output

- disp(‘ string’)
- disp(variable)
- Fprintf(‘string %d ‘,variable)

```
%d    integer (it stands for decimal integer)
%f    float (real number)
%c    character (one character)
%s    string of characters
```

- fprintf(‘ \n ‘)
- fprintf(‘number is %.2f’,a)
- fprintf(‘ the int is %10d and %.3f ‘,4,2.5)
- fprintf(‘ the number is \t %d ‘’ \\ ‘,5)
- vec = 2:5;
  fprintf('%d ',vec)
- mat = [4 5 8 ; 10 5 9]
- fprintf(‘ %d %d %d ‘,mat’)

Note :
Use disp to print matrix is more easier.

- error(‘ the number you entered is not valid’)

inputs

- Variable = input(‘value ‘)
- If the variable is a character or string

  Variable = input (‘ char or string’,’s’)

Reading and writing From a MAT-File

- load filename variables.
- save filename variables.
- save –append filename variable.
- who –file filename.

WRITING AND READING SPREADSHEET FILES

- xlswrite('filename',variable)
- [nums, txt] = xlsread('filename')

- **Selection Statements (if , switch , is-function, …).**
- **Looping (for – while).**

RELATIONAL EXPRESSIONS

Operator	Meaning
>	greater than
<	less than
>=	greater than or equals
<=	less than or equals
==	equality
~=	inequality

logical operators

Operator	Meaning
||	or
&&	and
~	not

Selection Statements

- The MATLAB® software has two basic statements that allow us to make choices.
- The if statement and the switch statement. The if statement has optional else and elseif clauses for branching. The if statement uses expressions that are logically true or false. These expressions use relational and logical operators.
- MATLAB also has “is” functions that test whether an attribute is true or not; these can be used with the selection statements.

IF

- **The if statement chooses whether another statement, or group of statements, is executed or not.**
- **The general form of the if statement is:**

```
if condition
action
end
```

```
>> num = –4;
>> if num < 0
num = 0
end
```

practice

PRACTICE 4.1

Write an **if** statement that would print "Hey, you get overtime!" if the value of a variable *hours* is greater than 40. Test the **if** statement for values of *hours* less than, equal to, and greater than 40. Will it be easier to do this in the Command Window or in a script?

if-else statement

- **The if-else statement is used to choose between two statements, or sets of statements. The general form is:**

```
if condition
  action1
else
  action2
end
```

```
if rand < 0.5
disp('It was less than .5!')
else
disp('It was not less than .5!')
end
```

Example

- Write a script to let the user to choice Y or N. If the user choose ‘Y’ display yes if he choose ‘N’ display no.

Note

Don’t discriminate between uppercase and lower case of user input so ‘N’ is ‘n’ and ‘Y’ is ‘y’.

Practice

PRACTICE 4.2

Write a script *printsindegorrad* that will:

- prompt the user for an angle
- prompt the user for (r)adians or (d)egrees, with radians as the default
- if the user enters 'd', the **sind** function will be used to get the sine of the angle in degrees; otherwise, the **sin** function will be used. Which sine function to use will be based solely on whether the user entered a 'd' or not ('d' means degrees, so **sind** is used; otherwise, for any other character the default of radians is assumed, so **sin** is used)
- print the result.

Here are examples of running the script:

```
>> printsindegorrad
Enter the angle: 45
(r)adians (the default) or (d)egrees: d
The sin is 0.71

>> printsindegorrad
Enter the angle: pi
(r)adians (the default) or (d)egrees: r
The sin is 0.00
```

The elseif Clause

```
if condition1
     action1
elseif condition2
     action2
elseif condition3
      action3
% etc: there can be many of these
else
    actionn      % the nth action
end
```

practice

```
function y = calcy(x)
% calcy calculates y as a function of x
% Format of call: calcy(x)
% y = 1          if    x < -1
% y = x^2        if    -1 <= x <= 2
% y = 4          if    x > 2

if x < -1
     y = 1;
elseif x <= 2
     y = x^2;
else
     y = 4;
end
end
```

Practice

- **Write a script to calculate the area of a circle depending on the desired radius from the user.**

Note

The radius can not be negative.

Practice

- The following function receives an integer quiz grade, which should be in the range from 0 to 10. The function then returns a corresponding letter grade, according to the following scheme: a 9 or 10 is an ‘A’, an 8 is a ‘B’, a 7 is a ‘C’, a 6is a ‘D’, and anything below that is an ‘F’. As the possibilities are mutually exclusive, we could implement the grading scheme using separate if statements. However, it is more efficient to have one if-else statement with multiple elseif clauses. Also, the function returns the letter ‘X’ if the quiz grade is not valid. The function assumes that the input is an integer.

THE SWITCH STATEMENT

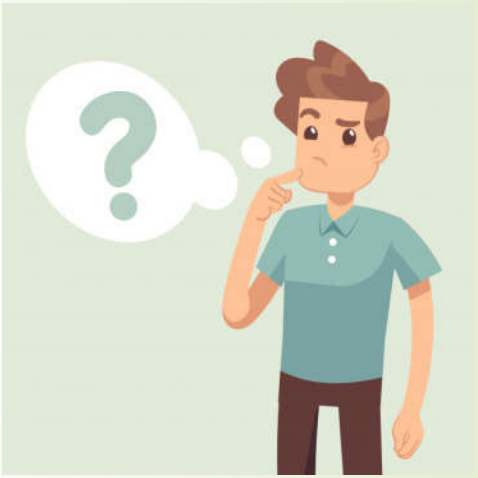
- A switch statement can often be used in place of a nested if-else or an if statement with many elseif clauses. Switch statements are used when an expression is tested to see whether it is equal to one of several possible values.

The general form of the **switch** statement is:

```
switch switch_expression
  case caseexp1
    action1
  case caseexp2
    action2
  case caseexp3
    action3
  % etc: there can be many of these
  otherwise
    actionn
end
```

Practice

- Do the previous practice using switch statement.



Is functions

- isletter(var)
- isempty(var)
- iskeyword(‘var’)
- Explore other is-functions.

For Loop Statements

- **The for statement, or the for loop, is used when it is necessary to repeat statement(s) in a script or function, and when it is known ahead of time how many times the statements will be repeated.**

- **The general form of the for loop is:**

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

THE PROGRAMMING CONCEPT

The loop could be entered in the Command Window, although, like **if** and **switch** statements, loops will make more sense in scripts and functions. In the Command Window, the results would appear after the **for** loop:

```
>> for i = 1:5
       fprintf('%d\n',i)
   end
1
2
3
4
5
```

QUICK QUESTION!

How could you print this column of integers (using the programming method):

```
  0
 50
100
150
200
```

QUICK QUESTION!

What would be the result of the following **for** loop?

```
for i = 4:2:8
    fprintf('I will not chew gum\n')
end
```

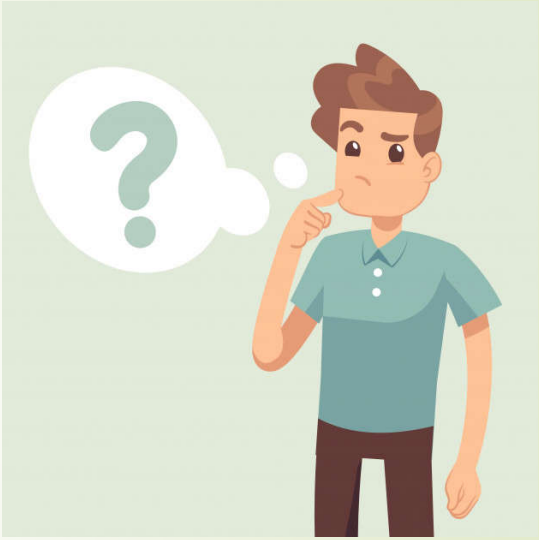
practice

- write a script that will sum the n numbers entered by the user.

n is a random number from 1:5

Practice

- Repeat the previous practice to get the product of the numbers.



practice

- Write a script to calculate how many of the numbers that the user entered were greater than the average?
  n is a random number from 1:5

NESTED FOR LOOPS

- The action of a loop can be any valid statement(s). When the action of a loop is another loop, this is called a nested loop.

The general form of a nested for loop is as follows:

```
for loopvarone = rangeone          ← outer loop
      % actionone includes the inner loop
      for loopvartwo = rangetwo     ← inner loop
            actiontwo
      end
end
```

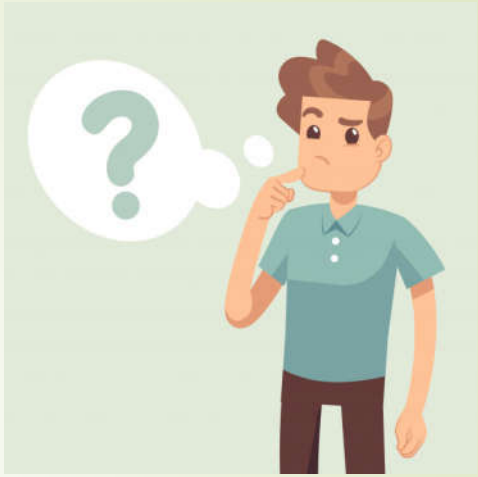
```
% Prints a box of stars
% How many will be specified by two variables
% for the number of rows and columns

rows = 3;
columns = 5;
% loop over the rows
for i=1:rows
    % for every row loop to print *'s and then one \n
    for j=1:columns
        fprintf('*')
    end
    fprintf('\n')
end
```

QUICK QUESTION!

How could this script be modified to print a triangle of stars instead of a box such as the following:

```
*
**
***
```



practice

```
% Displays the loop variables
for i = 1:3
    for j = 1:2
        fprintf('i=%d, j=%d\n',i,j)
    end
    fprintf('\n')
end
```

Practice

- Write a script to fill matrix(rows, columns) which elements are products of or number of row and number of column.

1	2	3	4	5
2	4	6	8	10
3	6	9	12	15

PRACTICE 5.3

For each of the following (they are separate), determine what would be printed. Then, check your answers by trying them in MATLAB.

```
mat = [7 11 3; 3:5];
[r, c] = size(mat);
for i = 1:r
    fprintf('The sum is %d\n', sum(mat(i,:)))
end
----------------------------------------
for i = 1:2
    fprintf('%d: ', i)
    for j = 1:4
        fprintf('%d ', j)
    end
    fprintf('\n')
end
```

WHILE LOOP

- **The while statement is used as the conditional loop in MATLAB; it is used to repeat an action when ahead of time it is not known how many times the action will be repeated.**

- **<u>The general form of the while statement</u> is:**

```
while condition
action
end
```

Practice

- As an example of a conditional loop, we will write a function that will find the first factorial that is greater than a certain value. For an integer n, the factorial of n, written as n!, is defined as n! ¼ 1 * 2 * 3 * 4 * … * n.

Practice

```
% Prompts the user and echo prints the numbers entered
% until the user enters a negative number

inputnum=input('Enter a positive number: ');
while inputnum >= 0
     fprintf('You entered a %d.\n\n',inputnum)
     inputnum = input('Enter a positive number: ');
end
fprintf('OK!\n')
```

Practice

```
% Prompts the user for positive numbers and echo prints as
% long as the user enters positive numbers

% Counts the positive numbers entered by the user
counter=0;
inputnum=input('Enter a positive number: ');
while inputnum >= 0
    fprintf('You entered a %d.\n\n',inputnum)
    counter = counter + 1;
    inputnum = input('Enter a positive number: ');
end
fprintf('Thanks, you entered %d positive numbers.\n',counter)
```

Practice

PRACTICE 5.5

Write a script *avenegnum* that will repeat the process of prompting the user for negative numbers, until the user enters a zero or positive number, as just shown. Instead of echo-printing them, however, the script will print the average (of just the negative numbers). If no negative numbers are entered, the script will print an error message instead of the average. Use the programming method. Examples of executing this script follow:

```
>> avenegnum
Enter a negative number: 5
No negative numbers to average.

>> avenegnum
Enter a positive number: -8
Enter a positive number: -3
Enter a positive number: -4
Enter a positive number: 6
The average was -5.00
```

- break.
- While true.
- return.
- continue.