

Day 11

- (Concept Question)
- Basic `if` concepts
- Relational operators
- Fancy `if`
- More practical `if` example
- User input
- (Exercises)

ME123 Computer Programming

Basic `if` concepts

`if` statements are used to check “if” a condition is true, and then do something special:

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=20;
5 -   if (b > a)
6 -       fprintf('b is bigger than a\n');
7 -   end
```

If b is bigger than a
we print a message

Command Window

b is bigger than a

ME123 Computer Programming

Basic `if` concepts

The simplest form of the `if` statement is just

`if condition is true`

Many different types of conditions can be tested

do this calculation

and this one

You can put as many calculations as you want here

and this one

`end`

This “end” statement ends the “if block” – that’s how Matlab knows which calculations go with the condition

ME123 Computer Programming

Relational operators

Relational operators are used in `if` statements

obvious:

- `<` less than
- `<=` less than or equal to
- `>` greater than
- `>=` greater than or equal to

less obvious:

- `==` is equal to (test for equality)
- `~=` is not equal to (test for inequality)
- `&&` and
- `||` or

ME123 Computer Programming

Relational operators

`==` is the *test* for equality, not `=`

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=20;
5 -   if (b==a)
6 -       fprintf('b is equal to a\n');
7 -   end
```

Using `=` when you should use `==` is a very common mistake.

ME123 Computer Programming

Relational operators

Use parentheses to create compound conditions

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=20;
5 -   c=30;
6 -   if ( (a<b) && (b<c) )
7 -       fprintf('b is between a and c\n');
8 -   end
```

We cannot use
`if (a<b<c)`
we must break it
into two simpler
pieces and use
`&&`

ME123 Computer Programming

Fancy `if`

This form of the `if` command lets us handle two possibilities:

```
if condition is true  
    do this calculation  
    and this one
```

The first part is the same as before.

```
else  
    do this calculation  
    and this one
```

But now we can specify what to do if the condition is false.

```
end
```

ME123 Computer Programming

Fancy `if`

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=20;
5 -   if (b>a)
6 -       fprintf('b is bigger than a\n');
7 -   else
8 -       fprintf('b is less than or equal to a\n');
9 -   end
```

Notice that there's no condition to test after the `else`

ME123 Computer Programming

Fancy if

This form of the `if` command lets us handle many possibilities:

```
if condition is true
    do this calculation
    and this one
elseif condition is true
    do this calculation
    and this one
else
    do this calculation
    and this one
end
```

The first part is the same as before.

This adds a second condition. You can have as many `elseif` sections as you want. `elseif` is ONE word.

This is what Matlab does if none of the other conditions was true.

ME123 Computer Programming

Fancy if

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=20;
5 -   if (b>a)
6 -       fprintf('b is bigger than a\n');
7 -   elseif (b==a)
8 -       fprintf('b is equal to a\n');
9 -   else
10 -       fprintf('b is less than a\n');
11 -   end
```

remember!
==

Still no condition on `else`– but we know `b` is less than `a` because it wasn't bigger or equal to `a`.

ME123 Computer Programming

Fancy if

The **first true section** within an `if` **executes** and **then we leave** the entire block— even if a later section is also true.

This is demonstrated on the next slide.

ME123 Computer Programming

Fancy if

```
Day11_InClassExample.m x
1 -   clc
2 -   clear variables
3 -   a=10;
4 -   b=10;
5 -   if (b>a) False
6 -       fprintf('b is bigger than a\n');
7 -   elseif (b<=a)
8 -       fprintf('b is less than or equal to a\n');
9 -   elseif (b==a)
10 -       fprintf('b is equal to a\n');
11 -   else
12 -       fprintf('b is less than a\n');
13 -   end
```

True—
print
and go to
end →

We never get here ←

```
Command Window
b is less than or equal to a
```

ME123 Computer Programming

More practical `if` example

The examples so far have been a bit silly, because we knew what the values of our variables were.

On the next slide is a script that prints a table of the time and altitude for a vertically launched projectile. It only prints if the projectile is still going up.

Notice that you can put an `if` block inside a `for` loop.

ME123 Computer Programming

More practical `if` example

```
Day11_InClassExample2.m x
1 -   clc
2 -   clear variables
3 -   g=9.81; % m/s^2
4 -   v0=80; % launch velocity m/s
5 -   fprintf('Time (s) Altitude (m) \n');
6 -   for t=0:1:100 %time in seconds
7 -       v=v0-g*t; %vertical velocity
8 -       y=v0*t-0.5*g*t^2; %altitude
9 -       if (v > 0)
10 -           fprintf(' %5.2f %6.2f \n',t,y);
11 -       end
12 -   end
```

Time (s)	Altitude (m)
0.00	0.00
1.00	75.09
2.00	140.38
3.00	195.85
4.00	241.52
5.00	277.38
6.00	303.42
7.00	319.65
8.00	326.08

ME123 Computer Programming

User input

We can add functionality to our scripts by asking for user input. The basic form of the command is

```
user_answer = input ('text to display');
```

The user's answer (generally a number) gets stored in this variable. Any valid variable name will work.

Whatever you type inside the quotes gets printed in the command window. Usually instructions to the user, such as 'Enter rocket velocity'

ME123 Computer Programming

User input

```
Day11_InClassExample3.m x
1 -   clc
2 -   clear variables
3 -   g=9.81; % m/s^2
4 -   v0=input('What is the launch velocity in m/s? ');
5 -   fprintf('Time(s) Altitude(m)\n');
6 -   for t=0:1:100           %time in seconds
7 -       v=v0-g*t;         %vertical velocity
8 -       y=v0*t-0.5*g*t^2; %altitude
9 -       if (v > 0)
10 -           fprintf(' %5.2f   %6.2f \n',t,y);
11 -       end
12 -   end
```

We ask the user what launch velocity to use and save the answer in the variable `v0`. Then we make a table for that `v0`.

ME123 Computer Programming

User input

When we run the program it *waits* for the user to respond:

Text goes to command window.

Nothing happens until user types an answer and hits enter.

Then the program continues, creating a table for that v_0

Time (s)	Altitude (m)
0.00	0.00
1.00	75.09
2.00	140.38
3.00	195.85
4.00	241.52
5.00	277.38
6.00	303.42
7.00	319.65

ME123 Computer Programming

User input

Other input examples

```
age=input('How old are you? ');
```

```
income=input('How much did you make last year?');
```

```
widgets_needed=input('How many widgets do we need?');
```

```
interest=input('What is the interest rate?');
```

ME123 Computer Programming