Loops

by Ahmet Sacan

looping statements, counted loops, conditional loops, action, iterate, loop or iterator variable, running sum, running product, factorial, preallocate, echo printing, nested loop, outer loop, inner loop, infinite loop, counting, error-checking

Looping Statements

- Loops are used to repeat actions.
- Counted Loops
  - **for**
- Conditional Loops
  - **while**

for loop

```
for loopvar = range
    action
end
```

```
for i = 1:6
  fprintf('hello');
end
```

```
for i = [ 1 2 3 4 5 6 ]
  fprintf('hello');
end
```

```
for i = 1:6; fprintf('hello'); end
```

```
for i = 1:6
  disp(i);
end
```

```
for i = [ 1 2 3; 4 5 6]
  disp(i)
end
```

Exercise: runningsum

- Write a function runningsum.m that takes an integer n, and returns the sum of numbers from 1 to n.
- Change the function runningsum so it takes 2 arguments “lower" and “upper" and returns the sum of numbers “lower" to “upper".
- If only “lower" is provided, return the sum of numbers from 1 to “lower".
  - Important programming concept: specifying default function arguments.
- Change the function runningsum so that the arguments “lower" and “upper" can be specified in any order.

Exercise: myprod

- Write a function myprod(v) that returns the product of the elements in v. Do not use the built-in prod() function.

Combining loops with ifs.

- Exercise: Write a function mymin(v) that returns the minimum value in the vector v. Use a for loop. Do not use min() function.

Input in a for loop

- Write a function inputnumbers(n) that asks the user to enter a number n times, and returns a vector of all the numbers the user enters.
  - Hint: pre-allocate

Nested for loops

```
for loopvarone = rangeone   ←   outer loop

    % actionone includes the inner loop

    for loopvartwo = rangetwo     ←     inner loop
        actiontwo
    end
end
```

- Exercise: write a function printrectangle(R,C) that prints a box of stars, with height R and width C.

```
>> printrectangle(3,4)
****
****
****
```

Exercise

- Write a function printtriangle(R) that prints a triangle of height R, with 1 star in the first row, and R stars in the last row.

```
>> printtriangle(4)
*
**
***
****
```

Exercise

- What will the following code print?

```
for i = 1:3
  for j = 1:2
    fprintf('i=%d, j=%d\n', i, j);
  end
end
```

Exercise: multtable

- Write a function multtable(R,C) that returns a matrix m where m(i,j) is equal to i*j.

Exercise

- Write a function mymatsum(m) that returns the sum of all elements in the matrix m. Use nested for loops.
- Create a random 10000x10000 matrix m in command window. calculate mymatsum(m). How long does it take matlab to calculate this?
  - Programming concept: tic, toc
- Can you re-write your function to run faster?
  - Programming concept: proximal/linear memory indexing
- Can you re-write your function to contain a single for loop?

Exercise: combining nested loops and if

- Write a function mymatsumifpos(m) that returns the sum of positive elements in the matrix m.

while loops

```
while condition
    action
end
```

- while loops are used when you don't know or cannot determine ahead of time how many times the loop will be executed.
- the action should at some point alter the condition to be false. otherwise you get an "infinite loop"
  - while true ; end
  - Use Ctrl+C to break out of an infinite loop. (Ctrl+C can also be used to stop execution of any long-running matlab command).

Exercise

- Write a function [n,f]=factgthigh(high) that returns the first integer n and its factorial that is greater than the input "high".

```
>> [n,f] = factgthigh (5000)
n =
  7
f =
  5040
```

Exercise

- Rewrite the following for loop using a while loop.

```
for i=1:5:100
    fprintf('i=%d\n', i);
end
```

Error checking user input in a while loop

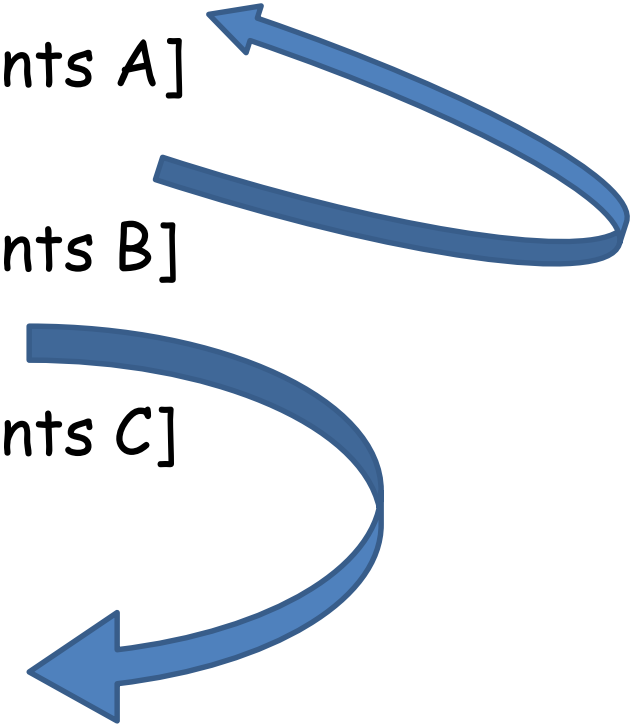
- Exercise: Write a function inputposnumber() that asks the user for a positive number and returns it.

```
>> x = inputposnumber
Enter a positive number: -5
Invalid! Enter a positive number: 5
OK!
x =
  5
```

- Programming Concept: "Error Handling"
  - What to do on error?
  - Do nothing
  - NaN / zero / default value
  - Raise error()

Flow control: continue, break

```
for range / while condition
    [statements A]
    continue
    [statements B]
    break
    [statements C]
end
```



Exercise

```
a='x';
while ~strcmp(a,'y')&&~strcmp(a,'n')
    a=input('Enter (y/n) :','s');
end
```

- Fill-in the body of the while loop so that the code is equivalent to above.

```
while true
    a=input('Enter (y/n) :','s');
    ....
    ....
```

Tips for Speed: "Preallocate" and/or "Avoid loops"

```
a=[];
for i=1:10000
        a(i)=log(i);
end
```

```
a=zeros(1,10000);
for i=1:10000
        a(i)=log(i);
end
```

```
a=log(1:10000);
```

```
tic; a=[]; for i=1:10000; a(i)=log(i); end; toc
tic; a=zeros(1,10000); for i=1:10000; a(i)=log(i); end; toc
tic; a=log(1:10000); toc
```