

Lesson 1

Teaching Your Kids
Programming with



About me

- Ana Jalba
- Associate Software Engineer
- BDP – Hotels.com
- Computer Science – King's College London

About you

- Name
- Role
- Why you chose to do this (short) course

Coming up

- Why knowing how to program is important for kids
- What programming is
- Code examples
- What it's like to be a programmer
- Introduction to Scratch
- Making a flying cat

Why is this important?

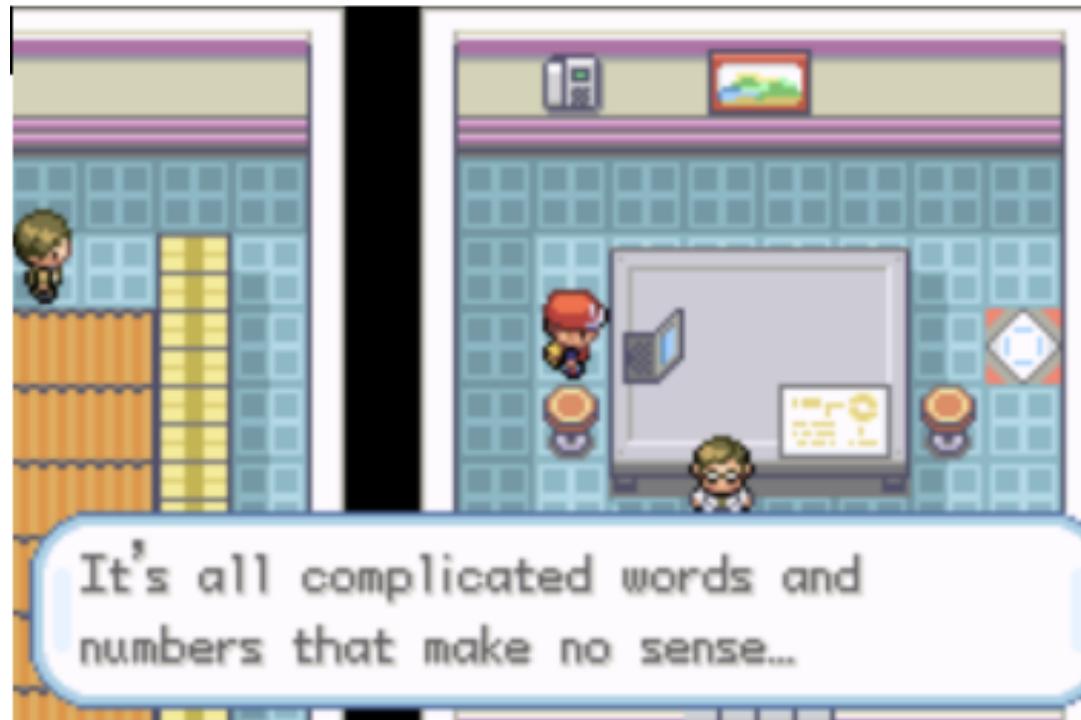
- Creativity
- Confidence
- Resilience / persistence
- Communication and collaboration
- Improve understanding of abstract concepts (like maths)
- Opens up many career options

Notes

- A lot of topics may not make sense the first time around. So, refrain from negative self-thoughts
- No practice = little to no improvement
- Need to strike the balance of what you tell the computer to do and what you think you told it to do
- It's important to not give the whole answer away
- If at any point you don't understand something, please **stop me and let me know**

What is programming?

When you start programming for
the first time



Source: reddit.com/r/programminghumor

Code examples (with popularity according to GitHub)

JavaScript (#1)

```
var x = 1;
if (x > 0) {
    console.log("x is positive");
} else if (x == 0) {
    console.log("x is 0");
} else {
    console.log("x is negative");
}
```

Java (#3)

```
int x = 1;
if (x > 0) {
    System.out.print("x is positive");
} else if (x == 0) {
    System.out.print("x is 0");
} else {
    System.out.print("x is negative");
}
```

Python (#2)

```
x = 1
if x > 0:
    print("x is positive")
elif x == 0:
    print("x is 0")
else:
    print("x is negative")
```

Ruby (#4)

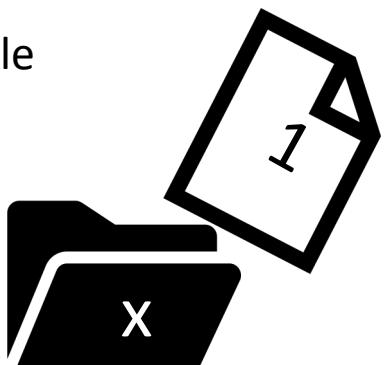
```
x = 1
if x > 0
    puts "x is positive"
elsif x == 0
    puts "x is 0"
else
    puts "x is negative"
end
```

Variables store values

JavaScript (#1)

```
var x = 1;  
if (x > 0) {  
    console.log("x is positive");  
} else if (x == 0) {  
    console.log("x is 0");  
} else {  
    console.log("x is negative");  
}
```

Variable



Conditional statements say if something is true or false

JavaScript (#1)

```
var x = -1;  
if (x > 0) {  
    console.log("x is positive");  
} else if (x == 0) {  
    console.log("x is 0");  
} else {  
    console.log("x is negative");  
}
```

Conditional statement:

If a condition is met, the statement evaluates to **true**, otherwise, to **false**

Printing something

JavaScript (#1)

```
var x = 1;  
if (x > 0) {  
    console.log("x is positive");  
} else if (x < 0) {  
    console.log("x is negative");  
} else {  
    console.log("x is 0");  
}
```

Printing statement:

Tells the console/terminal to log the text “x is positive”

Software developers google stuff all the time



Source: reddit.com/r/programminghumor

From xkcd comics: Wisdom of the ancients

NEVER HAVE I FELT SO
CLOSE TO ANOTHER SOUL
AND YET SO HELPLESSLY ALONE
AS WHEN I GOOGLE AN ERROR
AND THERE'S ONE RESULT
A THREAD BY SOMEONE
WITH THE SAME PROBLEM
AND NO ANSWER
LAST POSTED TO IN 2003



It's not always easy

The image shows two screenshots of the CS50 Sandbox interface. In both screenshots, a student has written a simple C program named 'hello.c' that prompts the user for their name and prints it back. The code is as follows:

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     string name = get_string("What is your name?\n");
6     printf("hello, name\n");
7 }
```

In the top screenshot, the terminal output shows a syntax error at line 5:

```
stdin
/usr/include/stdio.h:168:2: error: expected identifier before 'string'
extern struct _IO_FILE *stdin; /* Standard input stream. */
hello.c:5:11: error: expected identifier before 'string'
    string name = get_string("What is your name?\n");
               ^
```

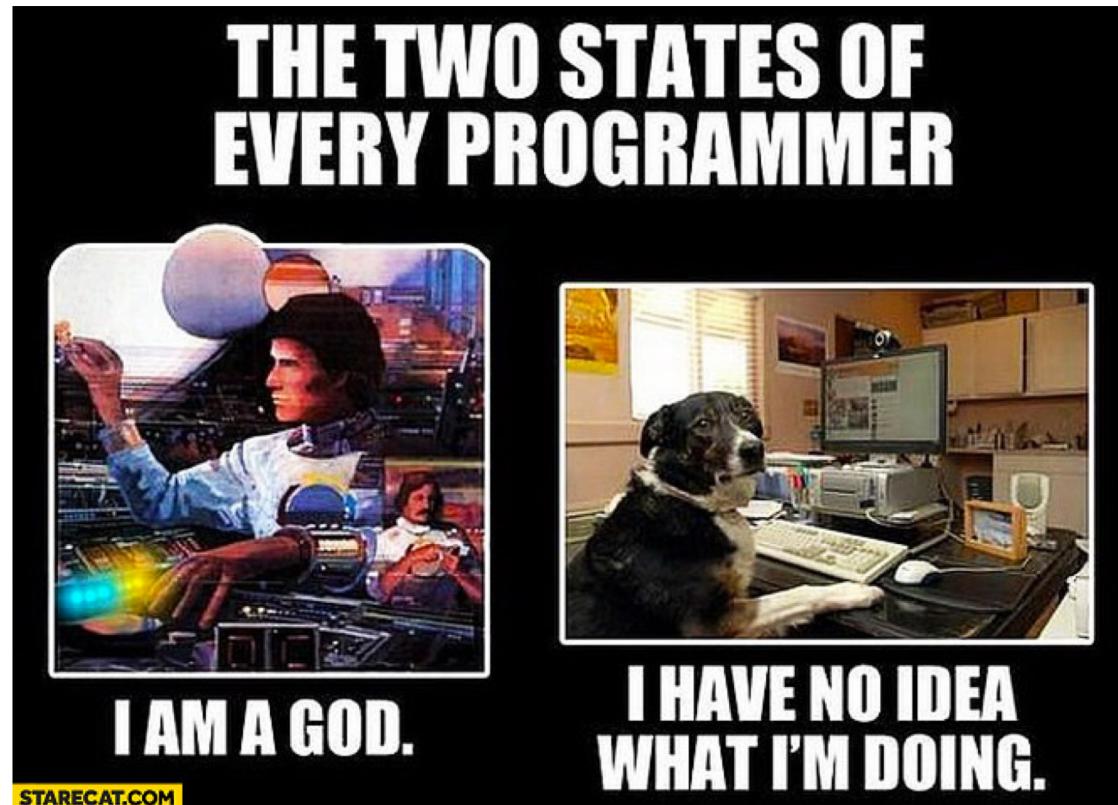
A large text overlay on this screenshot reads: "oh my god, I only wrote two lines of code."

In the bottom screenshot, the terminal output shows the same error, but the student has now added a closing brace at the end of the code:

```
stdin
/usr/include/stdio.h:168:2: error: expected identifier before 'string'
extern struct _IO_FILE *stdin; /* Standard input stream. */
hello.c:5:11: error: expected identifier before 'string'
    string name = get_string("What is your name?\n");
               ^
```

A large text overlay on this screenshot reads: "How do I have 20 lines of errors somehow?"

But when you make it work, it's very satisfying



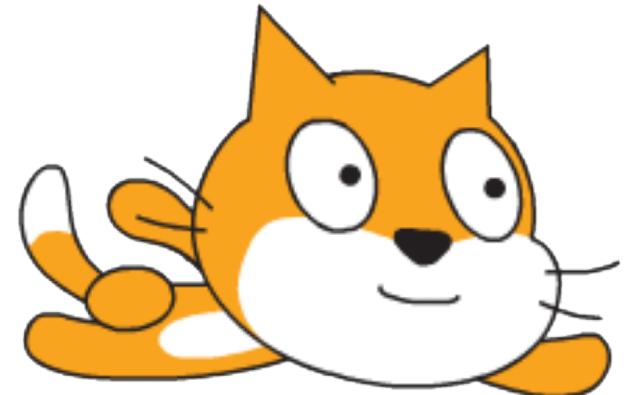
Introduction to Scratch:

- Scratch helps young people learn to:
 - think creatively
 - reason systematically
 - work collaboratively
- Lets you program your own interactive stories, games, and animations

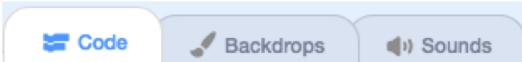


Let's make a flying cat

By using the “Make It Fly” tutorial



<https://scratch.mit.edu>

- Make an account (it's free)
-  Create Explore Ideas About click on create to start your project
-  Sprite = character – you can choose your own
-  Backdrop = background
-  – how to start and stop the program
-  – allow you to control what the *sprite* does (after you click on the *sprite* image)
-  – allow you to control what the *backdrop* does (after you click on the *backdrop* image)

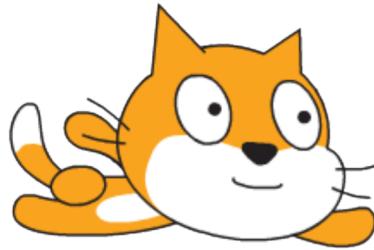
The image shows the Scratch programming environment with several key areas highlighted:

- Blocks palette (Left):** A vertical sidebar containing categories like Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The "Motion" category is currently selected, displaying blocks such as "move (10 steps)", "turn (15 degrees)", and "glide (1 sec to random position)".
- Scripts area (Center):** A large central workspace where scripts can be created. It features a pink background and contains the text "Scripts area".
- Stage (Right):** A green-bordered area representing the stage where sprites move. It contains a yellow cat sprite at [x: -240, y: 180] and [x: 240, y: 180]. The text "Stage" is overlaid on the stage area.
- Sprites list (Bottom Right):** A blue-bordered list showing the current sprites. It includes "Sprite1" (a yellow cat, x: 0, y: 0, size: 100, direction: 90), "Stage" (the green stage backdrop), and a "Backdrops" section with one item.

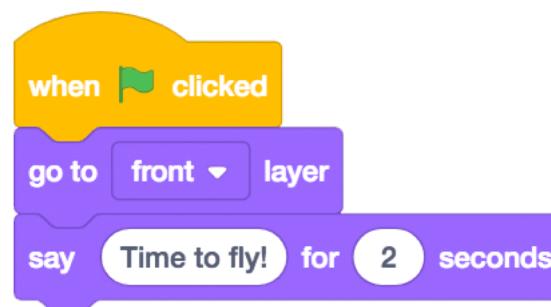
Large, bold text overlays are present in the center and bottom right areas:

- Scripts area** (in the center)
- Stage** (overlaid on the Stage area)
- List of sprites** (in the bottom right)

Prepare to fly



- From **Events**, drag to the sprite's script area
- From **Looks**, drag:
 -
 -
- attach them to “when flag clicked”
- It should look like this at the end



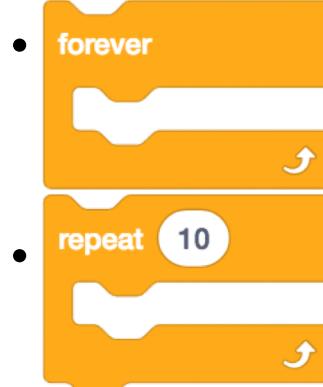
The scenery



- Add a new sprite in the shape of a building
 - new sprite → buildings
- Drag it to the side of the stage
- Now we're going to make it look like the cat is flying

Making the scenery move

- From **Control**, drag:

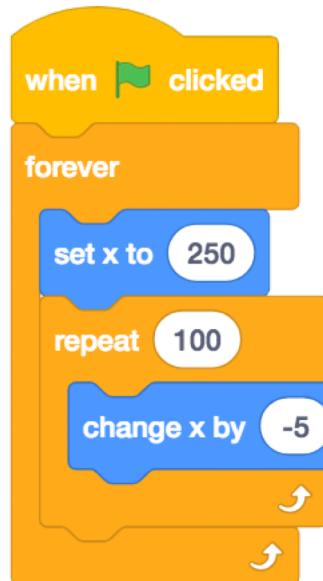


- From **Motion**, drag:

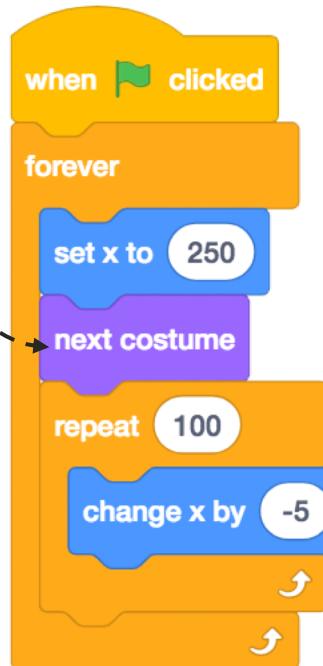
- **set x to 0** and set it to 250 (the right end of the stage)
- **change x by 10** and set it to -5 (moves to the left)

- Make sure you are working on the building sprite

Aiming for this:



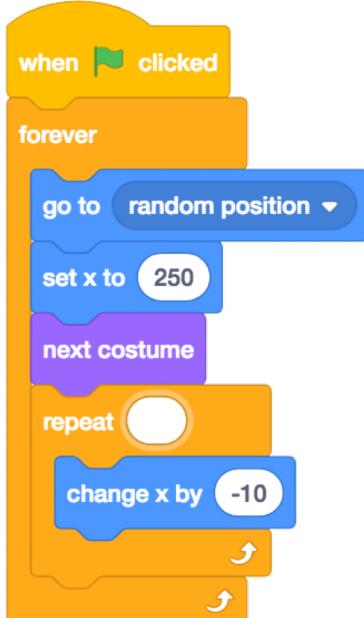
If you want to change the costume after each loop



Painting the sky



- Click the  icon to change the colour of the backdrop to blue
- Add other thing(s) (clouds, rainbows) to the sky as a new sprite
- To make the new sprite float, add a script like this to it:



*If we wanted the animation of buildings and clouds to stop at the same time, what's the number we should put for **repeat**?*

Next time

- Making a game in Scratch