“Get Techy”

Trinity Hall JCR Programming Club
Session 1

Sinéad McAleer - Tech Officer
mcaleesi@tcd.ie

“Coding is the closest thing we have to superpowers.”

- CEO of Dropbox

The Plan

WEEK 1	October 25th	Introduction to Coding & SCRATCH
WEEK 2	November 1st	SCRATCH & Intro to C
	November 8th	*READING WEEK*
WEEK 3	November 15th	C
WEEK 4	November 22nd	C
WEEK 5	November 29th	C
WEEK 6	December 6th	CODING CHALLENGE

After this we will regroup and decide what people would like to learn next. For example, next semester we could focus on web-based languages (HTML and Javascript) if there is interest. Also, let me know if anyone would like an ARM assembly workshop to help with your Computing modules.

Coding Challenge

- On the 6th Week we will hold a coding challenge.
- You will be given 30 minutes to complete a short C program.
- The winner will get a free ticket to next JCR event and 40 EURO
- The runner-up will get 15 EURO and the most improved coder will also get 15 EURO
- The Challenge will only be open to those who have attended at least 3 Programming Club nights.
- If you complete at least 4 Programming Club nights you will get a certificate AND a free can AND something amazing to put on your CV

The only way to learn how to code is to practice coding. You can’t learn from listening to an instructor or watching a video - you have to try yourself.

Github

- You will be able to find all the slides and code we talk about on Github. Github is used by computer scientists all around the world to share and collaborate on code
- On Github you can make suggestions to other people's code, you can work on code in a team or you can submit your own as a type of Technical CV

https://github.com/sineadmcaleer/get-techy

Intro to Coding

Week 1

Computer Programming

Also known as Coding

Coding is all around you. Whether you are sending a tweet, watching a YouTube video or swiping your debit card there is always lines of code doing the work.

Code is a “precise set of instructions that a computer can understand”. You could think of it as a recipe that makes a very specific dish.

```cpp
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

```
Log.Info("Entering method");

if (actualLogins == null)
    return new List<GroupData>();

Log.Info("Logins:  " + (actualLogins == null ? "null" : " not null"));
Log.Info("Host: " + Host);

var range = IntegerRange.Parse(actualLogins.Logins);
var grpNames = _managerCache.Data
    .FindAll(it => range.IsInRange(it.Login))
    .ConvertAll(it => it.Groups);

Log.Info("Groups " + grpNames.Count);
```

Languages

Just like how people from different countries speak different languages, there are hundreds of different languages a computer can understand. (https://en.wikipedia.org/wiki/List_of_programming_languages). Some of the more common ones are:

- C/C++
- Java
- Python

(The above are all used by Google)

- Javascript (used by Air BnB)
- PHP (used by Facebook)

- You may have heard of HTML - but this is not a coding language. It's a markup language which is a bit different, and not really as much fun as the kind of stuff we will be doing!

Languages

All languages come with their benefits and drawbacks.

The focus of coding is not learning one language. When you learn to code in one language you are learning the **art** of programming.

Once you can program in one language you can easily transfer the concepts you have learned.

The hard bit is learning your first language.

This is why we are starting off with Scratch.

Scratch

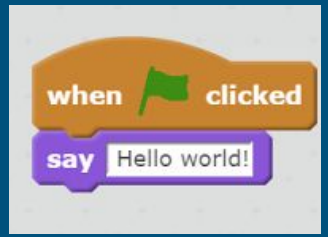
- Scratch is a visual programming language
- It was created about 15 years ago in MIT
- Seen as a common stepping stone into the world of coding, it will help you start thinking like a programmer
- It is fun!

Scratch is available online at:

https://scratch.mit.edu/

Scratch includes dragging and dropping puzzle pieces.

This is what Scratch looks like:



This is what C looks like:

```
#include <stdio.h>

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

Often the semi-colons and braces ({/}) can be distracting and intimidating and take away from the enjoyment of programming.

Note:

Computer Science is not the same as Computer Programming. Computer Science is theoretical and mathematical, while Computer Programming is practical.

If you are interested in Computer Science

Crash Course (12-part series):

https://www.youtube.com/playlist?list=PLME-KWdxI8dcaHSzzRsNuOLXtM2Ep_C7a

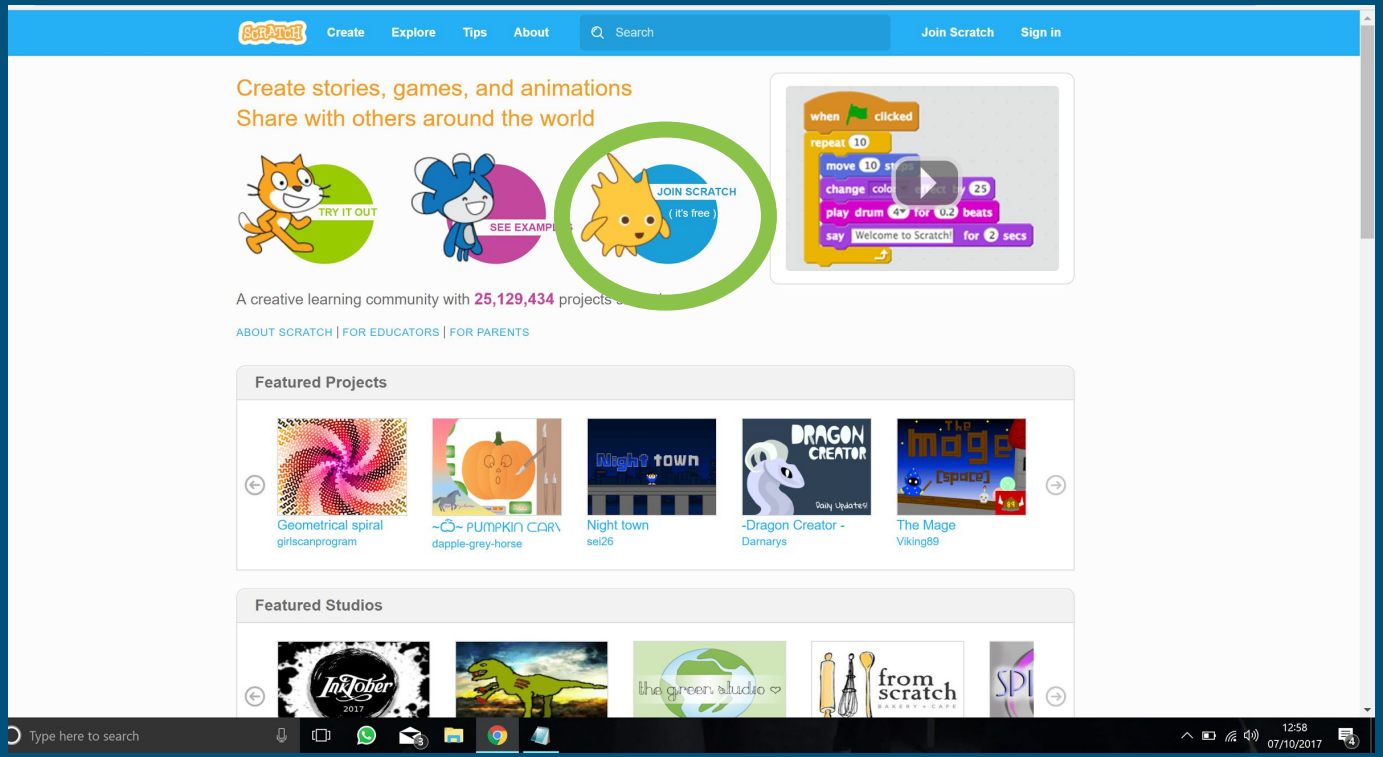
If you are interested in the Maths behind computer science:

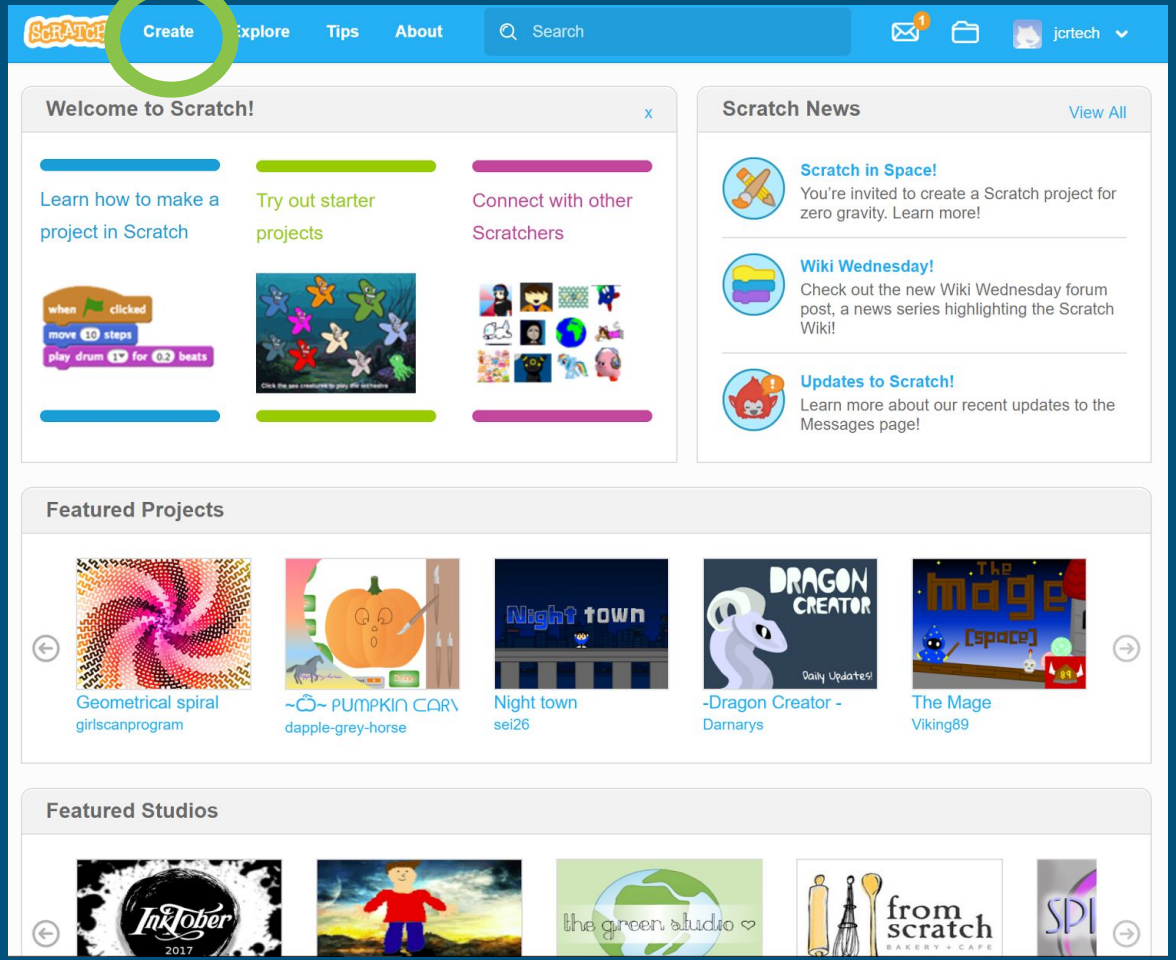
https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-042j-mathematics-for-computer-science-fall-2010/video-lectures/

Intro to Scratch

Week 1

Scratch - Create an Account https://scratch.mit.edu/




Create
Explore
Tips
About
Search
jcrtech
Welcome to Scratch!
x
Learn how to make a project in Scratch
Try out starter projects
Connect with other Scratchers
when clicked
move 10 steps
play drum 1 for 0.2 beats
Scratch News
View All
Scratch in Space!
You're invited to create a Scratch project for zero gravity. Learn more!
Wiki Wednesday!
Check out the new Wiki Wednesday forum post, a news series highlighting the Scratch Wiki!
Updates to Scratch!
Learn more about our recent updates to the Messages page!
Featured Projects
Geometrical spiral
girlscanprogram
~ PUMPKIN CARV
dapple-grey-horse
Night town
sei26
DRAGON CREATOR
Daily Updates!
-Dragon Creator -
Darnarys
The mage
[space]
The Mage
Viking89
Featured Studios
InkTober
2017
the green studio
from scratch
BAKERY + CAFE

This is our Sprite who we can control using instructions

This is the types of instructions we can use to control our sprite

These are the actual instructions

Instructions/Scripts:

Category	Notes
Motion	Moves sprites and changes angles and change X and Y values
Looks	Controls the visuals of the sprite; attach speech or thought bubble, change of background, enlarge or shrink, transparency, shade
Sound	Plays audio files and programmable sequences
Pen	Draw on the portrait by controlling pen width, color, and shade. Allows for turtle graphics.
Data	Variable and List usage and assignment

Category	Notes
Events	Contains event handlers placed on the top of each group of blocks
Control	Conditional if-else statement, "forever", "repeat", and "stop"
Sensing	Sprites can interact with the surroundings the user has created
Operators	Mathematical operators, random number generator, and-or statement that compares sprite positions
More Blocks	Custom procedures (blocks) and external devices control and can import from PicoBoard or Lego WeDo 1.0/2.0

How can we make our Sprite dance?

https://en.wikipedia.org/wiki/Scratch_(programming_language)

https://scratch.mit.edu/projects/178617528/

How can we make our Sprite dance?

Category	Notes
Motion	Moves sprites and changes angles and change X and Y values
Looks	Controls the visuals of the sprite; attach speech or thought bubble, change of background, enlarge or shrink, transparency, shade
Sound	Plays audio files and programmable sequences
Pen	Draw on the portrait by controlling pen width, color, and shade. Allows for turtle graphics.
Data	Variable and List usage and assignment

Category	Notes
Events	Contains event handlers placed on the top of each group of blocks
Control	Conditional if-else statement, "forever", "repeat", and "stop"
Sensing	Sprites can interact with the surroundings the user has created
Operators	Mathematical operators, random number generator, and-or statement that compares sprite positions
More Blocks	Custom procedures (blocks) and external devices control and can import from PicoBoard or Lego WeDo 1.0/2.0

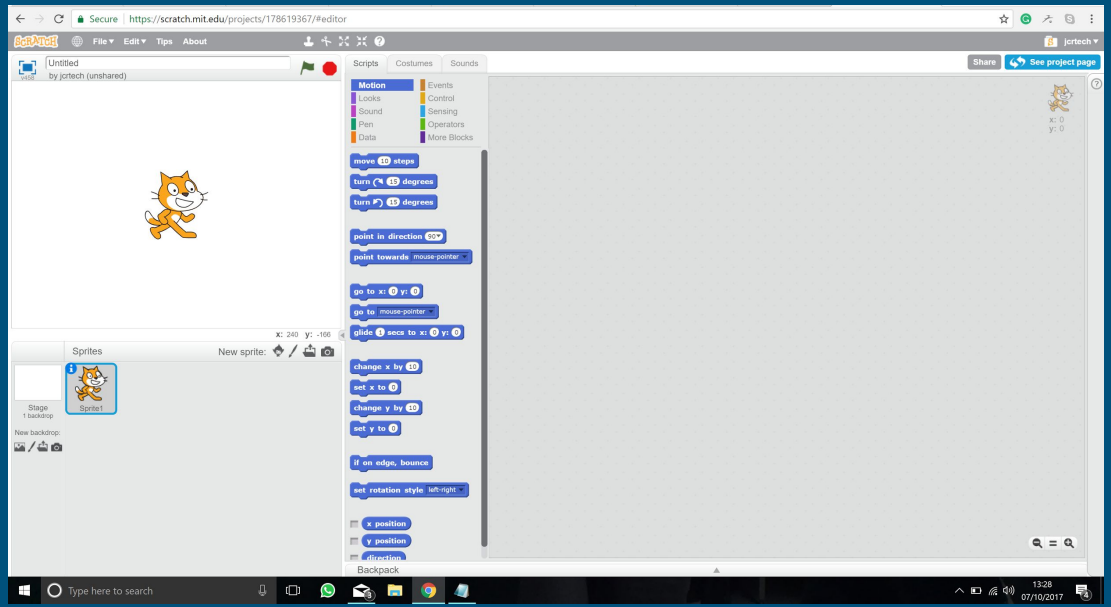
Problem #1 - Make our Sprite dance

We will need to use the categories “Motion”, “Sound”, “Look” and “Control”.

For motion, all the Sprite is doing is taking 10 steps forward and 10 steps back.

For sound, we simply play two drum beats after each other.

For control, we must repeat the movement and drum beats for 10 seconds.



Let’s try this on Scratch

Making our Sprite Dance

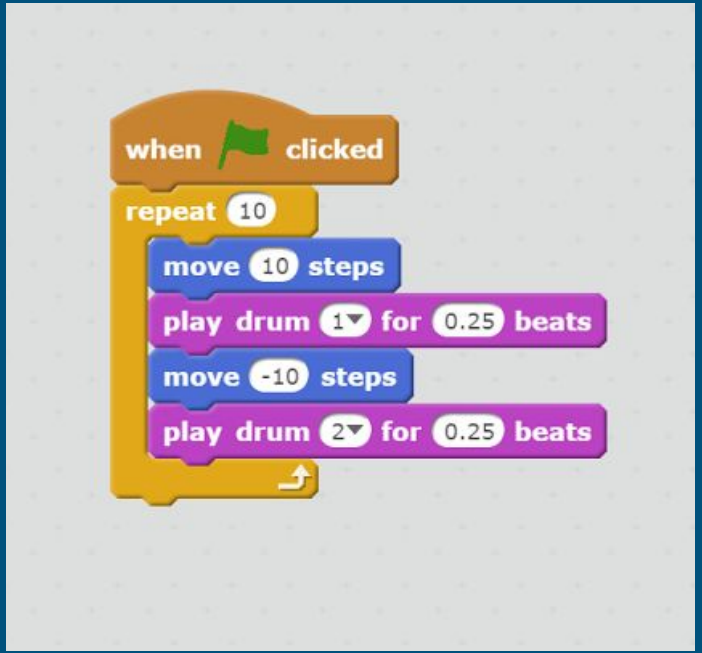
This is the code we just wrote.

This means for 10s after the green flag is selected the cat will dance and the beat will play.

What if we want the cat to dance for more than ten seconds?

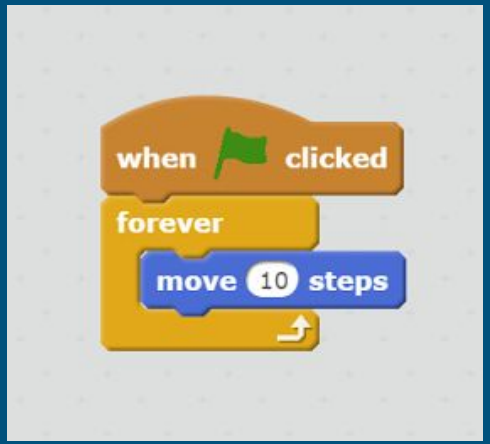
What if we want the cat to dance forever?

Pay close attention to these controls - which in time we will learn to call loops.



Problem #2- Make our cat walk...

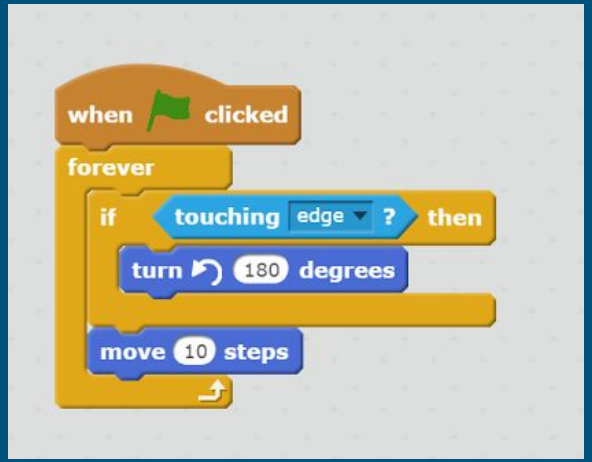
This code allows our cat to walk.



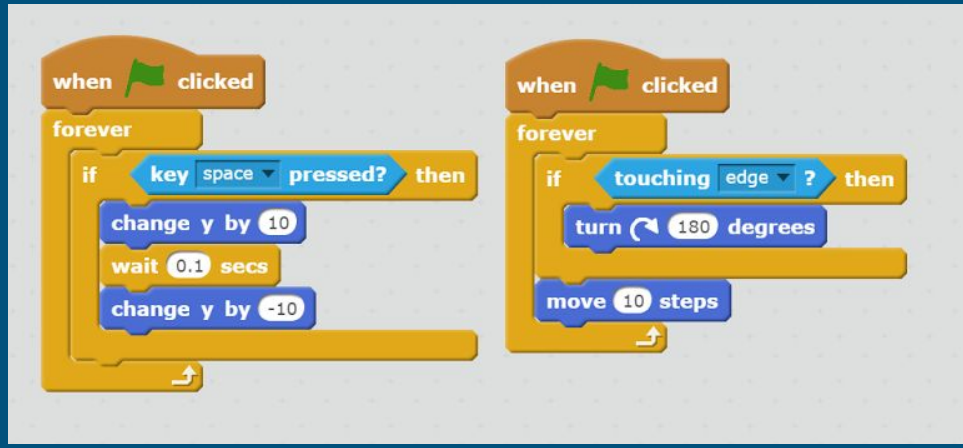
Try this on Scratch.

When he walks to the edge of the screen we lose him. How could we articulate this?

We want our cat to walk, but **if** he touches the side of the screen we want him to change direction. (Pay attention to the word IF here)



There is a lot you can do with IF



IF/ELSE

Imagine we want to say:

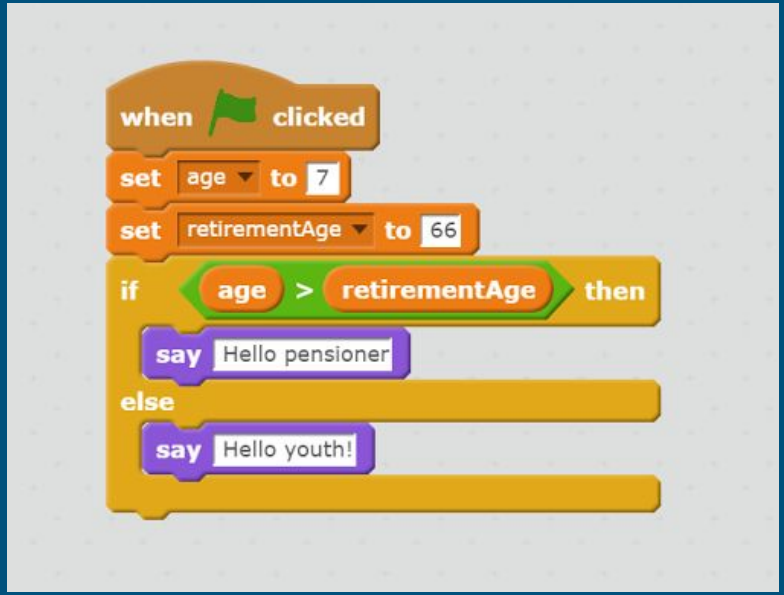
IF a person is older than 66 say “Hello pensioner”

Otherwise, we want our cat to say “Hello youth!”

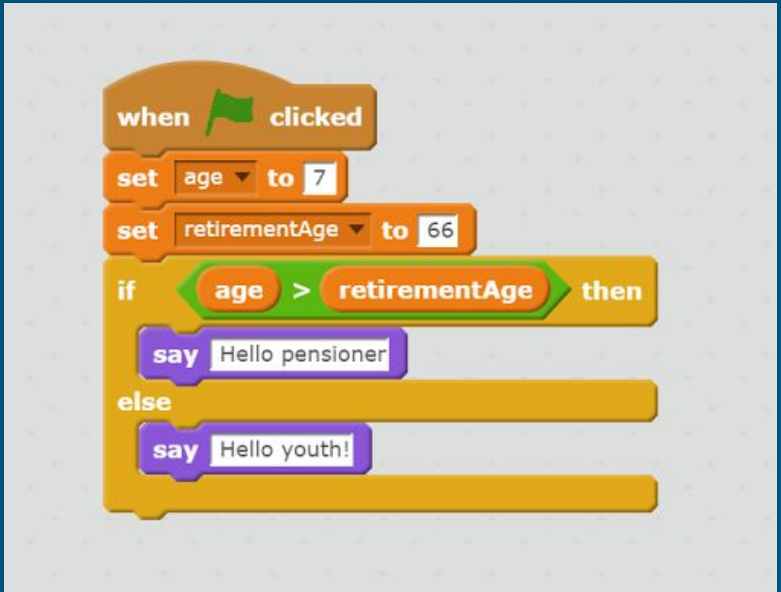
Instead of using the word “otherwise” we use

ELSE

This if statement also includes conditions. We will look at this again in the future.



Variables



Let’s look closer at lines 2 & 3:

We have created two VARIABLES. Variables store data. For example, name will store “Aoifé” and age will store “19”. We will talk more about this next week.

So far...

- Made our sprite move
- Played audio
- Done loops
- Made our sprite talk
- Completed an IF/ELSE statement
- Declared variables

We can combine them all into one program - Apple Game

https://scratch.mit.edu/projects/178650343/

Challenge #1: Make this game yourself

Next time...

- Boolean
- More variables
- Conditions
- Introduction to C

- In the next week try and play with Scratch at least twice - for about 30 minutes.
- Remember - the only way to learn how to code is to practice.
- For next Wednesday, **design a completely new game with at least three Sprites and at least four if statements**
- If you can show your game at the start of the next class you will be given a head-start hint for Week 6 Coding Challenge - giving you a better chance of winning the PRIZE