

Making Choices

Learning Objectives

- Learn about logical operators
- Learn about if, elseif and else
- Learn to test equality, AND, and OR conditions
- Learn to nest loops
- Learn about conditional vectors

Part 1: Logical operators

Intro to logical operators available in MATLAB

- Less than, greater than, equal to, less than / greater than, not equal to

What happens when you type:

```
1<2
```

```
ans =  
    1
```

```
1>2
```

```
ans =  
    0
```

```
1==2
```

```
ans =  
    0
```

```
35==35
```

```
ans =  
    1
```

```
35==70
```

```
ans =  
    0
```

- AND and OR (&& and ||)
- True, false

Part 2: If statements

The programs we have written so far always do the same things, regardless of what data they're given.

We want programs to make choices based on the values they are manipulating.

Conditionals.

Introduce if statements

```
num = 37;  
  
if num>100  
    disp('number is greater than 100');  
else  
    disp('number is not greater than 100');  
end
```

number is not greater than 100

```
disp('done')
```

done

Mini Challenge: Change num and see what happens.

AND, OR tests (use whiteboard)

- If (raining && going outside) => bring umbrella.
- If (I'm hungry || I'm bored) => Eat

AND statement (true if both statements are true)

```
% test if I need an umbrella  
  
raining=1; %it's raining!  
going_outside=0; % not going outside  
  
if raining && going_outside  
    disp('better bring an umbrella!')  
else  
    disp('no need for an umbrella!')  
end
```

no need for an umbrella!

```
% test if I should eat  
hungry=0;  
bored=0;  
  
if hungry || bored  
    disp('eat!')  
else  
    disp('dont eat....yet...')  
end
```

dont eat....yet...

OR statement (true if either or both statements are true)

```
% test OR statements

if (1>0)||(3<4)
    disp('at least one part is true')
end
```

at least one part is true

Part 3: Nesting Loops

Combining if statements with loops.

```
numbers = [-5, 3, 2, -1, 9, 6];    % list of numbers to test
total = 0;                        % initialise value to zero

for n = numbers %loops through each element in "numbers" vector
    if n >= 0     % checks if the number is positive
        total = total + n; %adds it to the total of positive numbers
    end % closes the "if" statement
end % closes the "for" loop

disp(['sum of positive values: ', num2str(total)]) %displays the sum
```

sum of positive values: 20

Part 3 – Conditional Vectors

**** maybe skip this section**

One of the great things about MATLAB is we don't always need to nest conditional statements in a loop. MATLAB can condition vectors directly.

Why do we use && and || in conditions?

Because & and | are for vector conditionals

```
vector1 = [1 0 0 1 0 1];
vector2 = [1 0 1 0 0 0];

vector1 & vector2
```

```
ans =
     1     0     0     0     0     0
```

```
vector1 | vector2
```

```
ans =
     1     0     1     1     0     1
```

We can do this vector conditioning with all our logical operators

```
numbers = [-5, 3, 2, -1, 9, 6];  
numbers < 0
```

```
ans =  
    1     0     0     1     0     0
```

```
numbers >= 0
```

```
ans =  
    0     1     1     0     1     1
```

```
numbers(numbers < 0)
```

```
ans =  
    -5    -1
```

```
numbers(numbers >= 0)
```

```
ans =  
     3     2     9     6
```

```
pos_total = sum(numbers(numbers >= 0));  
neg_total = sum(numbers(numbers < 0));  
disp(['sum of positive values: ', num2str(pos_total)]);
```

```
sum of positive values: 20
```

```
disp(['sum of negative values: ', num2str(neg_total)]);
```

```
sum of negative values: -6
```