

Hedy : A gradual

Programming Language

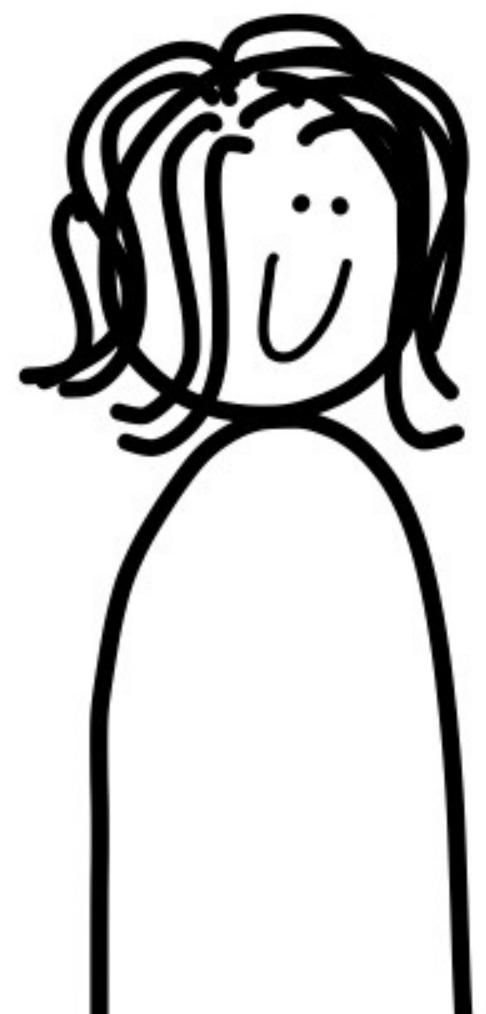


Felienne Hermans

Hedy is gradual

multi lingual

built for teaching



Felienne Hermans

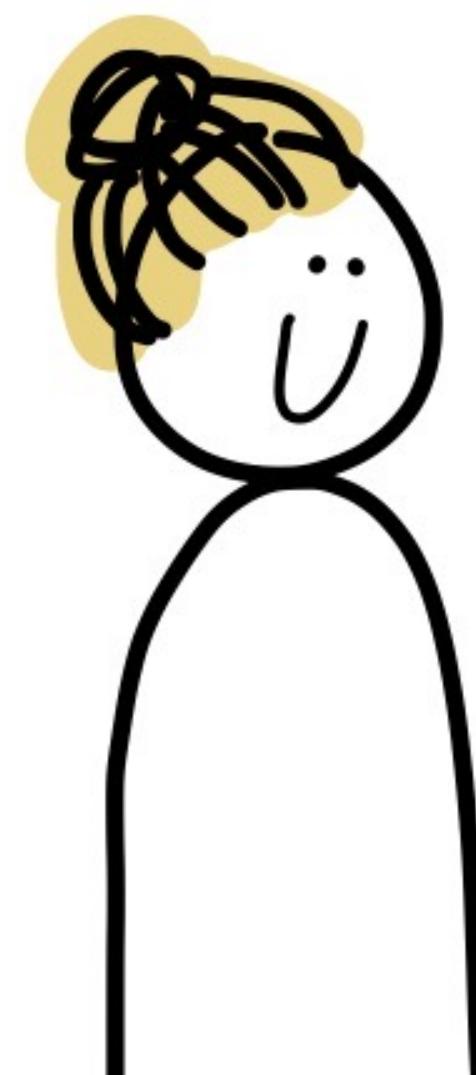
How to teach programming?



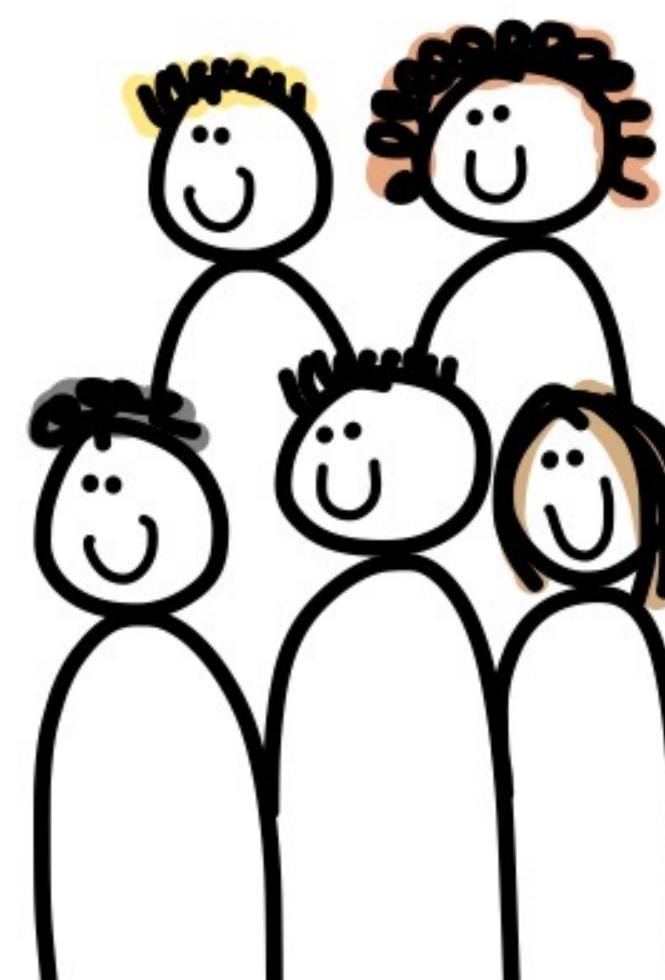
2013

middle schoolers

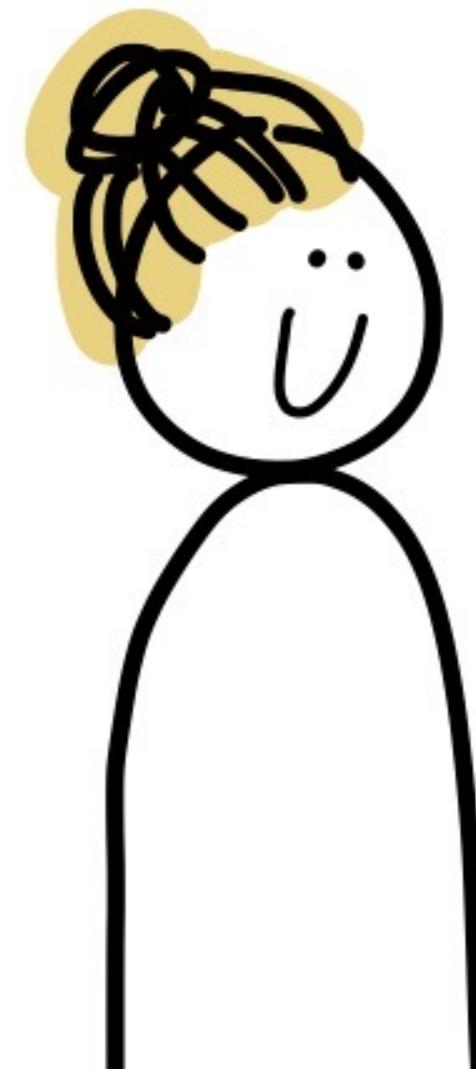
age 10 - 14



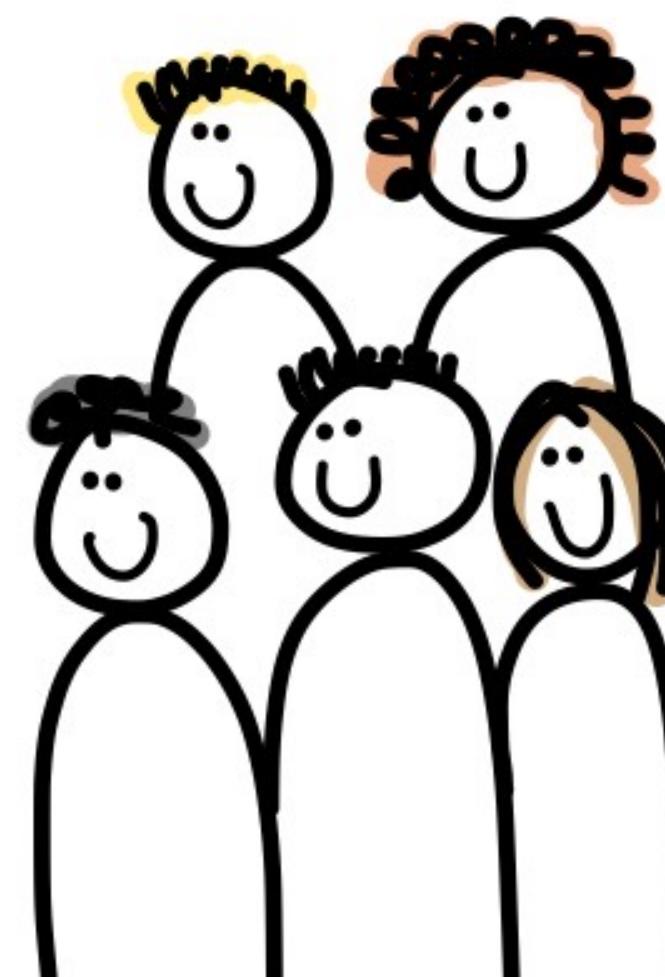
2013



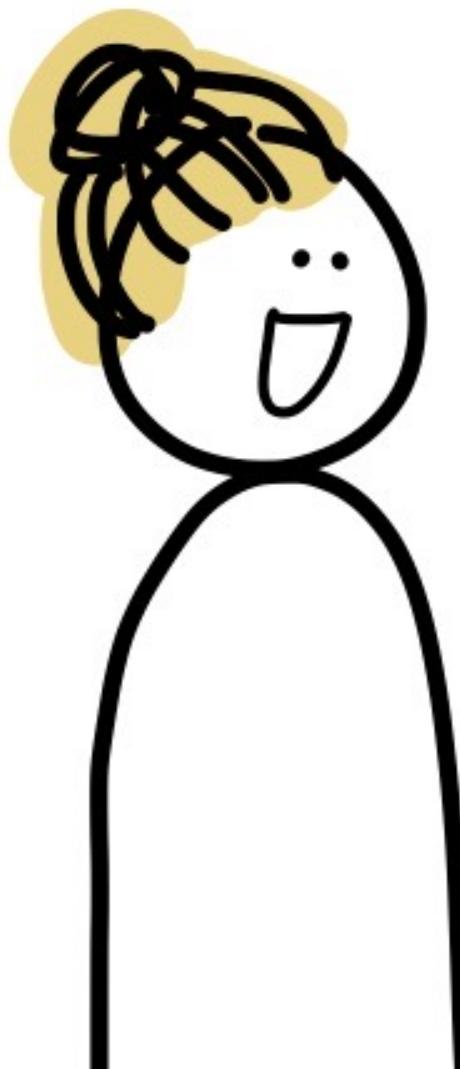
We'd love to learn
programming



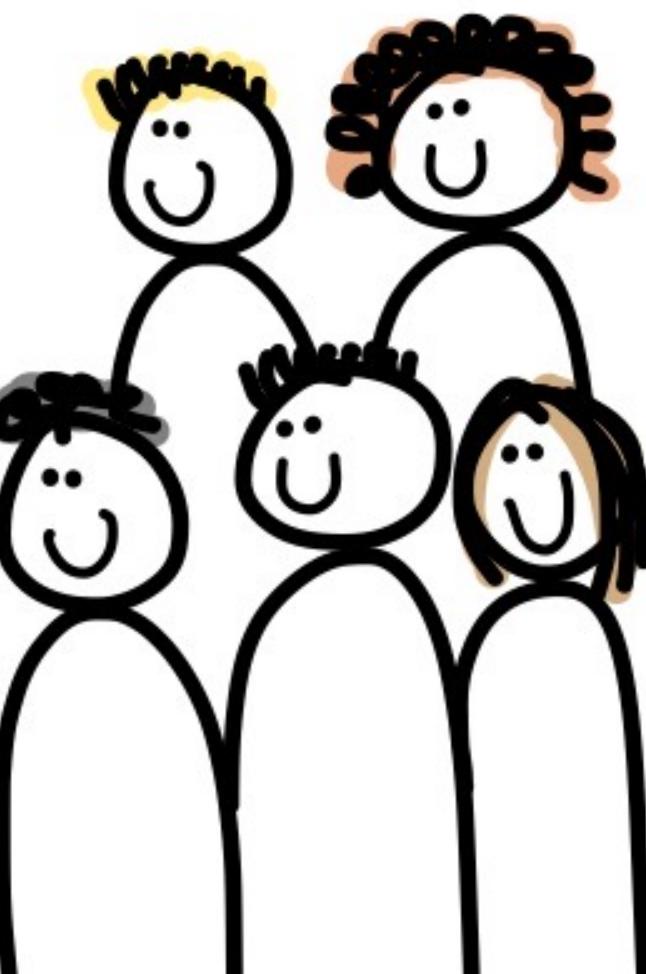
2013



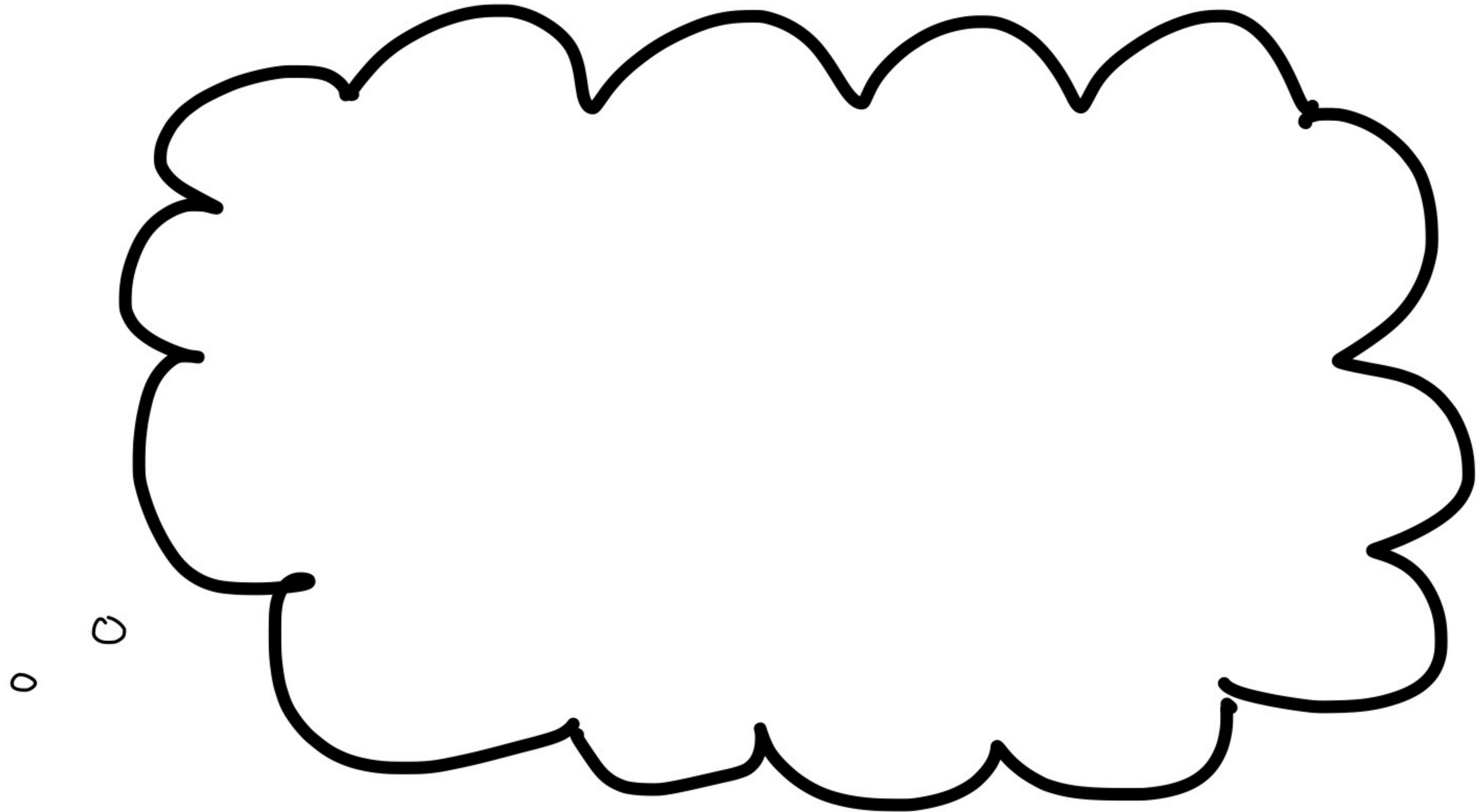
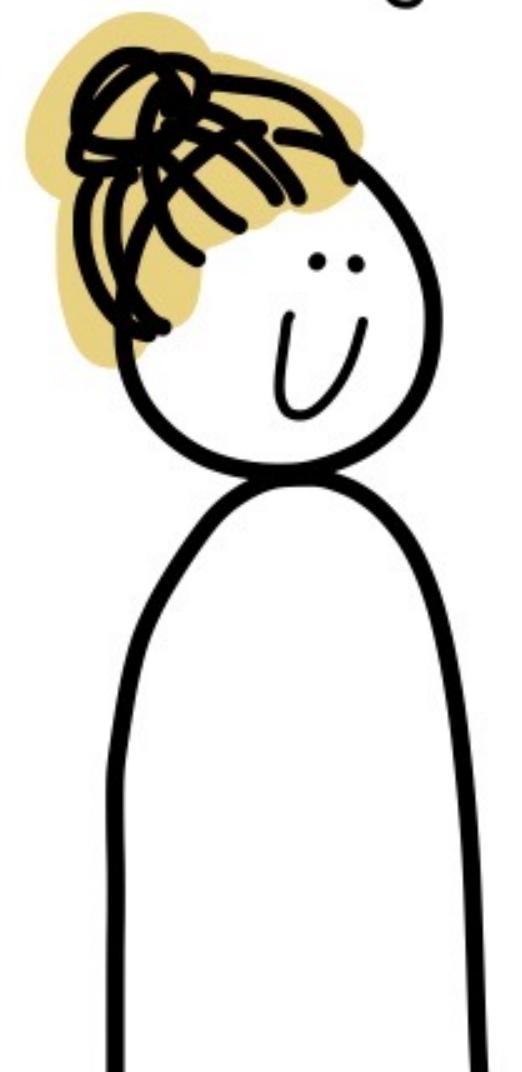
I'll teach you

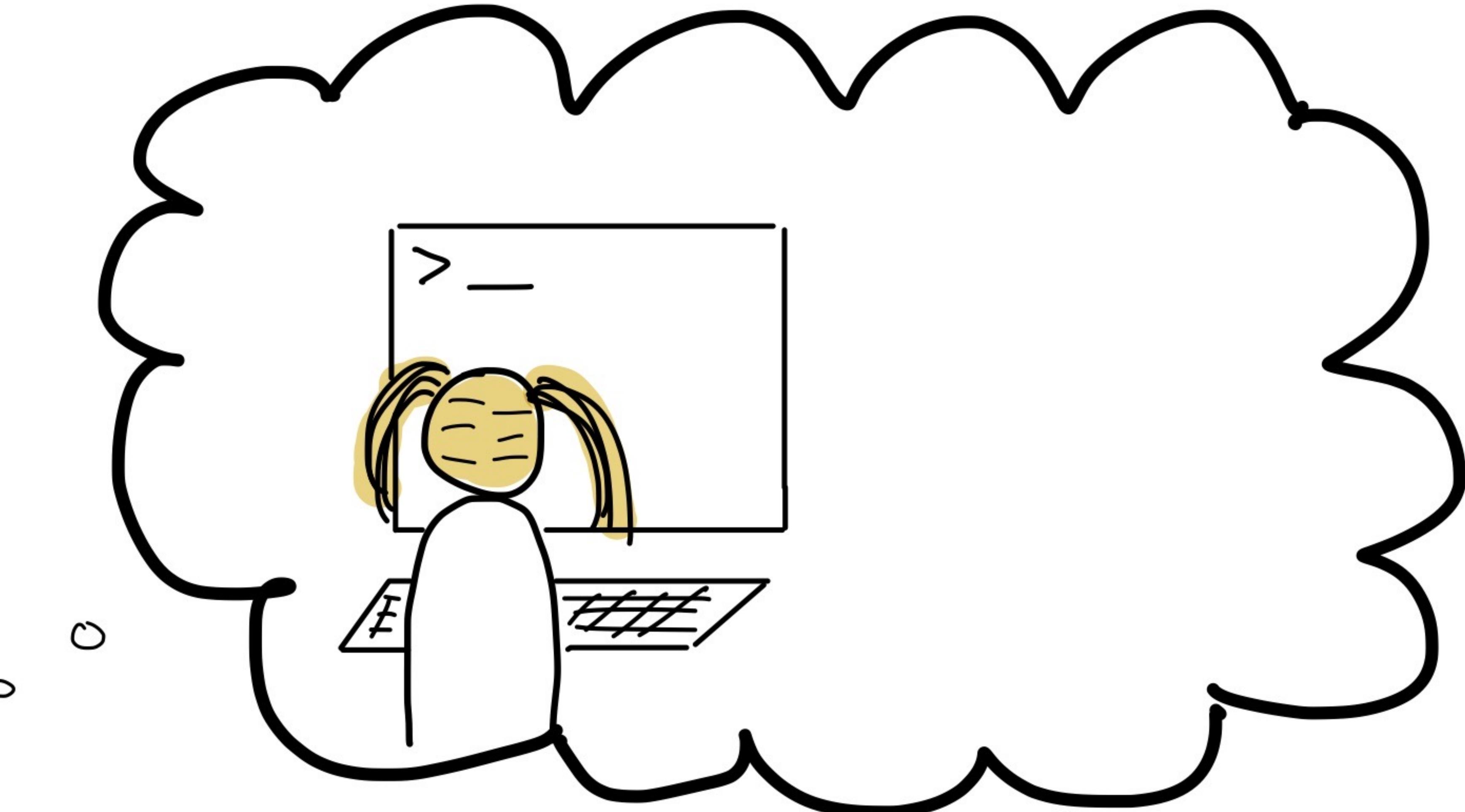
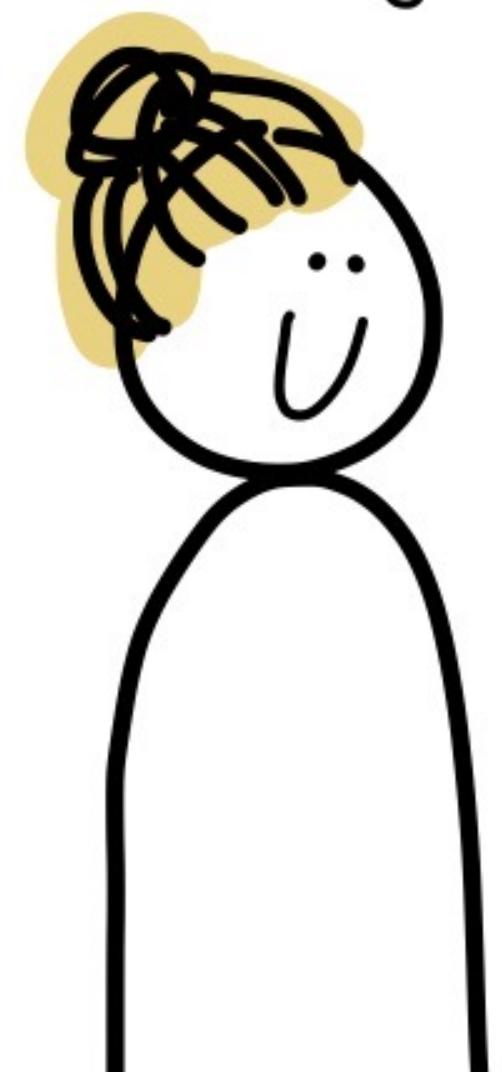


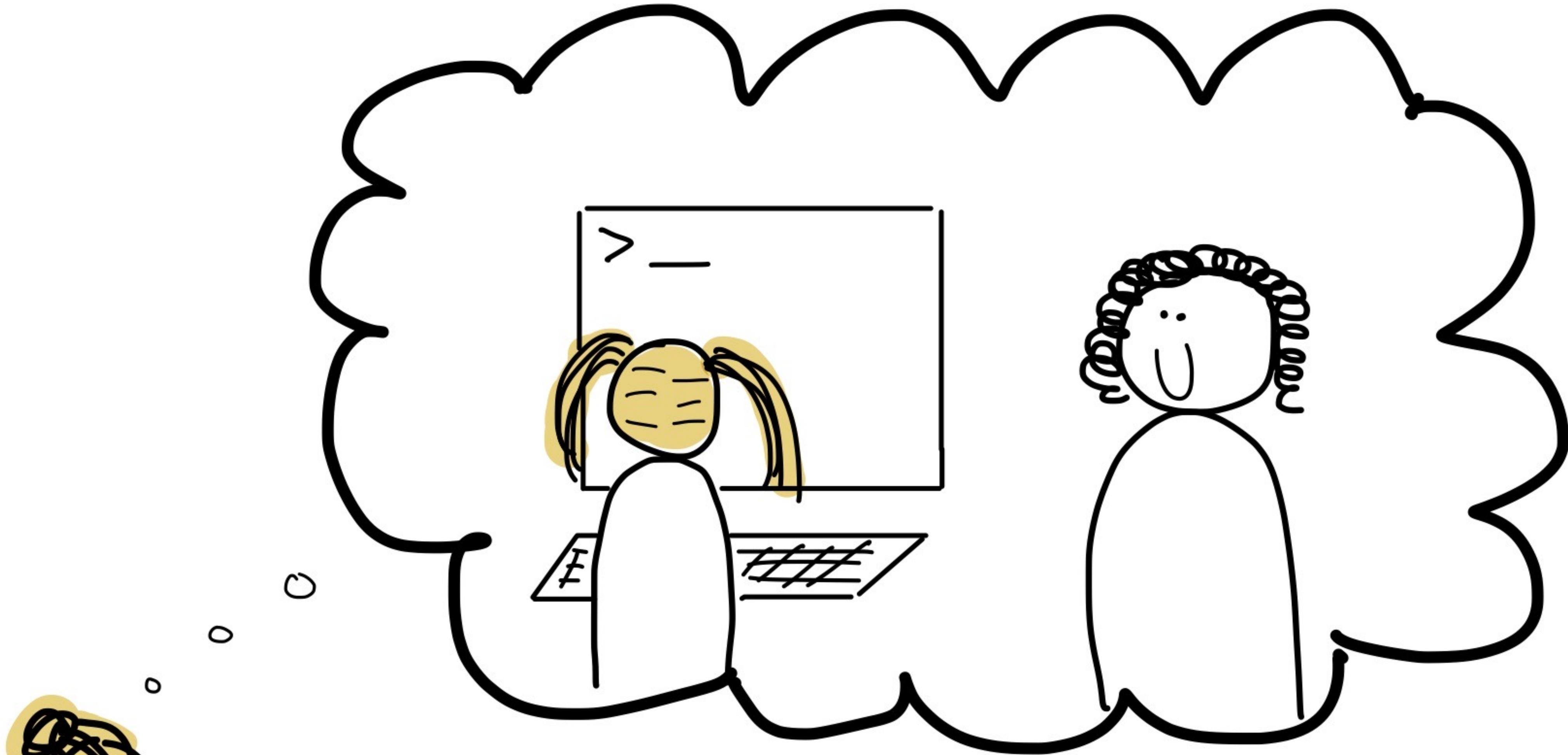
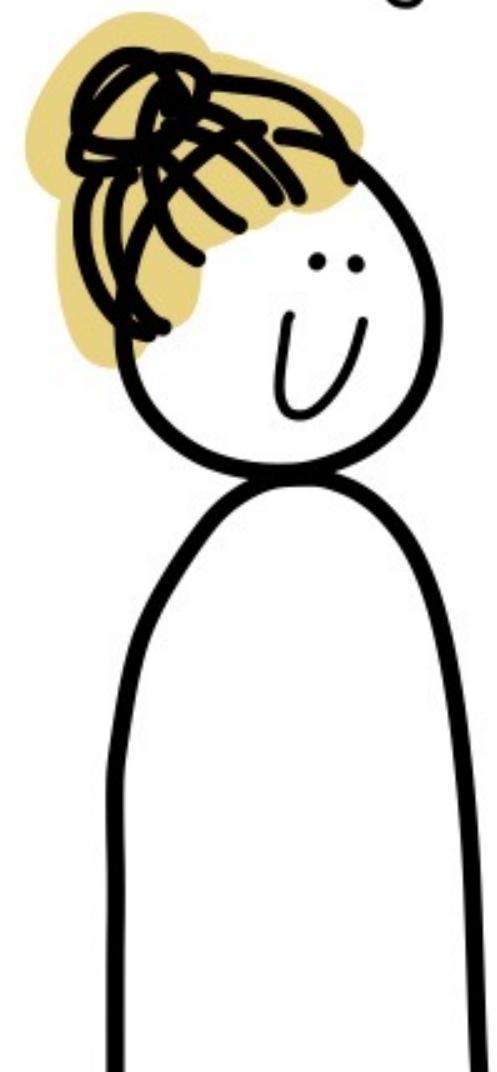
We'd love to learn
programming

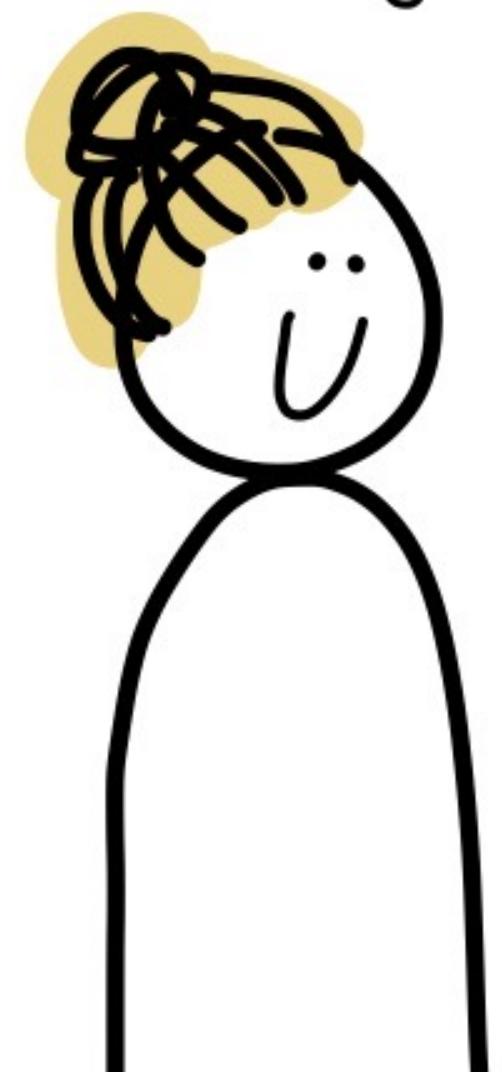


2013

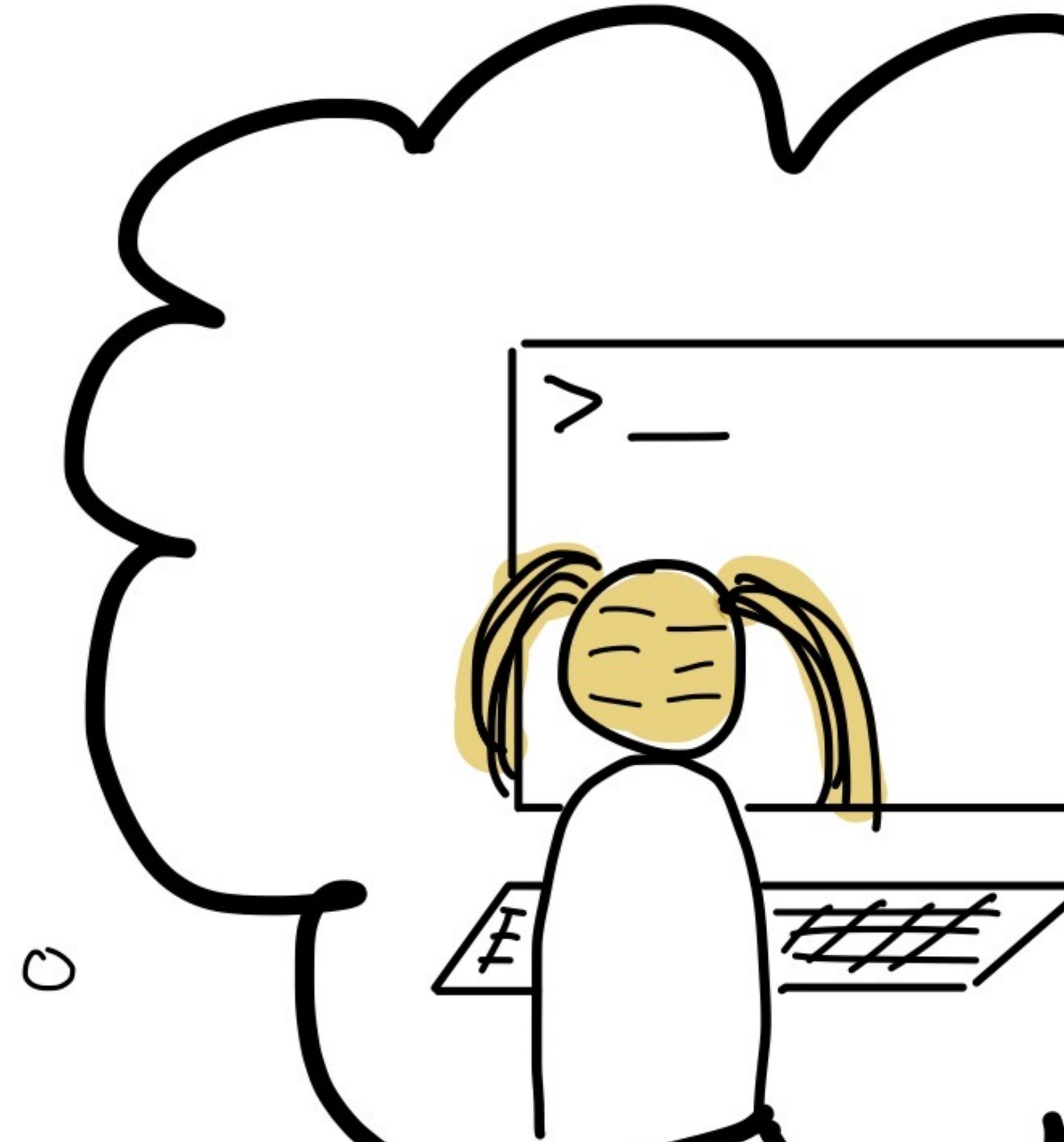
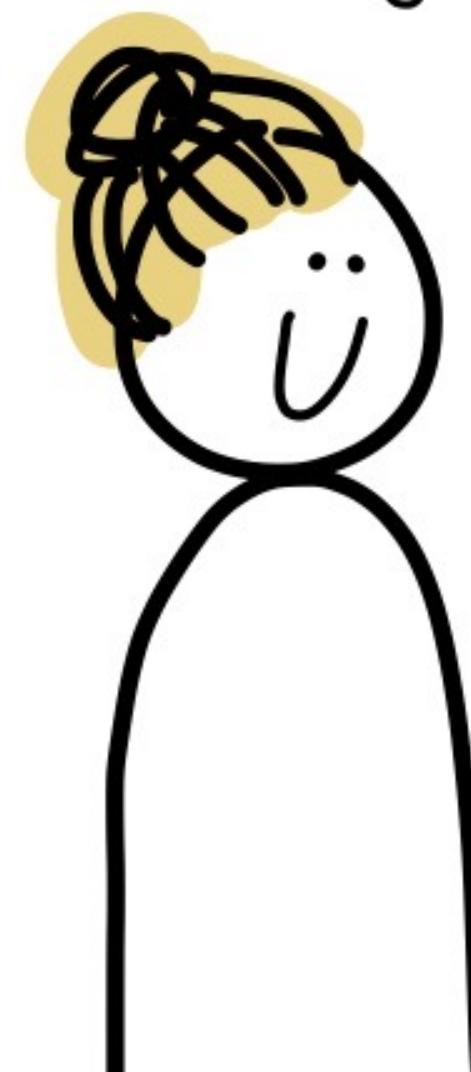




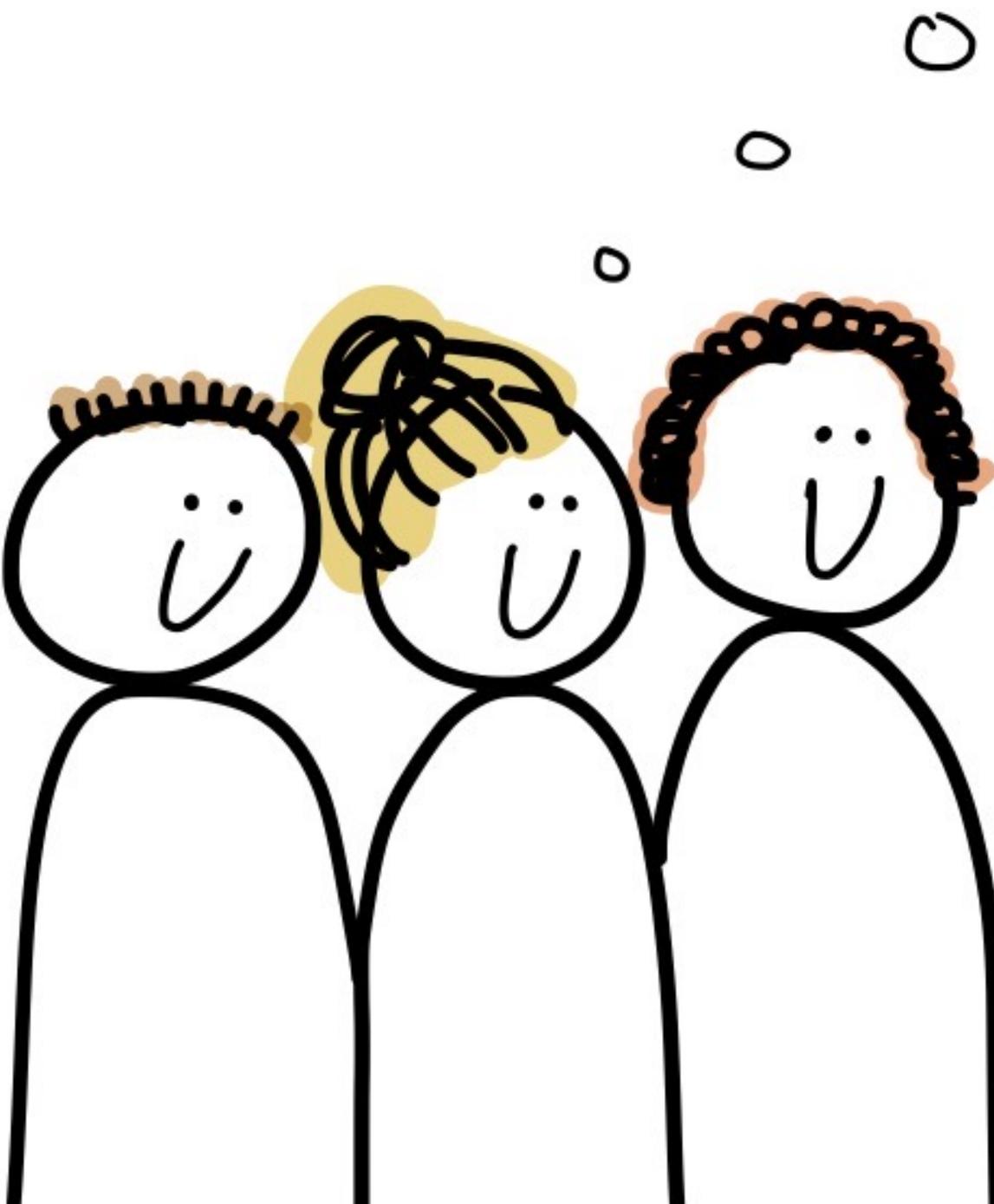




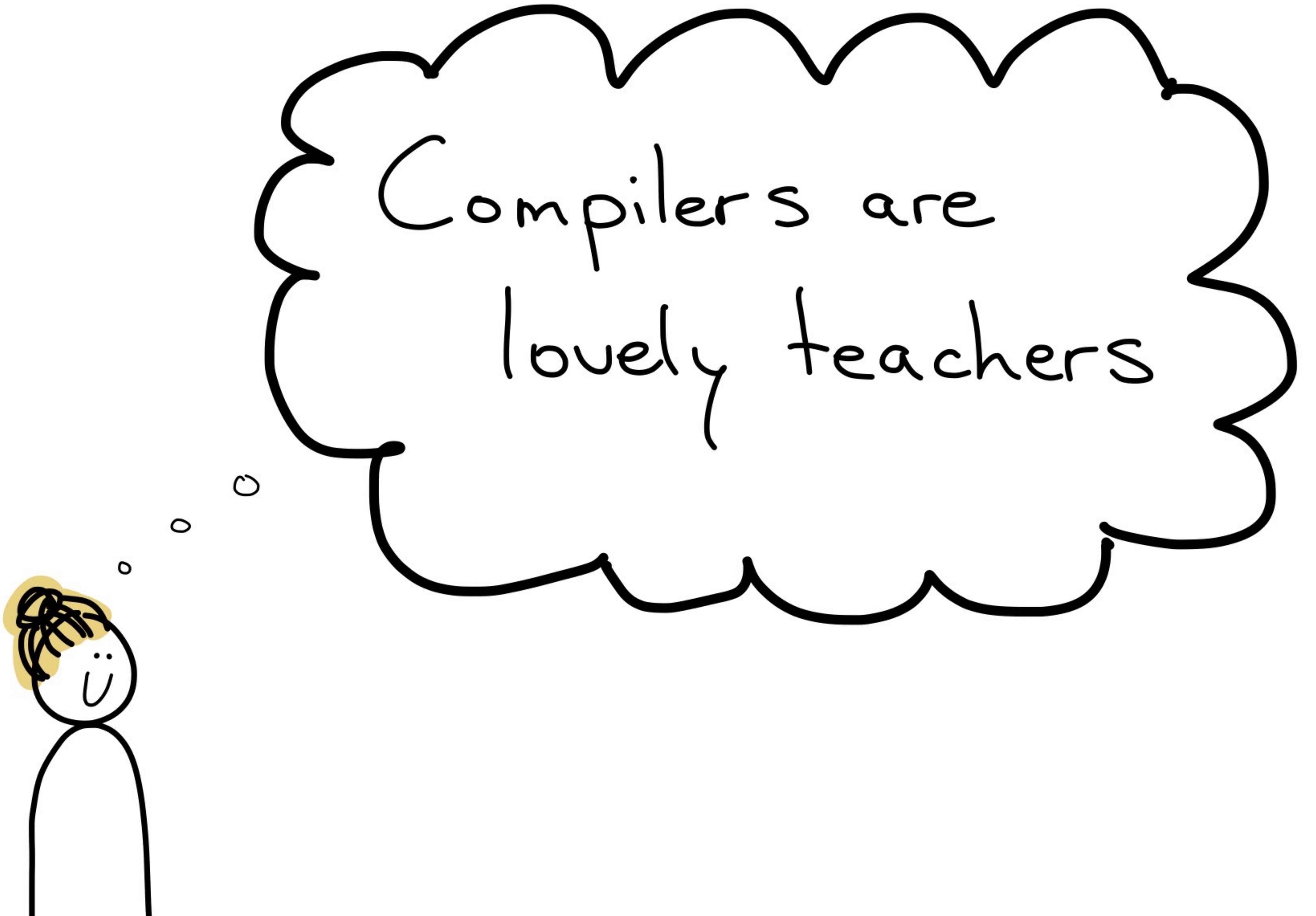




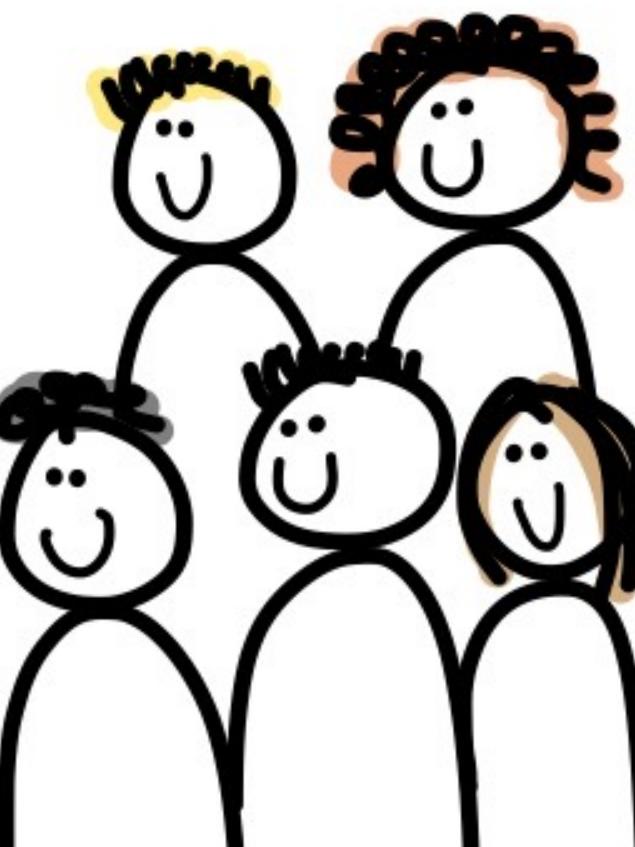
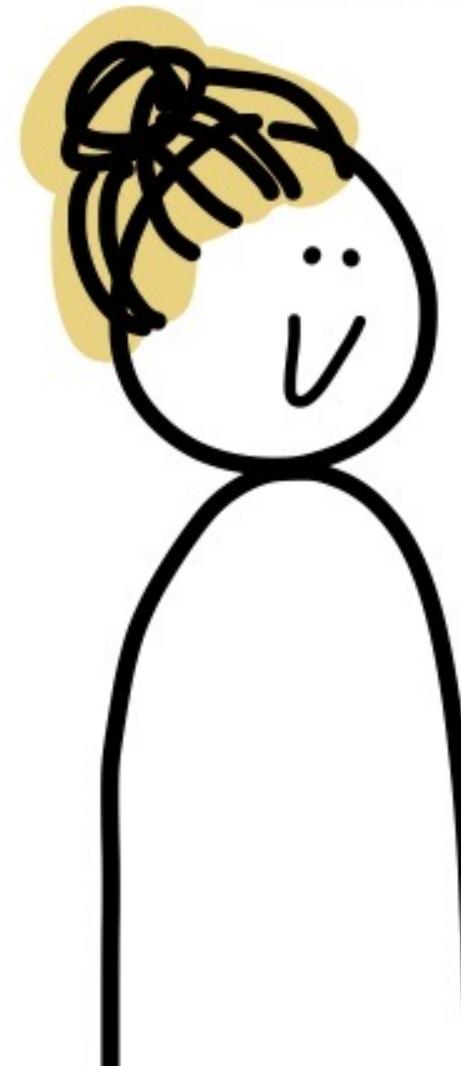
```
135 INPUT "TIME INCREMENT (SEC)";S2
140 PRINT
150 INPUT "VELOCITY (FPS)";V
160 PRINT
170 INPUT "COEFFICIENT";C
180 PRINT
182 PRINT "FEET"
184 PRINT
186 S1=INT(70/(V/(16*S2)))
190 FOR I=1 TO S1
200 T(I)=V*C^(I-1)/16
210 NEXT I
220 FOR H=INT(-16*(V/32)^2+V^2/32+.5) TO 0 STEP -.5
221 IF INT(H)<>H THEN 225
222 PRINT H;
225 L=0
230 FOR I=1 TO S1
240 FOR T=0 TO T(I) STEP S2
245 L=L+S2
250 IF ABS(H-.5*(-32)*T^2+V*C^(I-1)*T)>.25 THEN 270
260 PRINT TAB(L/S2);"0";
270 NEXT T
275 T=T(I+1)/2
276 IF -16*T^2+V*C^(I-1)*T<H THEN 290
280 NEXT I
290 PRINT
300 NEXT H
310 PRINT TAB(1);
320 FOR I=1 TO INT(L+1)/S2+1
330 PRINT ".";
340 NEXT I
350 PRINT
355 PRINT "0";
360 FOR I=1 TO INT(L+.9995)
380 PRINT TAB(INT(I/S2));I;
390 NEXT I
```

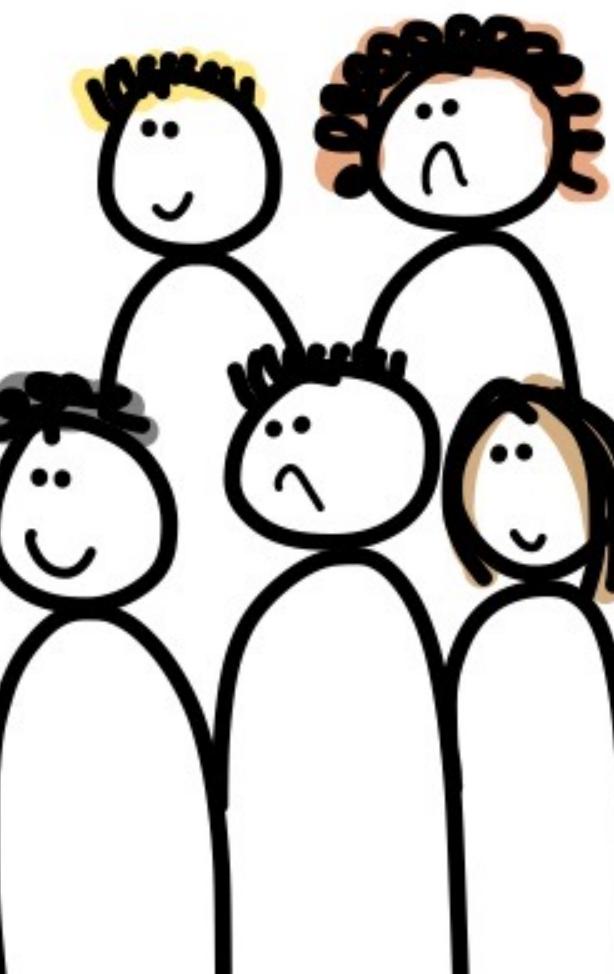


Compilers are
lovely teachers

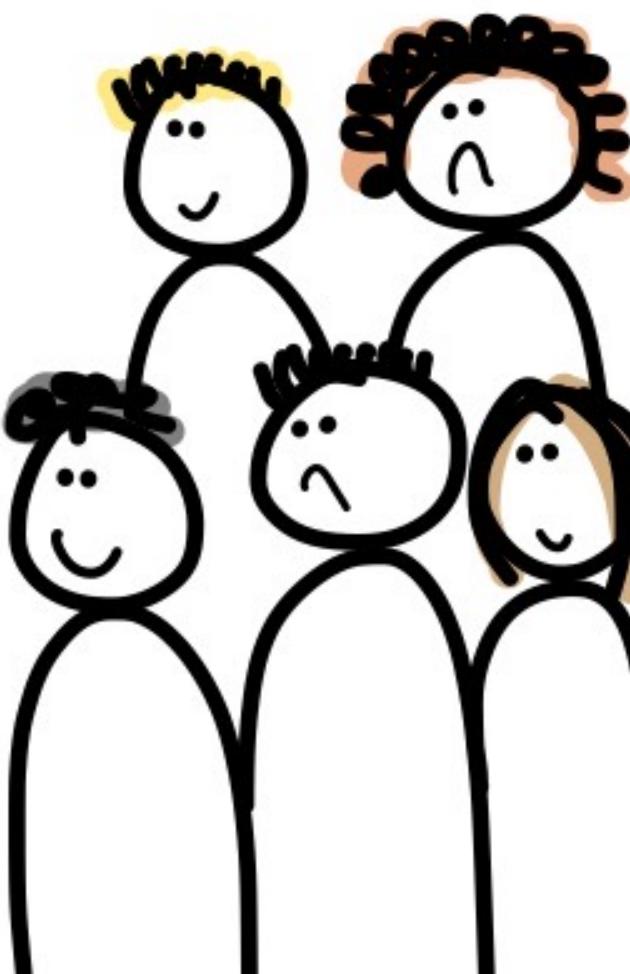
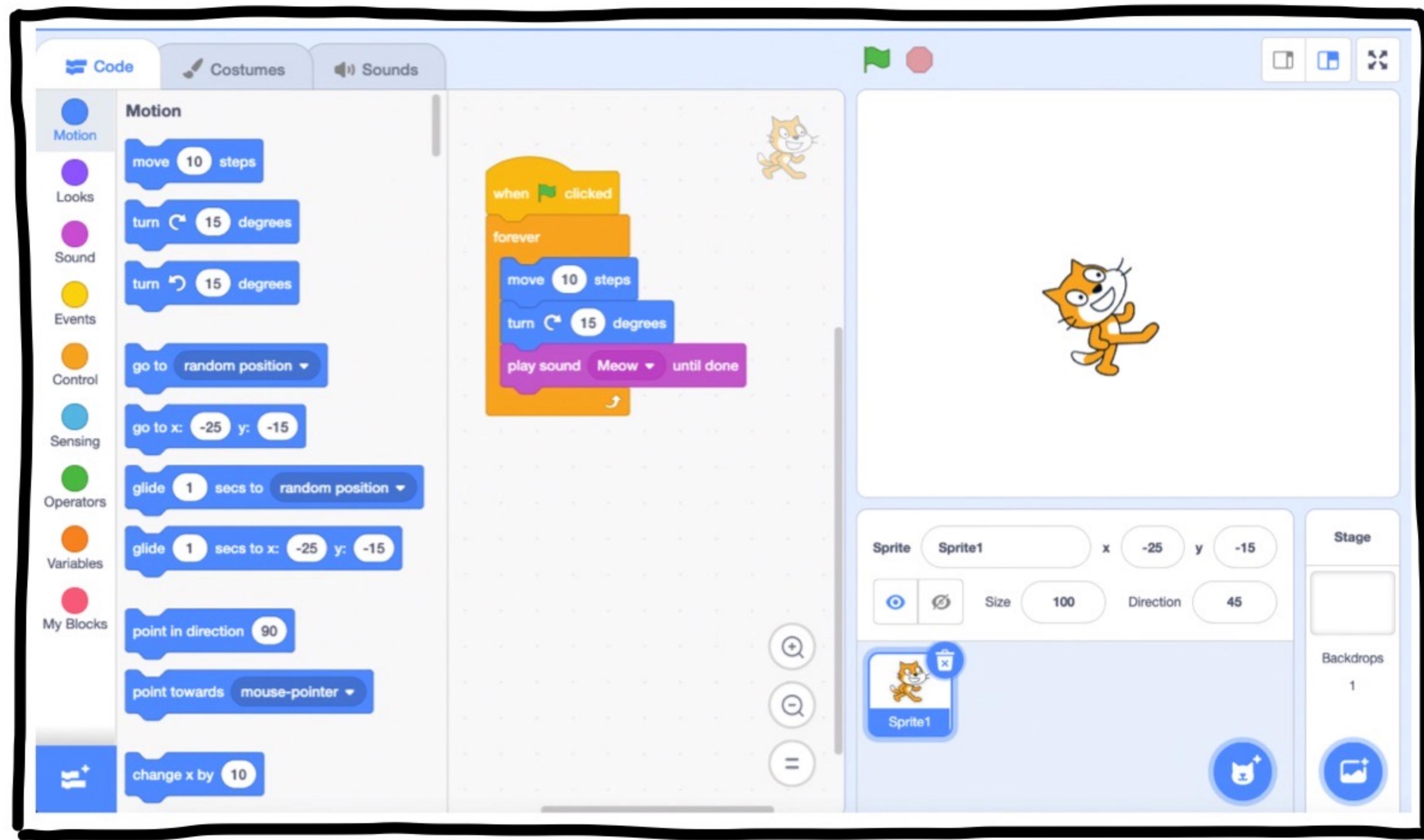


Compilers are
lovely teachers





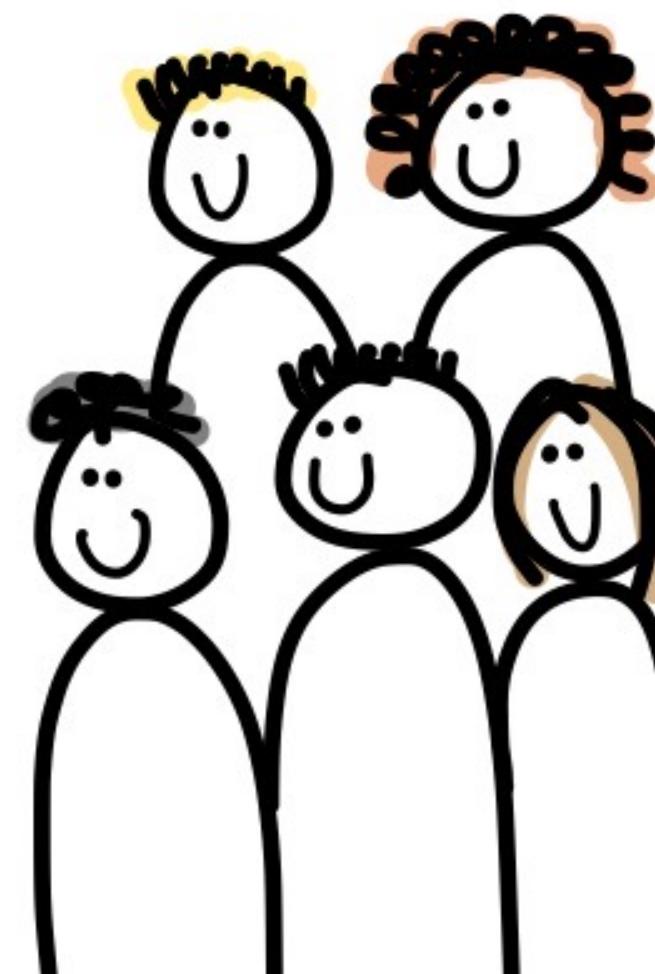
Scratch
is
for kids



A screenshot of a Python code editor interface. The top bar shows the user 'mevrHermans / HedyTester'. The left sidebar has a 'Files' tab selected, showing a folder icon and a file icon labeled 'main.py'. The main area displays the file 'main.py' with the following code:

```
1 print("Hello everyone!")
```

The right side of the interface includes tabs for 'Console' and 'Shell', with the 'Console' tab currently active. The console window is empty, indicated by a dark blue background and a small 'x' button.

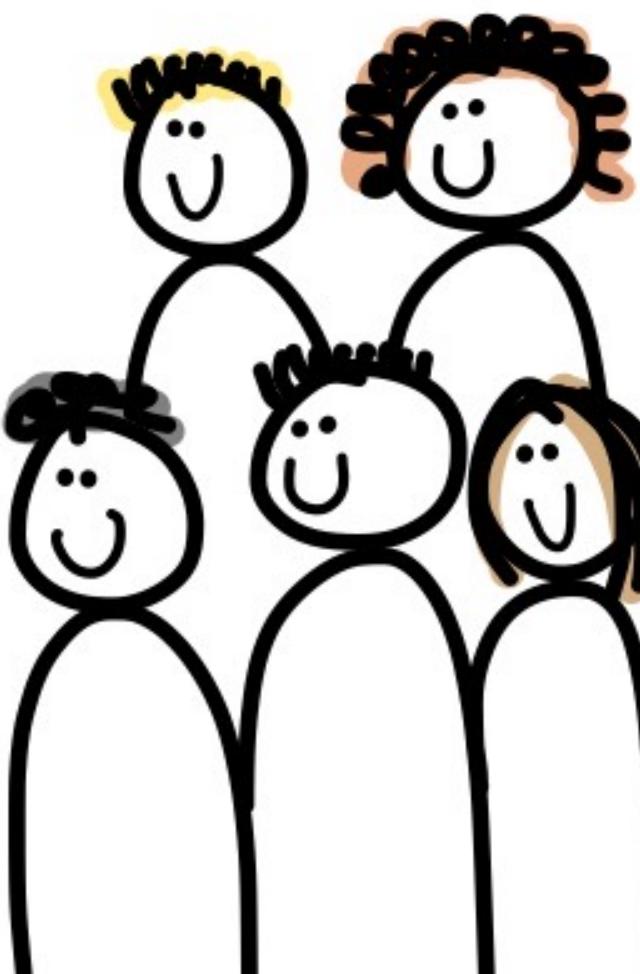


A screenshot of a Python code editor interface. The top bar shows the user 'mevrHermans / HedyTester' and various icons. The left sidebar has a 'Files' tab selected, showing a folder icon and a partially visible file 'mai...'. The main area displays a file named 'main.py' with the following code:

```
1 print("Hello everyone!")  
2  
3
```

To the right of the code editor is a 'Console' window with the output:

```
Hello everyone!  
▶ []
```



mevrHermans / HedyTester

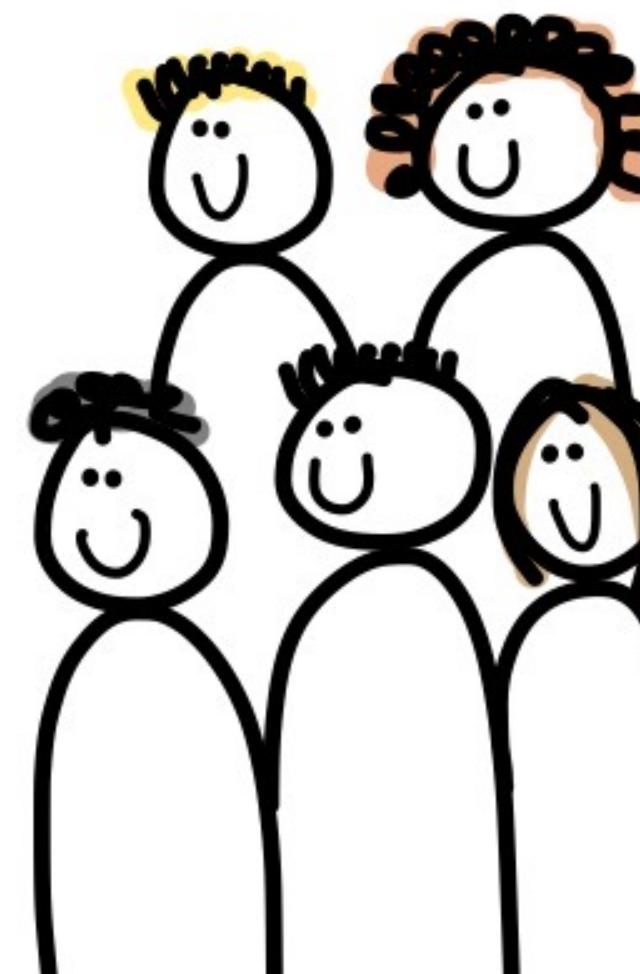
main.py

```
1 Print("Hello everyone!")  
2  
3
```

Files

Console Shell

Print("Hello everyone!")



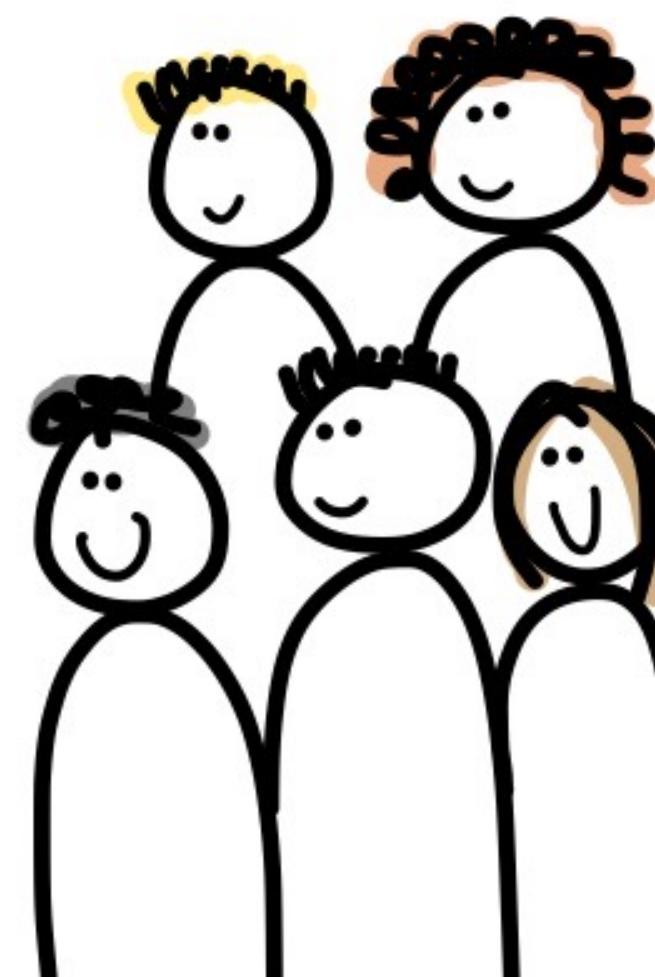
A screenshot of a Python code editor interface. The top bar shows the user 'mevrHermans / HedyTester' and a green play button icon. The left sidebar has a 'Files' tab selected, showing a folder structure with a file named 'main.py'. The main editor area displays the following code:

```
1 Print("Hello everyone!")  
2  
3
```

The word 'Print' is underlined with a red wavy line, indicating a spelling error. The status bar at the bottom of the editor shows 'File main.py 1 line'.

To the right of the editor is a 'Console' tab, which is active. It displays a traceback of the error:

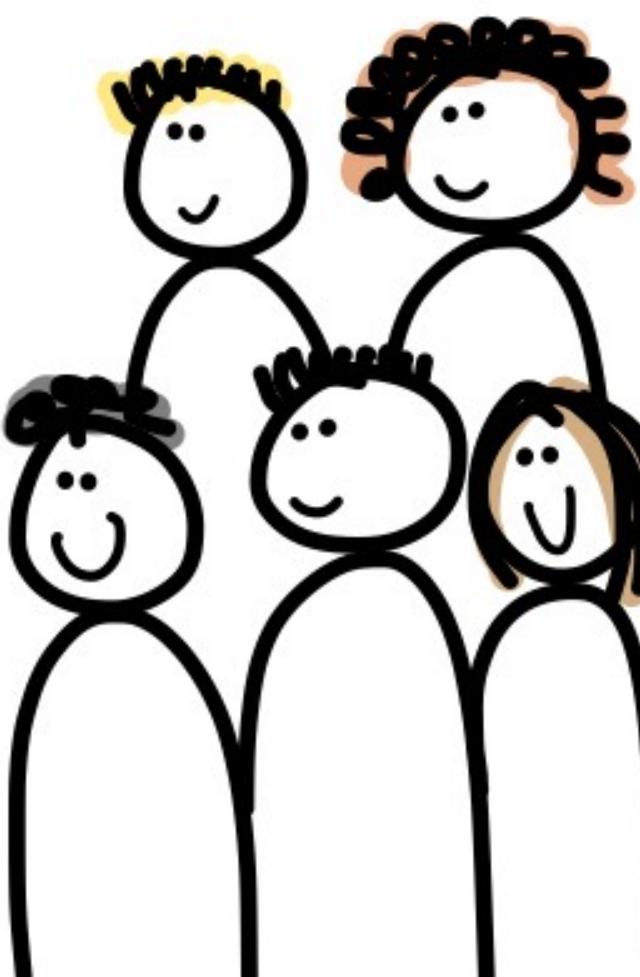
```
Traceback (most recent call last):  
  File "main.py", line 1, in <module>  
    Print("Hello everyone!")  
NameError: name 'Print' is not defined  
> []
```



A screenshot of a Python code editor interface. The top bar shows the user 'mevrHermans / HedyTester'. The left sidebar has a 'Files' tab selected, showing a folder icon and a file icon labeled 'mai...'. The main area displays a file named 'main.py' with the following code:

```
1 print("Hello everyone!")
```

The right side of the interface includes tabs for 'Console' and 'Shell', and a large dark blue rectangular area representing the terminal or console window.

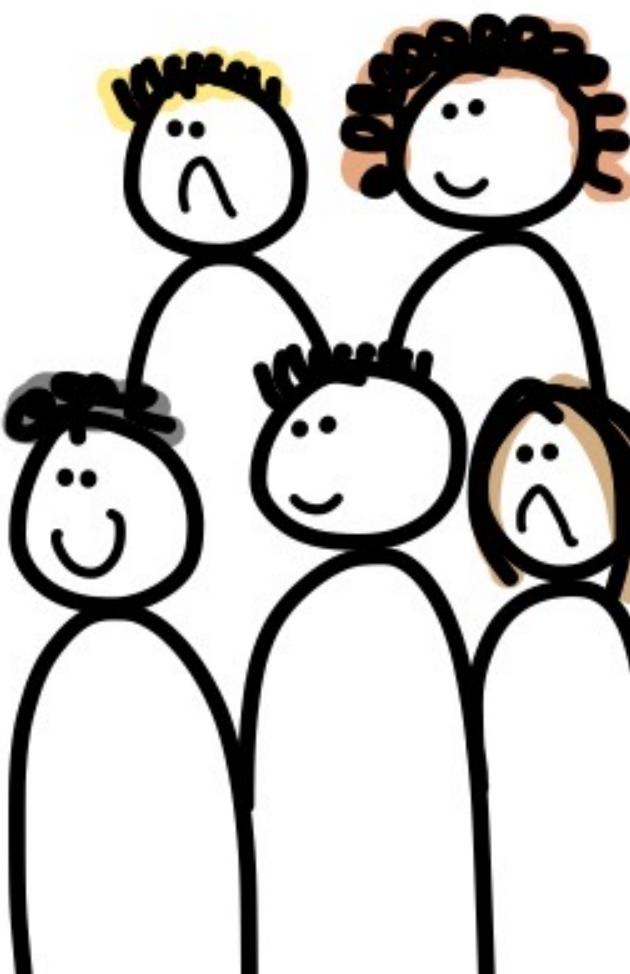
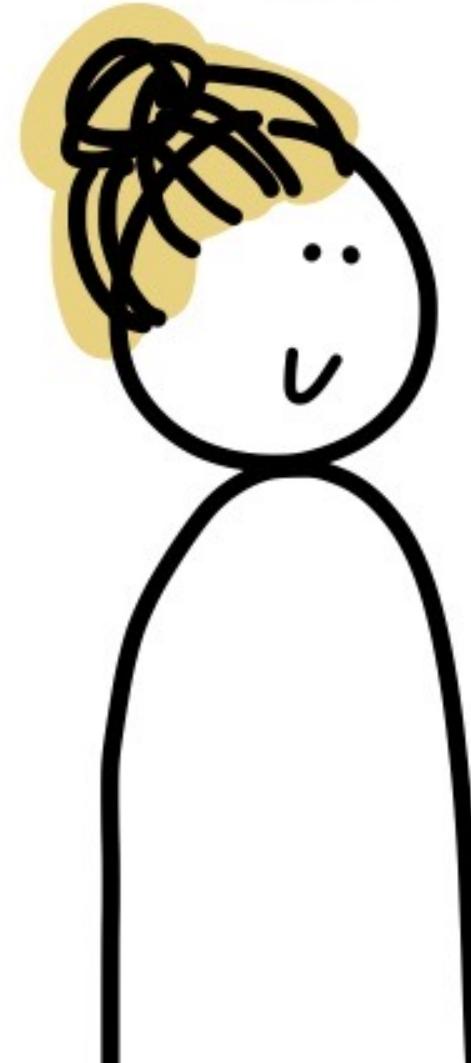


A screenshot of a code editor interface. The top bar shows the user 'mevrHermans / HedyTester' and a green play button icon. The left sidebar has a 'Files' tab selected, showing a folder icon and a file named 'main.py'. The main editor area shows the code:

```
1 print("Hello everyone!")  
2  
3
```

The console tab is active, displaying the error message:

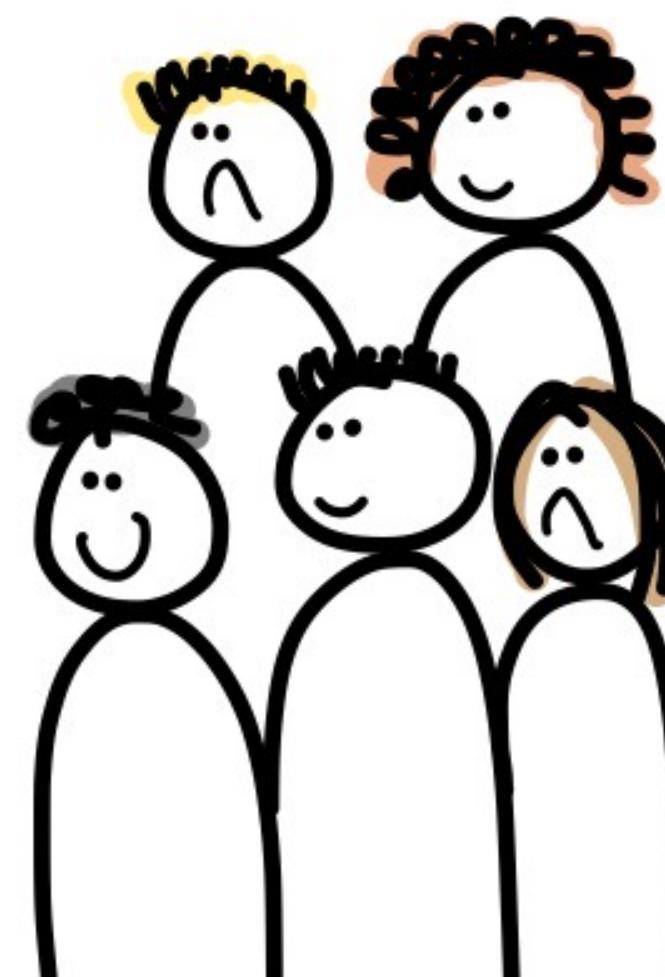
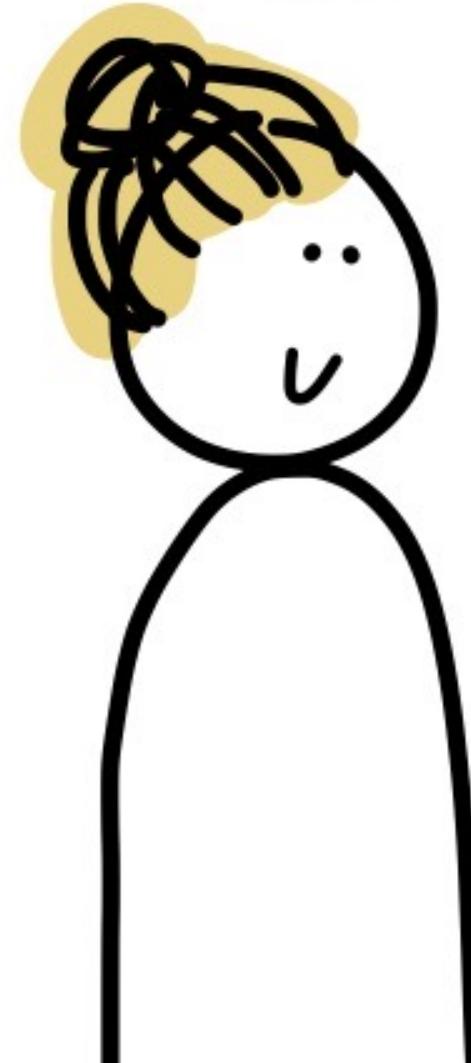
```
File "main.py", line 3  
    ^  
SyntaxError: unexpected EOF while parsing
```



A screenshot of a web-based Python code editor. The interface includes a top bar with user information (mevrHermans / HedyTester), a play button, and an invite button. On the left is a sidebar with icons for Files, Settings, and Help. The main area shows a file named 'main.py' containing the code:

```
1 print("Hello everyone!")
```

The code editor has tabs for 'Console' and 'Shell', but the console tab is currently active and is empty.



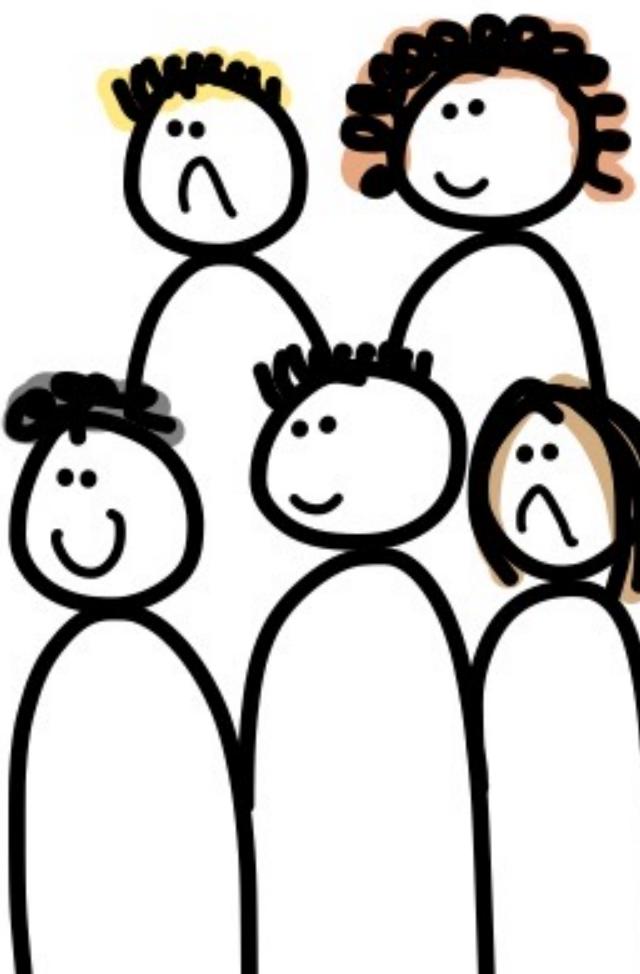
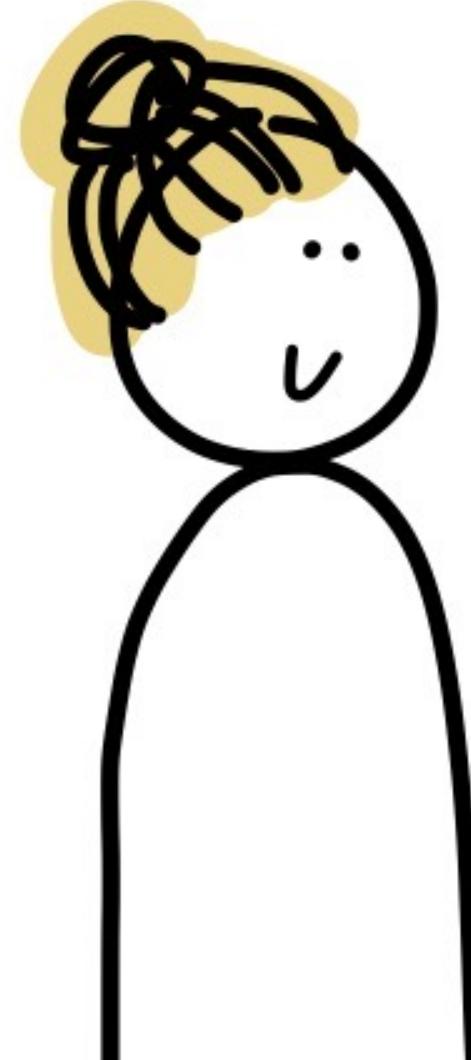
mevrHermans / HedyTester

Files main.py

```
main.py x
1 print("Hello everyone!")
```

Console Shell

Space



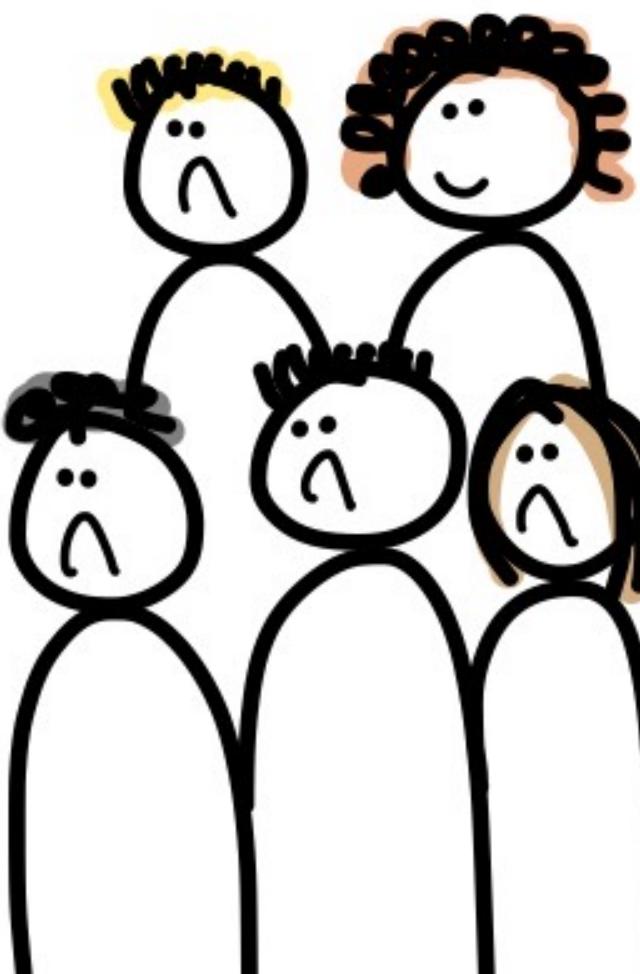
A screenshot of a code editor interface. On the left is a sidebar with icons for Files, Version Control, Directories, Lock, Settings, and Help. The main area shows a file named 'main.py' with the following content:

```
1 print("Hello everyone!")  
2  
3
```

An arrow points from the word "Space" to the blank line between line 2 and line 3. In the bottom right corner of the code editor, there is handwritten text: "Space".

The status bar at the top shows the user's name "mevrHermans / HedyTester" and some icons. To the right of the code editor is a terminal window titled "Console" with the following output:

```
File "main.py", line 1  
print("Hello everyone!")  
^  
IndentationError: unexpected indent  
:> |
```





Compilers are
lovely teachers



```
for i in range(4):  
    print(i)
```



Repetition



```
for i in range(4):  
    print(i)
```



Repetition



```
for i in range(4):  
    print(i)
```

Brackets

Spaces

Colon



Syntax creates
cognitive load

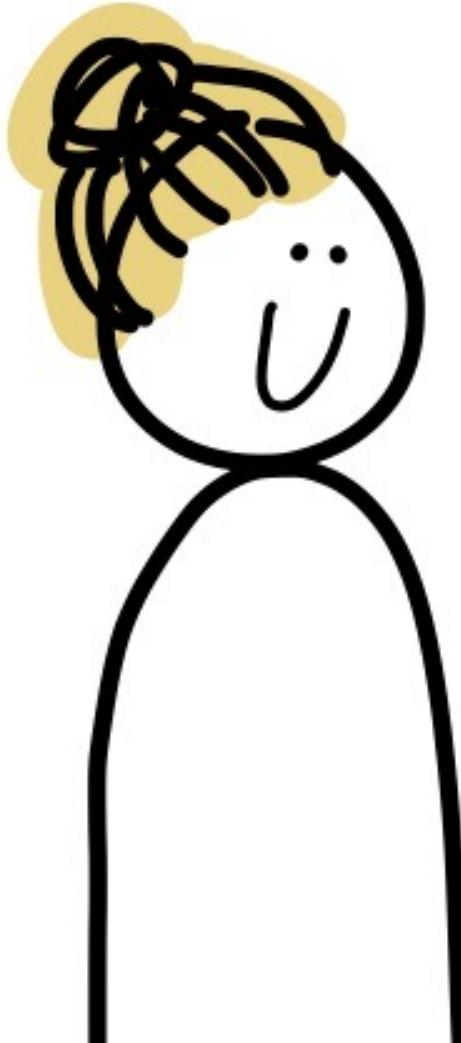


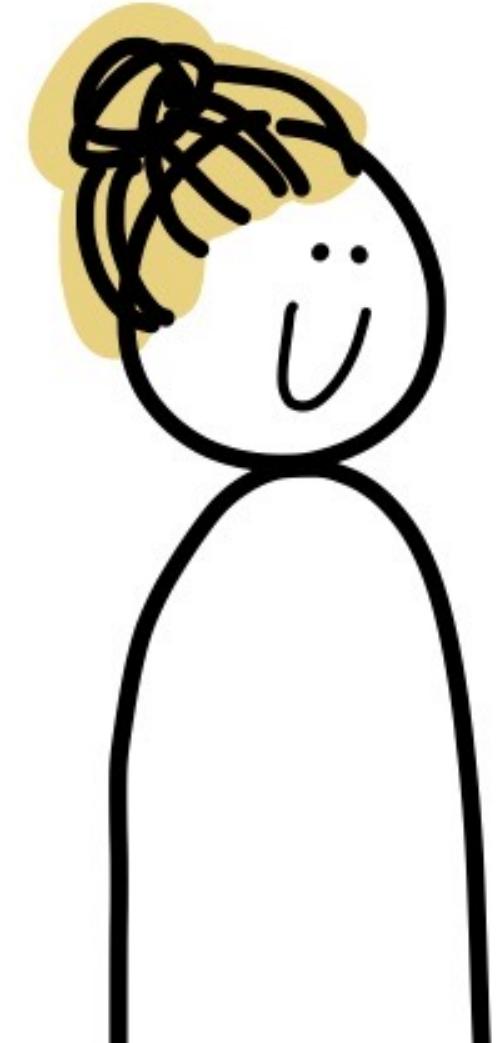
```
for i in range(4):  
    print(i)
```

Colon
Brackets
Spaces



How do other fields
manage cognitive load?





a in e

cat in tree

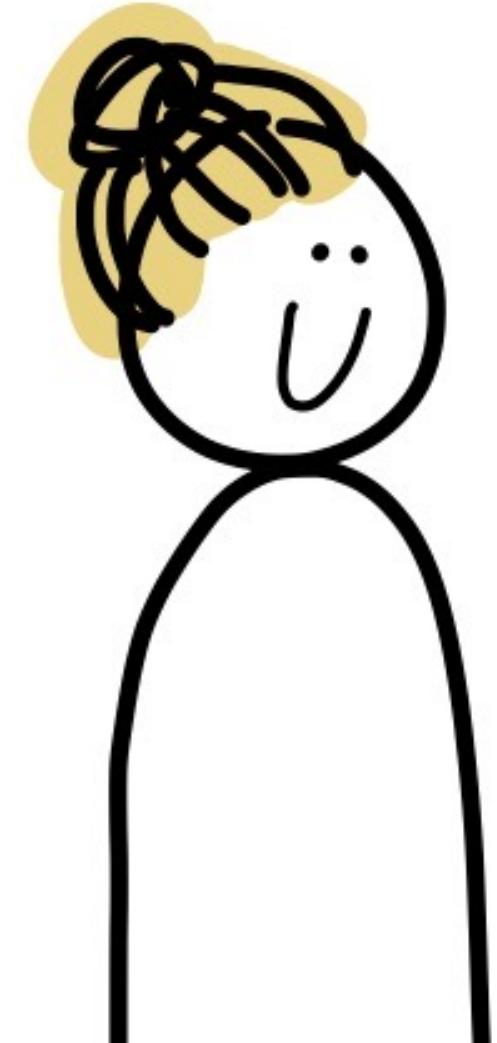
Cat in tree

Cat in tree.

The cat in the

tree.





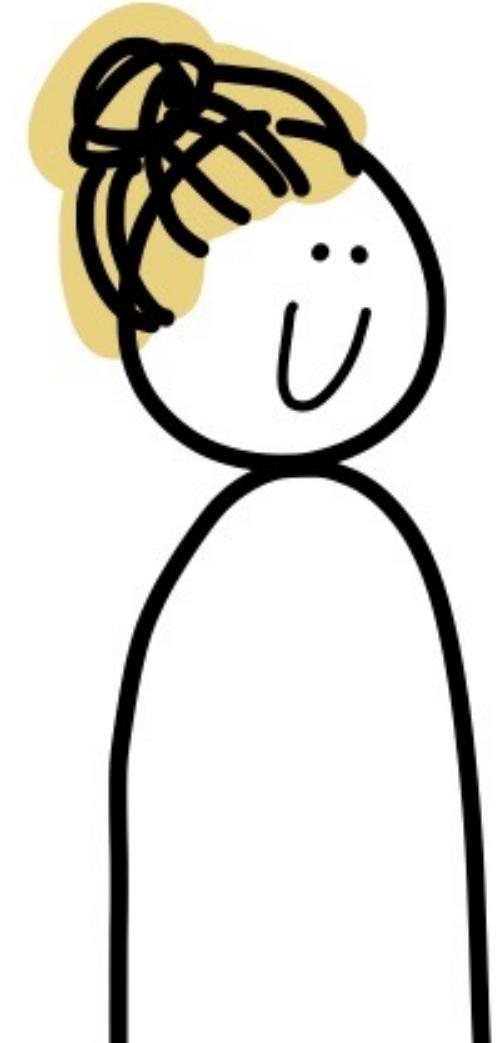
a in e
cat in tree
Cat in tree
Cat in tree.
The cat in the
tree.





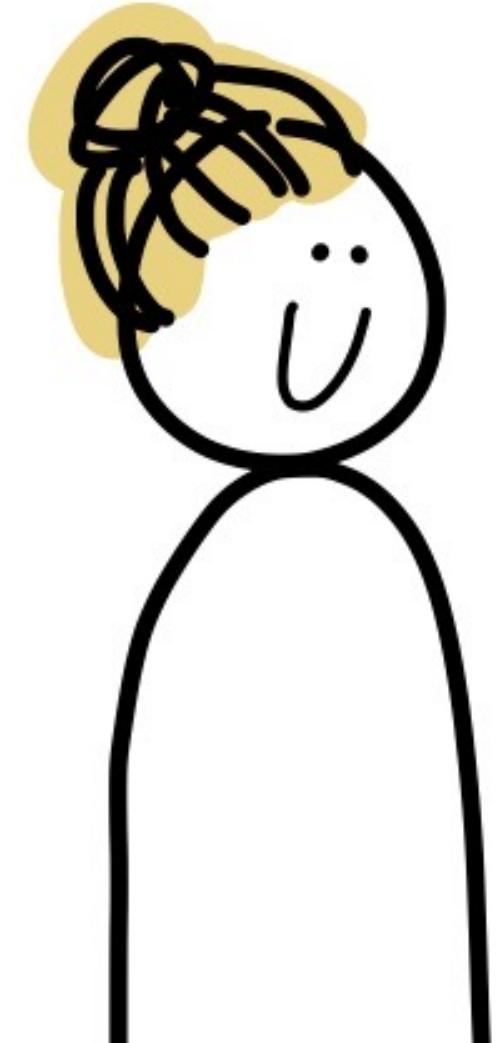
a in e
cat in tree
Cat in tree
Cat in tree.
The cat in the
tree.





a in e
cat in tree
Cat in tree
Cat in tree.
The cat in the
tree.

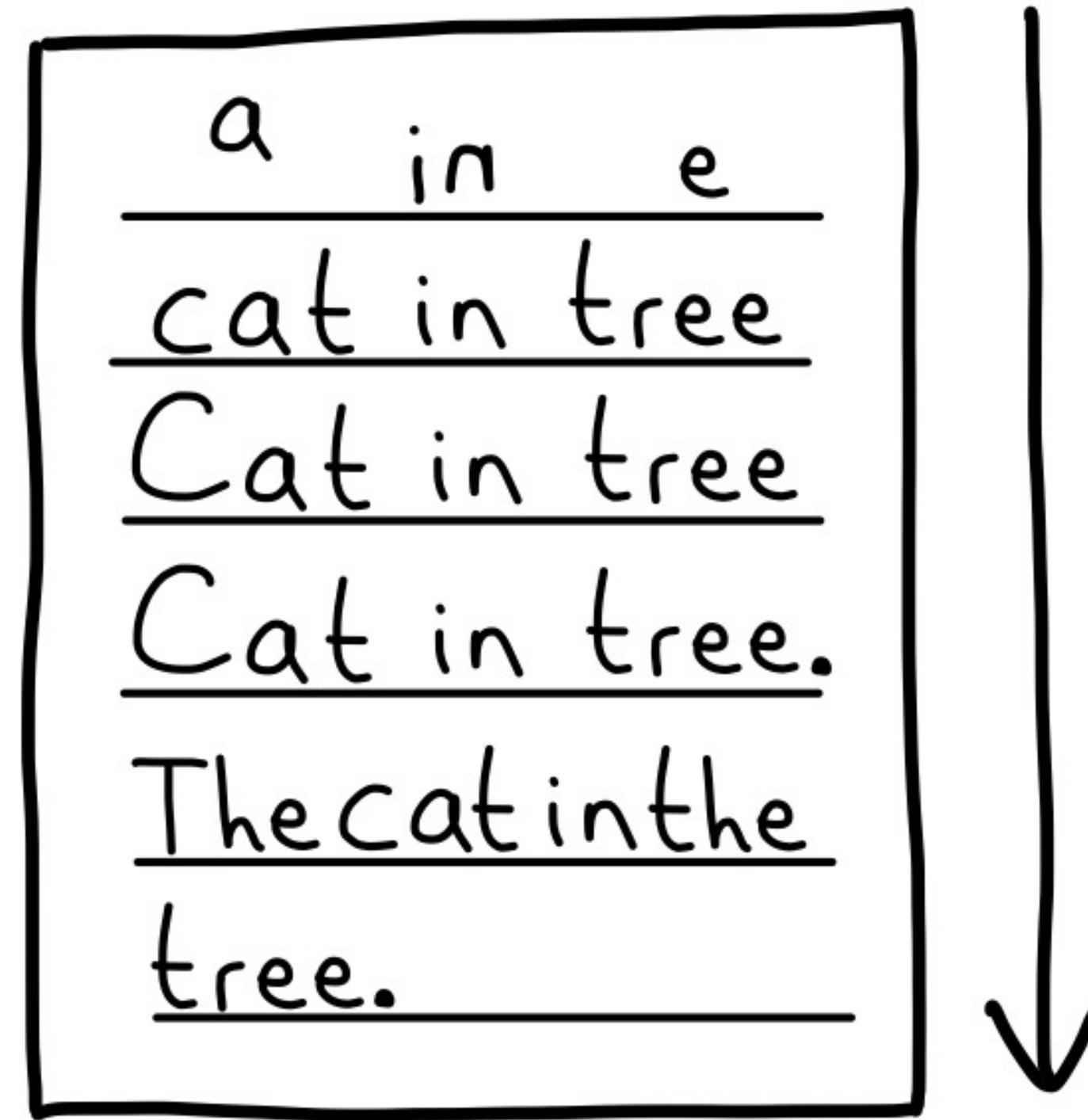
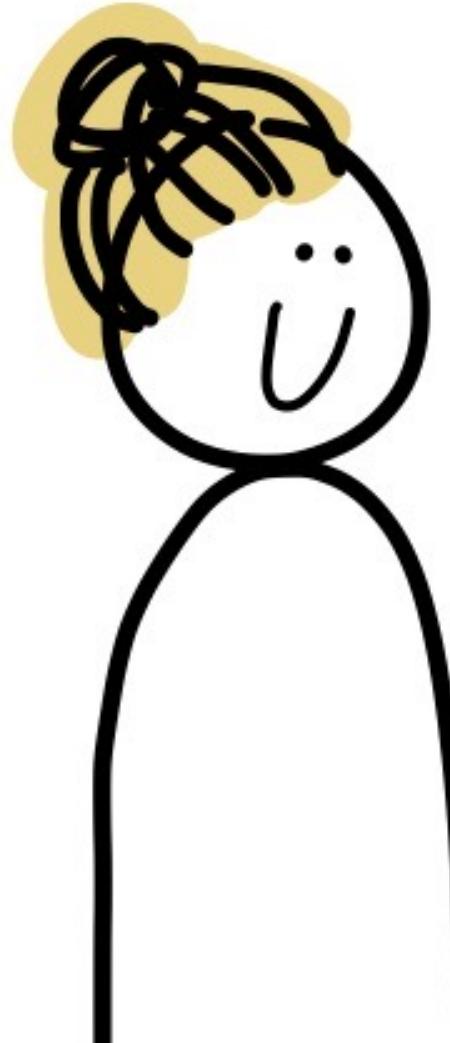




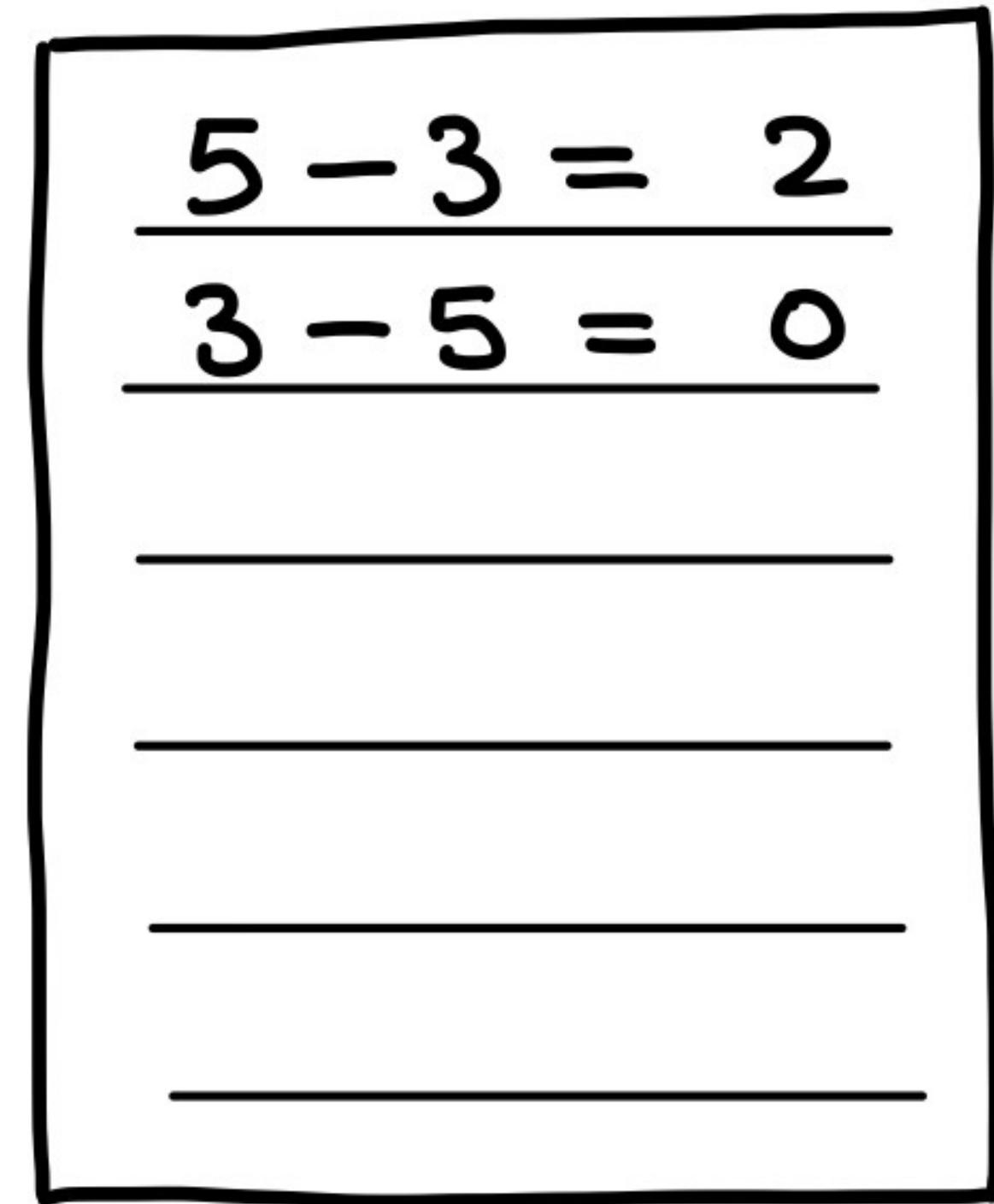
a in e
cat in tree
Cat in tree
Cat in tree.
The cat in the
tree.

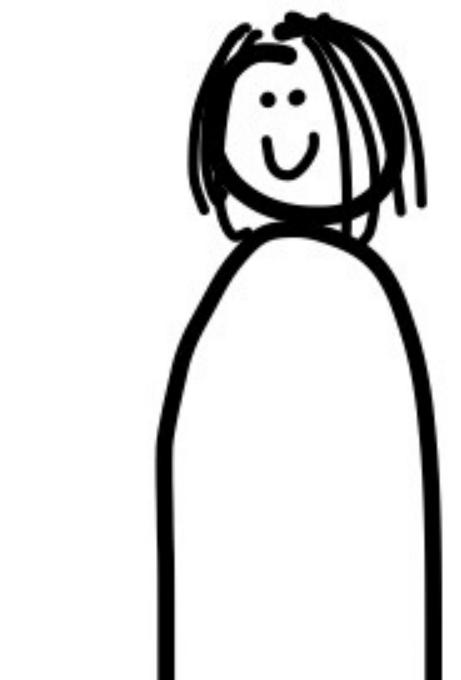
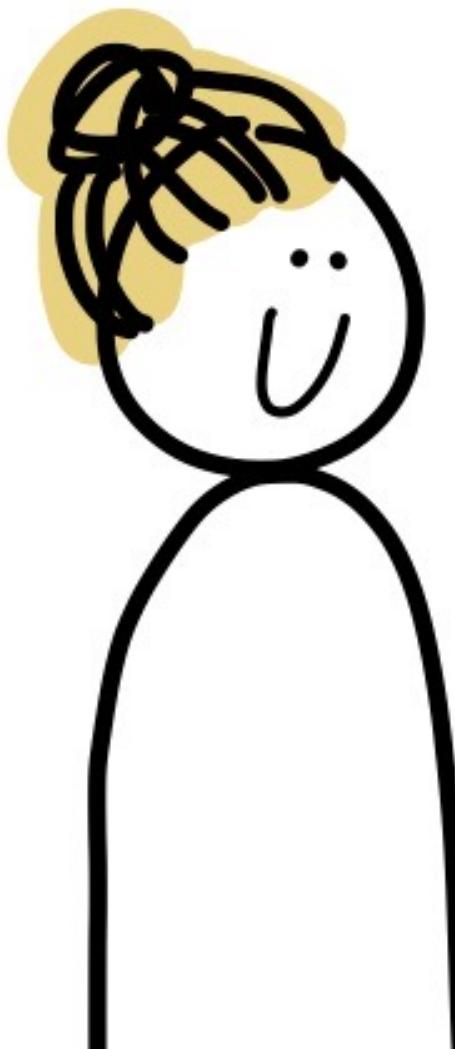


Rules gradually
change



Rules gradually
change
in math too!

$$\begin{array}{r} 5 - 3 = 2 \\ \hline 3 - 5 = 0 \end{array}$$




Rules gradually
change
in math too!



$$\begin{array}{r} 5 - 3 = 2 \\ \hline 3 - 5 = 0 \\ \hline 3 - 5 = -2 \end{array}$$

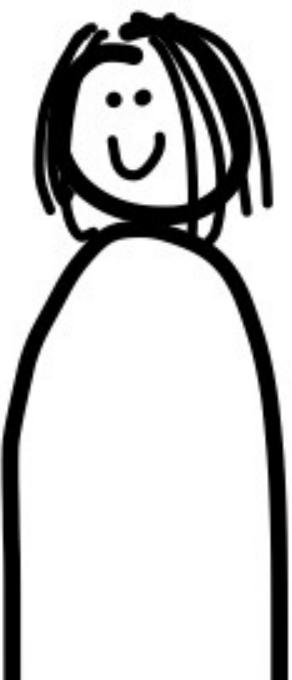
A rectangular frame containing three horizontal lines of mathematical equations. The first line shows 5 minus 3 equals 2. The second line shows 3 minus 5 equals 0. The third line shows 3 minus 5 equals negative 2. Below each equation is a blank line for notes.



Rules gradually
change
in math too!

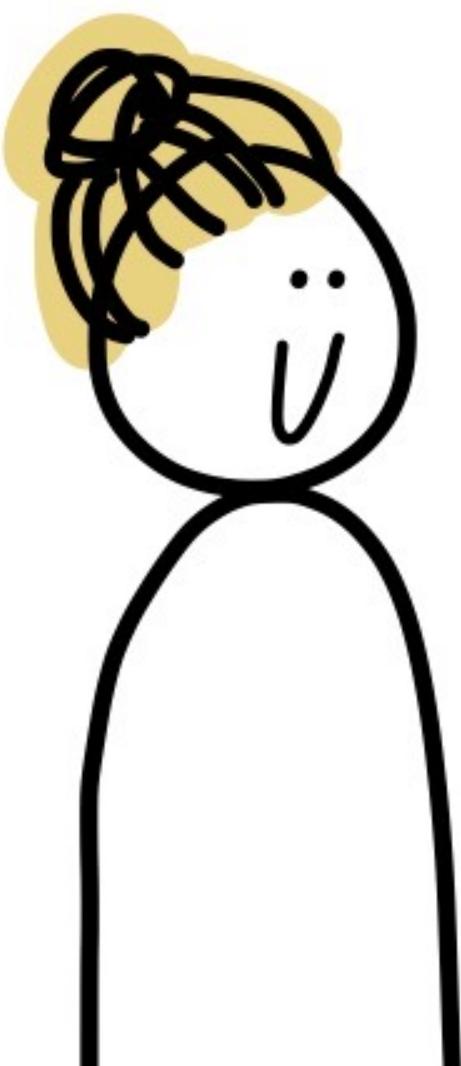


$$\begin{array}{r} 5 - 3 = 2 \\ \hline 3 - 5 = 0 \\ \hline 3 - 5 = -2 \\ \hline 8 / 3 = 2 \text{ r } 2 \\ \hline 8 / 3 = 2 \frac{2}{3} \\ \hline 8 / 3 = 2,666... \end{array}$$



Can we teach code
gradually?



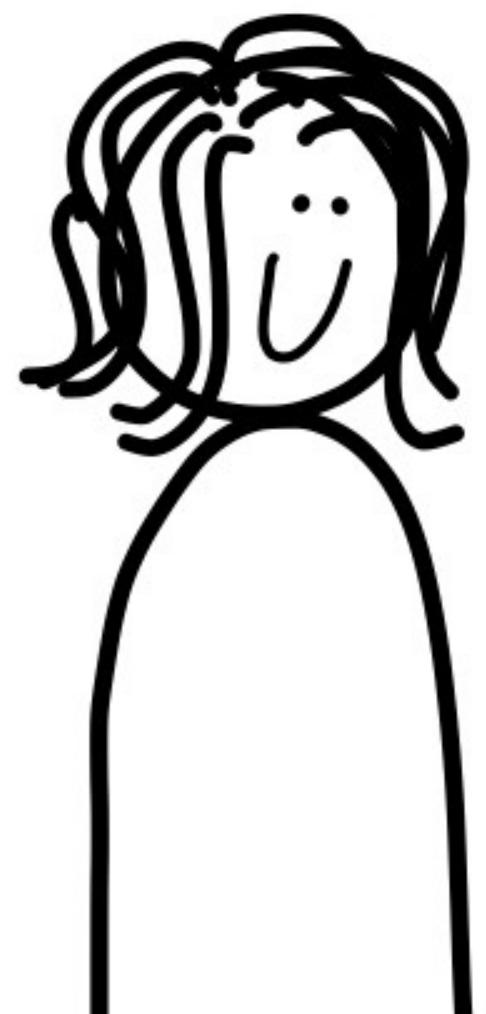


Let's look at
a demo!

Hedy is gradual ✓

multi lingual

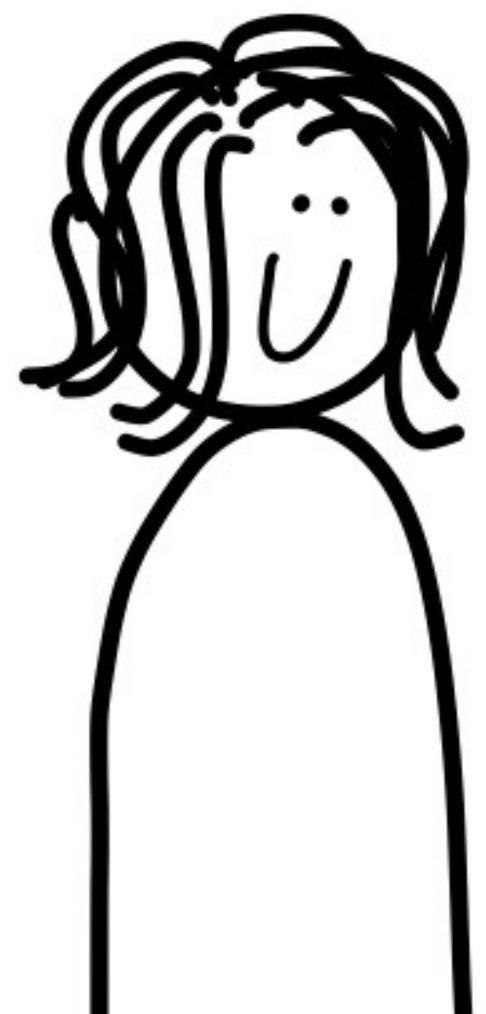
built for teaching



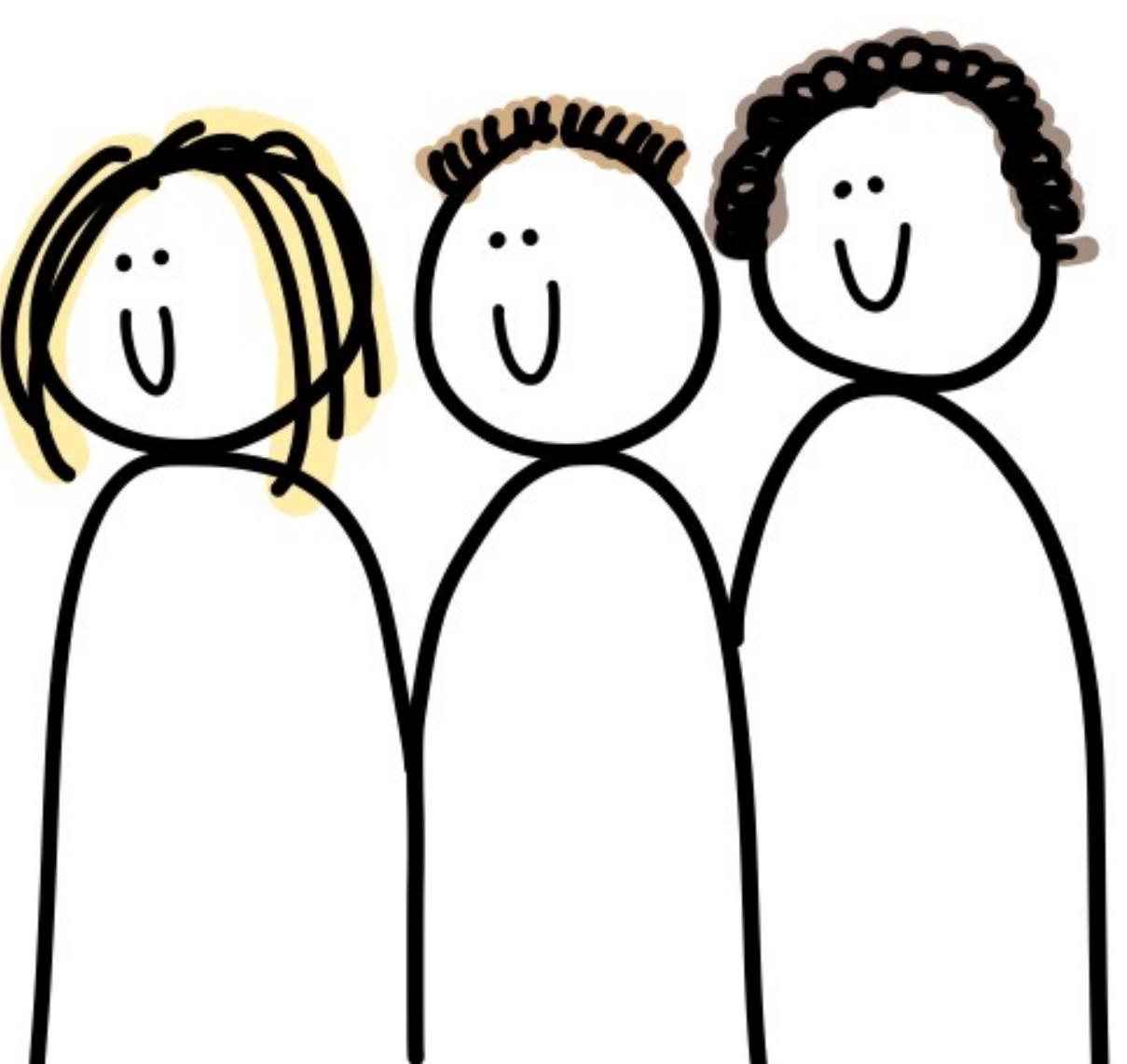
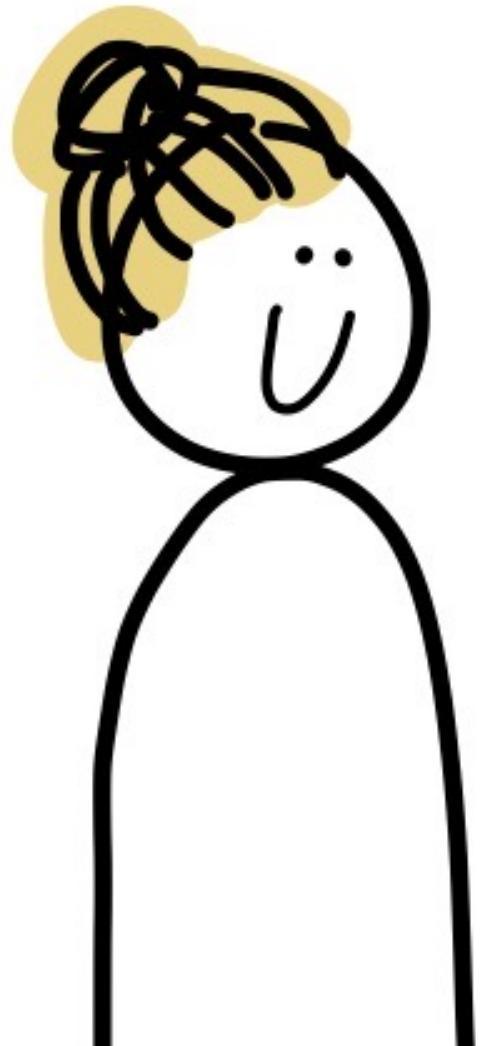
Hedy is gradual

multi lingual

built for teaching

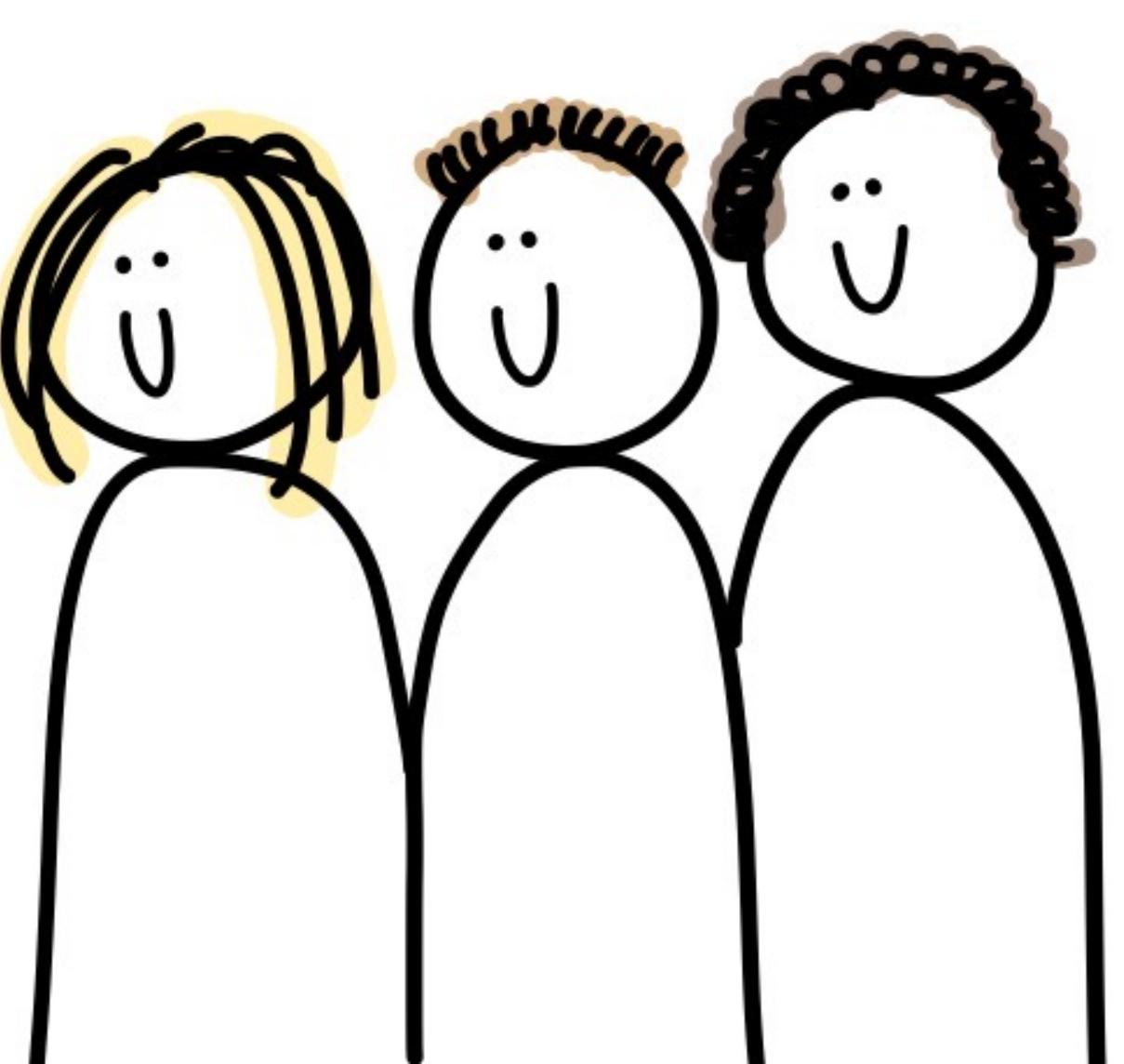
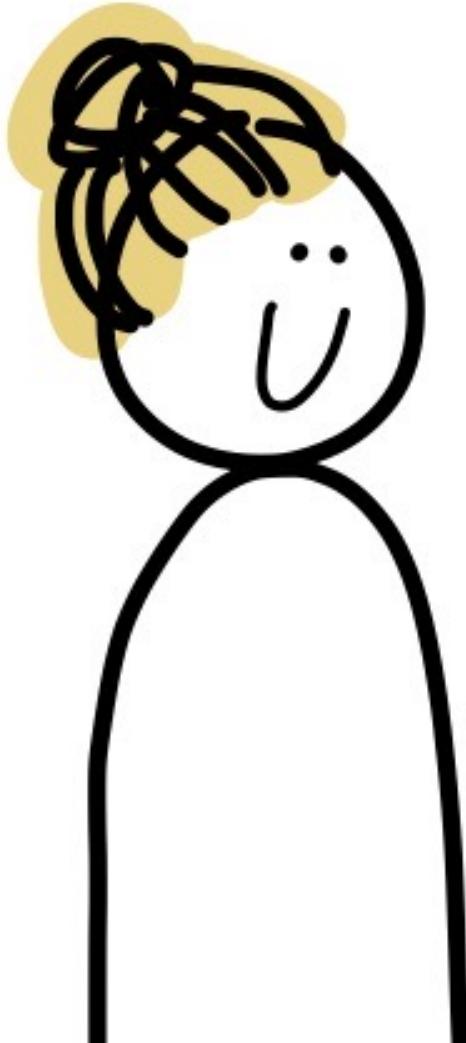


Do kids
like Hedy?

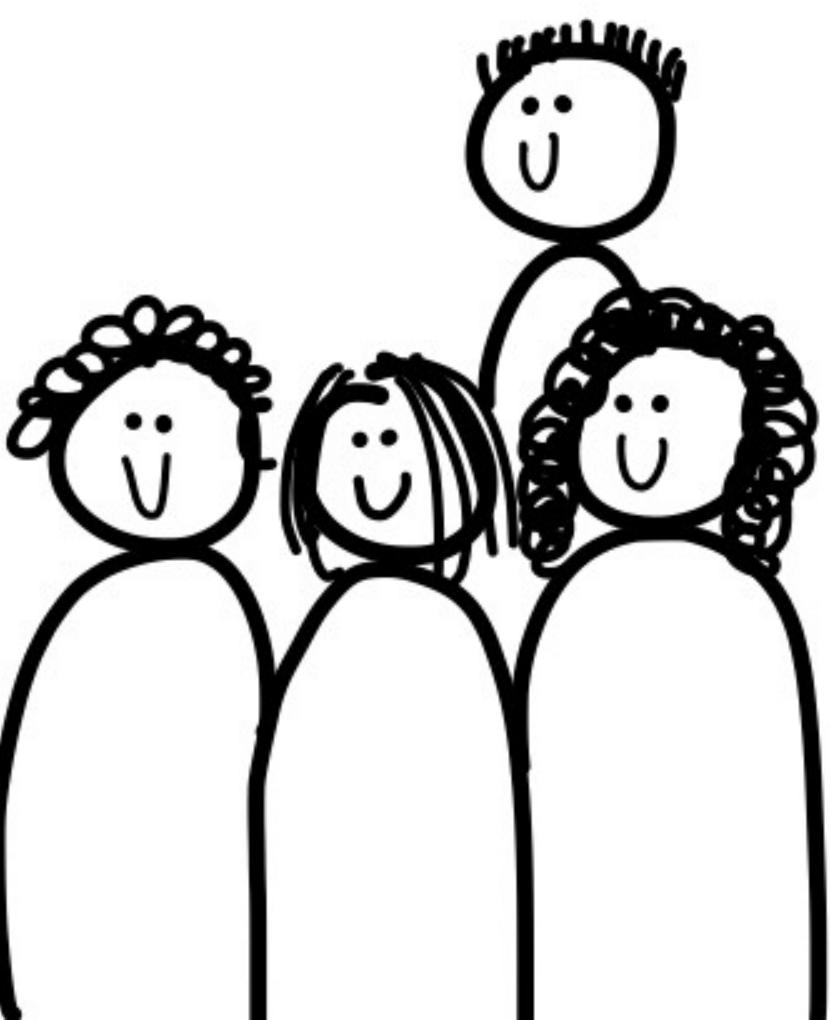
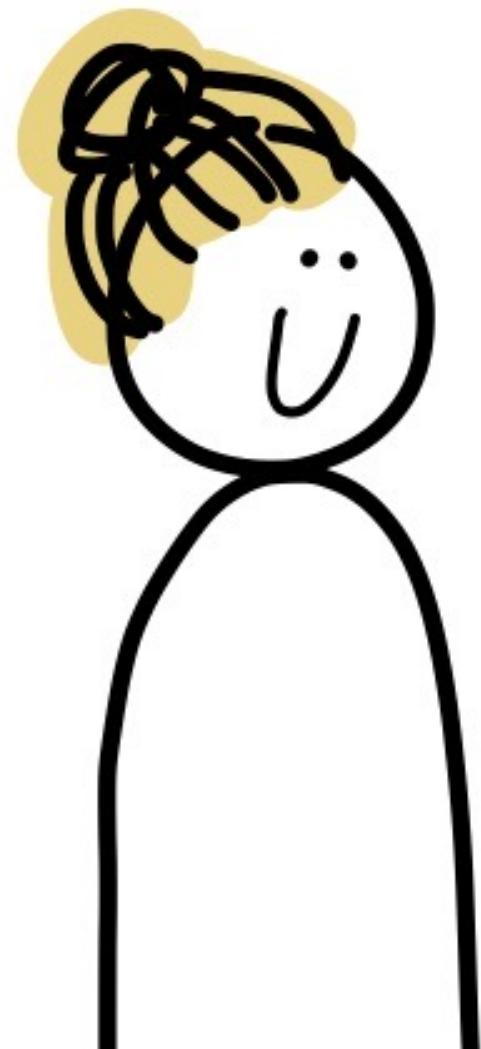


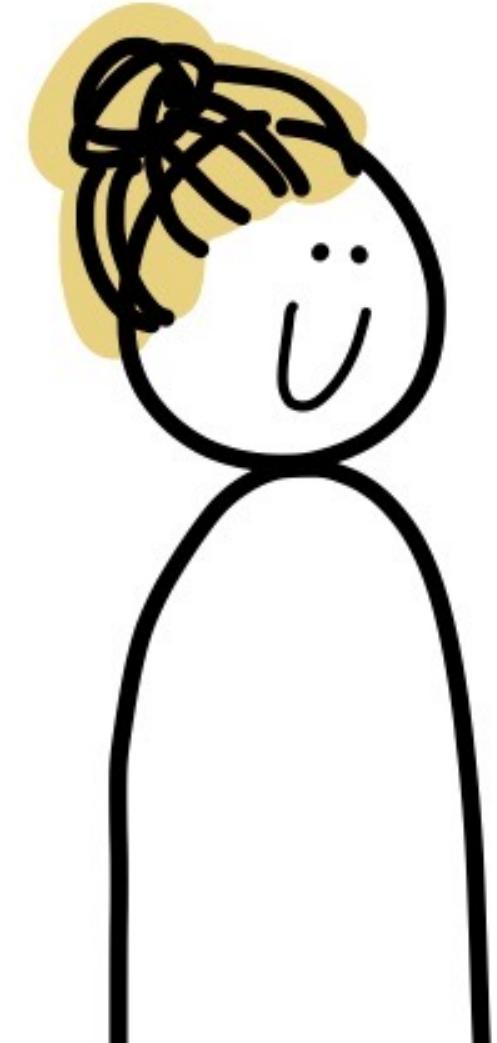
We ran a
user study!

Do kids
like Hedy?



39 kids

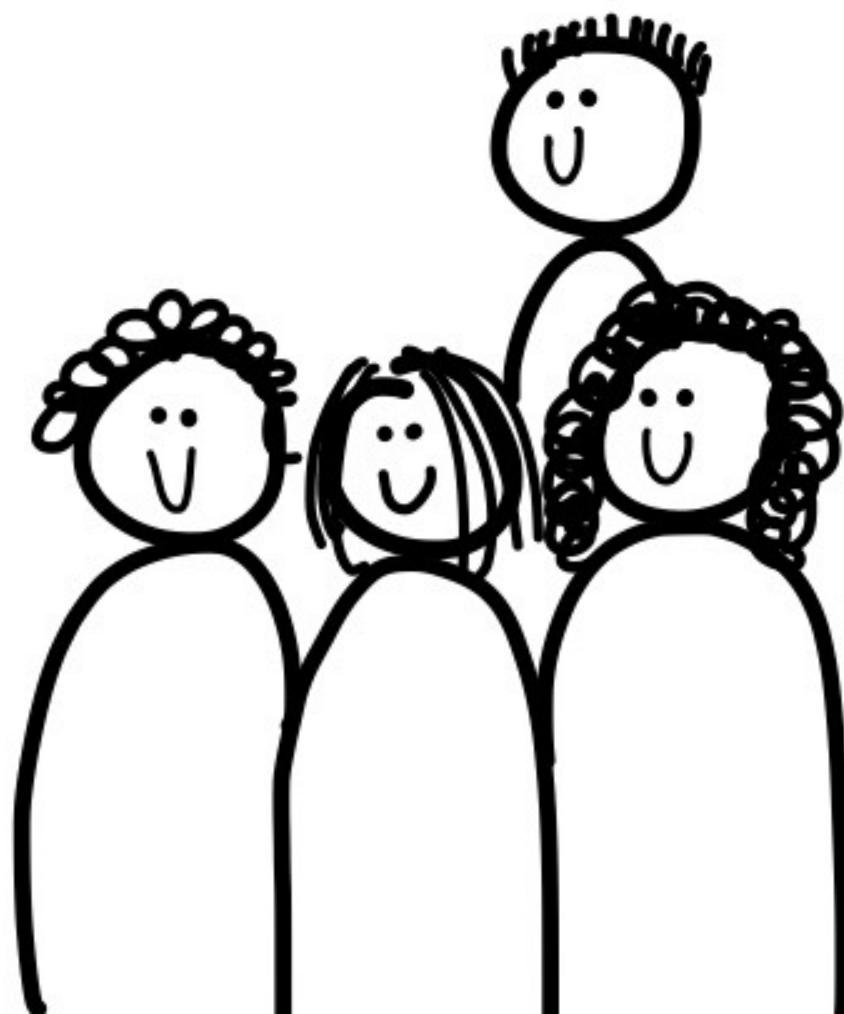




12 online
lessons



39 kids



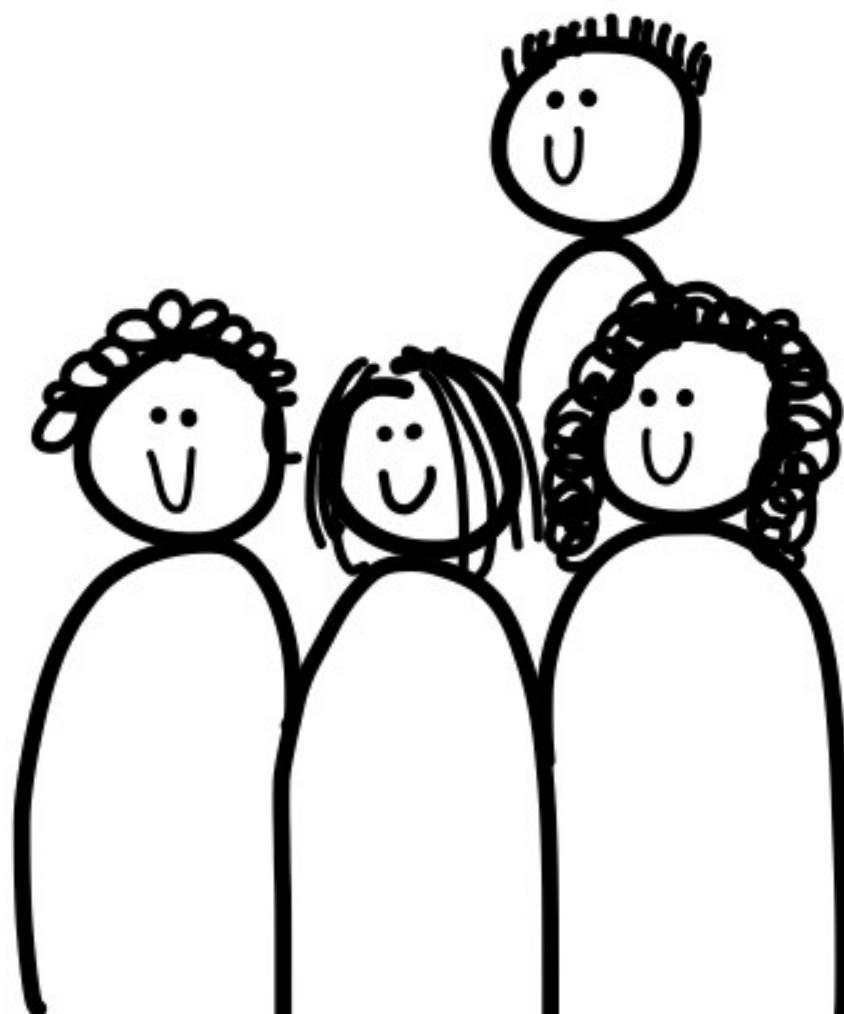
Benefits, challenges & improvements



12 online
lessons



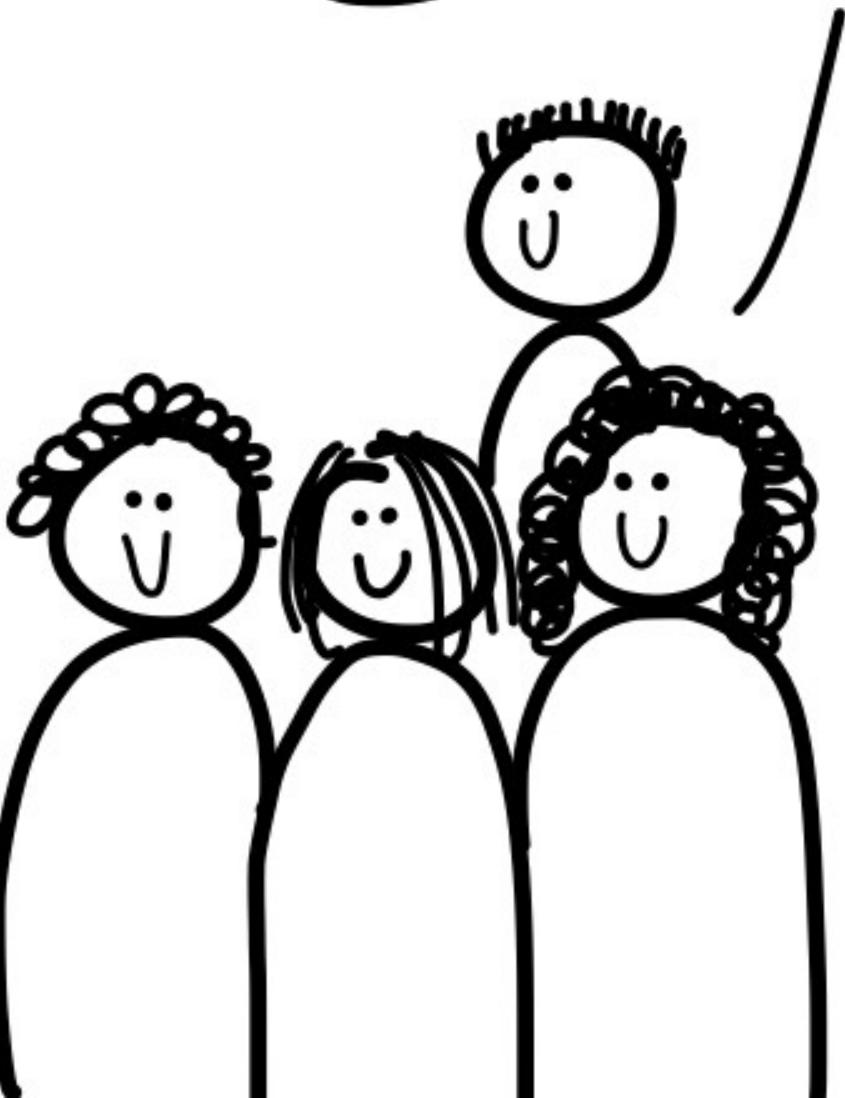
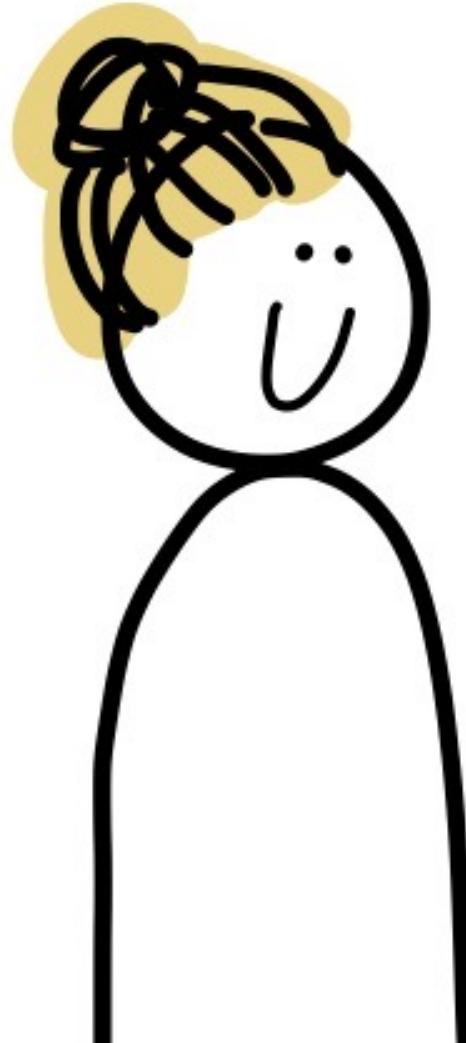
39 kids



Benefits, challenges & improvements

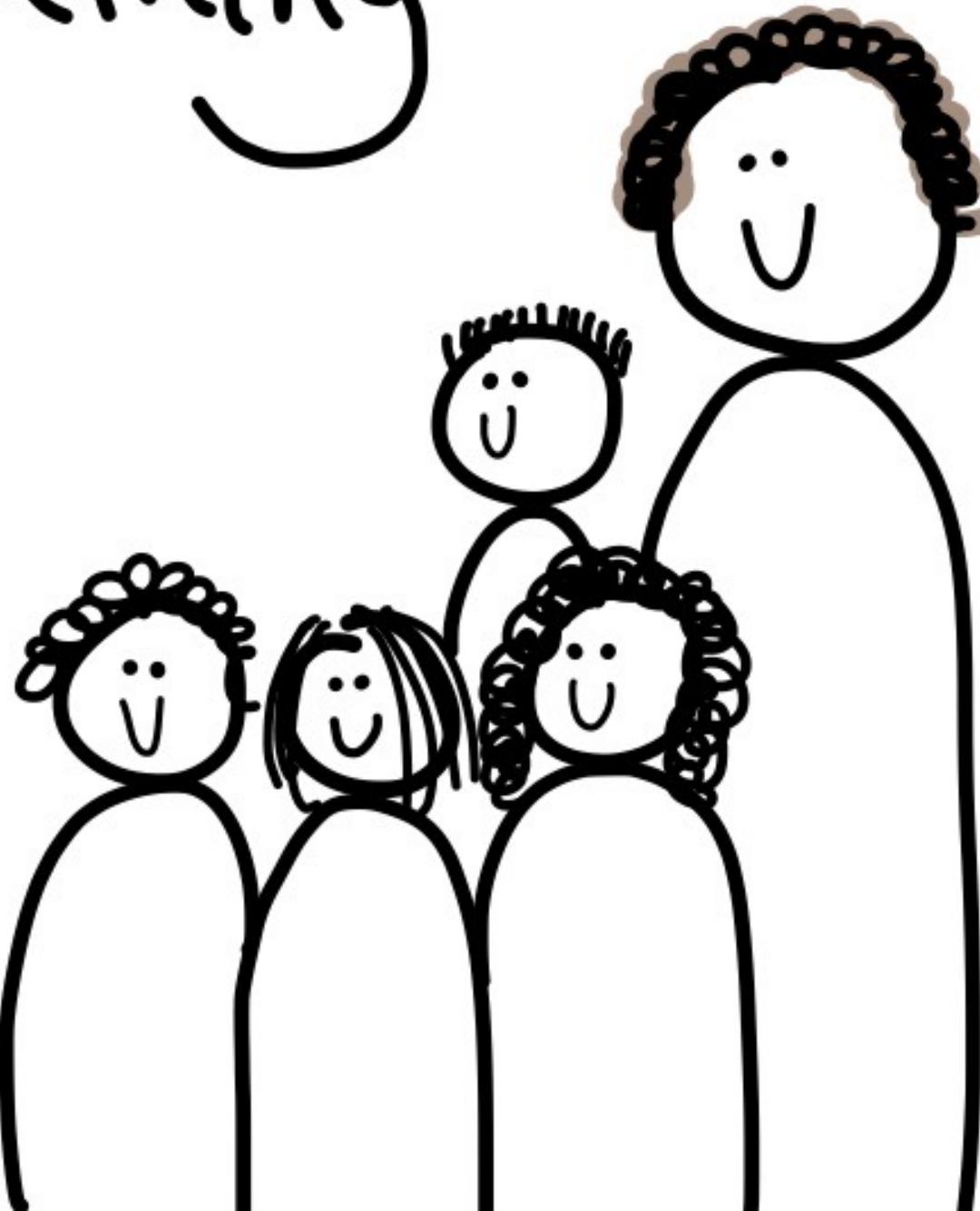
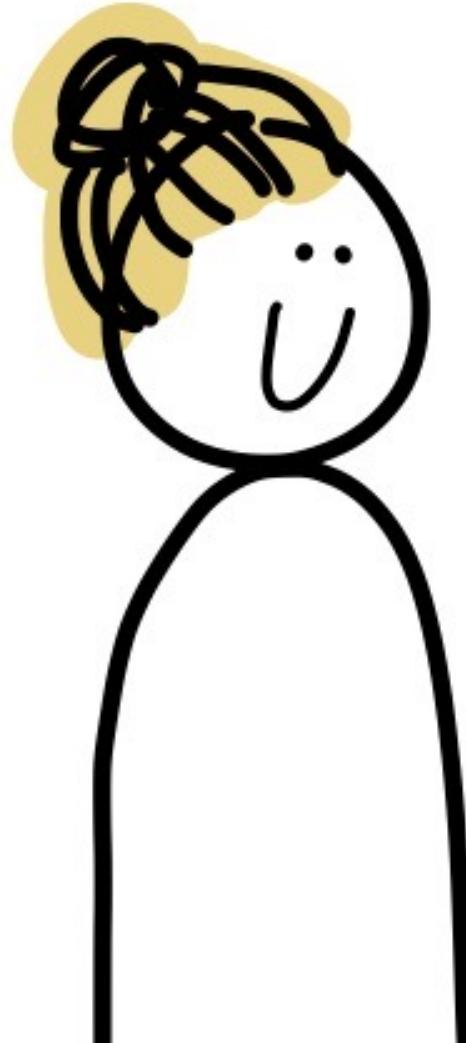


The levels are
a step by step
guide



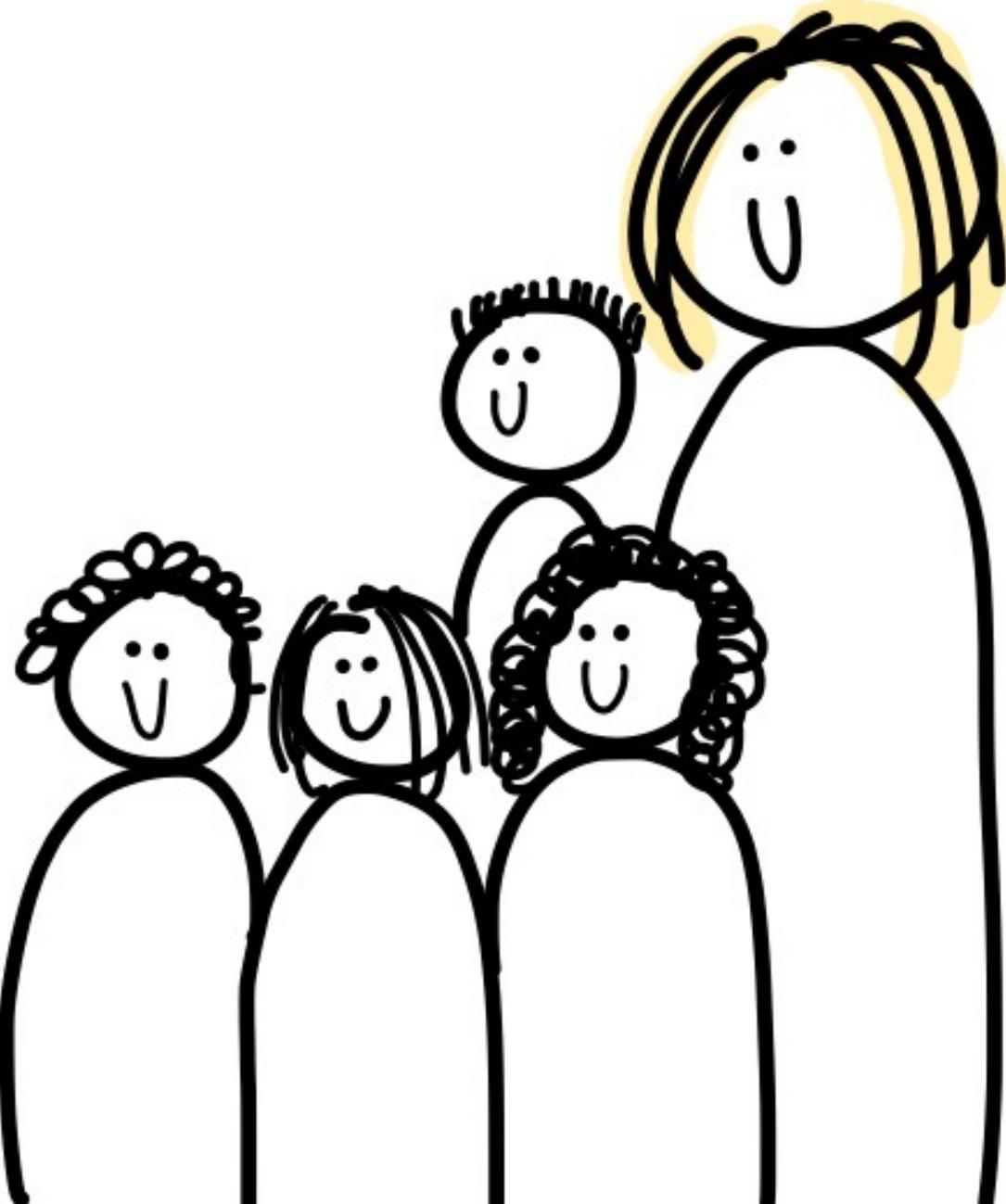
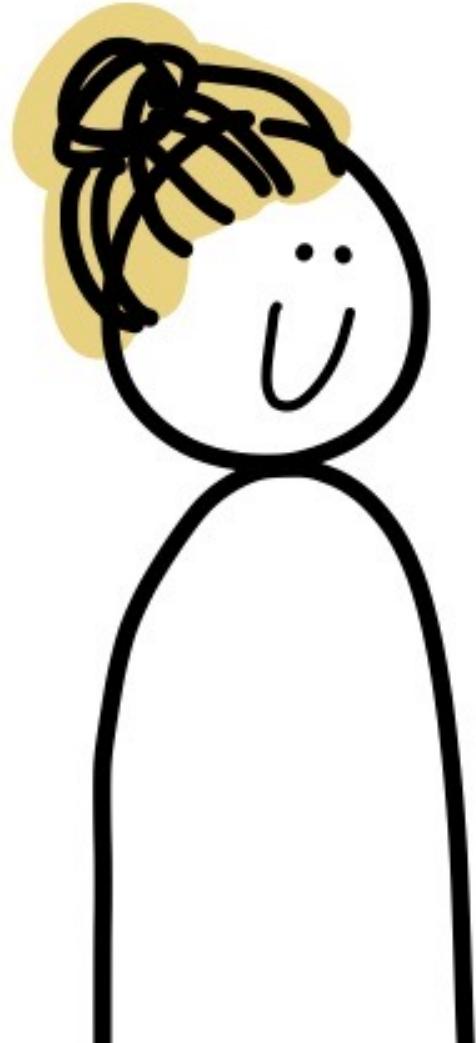
Benefits, challenges & improvements

With Hedy all kids
learn some programming



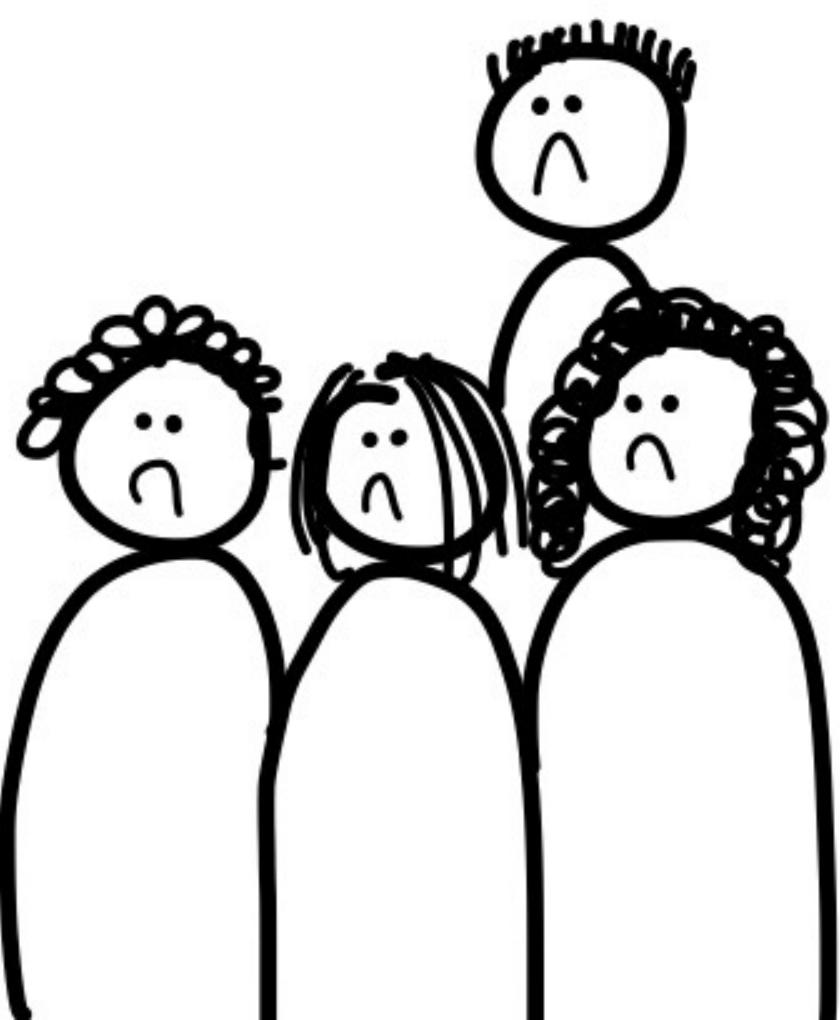
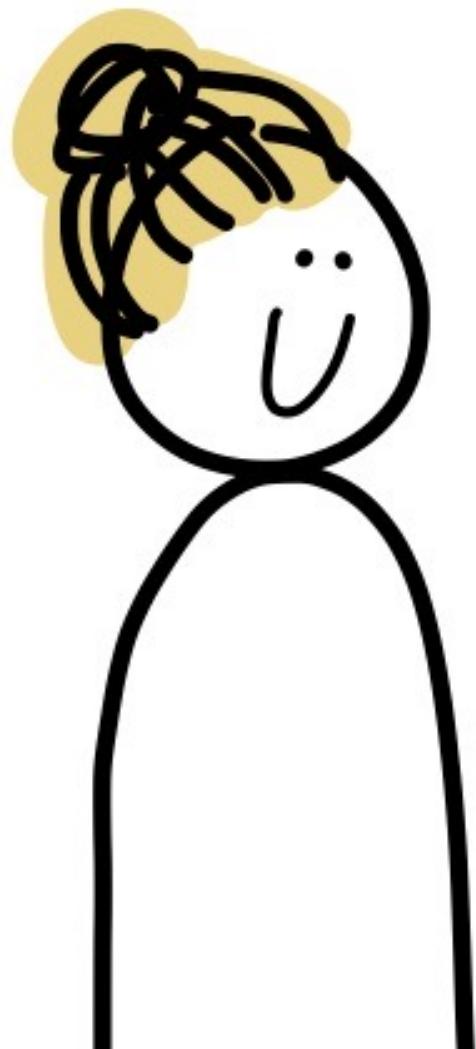
Benefits, challenges & improvements

Hedy gives girls
the confidence to
learn programming



Benefits, challenges & improvements

Better error
messages

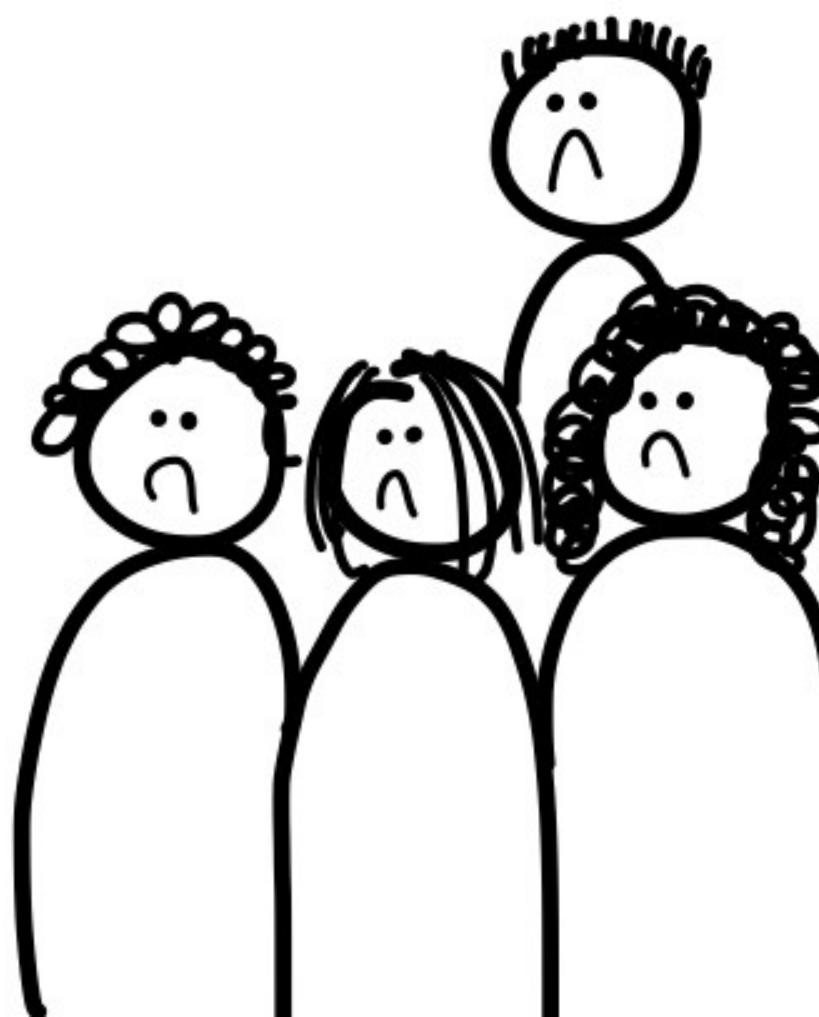


Benefits, challenges & improvements

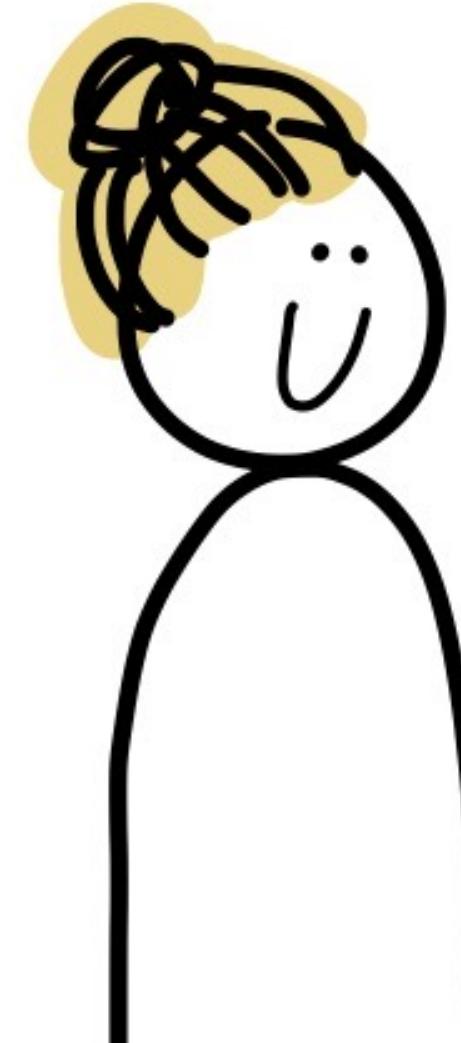


A screenshot of a code editor interface. At the top, there is a dark header bar with the text "1 print Hello, world". Below this is a large black code area. At the bottom, there is a red error message box. The message box contains an information icon (a red circle with a white "i") on the left and a close button ("X") on the right. The text inside the message box reads: "The code you entered is not valid Hedy code. There is a mistake on line 1, at position 12. You typed '!' but that is not allowed."

Better error
messages



Benefits, challenges & improvements

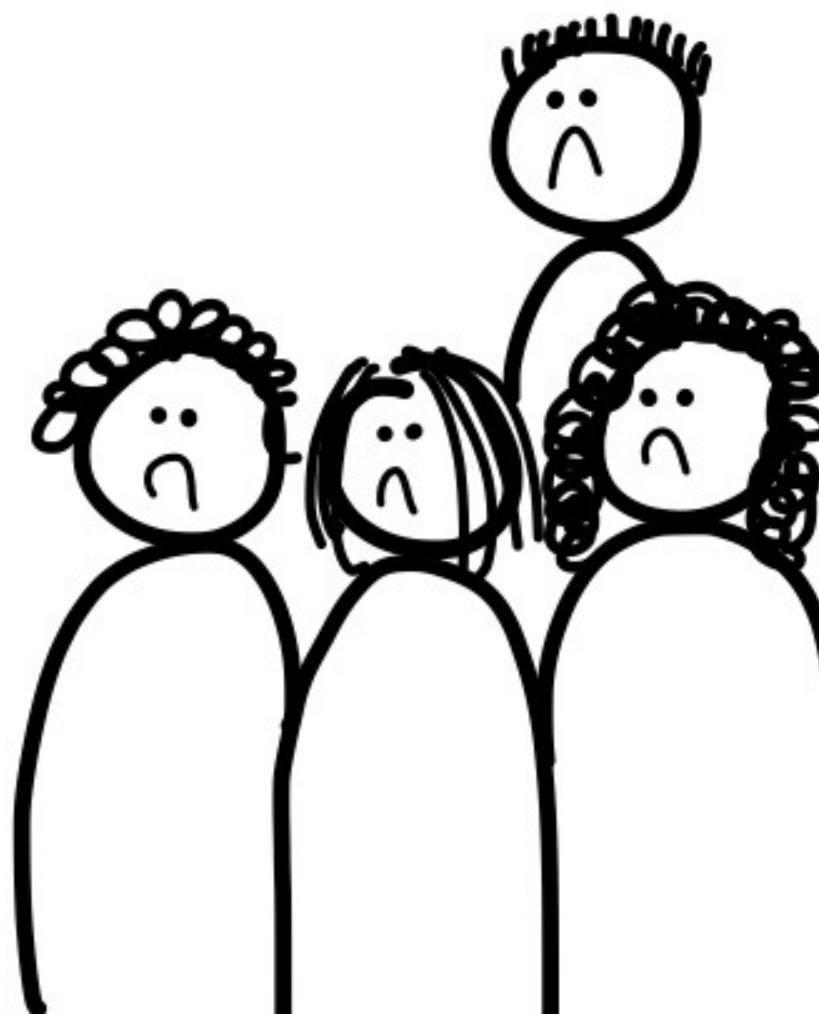


```
1 print Hello, world
```

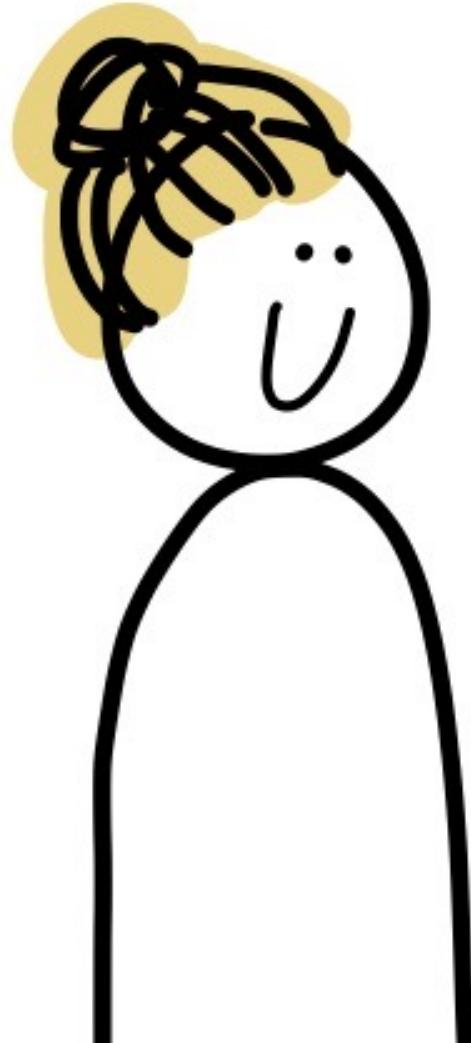
i The code you entered is not valid Hedy code. There is a mistake on line 1, at position 12. You typed , but that is not allowed.
a comma

A screenshot of a code editor showing a syntax error. The code "print Hello, world" is entered in a dark text area. Below it, a red error message box contains the letter "i" followed by the error message: "The code you entered is not valid Hedy code. There is a mistake on line 1, at position 12. You typed , but that is not allowed." A handwritten note "a comma" is written below the error message. In the bottom right corner of the error box is a small red "X".

Better error messages



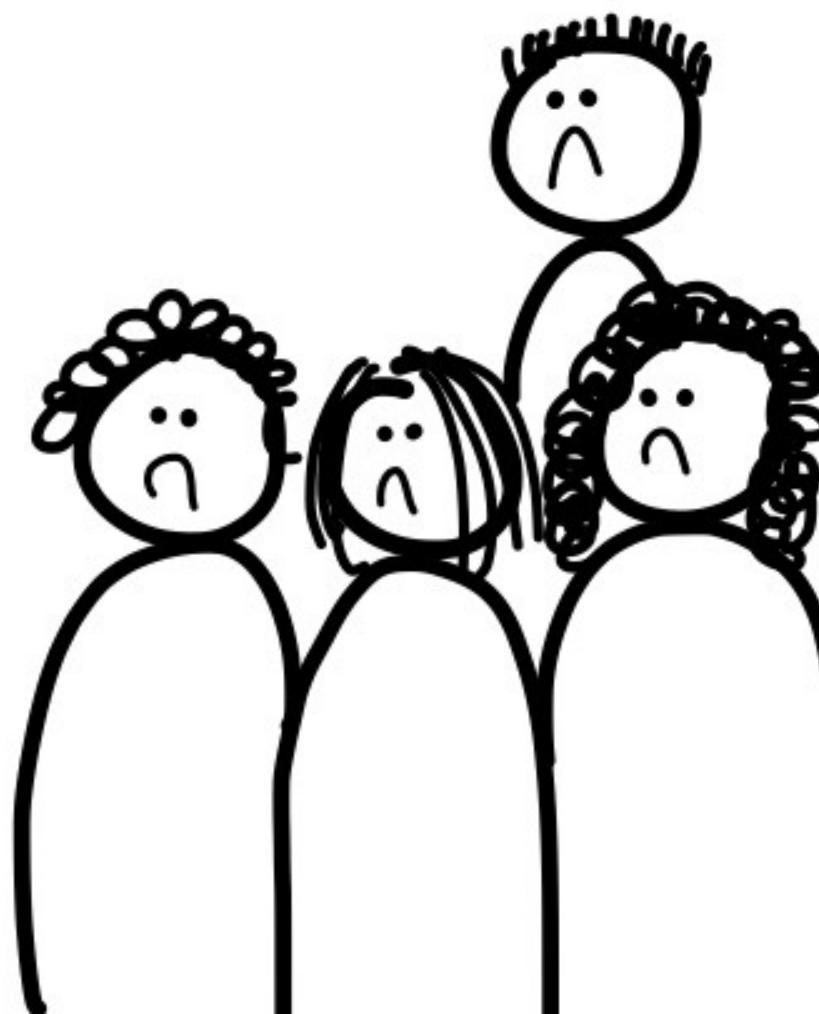
Benefits, challenges & improvements



```
1 name is Hedy
2 print welcome, name
3
```

The code you entered is not valid Hedy code. There is a mistake on line 2, at position 14. You typed a comma, but that is not allowed.

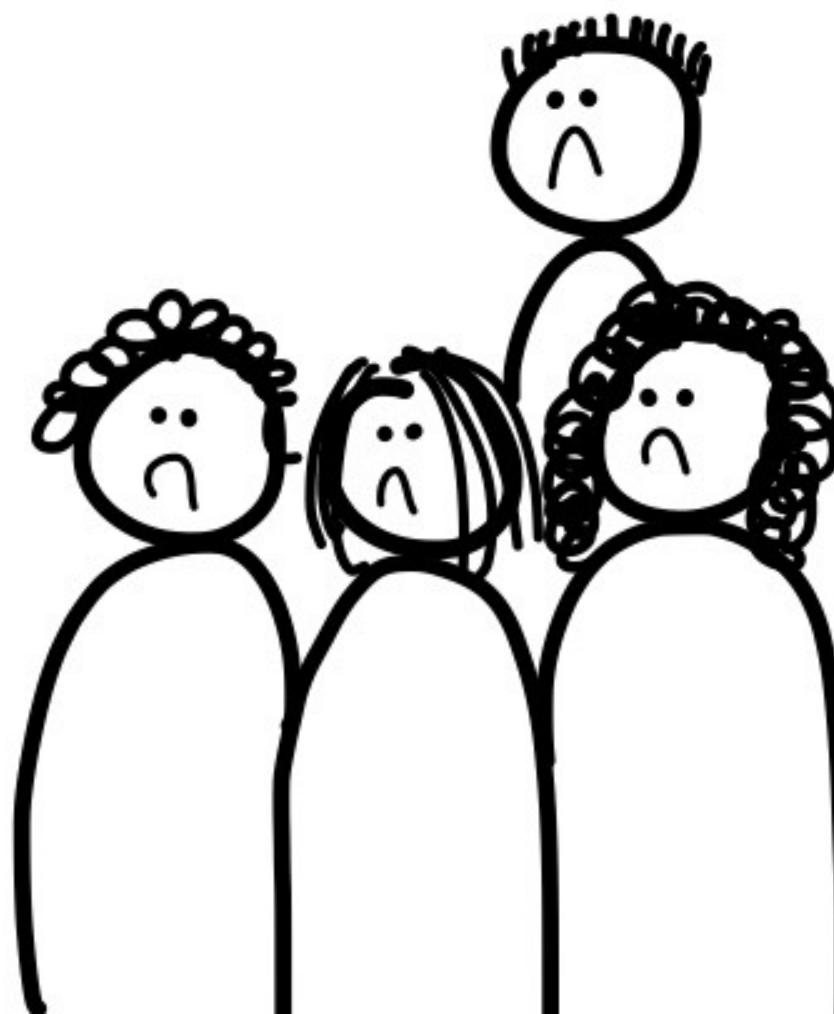
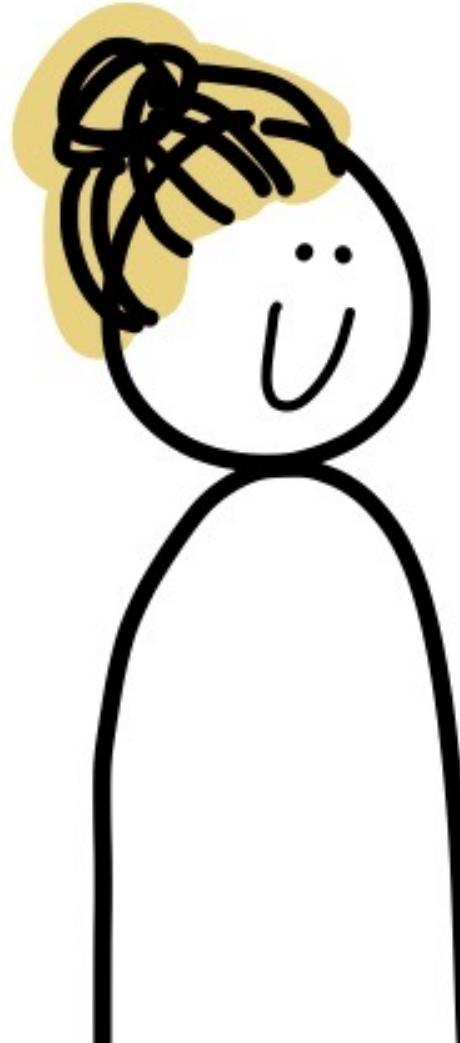
Better error messages



Benefits, challenges & improvements

Better error
messages

Dutch keywords

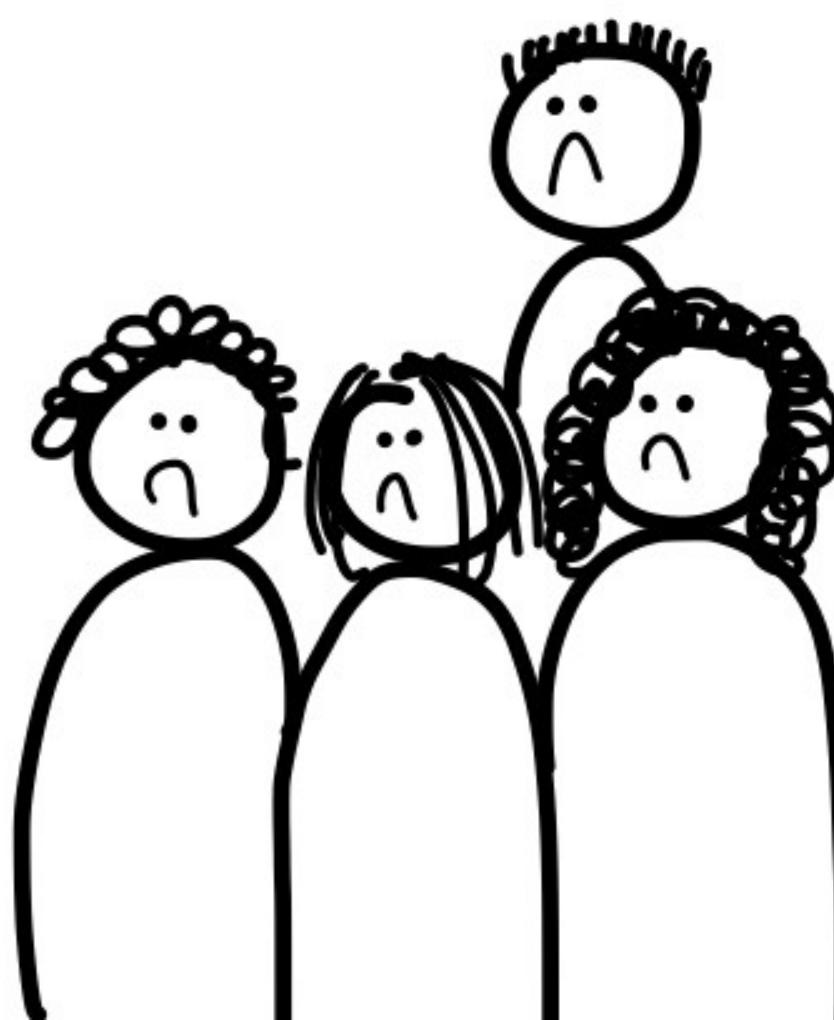


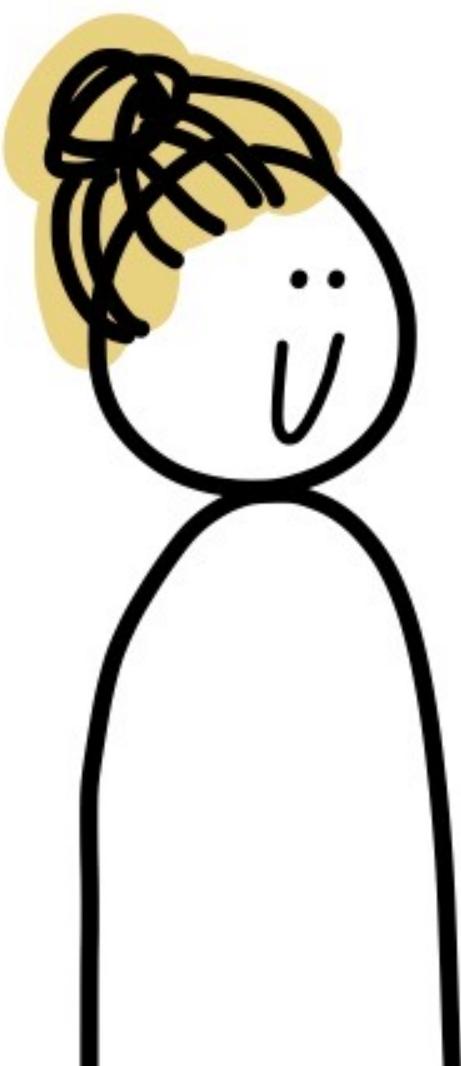
Benefits, challenges & improvements

Really?

Better error
messages

Dutch keywords



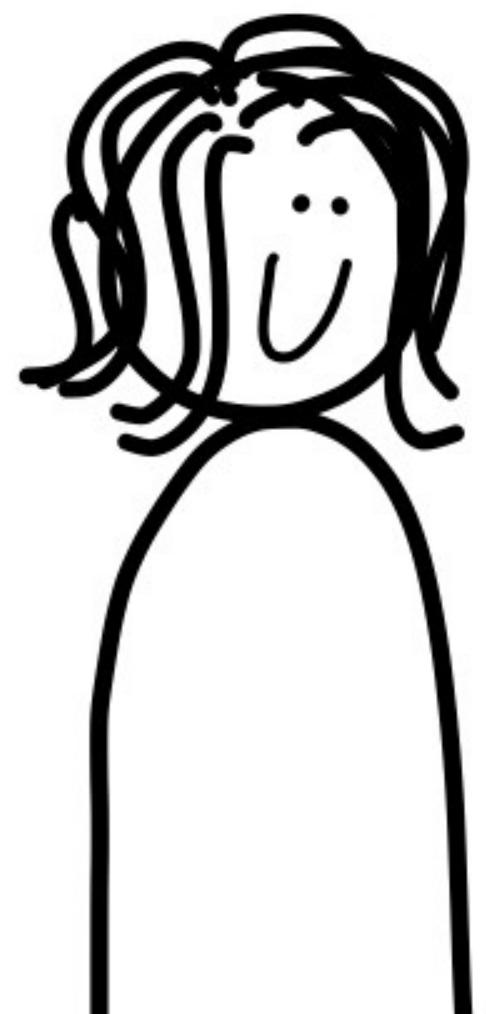


Let's look at
a demo!

Hedy is gradual

multi lingual ✓

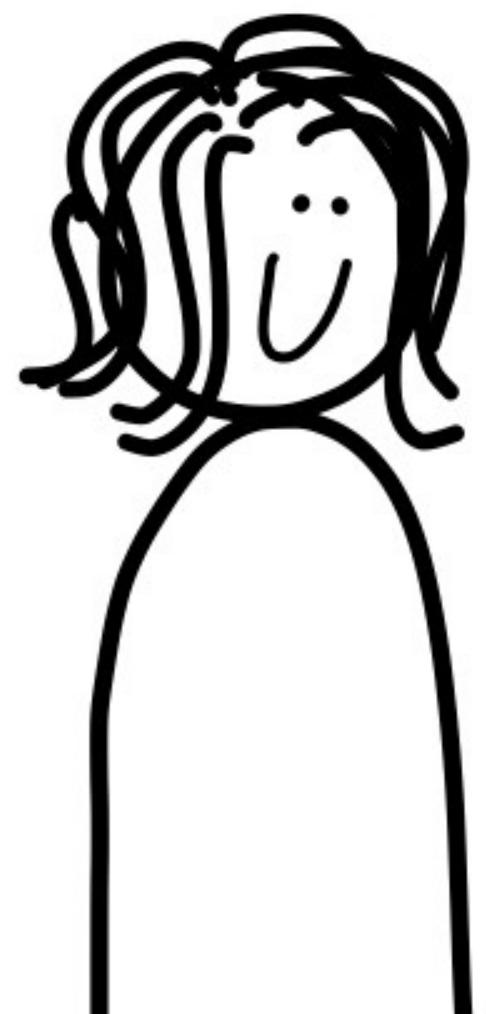
built for teaching



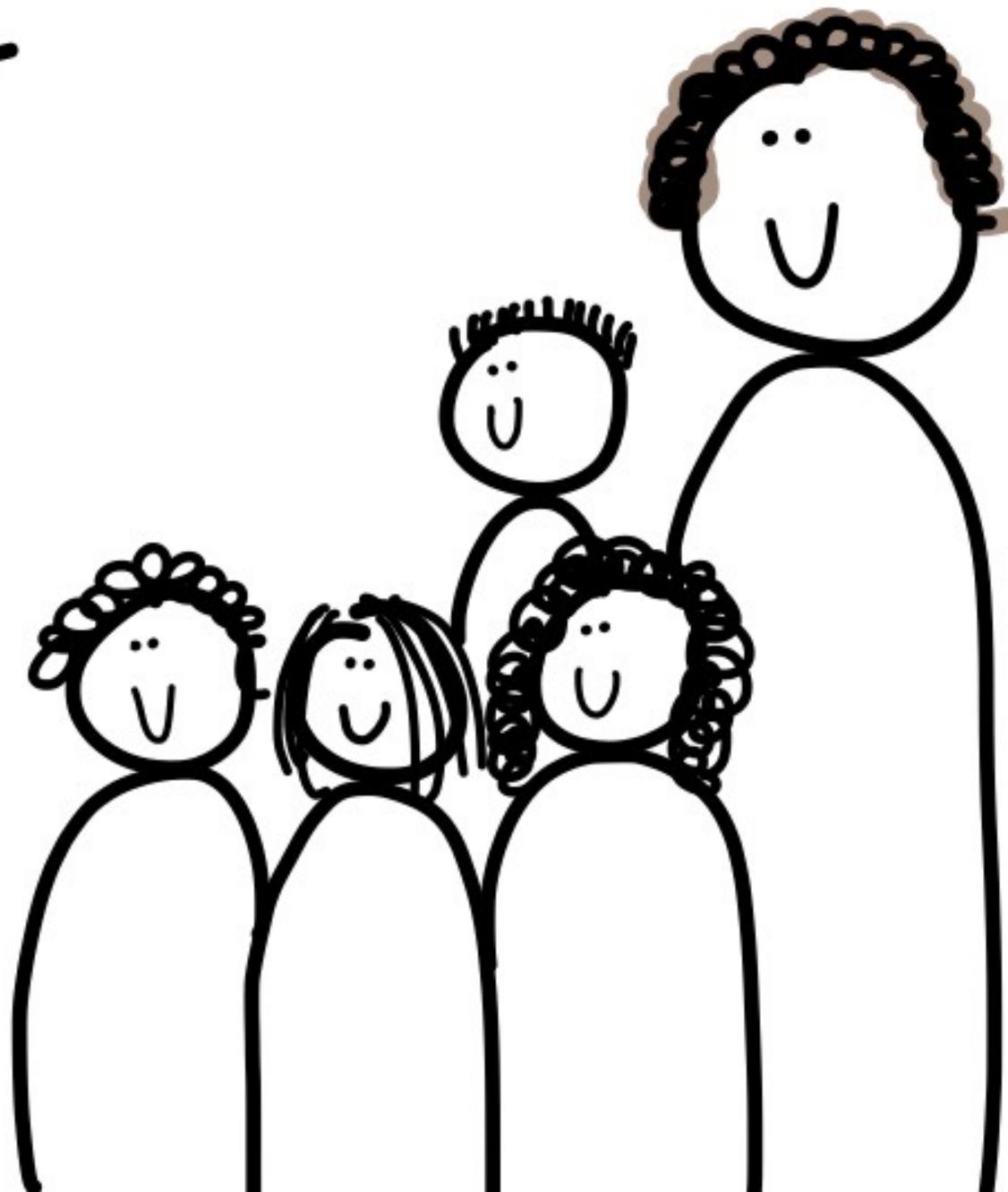
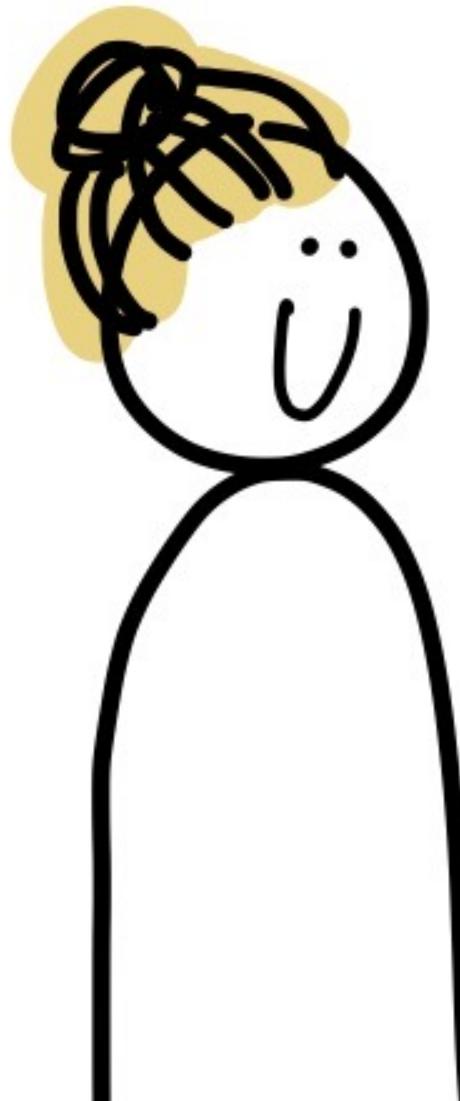
Hedy is gradual

multi lingual

built for teaching

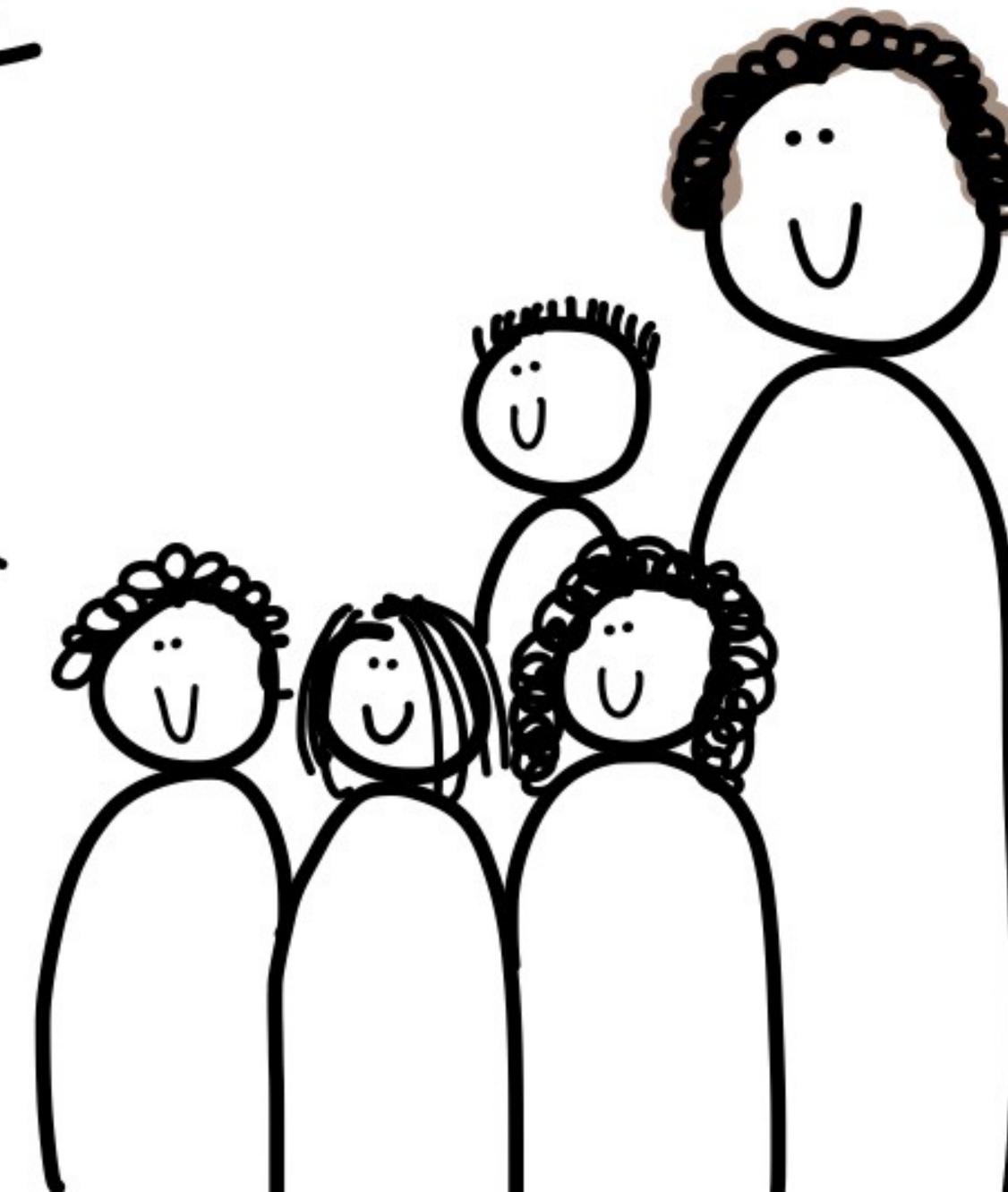
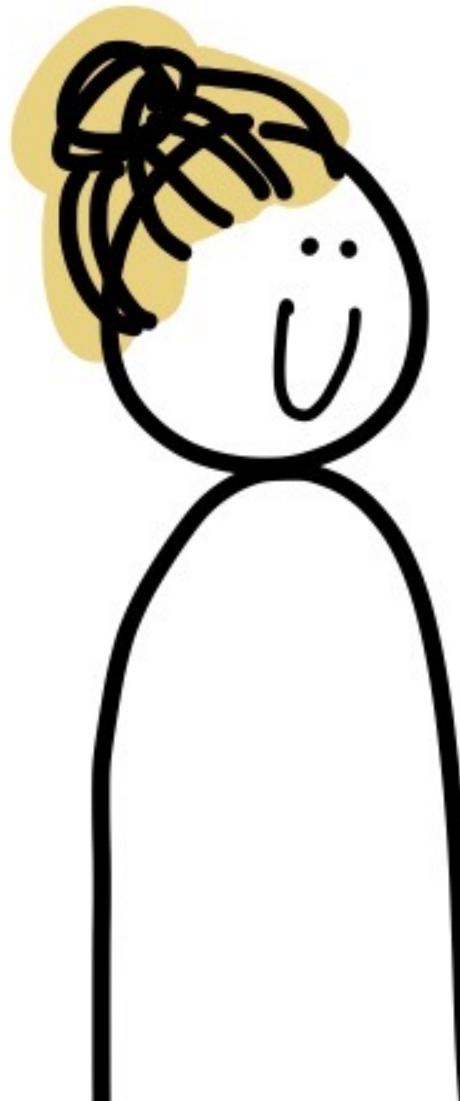


The levels are
a step by step guide



Less overwhelming for
me too!

The levels are
a step by step guide



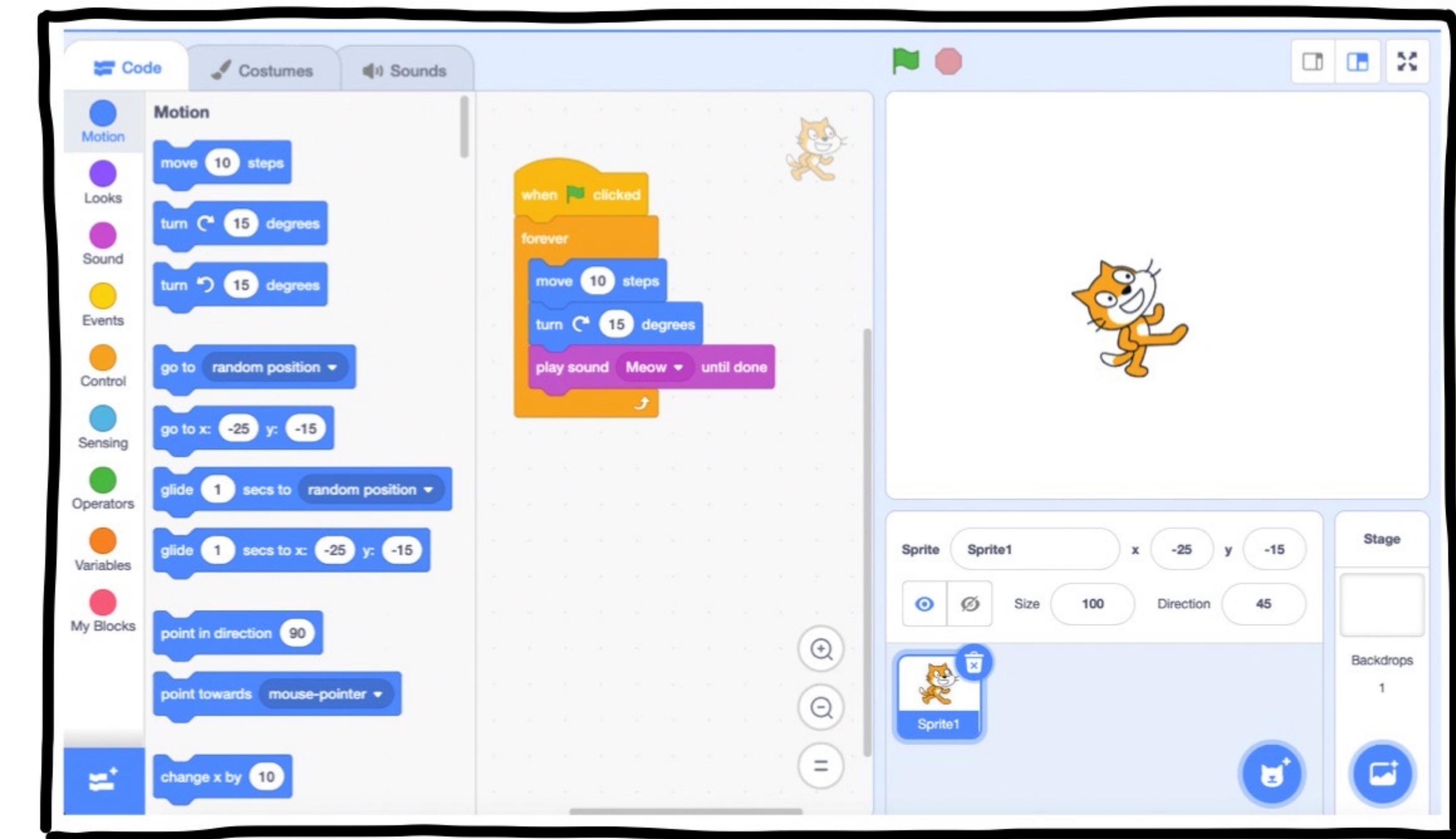
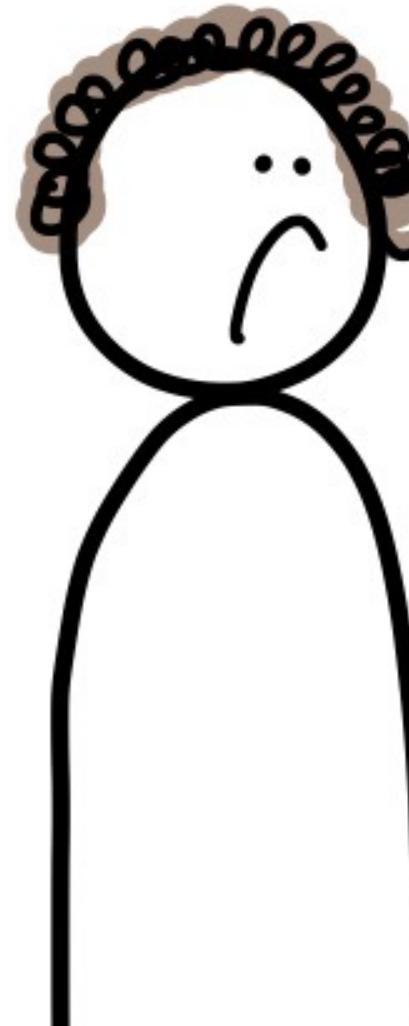
The Scratch interface is shown with the following details:

- Code Tab:** Selected tab.
- Motion Category:** Contains blocks for movement and rotation.
- Scratch Stage:** Shows a cat sprite running across the stage.
- Script Area:** A script is attached to the cat sprite:

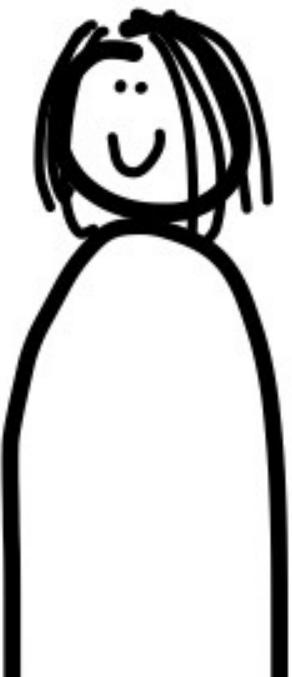
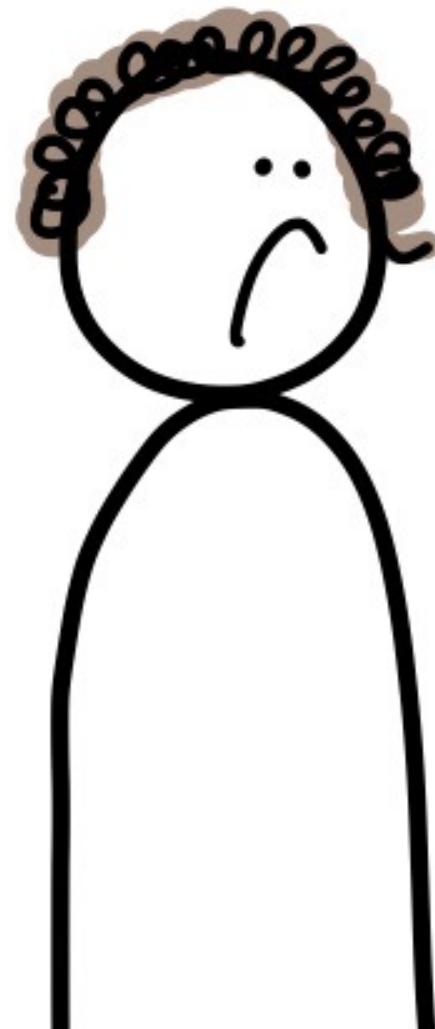
```
when green flag clicked
forever
  move (10) steps
  turn (15) degrees
  play sound [Meow v] until done
```
- Properties Panel:** Shows the cat's properties: **Sprite** (Sprite1), **x** (-25), **y** (-15), **Size** (100), and **Direction** (45).
- Stage Panel:** Shows the current backdrop (1).



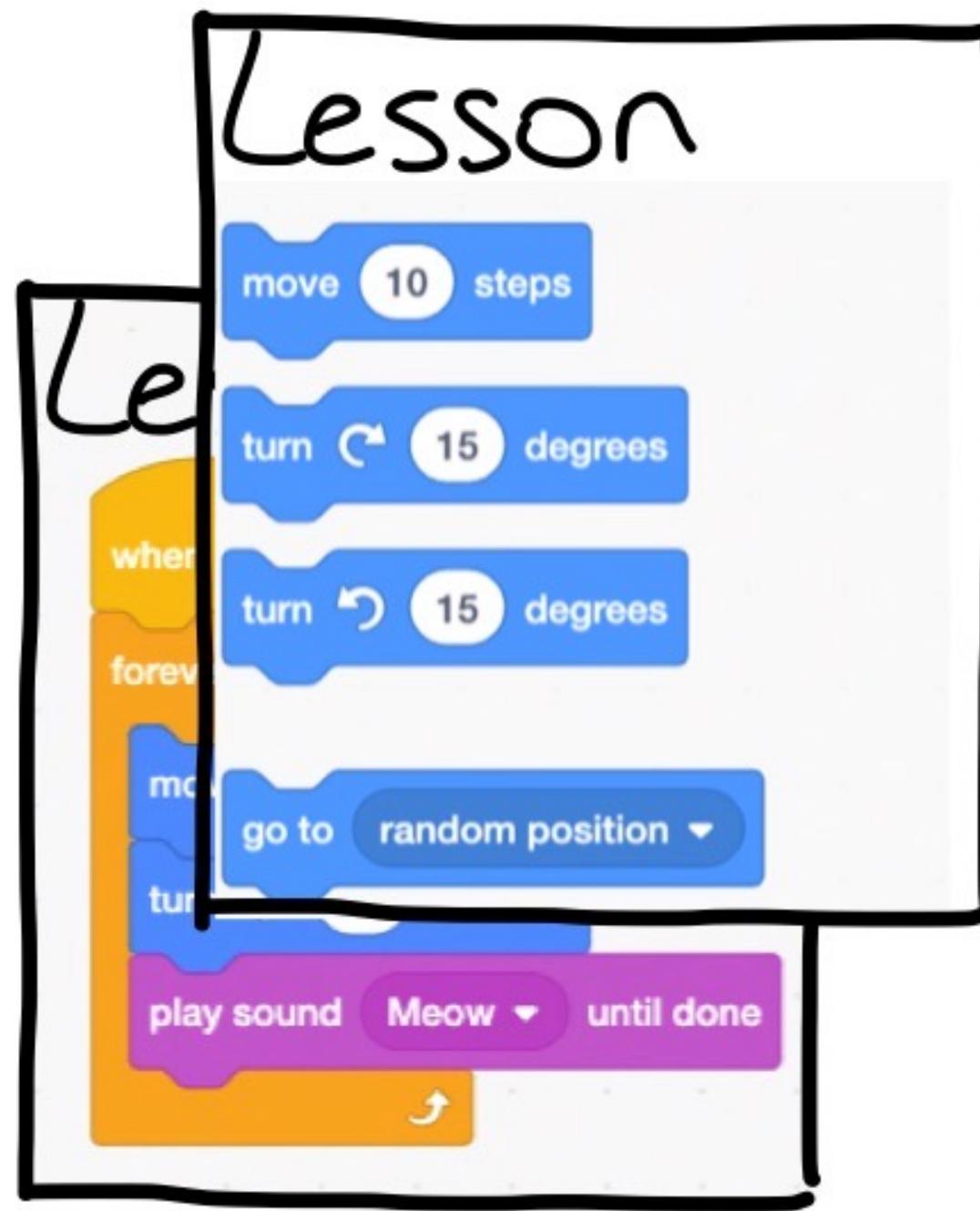
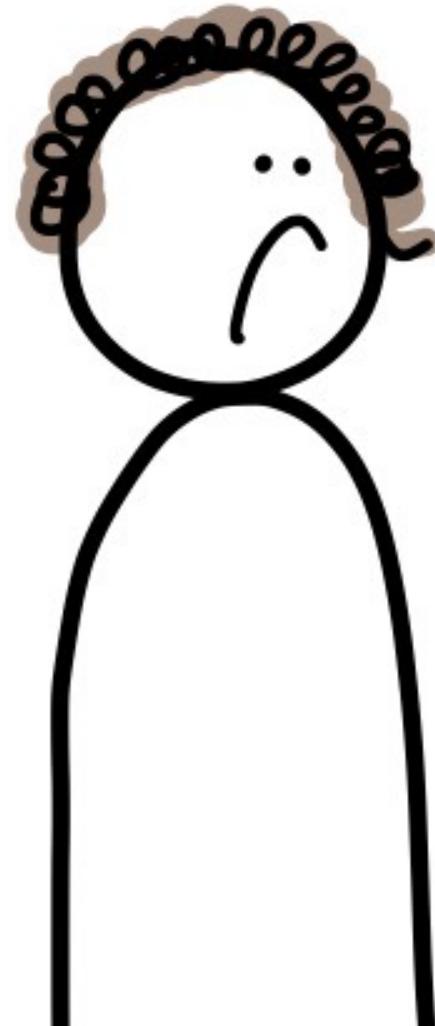
What can my Students build?



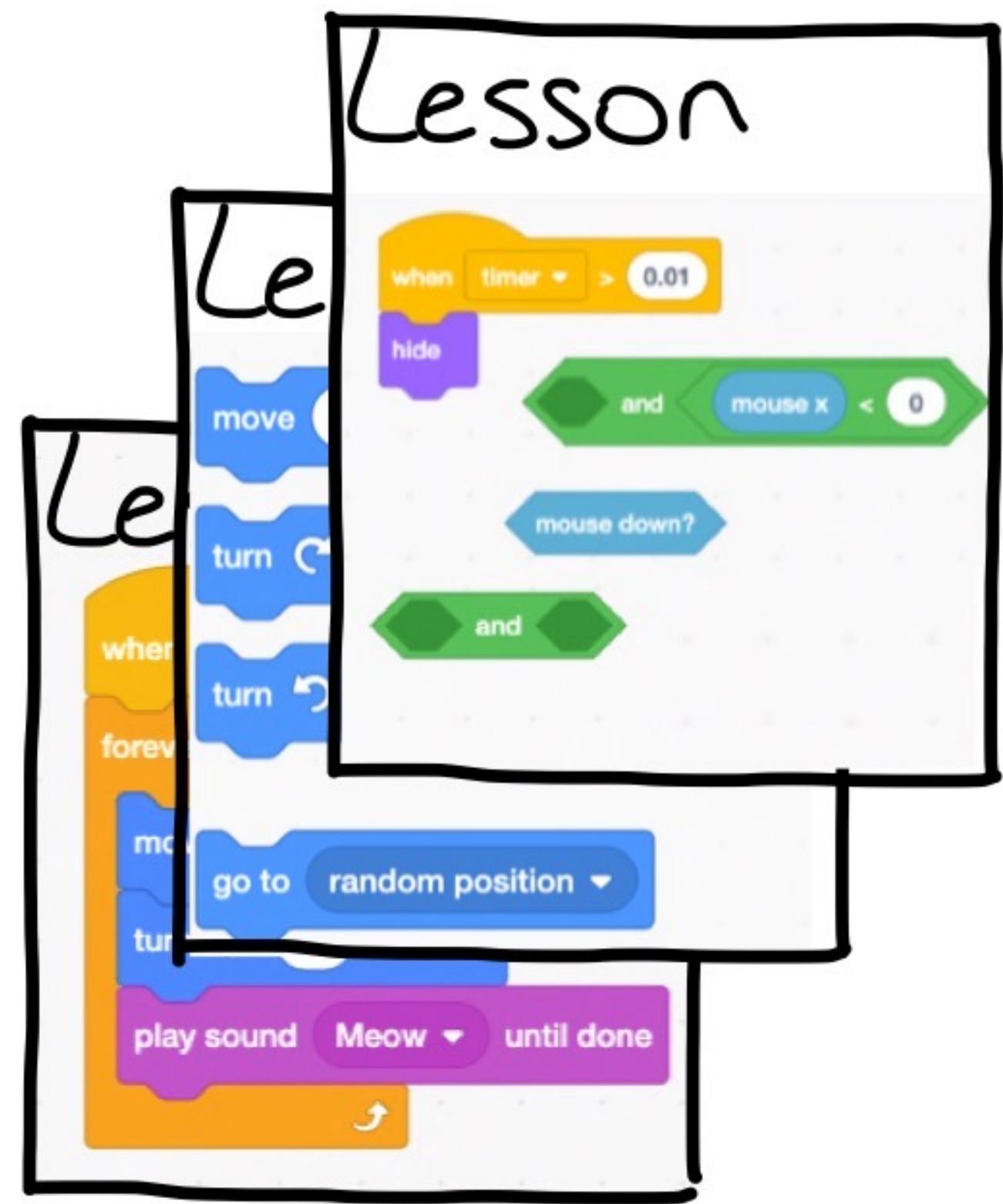
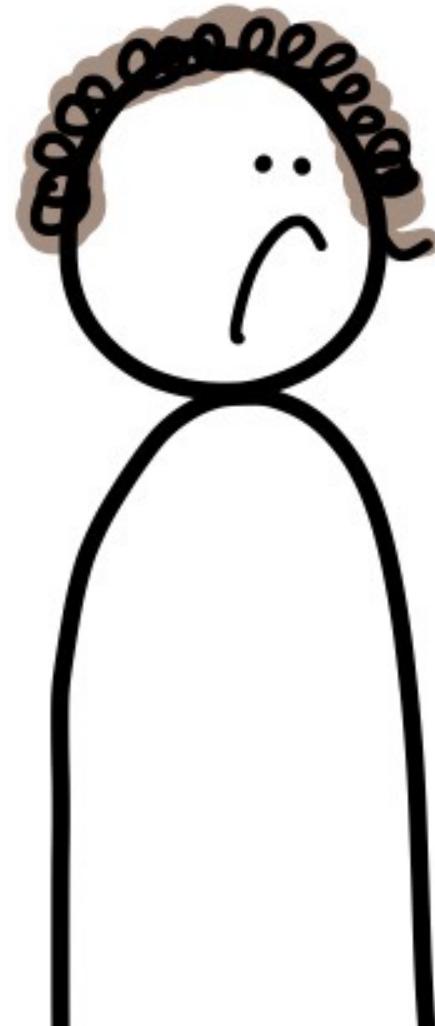
What can my Students build?



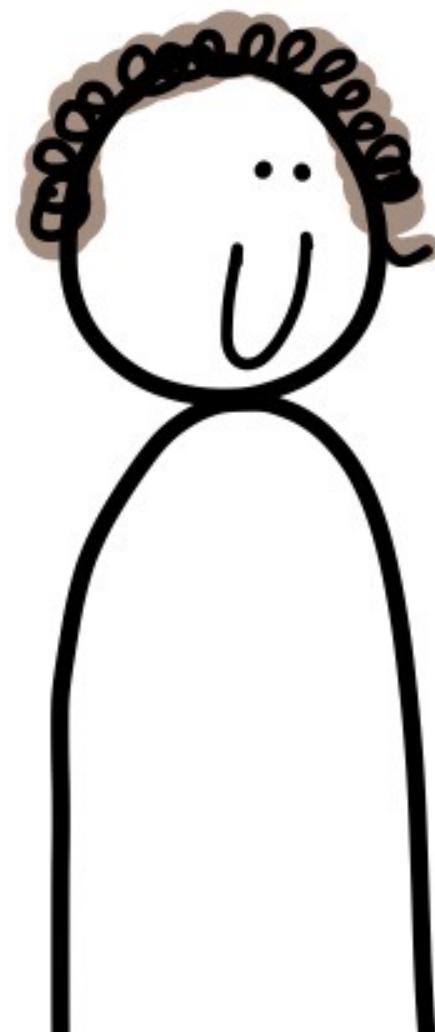
What can my Students build?



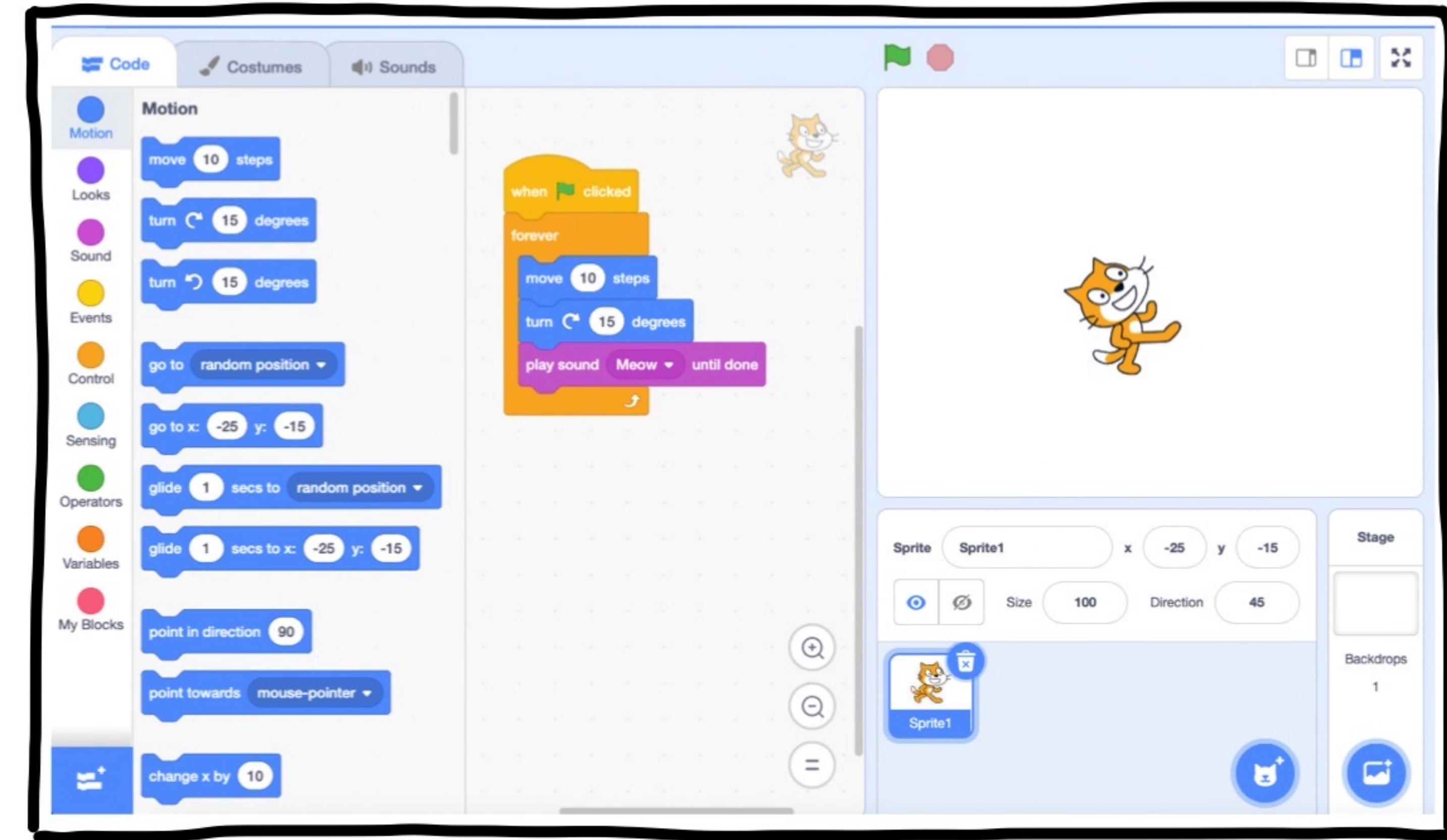
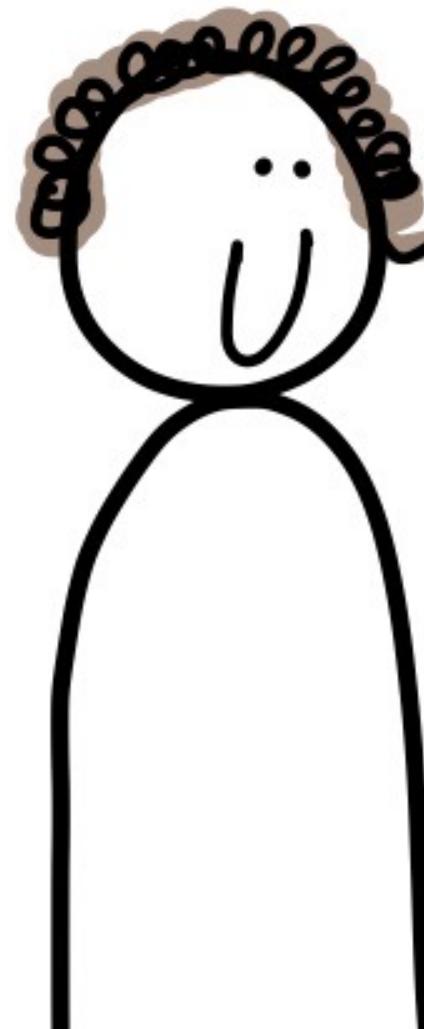
What can my Students build?



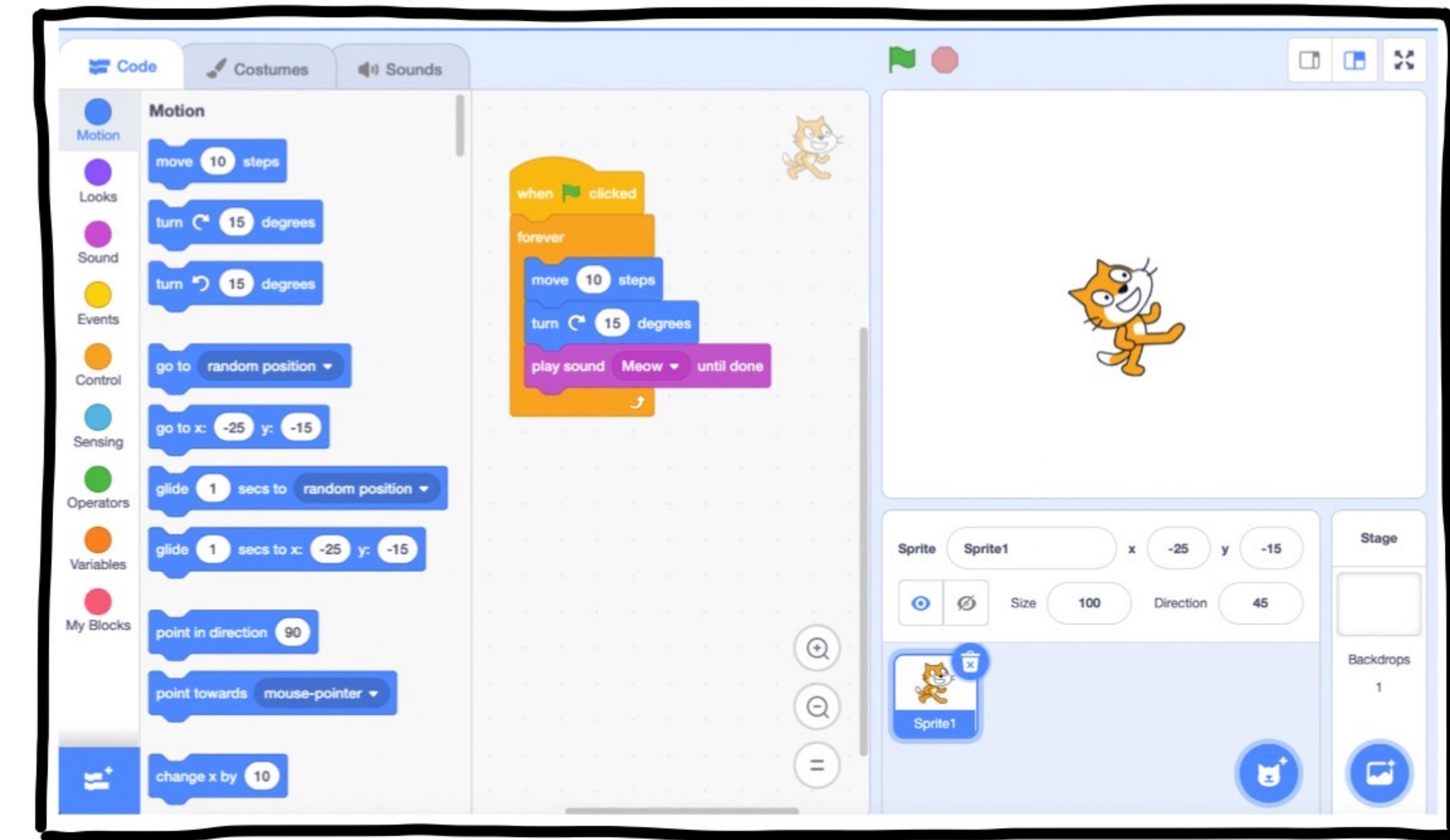
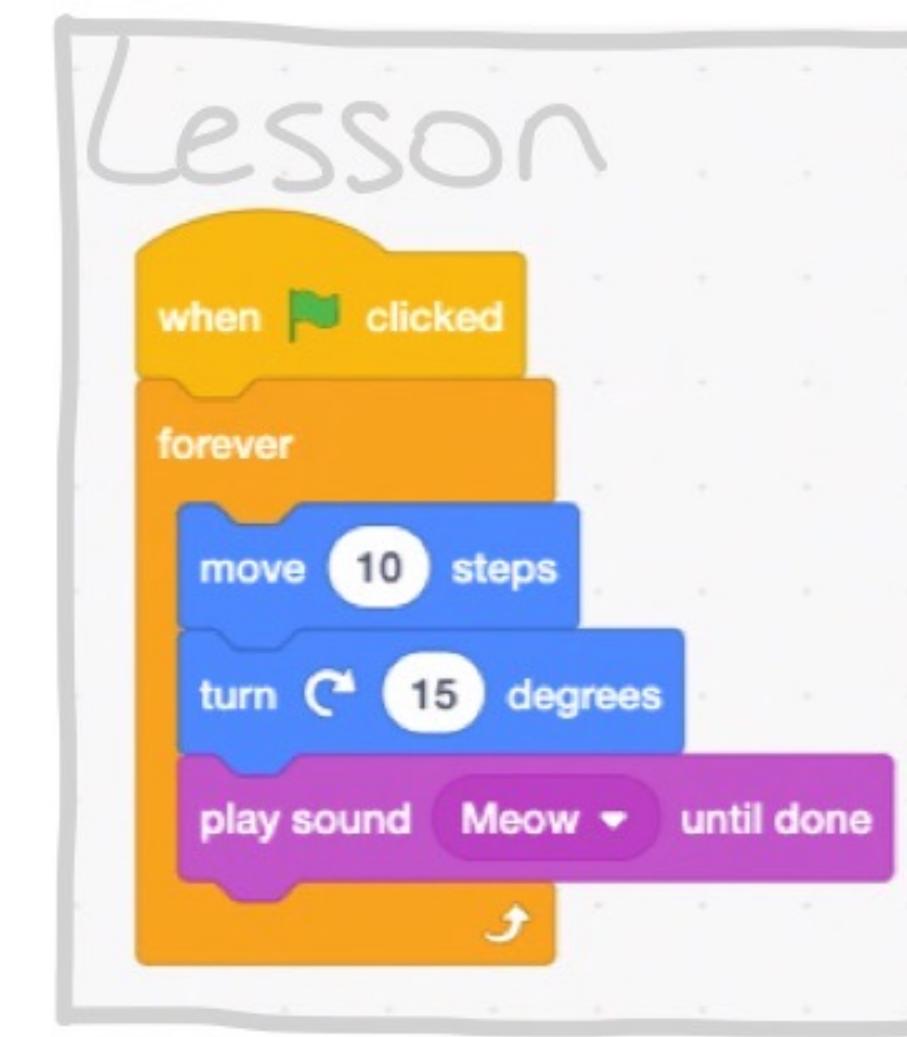
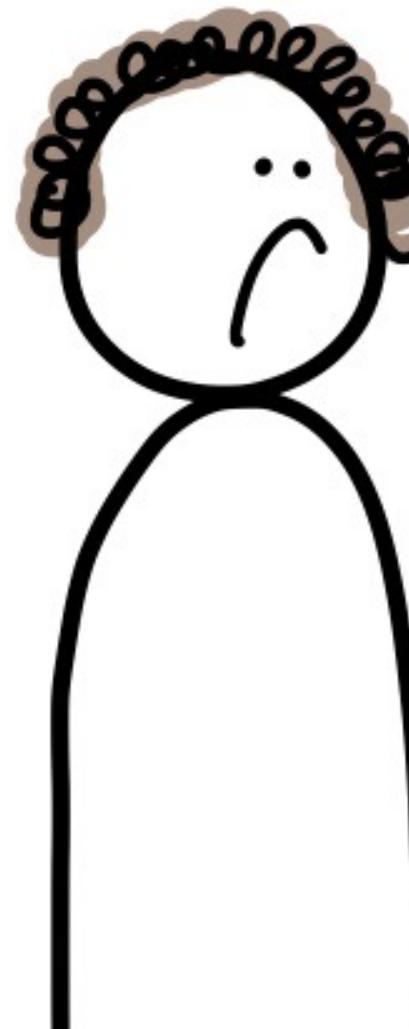
What can my Students build?



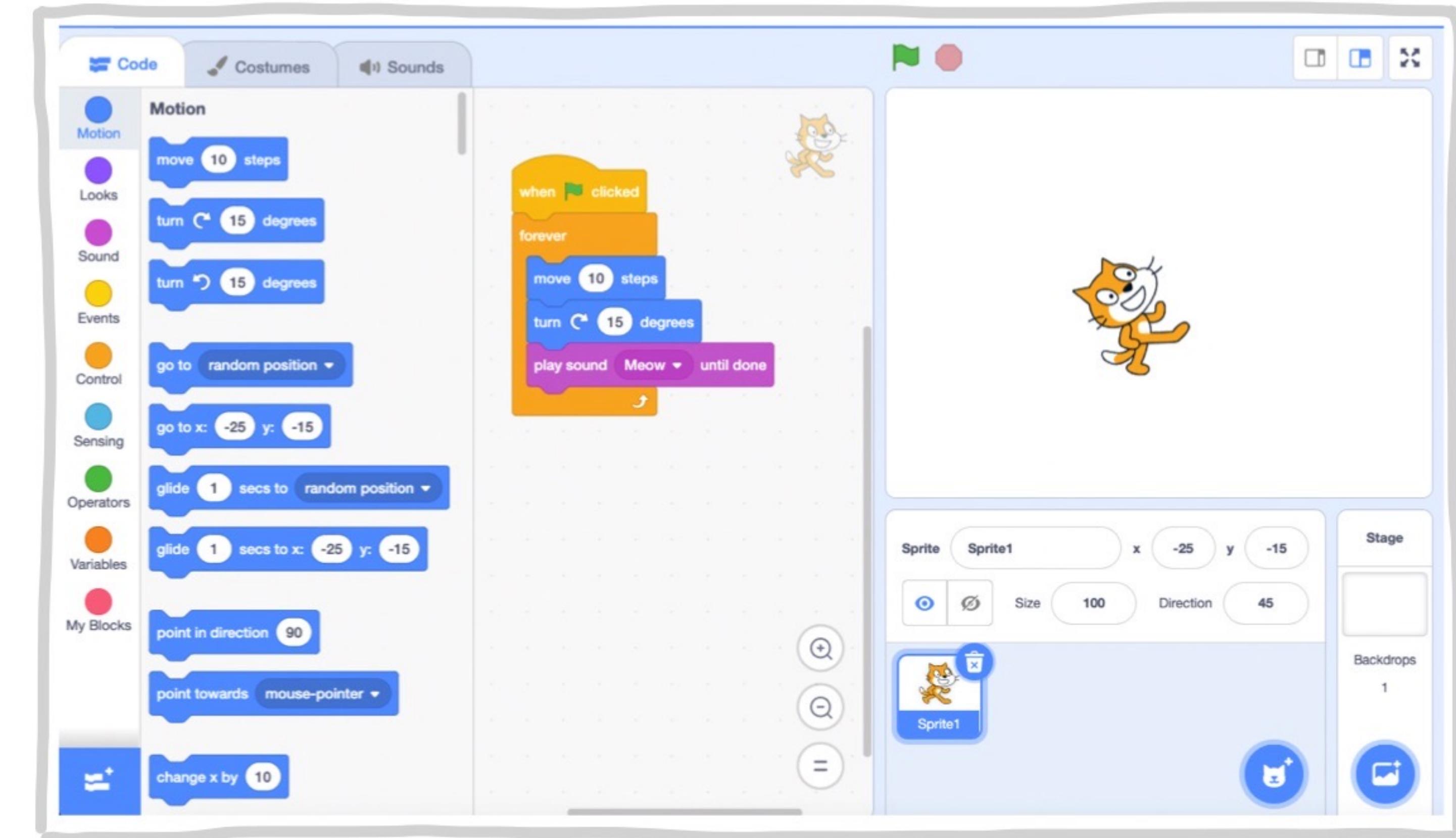
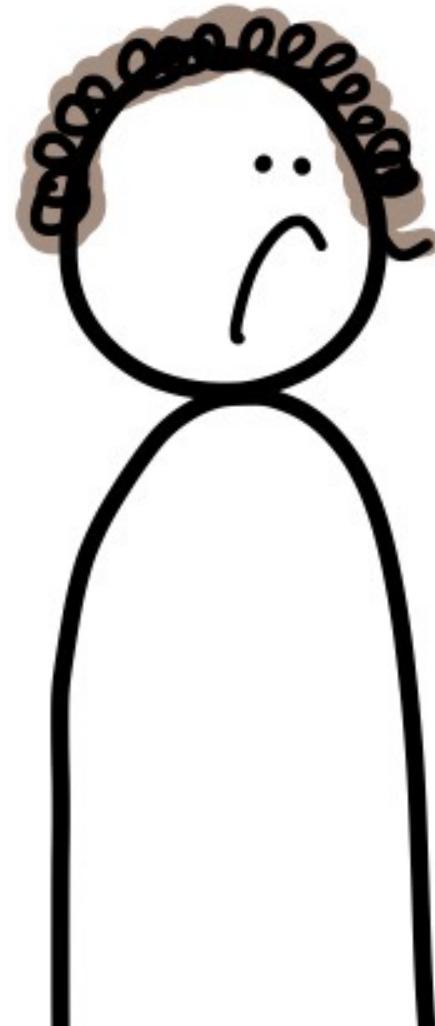
What can my Students build?



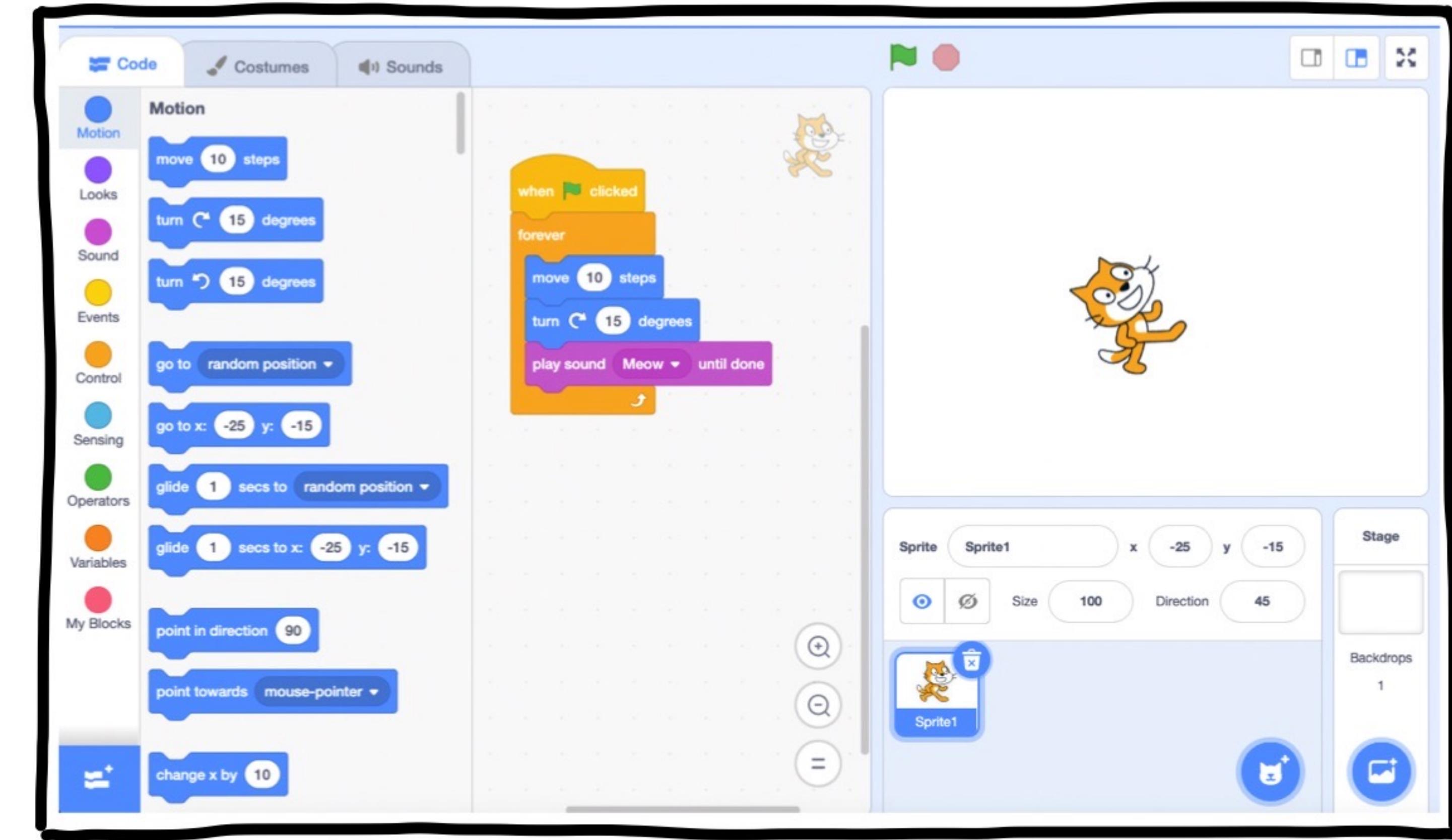
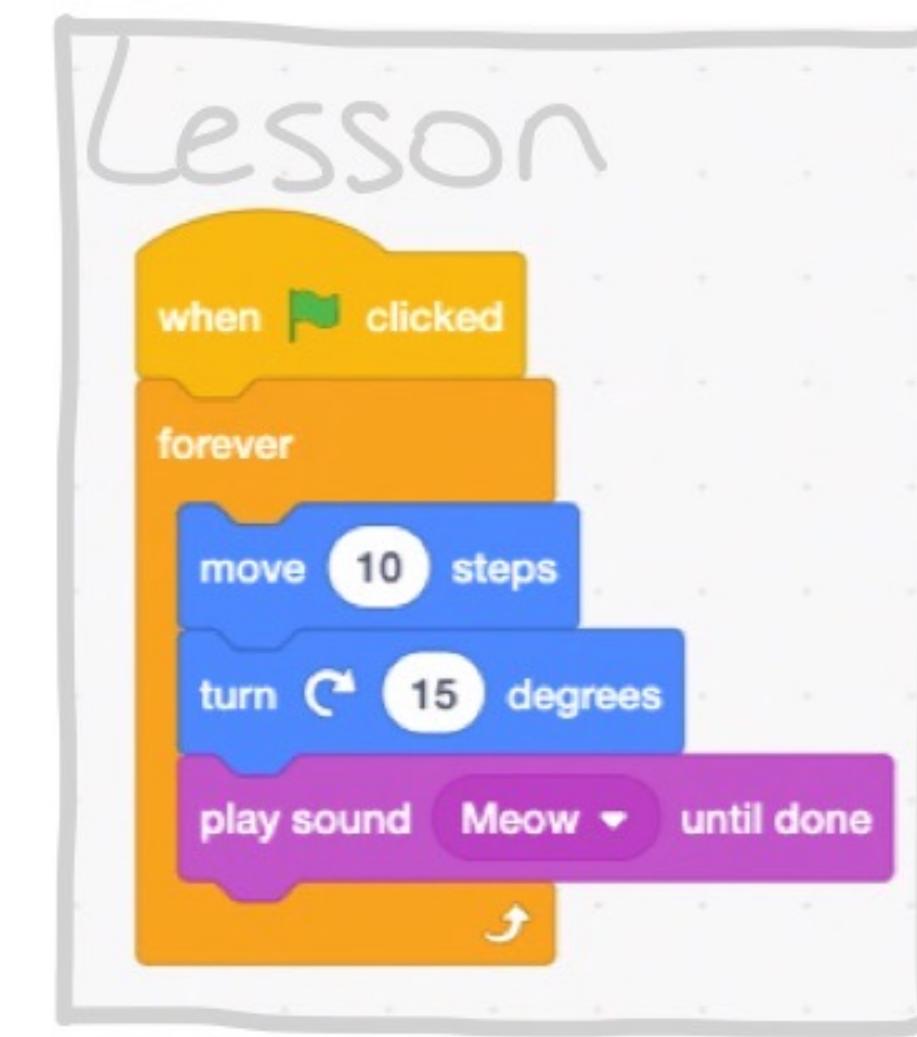
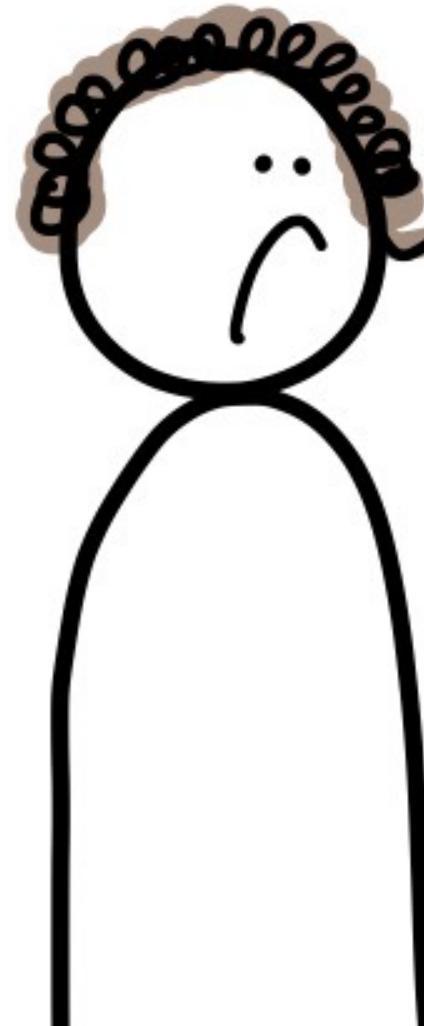
What can my Students build?



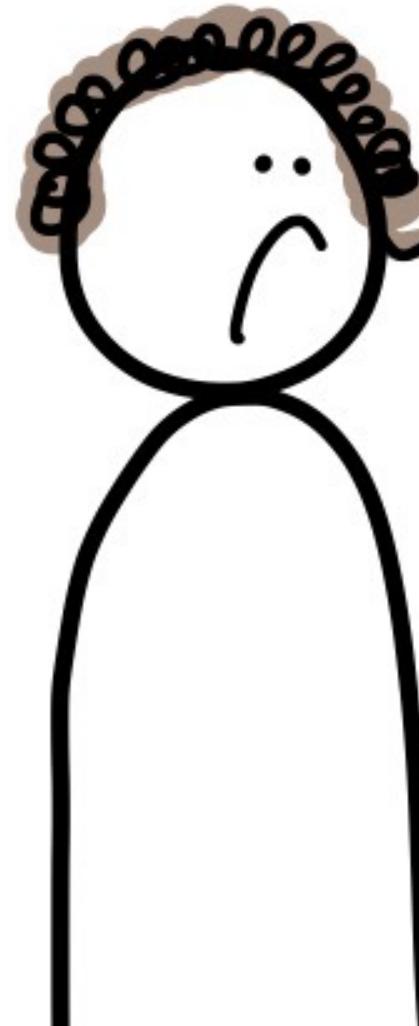
What can my Students build?



What can my Students build?



What can my Students build?



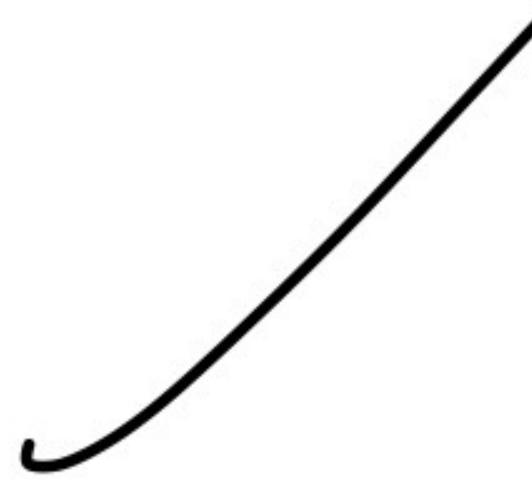
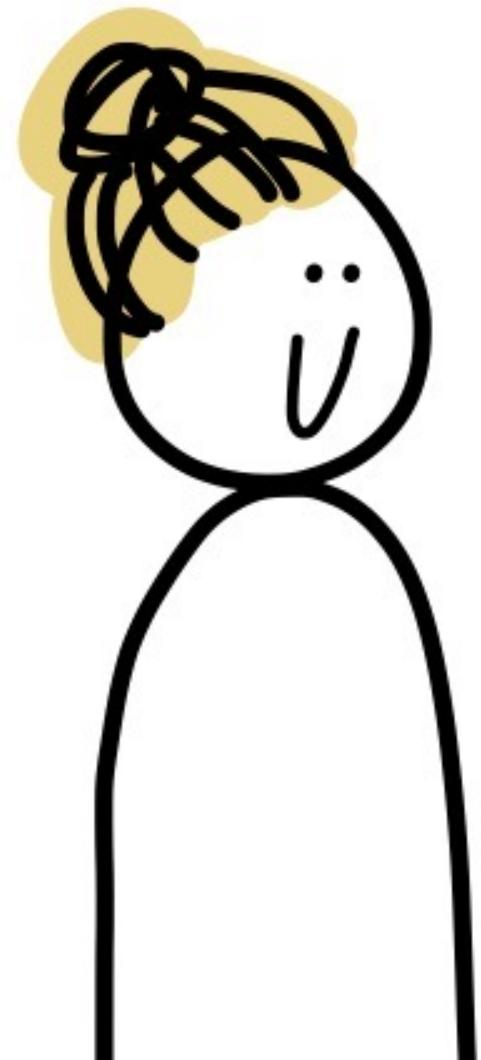
The screenshot shows the Scratch interface with a cat sprite on the stage. The script for the cat is:

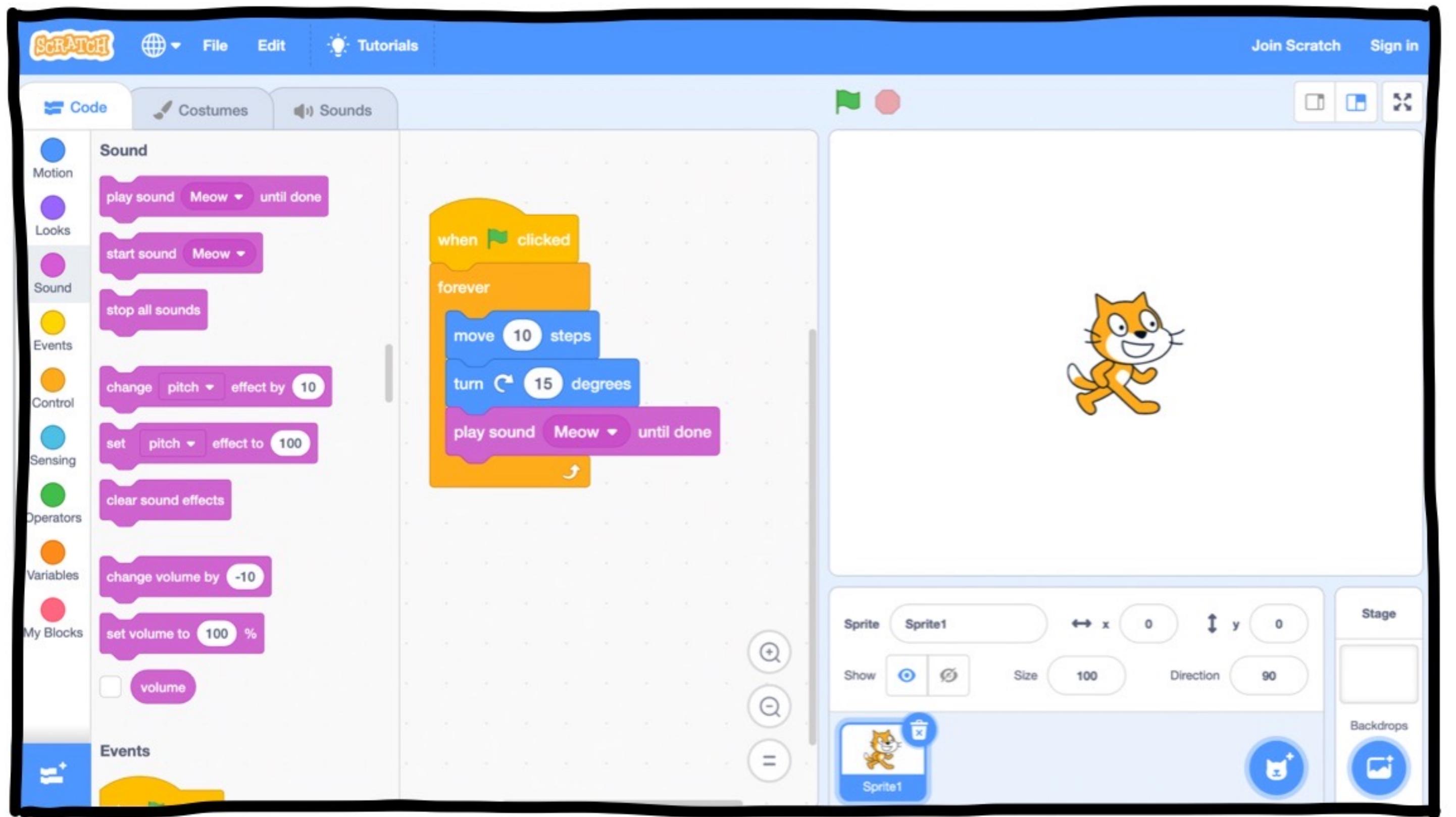
```
when green flag clicked
forever
  move (10) steps
  turn (15) degrees
  play sound [Meow v] until done
```

The stage and sprite properties are also visible.

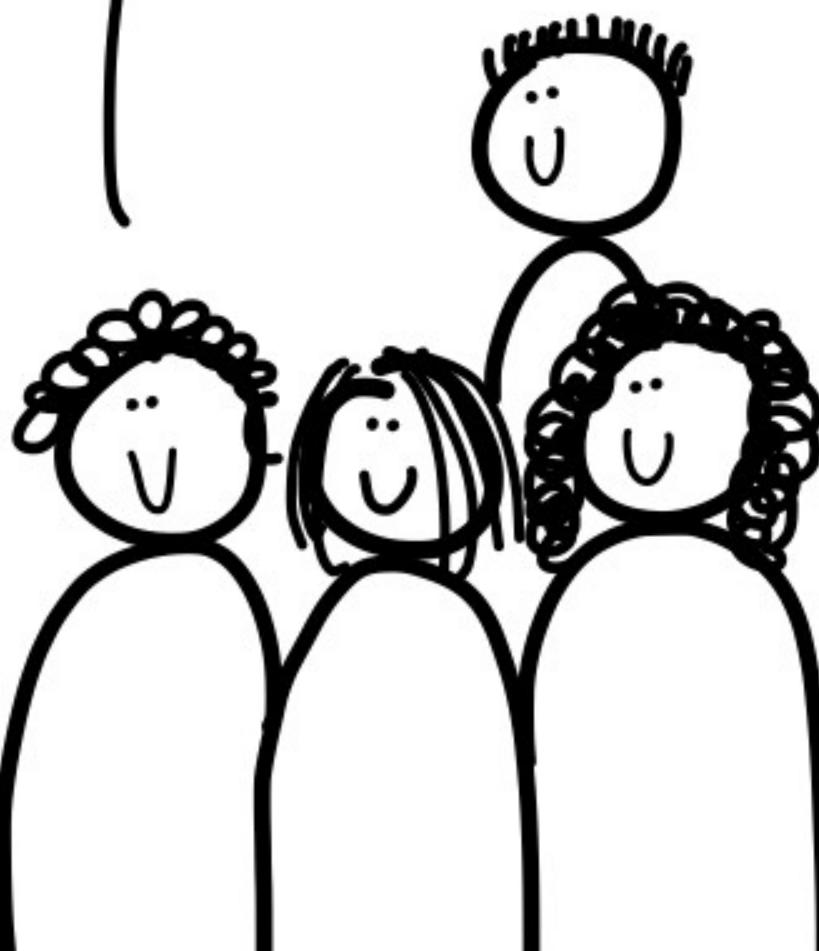


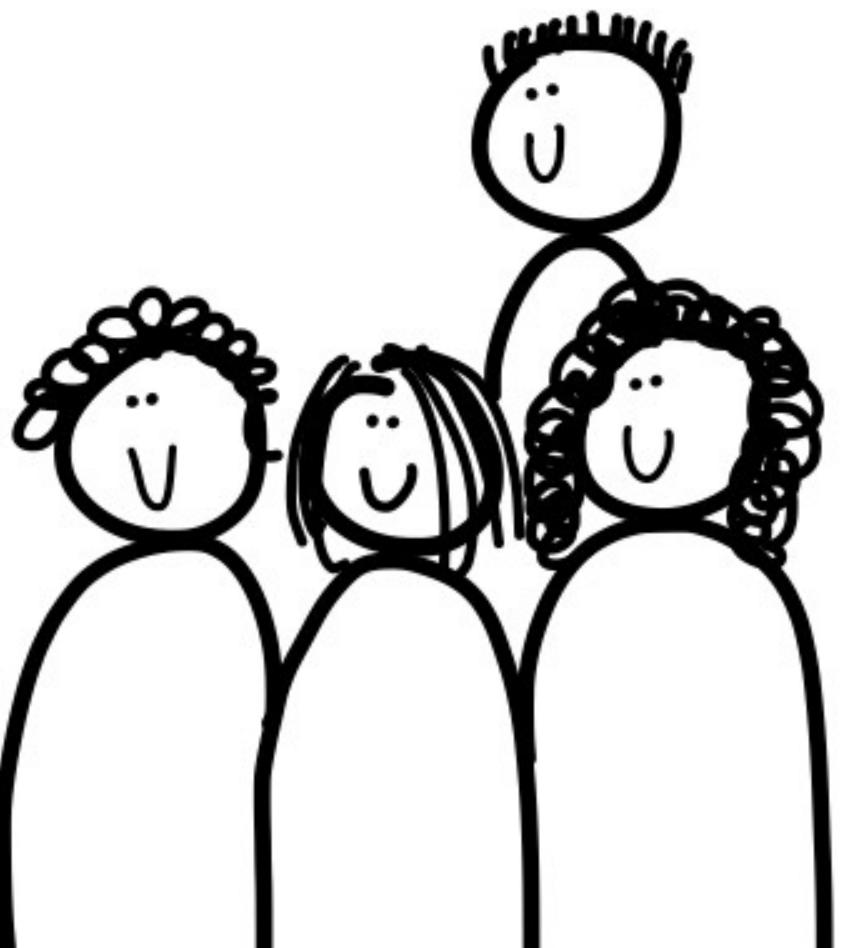
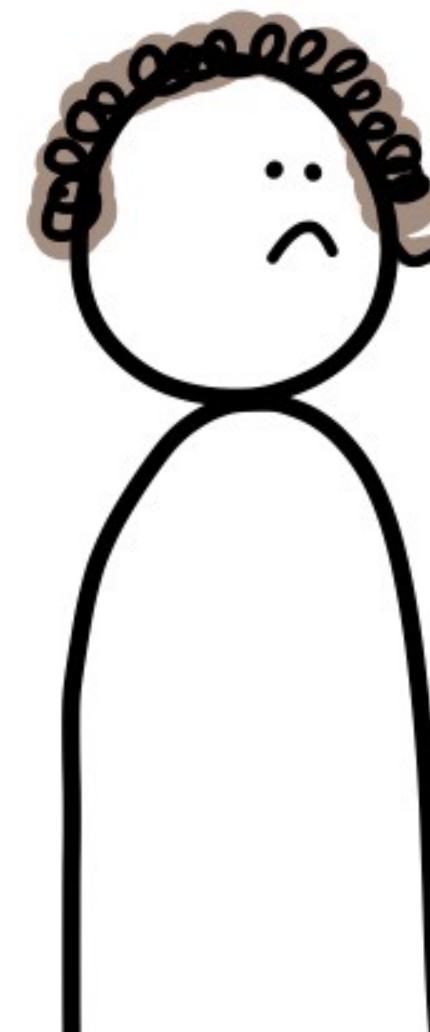
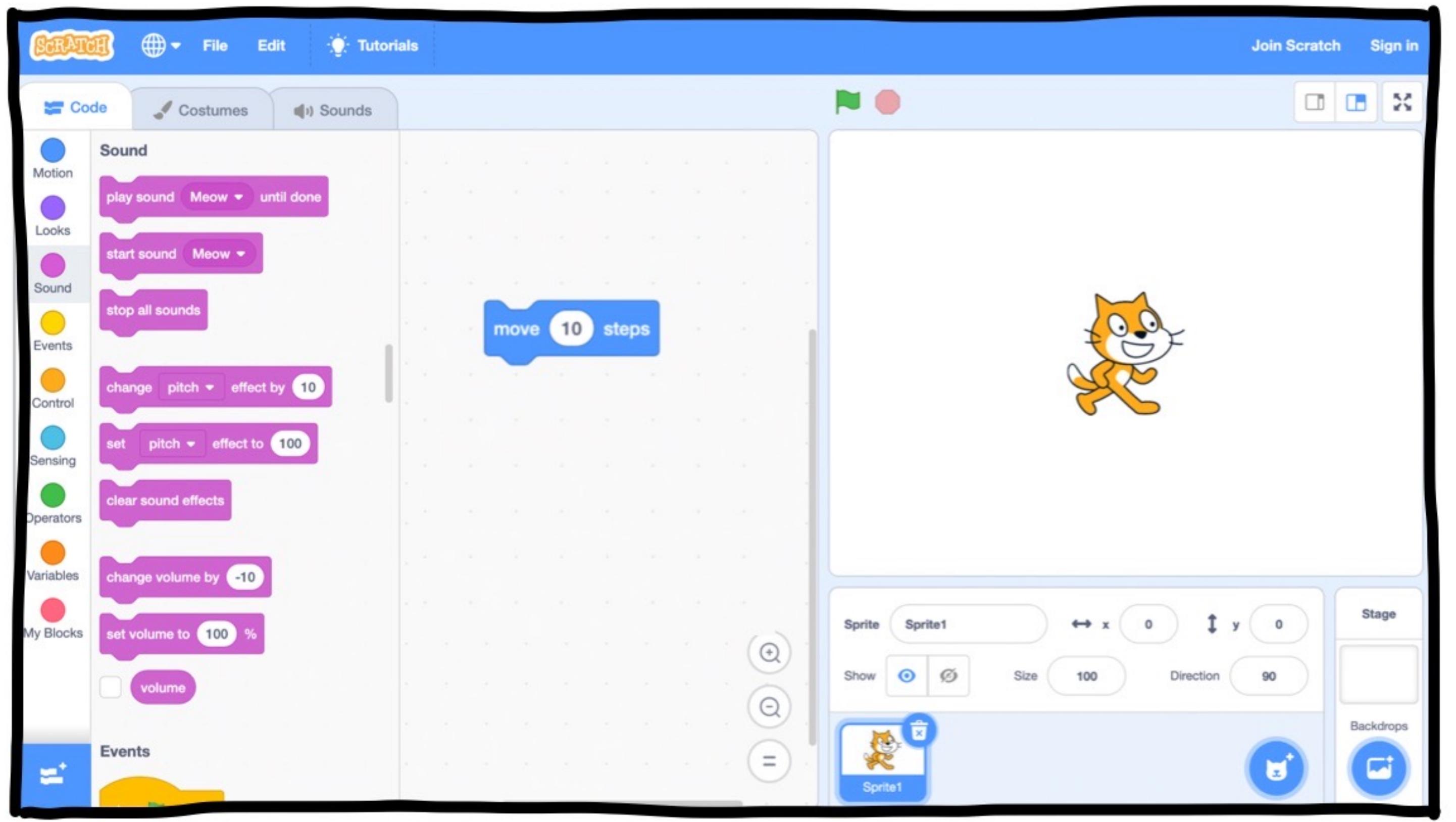
Adventures





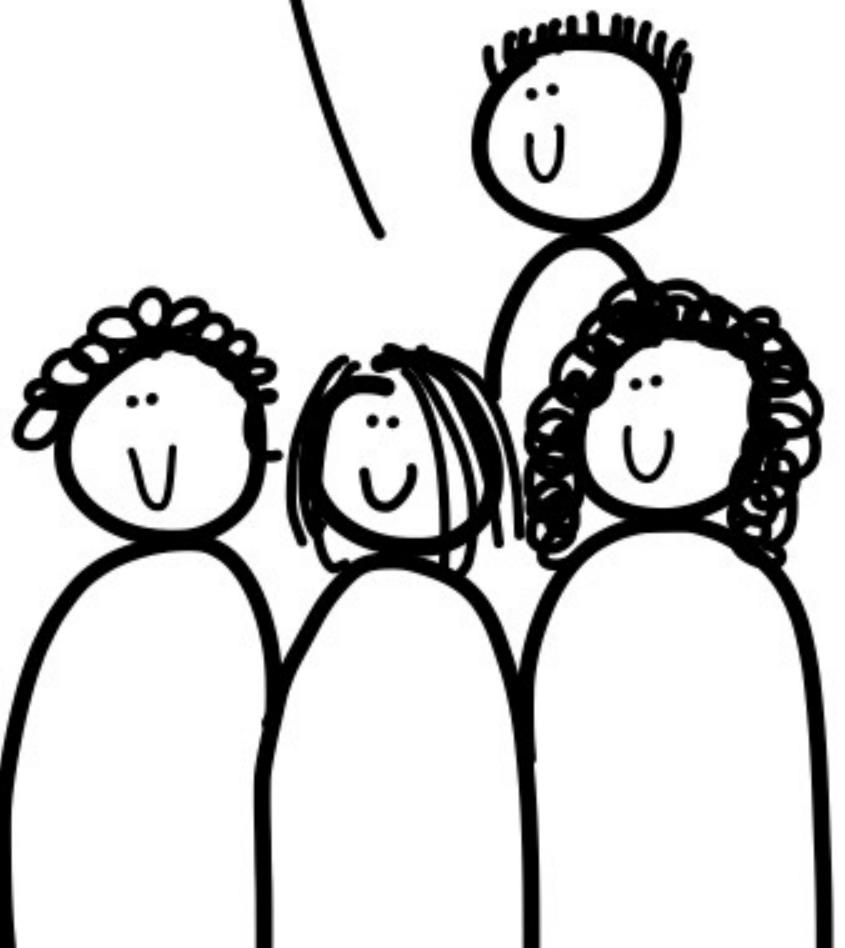
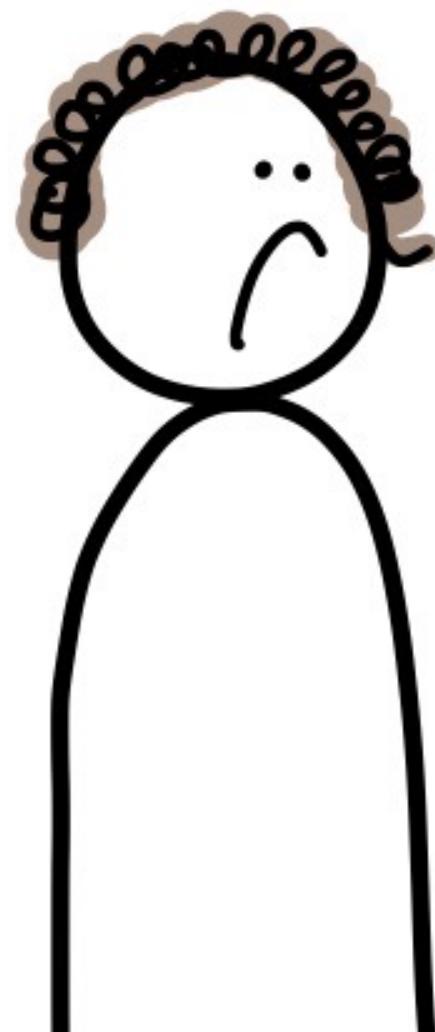
I did it!



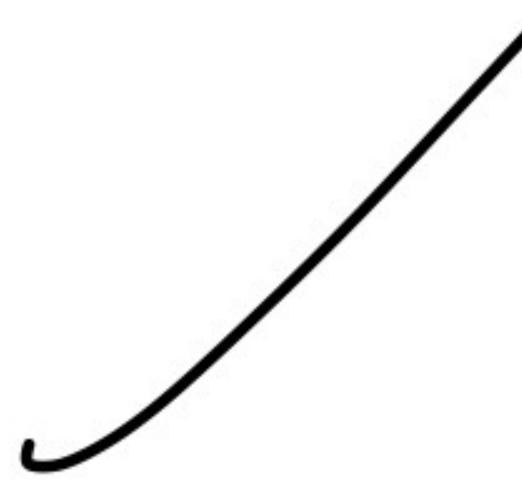
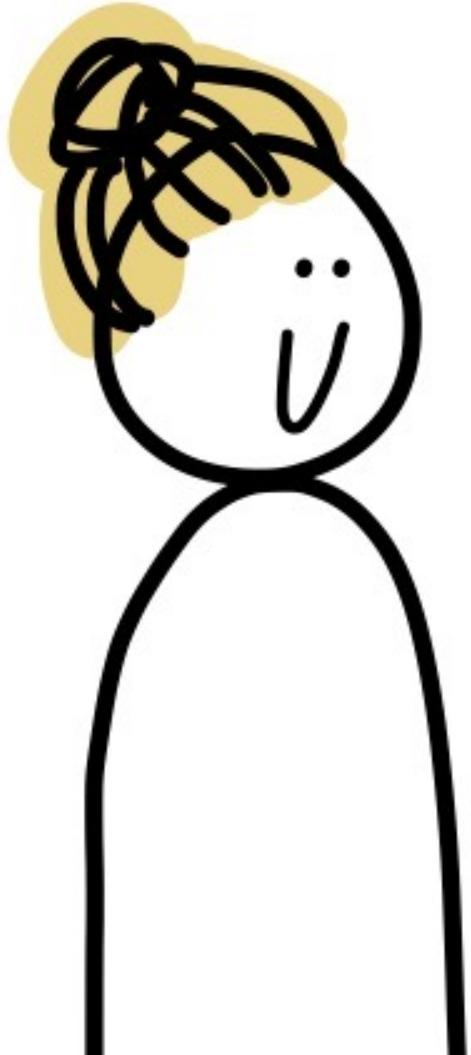


The image shows the Scratch programming environment. The stage background is titled "Robin Hood || An Adventure Part 5: Illusion Island". On the stage, there is a green knight-like sprite and a palm tree. The script area contains two scripts for the knight. The left script starts with "when green flag clicked" and includes blocks for "play sound Meow until done", "switch costume to: Introduction v.", "go to x: 0 y: 0", and a "forever" loop with "if key [left arrow v pressed? or mouse down? then" and "repeat (10) [change x by -5]". The right script also starts with "when green flag clicked" and includes "go to front + layer", "reset timer", and a "repeat (18)" loop with "when I receive [Intro Ending v]" and "set ghost effect to 0". The script palette on the left lists categories: Motion, Looks, Sound, Events, Control, Sensing, Operators, Variables, and My Blocks. The Sound category is currently selected, showing blocks like "play sound Meow until done", "start sound Meow", "stop all sounds", and "change pitch effect by 10".

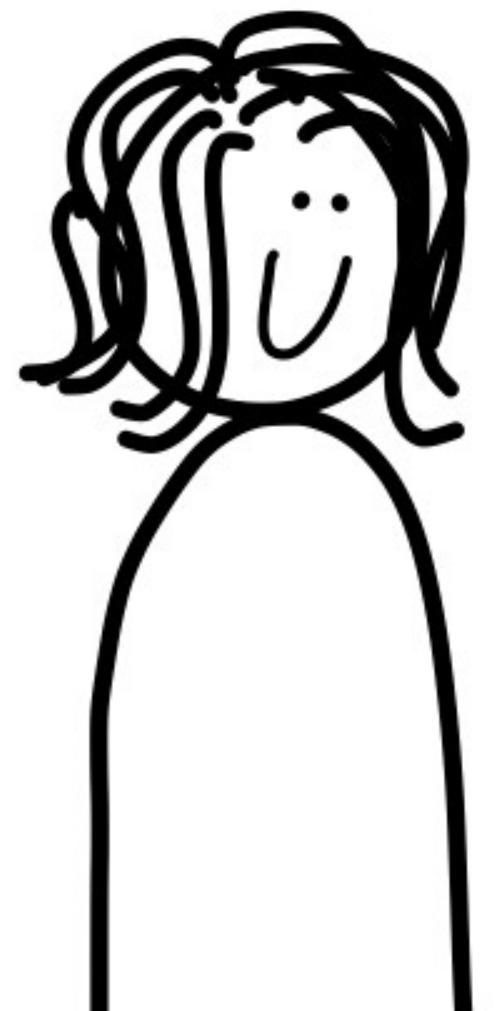
I built this!



Customization

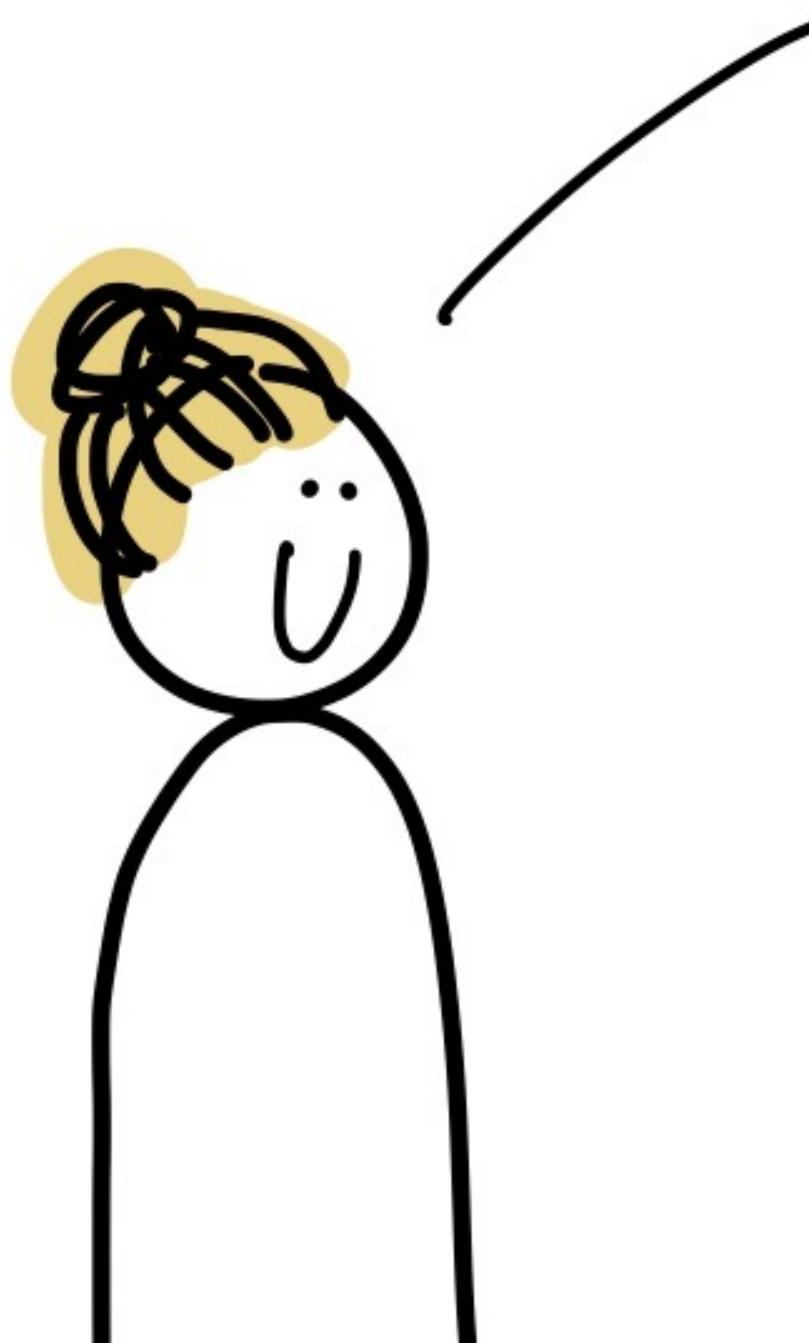


Hedy



Named after
Hedy Lamarr

We love your help!



Hedy is gradual



Want to work on an
innovative parser ?

Hedy is gradual



Journal of Computer Languages

Available online 20 September 2022, 101158

In Press, Journal Pre-proof



Design, implementation and evaluation of the Hedy programming language

Marleen Gilsing ^a, Jesús Pelay ^b, Felienne Hermans ^{a, c}

Show more

+ Add to Mendeley Share Cite

<https://doi.org/10.1016/j.col.2022.101158>

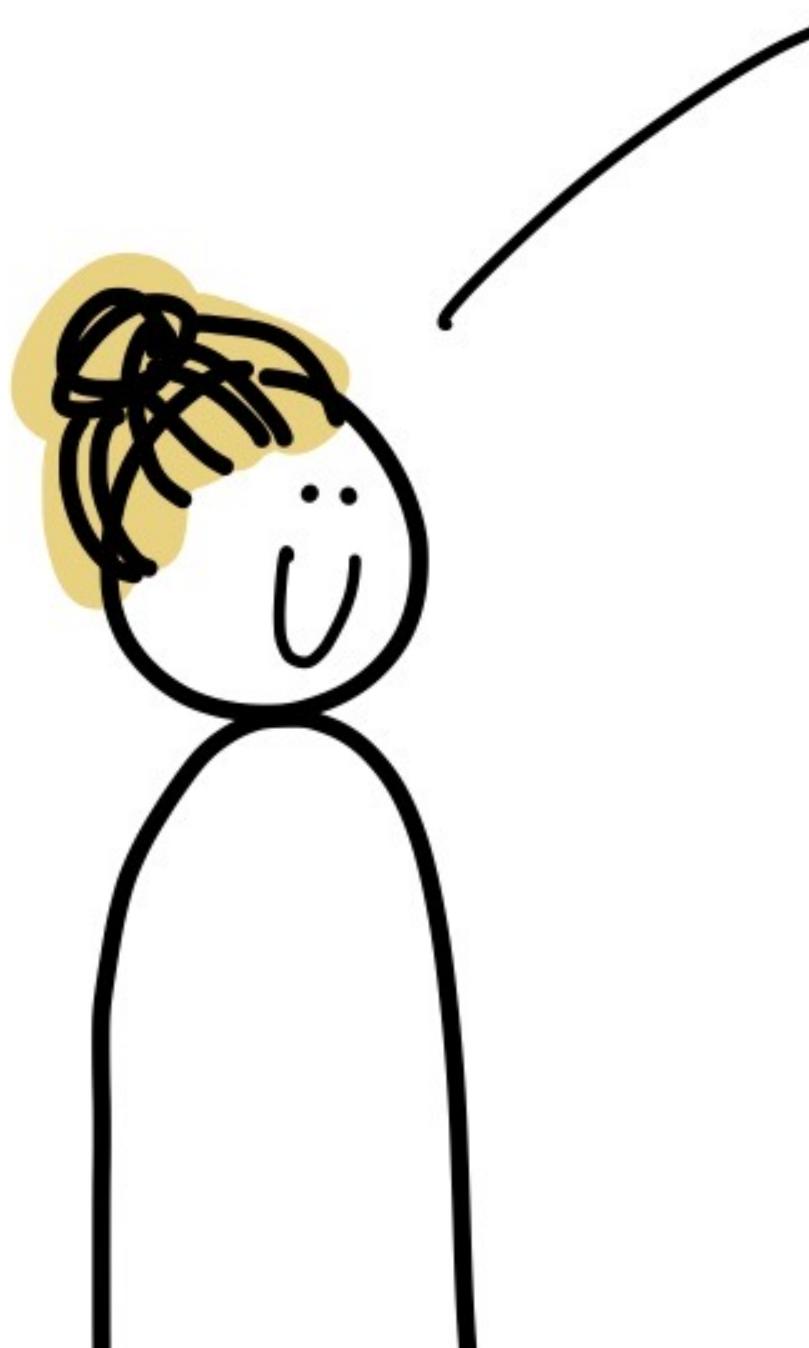
Under a Creative Commons [license](#)

Get rights and content

Open access



Hedy is gradual



Marleen Gilsing ^a, Jes

Show more ▾

+ Add to Mende

<https://doi.org/10.10>

Under a Creative Cor

Journal of Computer Languages

Available online 20 September 2022, 101158

In Press, Journal Pre-proof ⓘ



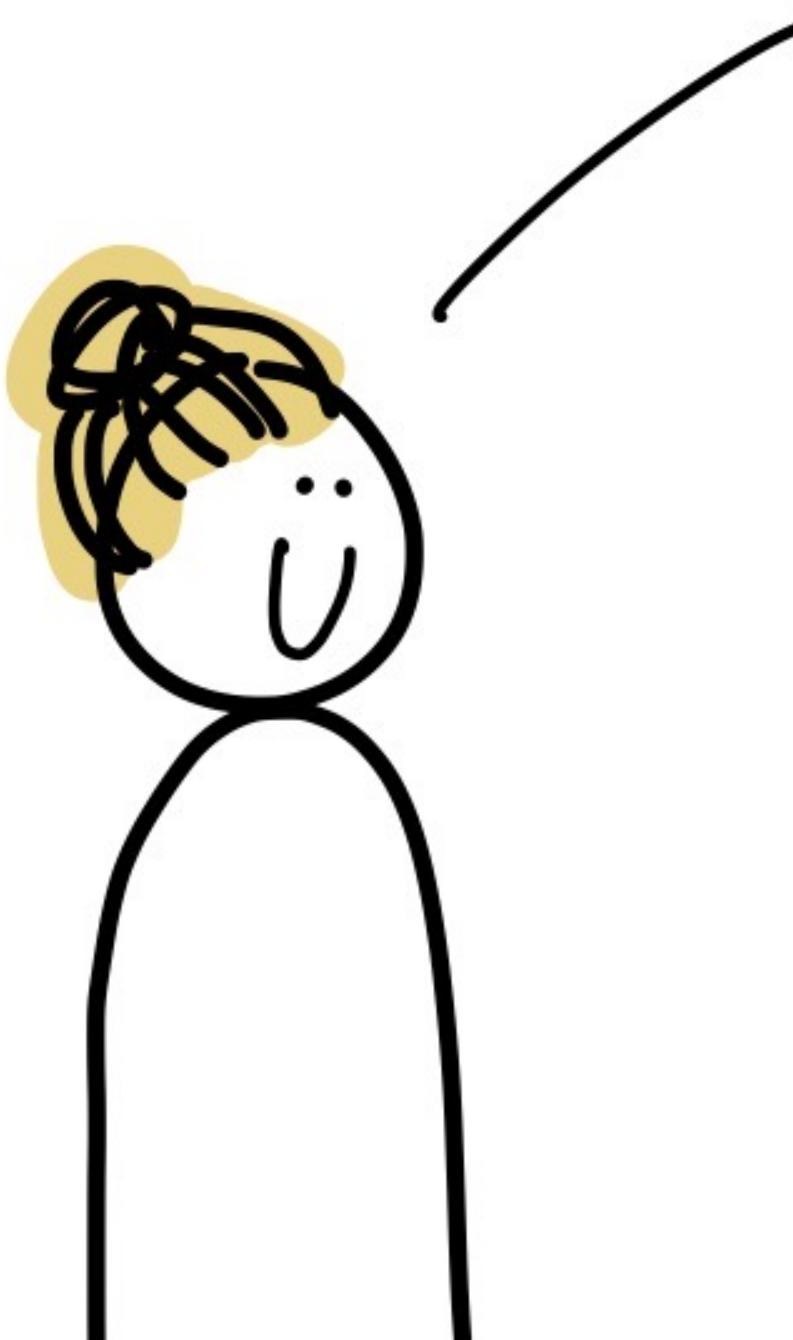
Design, implementation and evaluation of the
Hedy pro

Highlights

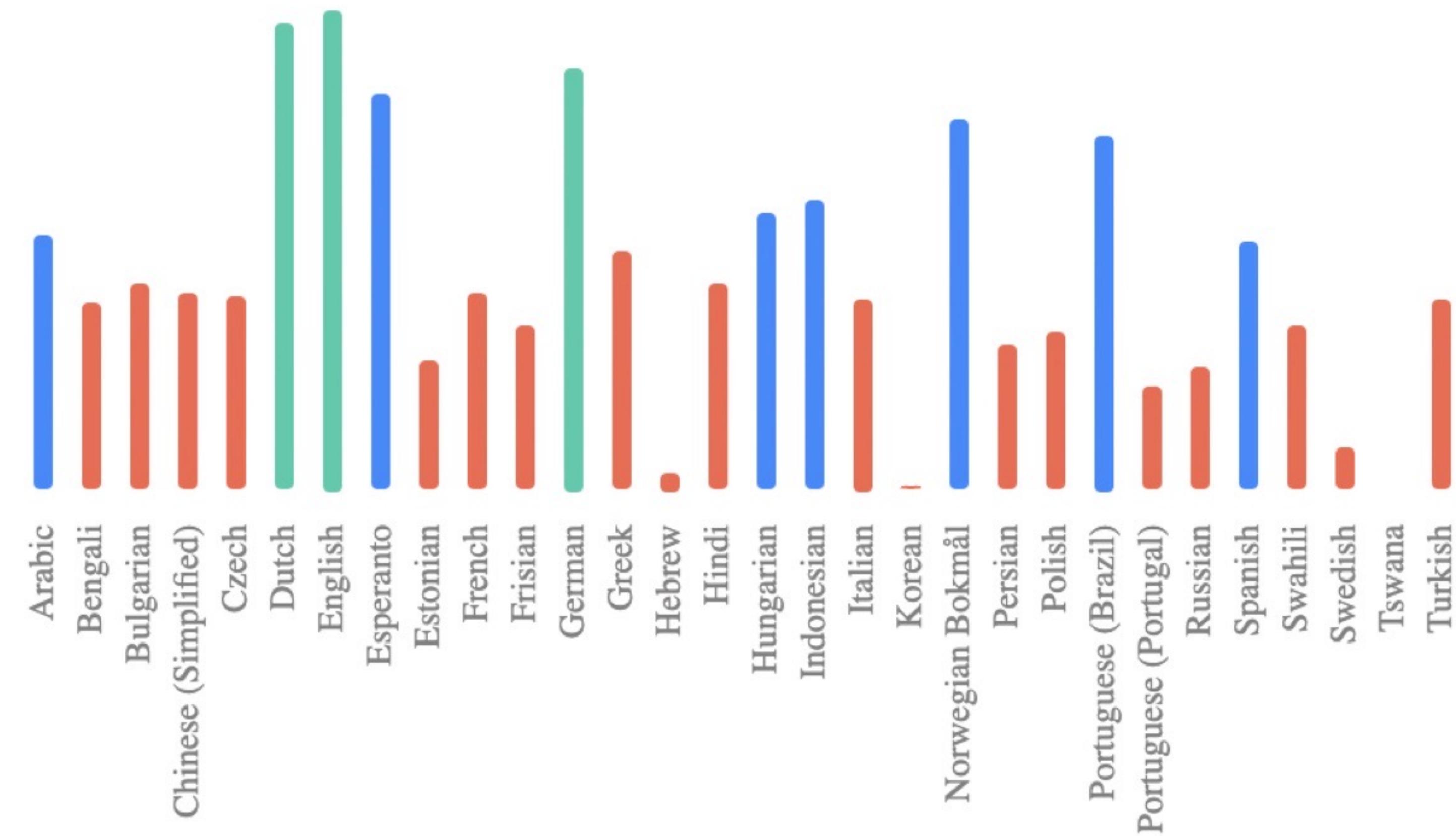
- Details the full implementation of Hedy; the first gradual language for programming education.
- Introduces an EBNF extension used for merging partial grammars to enable gradual language implementation.
- Describes the first user study on Hedy.

Hedy is multilingual

Want to add or
extend translations?



Hedy is multilingual



Hedy is built for teaching

Want to bring Hedy
to teachers?



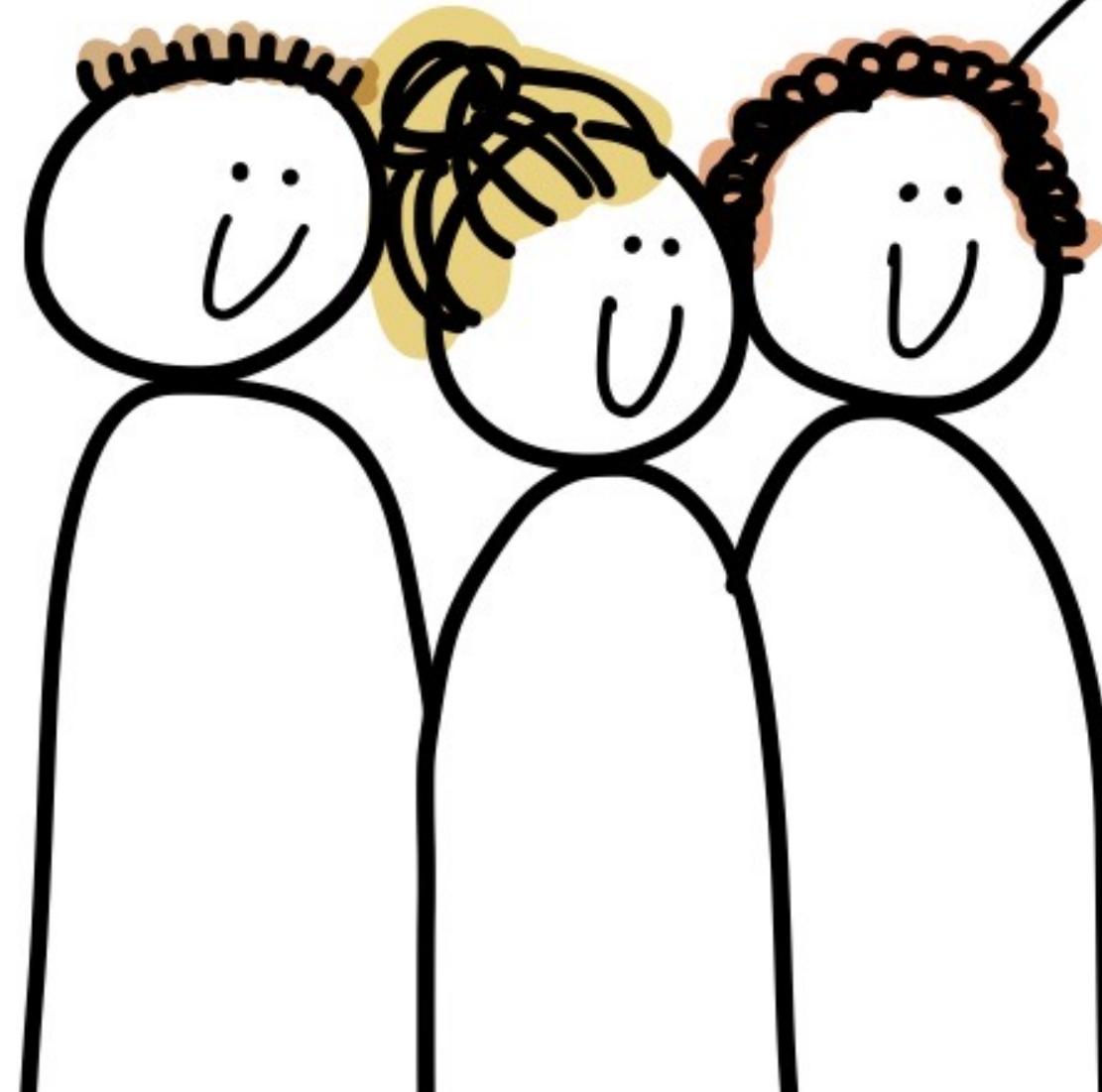
Hedy is built for teaching



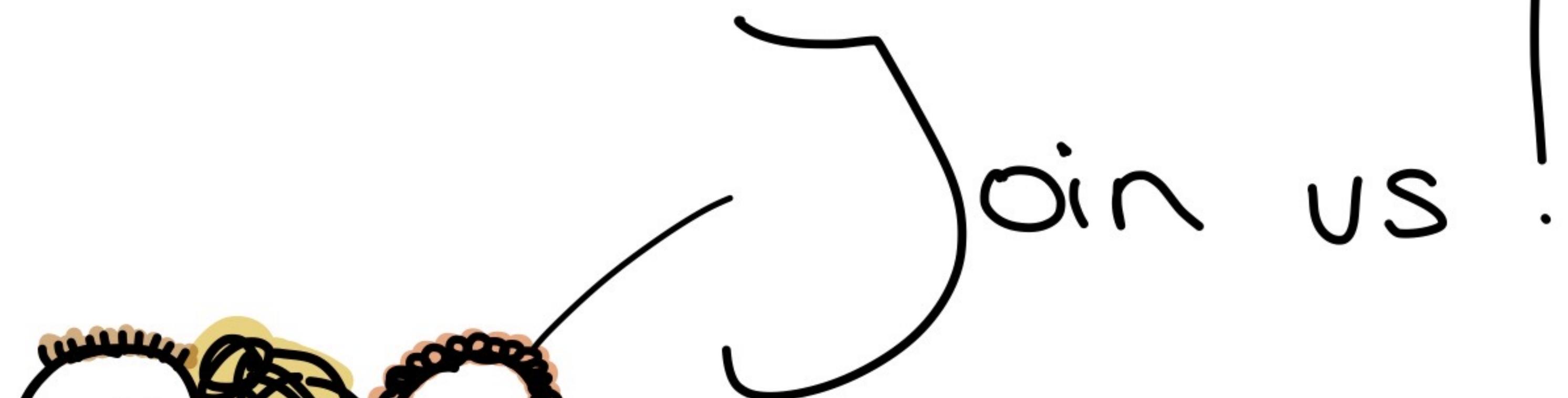
The screenshot shows a YouTube video player displaying a Hedy programming tutorial. The video frame shows a woman speaking. The Hedy interface is overlaid on the video. The interface has a blue header with 'Hedy' and 'Log in'. Below it is a navigation bar with 'Level 1' selected, and tabs for 'Introduction', 'Story', 'Parrot', 'Turtle', 'Rock, paper, scissors', 'Fortune teller', 'Restaurant', 'Haunted House', 'What's next?', 'Puzzle', and 'End quiz'. On the left, there's a text area with instructions about 'print', 'ask', and 'echo' commands. On the right, there are two examples of Hedy code. The first example shows 'print Hello!' and 'print Welcome to Hedy!'. The second example shows 'ask What is your name?' and 'echo hello'. At the bottom, there's a terminal window showing the code being run and the output 'Hello welcome to Hedy!', along with a 'Run code' button and a progress bar indicating the video is at 0:05 / 1:07.

Hedy Community

Join us !



Hedy Community



Join us !
www.hedy.org/join



Guido van Rossum ✅

@gvanrossum

...

Just discovered Hedy, a gradual programming language. It's a new idea on how to teach programming to beginners. Very cool! hedycode.com @hedycode

1:53 AM · Mar 30, 2022 · Twitter Web App

139 Retweets 14 Quote Tweets 691 Likes

