## Making Choices

## Learning Objectives

* Learn about if, elseif and else
* Learn to test equality, AND, and OR conditions
* Learn to nest loops

**Part 1 – 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

num = 37;

if num>100

disp('number is greater than 100');

else

disp('number is not greater than 100');

end

disp('done')

Try without else block

Other logical operators

>, <, >=, <=, == (test if something is true, not assigning)

Introduce elseif

Code to return sign of number

% check the sign of a number

if num>0

sign = 1;

elseif num==0

sign = 0;

else

sign = -1;

end

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)

EXPLAIN TRUE/FALSE IN MATLAB

True = 1

False = 0

Lower-case true and false are also ‘special’ words in MATLAB

% 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

% test if I should eat

hungry=0;

bored=0;

if hungry || bored

disp('eat!')

else

disp('dont eat....yet...')

end

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

% test if I should eat

hungry=0;

bored=0;

if hungry || bored

disp('eat!')

else

disp('dont eat....yet...')

end

When you use conditionals MATLAB tries to convert everything to either a 0 or a 1 (true or false)

Set hungry = 42

Set hungry = ‘hungry’

The code still works!

CHALLENGE:

% Write code to work out which number is bigger (and try changing the

% values of num1 and num2 to make sure your code always works)

num1 = 40;

num2 = 25;

CHALLENGE EXTENSTION:

Modify the code so that the correct statements are displayed for number 1 and number 2. Check your code with some different values of number 1 and number 2 – does it still work?

**Part 2 - Nesting**

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

CHALLENGE:

Edit the code so it also displays the sum of the negative values

CHALLENGE ANS:

pos\_total = 0;

neg\_total = 0;

for n = numbers

if n >= 0

pos\_total = total + n;

else

neg\_total = neg\_total + n;

end

end

disp(['sum of positive values: ', num2str(pos\_total)]);

disp(['sum of negative values: ', num2str(neg\_total)]);

EXTENSION:

Using the commands we have learnt previously, write a script that loads in some Pokemon data and select one Pokemon vector randomly (choose any Pokemon you like or use randi() command. HINT: remember how to slice a single vector out of a data matrix?). Analyse this Pokemon vector and display its Pokedex number if its before CP is greater than 10 or after CP is greater than 100 or player level is less than 30.

EXTENSION ANS:

clear;

clc;

% Concatenate CP data

Pokemon\_CP\_all = [Pokedex\_Num CP\_before CP\_after Level];

% Generate single random integer

my\_Pokemon\_Ind = randi(12000);

% select the Pokemon to analyse

my\_Pokemon = Pokemon\_CP\_all (my\_Pokemon\_Ind,:);

if (my\_Pokemon(2) > 10)

disp(['This Pokemon’s Pokedex number is: ', num2str(my\_Pokemon(2))]);

elseif (my\_Pokemon(3) > 100)

disp(['This Pokemon’s Pokedex number is: ', num2str(my\_Pokemon(2))]);

elseif (my\_Pokemon(3) < 30)

disp(['This Pokemon’s Pokedex number is: ', num2str(my\_Pokemon(2))]);

else

disp('This Pokemon doesn’t meet given criteria’)