Prepare figure
4  –   clf
5  –   set(gca,'XScale','linear')
6      %
7      % Determine and plot sequence and sequence length
8  –   for N = 1:m
9  ●       plot_seq = collatz(N);
10 ●       seq_length(N) = length(plot_seq);
11 ●       line(N,plot_seq,'Marker','.','MarkerSize',9,'Color','bl
12 –       drawnow
13 –   end
14     % This M-file has an intentional error at line 10 to demons
15     % The plot_seq variable should be seq_length(N).
```

Source: `http://www.mathworks.com/help/releases/`

Debugging (cont.)

3. Run the program. (Click the triangle *on the top*.)



Source: `http://www.mathworks.com/help/releases/R2013b/matlab/matlab_prog/run_command.png`

4. See how the program works.

Conceptual Errors

1. Scripting language v.s. programming language (Hung, 2012, sec. 1)

	Scripting language	Programming language
compilation	yes	no
development speed	faster	slower
execution speed	faster	slower

Conceptual Errors (cont.)

2. `if` is *not* a loop. (“Control Flow,” n.d.)
   - Loop: Repeatedly do something.
   - `if-elseif-else` statements are *conditionals*.
   - Can we do something for 100 times using `if-elseif-else` statements?

References

Control Flow. (n.d.). Retrieved February 23, 2014, from http://en.wikipedia.org/wiki/Loop_(computing)

Hung, C. K. (2012). A Brief Introduction to Scripting. Retrieved February 23, 2014, from http://user.frdm.info/ckhung/b/pr/scripting.php

Infinite Loop. (n.d.). Retrieved February 23, 2014, from http://en.wikipedia.org/wiki/Infinite_loop#Mathematical_errors

Yevick, D. (2012). *A Short Course in Computational Science and Engineering: C++, Java, and Octave Numerical Programming with Free Software Tools*. New York: Cambridge University Press.