MATLAB
Flow control: *if-then*

What is flow control?

- **Flow control** refers to how you tell Matlab which lines of code to execute, and how many times.

- Ordinarily, Matlab executes code from top to bottom, line-by-line (skipping over any comments indicated by the %)

Order of evaluation

- Ordinarily, Matlab executes code from top to bottom, line-by-line

```
>> disp('hi')
>> disp('bye')
```

will produce the result:

```
hi
bye
```

If-then statements

- In Matlab, you might have a line or lines of code that you only want executed if a condition is met.

- Example: If temperature is greater than 100, then print a warning to the screen.

```
>> if T>100
    disp('Emergency: boiling!')
end
```

If-then statements

- You may additionally have behavior that you only want carried out if the boolean evaluated to false.

- One way to get this effect is to flip the Boolean:

```
>> if T<=100
        disp('Ok, not boiling');
end
```

If-then-else statements

- Another way to get the same result is to add an *else* clause to the *if-then* statement:

```
>> if T>100
    disp('Emergency: boiling!')
else
    disp('Ok, not boiling');
end
```

If-then-else statements

```
>> if T>100
    disp('Emergency: boiling!')
else
    disp('Ok, not boiling');
end
```

The behavior of this *if-else* clause is different than

```
>> if T>100
        disp('Emergency: boiling!')
end
disp('Ok, not boiling');
```

---

if-then-elseif statements

- The last variant in Matlab is the *elseif* clause, where you can specify lines of code to be evaluated if the first clause is false, but the elseif is true:

```
>> if T>100
    disp('Emergency: boiling!')
elseif T>70
    disp('Warning: getting ...
hot');
else
    disp('Ok, not boiling');
end
```

Review

1. *if, end, else*, and *elseif* are reserved keywords in Matlab. Don't use them for variable or function names!

2. Using Boolean operators and *if-then* statements, we can control which lines of code Matlab will see.