Giovanni Malafarina
Istituto Tecnico Tecnologico

Code The Future

2019-1-TR01-KA229-074007

Soverato - Italy 9th-13th December 2019

1st Joint Staff Training

Prof. Franco Corapi

f.corapi@itmalafarina.edu.it

Scratch Integration to the current Curricula

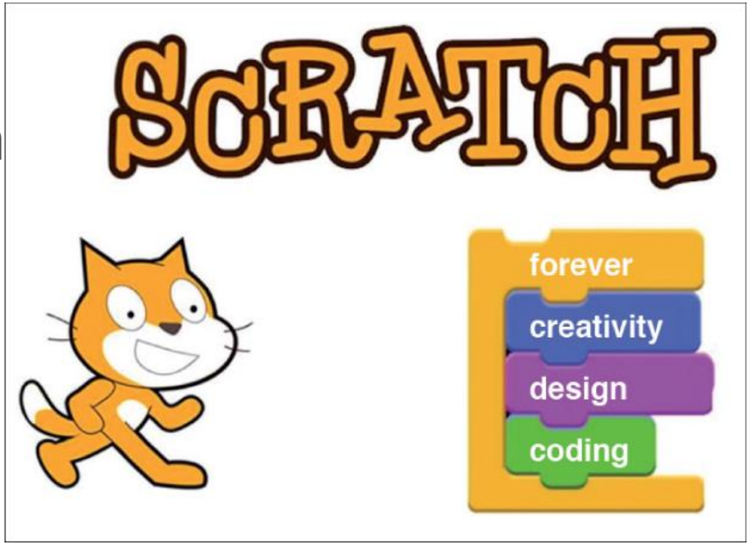
Co-funded by the Erasmus+ Programme of the European Union

About me: Franco Corapi

- Computer Science and Networking Teacher at ITT Malafarina -Soverato
- Teaching Innovation Team Member
- Expert in Educational Technology

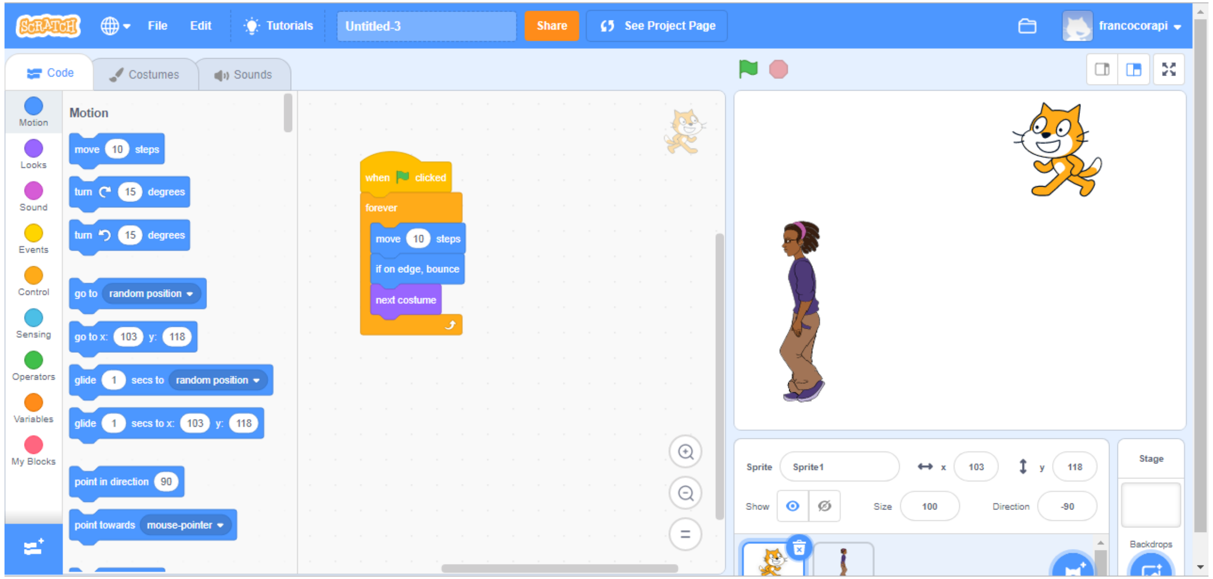
Goals for this lesson

- ❑ Knowing scratch
- ❑ Building basic script with scratch
- ❑ Integrating scratch into lessons



What is Scratch?

Scratch is a graphical programming language, developed by the Lifelong Kindergarten group at the Massachusetts Institute of Technology. Children can drag and combine code blocks to make a range of programs, including animations, stories, musical instruments and games.



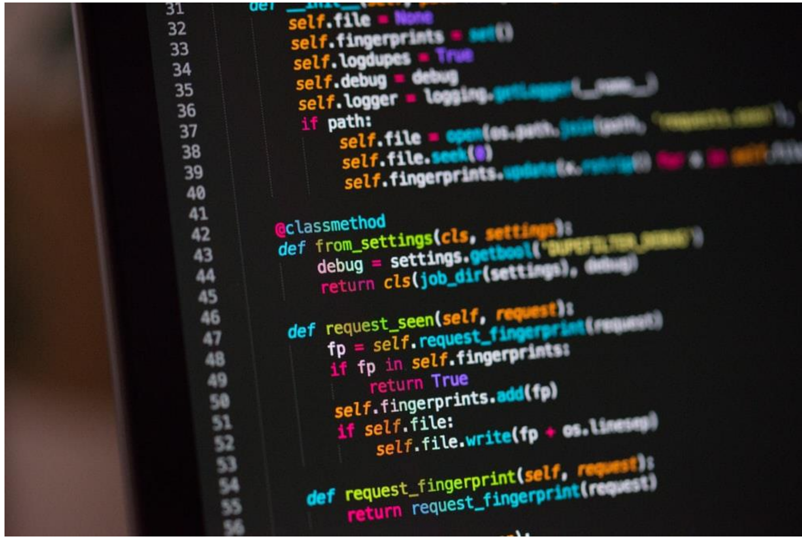
```
Motion
move (10) steps
turn ↻ (15) degrees
turn ↺ (15) degrees
go to (random position ▾)
go to x: (103) y: (118)
glide (1) secs to (random position ▾)
glide (1) secs to x: (103) y: (118)
point in direction (90)
point towards (mouse-pointer ▾)
```

```
when ⚑ clicked
forever
    move (10) steps
    if on edge, bounce
    next costume
```

Sprite: Sprite1, x: 103, y: 118, Size: 100, Direction: -90

What is programming?

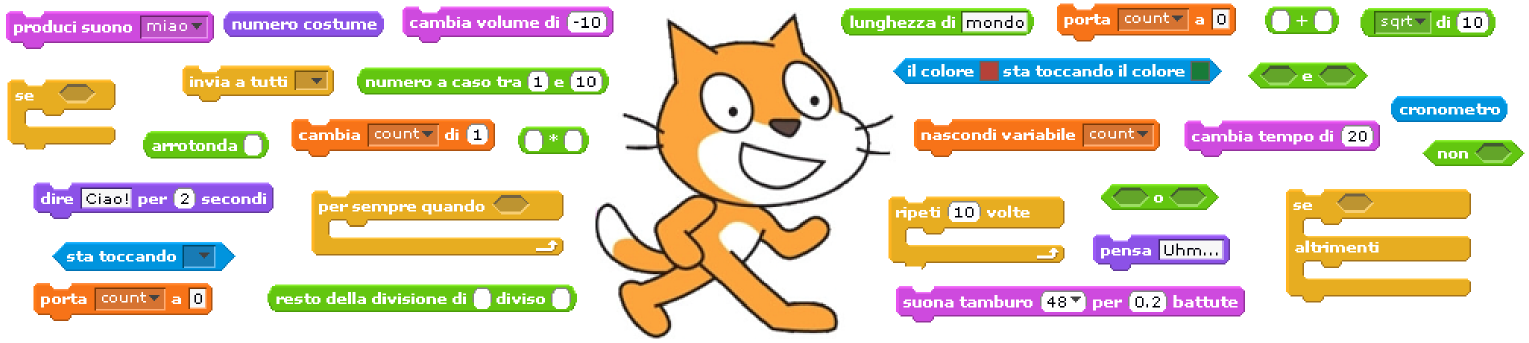
- Programming is the art of making a computer do what you want it to do.
- A computer program is simply a set of instructions to tell a computer how to perform a task.
- It is like a recipe: a set of instructions to tell a cook how to make a dish.



```
    def [illegible]
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, [illegible]
            self.file.seek(0)
            self.fingerprints.update([illegible]

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool([illegible])
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

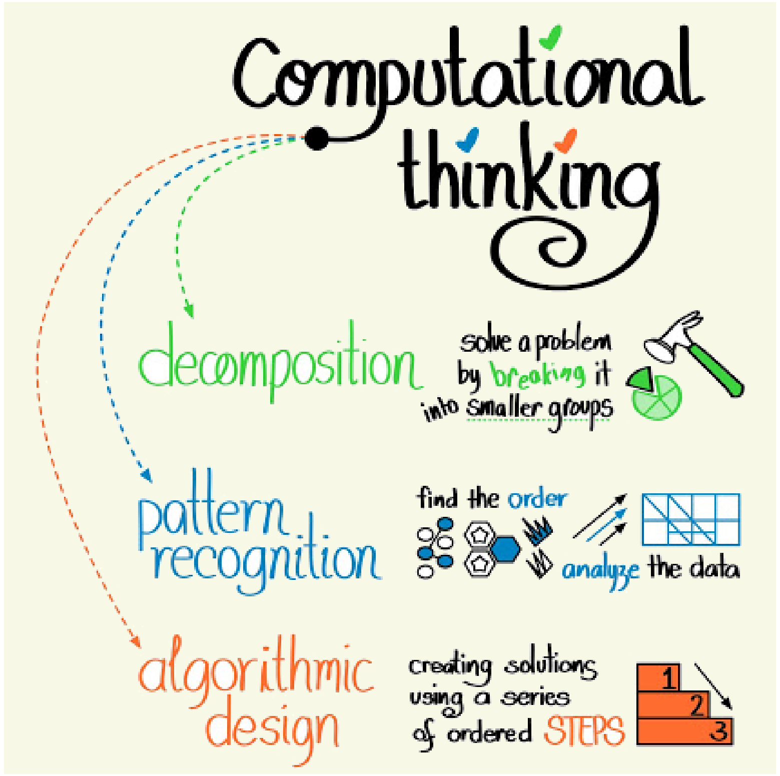
Why Scratch?

- Becouse Scratch editor allows students to learn coding concepts and create interactive projects without needing to learn a programming language.
- It’s FREE ☺



What can we learn from Scratch?

- We can learn important computational ideas.
- We can learn to think creatively.
- We can learn to reason systematically.

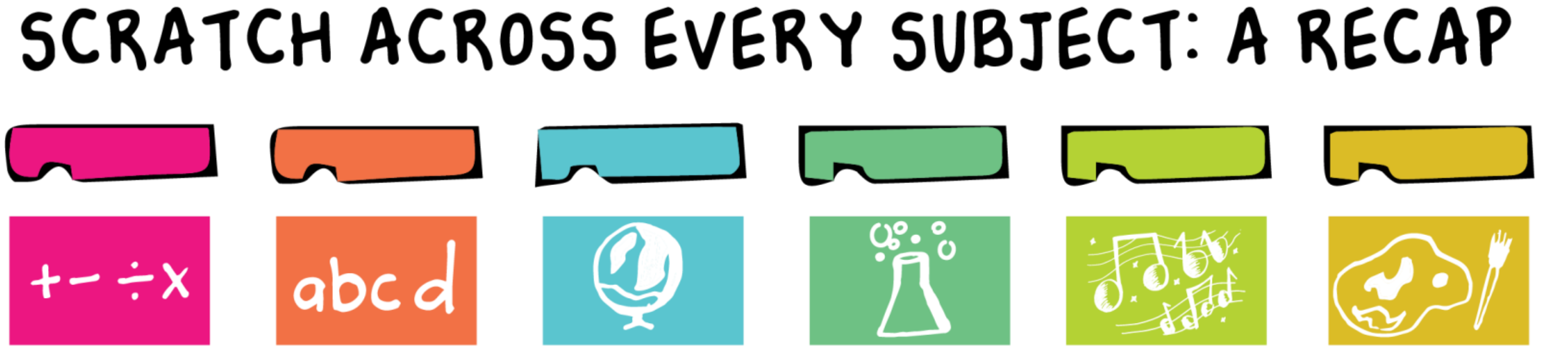


From Music to Math: Scratch Across Every Subject

Scratch is incredibly versatile–it's not just for Computer Science classes! Scratch can be incorporated into any content area in any classroom (English, Science, Math, History, Arts etc..

Scratch can contribute to your students' learning in a number of ways!

In the next slide there are some examples from our Scratch Across Every Subject series.

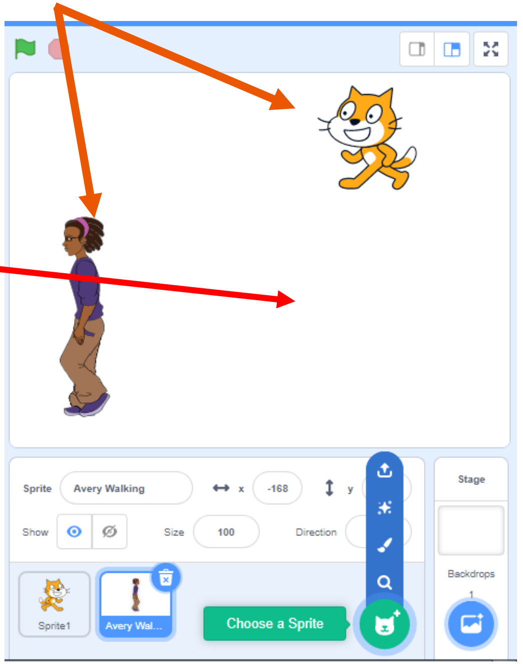


Basic Ingredients of a Scratch project

- Scratch projects are made up of a stage and objects called sprites.

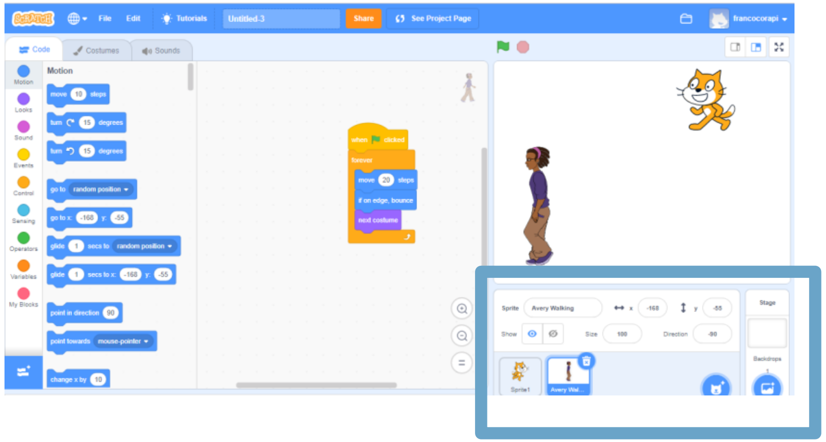
- The stage is where you see your stories, games and animations come to life.

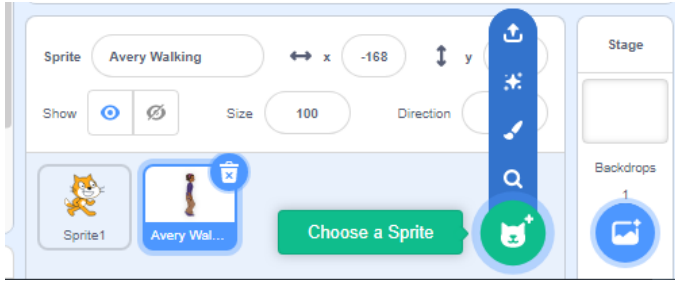
- Sprites move and interact with one another on the stage.



Sprites

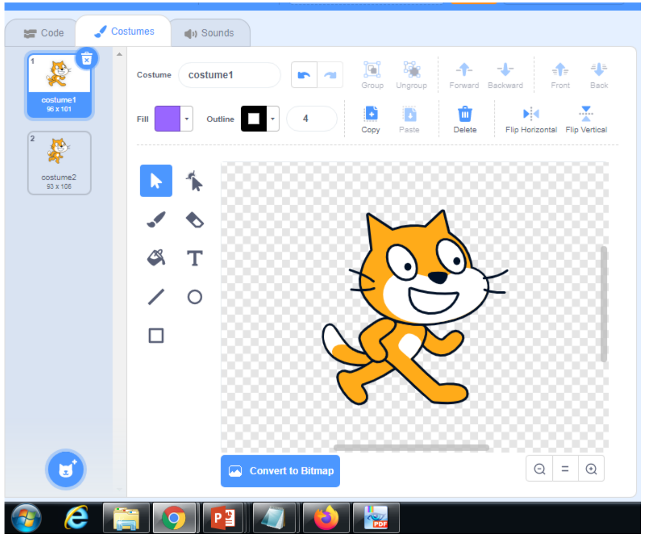
- You can create New sprites and find the sprites from the Sprite List

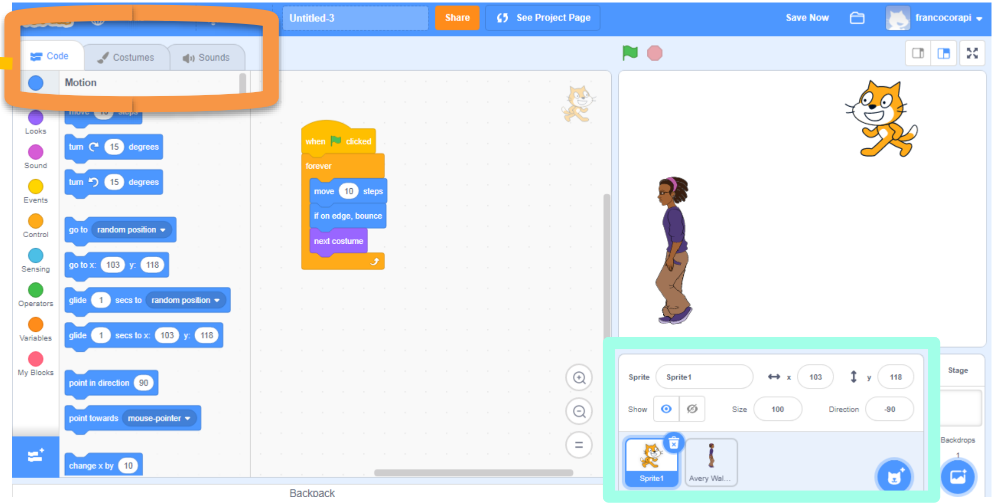




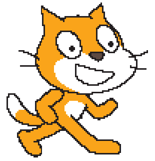
Current Sprite Info

- You can find the sprite's name, position, direction, lock state, pen color and rotation style

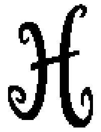




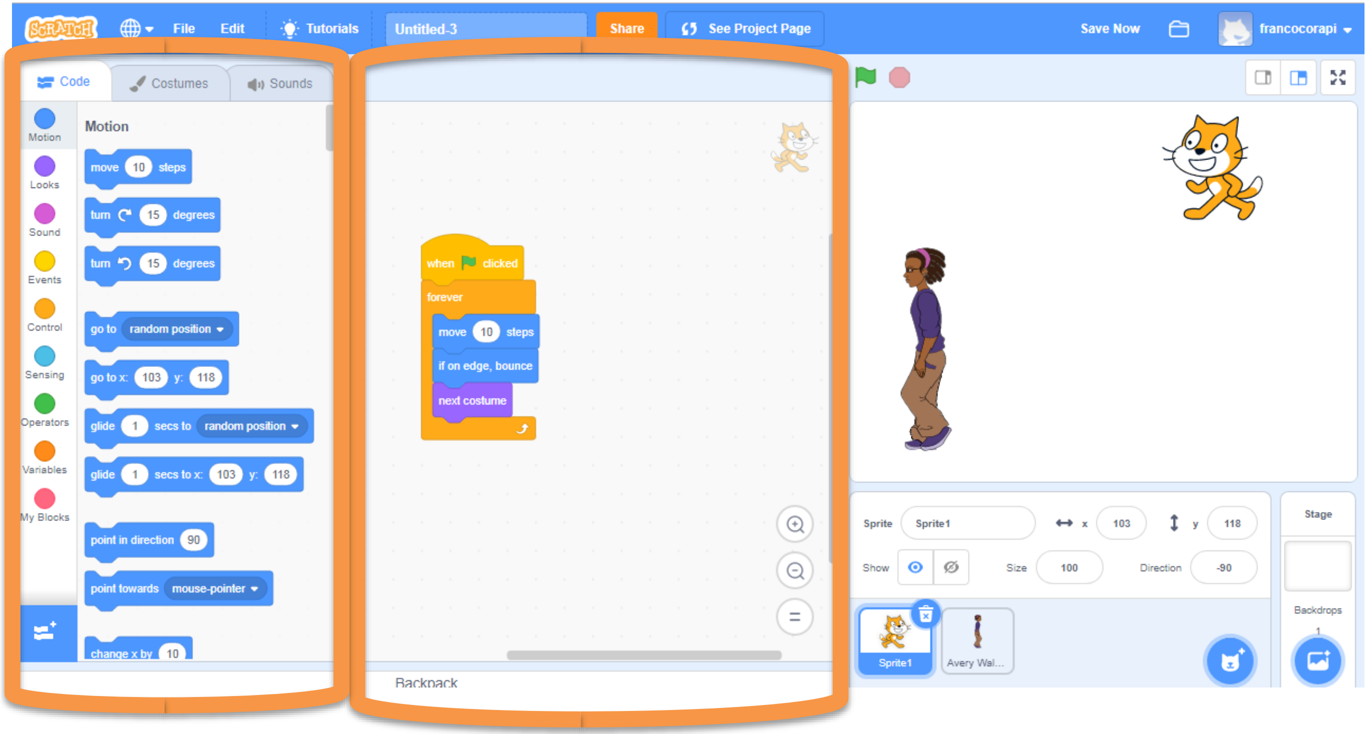
Costume



- You can change how a sprite looks like by giving it different costumes.
- You can make a sprite look like a person, a train, a butterfly or anything else.
- You can use any image as a costume.

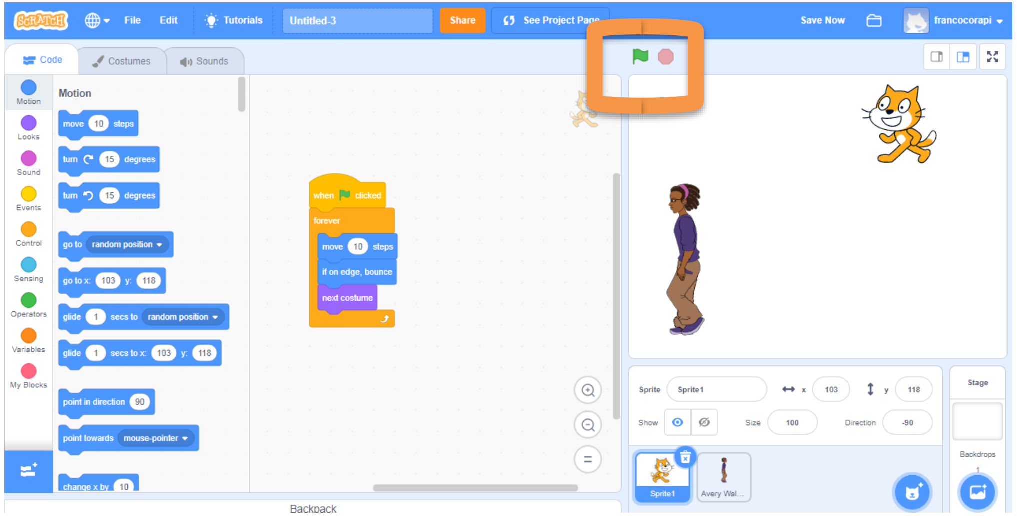


Blocks Palette and Scripts Area



Start and Stop Your Program

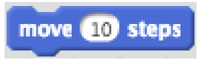
- Click the Green Flag to start all scripts that have **when 🏁 clicked** label at the top.
- Red Stop Sign stops all scripts



Exercise 1: Basic, basic script

Goal: Making Scratch sprite walk back and forth on the screen.

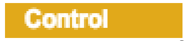
1. File → New
2. Click on

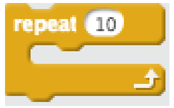
3. Drag 

over to the script area.
4. Double-click on it.

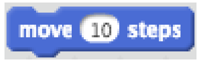
   - When you double-click on a script, it runs it once.

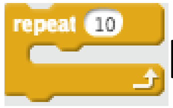
Basic, basic script (2)

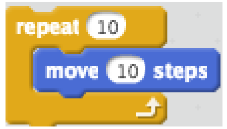
5. Click on 

, then drag 

into script area.

6. Put 

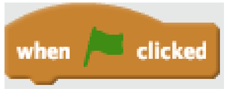
"inside" the 

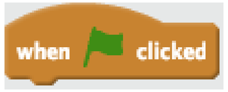
loop: 

7. Double-click on that. When cat gets to the right edge, drag it back to left edge with your mouse.

Basic, basic script (3)

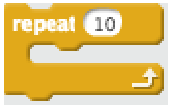
8. Click on

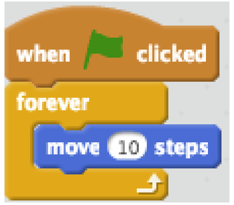
and drag 

into script area. Attach to the top of your script. (Notice that some things can only attach at the top, or bottom.)
9. Click on the above the canvas area.
10. 

is an "event handler": the script is run when the "green flag clicked event" happens.

Basic, basic script (4)

11. We want the cat to walk back and forth across the screen, changing direction when it hits the edges.
12. Detach all three instructions in your script so they are separated. Drag  back onto the area that shows the commands – this is how you delete something.
13. Do this instead ------------>



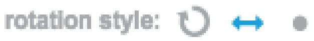
Basic, basic script (5)

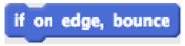
13. We need to make the cat reverse direction when it hits the edge. Click on **Motion** and find Add that to your script in the correct place.

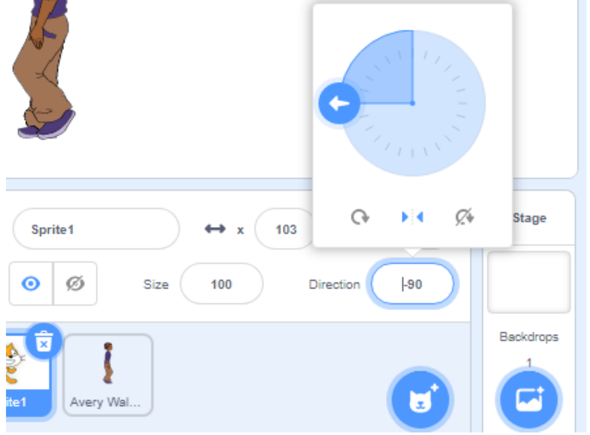
14. The cat flips upside down now.

    To fix this, click on DIRECTION in sprite section

13. Change the sprite's rotation style to this:



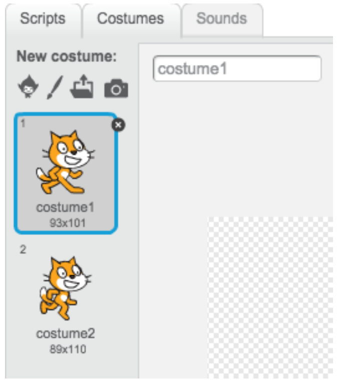



Sprite1 ↔ x 103

Size 100 Direction 90

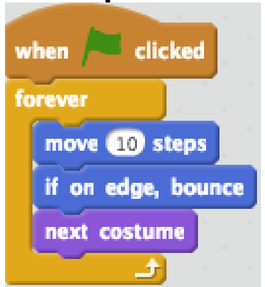
Basic, basic script (6)

16. Better… but the cat is sliding everywhere. Let's make it walk. Click on Costumes:
    Notice that it already has 2 costumes defined for it ----->.



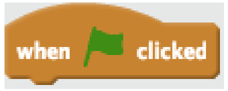
17. Click back on Scripts and click on
    In there, you'll find this:
    Add that to your script in the correct place.

18. Your script should look like this:



```
when [flag] clicked
forever
    move (10) steps
    if on edge, bounce
    next costume
```

Basic, basic script (7)

18. To demonstrate how we can do multiple scripts simultaneously, *right-click* on 

19. Insert a new sprite and drag&drop the duplicate block on new sprite
20. Try it!!

Play!

Try one or more of these:

- Click on the Stage and add multiple Backdrops, that change every 3 seconds.
- Make the sprite’s color change as it moves.
- Make music play in the background.

Exercise 2: More Control

In this lesson, we’ll learn about

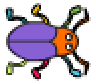
- if statements
- user interaction
- drawing with the pen

Goal: building a program that draws lines when you click on the stage.

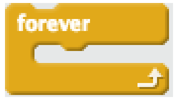
Exercise 2 (1)

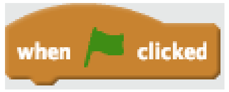
- From the main Scratch web page, click Create or, if you are already in the creation page, choose File → New.
- In the sprite section Choose the cat and click on the trash bin to delete it.
- Now, click on find sprite. You can choose anything you want, but I like the Beetle (under Animals) for this exercise. Click OK.

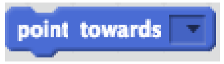
Exercise 2 (2)

- Let's make this Beetle smaller. Mine looks like this: 
- Now, think: we want the sprite to turn to face the mouse pointer at all times. We also want it to move to where the mouse is when we click. Finally, we want it to draw a line each time. So, there are 3 steps. Can you figure out how to write this script?
- Breaking a problem up into smaller problems is called *Problem Decomposition*.

Exercise 2 (3)

- First, let's make the sprite turn to face the mouse pointer. Get these onto the scripting area but don't connect them to each other: 

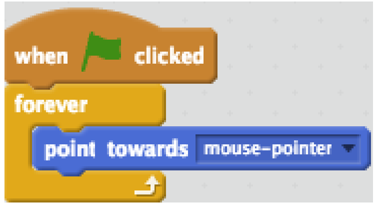
, 

, and 

.
- Click on the down arrow in "point towards" and choose "mouse-pointer". Double-click on this block now, and watch your sprite.
- Now, connect these blocks and see if you can get the sprite to follow your every move!

Exercise 2 (4)

- Your code should look like this:

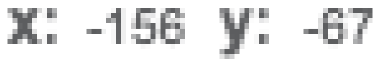


```
when [flag] clicked
forever
    point towards [mouse-pointer v]
```

- Notice that each sprite has: an (x, y) location, a direction, a rotation style, and whether it is visible or not.
  - Which direction in Scratch is northwest? Where is direction 0?

Exercise 2 (5)

- Click on your sprite and move this on the stage
- Look above the sprite area and below the stage and you'll see something like this:

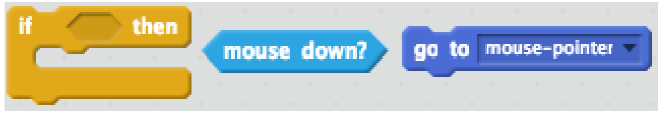


x: -156 y: -67

- What are these numbers?
- Watching these numbers, figure out how big the canvas is:
  - What is the smallest x value? ____ Largest? ____
  - What is the smallest y value? ____ Largest? ____
- Now, drag your sprite to (0, 0). What part of the sprite is exactly at (0, 0)?

Exercise 2 (6)

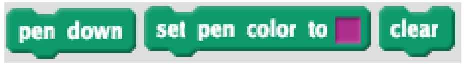
- Back to the exercise: now we want the sprite to move to wherever we click. For this step, we'll need:



- (If you have problems finding these, their colors should help you figure out what group they are in.)
- Put them together and then put the whole thing into your forever loop.

Exercise 2 (7)

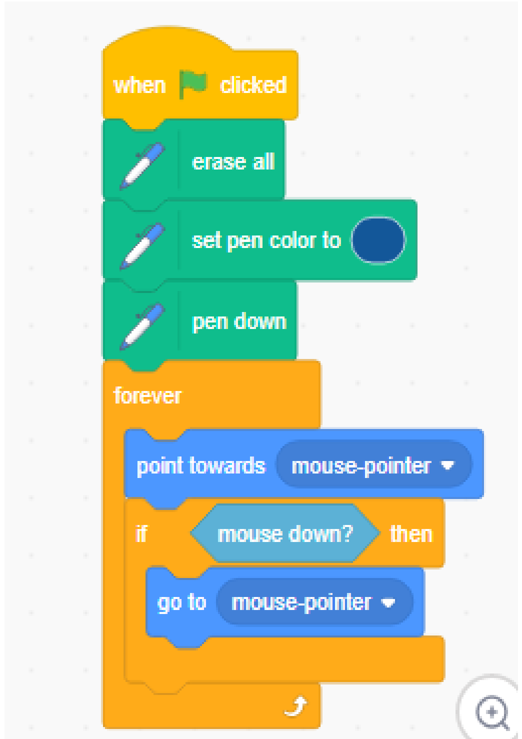
- Final step: make the sprite draw its path whenever it moves. For this, you'll need these:



- To add this block, click in botton on *add extension* and choose pen
- Figure out the order of these and where to attach them in the script. When you are done, you should be able to:
  - draw whenever you click.
  - stop the script and then when you start, it should be on a clear canvas.
- What happens if you keep the mouse button down while you move it around?

Exercise 2 (8)

- My solution looks like this:



```
when 🏁 clicked
erase all
set pen color to ●
pen down
forever
    point towards mouse-pointer ▾
    if mouse down? then
        go to mouse-pointer ▾
```

Notice a few things:

- Under , there is code to *initialize* the setup.
- Then there is the with code in it.
- This organization is very common.

Play!

Try one or more of these:

- Make the color of the pen change (constantly, or whenever you click).
- Set or change the width of the pen.
- Make the sprite always start at (0, 0) when you start.
- Make the sprite “glide” to where you have clicked, instead of “zipping” right there.
  - Note that there is no “glide toward” command…

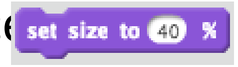
Exercise 3: More interaction

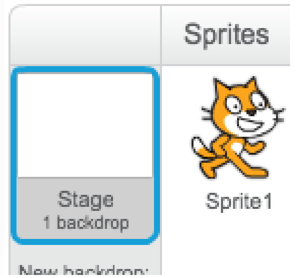
In this exercise we'll learn about:

- creating custom backdrops
- if-else statements
- sensing colors
- reacting to keyboard "events".

Goal: mak a program that lets you navigate the cat through a maze.

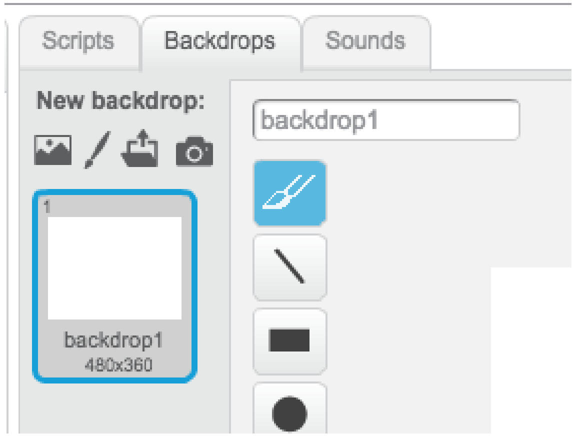
Exercise 3 (1)

- Shrink your cat down to a pretty small size. I made mine 40% of initial size. (You can do this quickly by dragging this  to your script area and double-clicking on it once
- Drag your cat to the lower left corner of the screen.
- Select the Stage to the left of your cat Sprite:

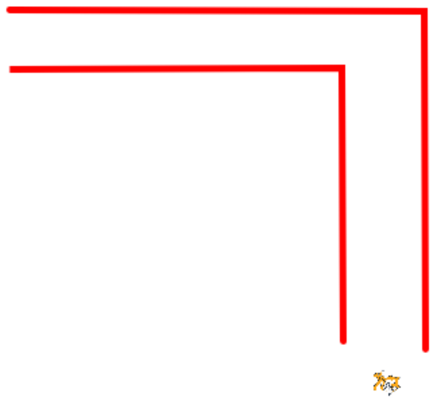


Exercise 3 (2)

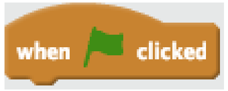
- Now, click on the Backdrops tab:
- Click on the Line tool.
- Choose a color. I chose red.
- Create a path for your cat to have to negotiate through. Make sure the hallways are wide enough for your cat to fit through.

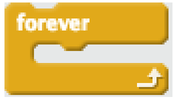


My path



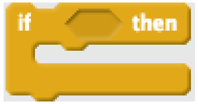
Exercise 3 (4)

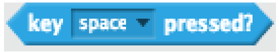
- Click on the cat again, and get into the Script writing "mode".
- Put a 

block and 

block in and connect them.
- We need to check if the user pressed the up-arrow key, and if so, the cat moves up. See if you can figure out how to do this.

Exercise 3 (5)

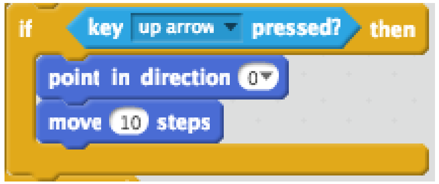
- You need the 

. This is a *conditional* block: if the "condition" is true, then the code inside it runs. Otherwise, it doesn't.
- The condition we need to check is the up-arrow key being pressed: 

- Now, if that condition is true, we want to make the cat face up and then move 10 steps.
- See if you can figure this out.

Exercise 3 (6)

- My code now looks like this (inside the forever loop):

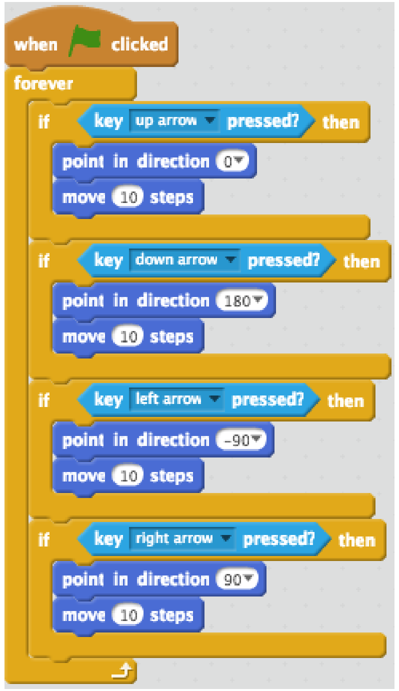


```
if <key [up arrow] pressed?> then
    point in direction (0)
    move (10) steps
```

- Try it!
- Right-click on the and duplicate the code. Fix those replacements so that they handle the down, left, and right arrow keys.

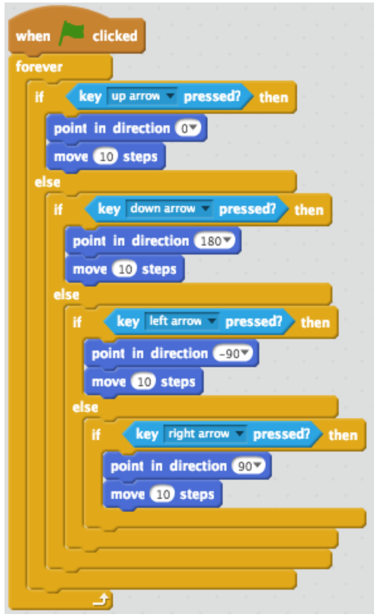
Exercise 3 (7)

My code looks like:



```
when [flag] clicked
forever
    if key up arrow pressed? then
        point in direction 0
        move 10 steps
    if key down arrow pressed? then
        point in direction 180
        move 10 steps
    if key left arrow pressed? then
        point in direction -90
        move 10 steps
    if key right arrow pressed? then
        point in direction 90
        move 10 steps
```

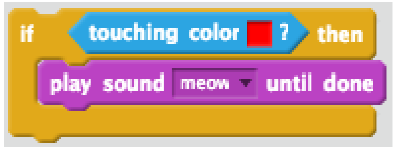
Alternatively we could use if-else blocks:



```
when [flag] clicked
forever
    if key up arrow pressed? then
        point in direction 0
        move 10 steps
    else
        if key down arrow pressed? then
            point in direction 180
            move 10 steps
        else
            if key left arrow pressed? then
                point in direction -90
                move 10 steps
            else
                if key right arrow pressed? then
                    point in direction 90
                    move 10 steps
```

Exercise 3 (9)

- Look in the Sensing area. How can we tell if the cat is touching a wall?
- What should happen when the cat touches a wall?
- Here is my code:
- Where should this code be put?



```
if <touching color 🟥 ?> then
    play sound meow until done
```

- Perhaps the cat should also say "Ouch!" **Try it!**
- Each time you restart your game, you cat should be moved to the start position.
- Add code to do this.

Exercise 4: Quiz

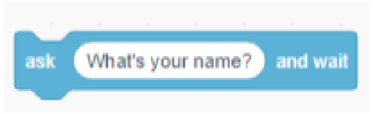
In this lesson, we’ll learn about…

- creating custom quiz.
- getting user input
- using variables.

Goal of the project: creating interactive quiz.

Exercise 4 (1)

- Create a new project
- Add a “When Flag Clicked” block that can be found in the “Events” tab.
- Add some kind of greeting. The “Say” block can be found in the “Looks” tab.
- This is where things start to get a little complicated! But don't worry! It's all very easy if you take it one at a time.
  First you need to go into the “Sensing” tab and grab this block:

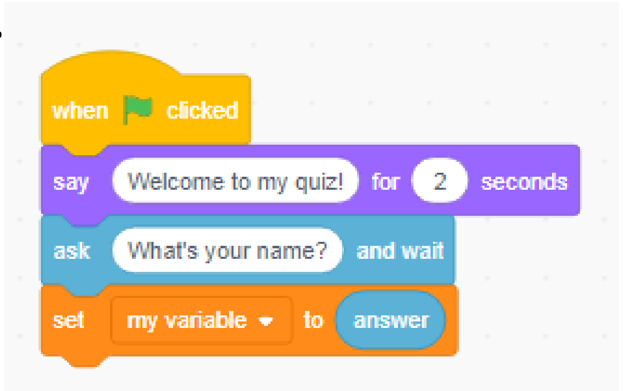


Exercise 4 (2)

We're going to use this and a name variable to save the person's name. So, next go over to the “Data” tab and click “Make a Variable”. Call it “name” and make sure the “for all sprites” button has been clicked. Once you've made it, unclick the tick beside the variable in the “Data” tab. Then get the “set name to _” and add it to your script.

Finally, to save the name we need to replace the 0 with an “answer block which can be found in the ”Sensing" tab.

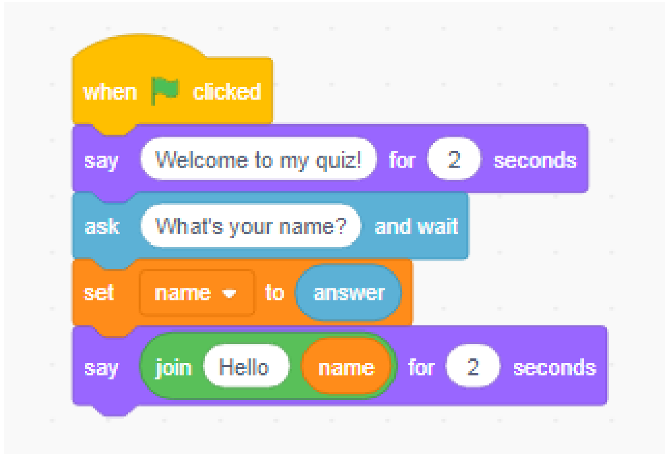
By now it should look like this:



```
when clicked
say Welcome to my quiz! for 2 seconds
ask What's your name? and wait
set my variable to answer
```

Exercise 4 (3)

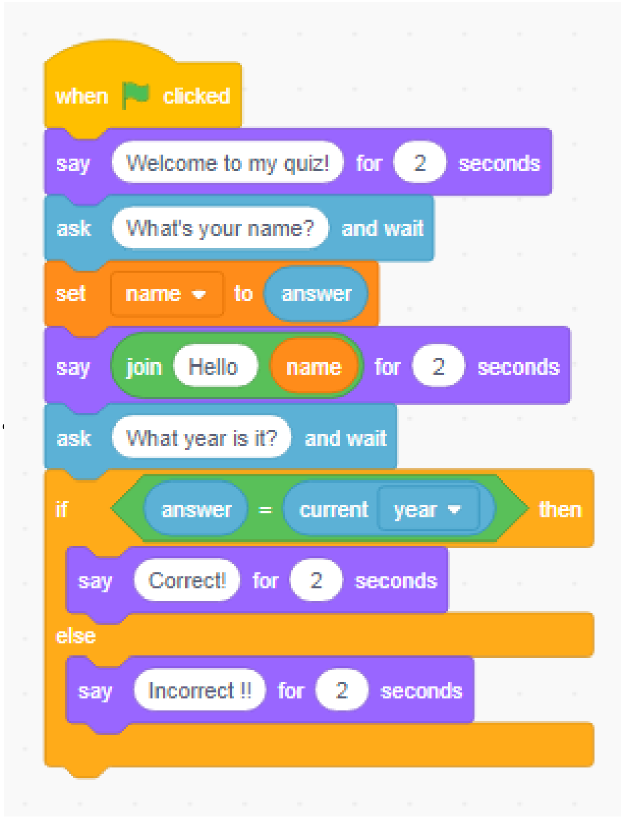
- Using The Name
  To use the name, we need to have our sprite say it. So, get another "say" block from the "Looks" tab. Then go to the "Operators" tab and get a "join" block. Put it into the white space beside "say". It should look like this:



```
when 🏳 clicked
say (Welcome to my quiz!) for (2) seconds
ask (What's your name?) and wait
set [name ▾] to (answer)
say (join (Hello) (name)) for (2) seconds
```

Exercise 4 (5)

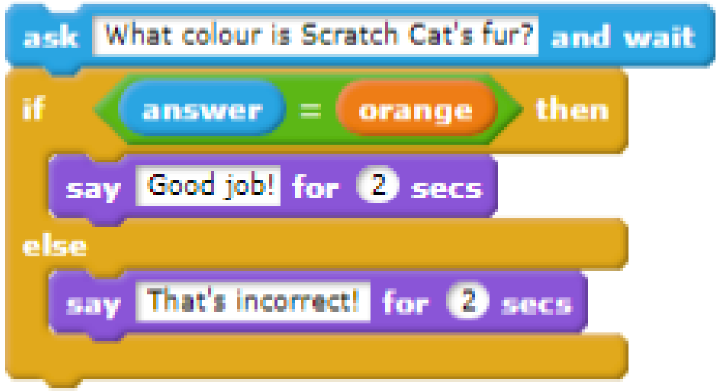
- Now that we know how to save and record a name, let's start asking some questions. To add a question, go back to the "Sensing" tab and get another "ask ___ and wait" block. Our first question is going to be "What year is it?"
- Now, to see if our answer is correct, we need to get an "if else" block which can be found in the "Control" tab.
- Then, in the space beside if, add a "_=_" block.
- Then get an "answer" and a "current _____" block from the "Sensing" tab.
- Drag them into the spaces beside the equals sign. Select year in the dropdown menu beside "current". You should now have this:



```
when 🏴 clicked
say (Welcome to my quiz!) for (2) seconds
ask (What's your name?) and wait
set [name ▾] to (answer)
say (join (Hello) (name)) for (2) seconds
ask (What year is it?) and wait
if <(answer) = (current [year ▾])> then
    say (Correct!) for (2) seconds
else
    say (Incorrect !!) for (2) seconds
```

Exercise 4 (4)

Great! Now add some more questions! Do the same things you did in the previous steps



```
ask [What colour is Scratch Cat's fur?] and wait
if <(answer) = (orange)> then
    say [Good job!] for (2) secs
else
    say [That's incorrect!] for (2) secs
```

Exercise 4 (5)

PLAY !!!

- Add score for each question
- Add feedback based on the final score

Thanks for watching

Prof. Franco Corapi

“Giovanni Malafarina” Technic Tecnological Institute

Co-funded by the
Erasmus+ Programme
of the European Union

Giovanni Malafarina
Istituto Tecnico Tecnologico