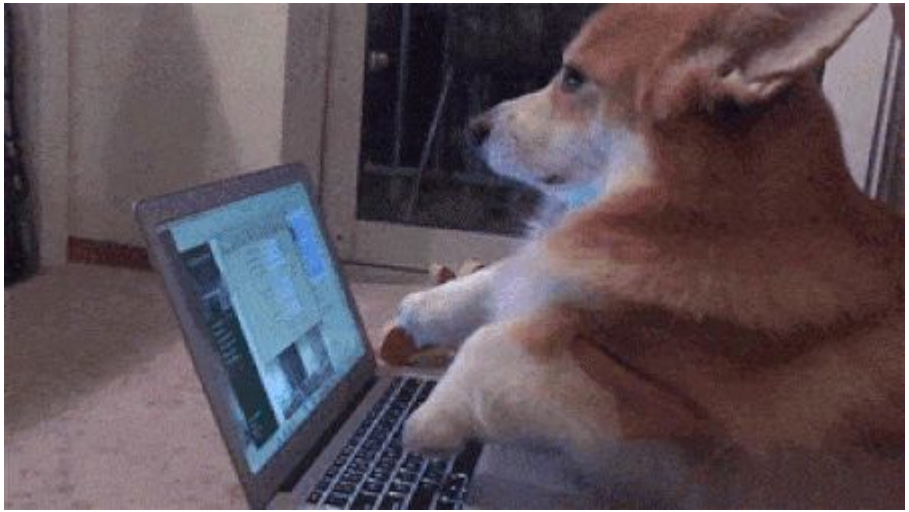
bit.ly/clps950

happy friday!



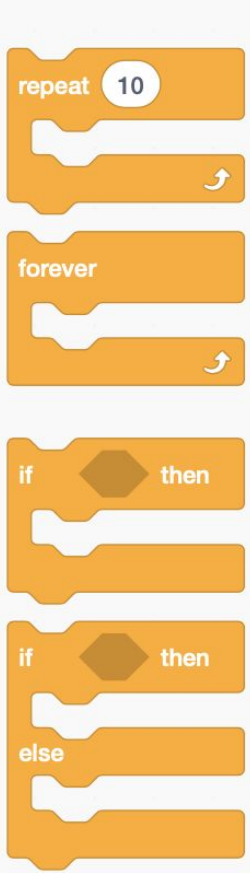
Conditionals

now that we know about conditionals (making things TRUE or FALSE) we can use them to decide whether to run code.

these are called **conditional statements** and in matlab they are if, else, and elseif

until now we have run all code we see – this will allow us to run some code only if certain conditions are true!

you may have used these in your scratch project!



Conditionals: if

if is our first building block. this checks whether the code after it is true or false

let's look at a function to see if it is snowing (if it is below 32 F AND there is precipitation)

*see **end** statements!*

```
function [snow_or_no] = isitsnowing(temperature_farenheight,precipitation)
% written by haley keg 2.4.22
% decide if it it snowing in providence
if (temperature_farenheight < 32) && (precipitation == true)
    snow_or_no = true;
end
if (temperature_farenheight >= 32) && (precipitation == true)
    snow_or_no = false;
end
if (temperature_farenheight < 32) && (precipitation == false)
    snow_or_no = false;
end
if (temperature_farenheight >= 32) && (precipitation == false)
    snow_or_no = false;
end
end
```

Conditionals: else

if statements are powerful, but it would be tedious to write out the opposite of every if statement. so, we can use **else** to tell the computer what to do when our first condition is false!

the **else** goes before the end because it is part of the same logic question

```
function [snow_or_no] = isitsnowing(temperature_farenheight,precipitation)
% written by haley keg 2.4.22
% decide if it it snowing in providence
if (temperature_farenheight < 32) && (precipitation == true)
    snow_or_no = true;
else
    snow_or_no = false;
end
end
```

Conditionals: elseif

we might have more than one thing to check for. now I'm going to make 'snow or no' into a percent chance instead of a boolean, and if it's below 32 degrees but no precipitation, we should make it a 50% chance.

```
function [snow_or_no] = isitsnowing(temperature_farenheight,precipitation)
% written by haley keg 2.4.22
% decide if it it snowing in providence
if (temperature_farenheight < 32) && (precipitation == true)
    snow_or_no = 100;
elseif (temperature_farenheight < 32) && (precipitation ~= true)
    snow_or_no = 50;
else
    snow_or_no = 0;
end
end
```

Conditionals: elseif

elseif is really powerful to let you not type out so many conditional statements. let's say you are building a computer vision algorithm which takes in pictures of animals to decide if they can fly. you know if the animal doesn't have wings, it definitely can't fly. but, some animals with wings can't fly (sorry, penguins). we can use elseif to code this situation!

```
function [can_fly] = flying_animal(has_wings,is_penguin)
% written by haley keg 2.4.22
if ~has_wings
    can_fly = false;
elseif is_penguin
    can_fly = false;
else
    can_fly = true;
end
end
```

Common Bug: Conditionals

remember that once a conditional ends, it doesn't remember what was going on above unless you save the information into a variable. a very common bug (even for experienced programmers!) is to chain two if statements instead of remembering elseif.

what would happen if we passed a **dog** into this function?

```
function [can_fly] = flying_animal(has_wings,is_penguin)
% written by haley keg 2.4.22
if ~has_wings
    can_fly = false;
end
if is_penguin
    can_fly = false;
else
    can_fly = true;
end
end
```

group work

Emily @_emilyliu_ · Feb 2

deleting my unit tests because they're not passing

105 348 4K

Friday Problems: Conditionals

uh oh! you forgot about ostriches which also have wings but can't fly. how would you change the code to include them?

what about flying squirrels, who have no wings but can fly?

```
function [can_fly] = flying_animal(has_wings,is_penguin)
% written by haley keg 2.4.22
if ~has_wings
    can_fly = false;
end
if is_penguin
    can_fly = false;
else
    can_fly = true;
end
end
```

Friday Problems: Conditionals

how would you combine this isitsnowing function with convcelc from class on wednesday if you only know the temperature in celsius?

```
function [F] = convcelc(C)
multiplier = 1.8;
bias = 32;
celc = multiplier * C;
F = celc + bias;
end
```

```
function [snow_or_no] = isitsnowing(temperature_farenheight,precipitation)
% written by haley keg 2.4.22
% decide if it it snowing in providence
if (temperature_farenheight < 32) && (precipitation == true)
    snow_or_no = true;
else
    snow_or_no = false;
end
end
```

Friday Problems: Logic

Make a function for each of the following statements. Create the variables P and Q, and test the function for different values of P,Q. Return a variable with your choice of name!

- ~(P||Q)==~P&&~Q
- ~(P&Q)==~P|~Q
- P&~(Q&&~P)

Friday Problems: Function Writing and Logic

you work in a research lab where participants play a computer game while you record EEG signals. on each trial, you know how fast the participant answered in ms and whether they got the question right or wrong. before you look at the EEG data, you want to separate the types of trials to use – the brain signal you're looking for only happens on **correct** trials. additionally, you know that if the subject took less than 200ms to respond they didn't have time to actually do the task (impossibly fast) and that if they took more than 4000ms, they weren't paying close attention, so you also want to exclude the trials outside these bounds.

write a function which takes in trial information and outputs a boolean variable which tells you if a trial is good to use or not.

Friday Problems: Environment Diagram (Challenge)

set a to some number <20 and pick b = 3 or 4. how would you draw an environment diagram for a function that calls itself? hint: you will need multiple houses!

```
function c = mystery(a, b)
% a function that calls itself!
% an example of recursion... we won't cover in this class
    if b == 1
        c = a;
    else
        c = a + mystery(a, b-1);
    end
end
```