Basic flow control in MatLab

Relational operators

`if` blocks

`while` loops

Early termination commands

Function files

Relational operators

a) < less than

b) <= less than or equal to

c) == equal to

d) ~= not equal to

e) >= greater than or equal to

f) > greater than

Return 0 if false

Return 1 if true

They work as expected with scalars (1x1 arrays) and return a single value of 0 or 1.

`if` Blocks

```
if test
    block
end
```

If *test* evaluates true (1), execute statements in *block*, then skip to first statement after **`end`**.

Otherwise, skip straight to first statement after **`end`**.

“Do-if-true” rule.

if Block Example

An example of an if block:

```
clear                  % clear out workspace

balance = input(‘Enter your balance amount: ’)

if balance < 50
    disp(‘Warning, you are below 50’)
end
```

`while` Loops

`while` *test*

        block

`end`

Used for looping through statements in *block* an unknown number of times.

“Do-if-true” rule still applies.

`while` Loops Cont.

The *test* is evaluated first.

If *test* is false, the *block* is not evaluated and MatLab skips to first statement after `end`.

If *test* is true, then *block* is executed and MatLab returns to the start of the loop and rechecks `test`.

`while` Loops Cont.

Any necessary variables and values must be available before you enter the `while` loop.

The normal way for the `while` loop to terminate is for something to change inside the loop so that `test` will evaluate as false the next time it is checked.

while Loop Example

An example of a while loop:

```
clear                            % clear workspace
balance = 100;                   % initial balance

while balance > 0

    balance = balance - input(‘Enter amount spent: ’)

    if balance < 50
        disp(‘Warning, you are below 50’)
    end

    disp(‘You still have some money left’)

end
```

`break` Command

You can jump out of a `while` loop part way through with the `break` command.

You skip the rest of the lines in the `block` and immediately jump to the first line after the `end` statement.

break Command Example

```
clear                            % clear workspace
balance = 100;                   % initial balance

while balance > 0

    balance = balance - input(‘Enter amount spent: ’)

    if balance < 50
        disp(‘Warning, you are below 50’)
        break
    end

    disp(‘You still have some money left’)

end
```

`continue` Command

Part way through a `while` loop you can skip the remaining commands in the block, and start a new pass through the loop, if appropriate.

continue Command Example

```
clear                            % clear workspace
balance = 100;                   % initial balance

while balance > 0

    balance = balance - input(‘Enter amount spent: ’)

    if balance < 50
        disp(‘Warning, you are below 50’)
        continue
    end

    disp(‘You still have some money left’)

end
```

Simple Function Files

```
function y = theFunctionName(x)
block of code to do cool stuff;
y = exp(x);
```

The file name must be the same as `theFunctionName` but with the `.m` extension.

`function` File Example

The following code is in a file named **`theFunction.m`**:

```
function y = theFunction(x)
y = sin(x) - x;
```

In the main program, it is called as follows:

```
x = 7;
z = theFunction(x);
```

Or even just

```
theFunction(pi)
```