“Get Techy”

Trinity Hall JCR Programming Club
Session 2

Sinéad McAleer - Tech Officer
mcaleesi@tcd.ie

Last Week

- Intro to Programming
- Made our sprite move
- Played audio
- Loops
- Made our sprite talk
- Completed an IF/ELSE statement
- Declared variables

Challenge:

- *Create apple game*
- *Create OWN game for extra credit*

Scratch

Week 2

Boolean

A boolean is a type of variable. (More on this later!)

A boolean is either TRUE or FALSE - we could also refer to this as YES or NO or 1 or 0.

Examples:

A person is awake - YES or NO.

A deposit is paid - TRUE or FALSE.

A person likes crisps - 1 or 0.

Combining this with knowledge from last week, we can use booleans as the condition for our if statements.

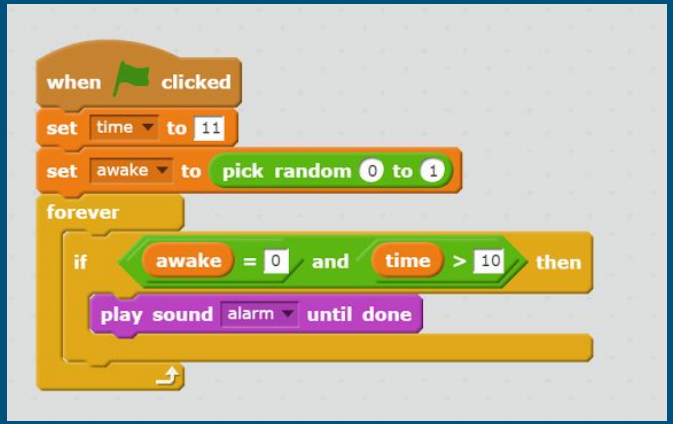
IF a person is asleep and the time is past 10am
then wake them up.

Let's look at this on Scratch.

If awake...

Awake = 0 is the same as AWAKE = FALSE

We can set the variable time to any value we want



https://scratch.mit.edu/projects/180133091/

The random function picks a random value between 0 and 1.

Recap on CONDITIONS

- All of our IF statements have a condition.
- Let's look at all the conditions we have used so far

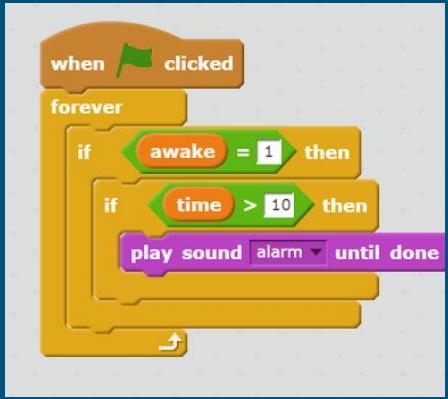
We can use these justifying symbols ">" or "<" or "=" when comparing variables in our IF statement. This is pretty straightforward.

For the two bottom conditions, it is a type of boolean statement. IF touching edge is TRUE then... / IF key space pressed is TRUE then...

This is the same as saying IF awake is TRUE then...

Recap on CONDITIONS

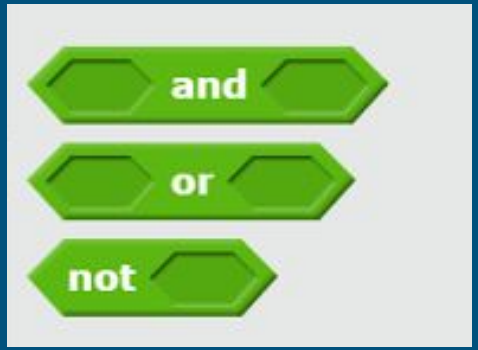
- We have also came across cases were we want to say IF x AND y then…
- We said, IF not awake AND the time is greater than 10 then...



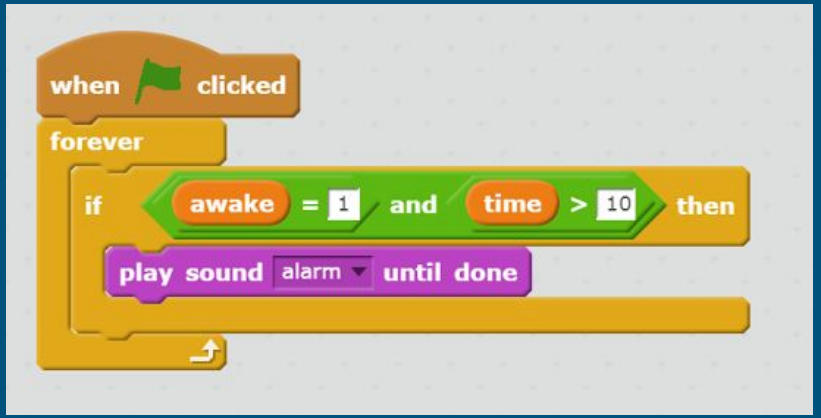
- In Scratch we could simply tuck these two conditions inside each other
- The pink line (play sound) will not run UNLESS both IF statements are satisfied.

- We may also want to so IF x OR y then...
- How would we go about this?

Hint:



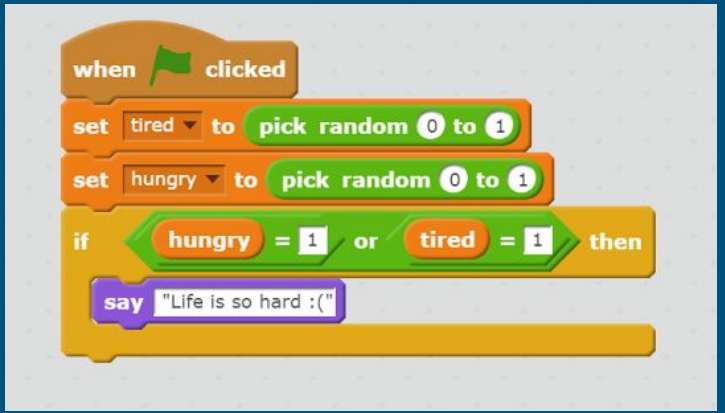
These three operations can save us time, as we do not have to nest the loops.



Try this:

Write a short programme…

If our sprite is hungry or tired (or both) when we run our programme, he complains.



https://scratch.mit.edu/projects/180194502/

Any questions on Scratch?

- Scratch is great for getting you to think like a programmer.
- It's intuitive user interface is great to play with, so definitely try and use your knowledge to make something cool.
- If you make a cool game and send it to me in the next weeks I will give you a head-start/hint for the CODING CHALLENGE

C

Week 1

C

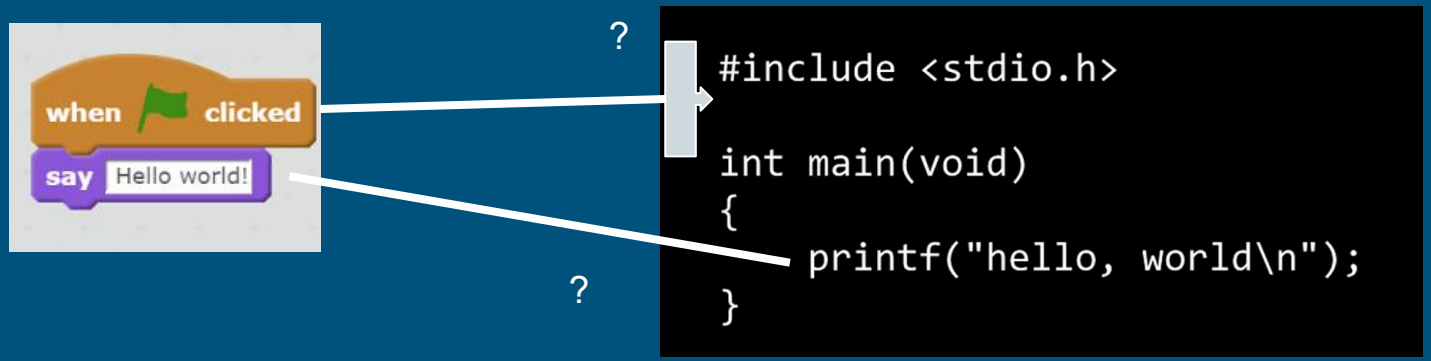
C was developed between 1969 and 1973. Many later languages have borrowed directly or indirectly from C, such as, C++, D, Go, Rust, Java, JavaScript, C#, Objective-C, Perl, PHP, Python, Swift, Verilog.

This means a lot of them are very syntactically similar to C, and the underlying structures are easily recognisable.

Therefore, learning C is a great place to start.

Let’s go back to where we started.

Here we see the same code in Scratch vs in C



Hello World

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

Let’s look at this closely:

Line #1: We didn’t see anything like #include in Scratch.

This just mean we are using the **standard input/output library**. A library is much like a library in the real world. We just borrow what we need from whatever library we want. How do we know what’s in the library? GOOGLE is your friend...

This library simply means that it deals with input (like from the keyboard) and output (printing characters to the screen). We can now use commands from this library to take in input and print out (you guessed it) output!

Hello World

```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

Line #2:

Again, this idea of “int main()” or “int main(void)” wasn’t introduced to us in Scratch.

“Main” is just the word that someone decided would indicate the “main function” of a program. This means the function (the lines of code) that will be called first and foremost when we run our program.

Think of Lines 1 and 3 as our little green flag from Scratch - they are what we need to get started!

Hello World

```c
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

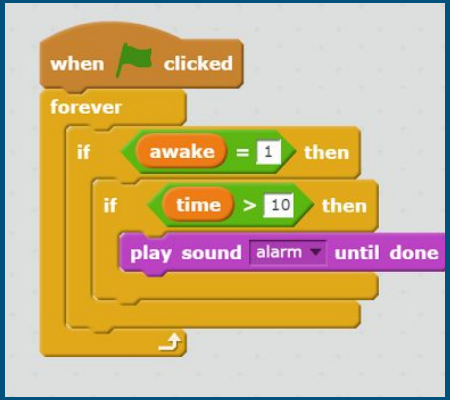
Those squiggly lines on Line #4 and #6?????

What do you think they might be? We saw this concept on Scratch.

These “braces” are very important. **They group a set of statements**.

We use them for loops and conditional statements. Think how in Scratch our blocks of instructions nested into our yellow blocks.

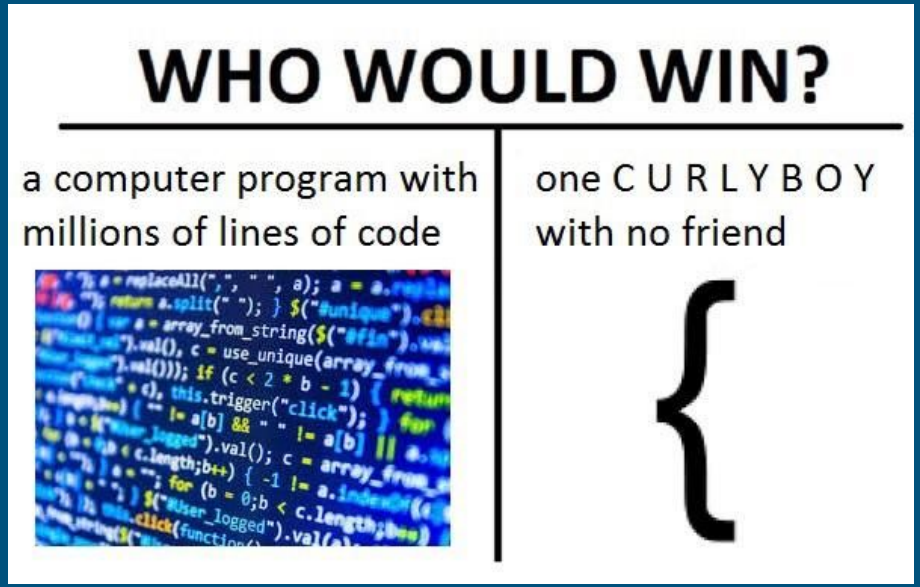
This is the purpose of curly brackets. To **contain** code.



Curly Braces

- Curly Braces are so important.
- Where you have one curly brace you **must** have another. They must be in pairs or your code will break.

- When you are using an IDE (Integrated Development Environment - basically pretty software to run and write code on) the **compiler** will catch your errors. We will be using Codeboard.io as our IDE!



Hello World

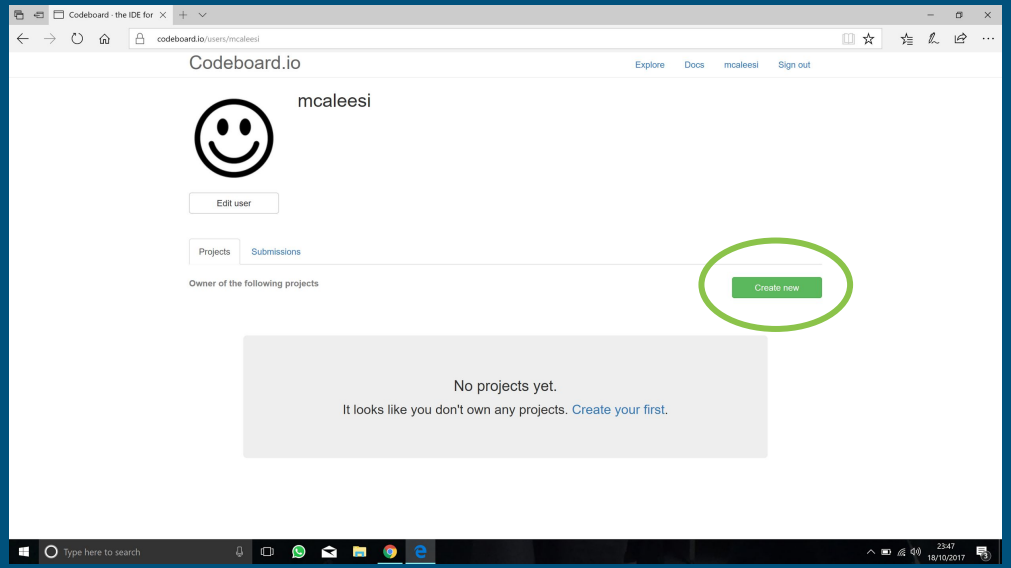
For the curious:

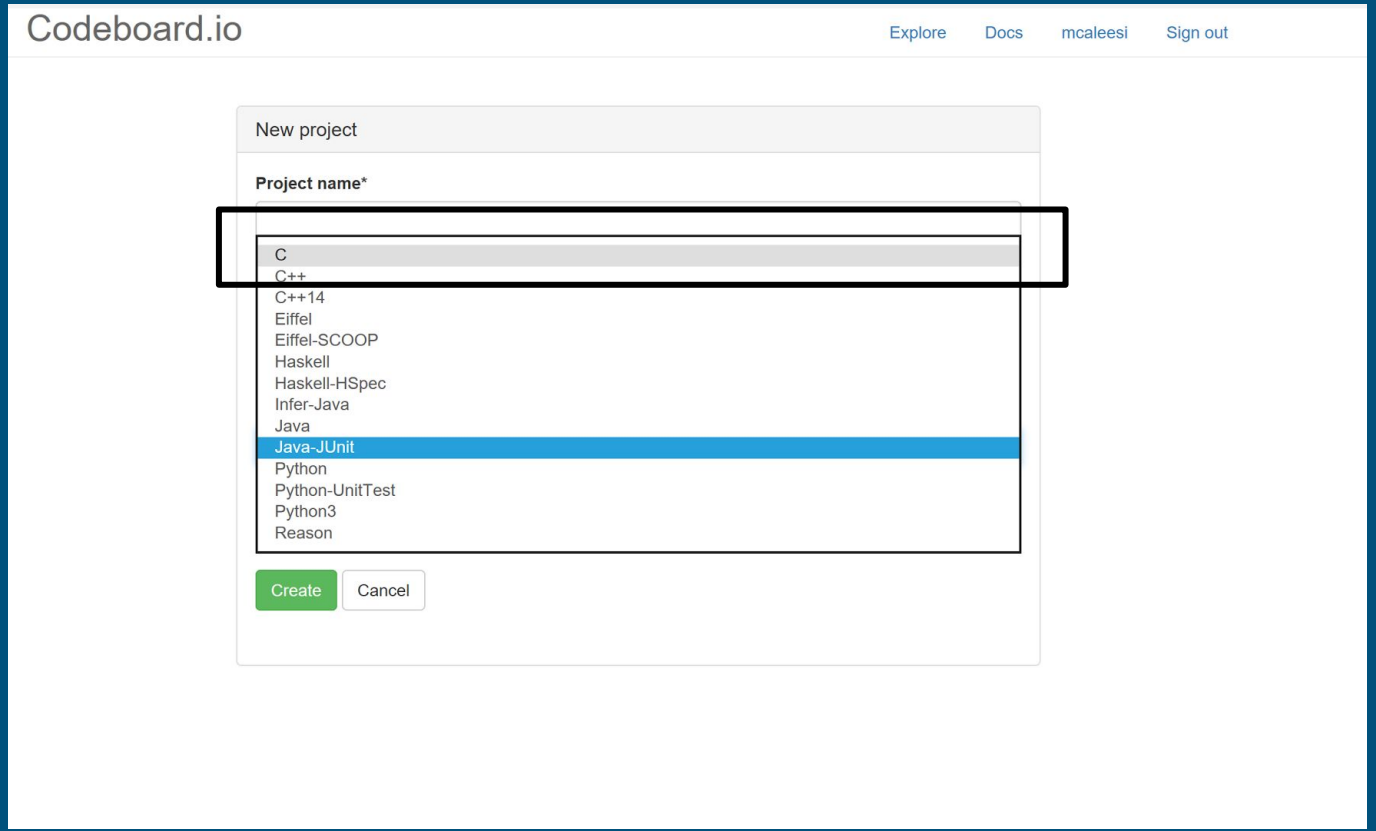
In Line #3 the meaning of the main(**void**) implies that the main can only be called without any *parameters*. It is technically better than writing main() but both will work.

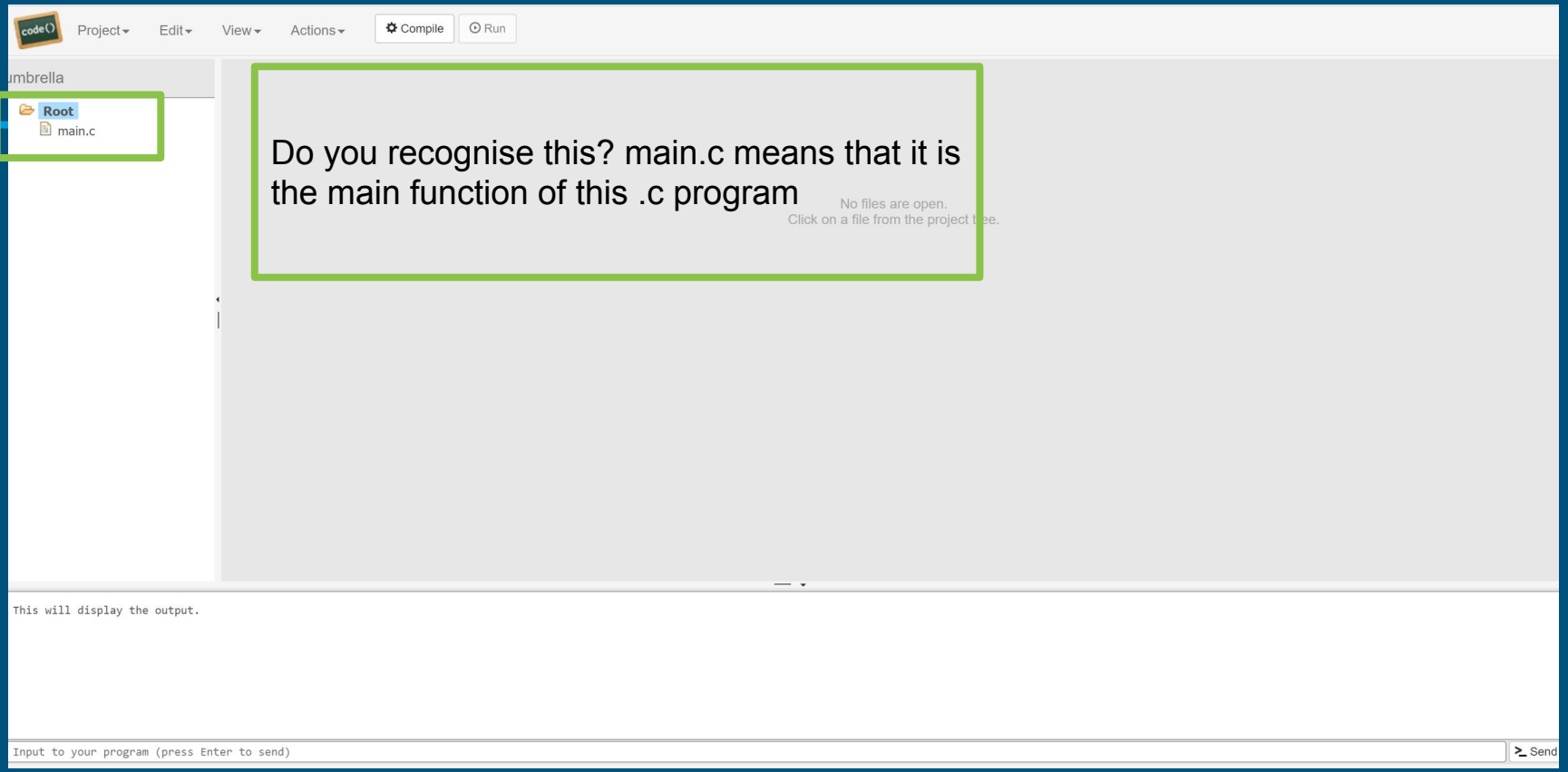
```
#include <stdio.h>

int main(void)
{
    printf("hello, world\n");
}
```

Codeboard.io Challenge #1

- Create an account & your first challenge is to try and get a “Hello World” program running.




Codeboard.io
Explore
Docs
mcaleesi
Sign out
New project
Project name*
C
C++
C++14
Eiffel
Eiffel-SCOOP
Haskell
Haskell-HSpec
Infer-Java
Java
Java-JUnit
Python
Python-UnitTest
Python3
Reason
Create
Cancel


Project
Edit
View
Actions
Compile
Run
umbrella
Root
main.c
Do you recognise this? main.c means that it is the main function of this .c program
No files are open.
Click on a file from the project tree.
This will display the output.
Input to your program (press Enter to send)
Send

Hello World

- You will see the code has already been written for you (with some additions)
- The task:
  - Delete the code you aren't familiar with (ie the extra #include and the return statement)
  - Insert the word void where we discussed
  - Find the COMPILE button - does your compilation succeed?
  - If yes - select Run.

- Congrats! You've just ran your very first C program!

Compile

For the curious:

We must press the compile button every time we wish to run a program? Why?

We code in C, but computers only understand binary (they only understand 1’s and 0’s). The compile button translates our code to a language the computer can understand and subsequently execute. It saves us a lot of time and effort.

If you want to learn more about this:
https://www.youtube.com/watch?v=CSZLNYF4Klo

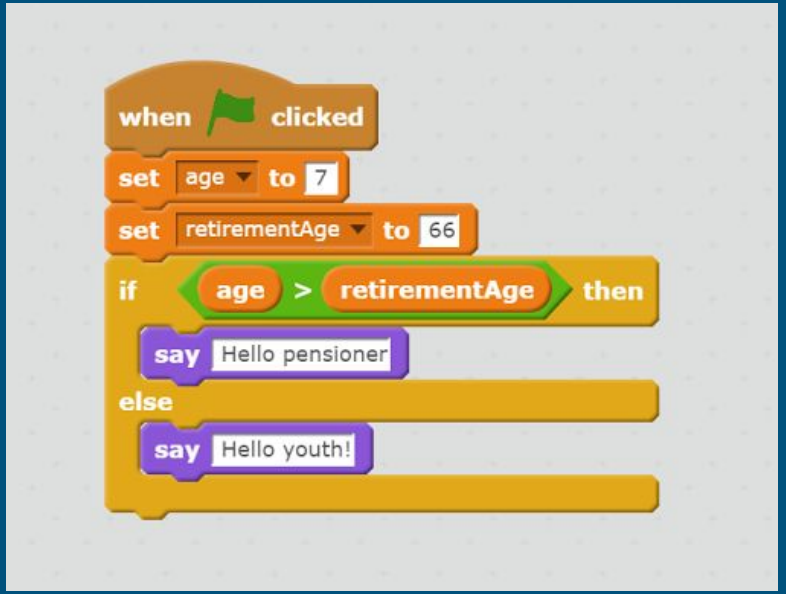
More Scratch translation

We know the block with the green flag will look like:

```
#include <stdio.h>
int main(void){                    }
```

The IF/ELSE is also straightforward:

```
if(age > retirementAge)
{
        printf(“Hello pensioner”\n);
}
else
{
        printf(“Hello youth!”\n);
}
```



Scratch Translation

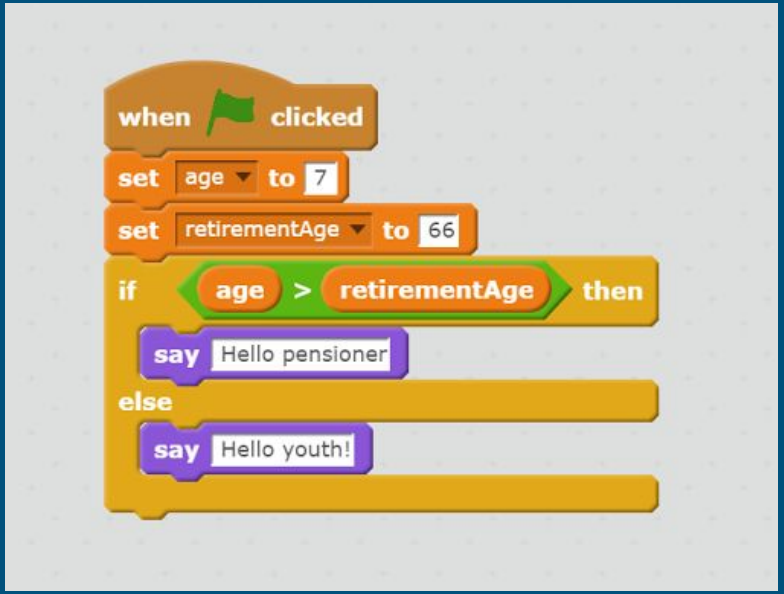
What about the two orange blocks?

We need to **define these variables**.

In C we write…

```
int age = 22;
```

… to define a variable. But what exactly is a variable?



Variables

- Variables are simply names used to refer to some location in memory – a location that holds a value with which we are working.
- This sounds a bit complicated so to put it simply…

- A **variable** is a (usually) short name we use to refer to a **value**. We can change, manipulate and work with this value - calling it by its variable name.
- Still not sure? Let’s look at an examples.

Variables

- Say we want to store someone's age in our program, we can define the age as a variable.

```
int age = 19;
```

- Now, when we want to do something to this age (perform an operation or use it as a condition) we just refer to it as `age`.

```
age = age + 1;
```

Variables

- What other things might we want to define?

Aoifé

3.14

true/false

86A

F

18869

What Data Types are Different Variables?

“6” or “19” or “18869” are ints (integer).
We define these: int age = 19;
Or int height = 170;

“3.14” or “0.001” are doubles
(They may be called floats but for now we will call them doubles!)
We will define these: double pi = 3.14;
Or double temperature = 14.3;

“F” or “a” or “&” are all char (think of character)
We define these: char grade = “F”;
Or char answerSelected = “a”;

“Aoifé” or “dog” or “Trinity Hall, Dartry” are a bit different. These values are all made up of many chars in a row. So, when defining the variable we store them all in a list(or array) and point to the first one in the list, as follows:

char * address = “Trinity Hall, Dartry”;

char * name = “Aoifé”;

Data Types

So now we have learnt the three main data types in C.

- Int
- Char
- Double

And we also have learnt

- Char * points to the first char in a sequence of chars.

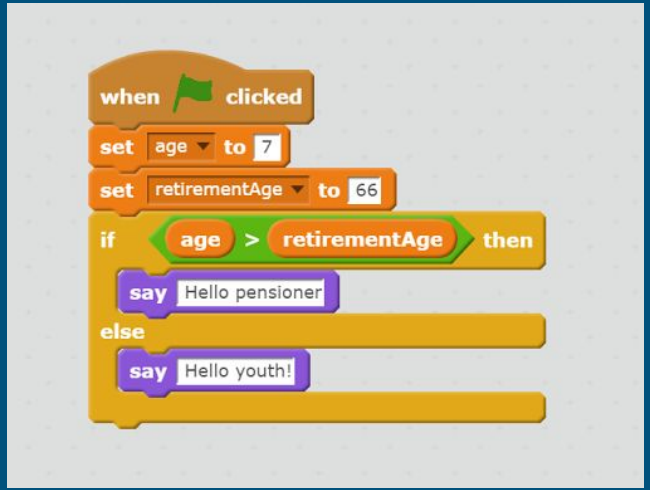
Program #2

- So we know we can define a number using int
- We know how to write IF/ELSE statements

int age = 19;

```
if (x > y)
{
        printf("Hello world!\n");
}
else
{
        printf("Goodbye world.\n");
}
```

SO, can you code, compile and run this --->



Your Task for Next Week:

- Write a program which determines if the user should bring an umbrella when they go out (if it looks like it might rain or is raining) and if so whether they should put it up (if it is raining).
- What you will need:
  - Two Booleans for whether or not it is raining and whether or not is looks like it will rain (in C booleans are simply integers that are 1 or 0)
  - You will need an IF/ELSE statement
  - You will need two printf statements telling the user whether or not to bring an umbrella.
  - You can decide whether or not it is raining! Change it up and see if it works for all possible outcomes.