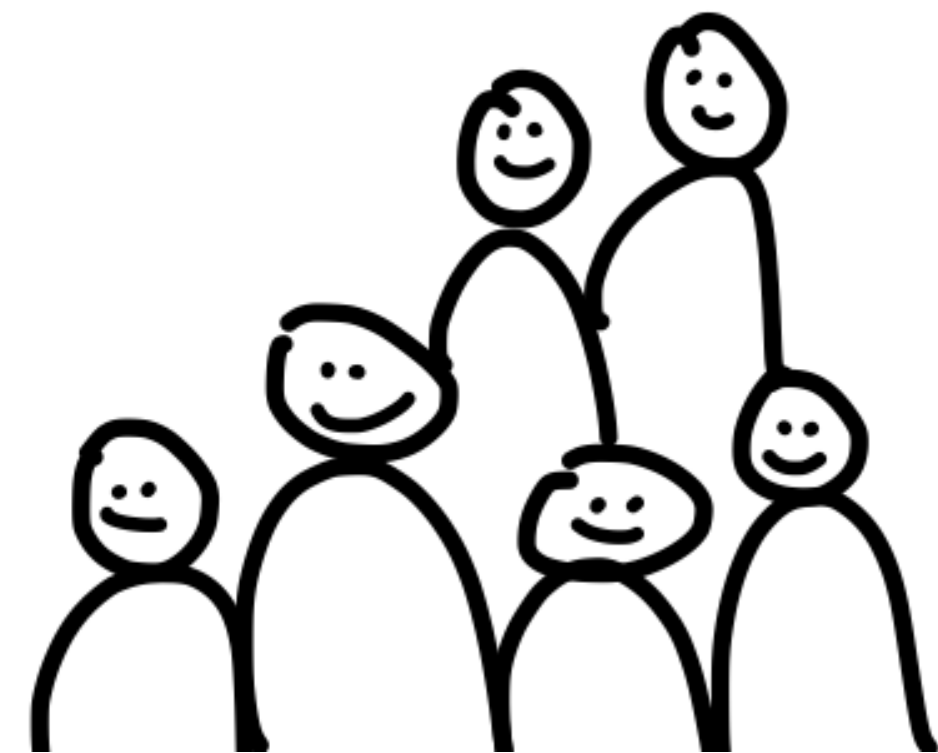


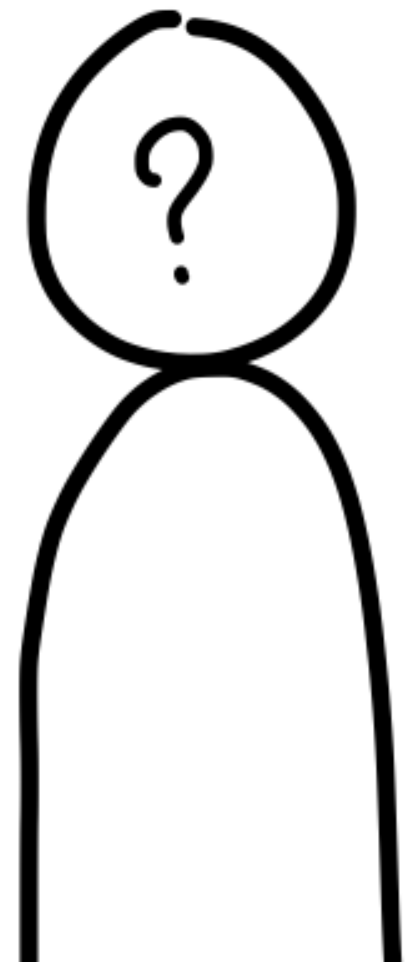
How to teach programming (and other things)



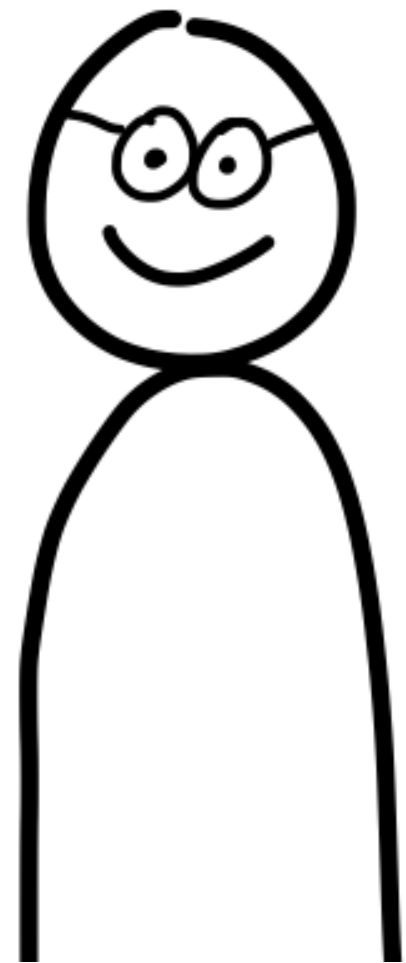
@Feliienne



"Everyone should
learn programming"



"Everyone should
learn programming"



- Every programmer ever

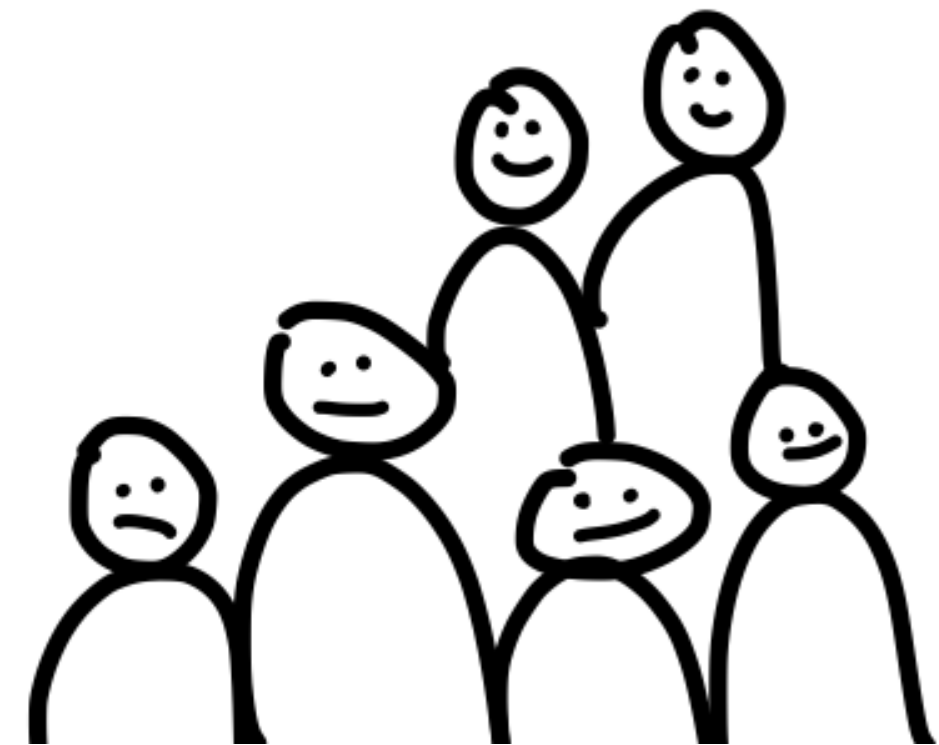


2013

We'd love to
learn programming



2013

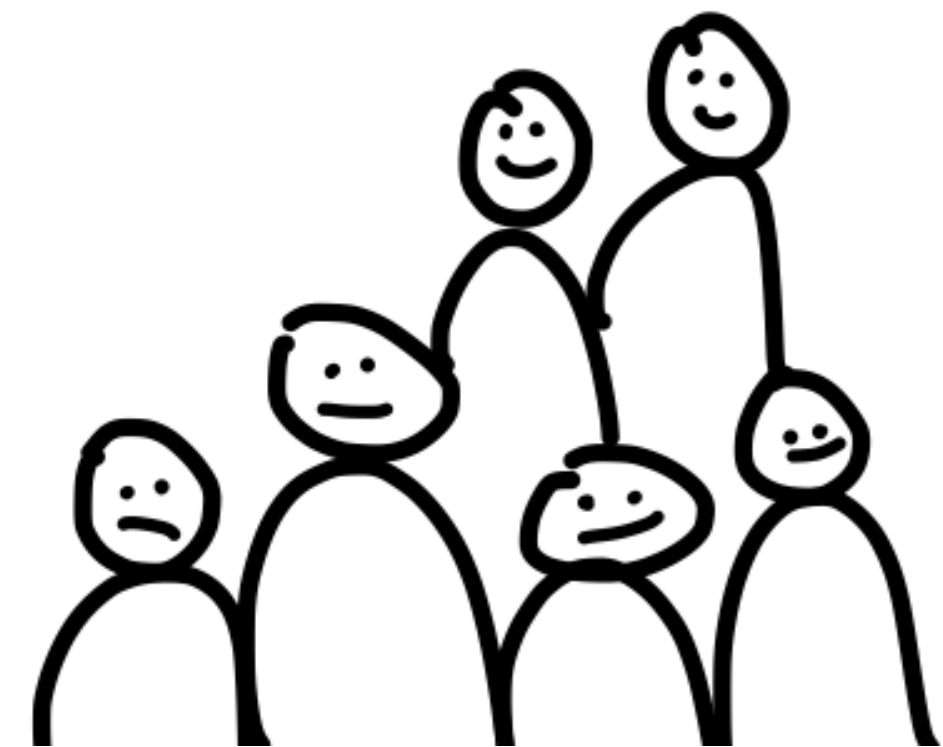


Yeah!

We'd love to
learn programming



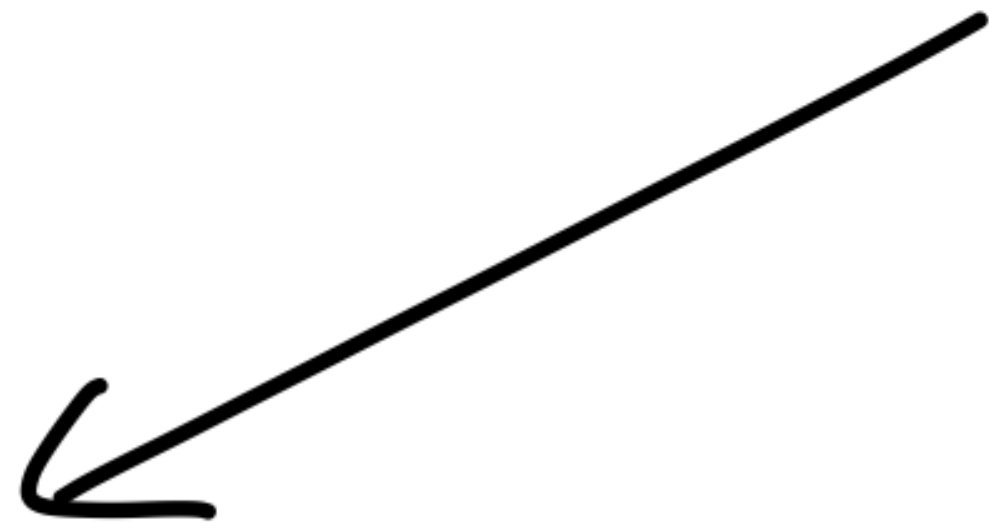
2013



Yeah!

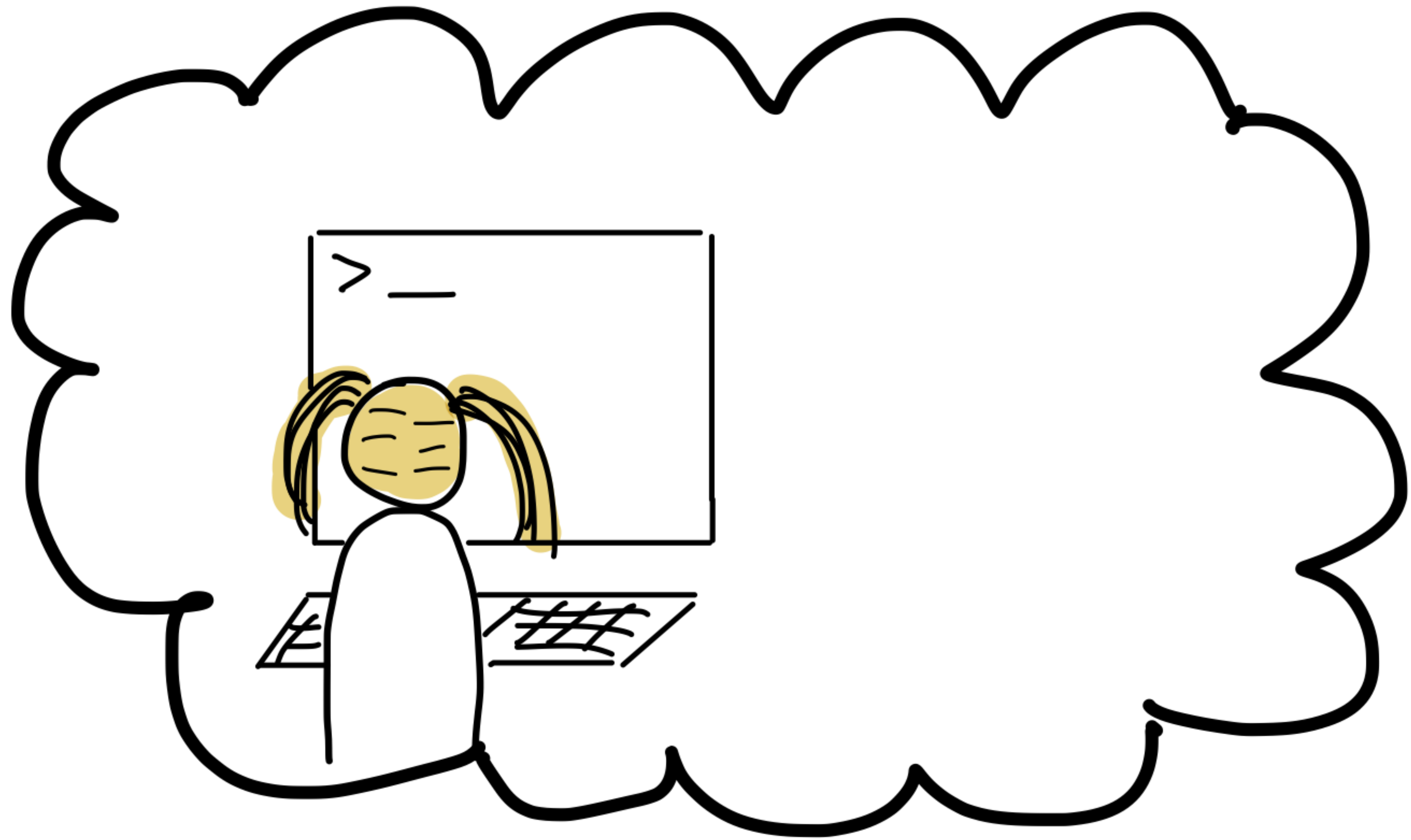


Doesn't know anything
about teaching





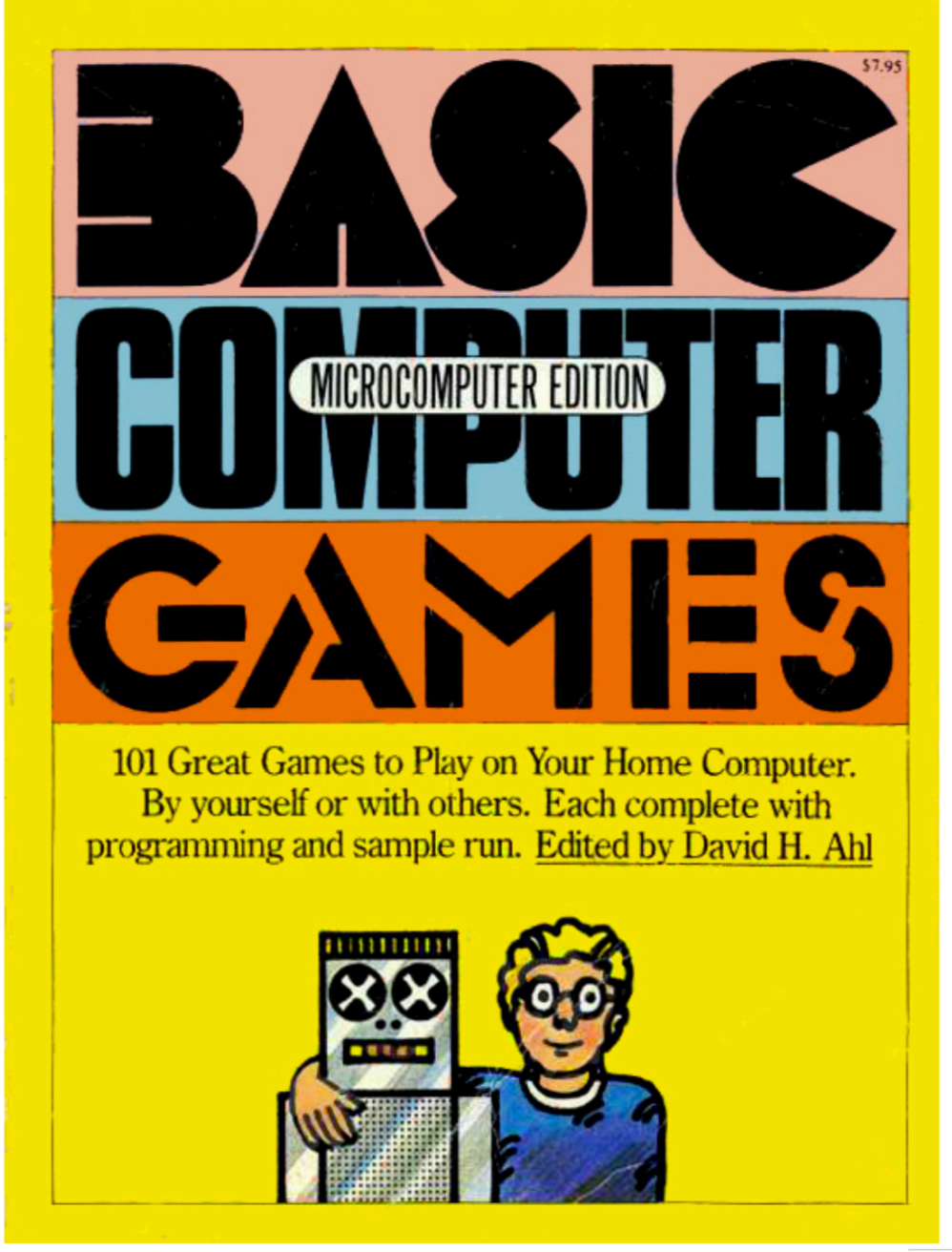
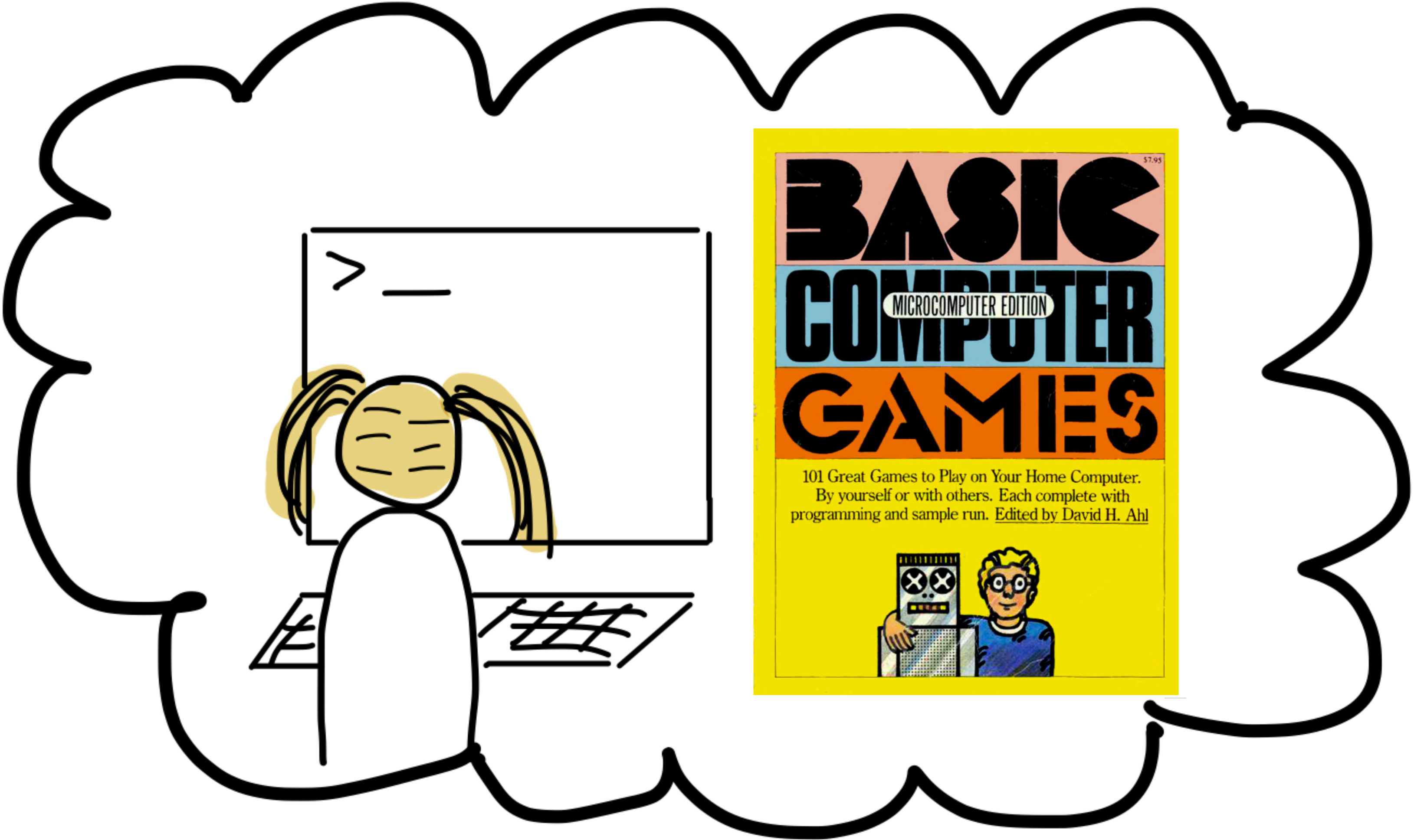
o
o
o





o

o

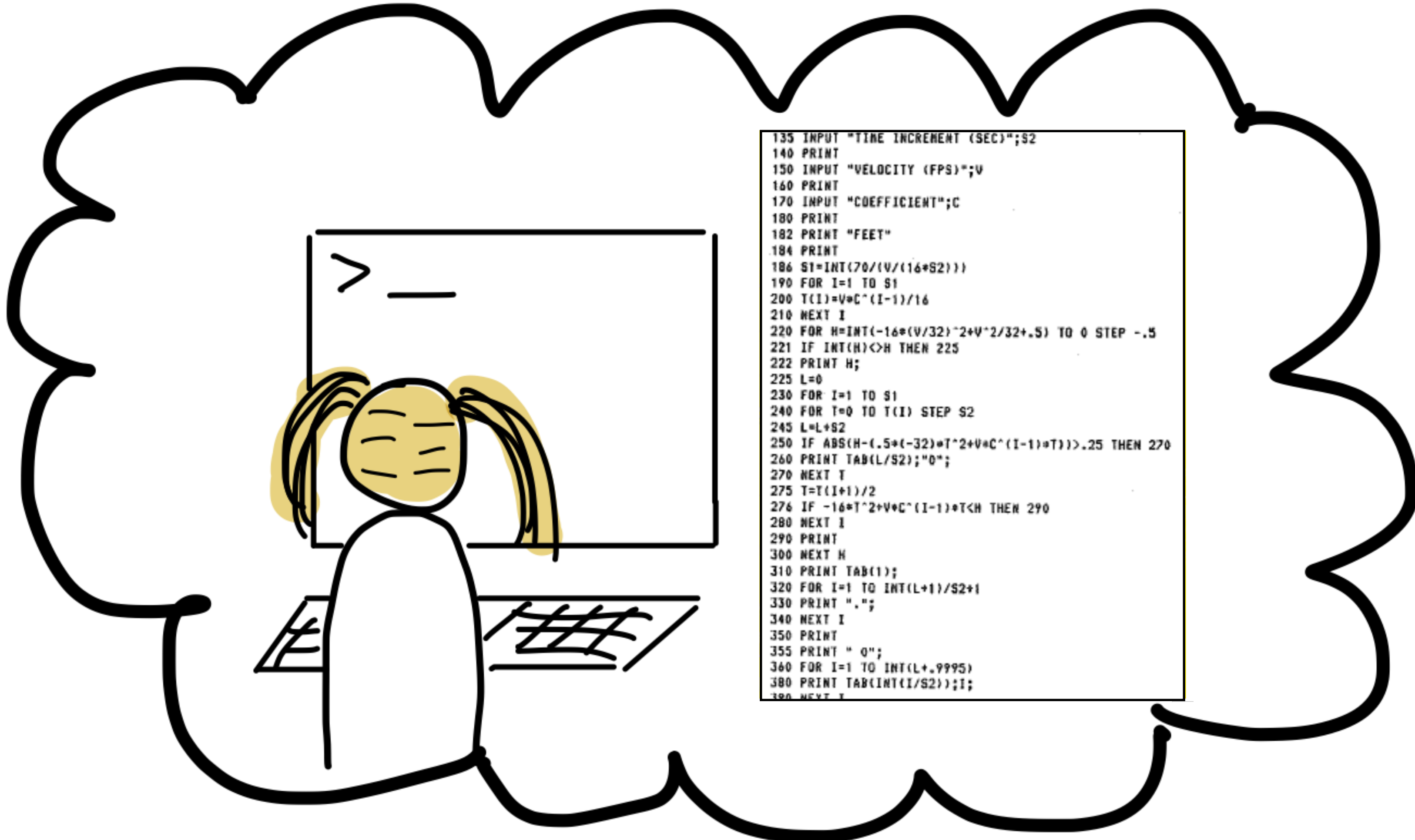




o

o

o



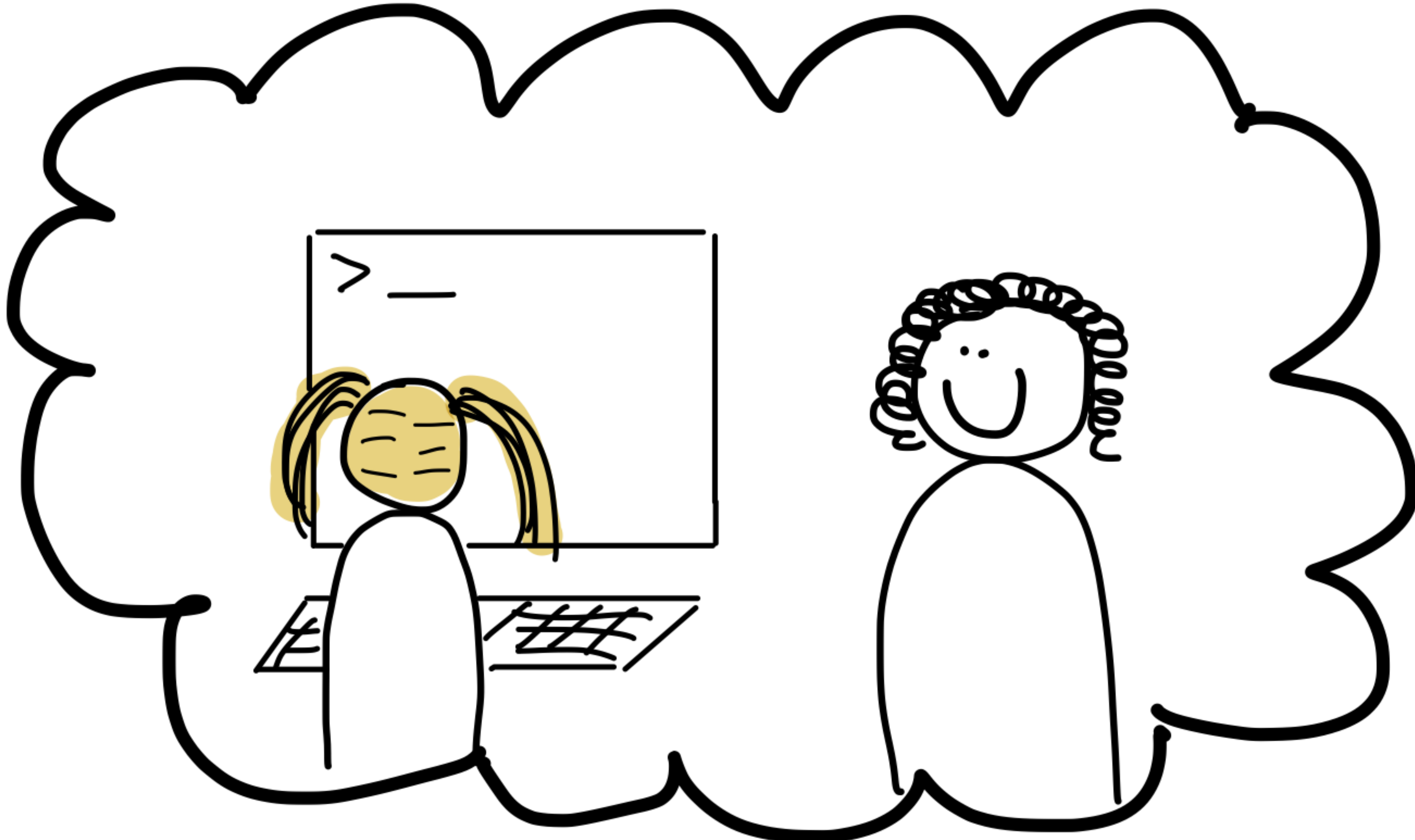
```
135 INPUT "TIME INCREMENT (SEC)";S2
140 PRINT
150 INPUT "VELOCITY (FPS)";V
160 PRINT
170 INPUT "COEFFICIENT";C
180 PRINT
182 PRINT "FEET"
184 PRINT
186 S1=INT(70/(V/(16*S2)))
190 FOR I=1 TO S1
200 T(I)=V*C^(I-1)/16
210 NEXT I
220 FOR H=INT(-16*(V/32)^2+V^2/32+.5) TO 0 STEP -.5
221 IF INT(H)<>H THEN 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
```



o

o

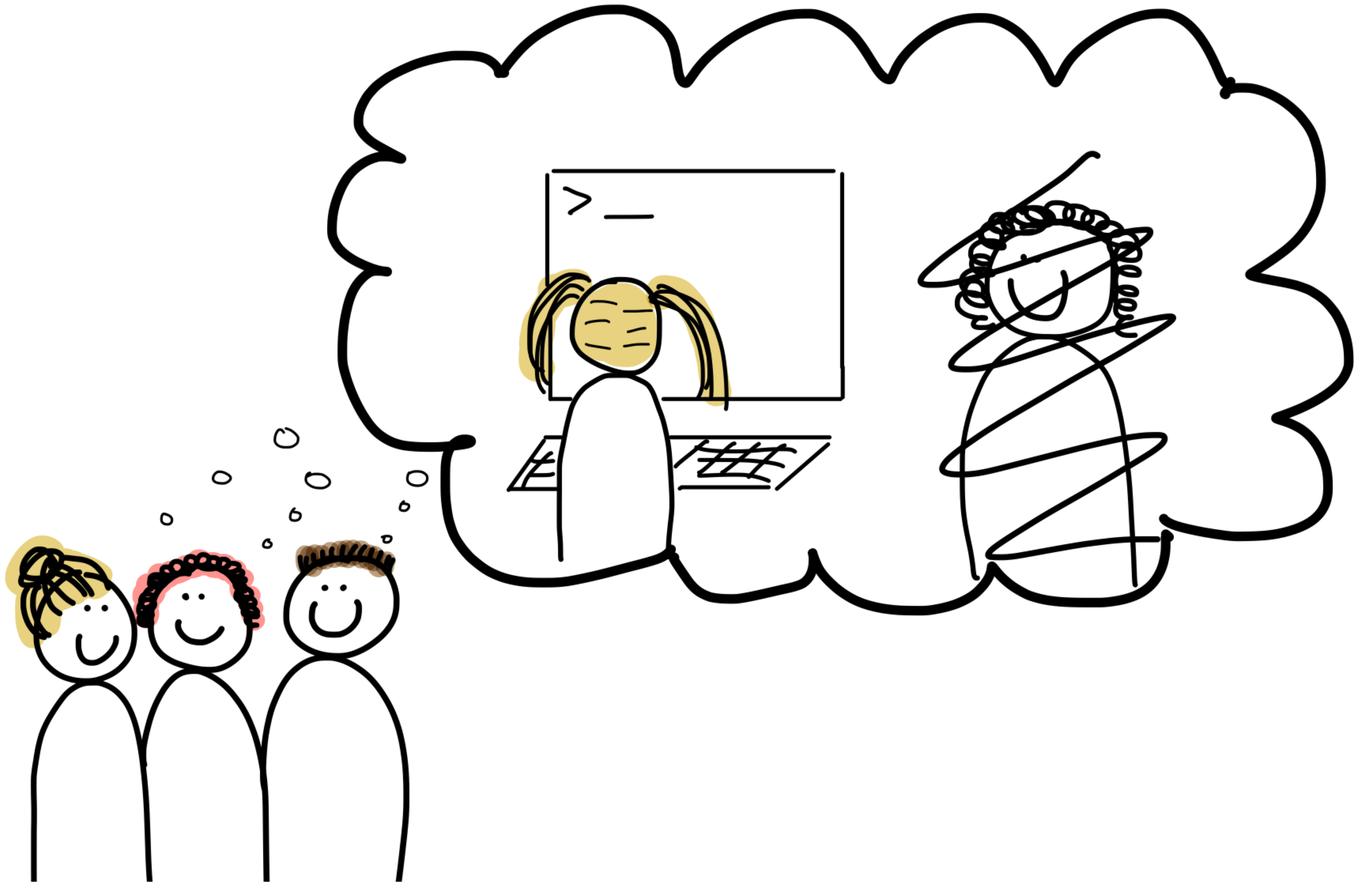
o





o
o
o



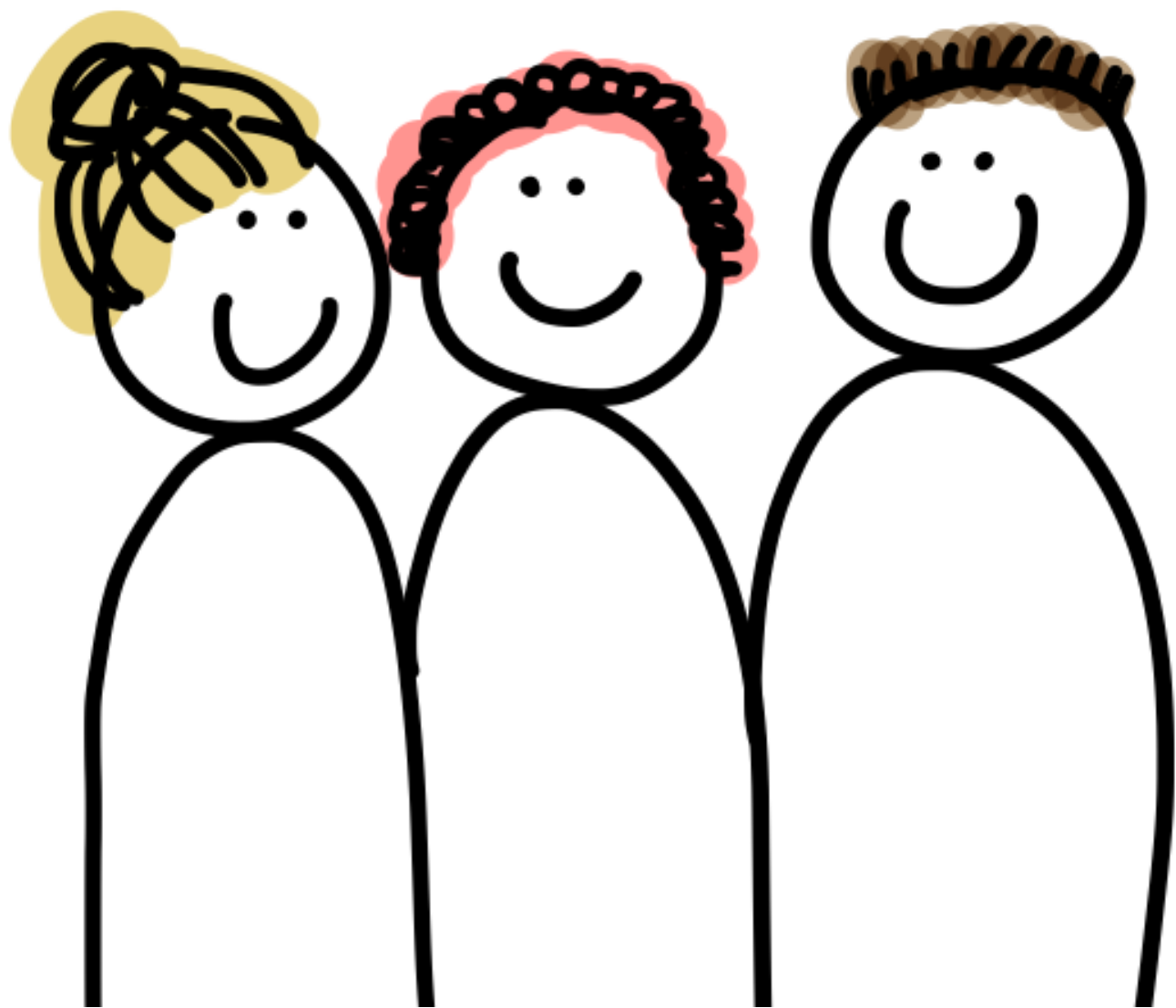
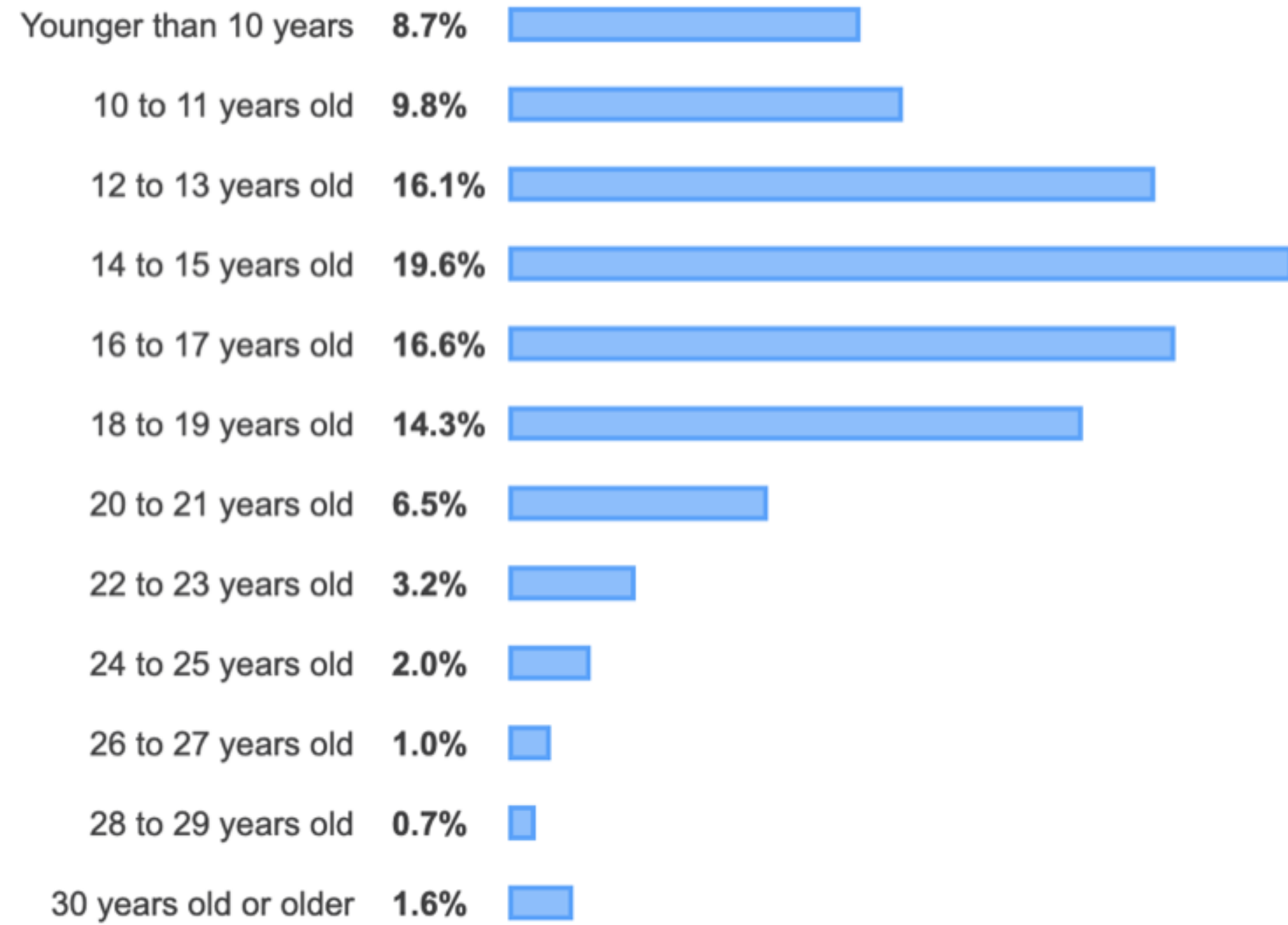


Writing That First Line of Code

Overall

★ By Country

By Gender



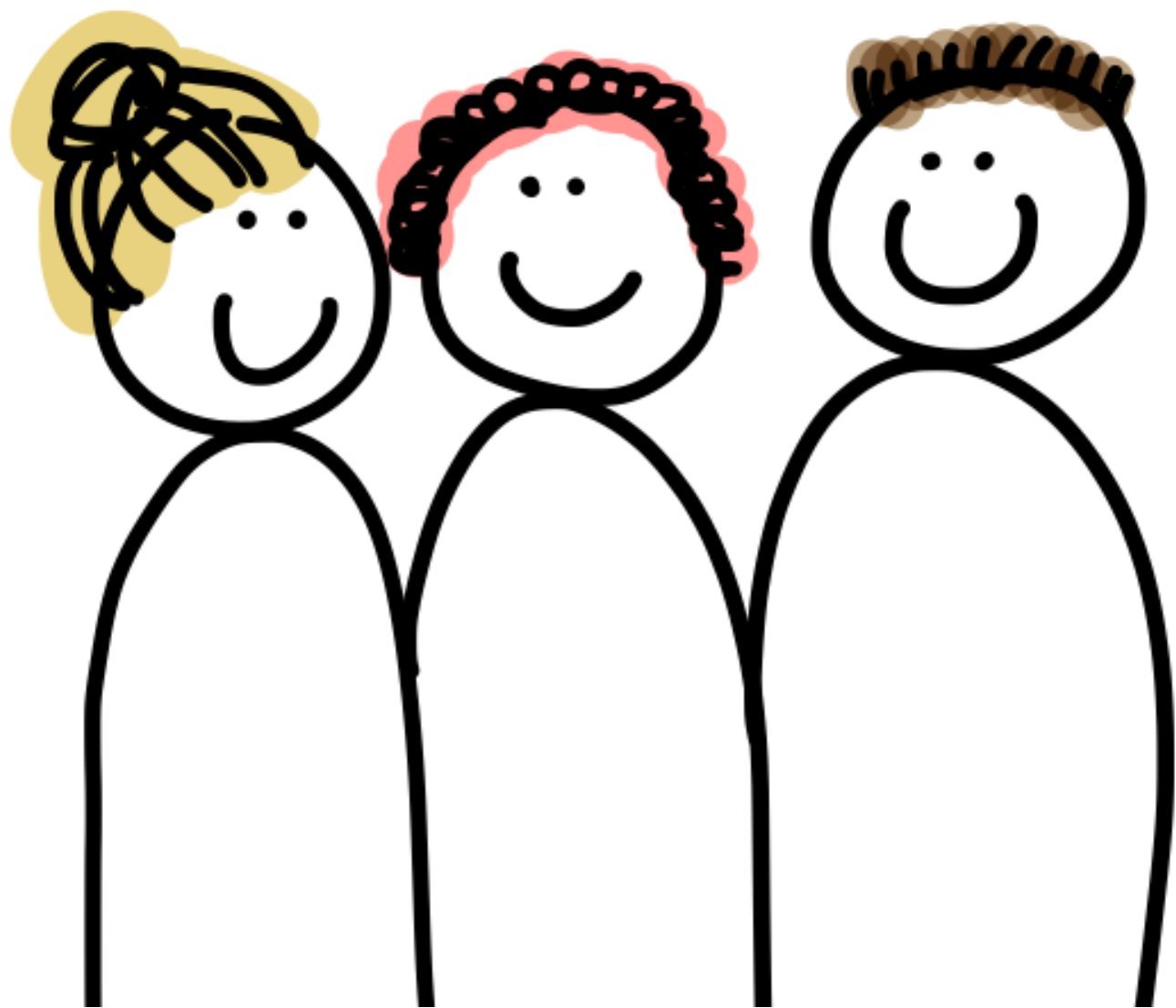
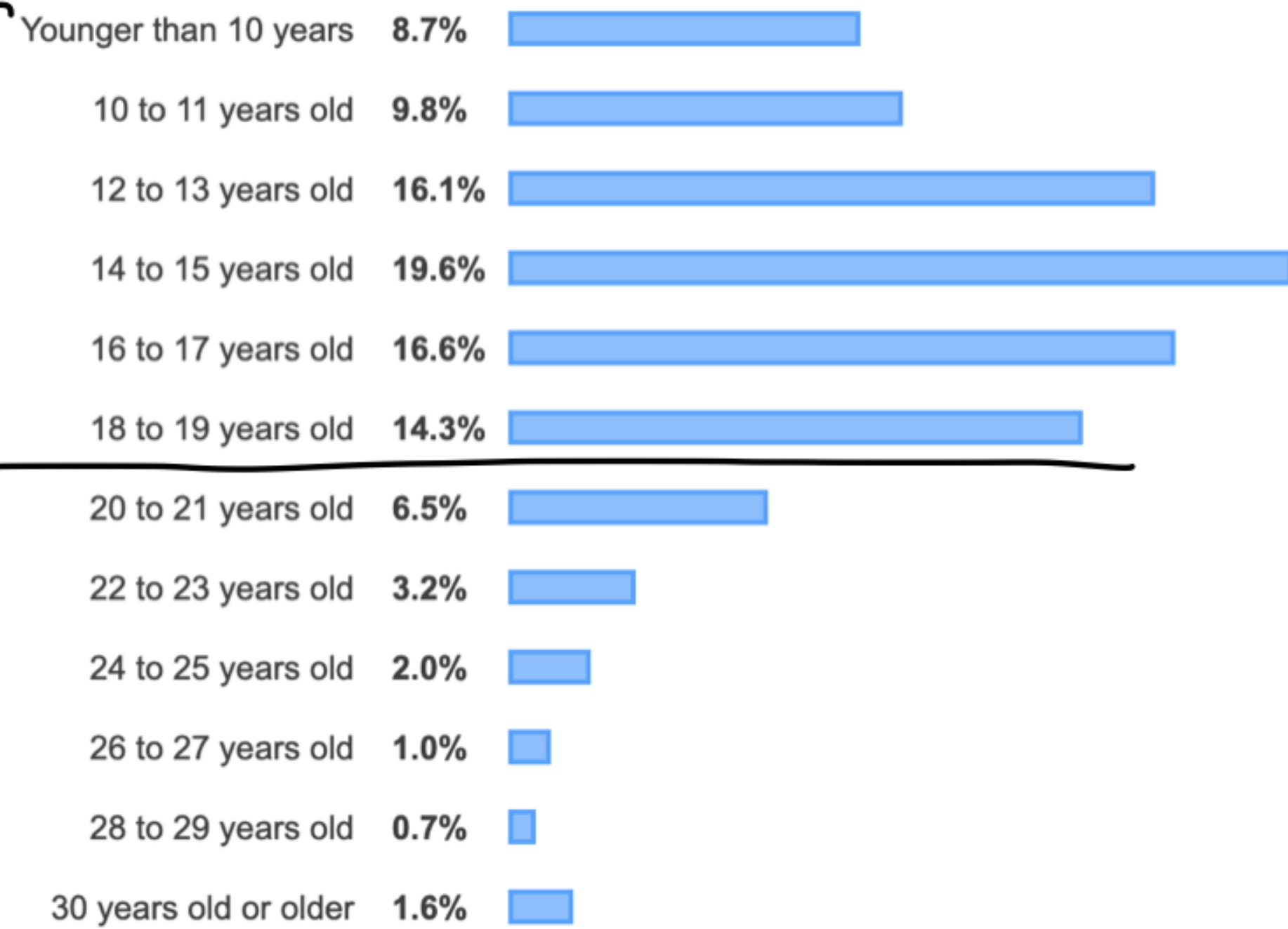
85%

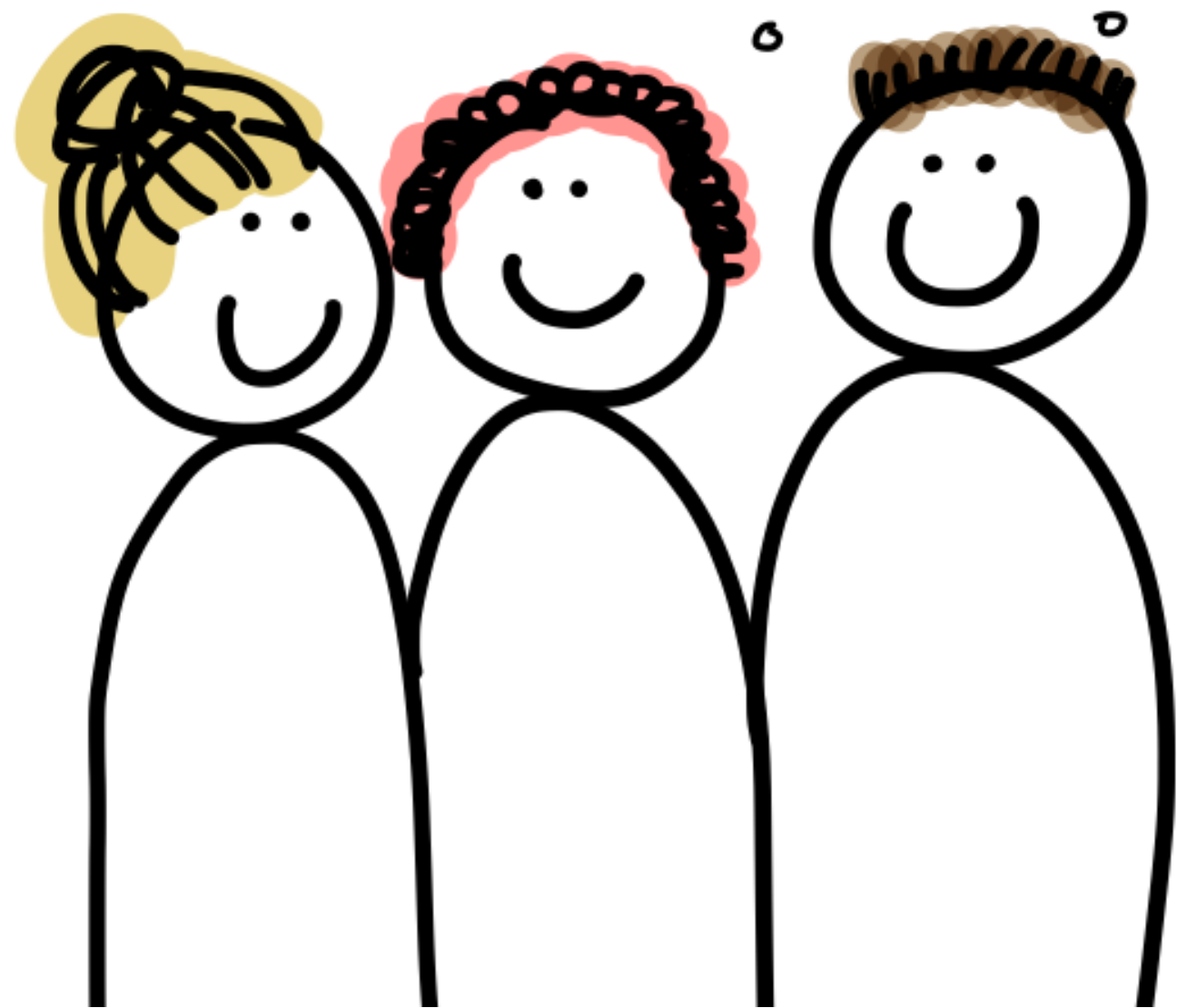
Writing That First Line of Code

Overall

★ By Country

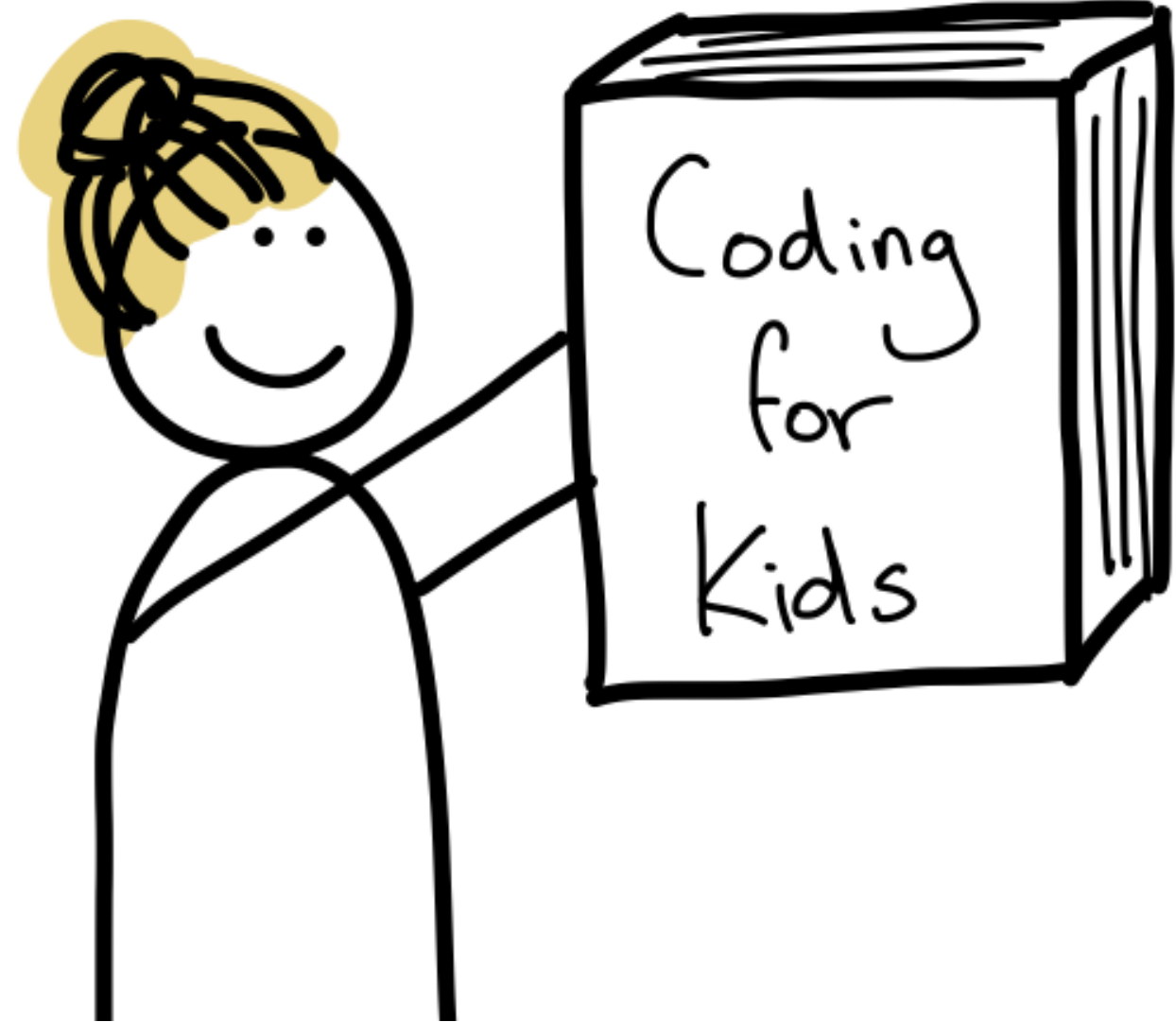
By Gender





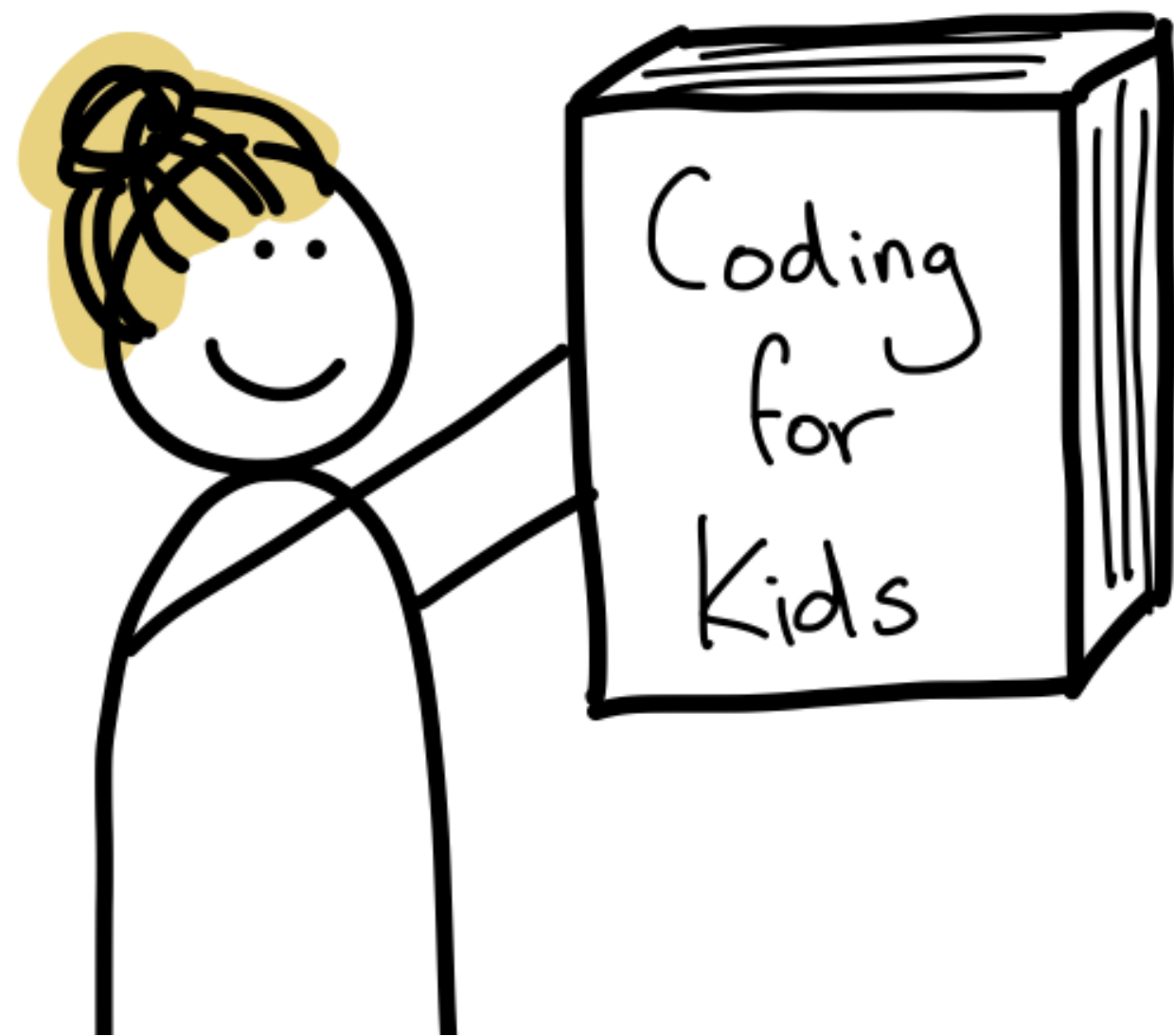
Just let
them explore!



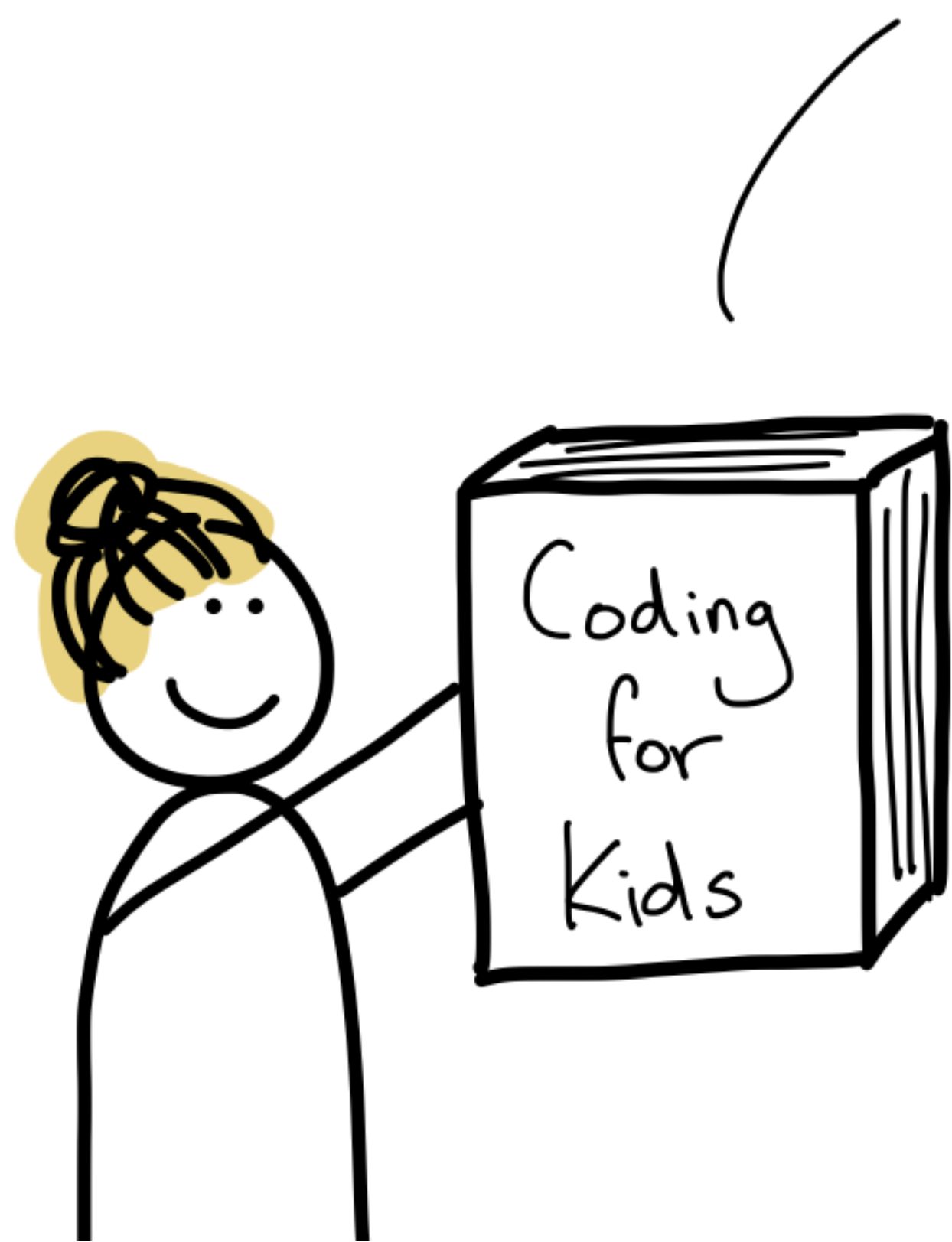


OH!
ER ZIT
EN BUG!

Af en toe kom je een probleem tegen als je je project start. Dit heet een *bug*. Dit is het leukste van programmeren: een fout ontdekken in iets dat je gemaakt hebt, en die dan oplossen.



"The best part of programming is finding mistakes"

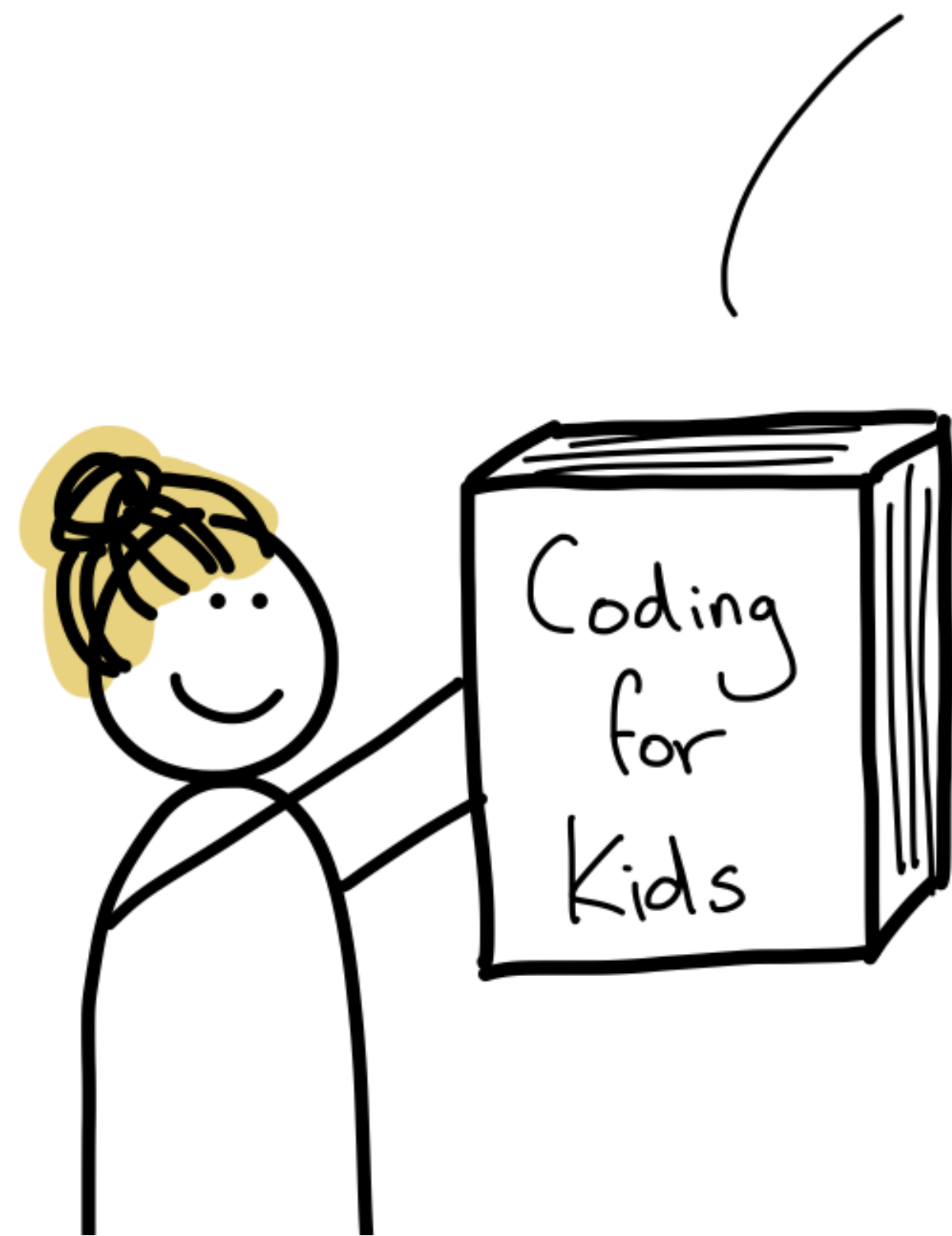


Reken erop dat je veel fouten zult maken en veel van dit soort foutmeldingen te zien krijgt. Denk erom, programmeurs leren alleen maar meer door het maken van fouten en door uit te zoeken waar de fout zit. Zelfs de beste programmeurs ter wereld maken fouten, zoals het vergeten van een " in hun code.

"Programmers only learn from making mistakes"

Heb plezier!

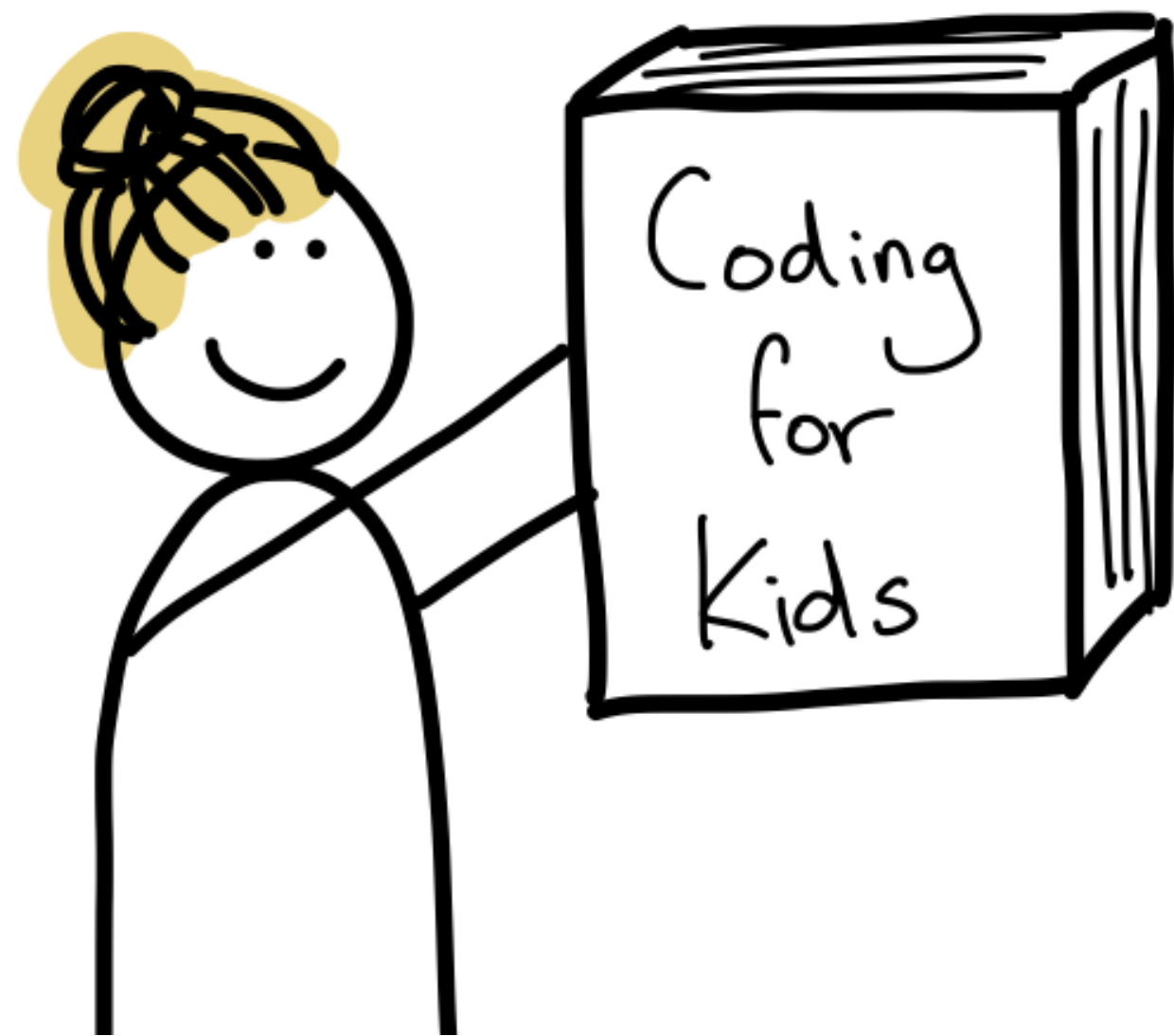
Coderen lijkt veel op puzzels oplossen. Het is een uitdaging, vaak lukt het niet, en soms is het frustrerend. Maar je beleeft ook een doorbraak als je een probleem weet op te lossen en het geeft een kick als je codering werkt. Werk op je eigen niveau: als een



"You will fail often,
and it will be frustrating"

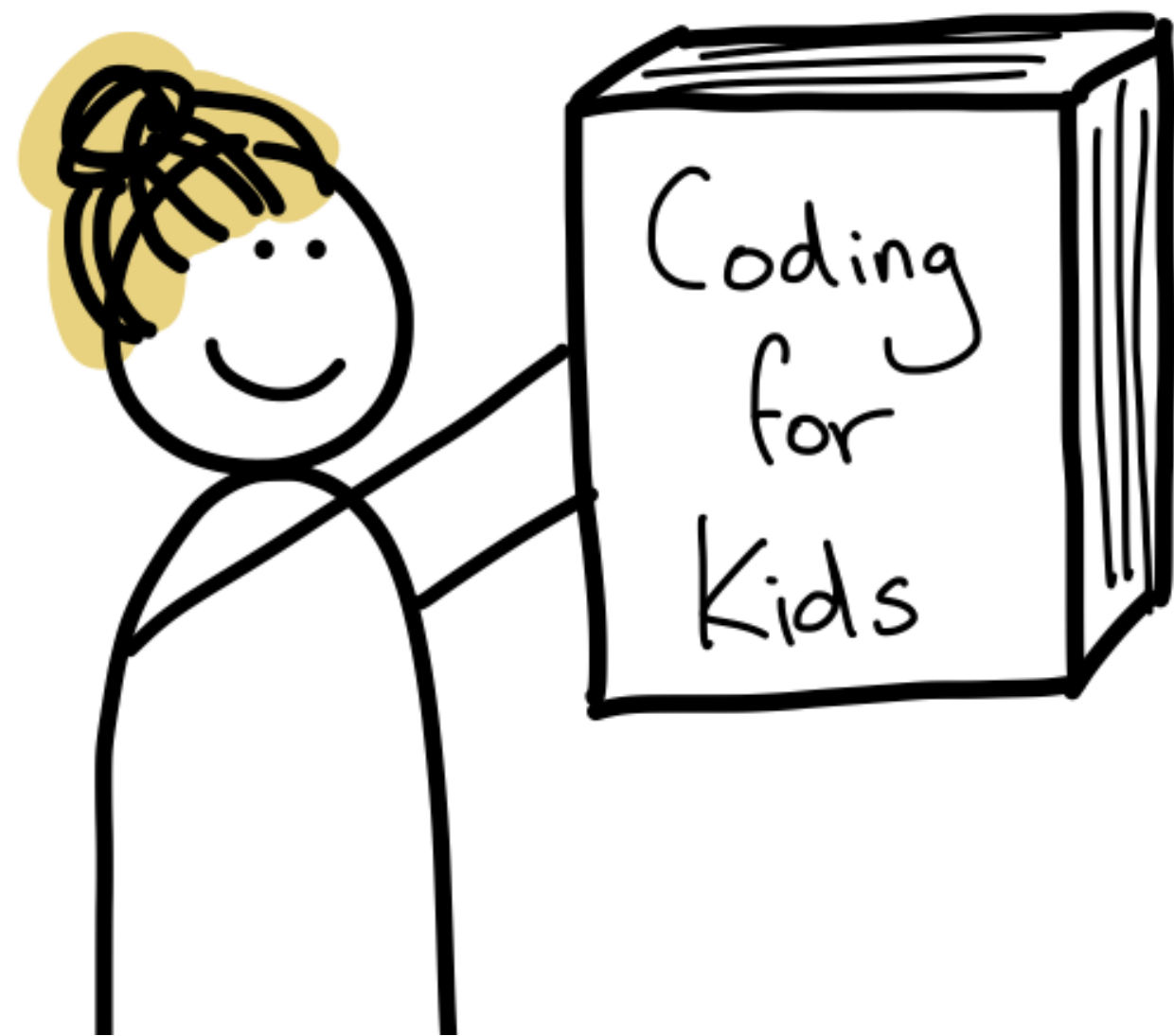
belangstelling. Wees nooit bang om te rotzooien, prutsen en experimenteren en houd je niet aan de regels, maar blijf nieuwsgierig. En het belangrijkste: het is leuk!

"Never be afraid to mess around and experiment"



belangstelling. Wees nooit bang om te rotzooien, prutsen en experimenteren en **houd je niet aan de regels**, maar blijf nieuwsgierig. En het belangrijkste: het is leuk!

"Break the rules"



The first character, "F", is in position 0

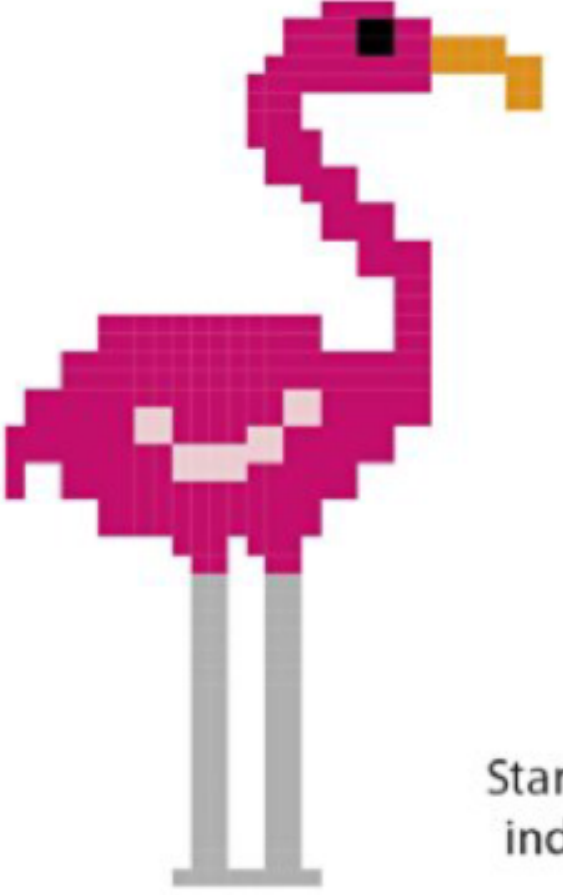
3 "Slicing" Two indexes can be used to pull out a part of the string or "slice" it. The letter in the last position isn't included.

```
>>> a[1:7]
```

Colon defines the range of characters

'LAMING'

A slice from index 1 to index 6 of variable "a"



4 From the start or the end If you leave off the start or end index, Python will automatically use the first or the last character of the string.

```
>>> a[:3]
```

'FLA'

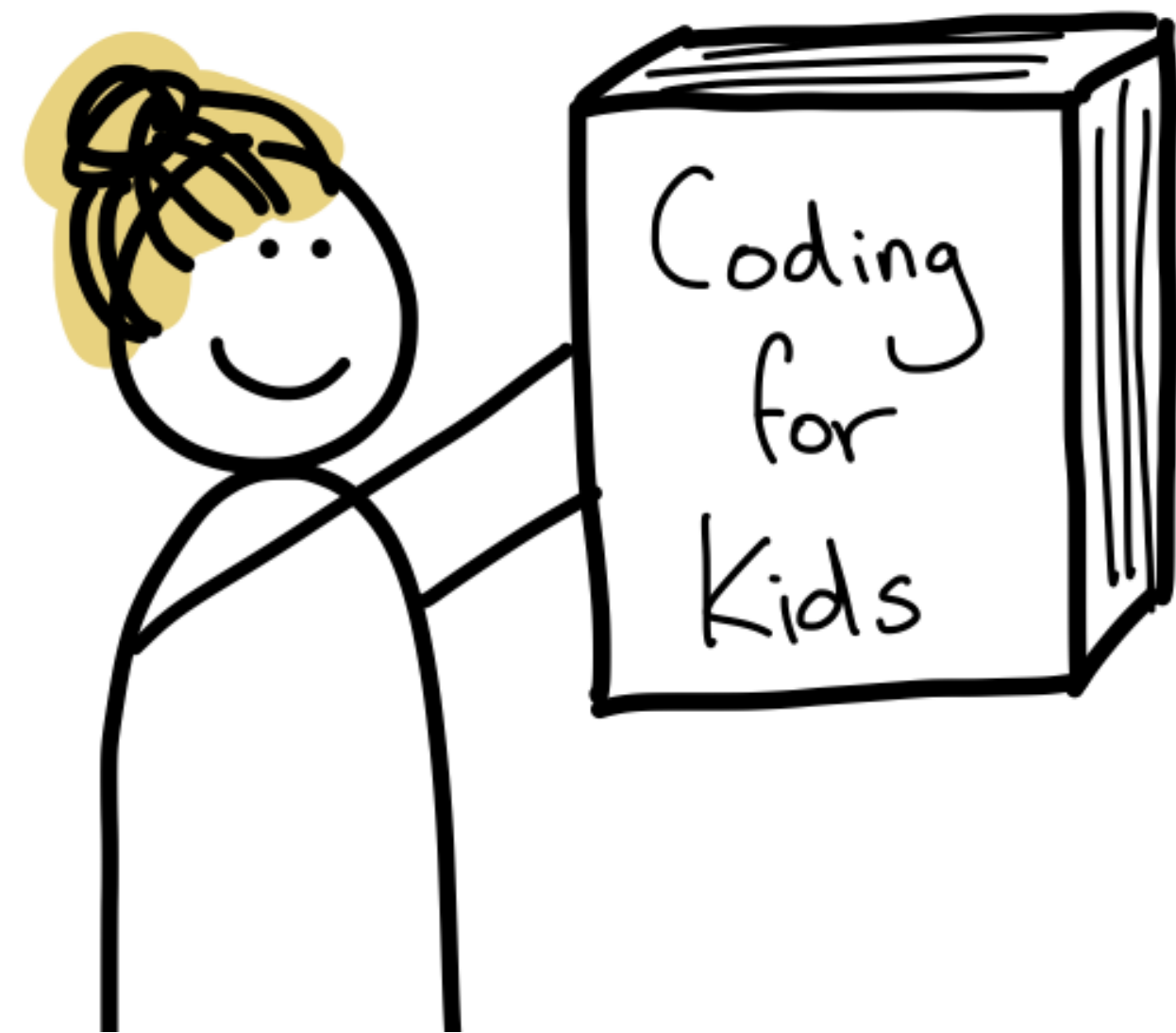
```
>>> a[3:]
```

'MINGO'

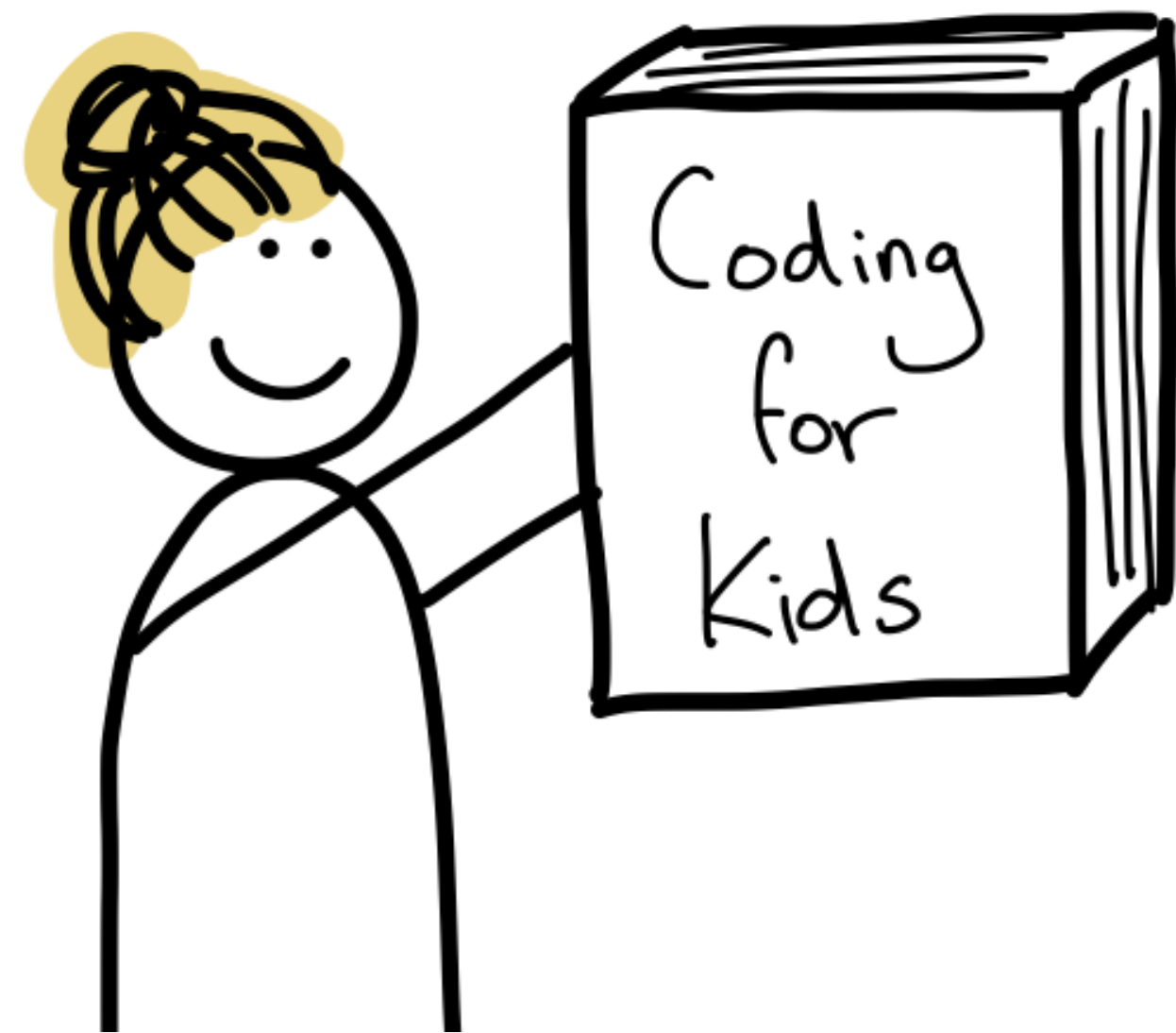
Starts at index 0

Ends at index 7

"Break the rules"



belangstelling. Wees nooit bang om te rotzooien, prutsen en experimenteren en houd je niet aan de regels, maar blijf nieuwsgierig. En het belangrijkste: het is leuk!



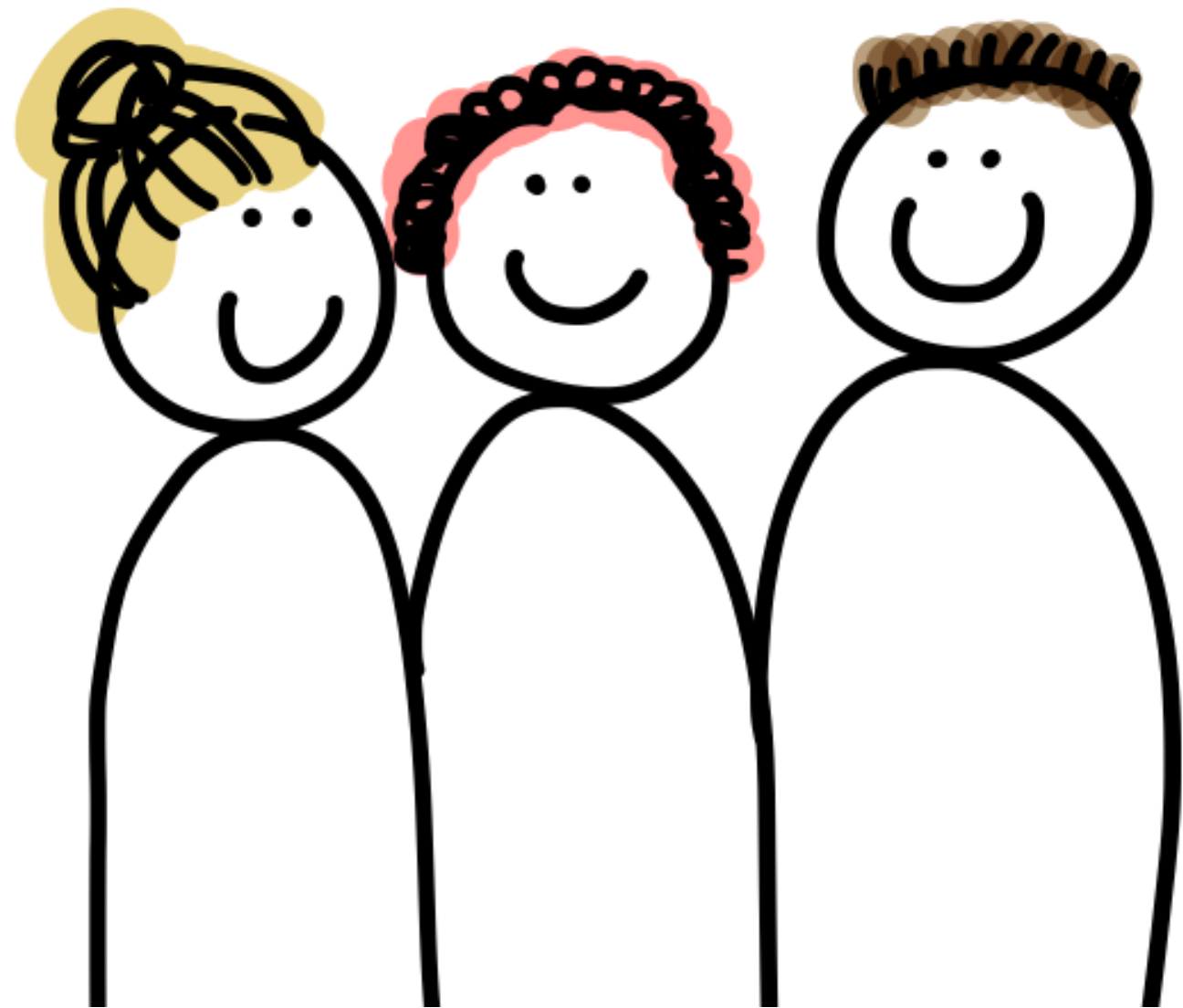
"And most importantly:
it's fun"

Just let
them explore!



How is programming
taught at code clubs?
Aivaloglou & Hermans '19

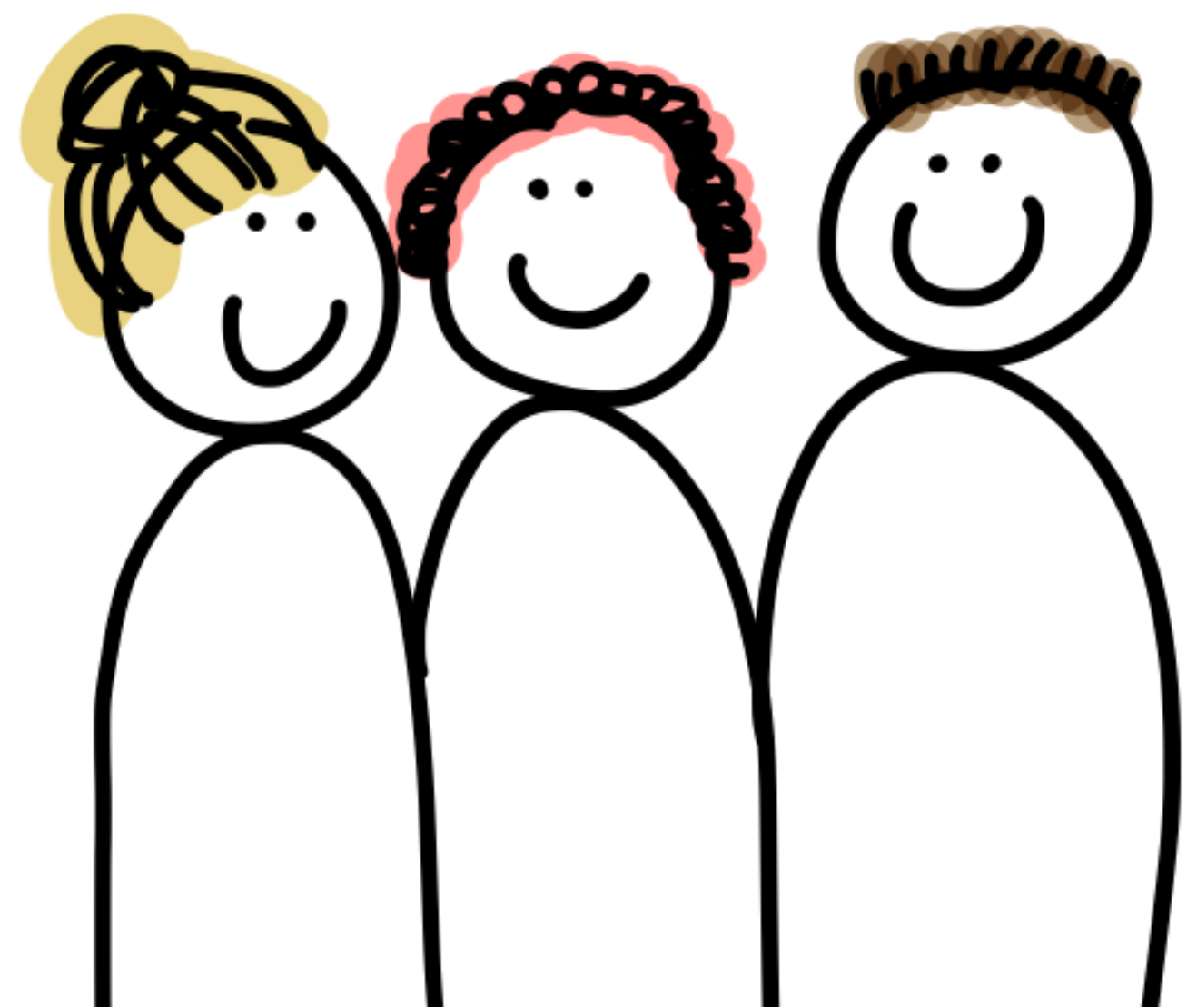
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

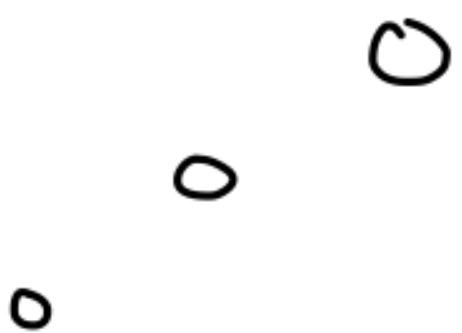


In 71% of clubs
students work
independently

How is programming
taught at code clubs?
Aivaloglou & Hermans '19

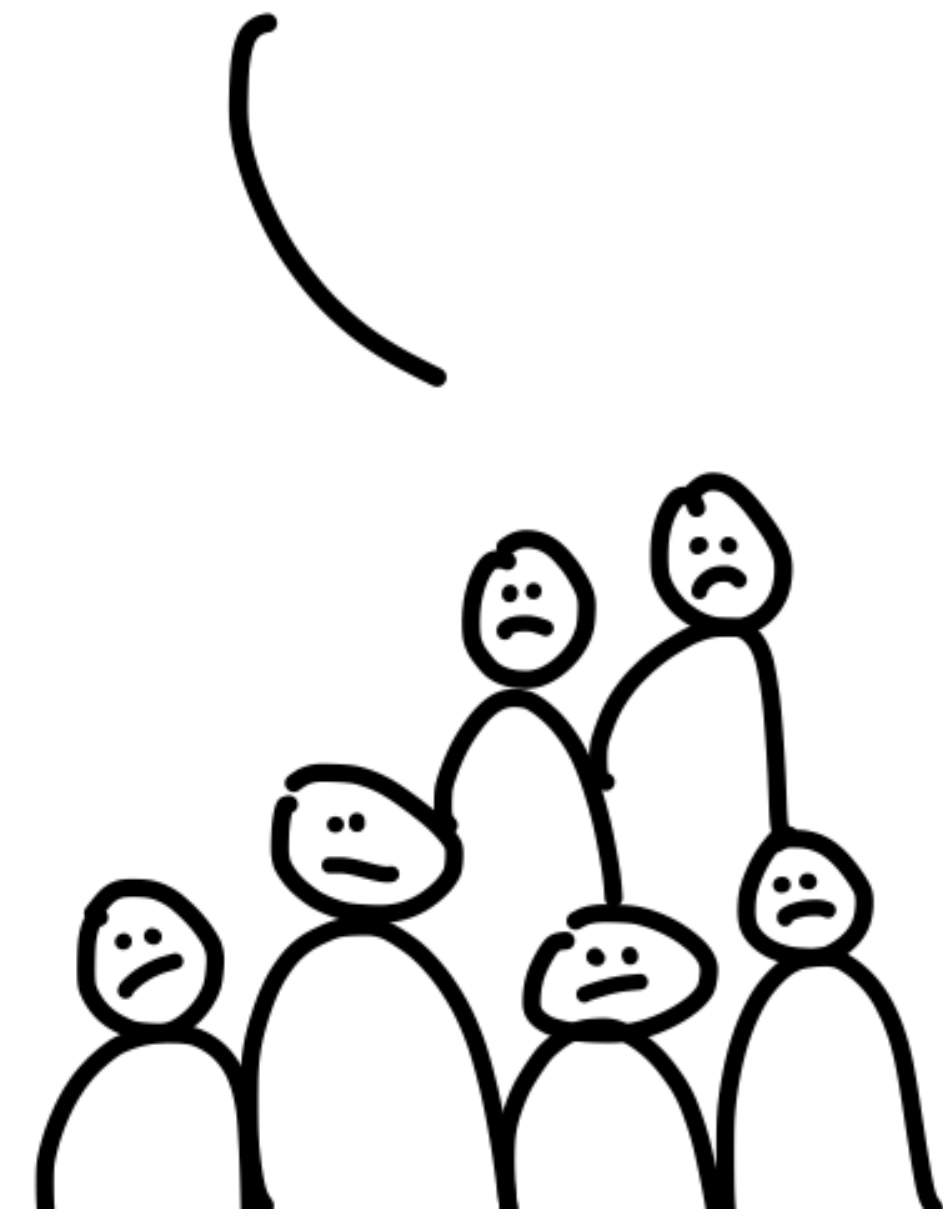
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____



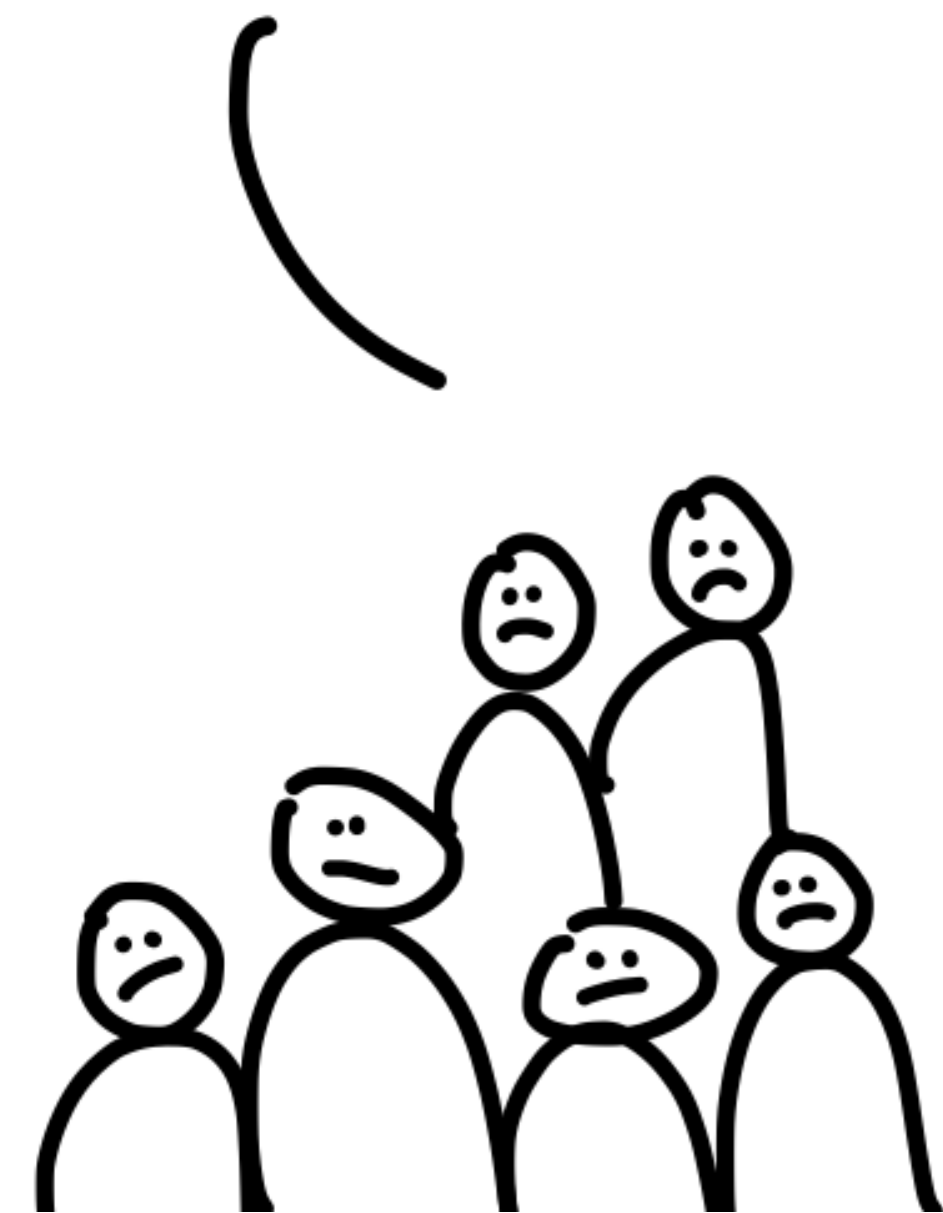


Just let
them explore!

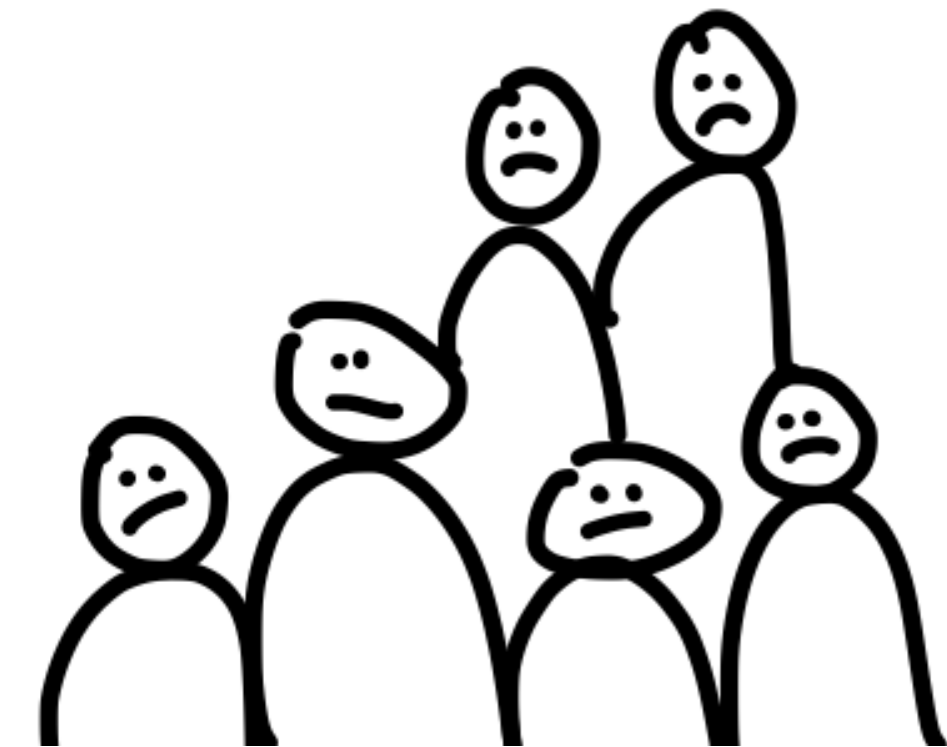
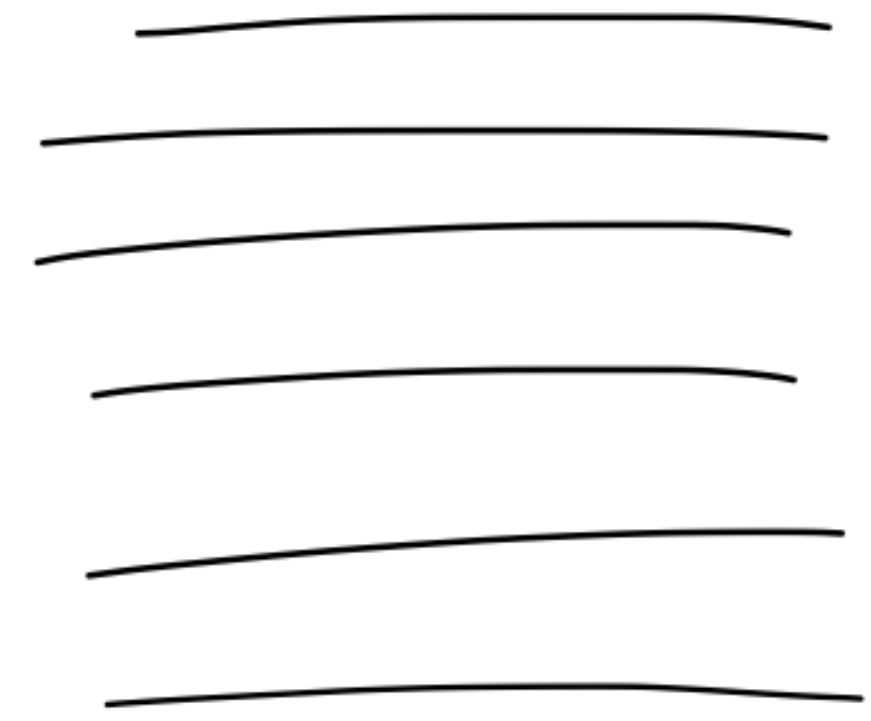
This is hard!



This is hard!

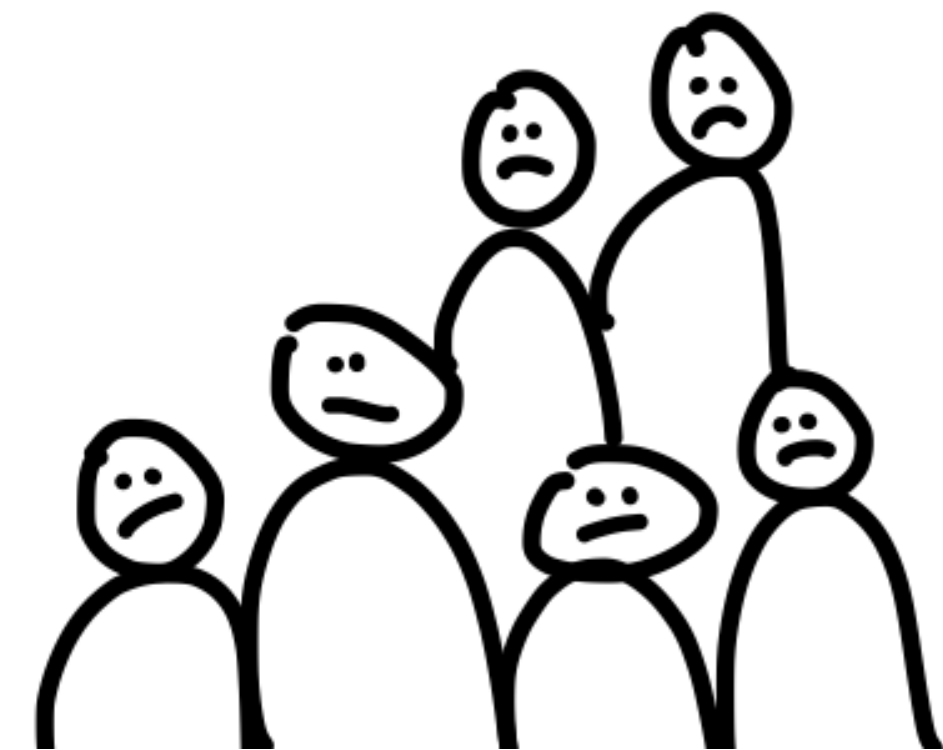
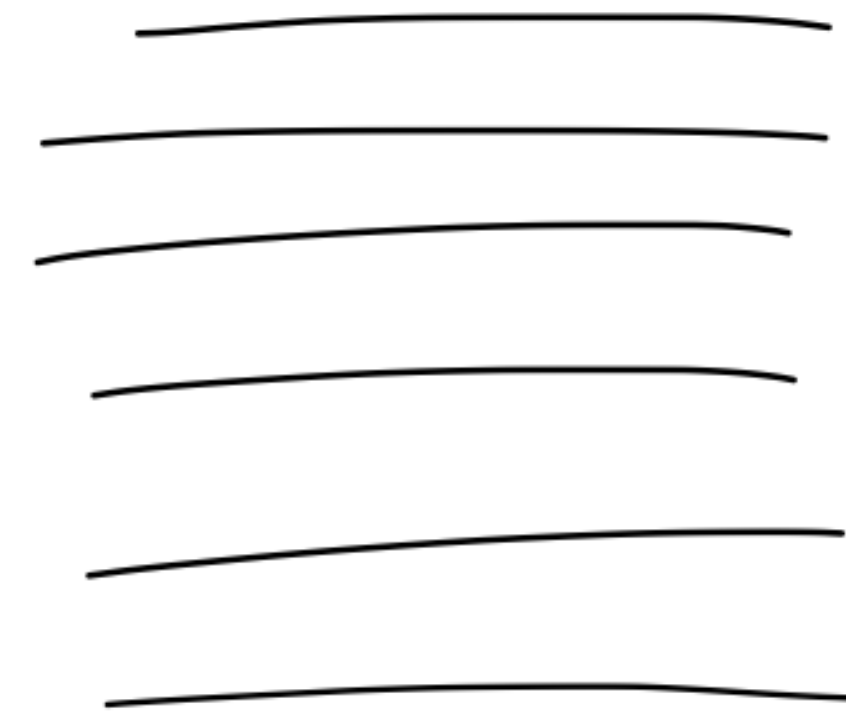


Skill progression
demonstrated by
users in Scratch
Scaffidi & Chambers



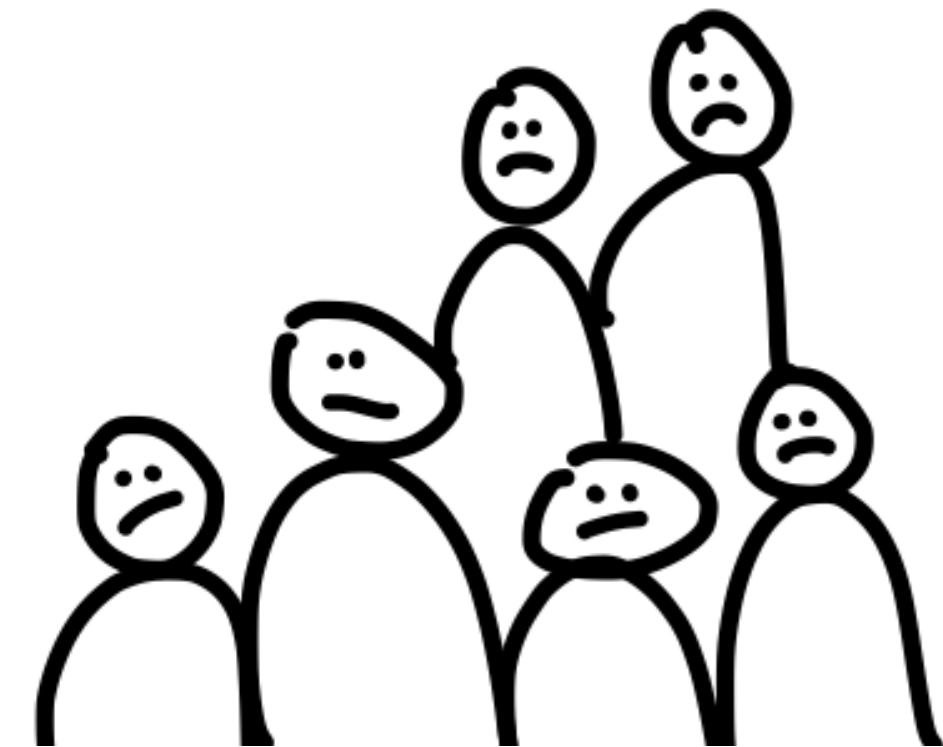
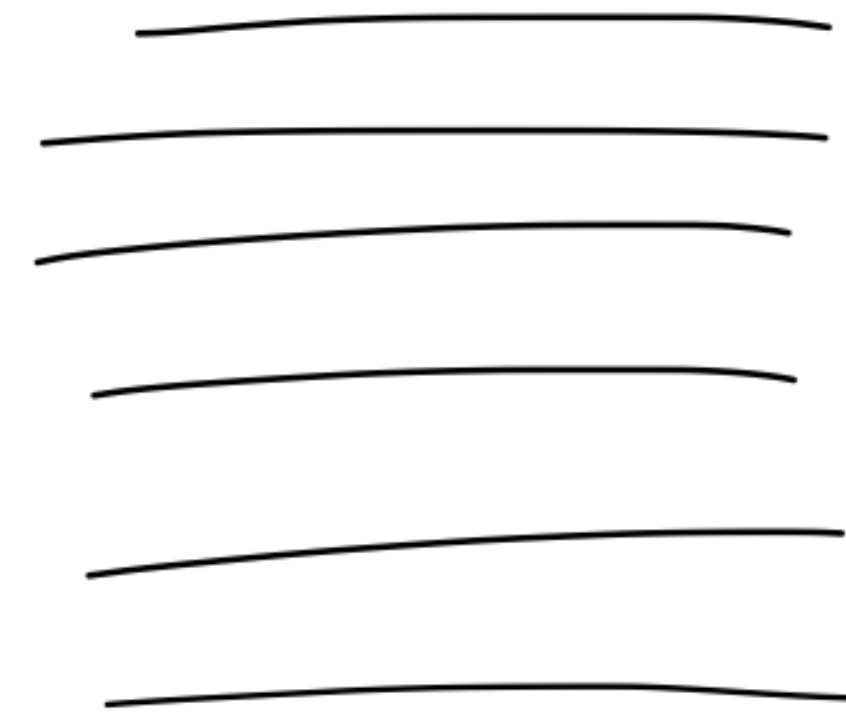
— High dropout

Skill progression
demonstrated by
users in Scratch
Scaffidi & Chambers

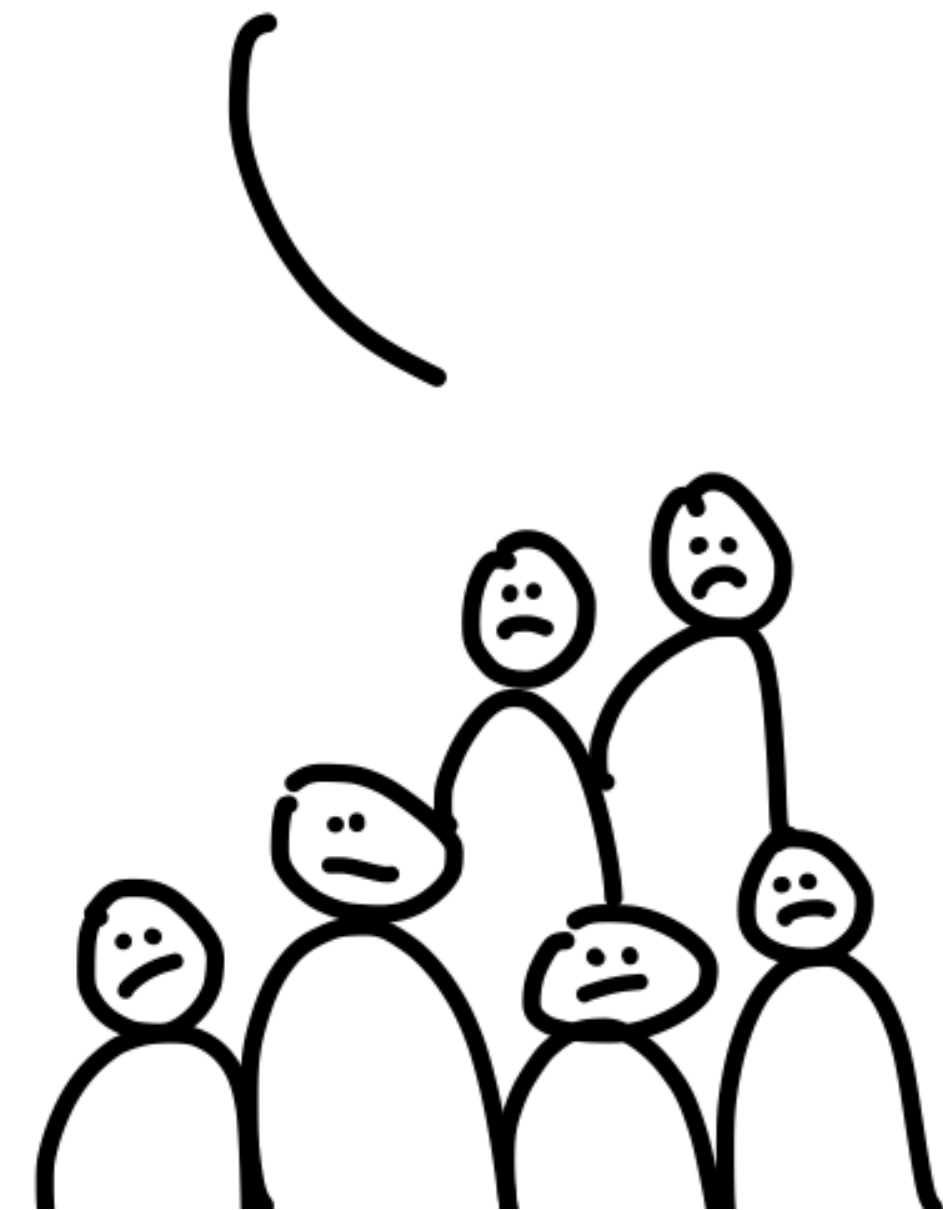


- High dropout
- Flat skill level

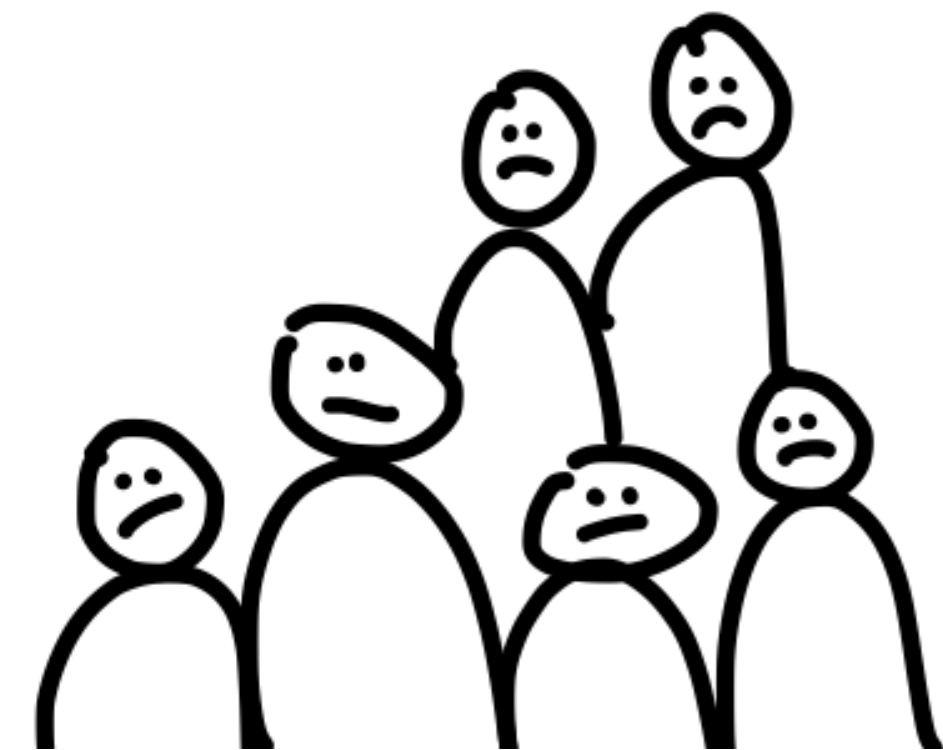
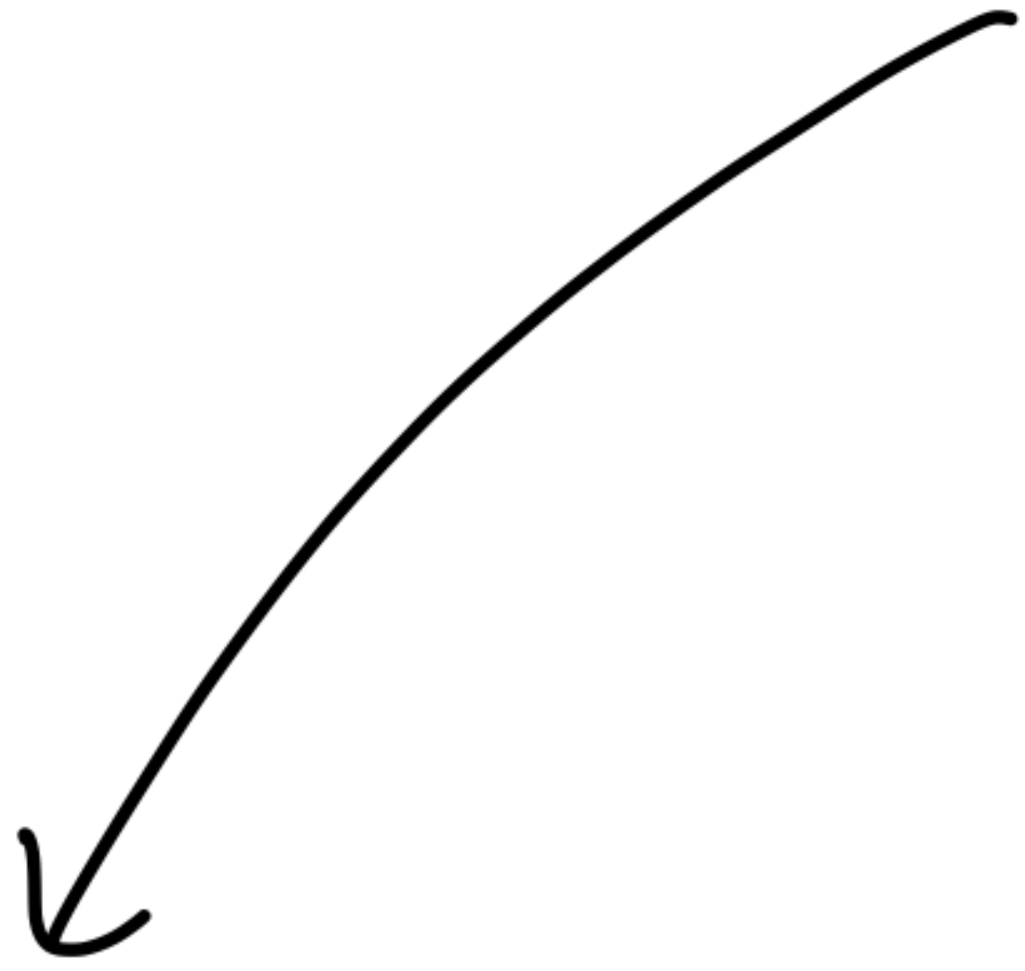
Skill progression
demonstrated by
users in Scratch
Scaffidi & Chambers



This is hard!



Doesn't know anything
about teaching



4 years in
School



4 years in
School



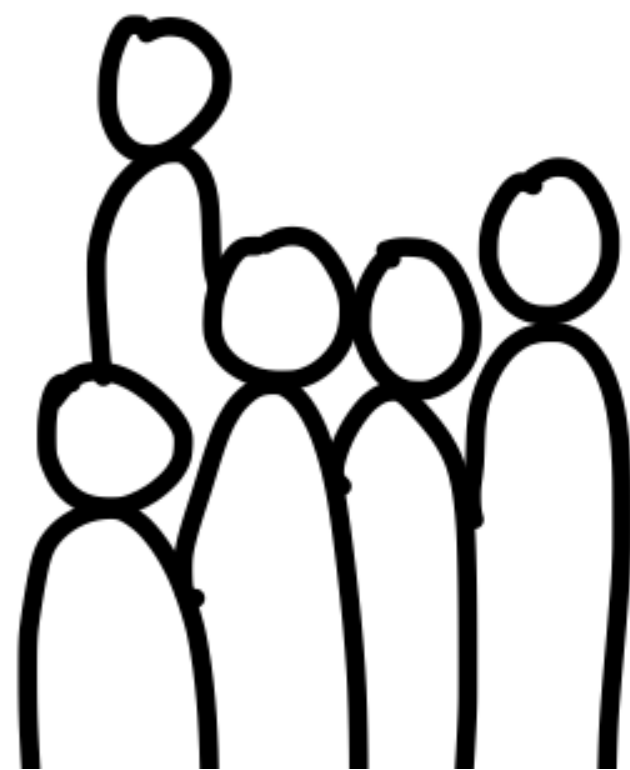
5 years in
School



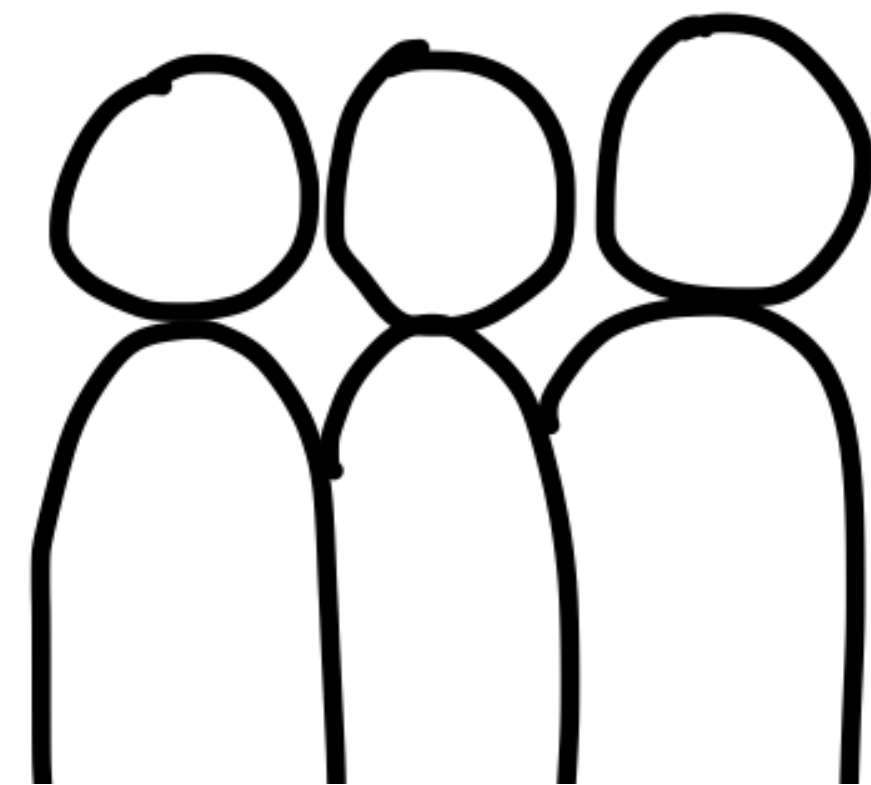
4 years in
School



5 years in
School



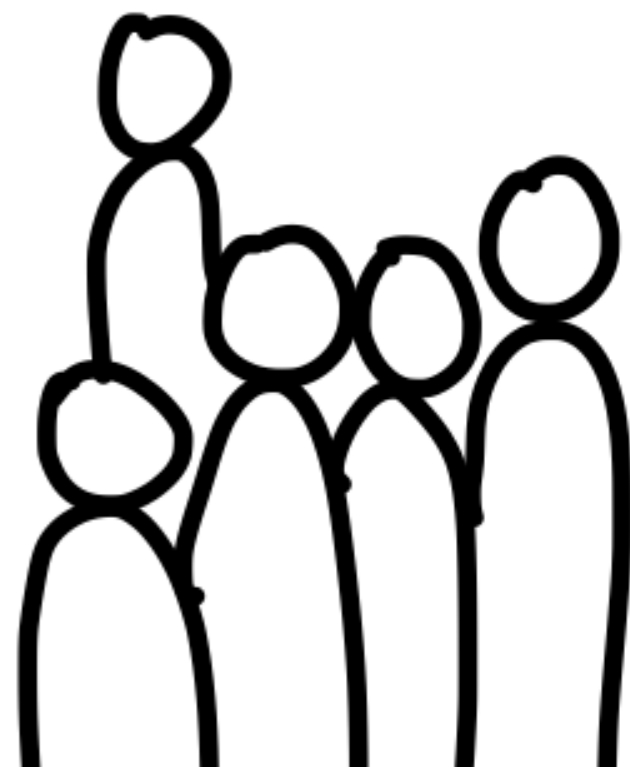
PhD



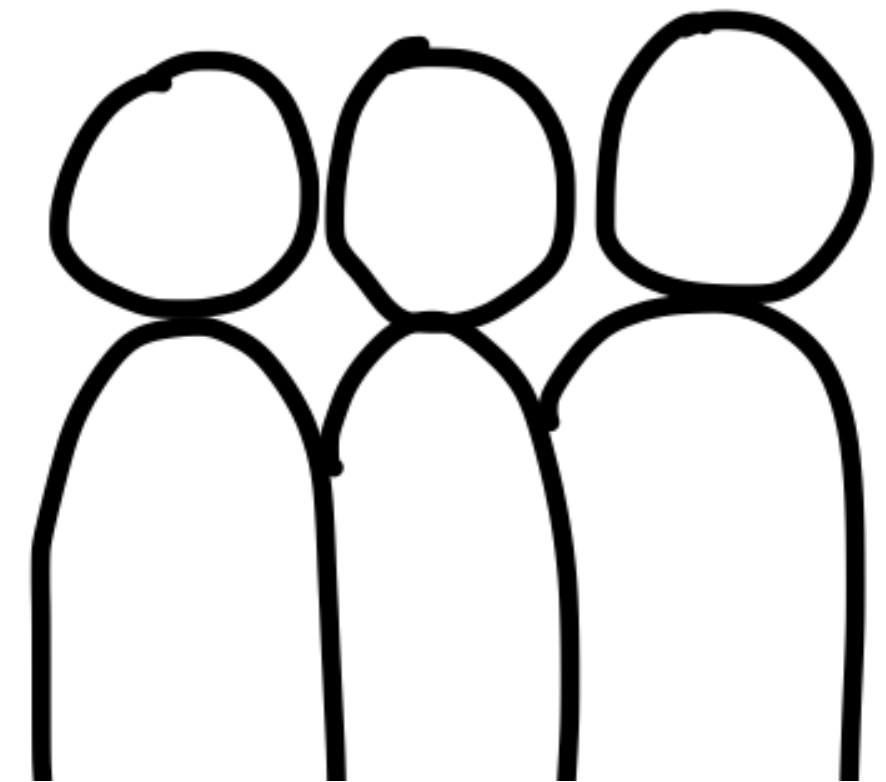
4 years in
School



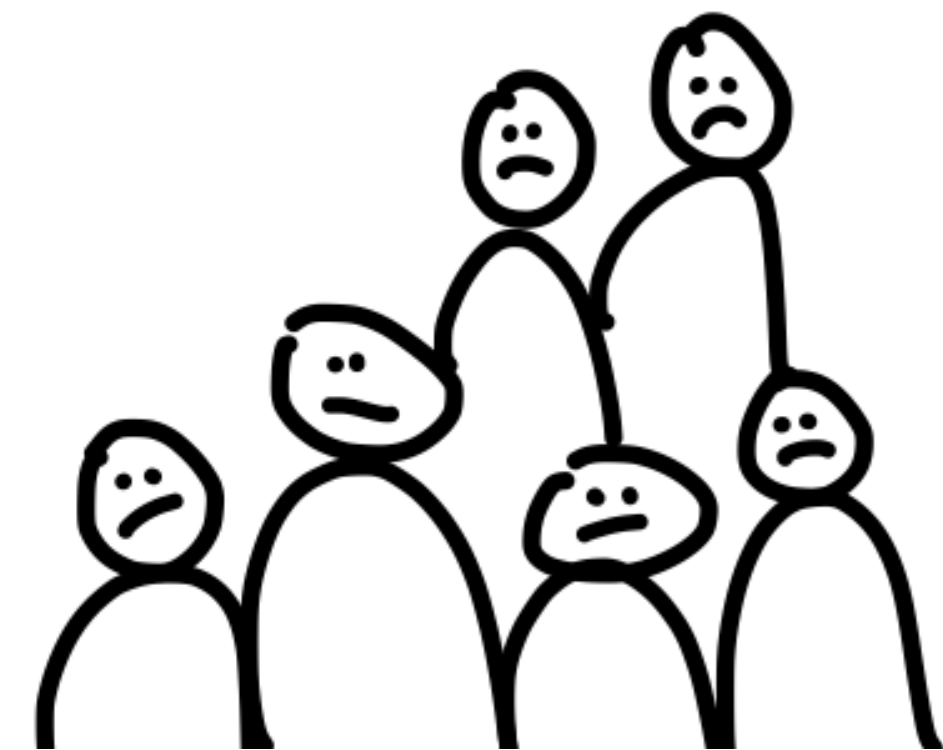
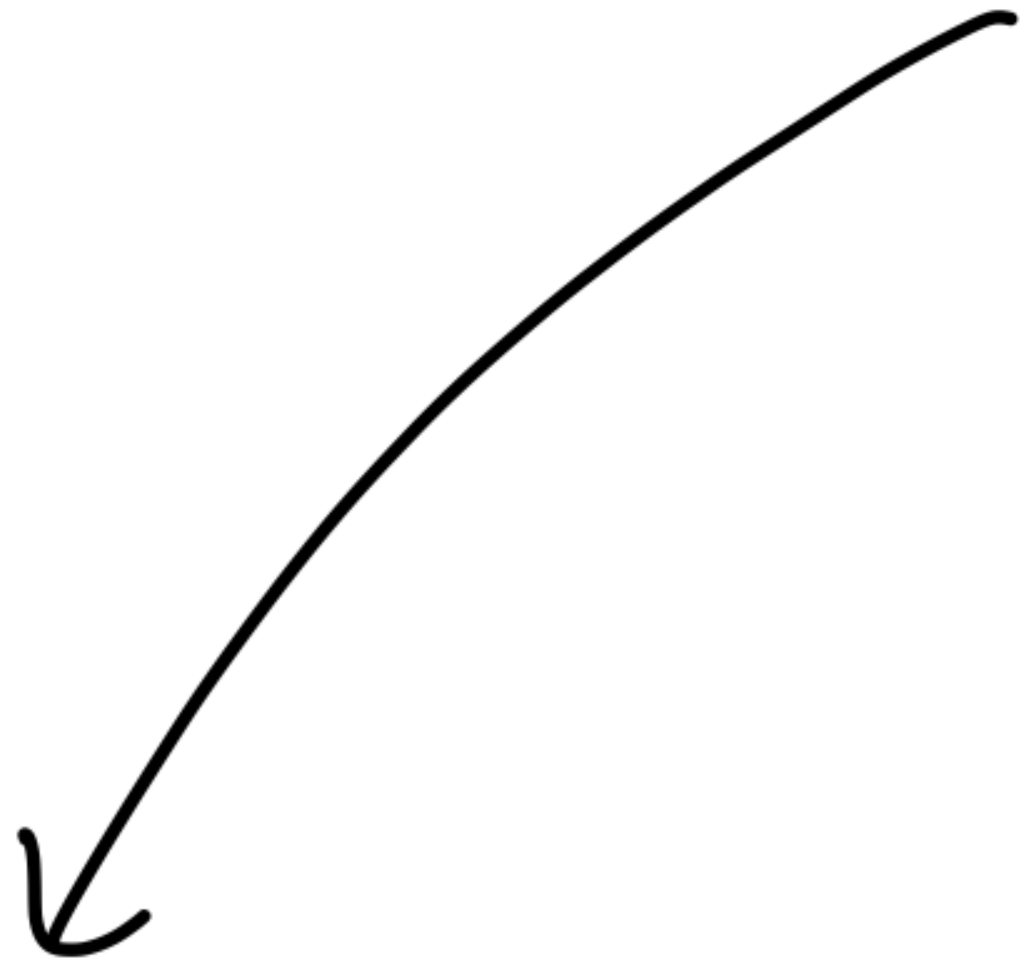
5 years in
School



PhD +
12 afternoons
of Bloom



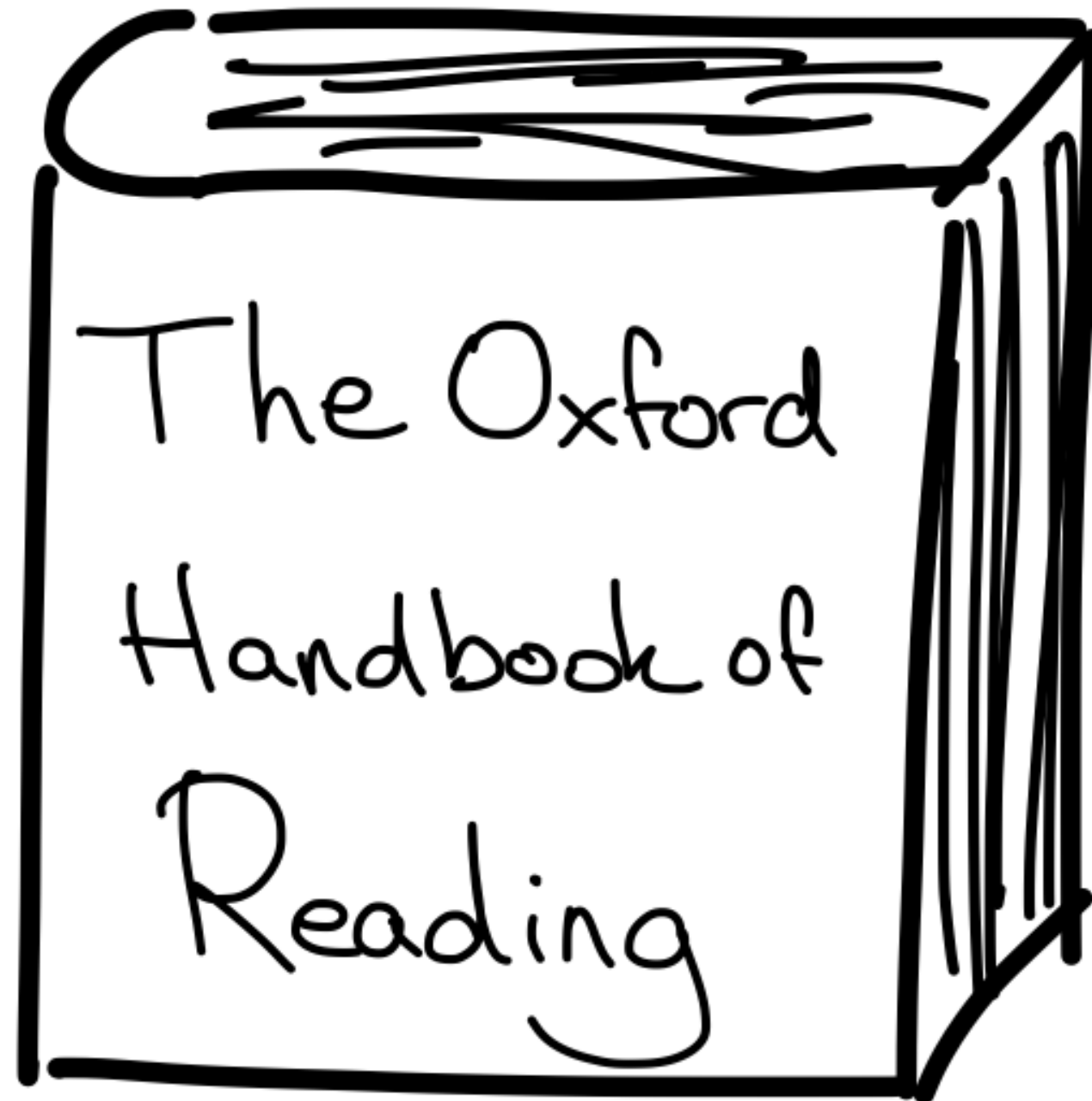
Doesn't know anything
about teaching



How do we

teach things?





1) We know nothing

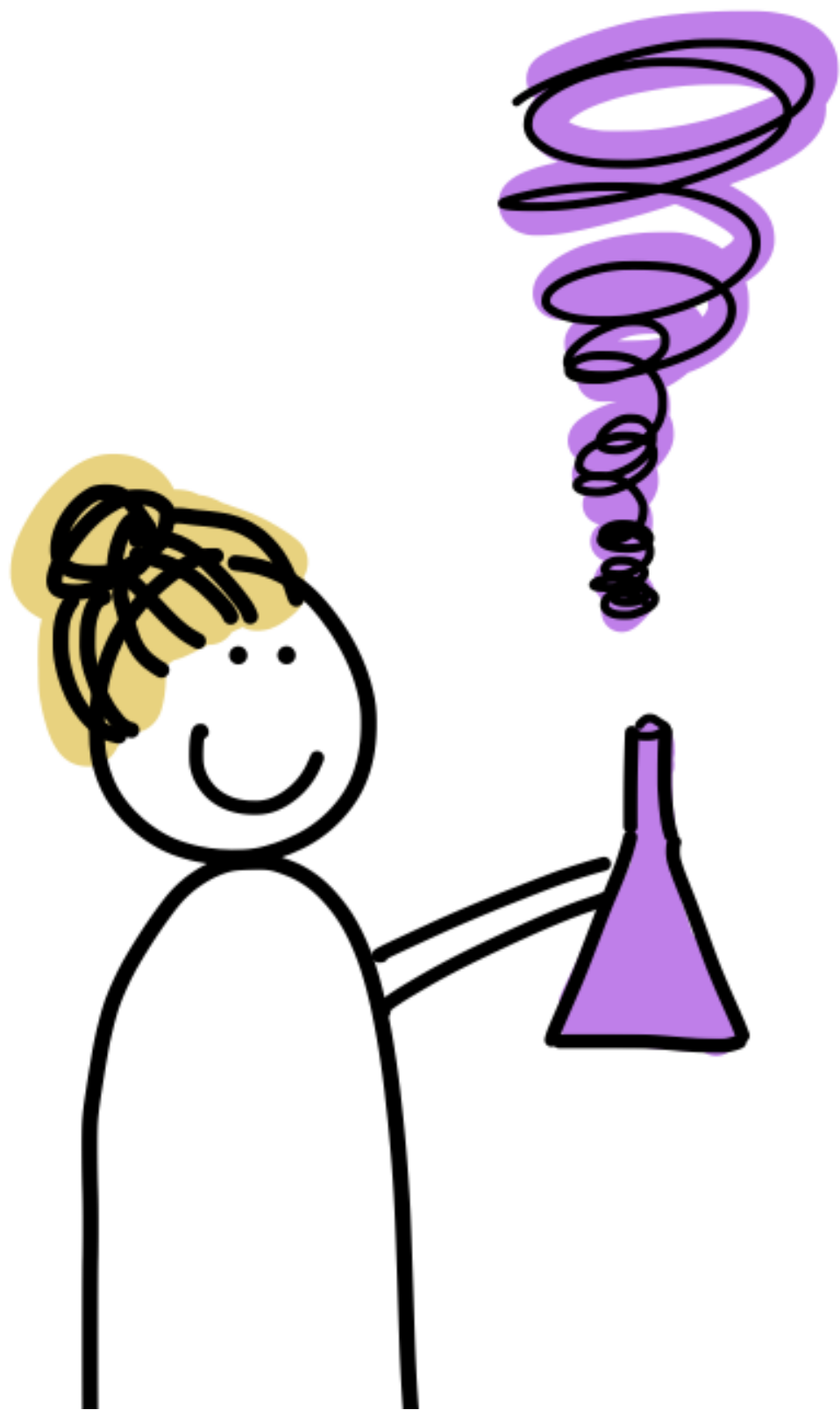


Sound matters

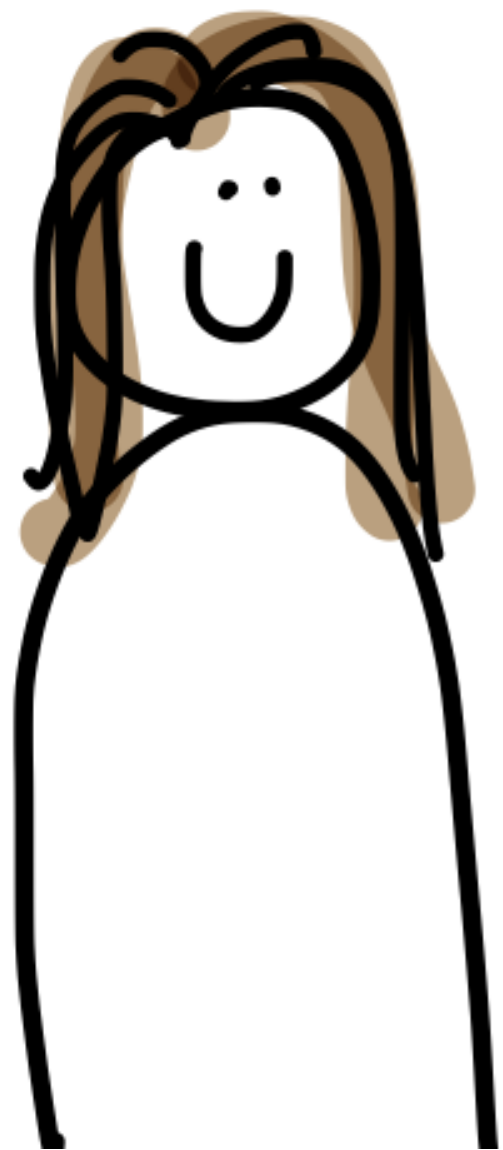


Sound matters

Let's do
an experiment!

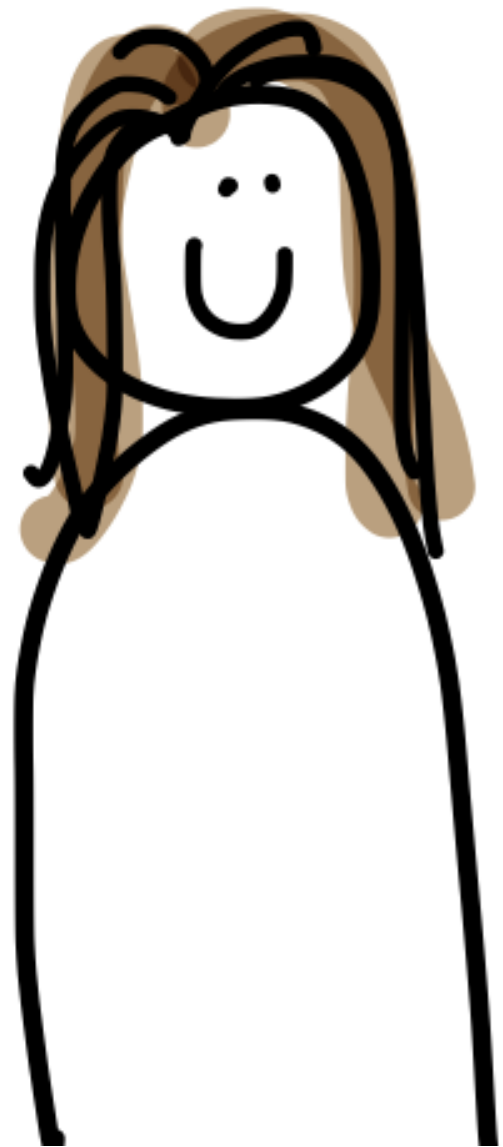


Sound matters



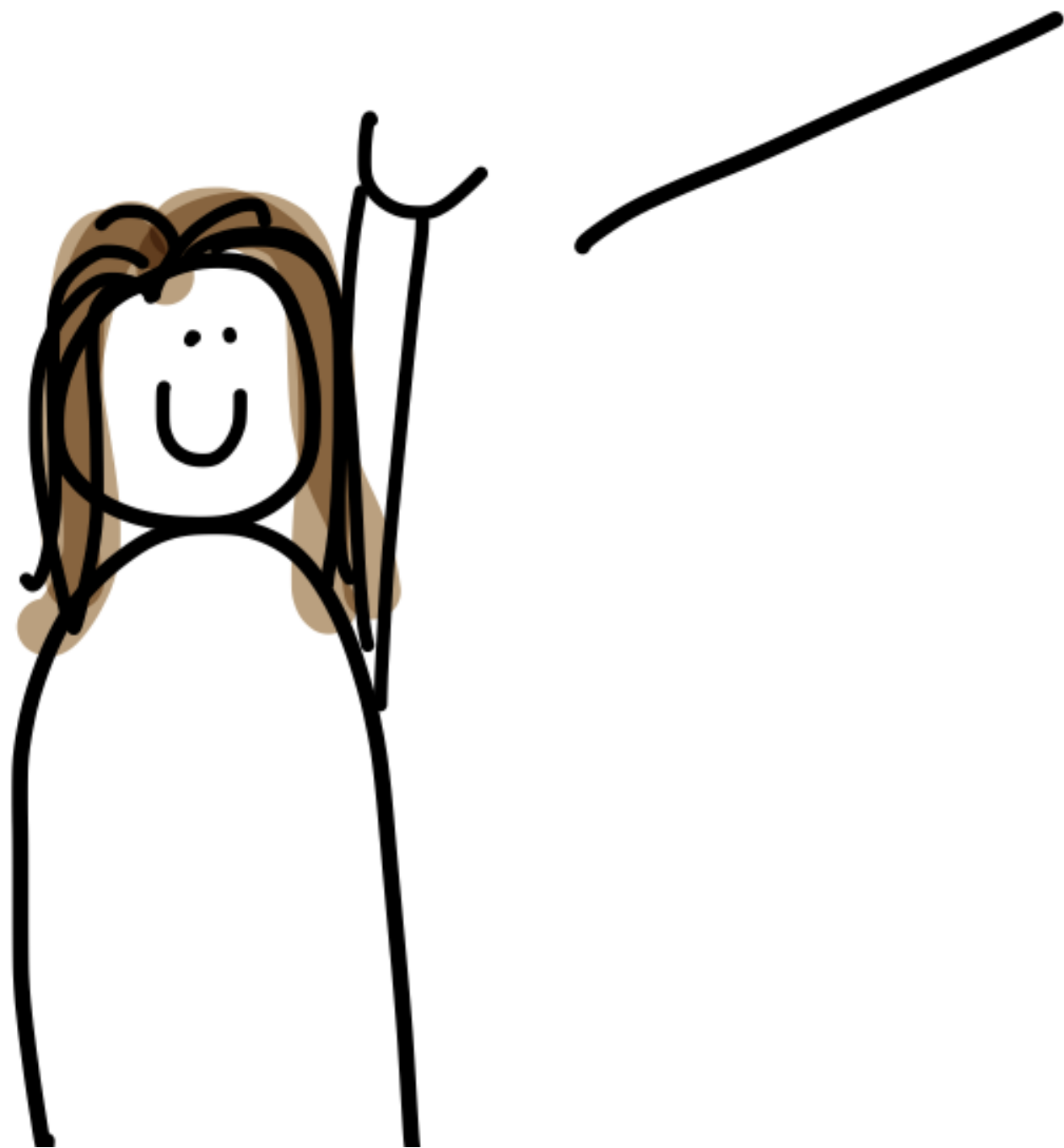
Sound matters

There is a cat
... in the tree



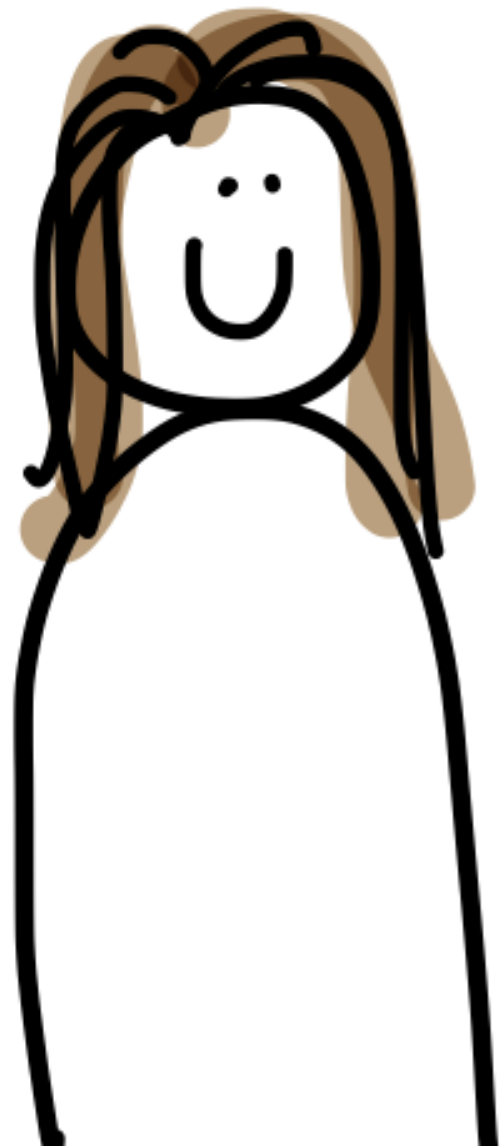
Sound matters

There is a cat
... in the tree



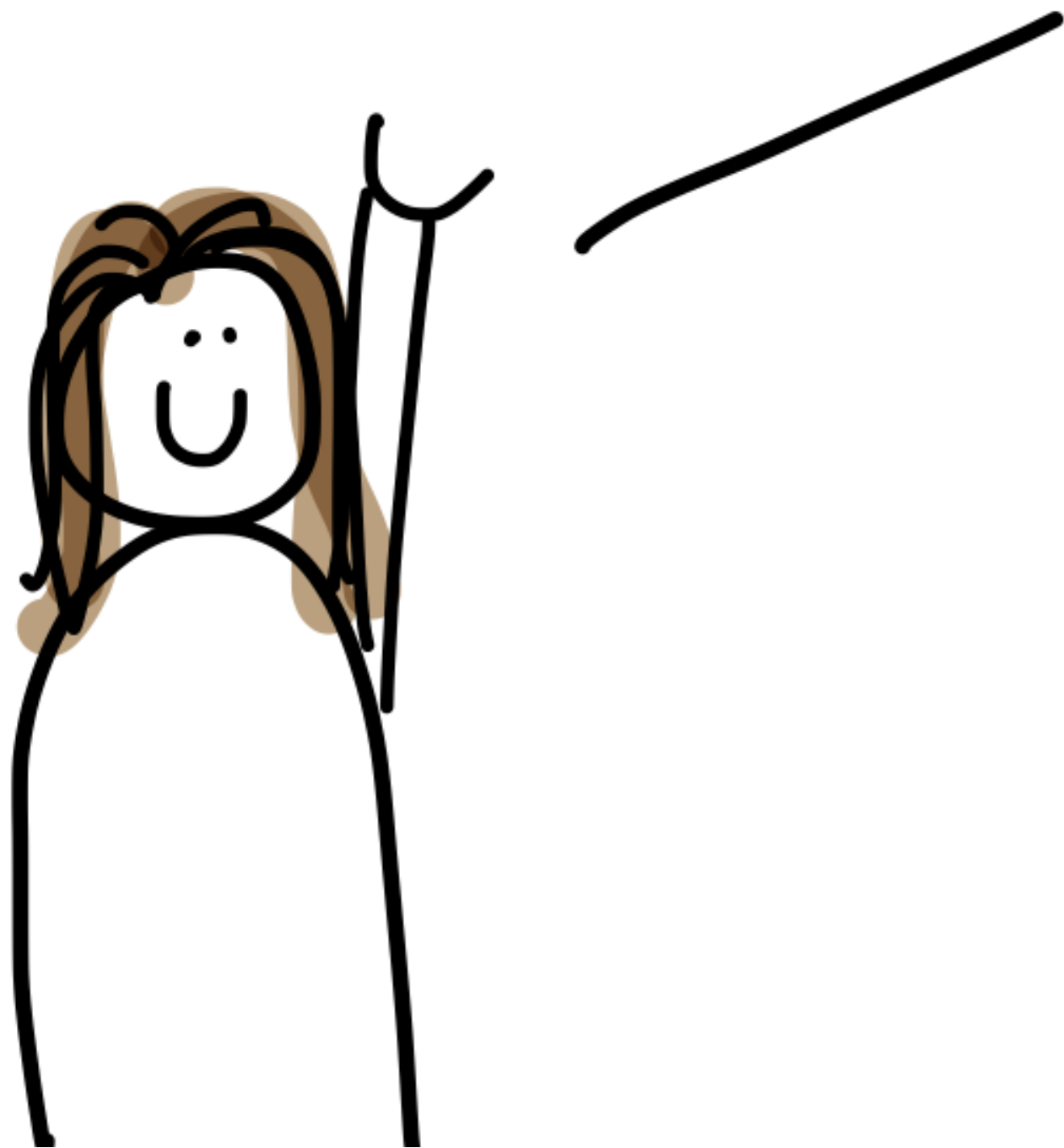
Sound matters

There is a cat



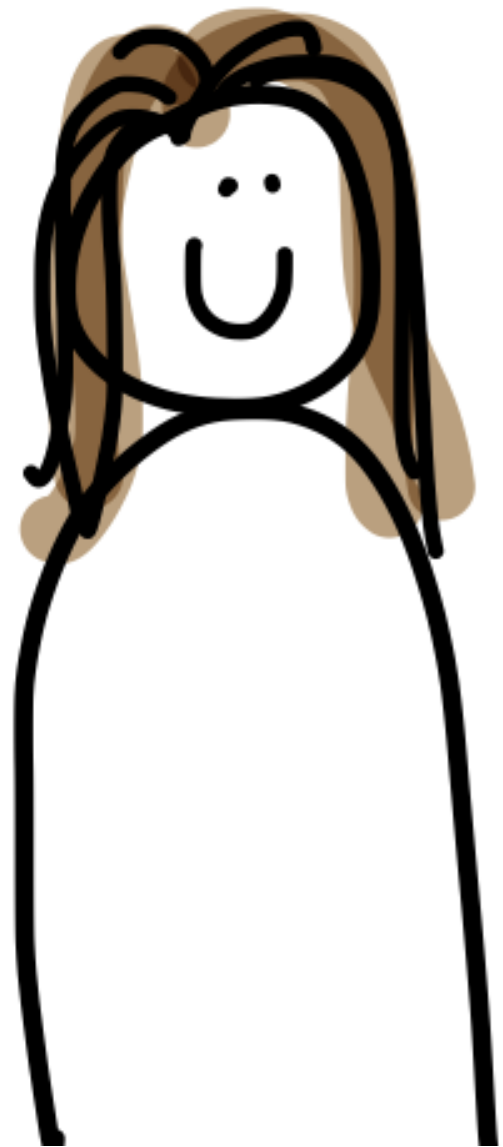
Sound matters

There is a cat
... in the tree



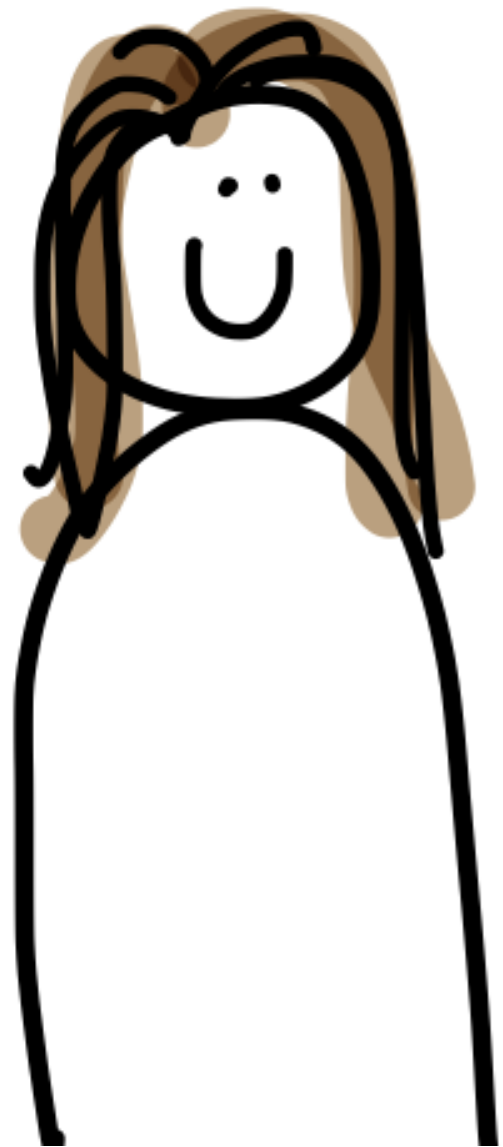
Sound matters

A turtle swims



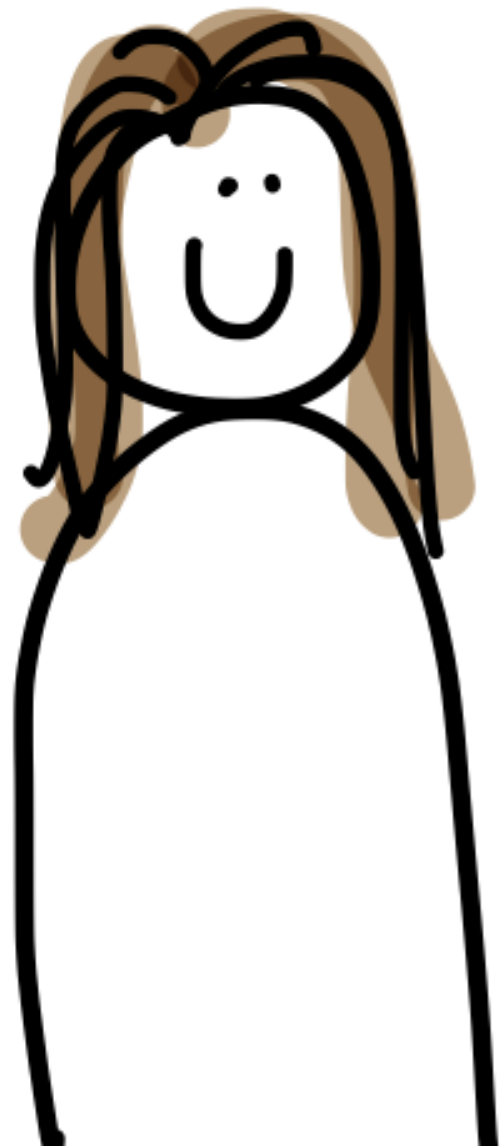
Sound matters

A turtle swims
... in the lake



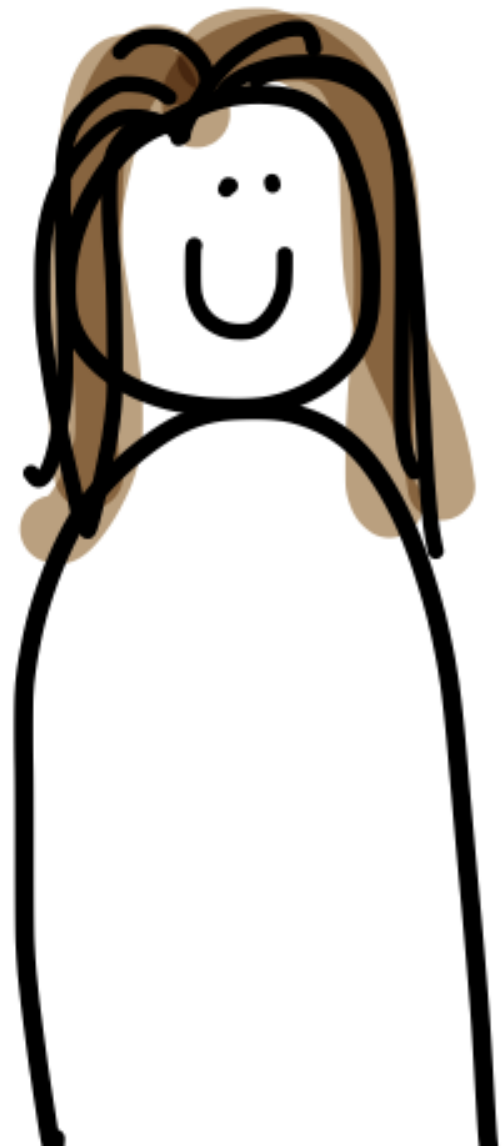
Sound matters

I have a tear

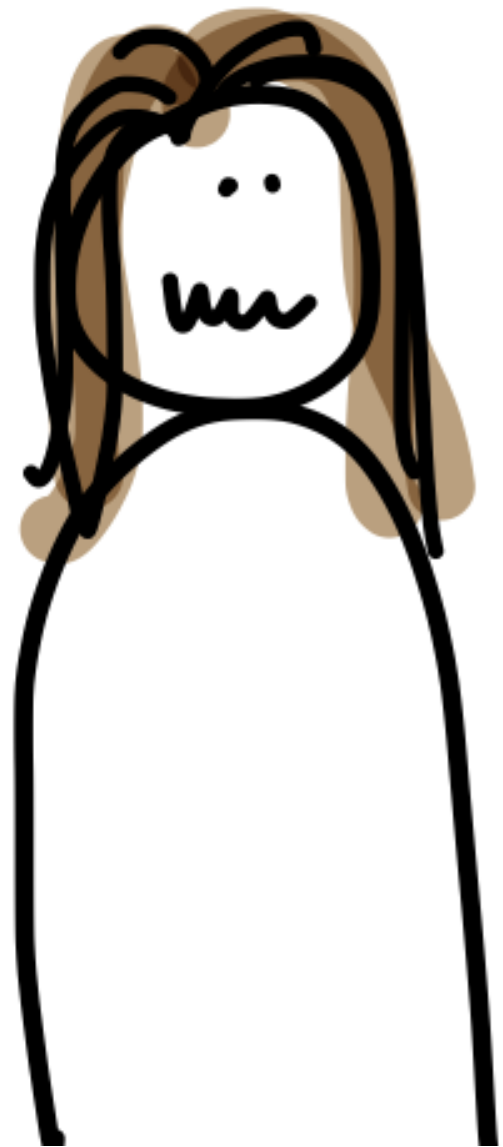


Sound matters

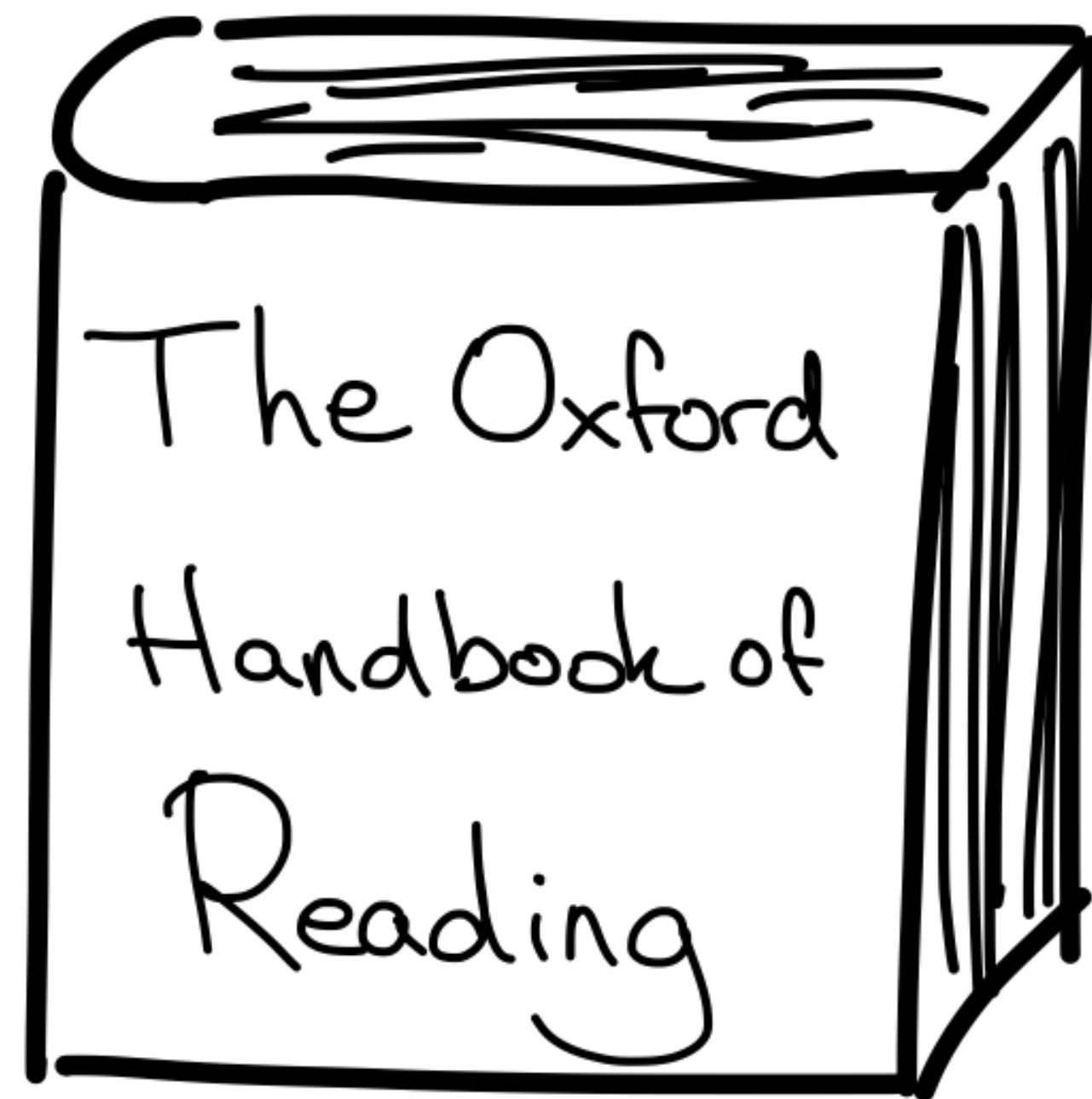
I have a tear
... in my pants



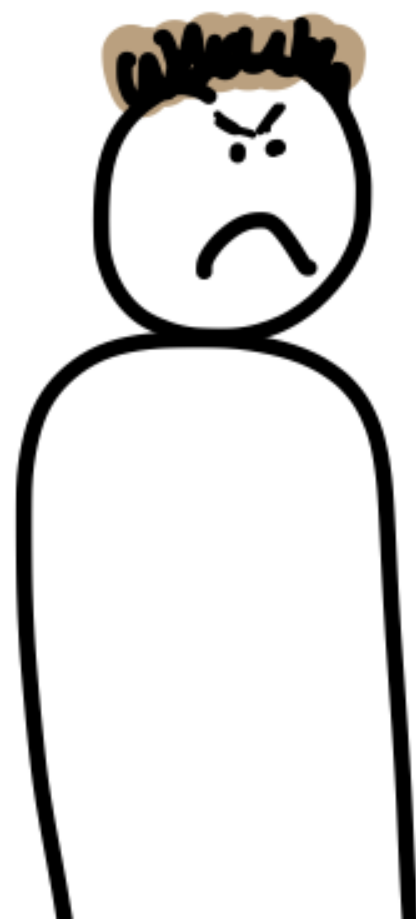
Sound matters



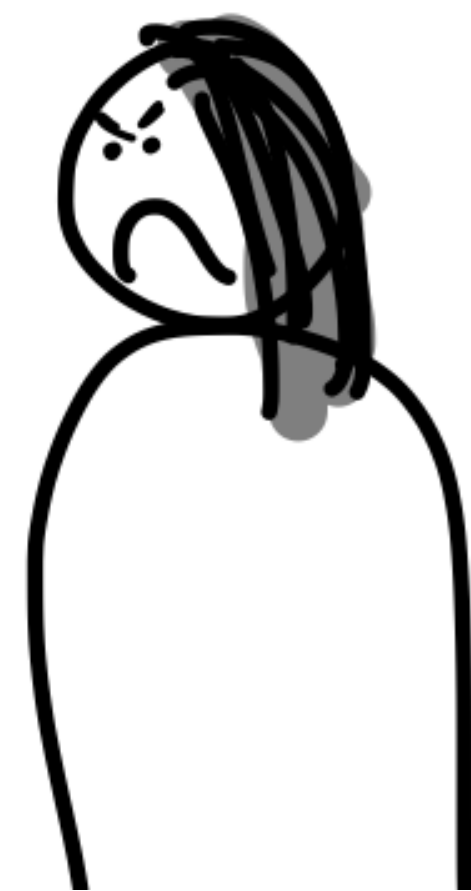
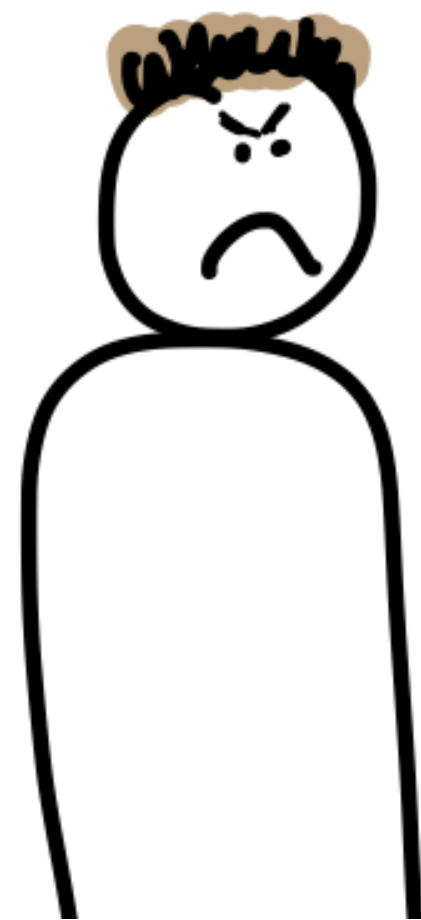
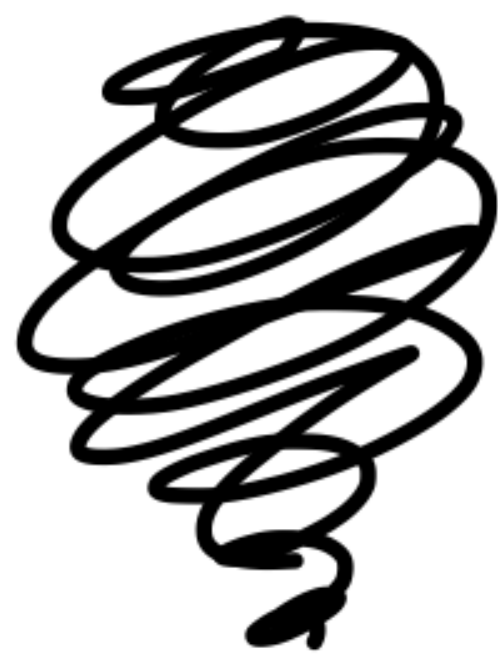
1) We know nothing



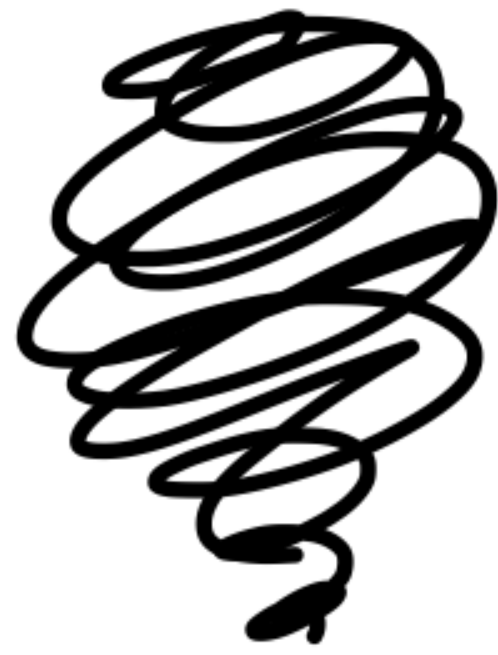
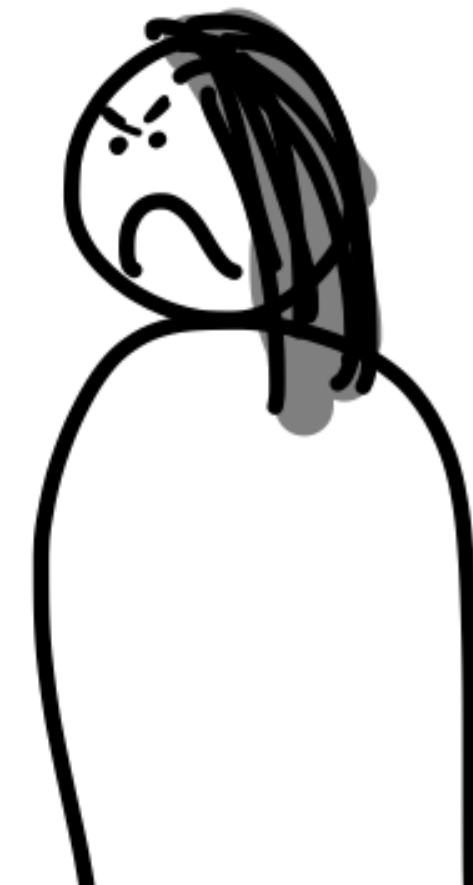
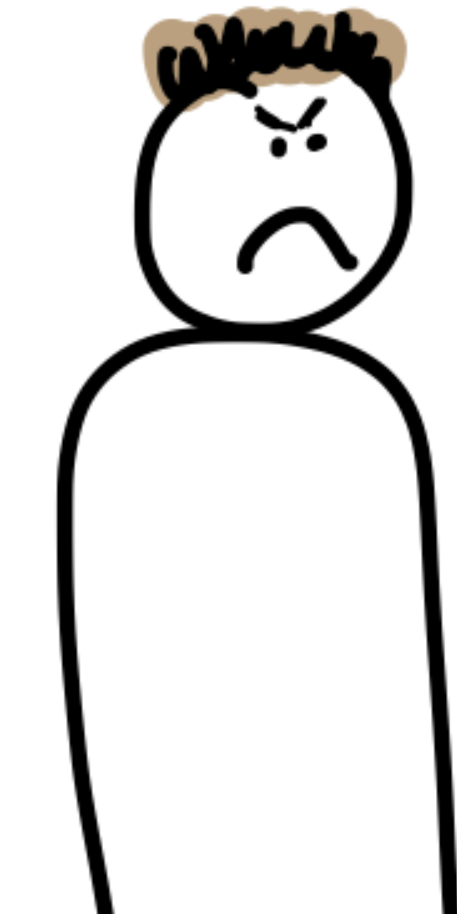
2) We lack fights



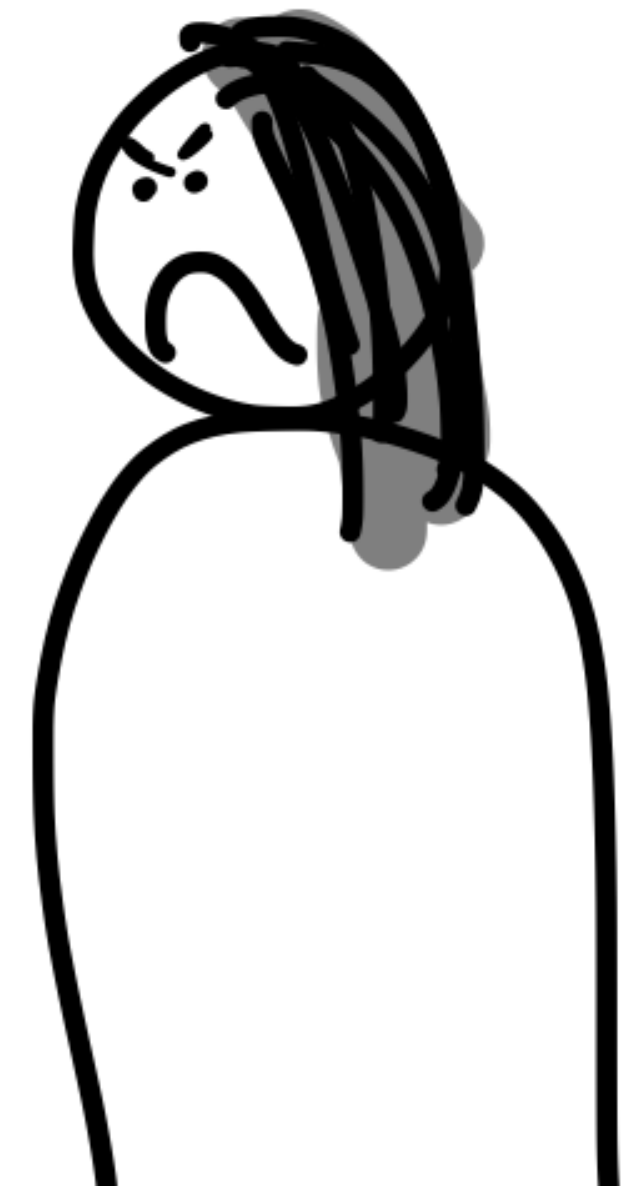
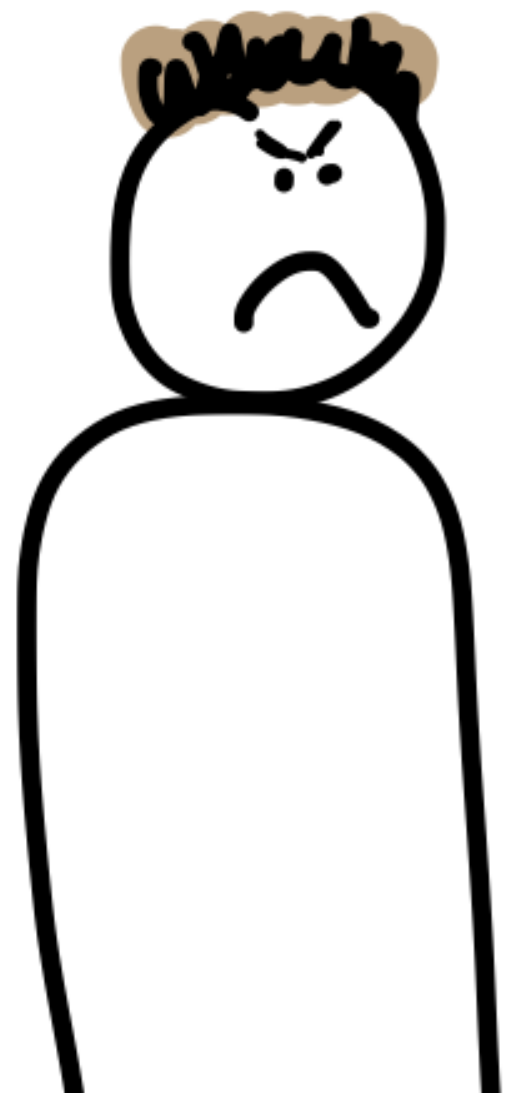
2) We lack fights



2) We lack pedagogical diversity

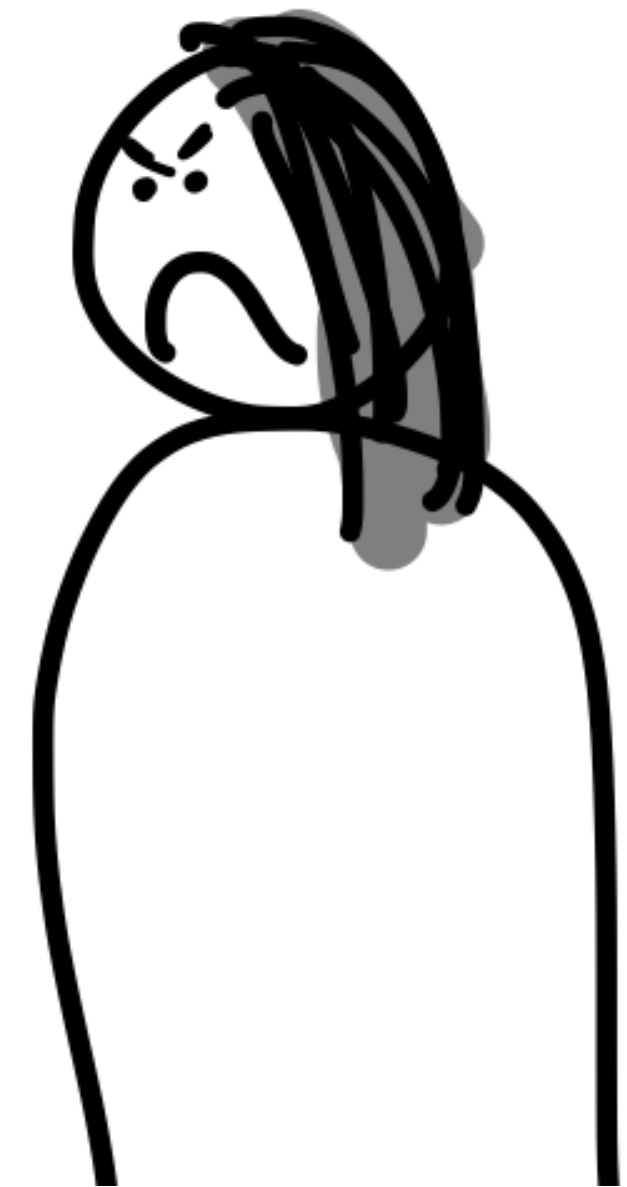
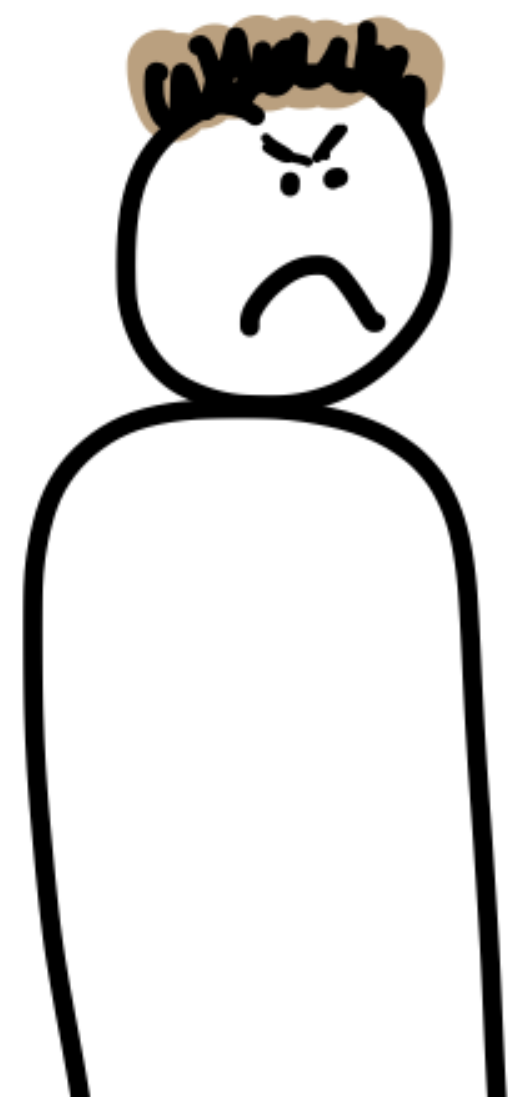


Phonics vs. Whole Language



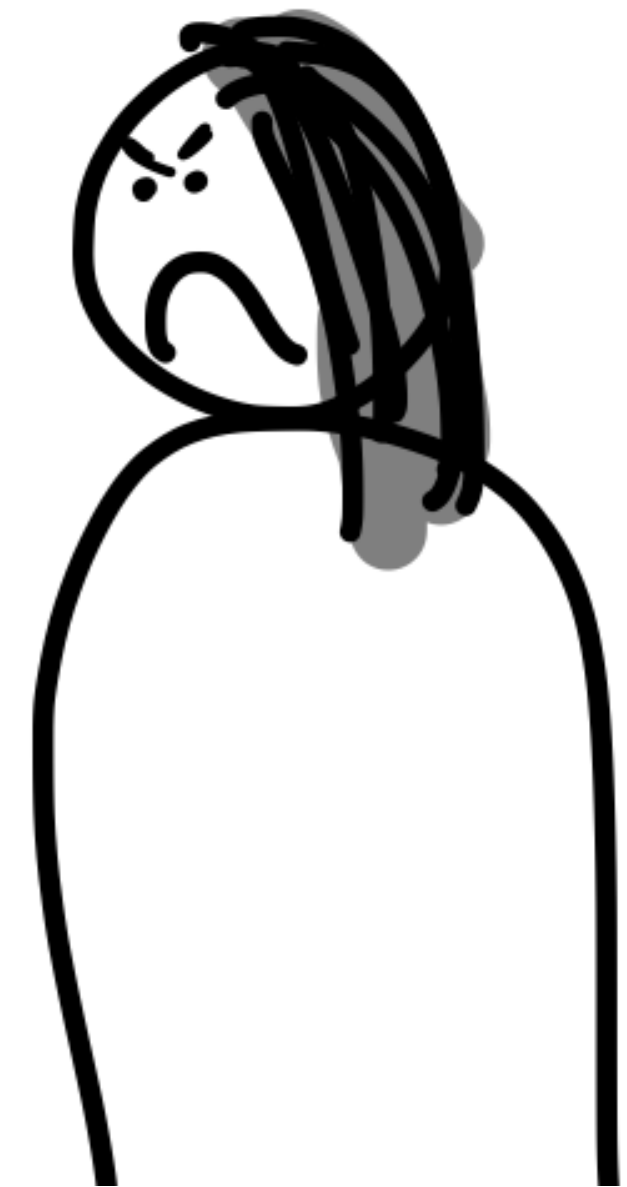
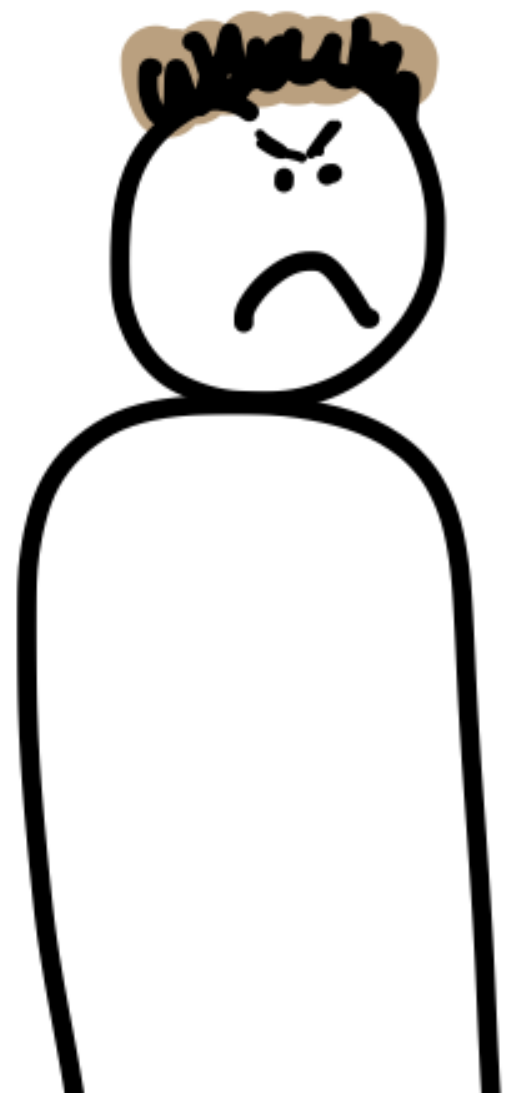
Phonics vs. Whole Language

r · e a · d



Phonics vs. Whole Language

I love books ~



The math wars



The math wars

Explanation
&
practice



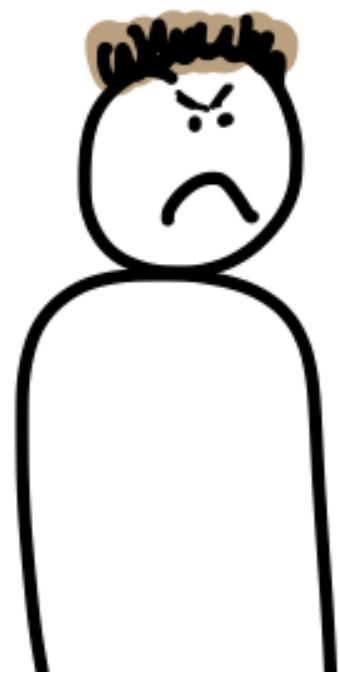
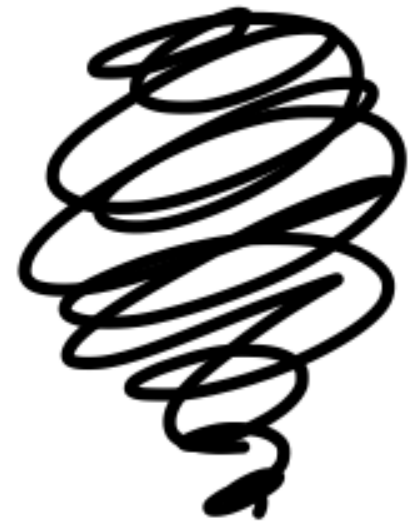
The math wars

Exploration
&

Problem
solving



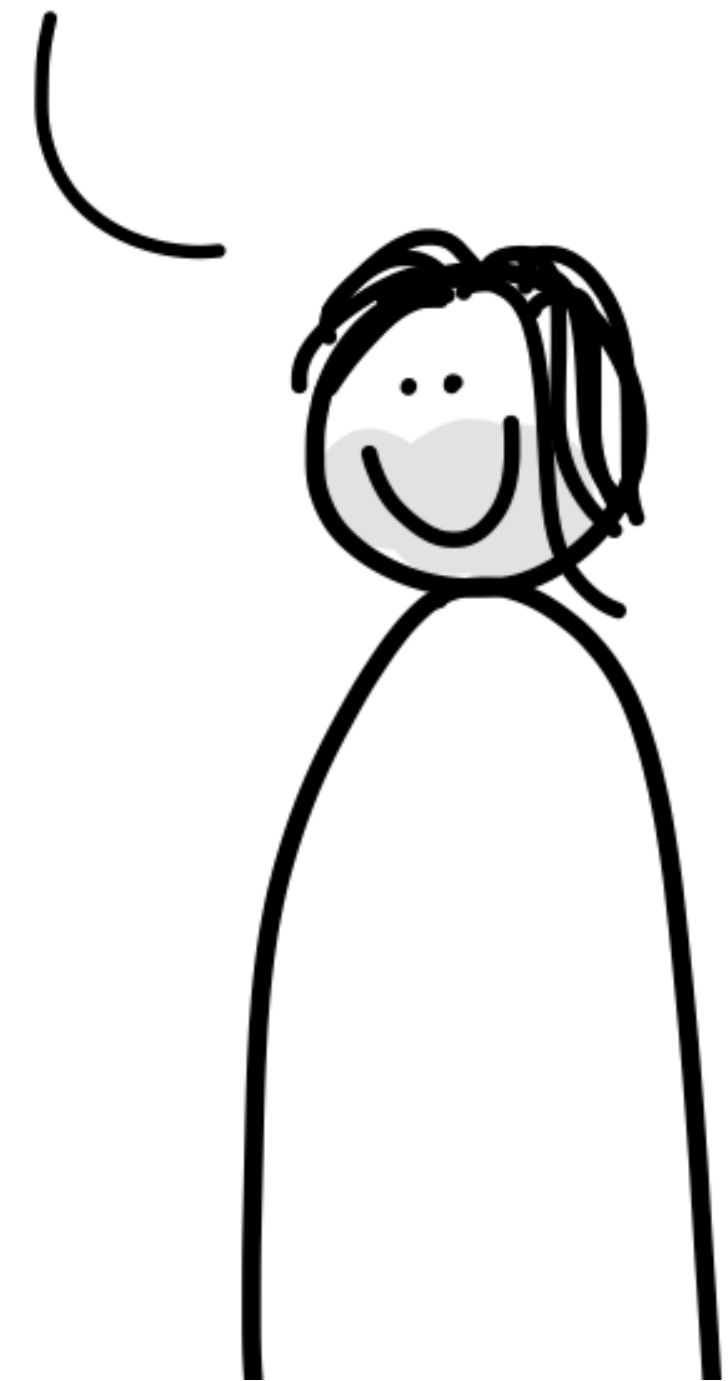
Where is programming's
pedagogical debate?



How to teach programming?

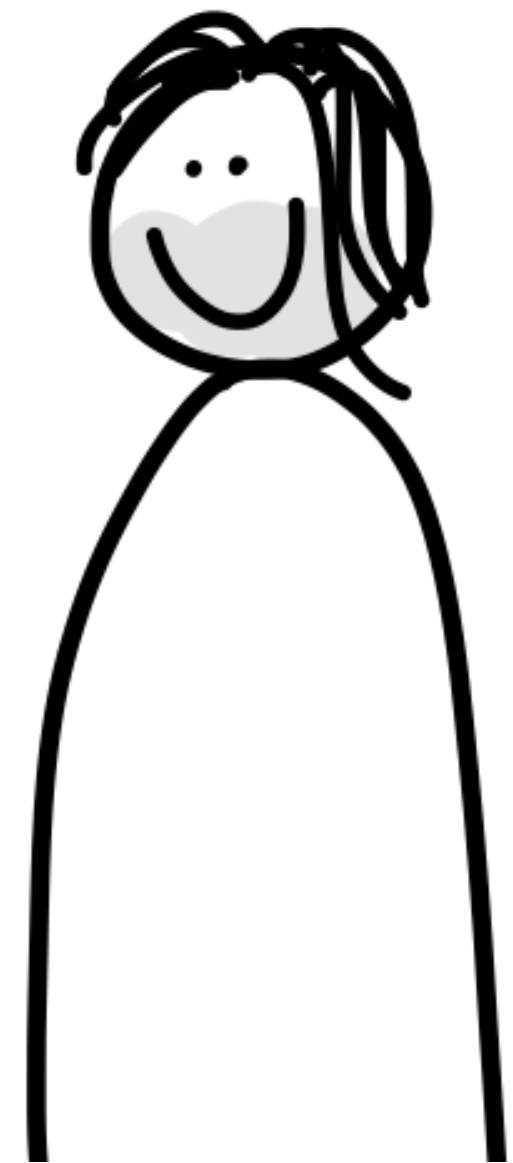
How to teach programming?

I created LOGO



How to teach programming?

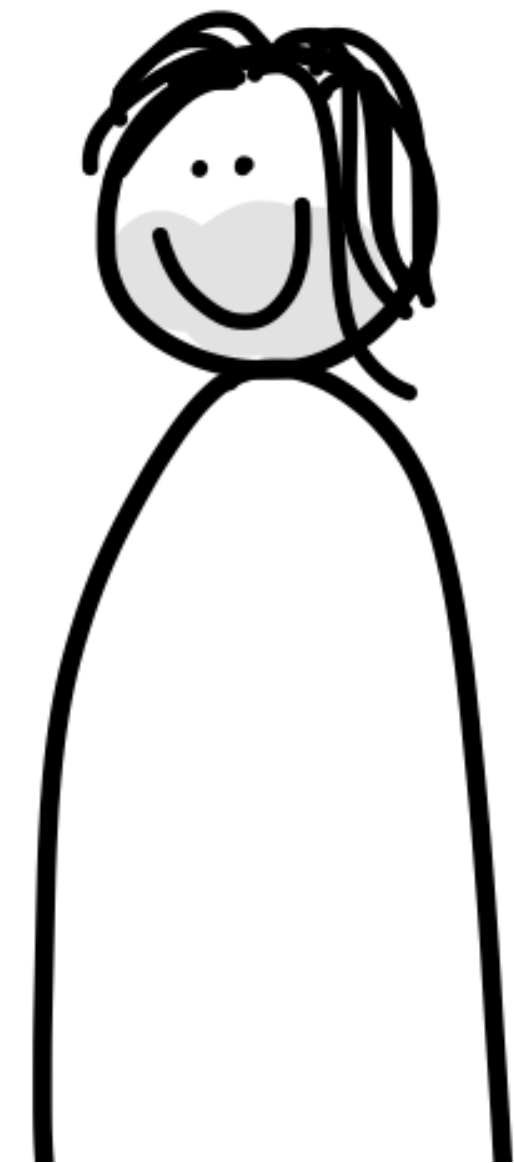
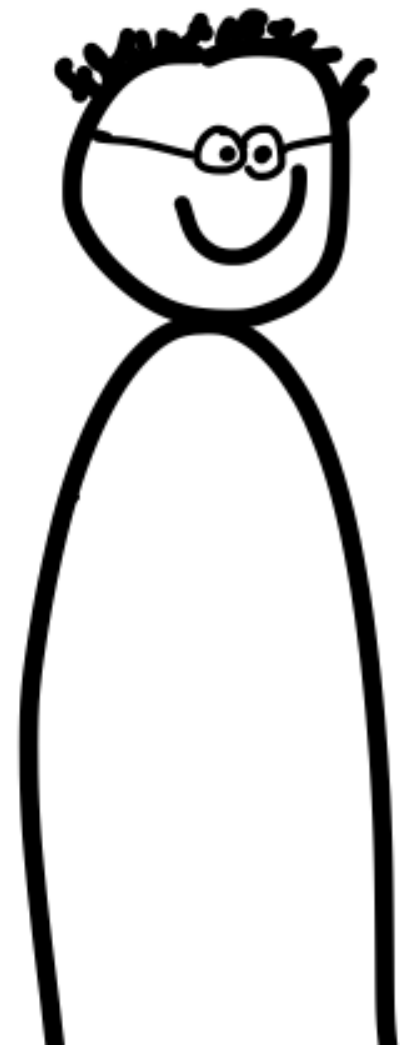
I am Seymour Papert



How to teach programming?

I am Seymour Papert

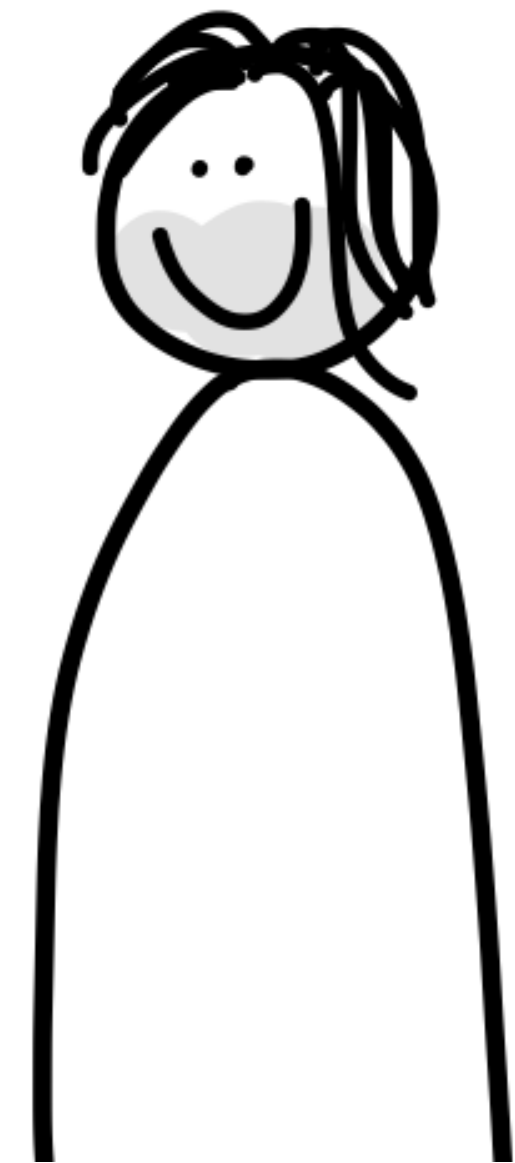
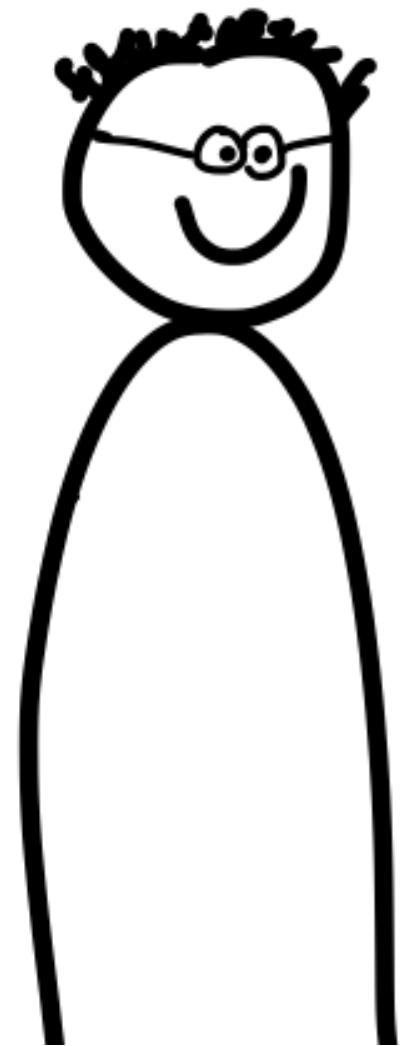
I am Jean Piaget



How to teach programming?

I am Seymour Papert

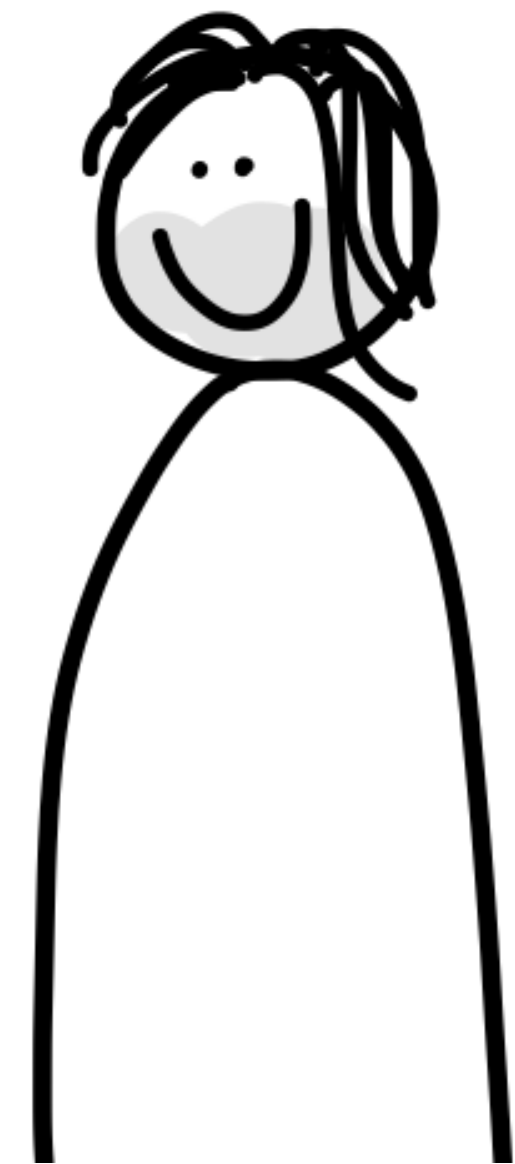
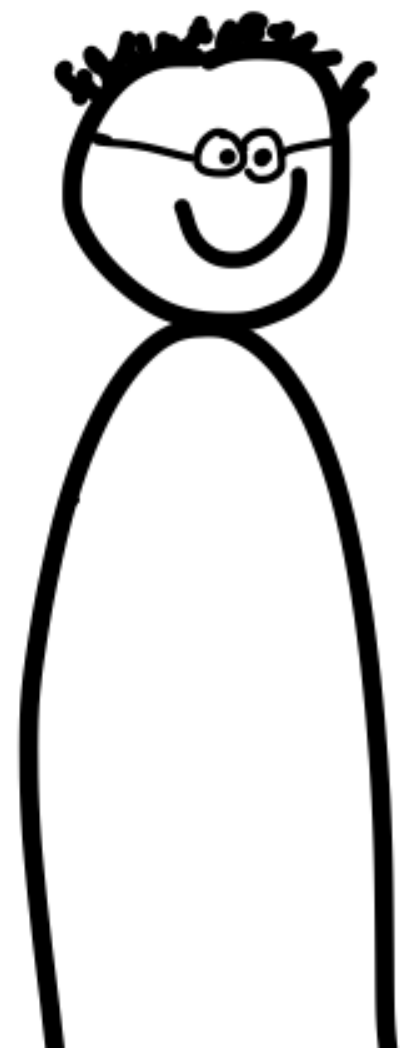
I founded
constructivism



How to teach programming?

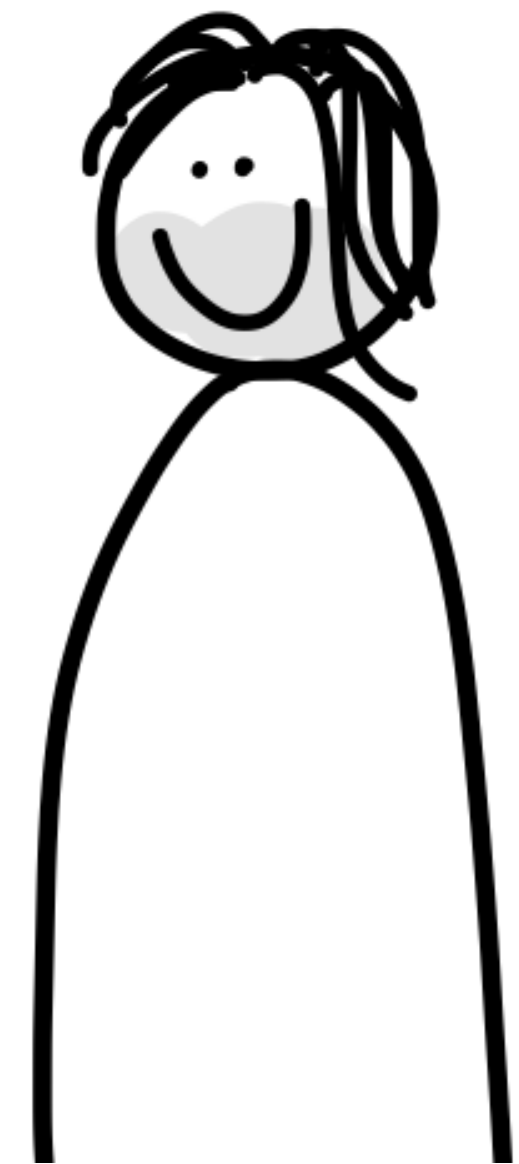
I am Seymour Papert

Nobody understands
my work better

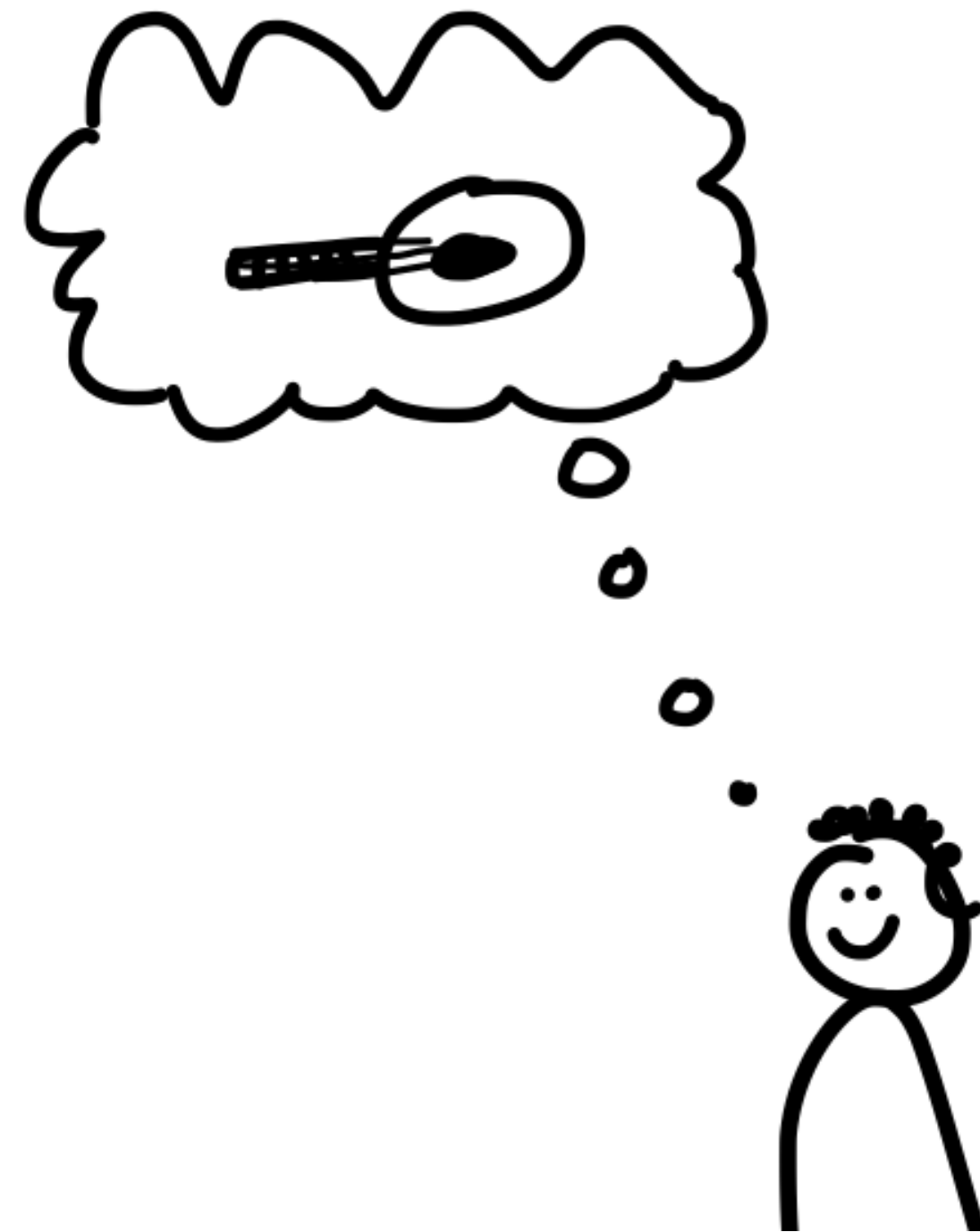


How to teach programming?

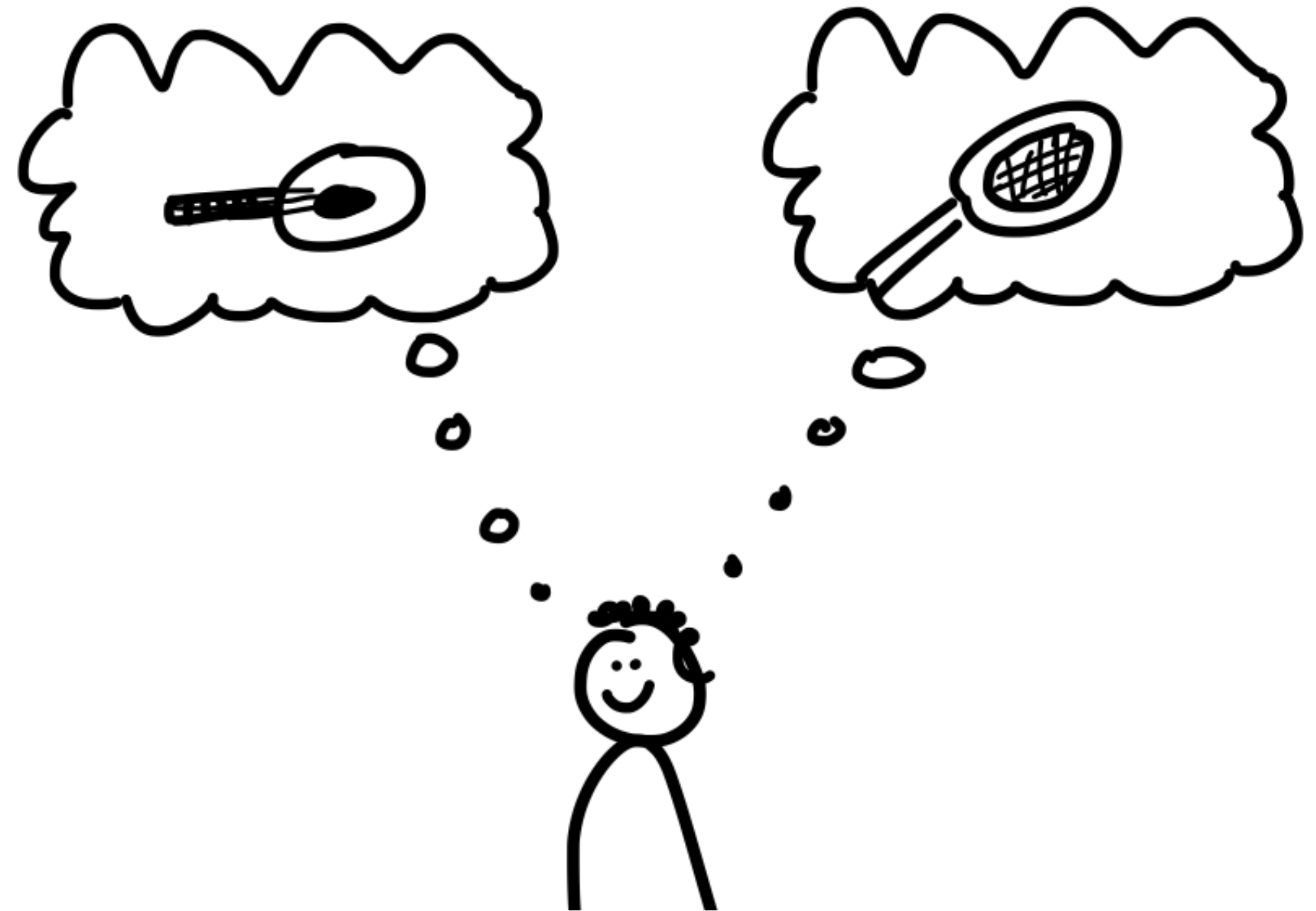
"By explaining something
you take away the
opportunity for a
child to discover it"



How to teach other things?

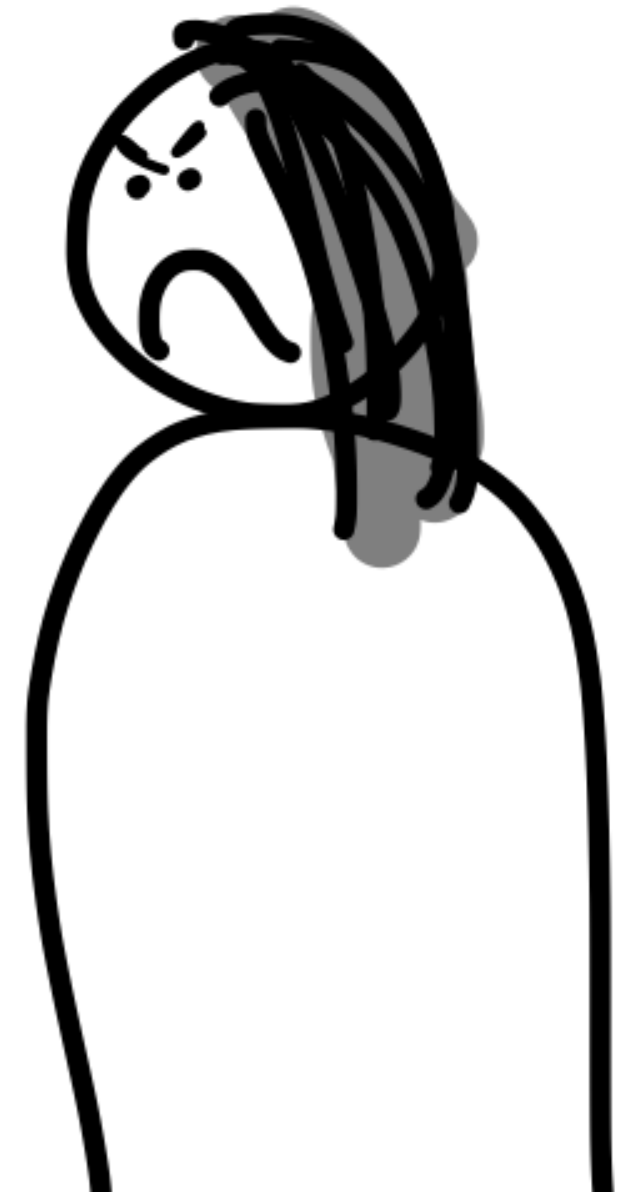
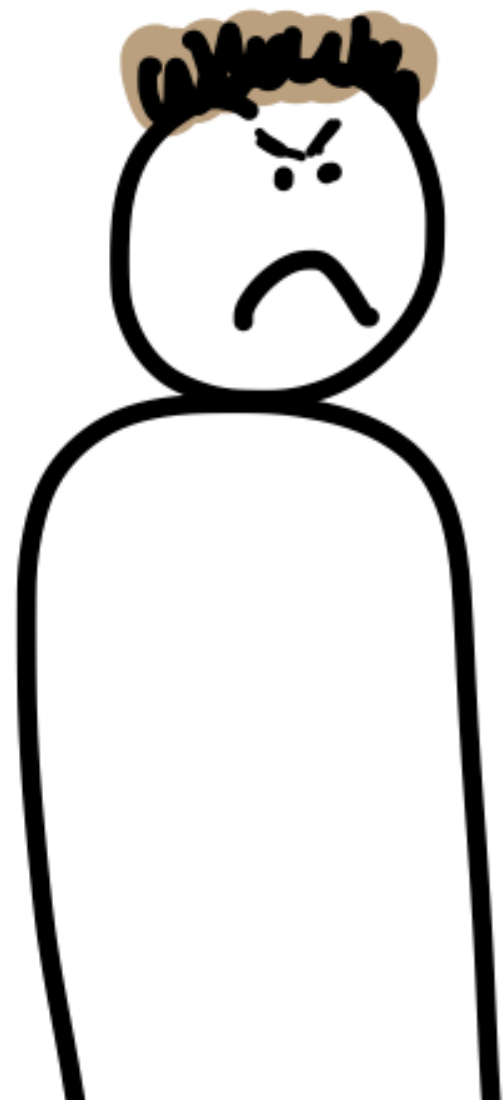


How to teach other things?



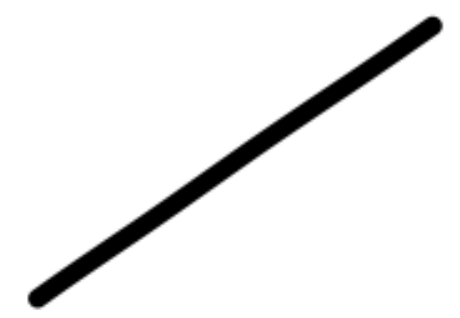
How to teach other things?

— r · e a · d

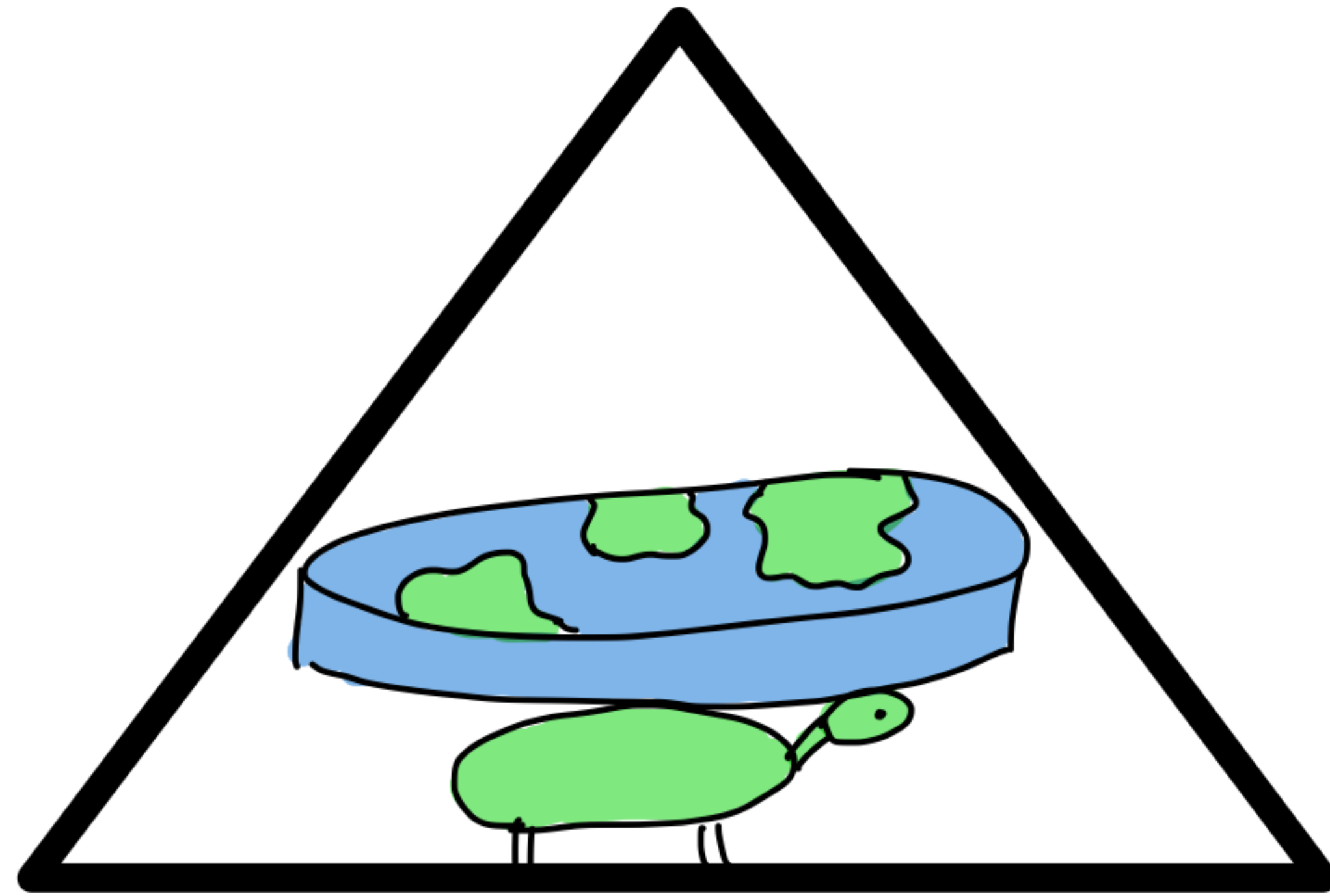


How to teach other things?

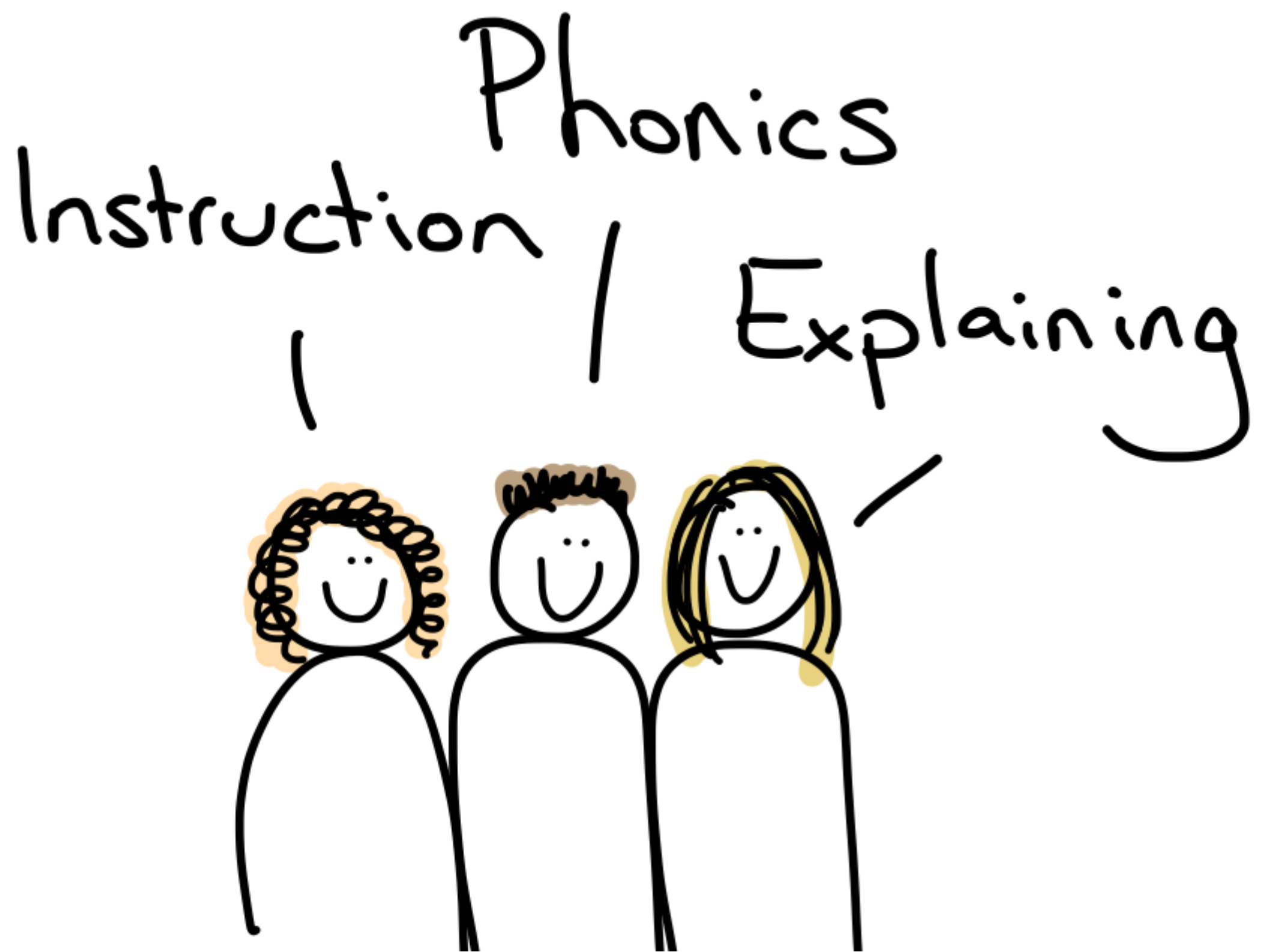
Explanation
&
practice



Beware! Controversy ahead



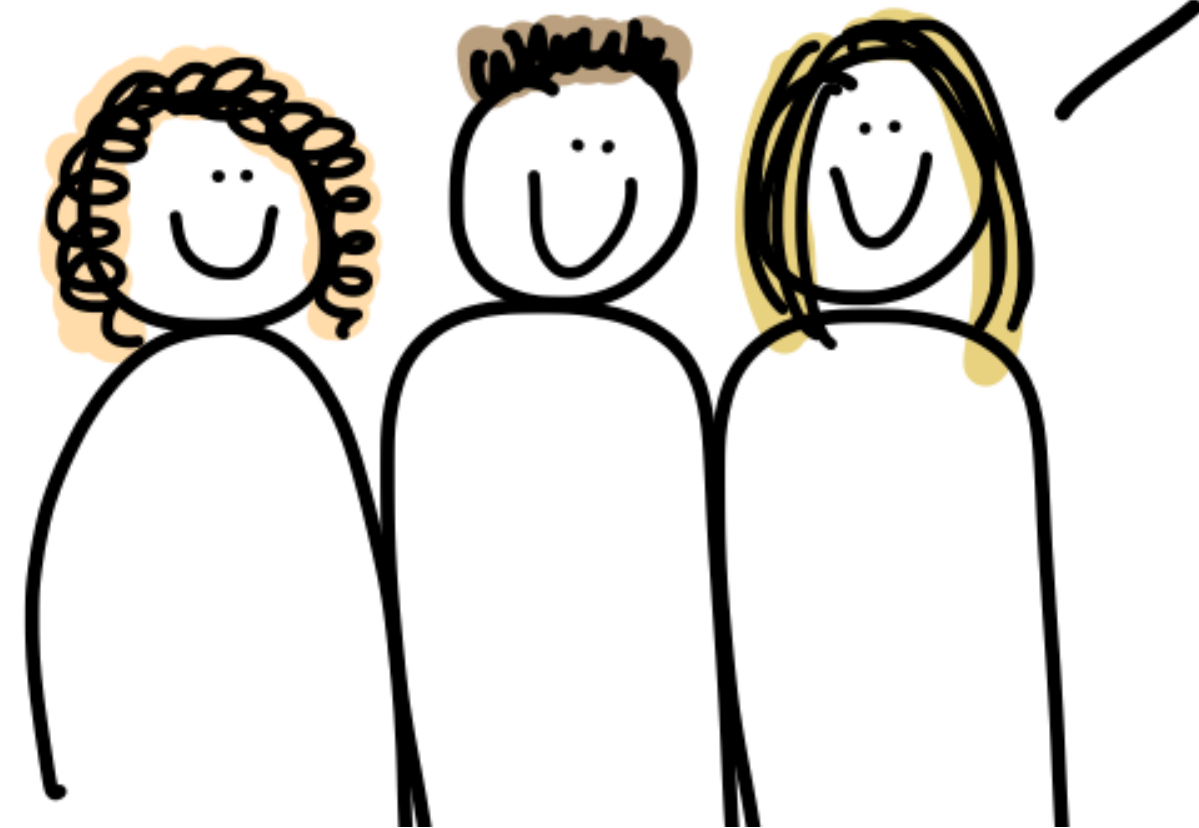
How to teach other things?



How to teach other things?



Phonics
Instruction / Explaining



Whole language
Inquiry / Exploration



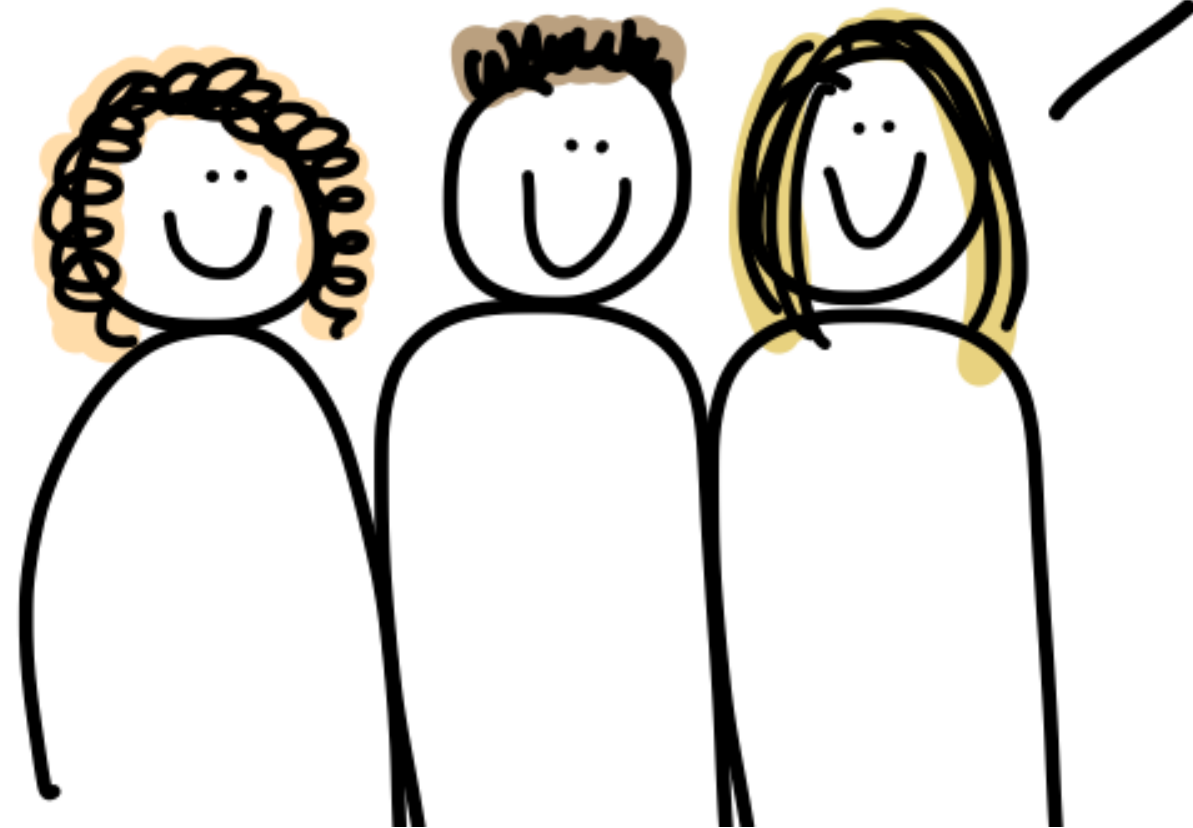
How to teach other things?



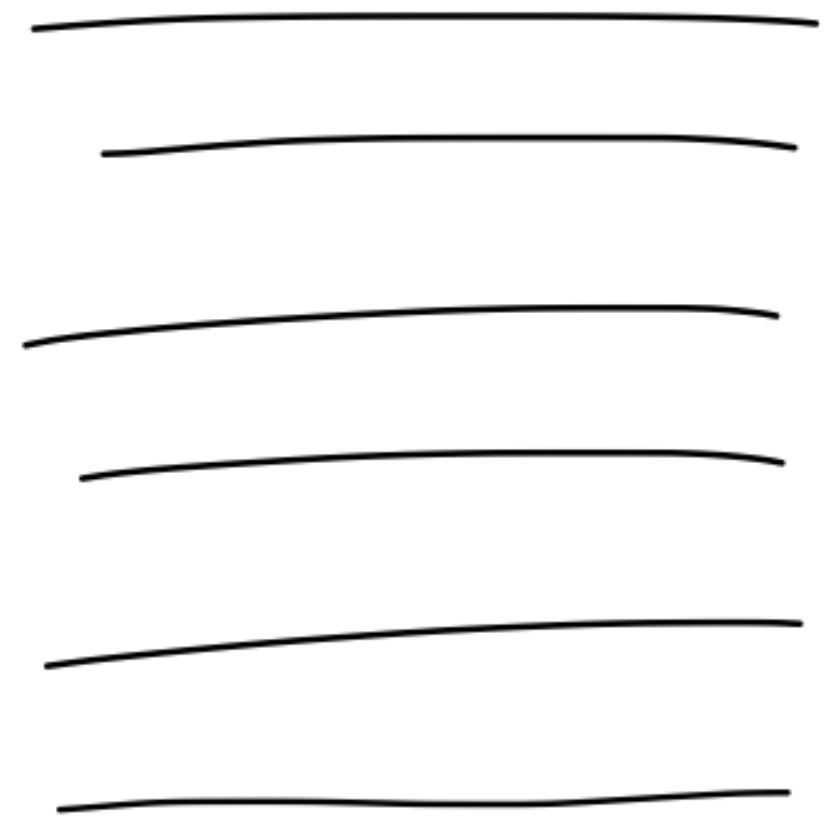
Phonics

Instruction

Explaining



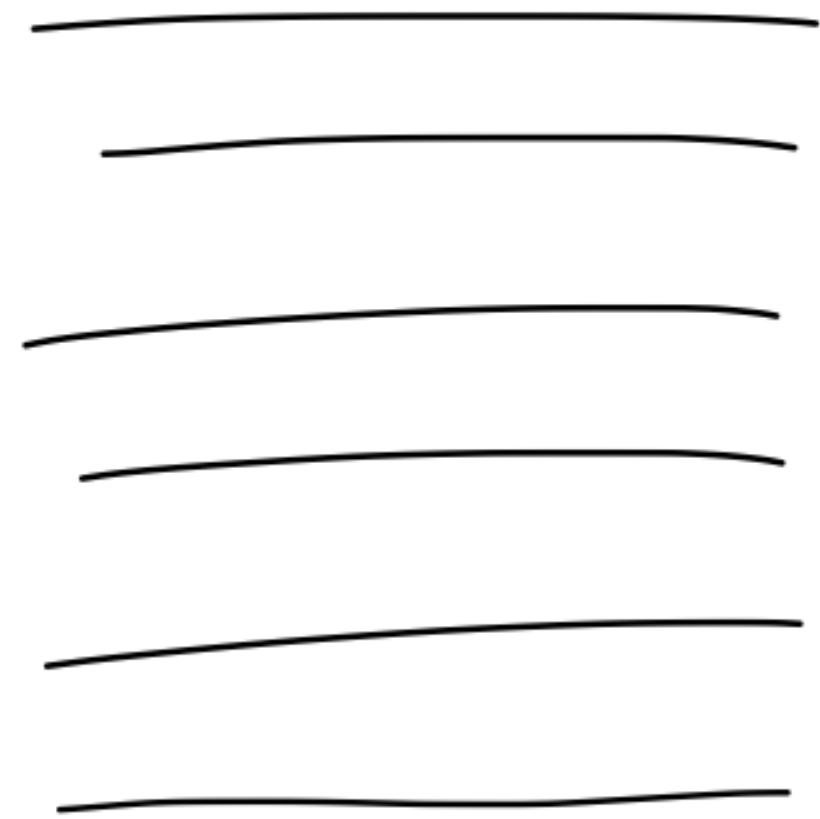
Why minimal guidance
does not work
Kirschner et al.



How to teach other things?

The use of worked
examples for Problem Solving

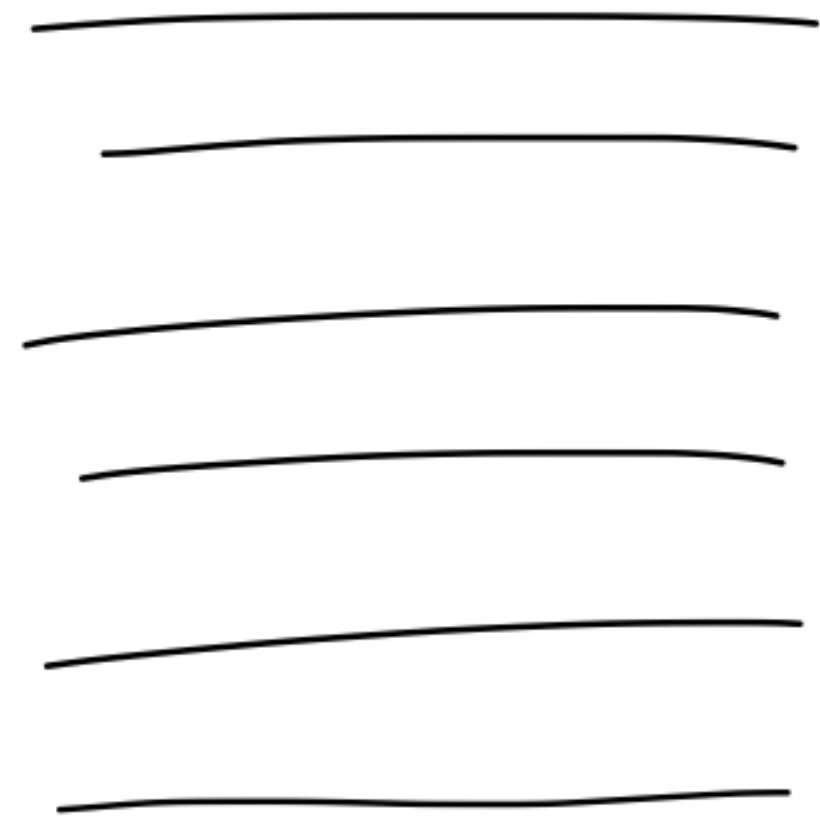
Sweller & Cooper





The use of worked
examples for Problem Solving

Sweller & Cooper



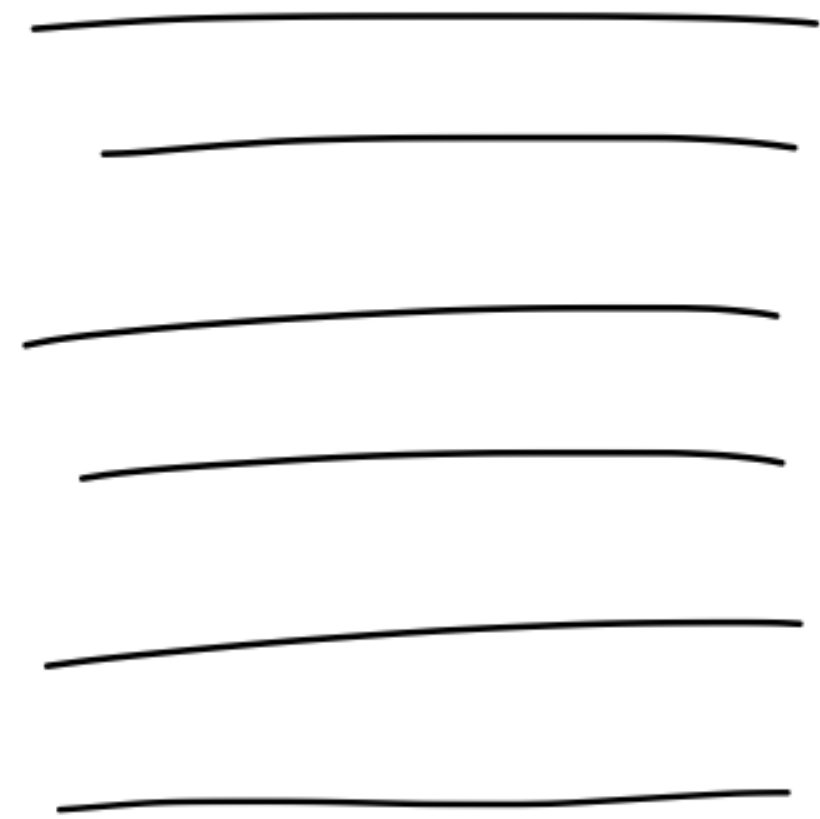


$a = 7 - 4a$
what is a ?



The use of worked
examples for Problem Solving

Sweller & Cooper



$$a = b - ca$$
$$a + ca = b$$
$$a(1+c) = b$$
$$a = \frac{b}{1+c}$$

$a = 7 - 4a$
what is a ?

The use of worked examples for Problem Solving

Sweller & Cooper



$$\begin{aligned} a &= b - ca \\ a + ca &= b \\ a(1+c) &= b \\ a &= \frac{b}{1+c} \end{aligned}$$

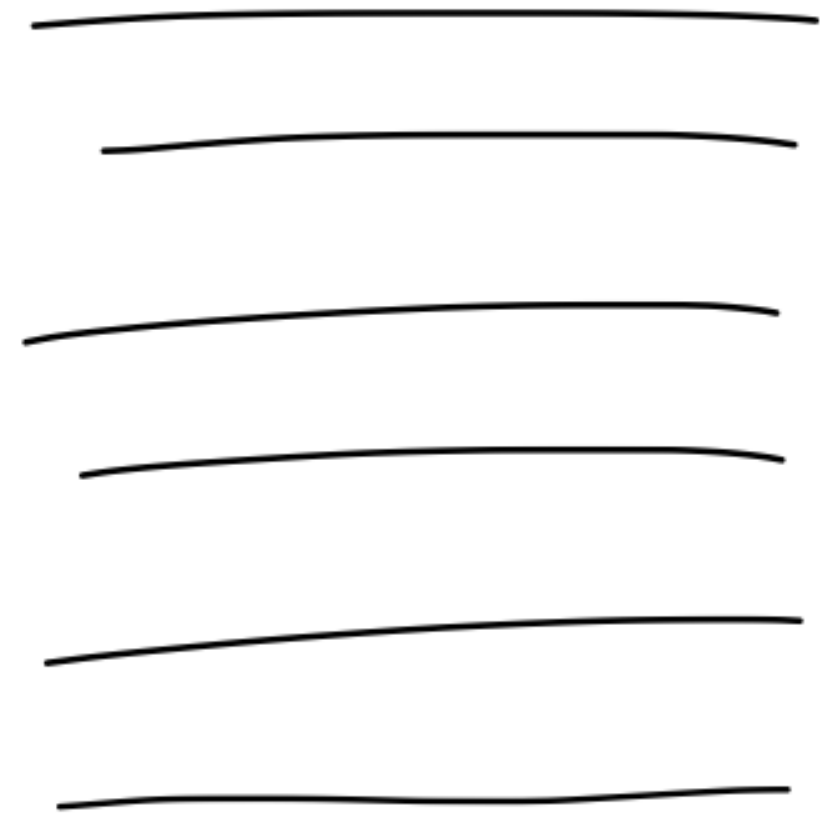
$a = 7 - 4a$
what is a ?

1

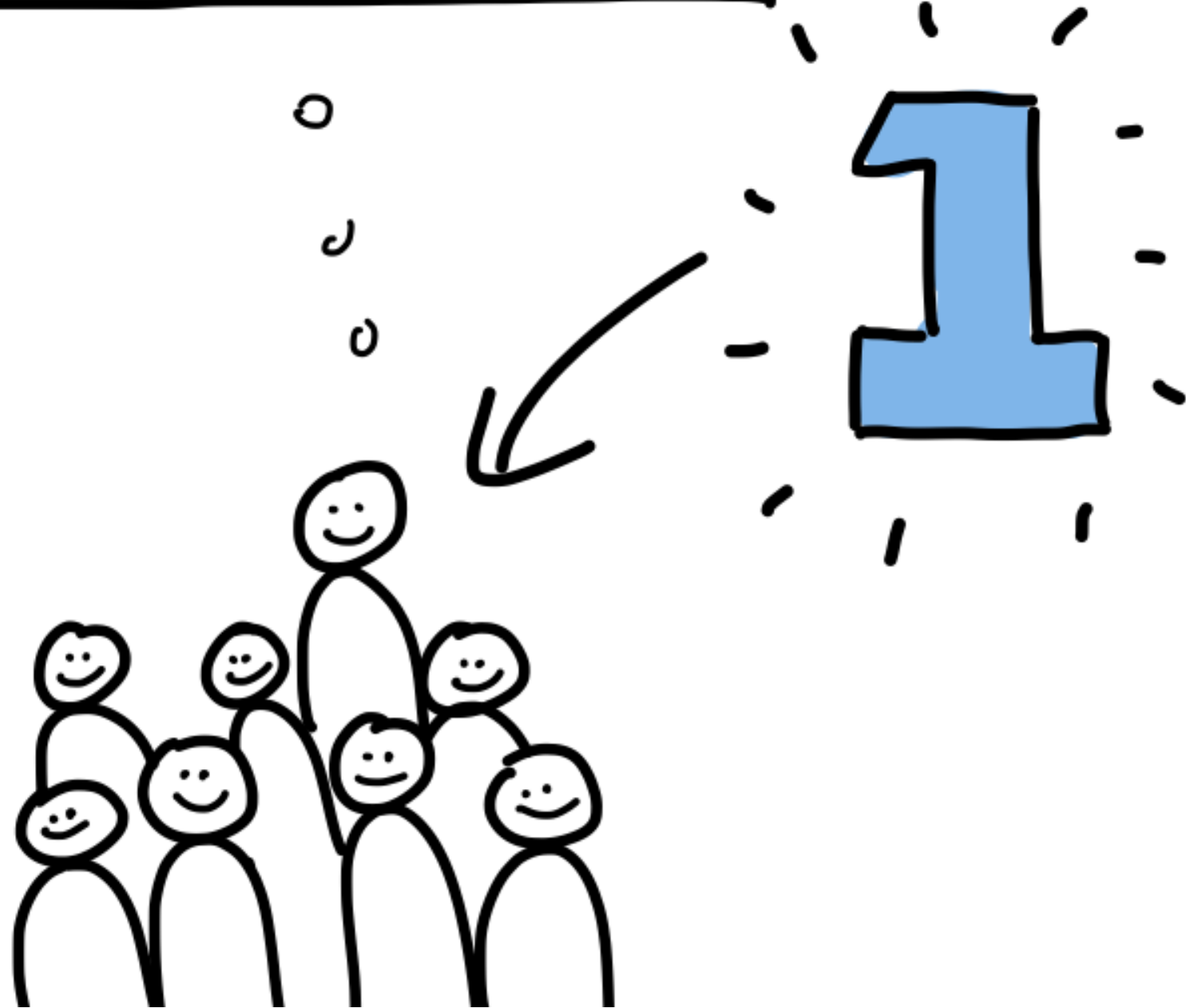


The use of worked
examples for Problem Solving

Sweller & Cooper



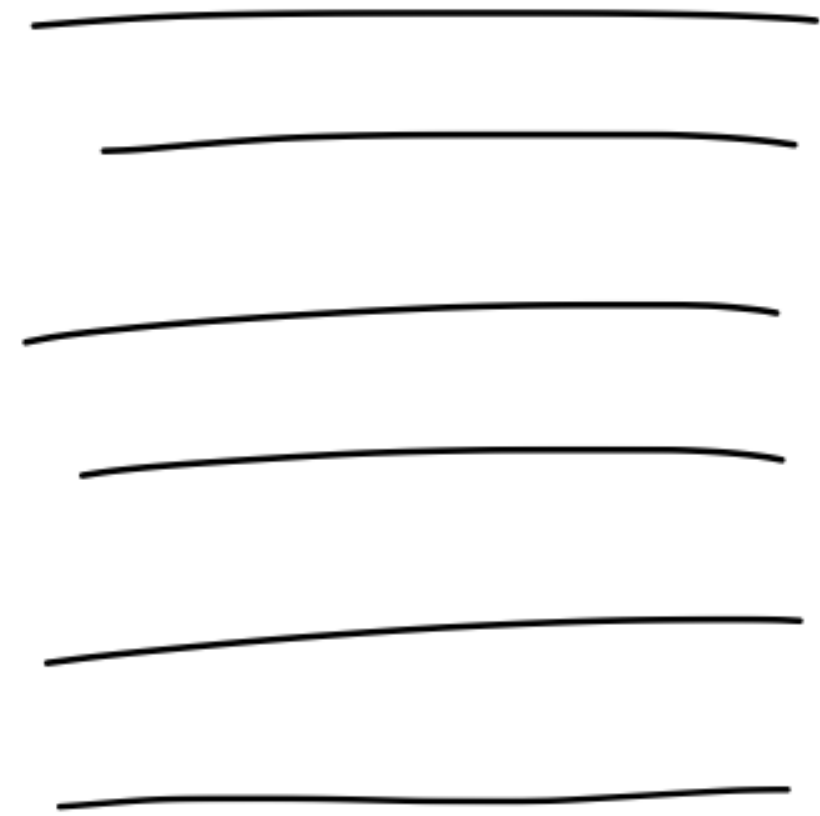
$$\begin{aligned} a &= b - ca \\ a + ca &= b \\ a(1+c) &= b \\ a &= \frac{b}{1+c} \end{aligned}$$



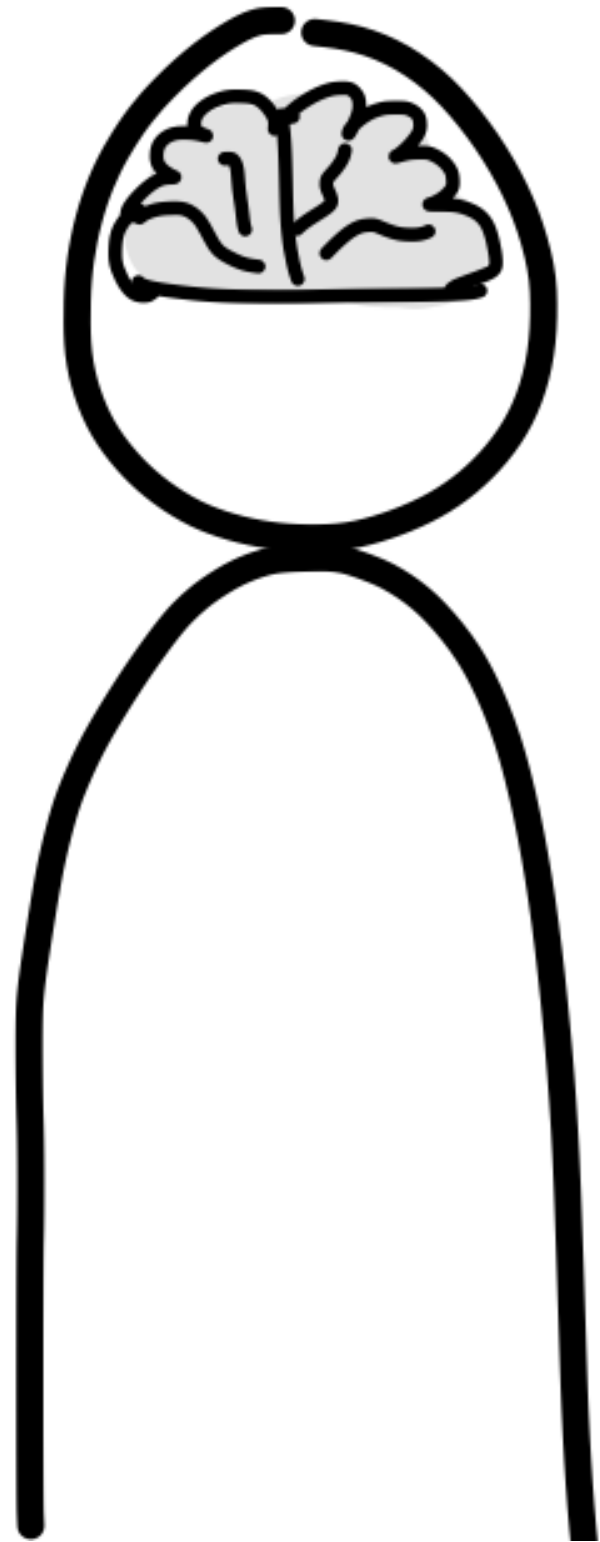
Also on
different problems

The use of worked
examples for Problem Solving

Sweller & Cooper

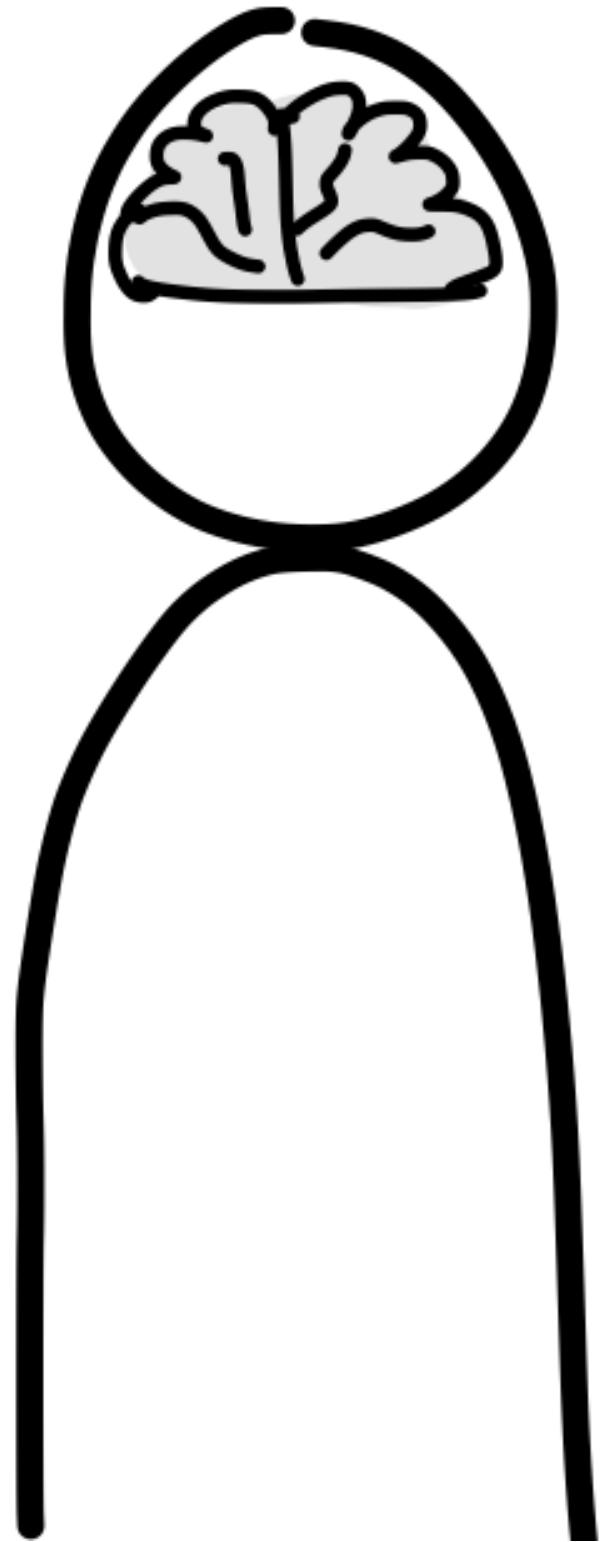


Short term memory

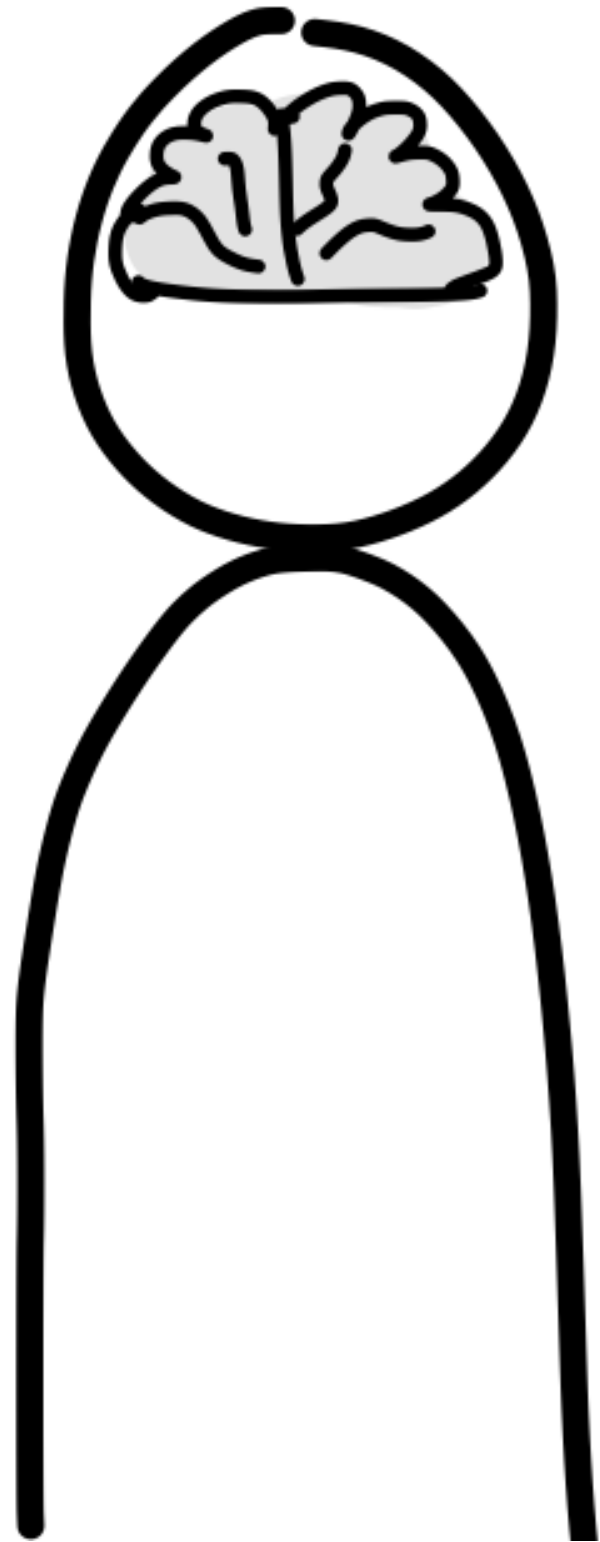


Short term memory

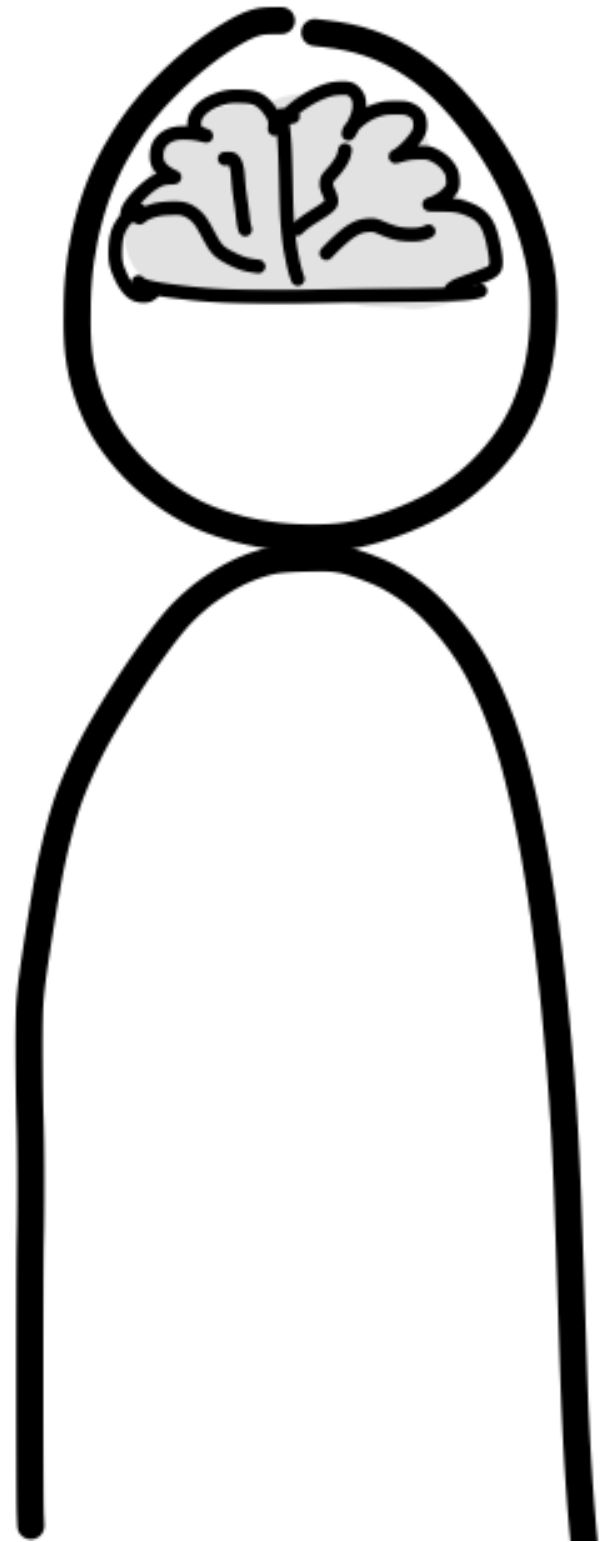
○ | 3 ◇ \ C < ◇ ↓ | 3)



Short term memory

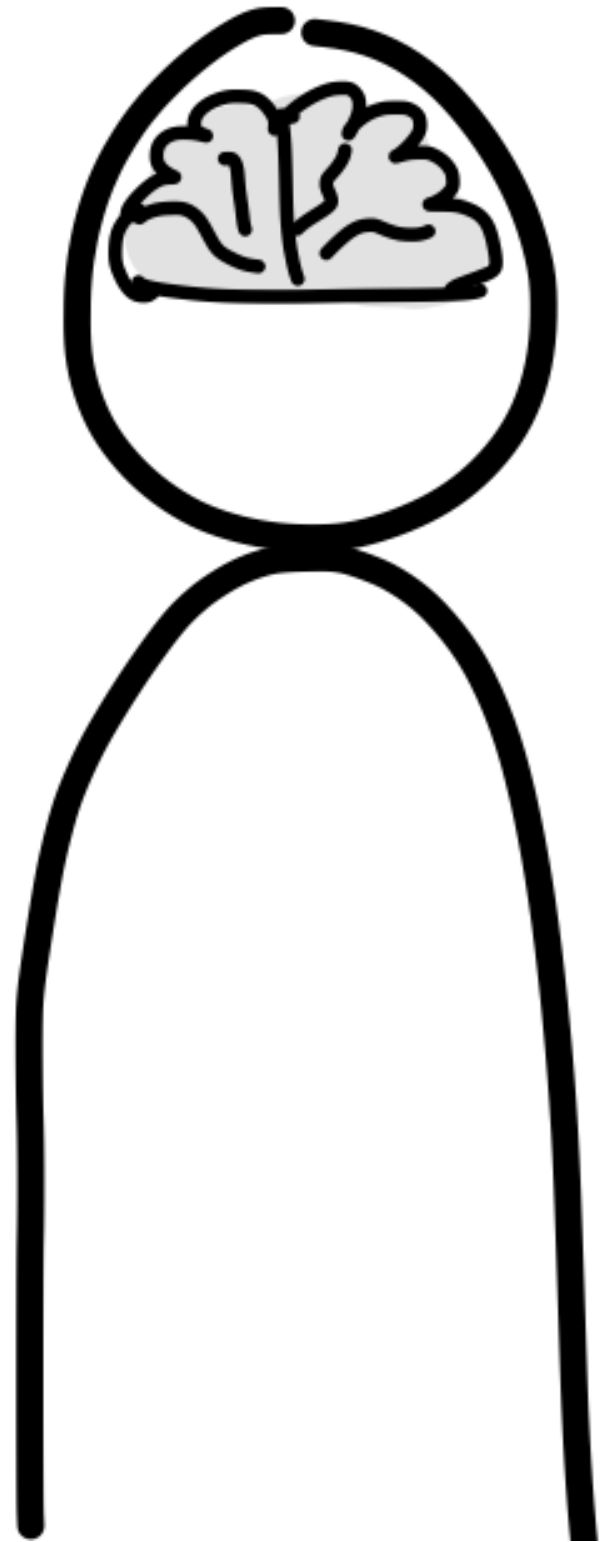


Short term memory

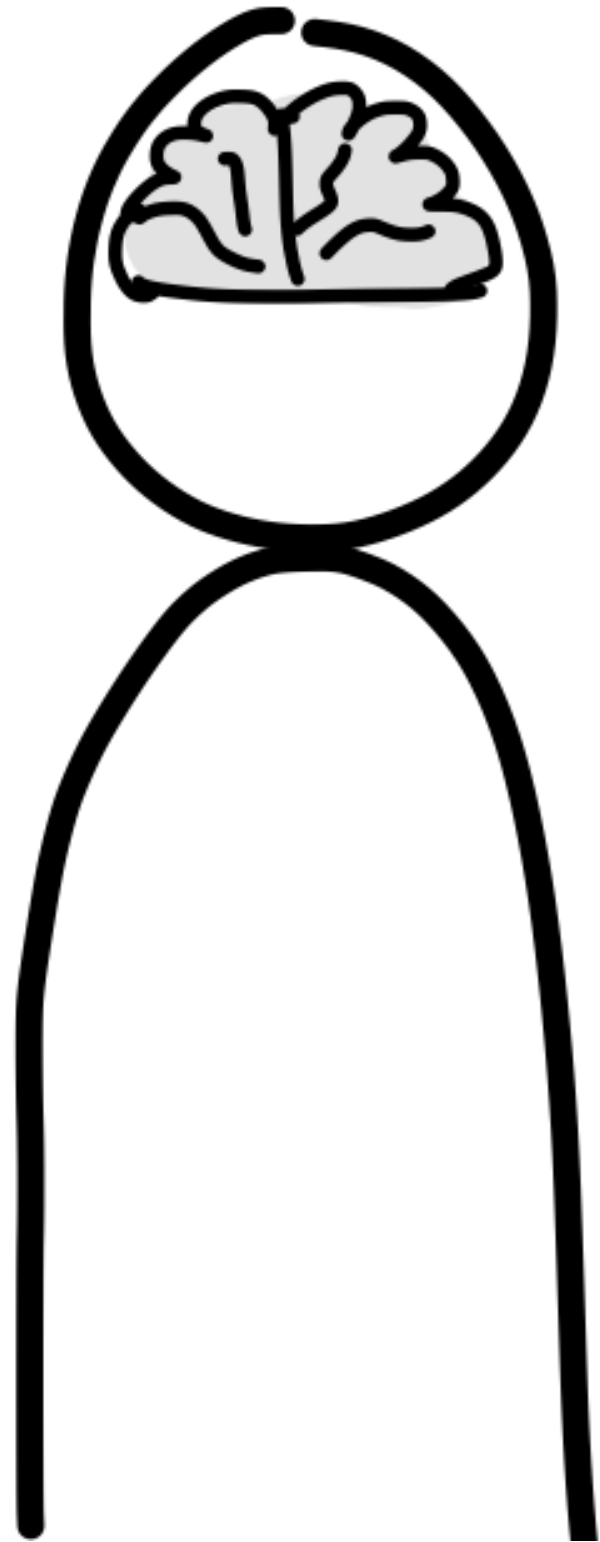


abk mrtpi gbar

Short term memory

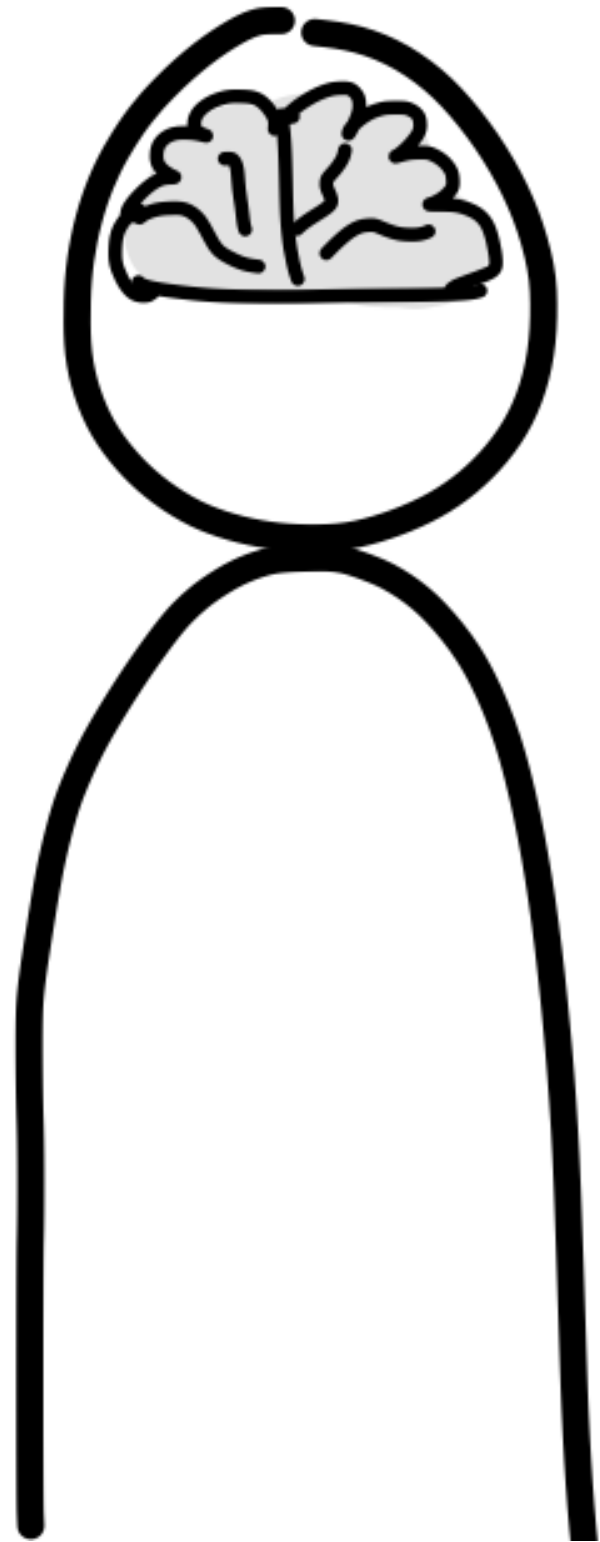


Short term memory

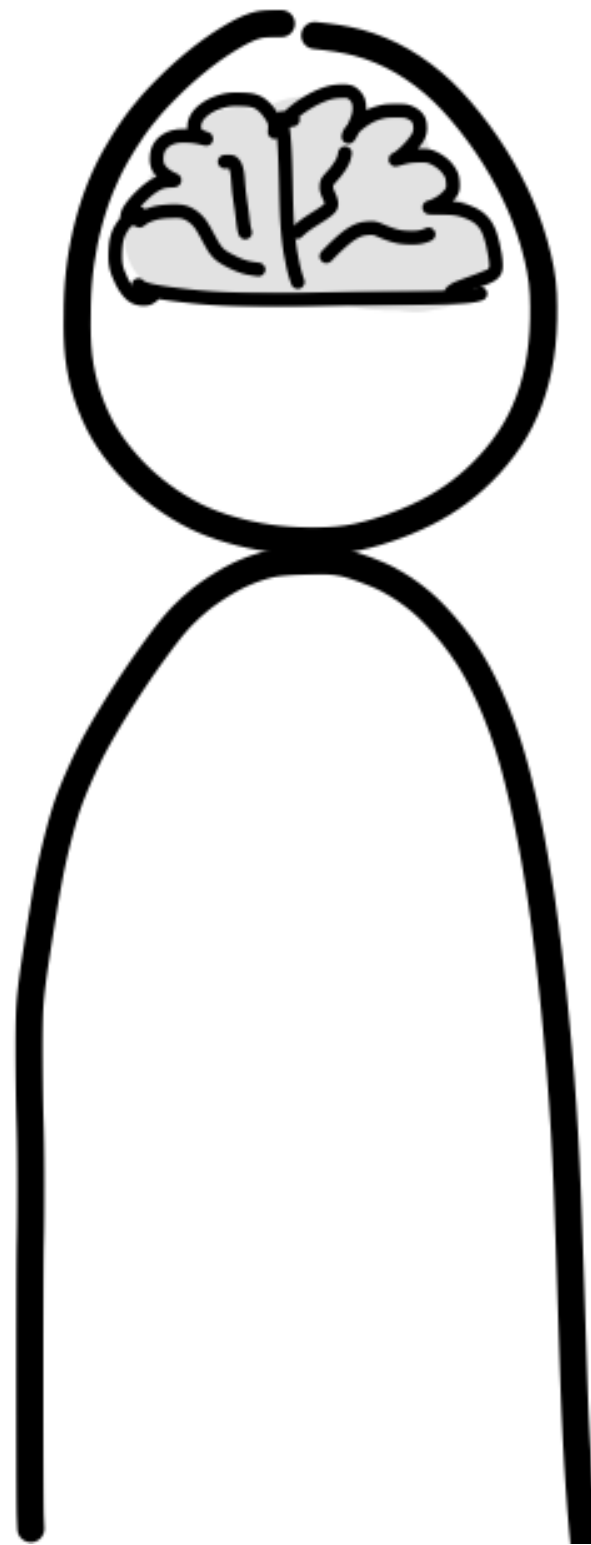


cat loves cake

Short term memory

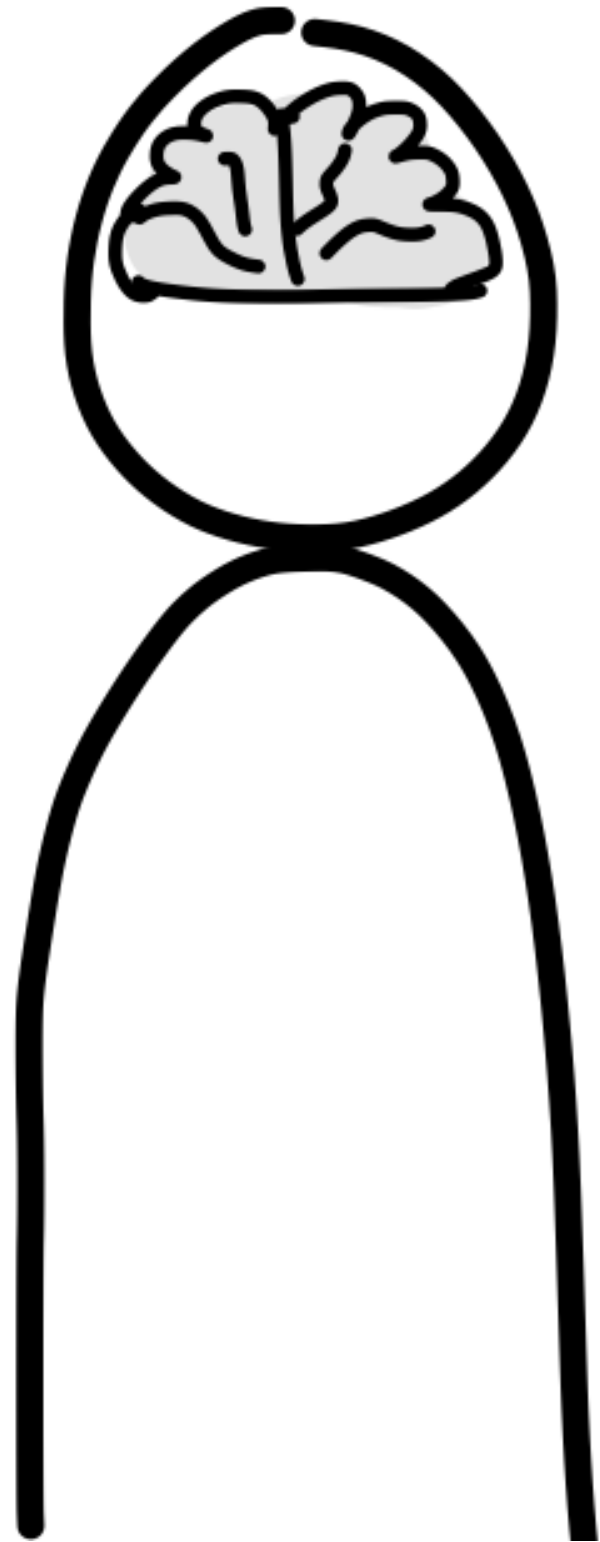


Short term memory



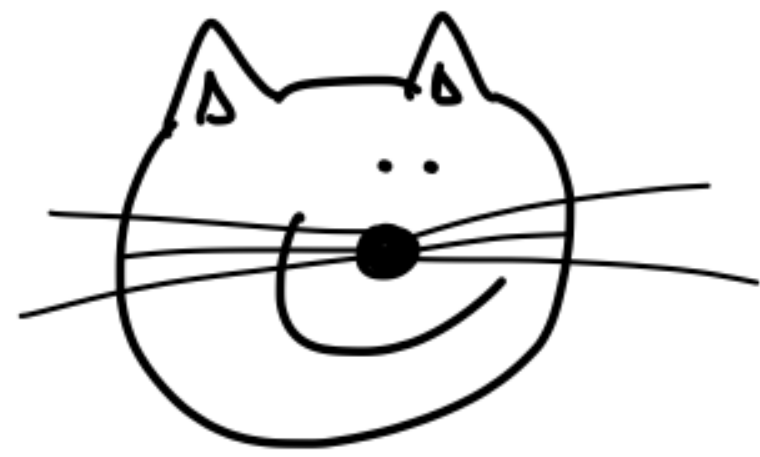
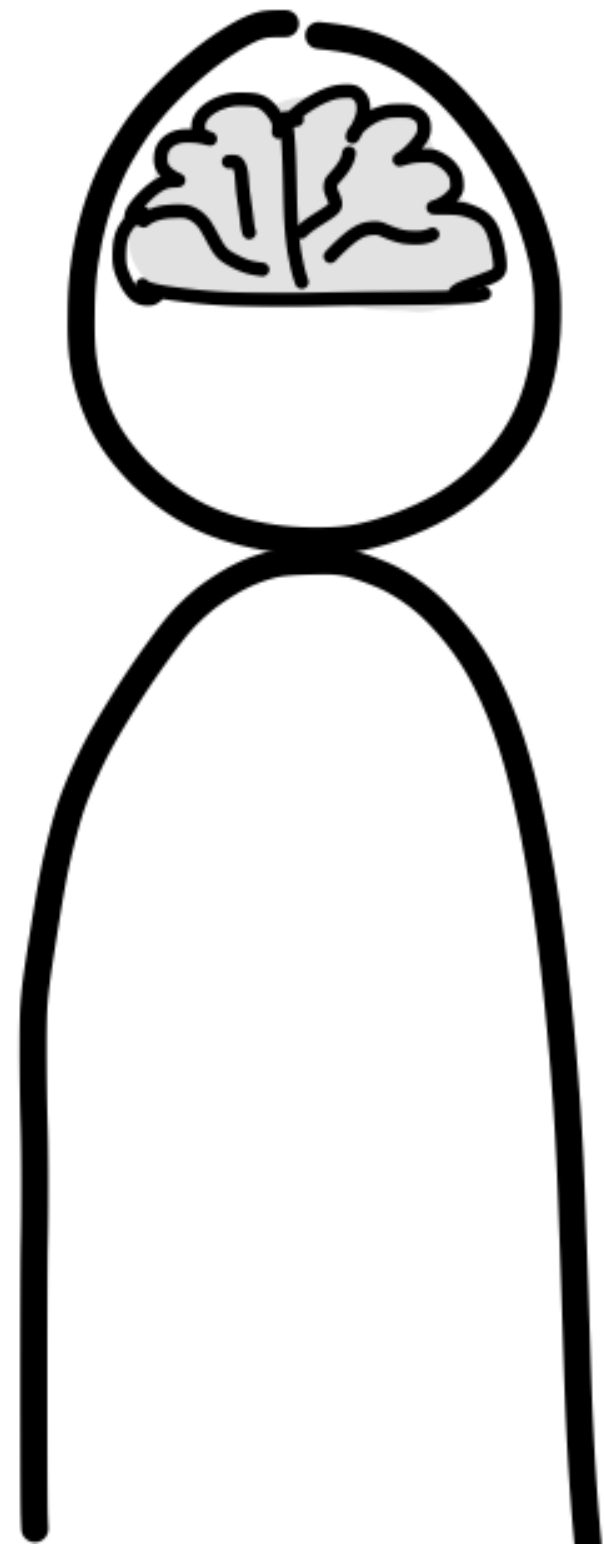
— Can only hold 7 ± 2 items

Short term memory



— Can only hold ⁴~~7~~ ± 2 items

Short term memory



cat loves cake



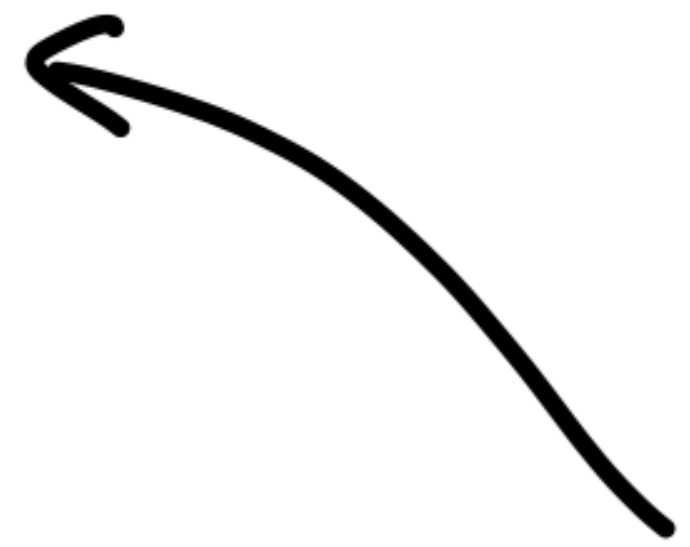
$\odot \mid \approx$

$\diamond \setminus C < \diamond$

$\downarrow \mid \approx \cup$



$\odot \mid \exists$ $\diamond \setminus C < \diamond$ $\downarrow \mid \exists \cup$



Cognitive load

How to teach other things?

1



$a = aq + b$
 $a - aq = b$
...

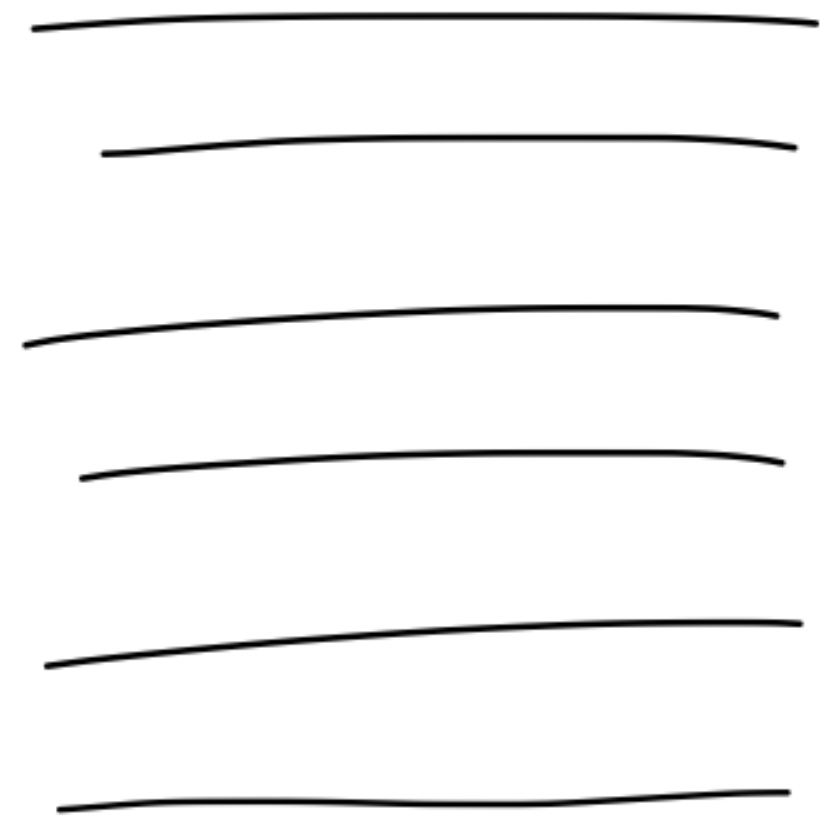


$a = 4a + 7$
solve for a



The use of worked examples for Problem Solving

Sweller & Cooper



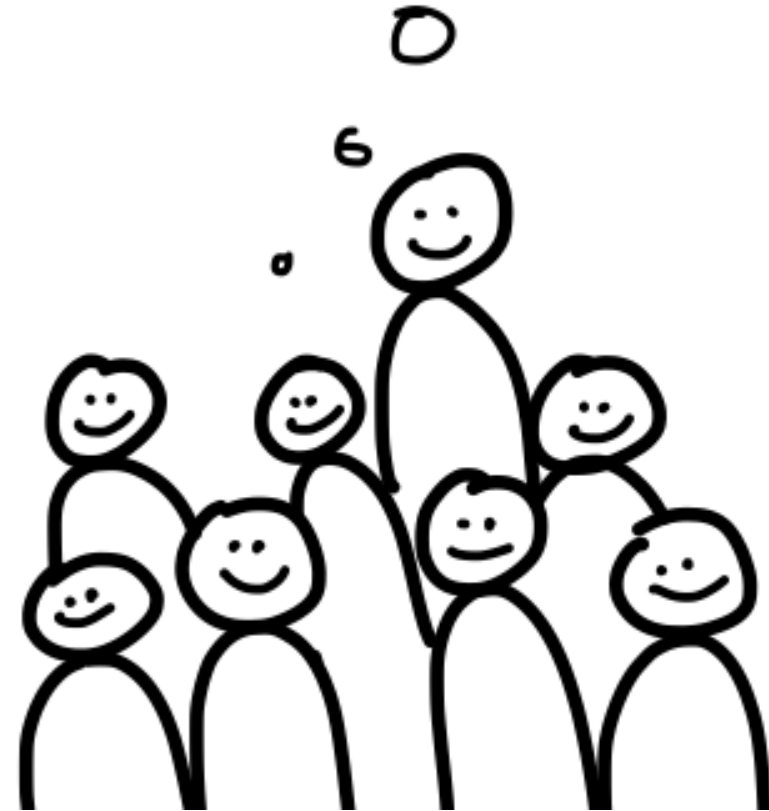
How to teach other things?

Cognitive load
too high

$$\begin{aligned} a &= aq + b \\ a - aq &= b \\ \dots \end{aligned}$$

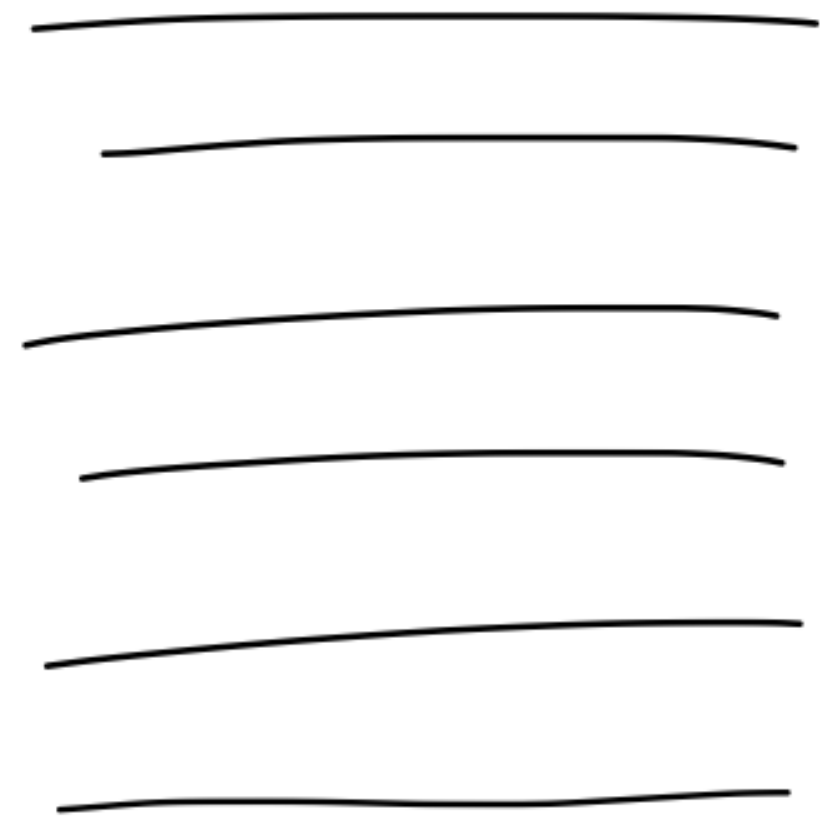


$$\begin{aligned} a &= 4a + 7 \\ \text{solve for } a \end{aligned}$$



The use of worked
examples for Problem Solving

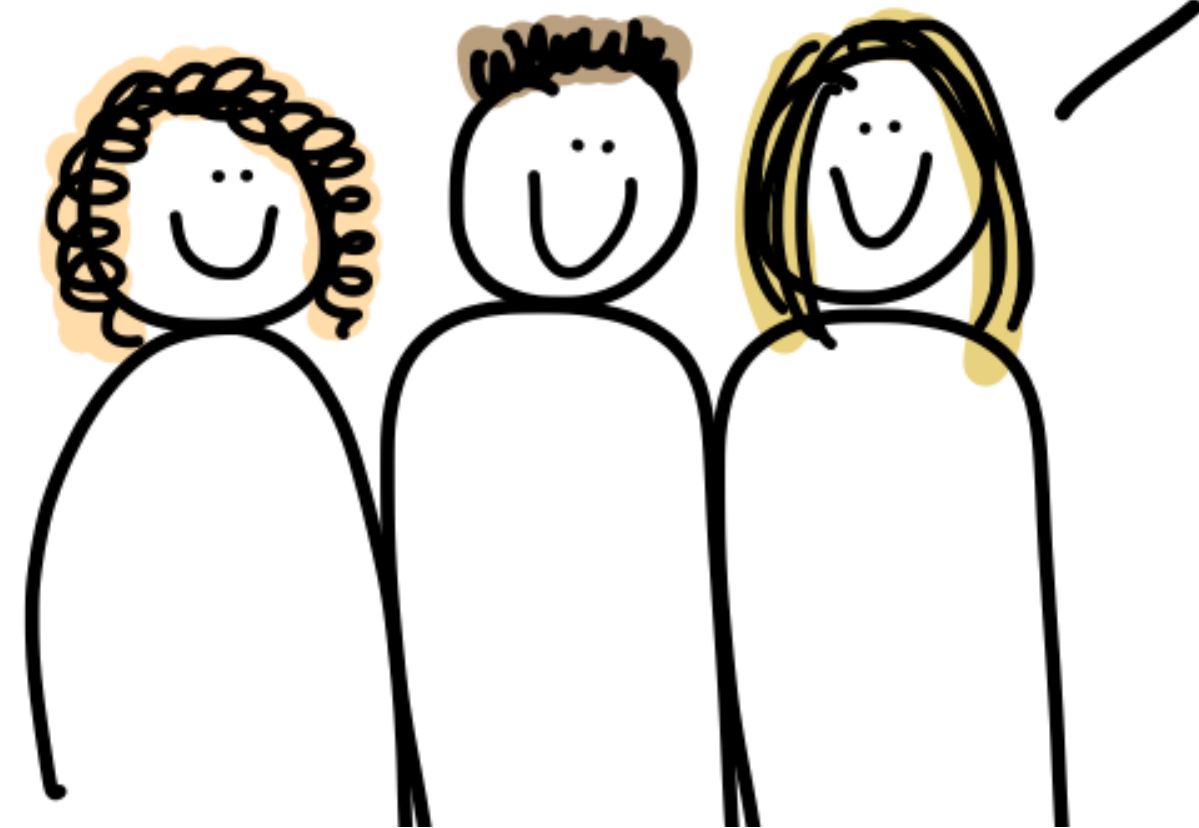
Sweller & Cooper



How to teach other things?



Phonics
Instruction / Explaining



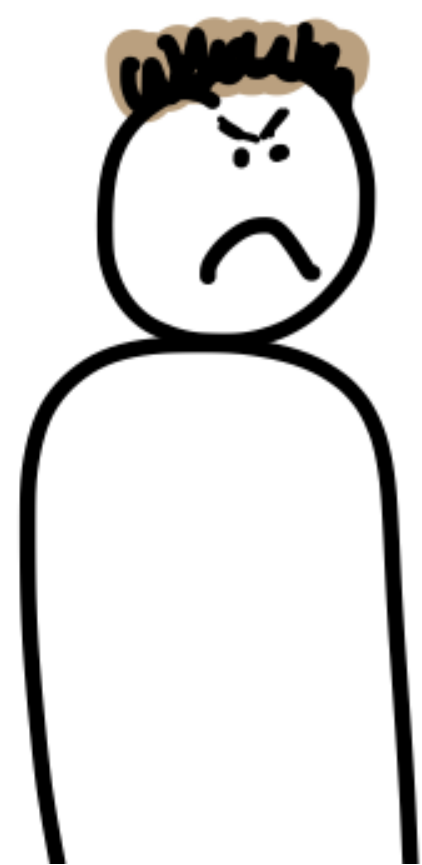
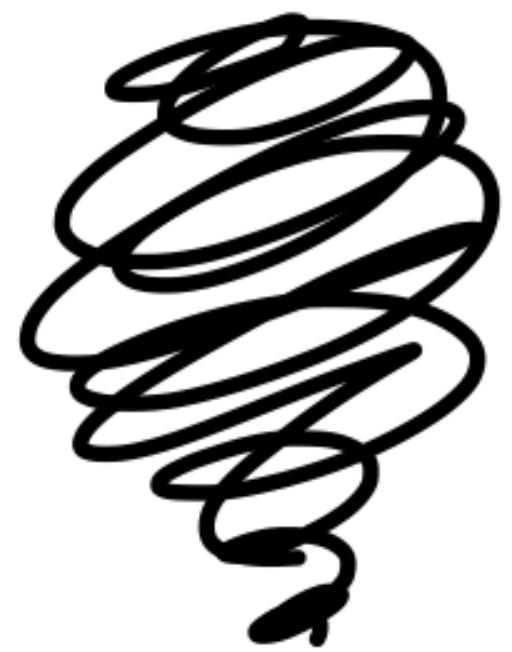
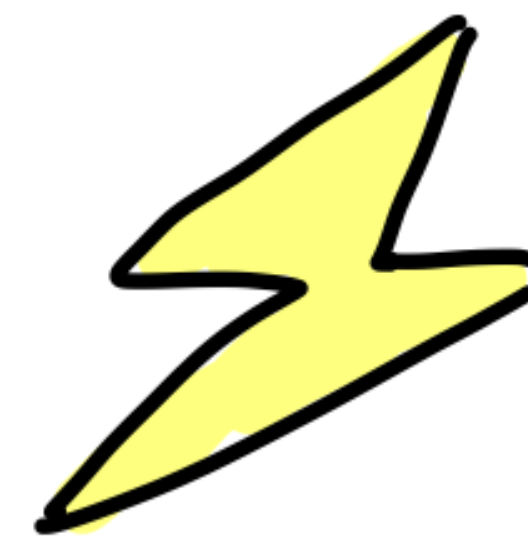
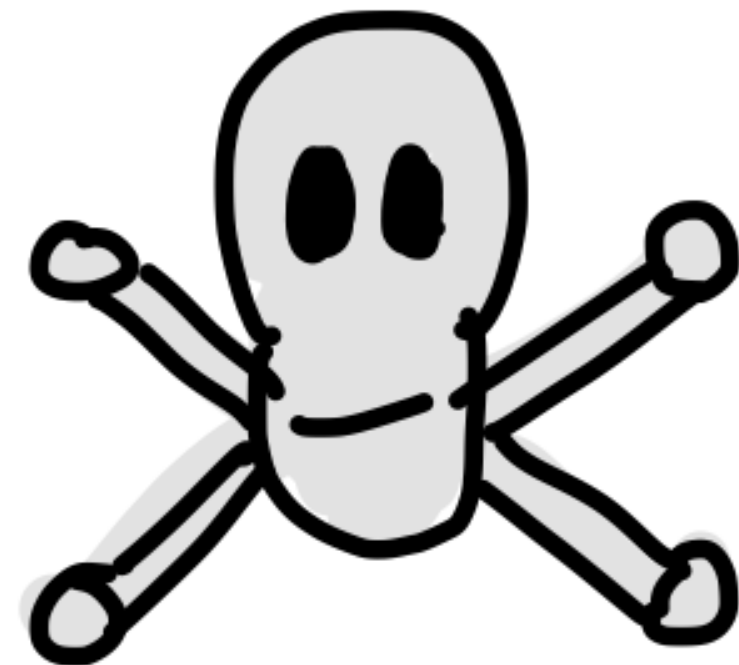
Whole language
Inquiry / Exploration



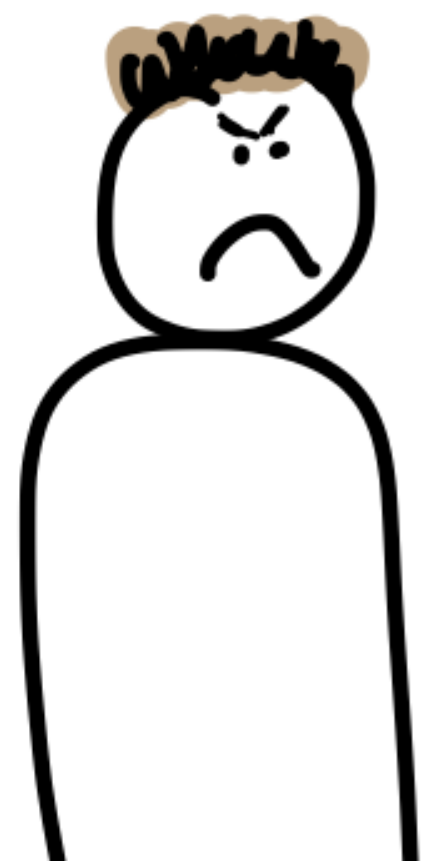
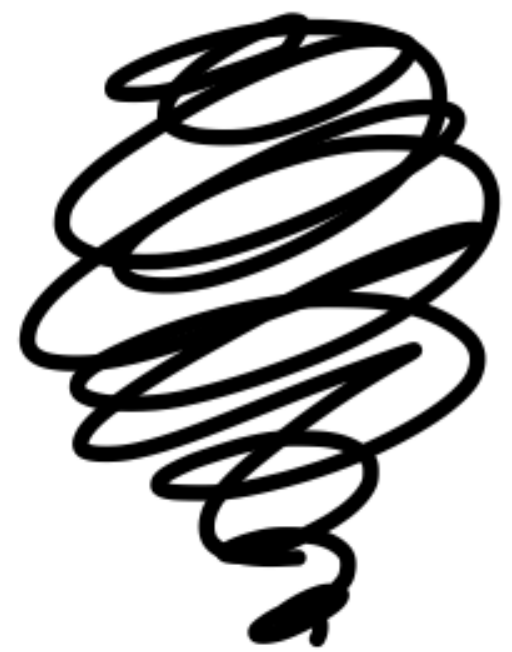
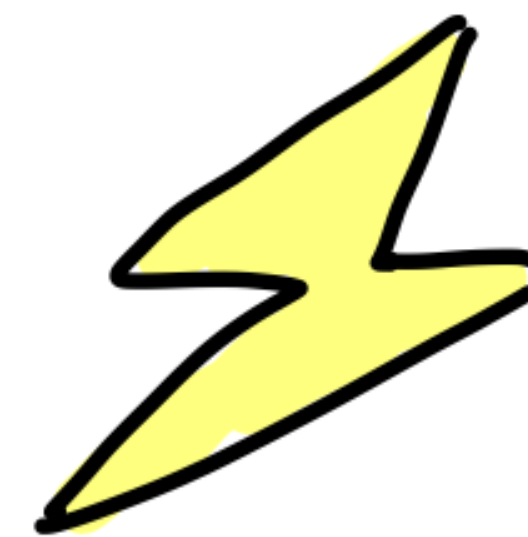
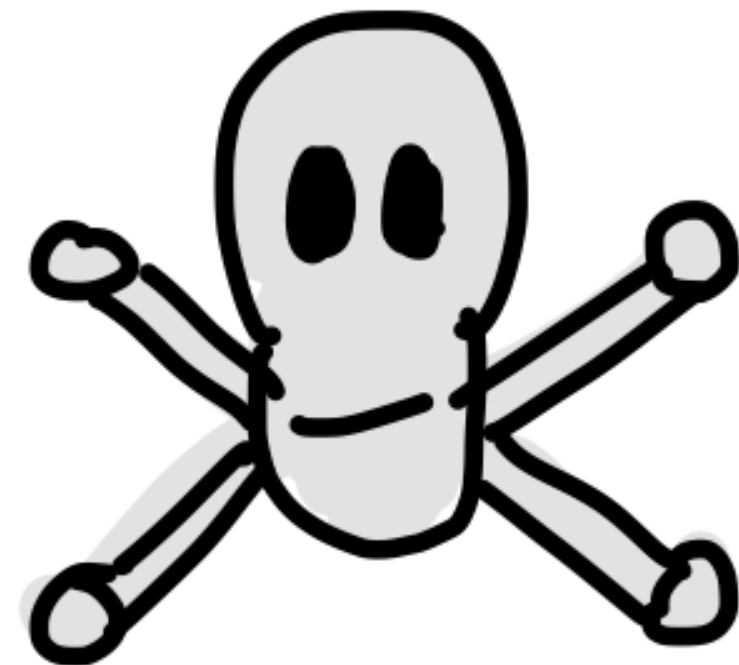
You don't become
an expert by
doing expert things



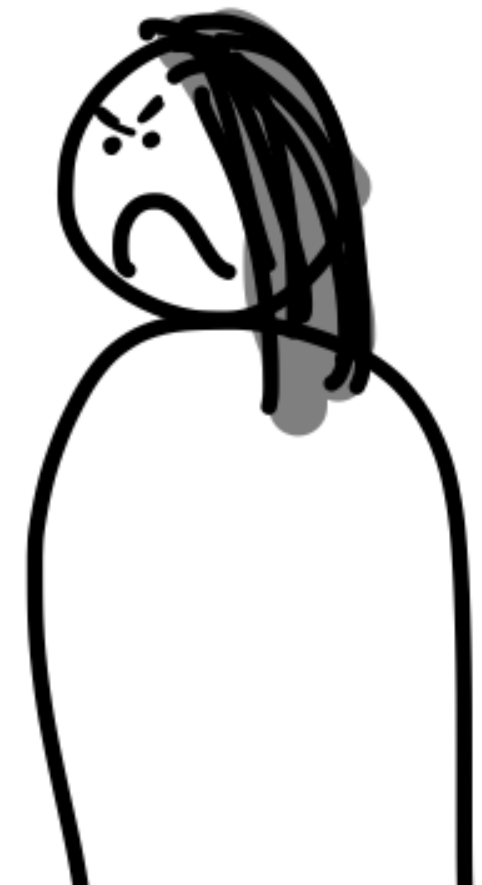
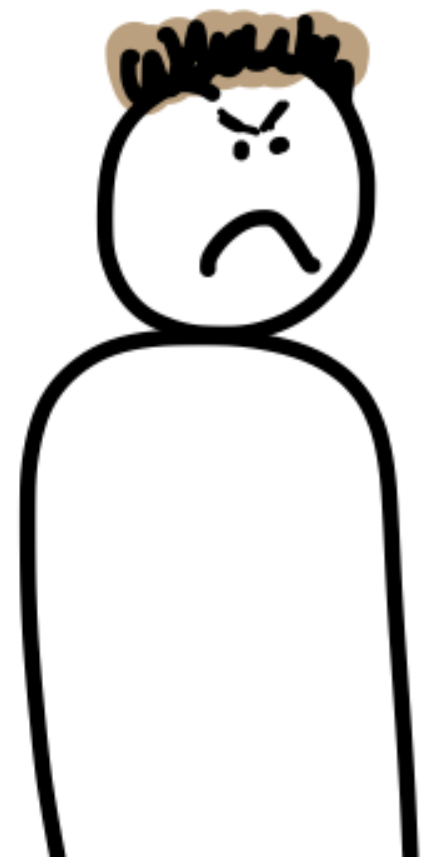
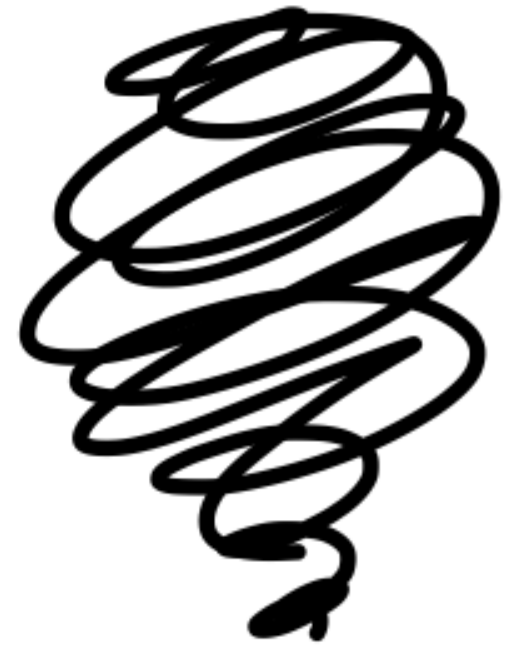
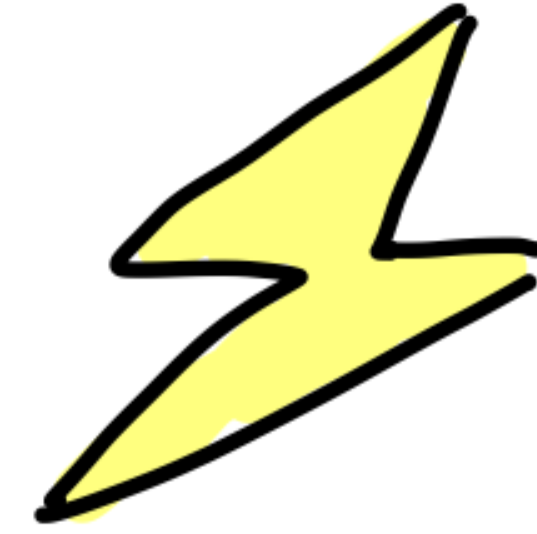
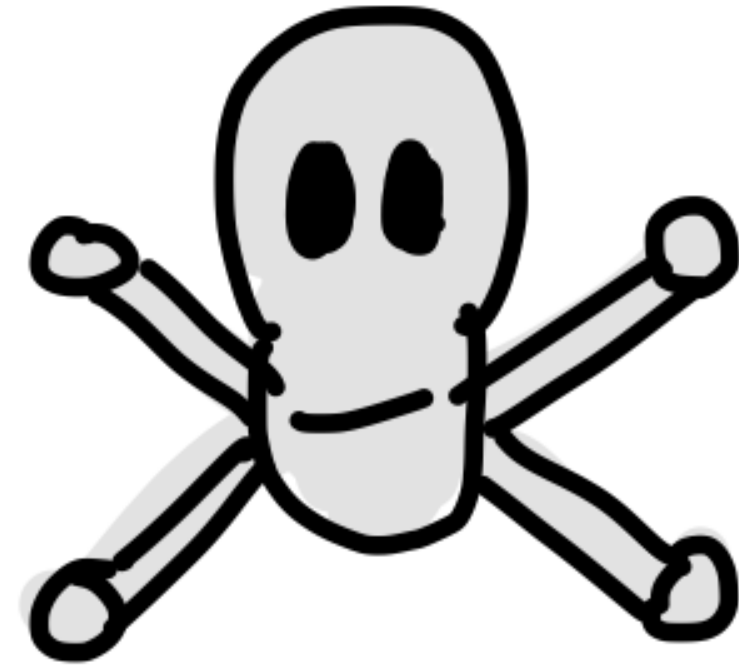
Let's start a fight



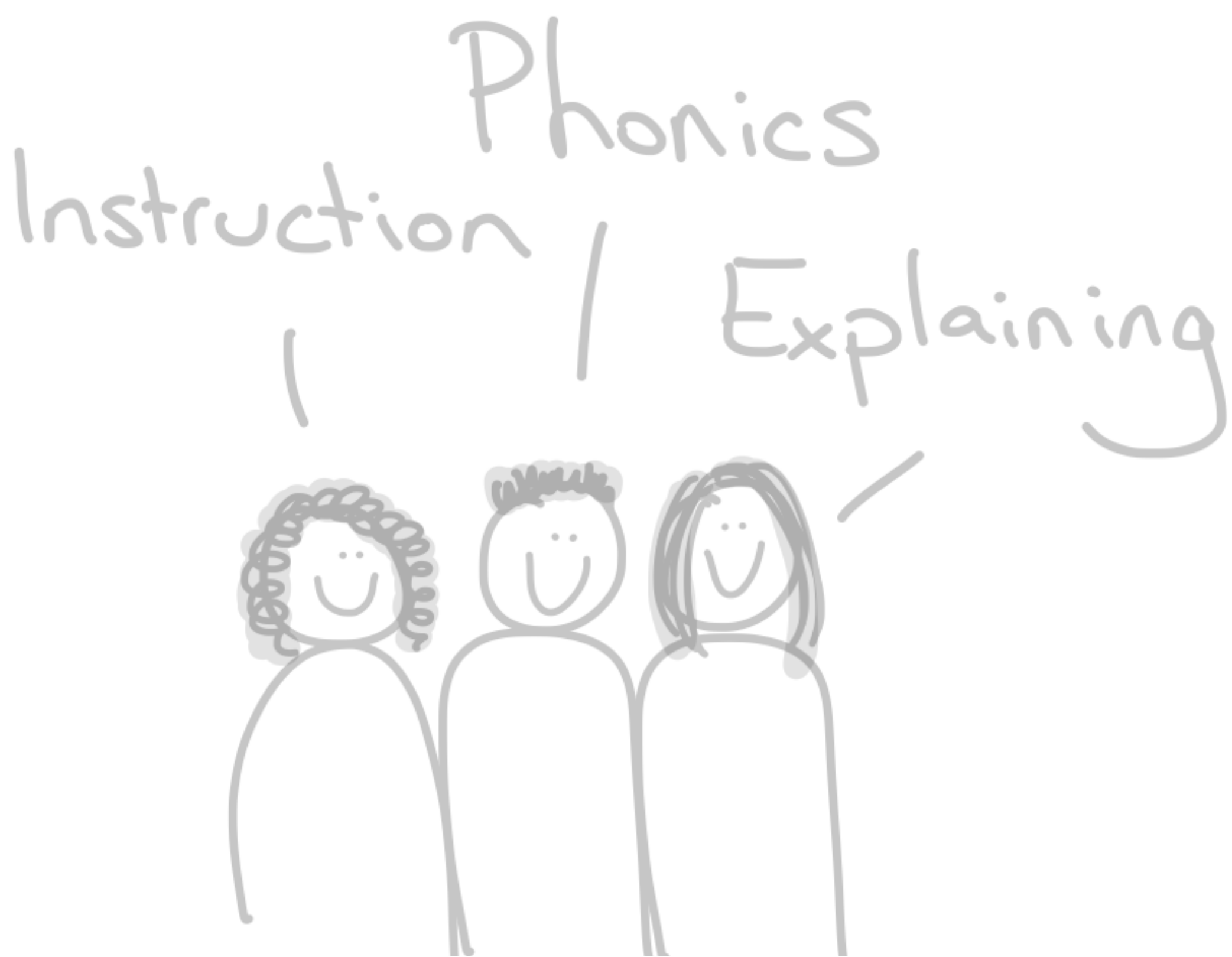
Let's start a ~~flight~~



Let's start a pedagogical debate



How to teach programming?



How to teach programming?

What would direct instruction
for programming look like?



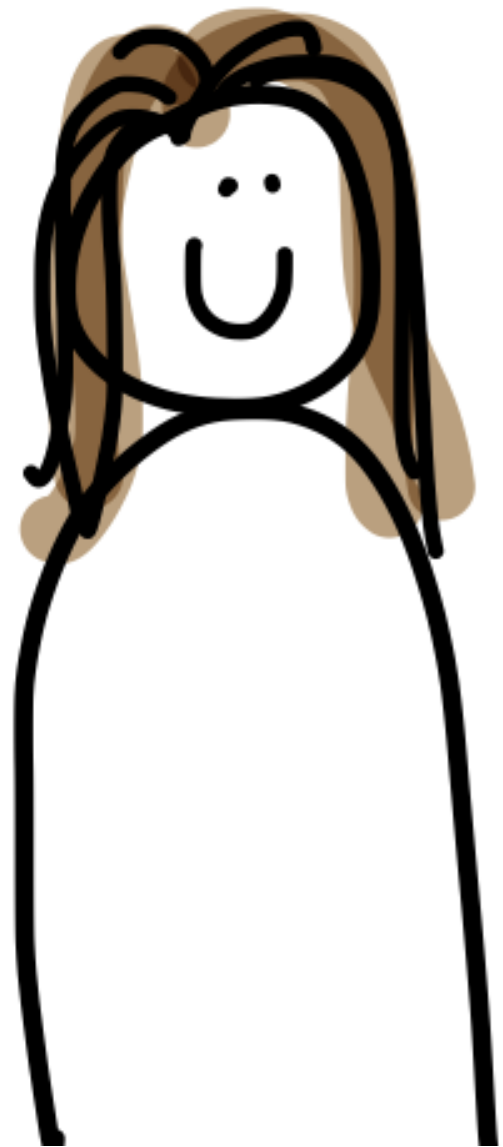
How to teach programming?

Vocalise syntax



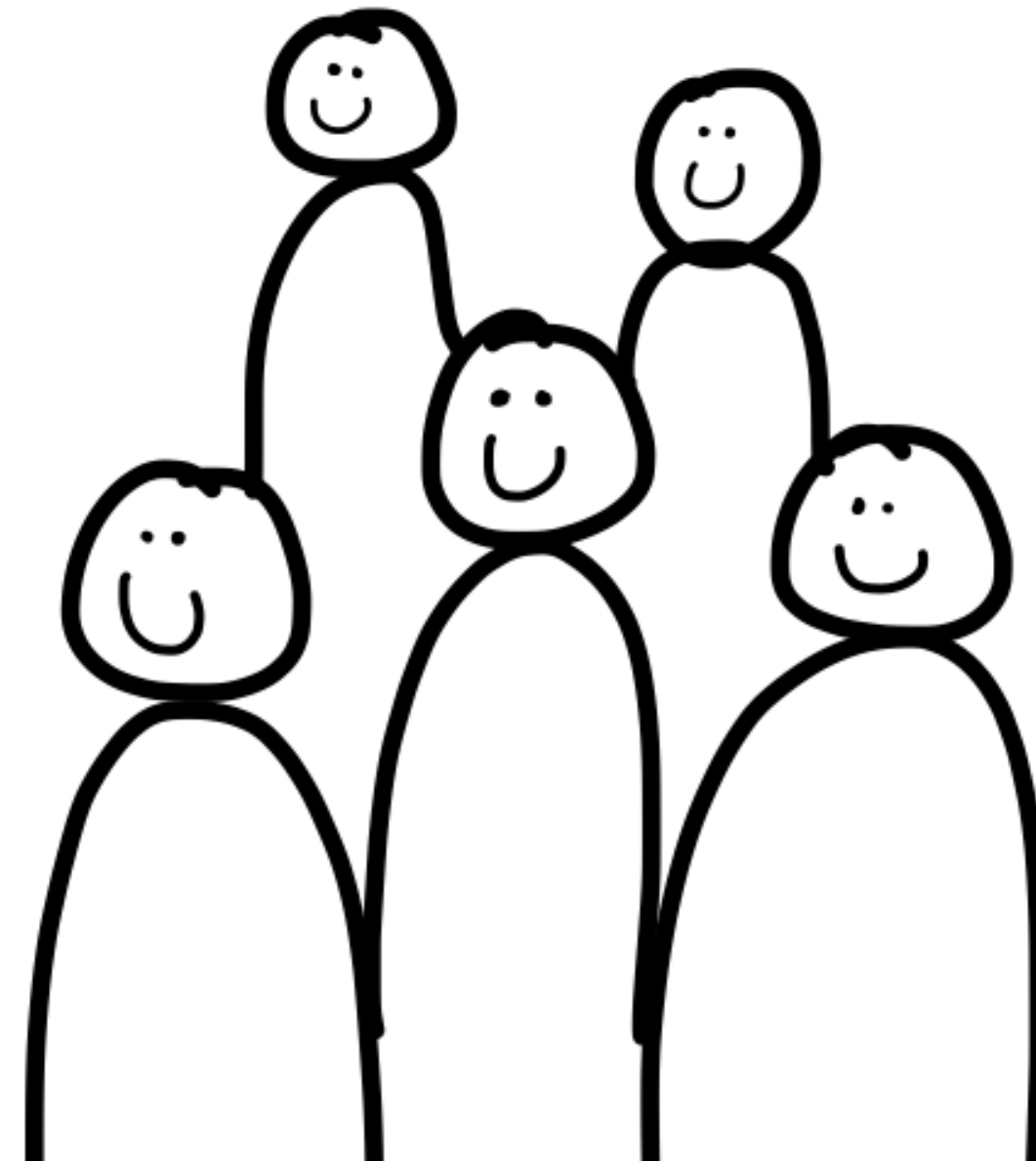
Sound matters

I have a tear
... in my pants



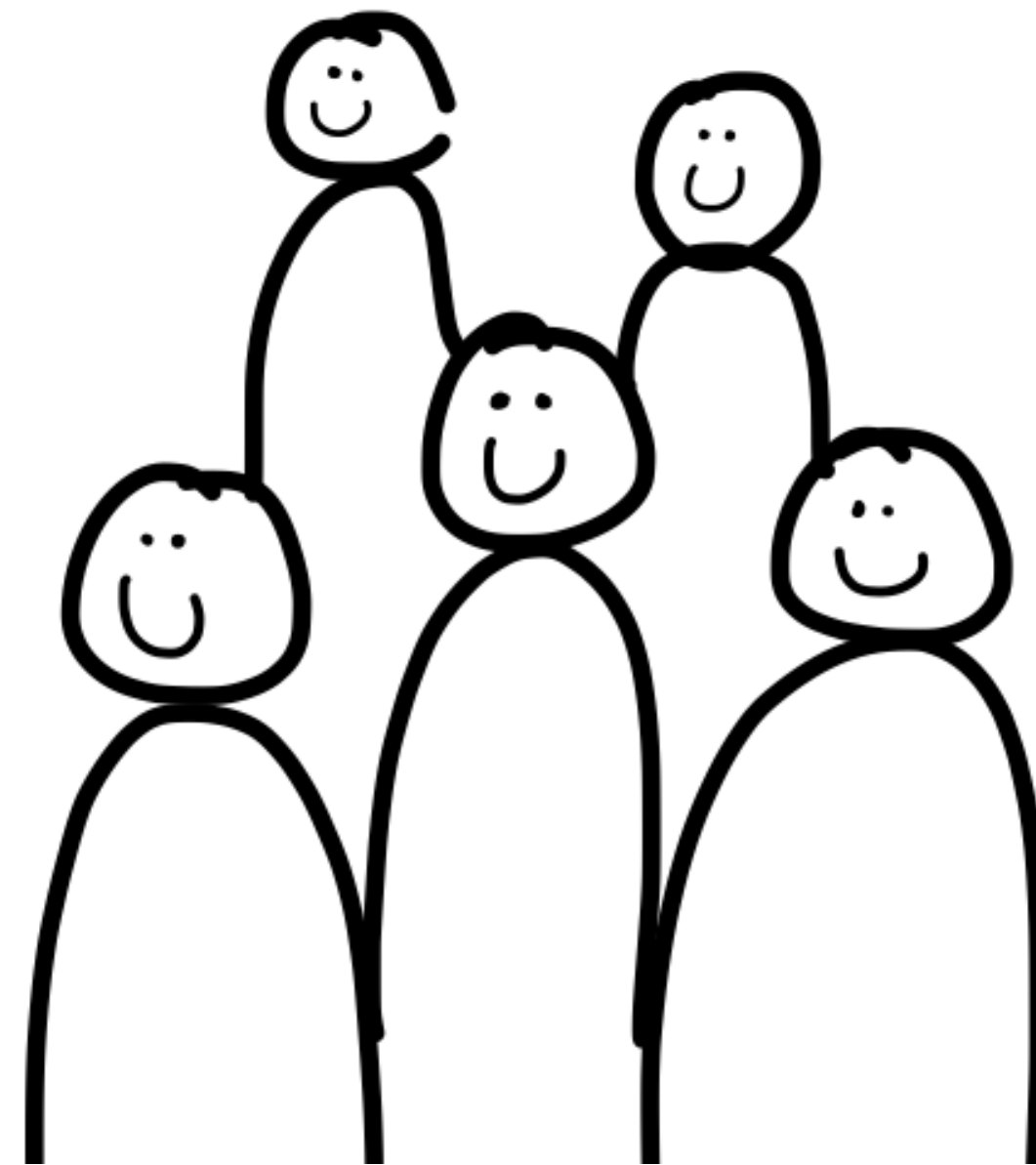
How does code sound?

we asked 10 novices



How does code sound?

$x = 5$

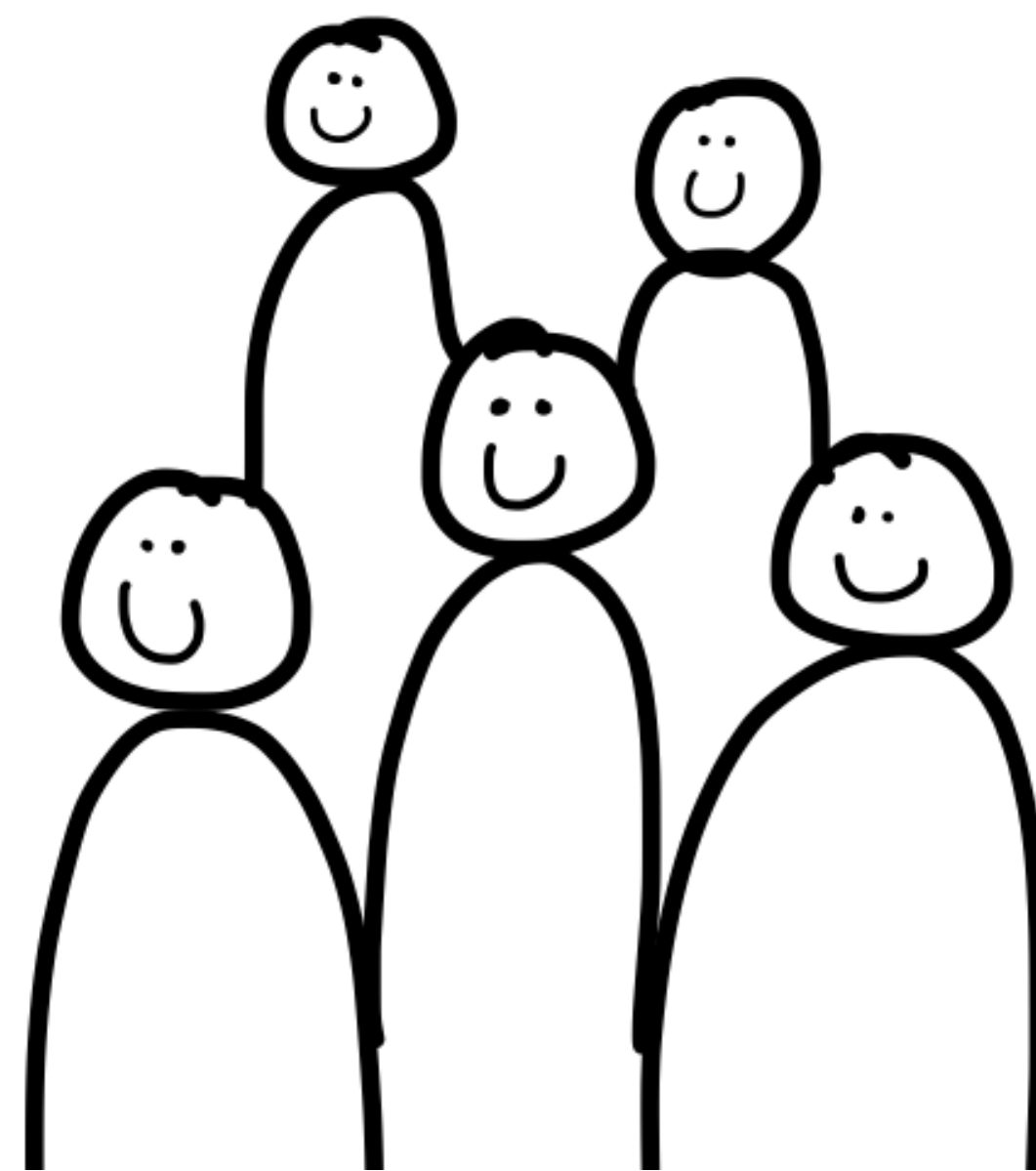


How does code sound?

$x = 5$



x is 5

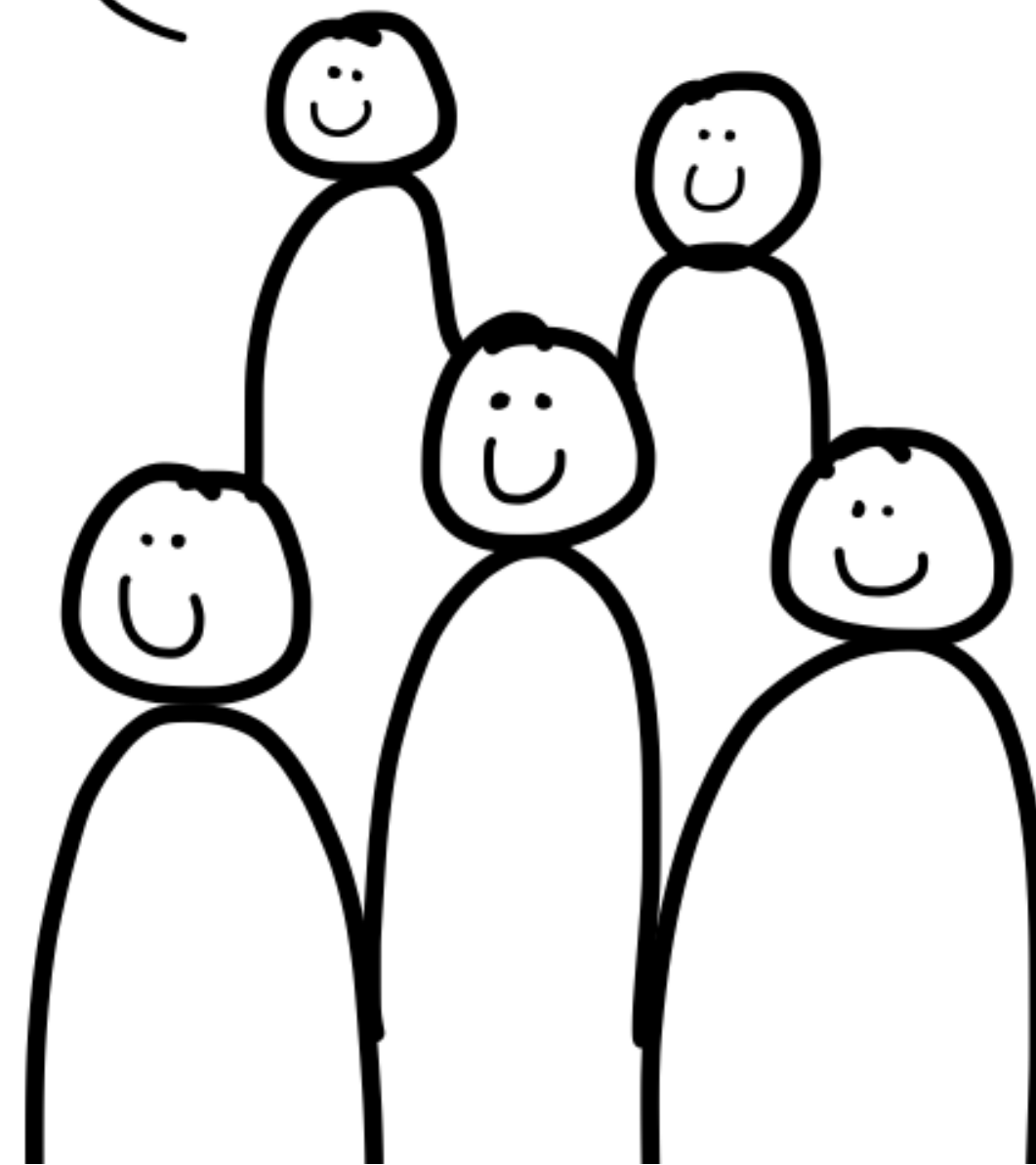


How does code sound?

$x = 5$

x equals 5

x is 5



How does code sound?

x = 5

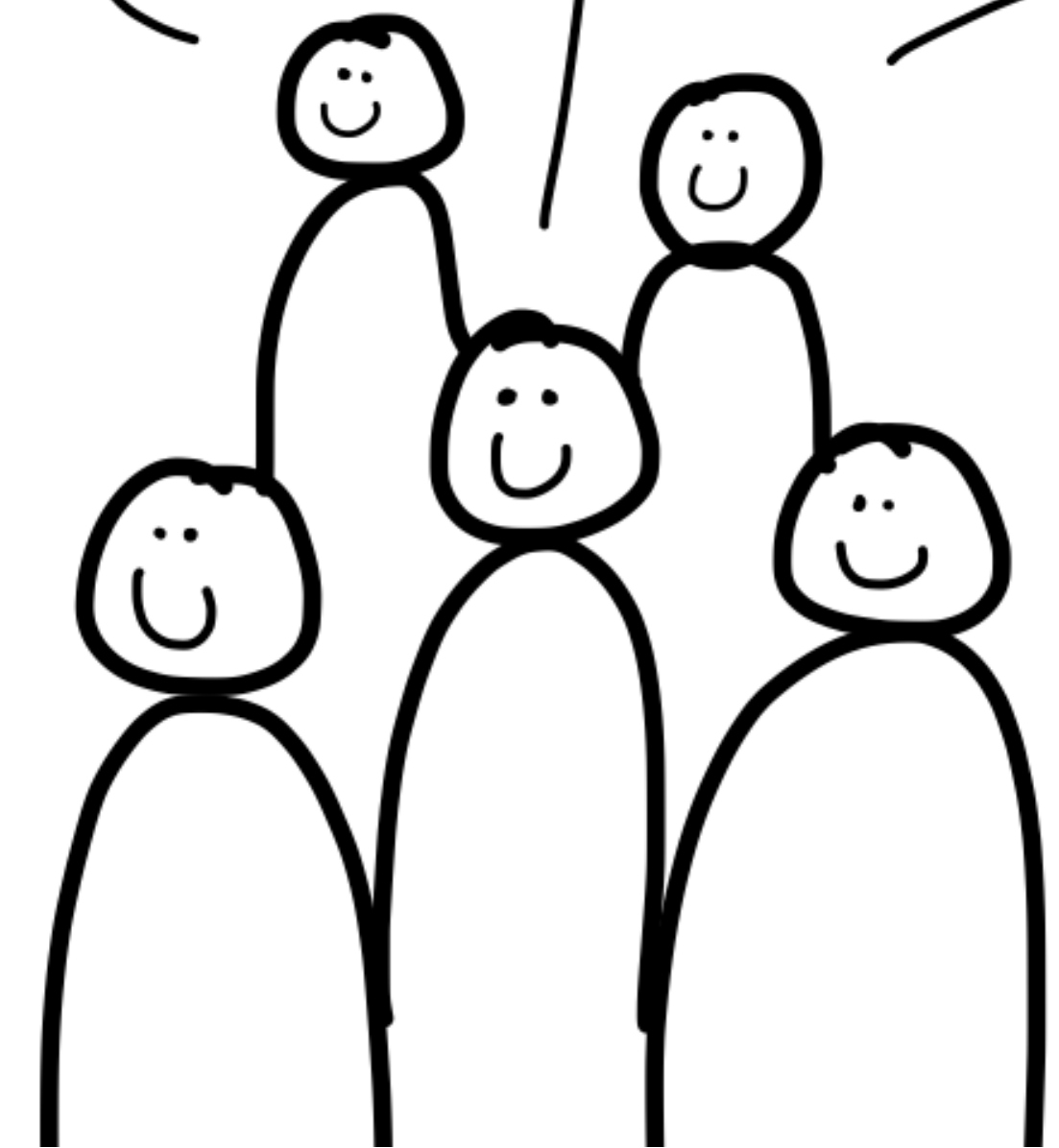
x gets 5

x equals 5

set x to 5

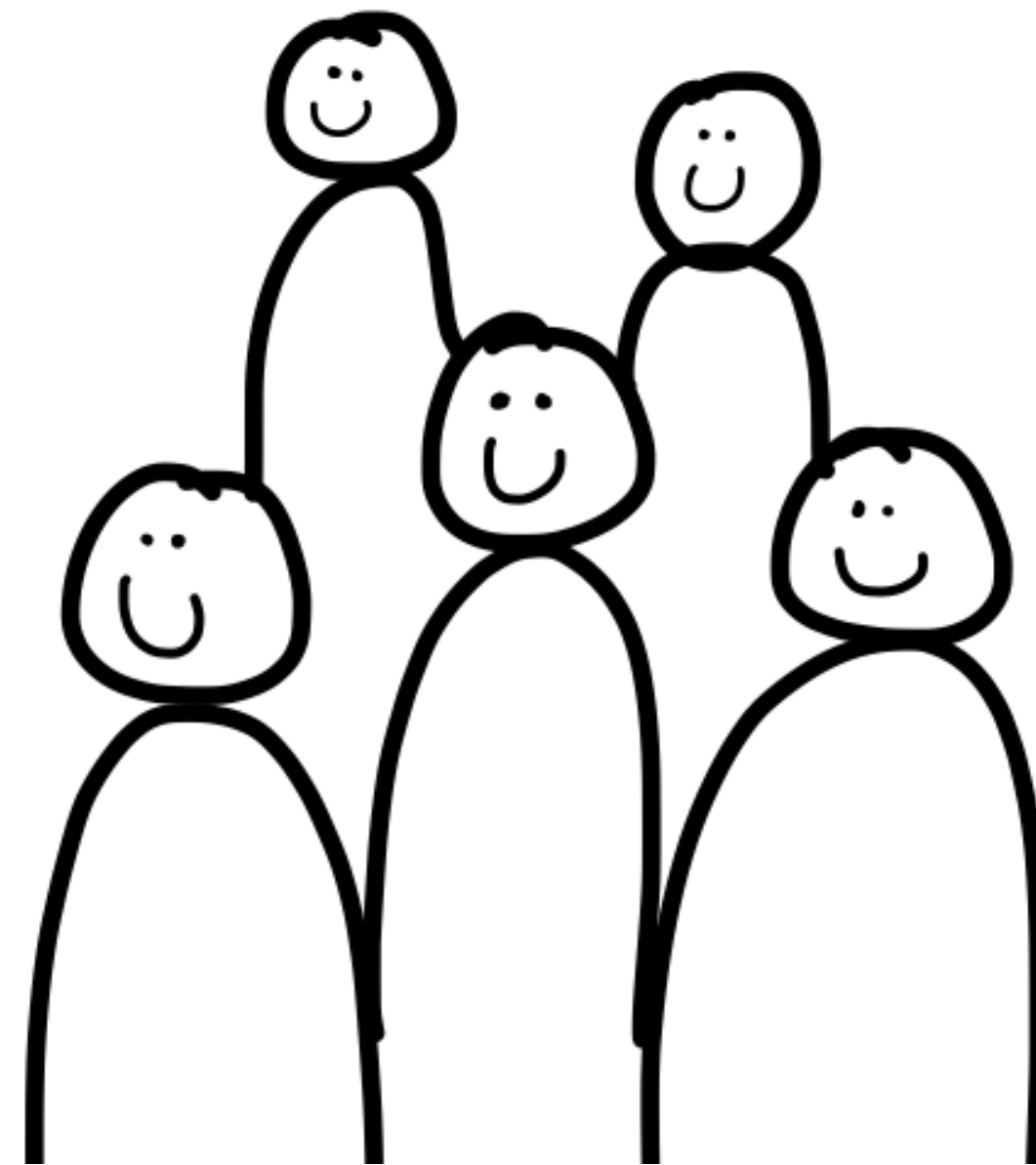
x is 5

assign 5 to x



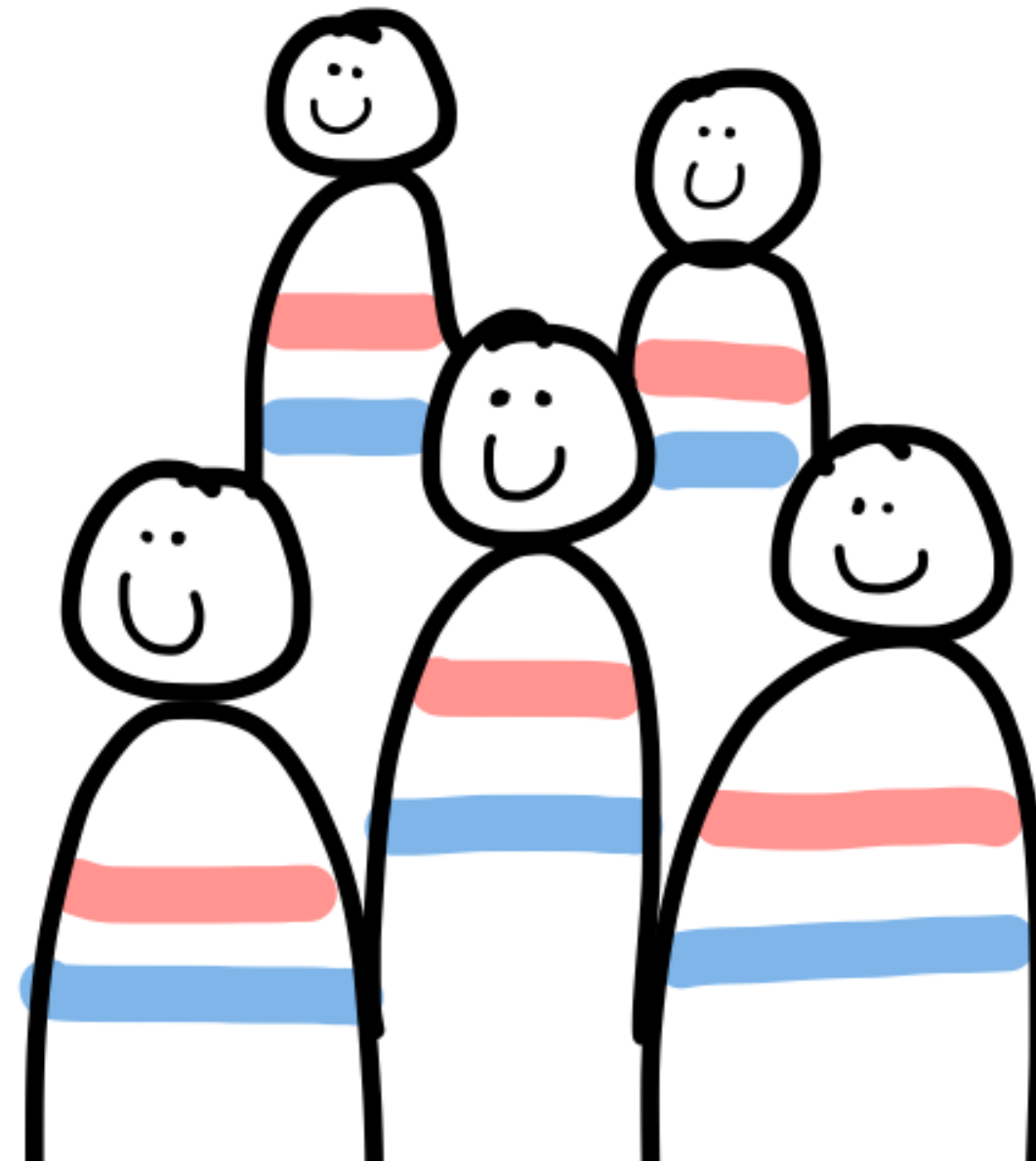
How does code sound?

we asked 10 novices



How does code sound?

we asked 10 novices
^
Dutch



Native tongue plays a role

```
for i in range(3):  
    print(i)
```



Native tongue plays a role

— "ie" as in tie

```
for i in range(3):
```

```
    print(i)
```



Native tongue plays a role

— "ee" as in creep

```
for i in range(3):  
    print(i)
```

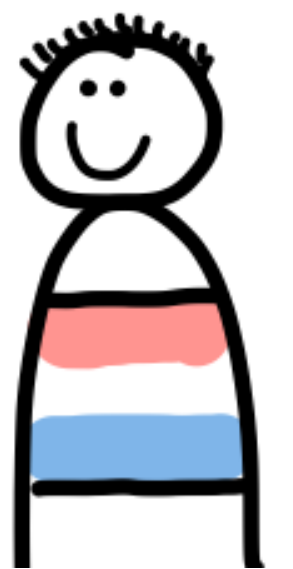


Native tongue plays a role

— "ee" as in creep

```
for i in range(3):  
    print(i)
```

```
for ie in range...  
    print ee
```

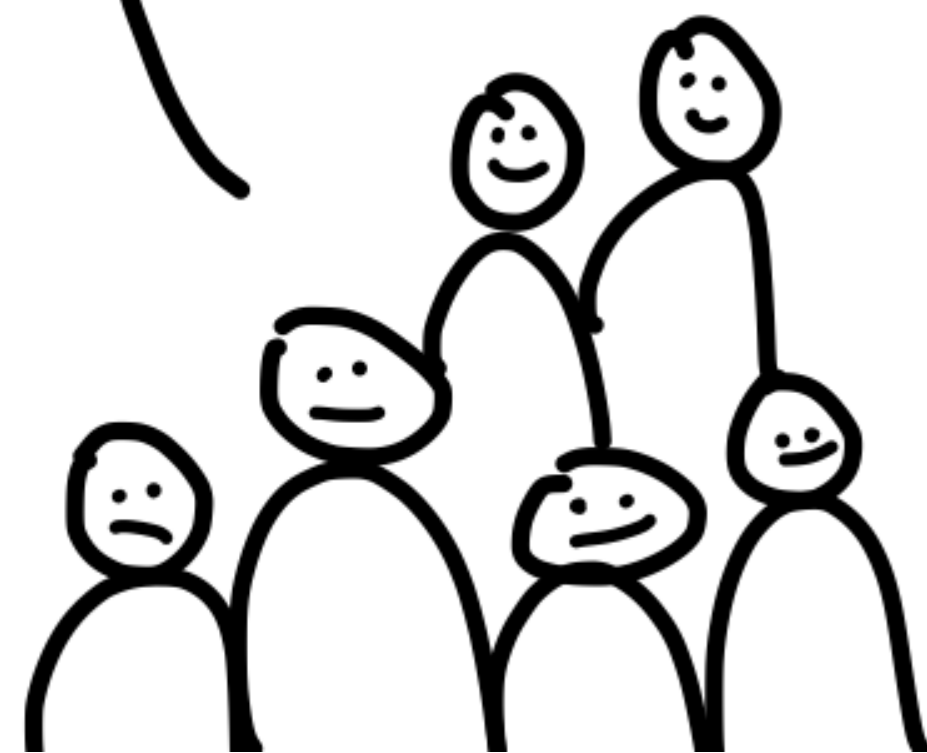


We should tell kids

repeat after me: $x = 5$



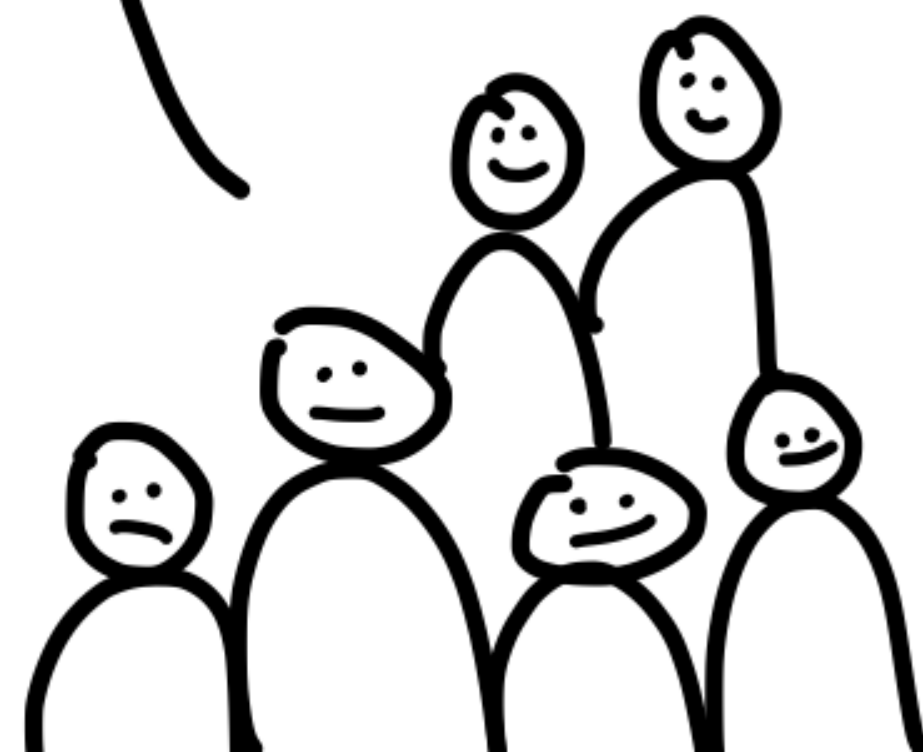
"x is 5"



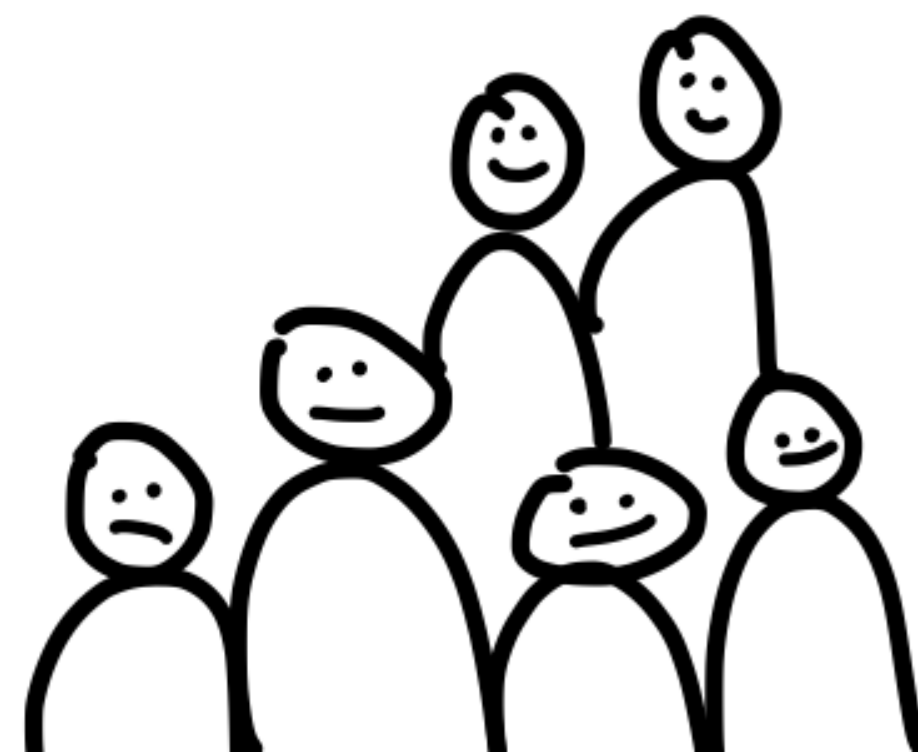
repeat after me:

for i in range(4):

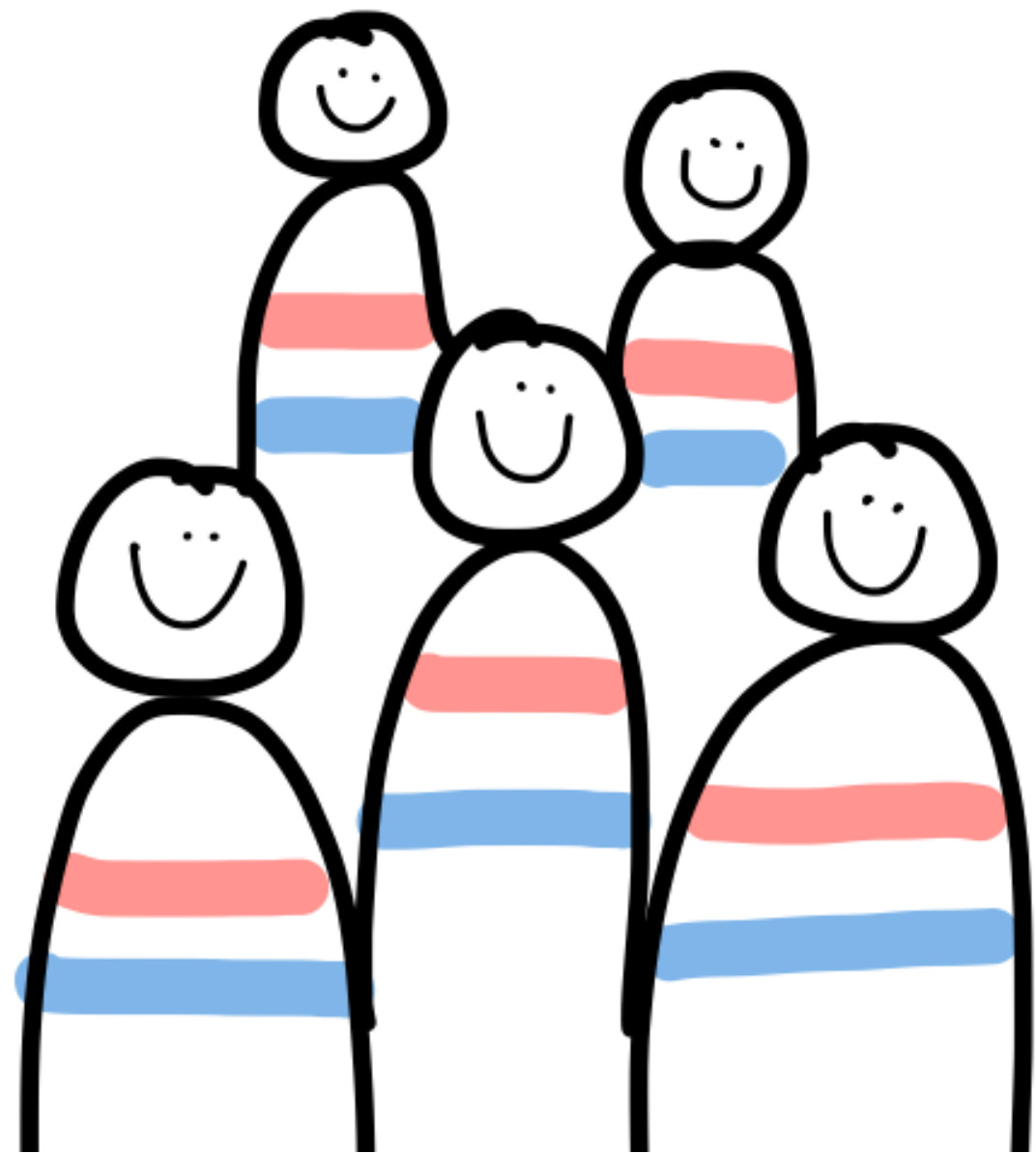
"for ie in reeng..."



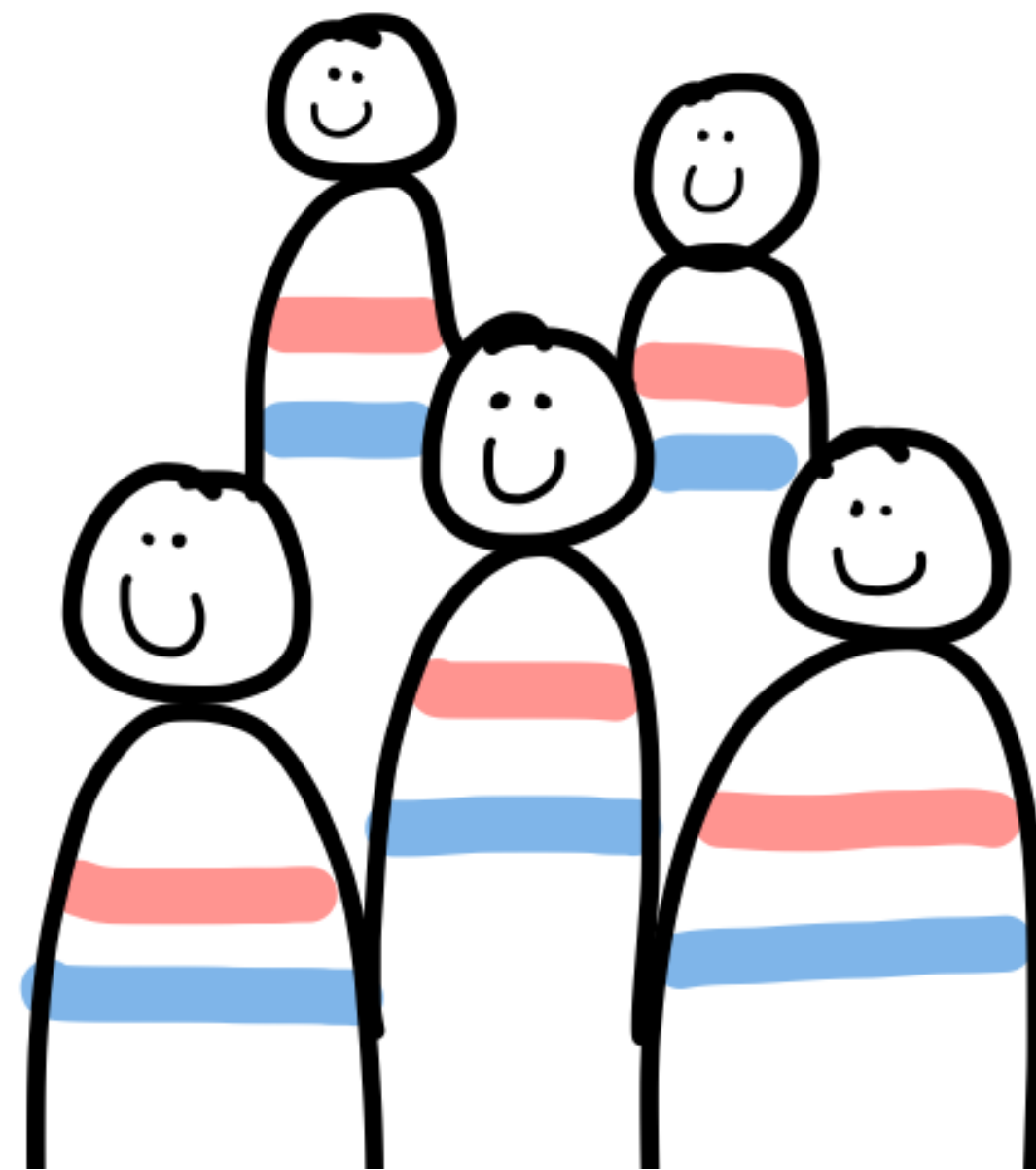
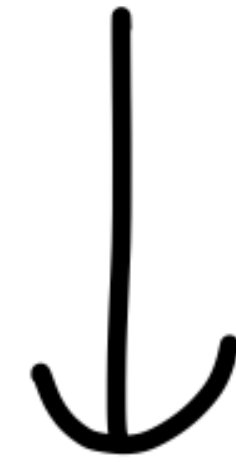
Practicing this helps!

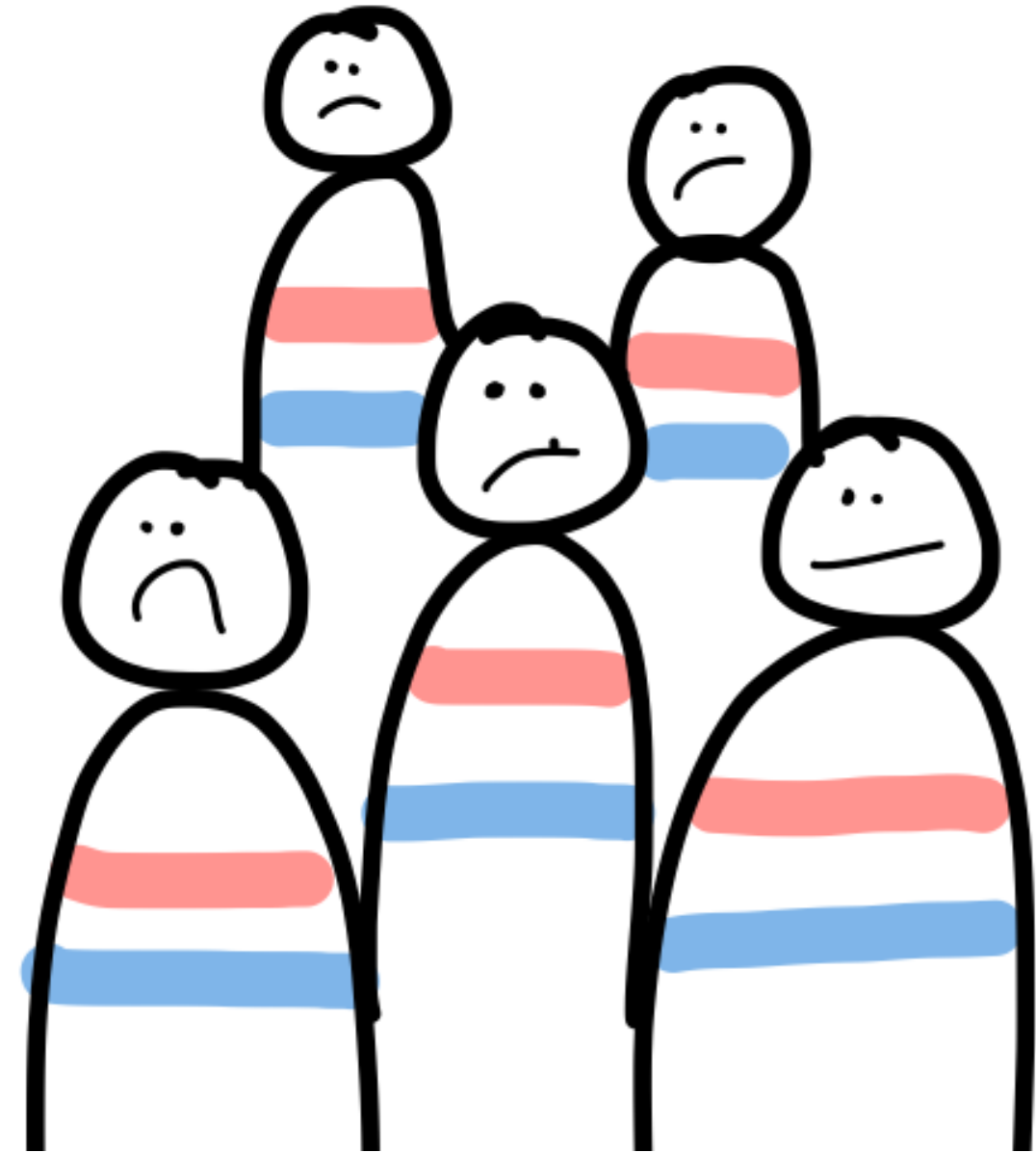
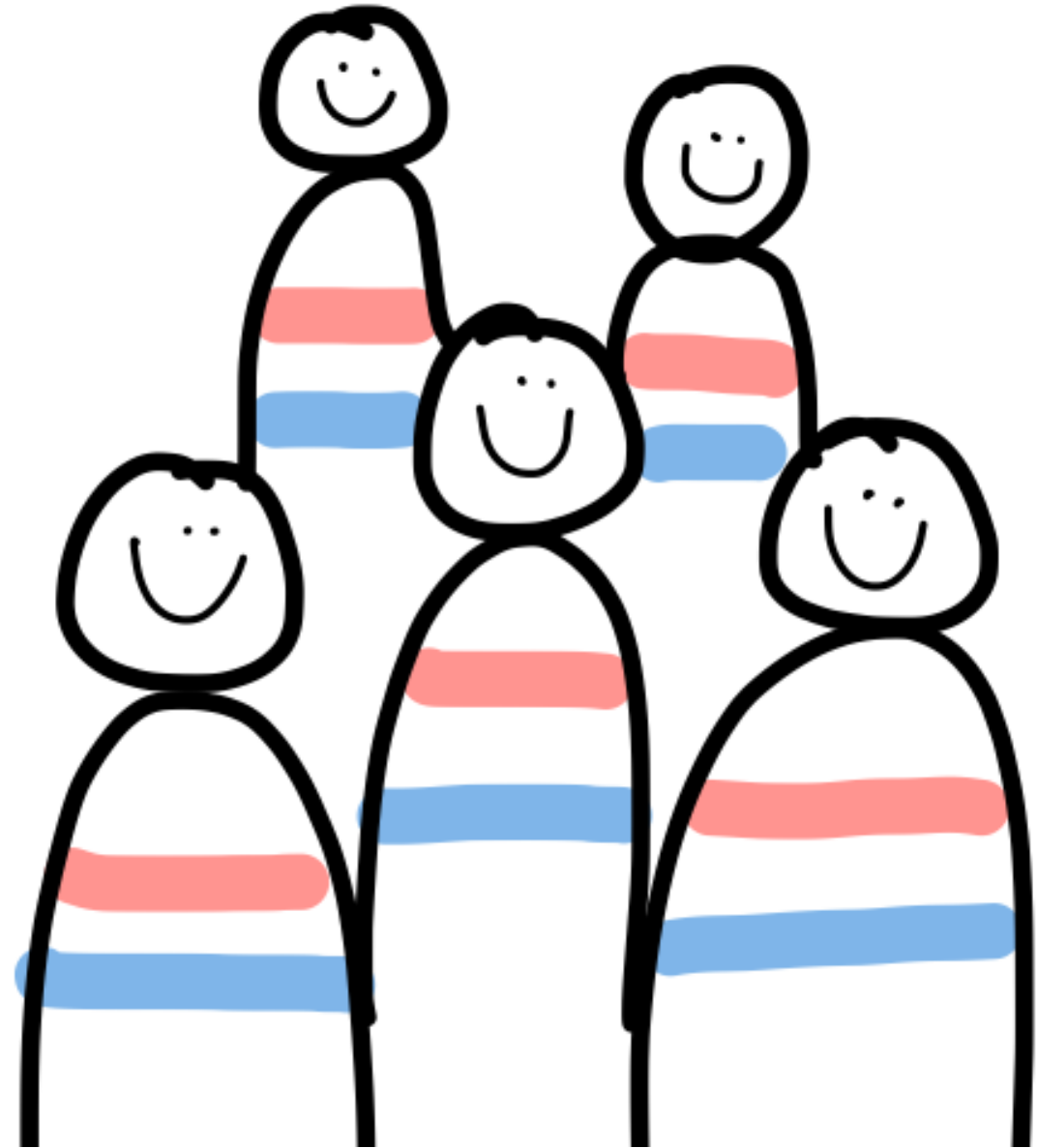
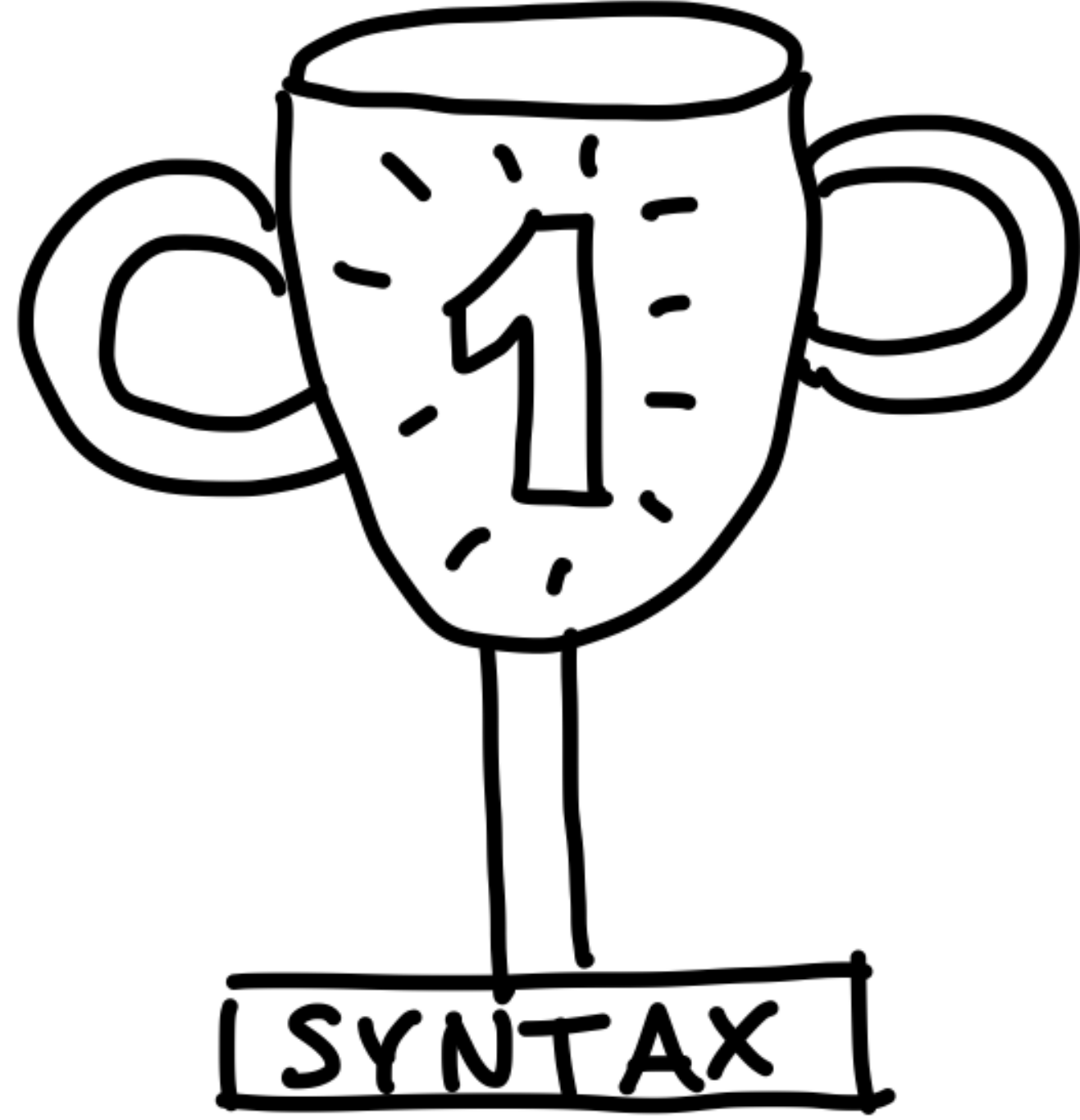


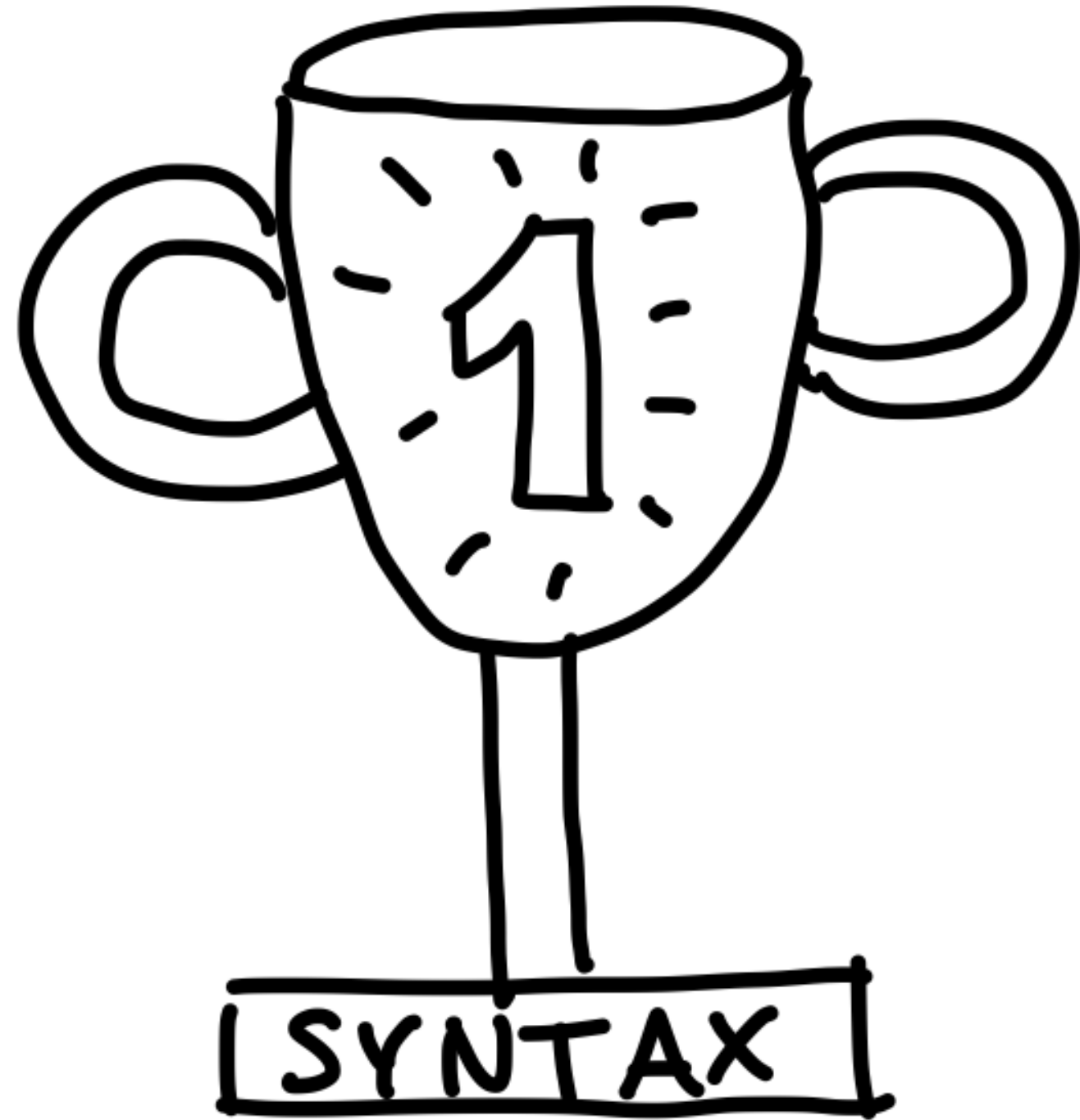
Read code aloud
every week



Control group

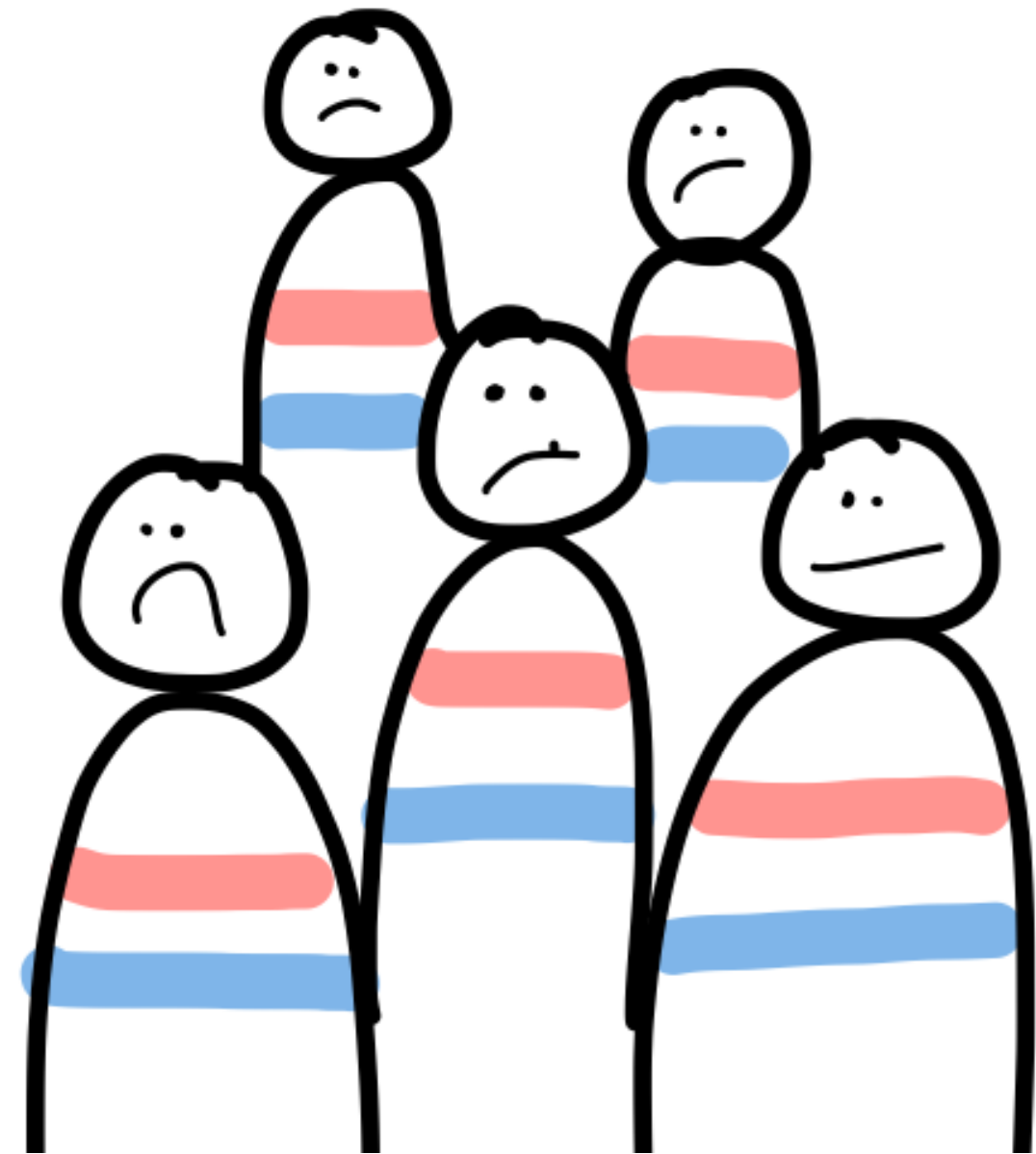
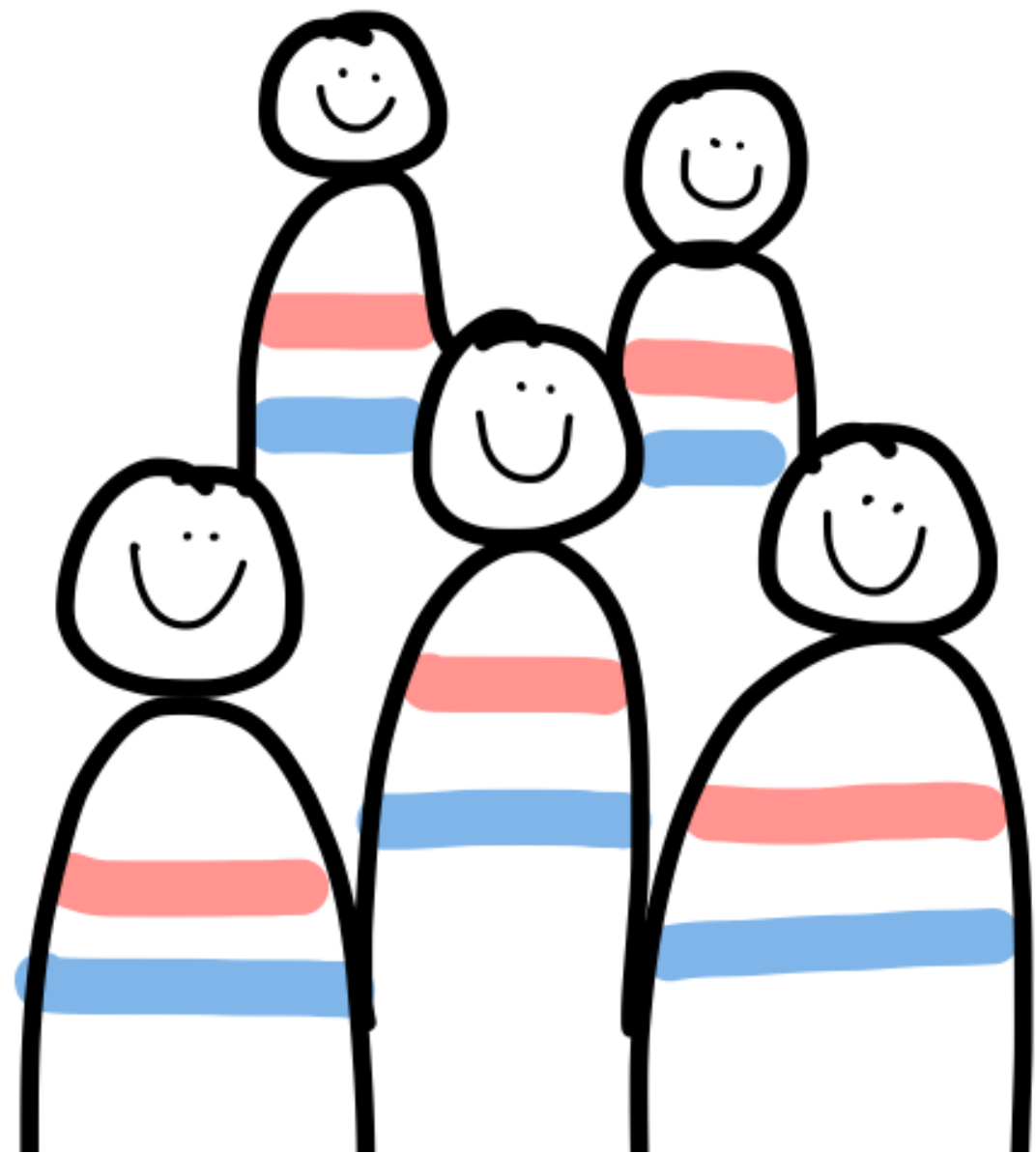






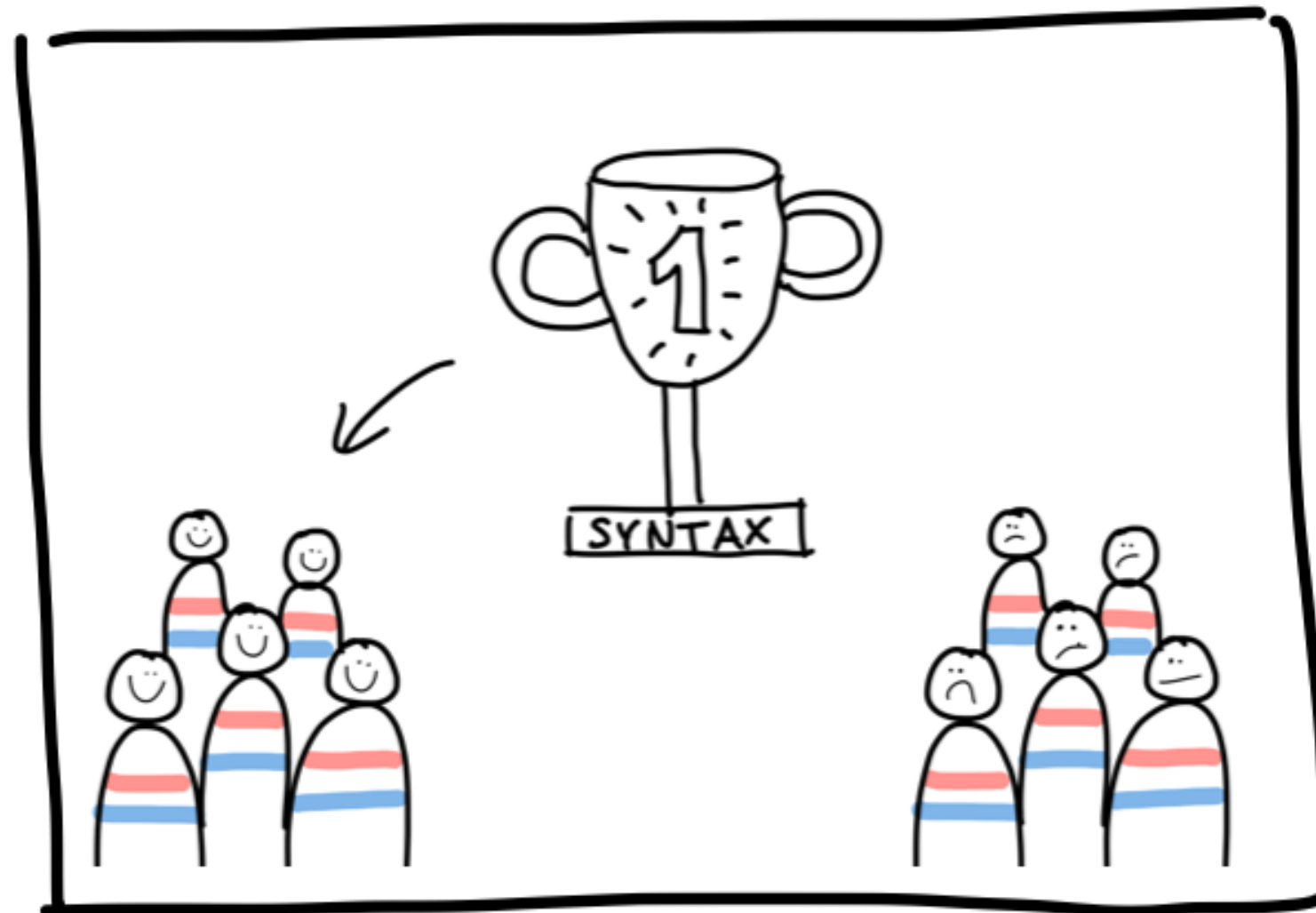
The effect of reading code
aloud on Comprehension
Swidan & Hermans '19

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____



How to teach programming?

Vocalise syntax



How to teach programming?



How to teach programming?

Tell students how
to solve problems!



How to teach programming?

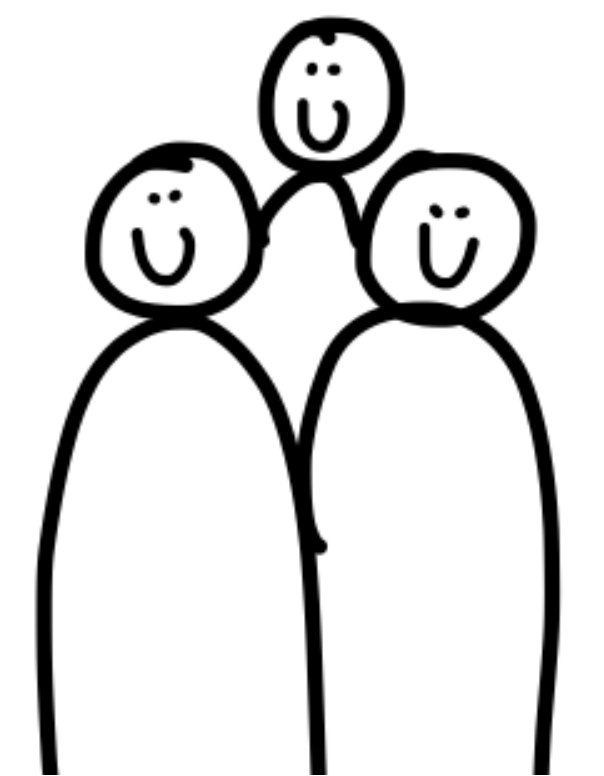
The case for case studies in Pascal
Linn & Clancy '92

_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____



How to teach programming?

write own code
&
read expert code

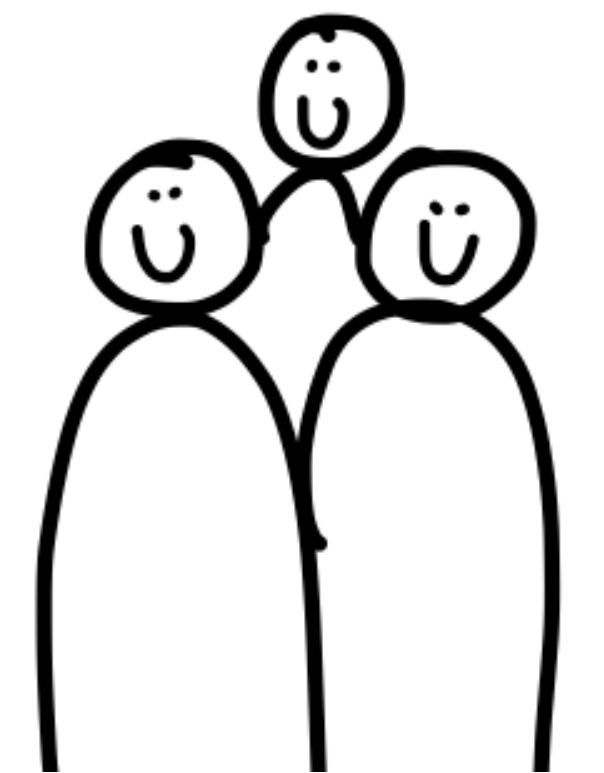


How to teach programming?

write own code

&

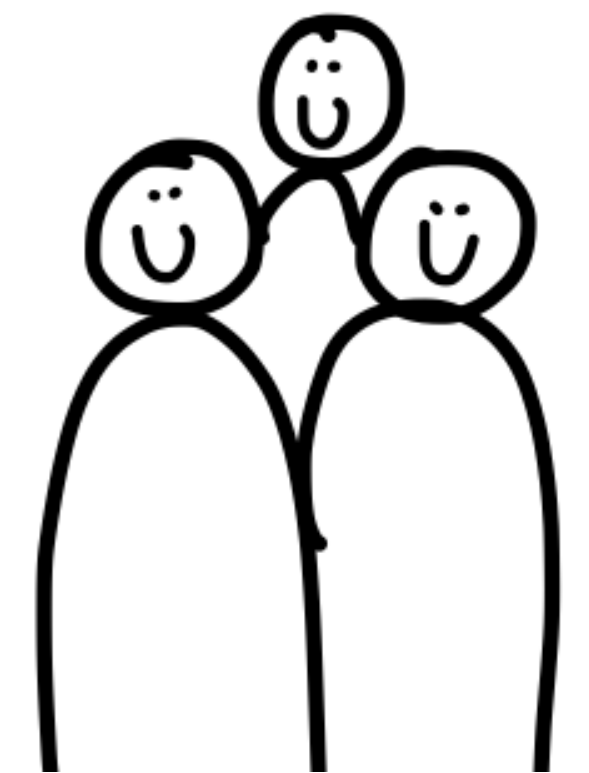
read expert code + explanation



How to teach programming?

~~write own code~~

read expert code + explanation

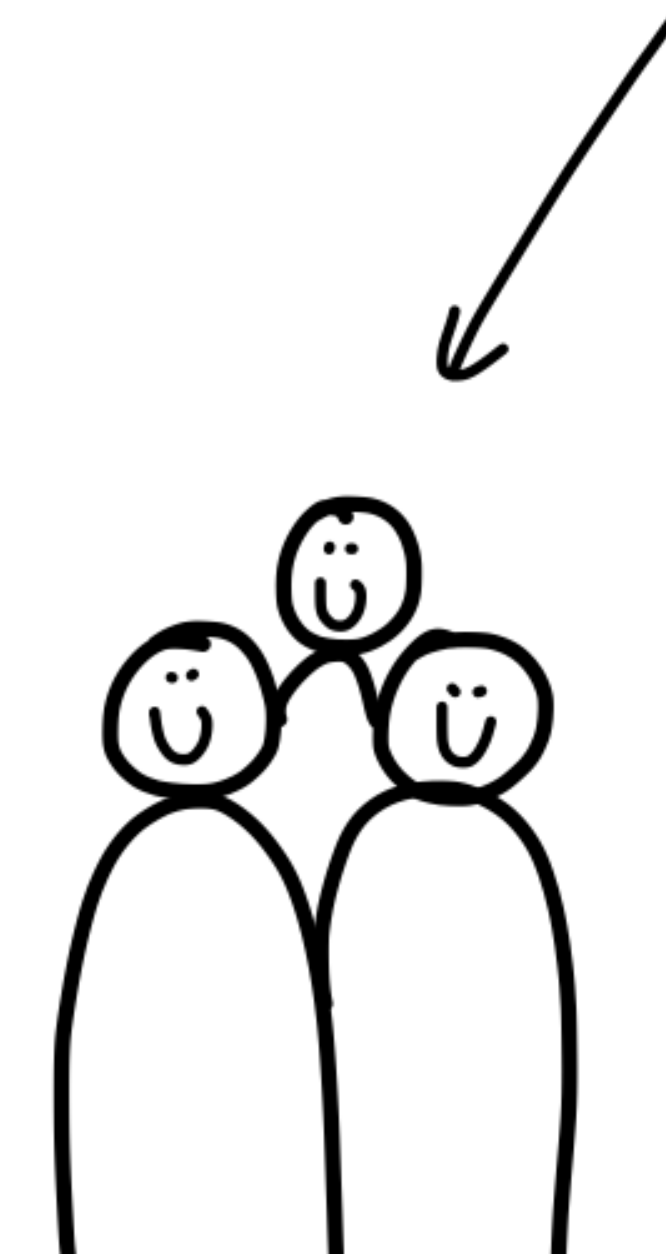
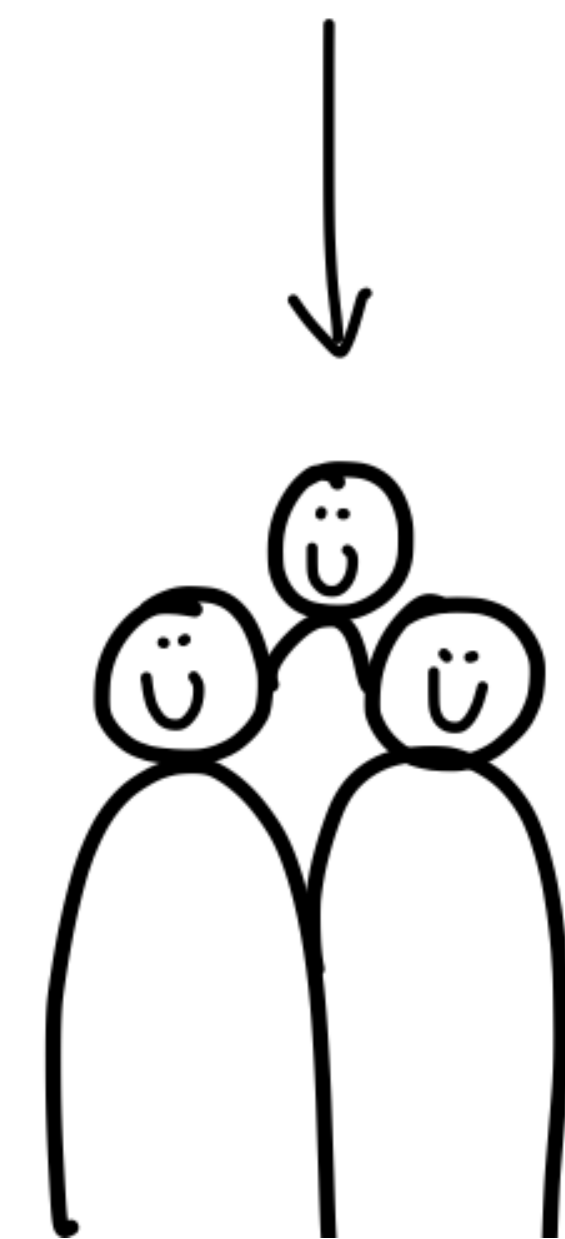


How to teach programming?

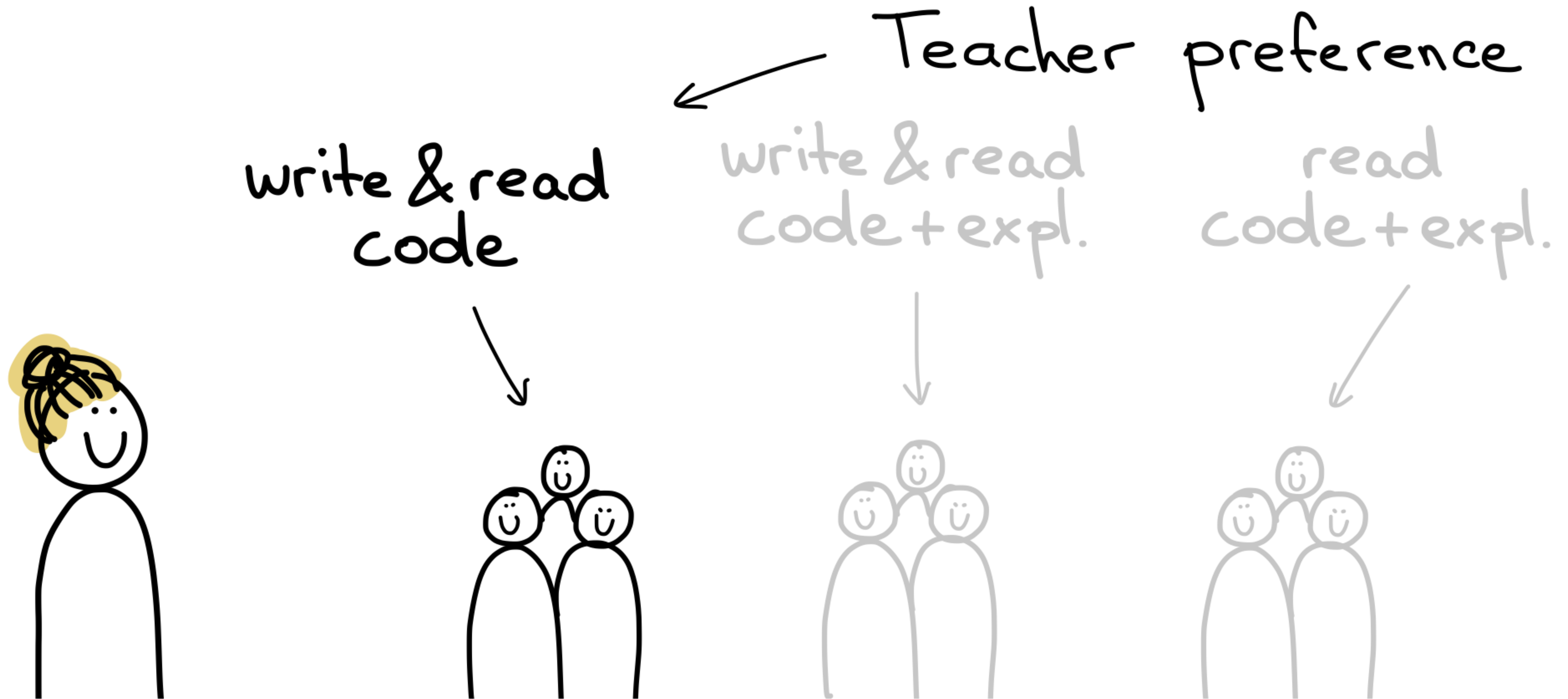
write & read
code

write & read
code + expl.

read
code + expl.



How to teach programming?

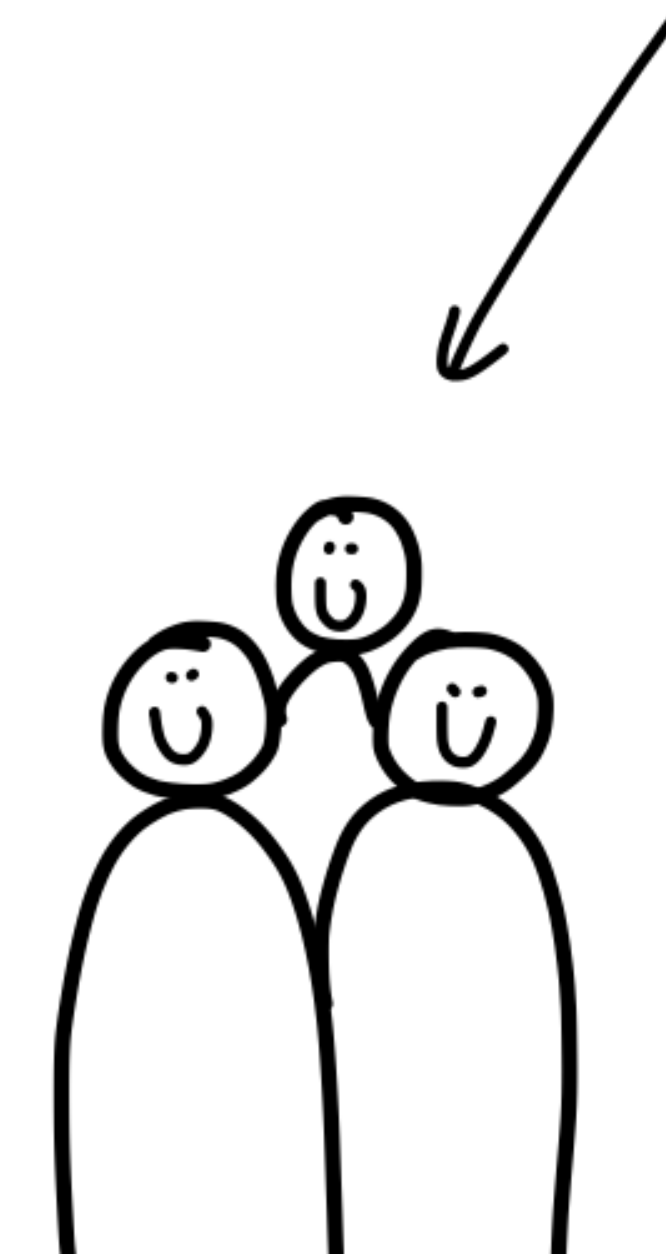
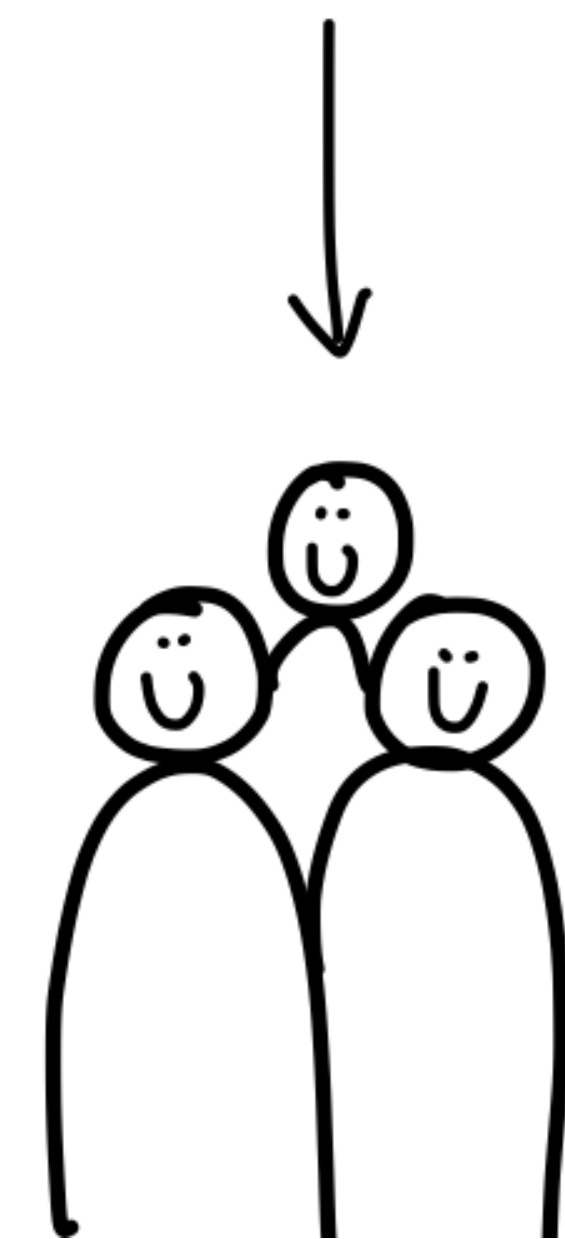


How to teach programming?

write & read
code

write & read
code + expl.

read
code + expl.

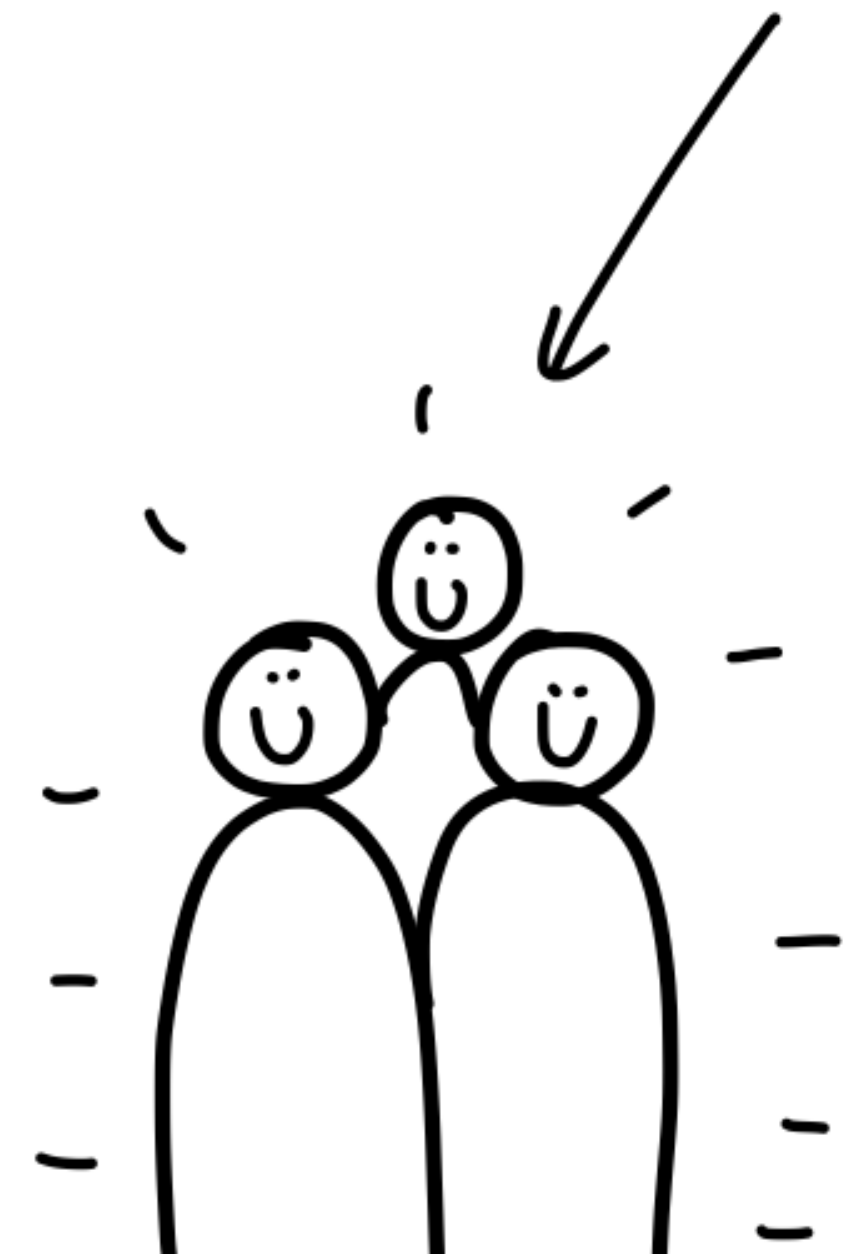
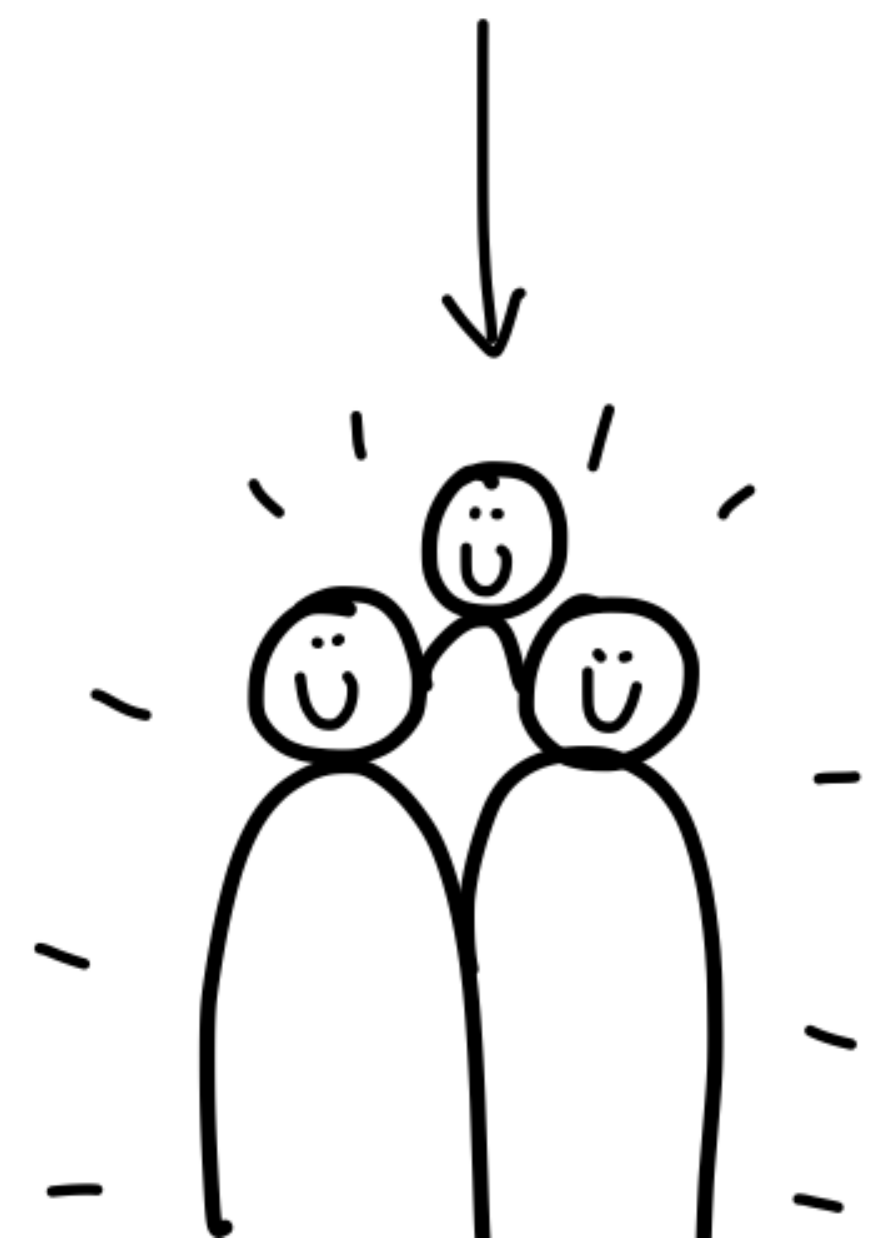


How to teach programming?

write & read
code

write & read
code + expl.

read
code + expl.



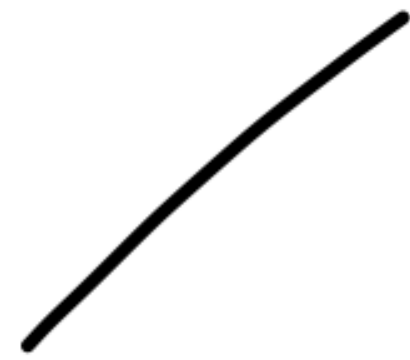
How to teach programming?

Explicit strategies



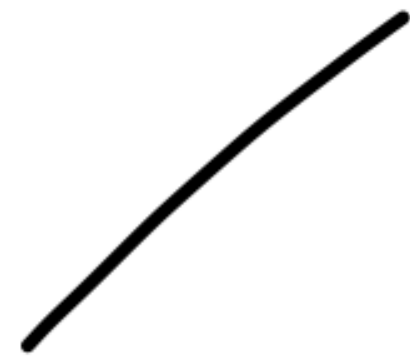
How to teach programming?

Assessment



How to teach programming?

Assessment



Which one is a variable? Circle that one.

number of kids

say number of kids for 2 seconds

wait 2 seconds

You want to set the variable points to 0.

Which block does that?

set points to 0

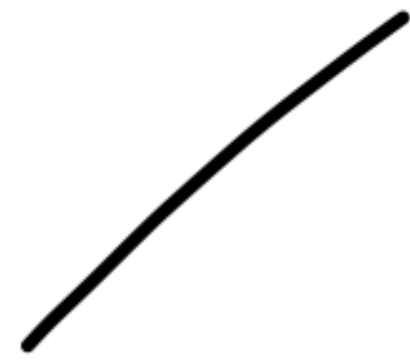
change points by 0

say points for 0 seconds

How to teach programming?

Assessment

We need you!



How to teach programming?

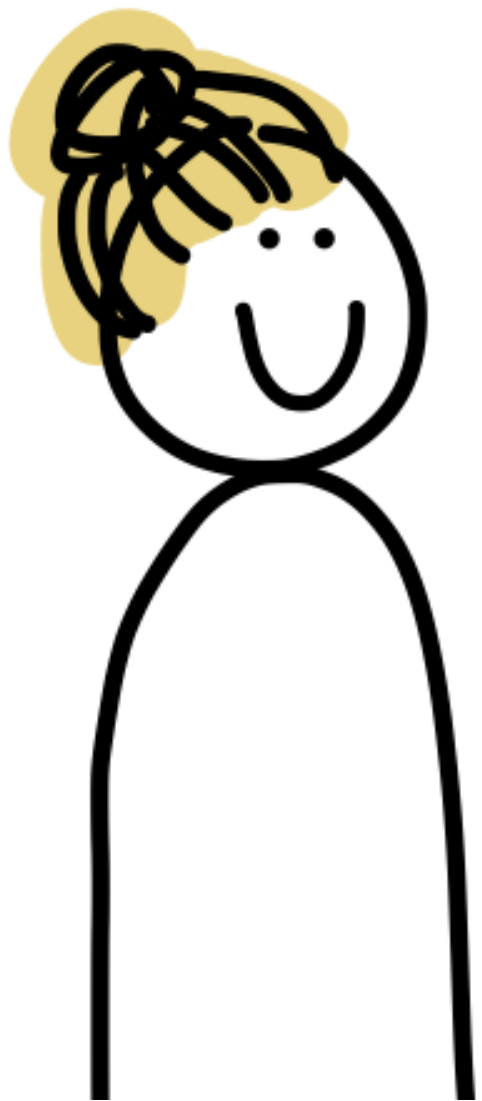
Assessment

We need you!

bit.ly/LizzysAfstuderen



How to teach programming?



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Uitleg over het materiaal

De Python in de klas lessen zijn geschikt vanaf ongeveer brugklas. Leerlingen hoeven helemaal geen voorkennis te hebben en de leerkracht eigenlijk ook niet. Het materiaal bestaat uit 3 modules: Nederlands, kunst en aardrijkskunde waarin leerlingen achtereenvolgens een interactief verhaal, een kunstwerk en een datavisualisatie maken.

Klinkt dit leuk? Lees dan [een uitgebreidere handleiding](#).

Module Nederlands

Slides

- [Module Nederlands, les 1](#)
- [Module Nederlands, les 2](#)
- [Module Nederlands, les 3](#)
- [Module Nederlands, les 4](#)
- [Module Nederlands, les 5](#)
- [Module Nederlands, les 6](#)

How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Titel over het materiaal

NAAM NIET GEDEFINIEERD

Soms maak je een foutje, en gebruik je een variabele die niet is gedefinieerd.

```
naam = 'Feliene'  
print(voornaam)
```

- Python zoekt nu voornaam, maar die is er niet!
- Je krijgt dan een foutmelding



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Titel over het materiaal

DEMO

```
main.py  open in repl.it
1 naam = 'Feliene'
2 print(voornaam)
3
4
5
6
7
8
```

Python 3.7.4 (default, Jul 9 2019, 00:06:43)
[GCC 6.3.0 20170516] on linux



- Module Nederlands, les 6



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Titel over het materiaal

DEMO

```
main.py  open in repl.it
1 naam = 'Feliene'
2 print(voornaam)
3
4
5
6
7
8
```

Traceback (most recent call last):
File "main.py", line 2, in <module>
print(voornaam)
NameError: name 'voornaam' is not defined



- Module Nederlands, les 6



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Waardes zoeken

1) In Python kun je woorden zonder aanhalingstekens gebruiken. Dat zijn variabelen. Als Python die tegenkomt gaat hij omhoog zoeken in het programma naar de *definitie* van de variabele. Dat betekent waar de variabele wordt ingesteld met een is-teken:

```
naam = 'Feliene'
```

Schrijf deze codes over. Zet een pijltje tussen het gebruik van een variabele en zijn definitie.

Voorbeeld: `klas = 'c'`
`print('Hallo', klas)`

```
1. les = 'Coderen'  
   print('welkom', 'bij', les)
```

```
2. klas = '1c'  
   print('Hallo', klas)
```


How to teach programming?

← → ↻ lyceumkralingen.nl/de-school/onderwijs/codasium/



HEEL DE WERELD WORDT JE THUIS

technasium lyceum kralingen contact

🔍 typ en druk op enter



De School ▾

Onderwijs ▾

Leerlingen ▾

Ouders ▾

Groep 8 ▾

Nieuws

Aanmelden?



Lyceum Kralingen

Visie en missie

Unesco School

Organisatie

Onderwijs

Coderen

Codasium

Wereldburgerschap

Technasium

Partners

Vakken

Codasium; Op het Lyceum Kralingen leiden wij je op tot wereldburger: iemand die zich overal thuis voelt en zich inspant om het samenleven in de wereld te verbeteren. Dan hebben we het natuurlijk over de ;

wereld van de toekomst > wanneer jij van school komt en over de

toekomst van de wereld > die jij samen met anderen gaat bepalen.

Voor die toekomst is het niet alleen handig, maar zelfs noodzakelijk dat je goed de weg weet in de digitale wereld van internet, computers en smart phones. De leukste en snelste manier om heel veel over digitale technologie te weten te komen is: zelf leren coderen. Vroeger noemden ze dit ook wel 'programmeren'.

coderen = het omzetten van een idee in een taal die je computer begrijpt

We beginnen met games. Je leert eerst wat je moet doen (welke codes je moet

How to teach programming?

lyceumkralingen.nl/de-school/onderwijs/codasium/

HEEL DE WERELD WORDT JE THUIS



De School ▾ Onderwijs ▾



Lyceum Kralingen

Visie en missie

Unesco School

Organisatie

Onderwijs

Coderen

Codasium

Wereldburgerschap

Technasium

Partners

Vakken

Codasium; Op het Lyceum Kralingen

zich overal thuis voelt en zich ins

verbeteren. Dan hebben we het r

wereld van de toekomst > want

toekomst van de wereld > die j

Voor die toekomst is het niet alle

de weg weet in de digitale wereld

leukste en snelste manier om hee

komen is: zelf leren coderen. Vro

coderen = het omzetten van eer

We beginnen met games. Je leert eerst wat je moet doen (welke codes je moet

Programma i&i conferentie - donderdag 7 november 2019					
	BBG-020	BBG-069	BBG-071	BBG-075	BBG-077
09.00 - 10.00	Foyer: Inloop en registratie - koffie/thee en bezoek informatiemarkt				
10.00 - 10.15	Opening in Auditorium Cosmos, Welkom door i&i bestuur				
	Toelichting landelijke nascholing keuzedomeinen door Bestuur en Martin Bruggink				
10.15 - 11.00	Toelichting programma i&i - conferentie				
	Keynote "Effectief leren programmeren" door Felienne Hermans				
11.00 - 11.45	Keuzedomein H: Databases - Eelco Dijkstra & Frans Peeters	Security - Erik Poll	Toetsing van Grondslagen (domein B) en programmeren (domein D - Renske Weeda & Martin Bruggink	Toelichting Digitale Geletterdheid uit Curriculum.nu - Koen Buiter	Codasium; programmeeronderwijs in onder- én bovenbouw - Luc van Leeuwe
11.45 - 12.30			Praktische opdracht bij domein R computational science - Nataša Grgurina	Digitaal geletterd met DIGIT! - Hafida Makioul (Instruct)	Lessen en lesmaterialen over eindige automaten (domein B: grondslagen) - Philip Braas
12.30 - 13.30	Lunch en bezoek informatiemarkt				
13.30 - 14.15	Keuzedomein R: Computational Science - Nataša Grgurina & René van der Veen	Leren programmeren door muziek te maken - Christian Köppe	IOT Netwerken en slimme oplossingen - Willem-Jan Derks (Cisco)	Computer Science Certificate - Basisexamen programmeren VO - VHTO	Studententeam Project MARCH (TU Delft) maakt exoskelet
14.15 - 15.00				De didactiek van genderinclusief programmeren - VHTO	Virtuele race met zelfrijdende auto's- Karel Bruneel"
15.00 - 15.15	koffie/thee en bezoek informatiemarkt				
15.15 - 16.00	Keuzedomein J: Programmeerparadigma's / Functioneel programmeren - Kees Huizing & Hans van den Burg	Rekenen met de Quantumcomputer - CWI - Christian Schaffner	Beauty and Joy of Computing - Casper Kole	Fundament Informatica, docenten aan het woord - Adriaan Gijssen (Instruct)	Raspberry Pi Competities- Ernst Brand (PA Consulting) & Krijn Hoogendorp
16.00 - 16.45					Computational Thinking in de Klas - Eljakim Schrijvers
16.45-17.30	Foyer: Napraten met hapje en drankje - bezoek informatiemarkt				

How to teach programming?

lyceumkralingen.nl/de-school/onderwijs/codasium/

HEEL DE WERELD WORDT JE THUIS



De School ▾ Onderwijs ▾



Lyceum Kralingen

Visie en missie

Unesco School

Organisatie

Onderwijs

Coderen

Codasium

Wereldburgerschap

Technasium

Partners

Vakken

Codasium; Op het Lyceum Kralingen

zich overal thuis voelt en zich ins

verbeteren. Dan hebben we het r

wereld van de toekomst > want

toekomst van de wereld > die j

Voor die toekomst is het niet alle

de weg weet in de digitale wereld

leukste en snelste manier om hee

komen is: zelf leren coderen. Vro

coderen = het omzetten van eer

We beginnen met games. Je leert eerst wat je moet doen (welke codes je moet

Programma i&i conferentie - donderdag 7 november 2019					
	BBG-020	BBG-069	BBG-071	BBG-075	BBG-077
09.00 - 10.00	Foyer: Inloop en registratie - koffie/thee en bezoek informatiemarkt				
10.00 - 10.15	Opening in Auditorium Cosmos, Welkom door i&i bestuur				
	Toelichting landelijke nascholing keuzedomeinen door Bes en Martin Bruggink				
10.15 - 11.00	Keynote "Effectieve programmeren"				
11.00 - 11.45	Keuzedomein H: Databases - Eelco Dijkstra & Frans Peeters	Security - Erik Poll	praktische opdrachten (domein R) - Renske Weeda & Martin Bruggink	Cursus Programmeren - Jeroen van der Vliet	Codasium; programmeeronderwijs in onder- én bovenbouw - Luc van Leeuwe
11.45 - 12.30			Praktische opdracht bij domein R computational science - Nataša Grgurina		Digitale leeromgeving met DIGIT! - Nataša Grgurina
12.30 - 13.30	Lunch en bezoek informatiemarkt				
13.30 - 14.15	Keuzedomein R: Computational Science - Nataša Grgurina & René van der Veen	Leren programmeren door muziek te maken - Christian Köppe	IOT Netwerken en slimme oplossingen - Willem-Jan Derks (Cisco)	Computer Science Certificate - Basisexamen programmeren VO - VHTO	Studententeam Project MARCH (TU Delft) maakt exoskelet
14.15 - 15.00				De didactiek van genderinclusief programmeren - VHTO	Virtuele race met zelfrijdende auto's- Karel Bruneel"
15.00 - 15.15	koffie/thee en bezoek informatiemarkt				
15.15 - 16.00	Keuzedomein J: Programmeerparadigma's / Functioneel programmeren - Kees Hulzing & Hans van den Burg	Rekenen met de Quantumcomputer - CWI - Christian Schaffner	Beauty and Joy of Computing - Casper Kole	Fundament Informatica, docenten aan het woord - Adriaan Gijssen (Instruct)	Raspberry Pi Competities- Ernst Brand (PA Consulting) & Krijn Hoogendorp
16.00 - 16.45					Computational Thinking in de Klas - Eljakim Schrijvers
16.45-17.30	Foyer: Napraten met hapje en drankje - bezoek informatiemarkt				



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Uitleg over het materiaal

De Python in de klas lessen zijn geschikt vanaf ongeveer brugklas. Leerlingen hoeven helemaal geen voorkennis te hebben en de leerkracht eigenlijk ook niet. Het materiaal bestaat uit 3 modules: Nederlands, kunst en aardrijkskunde waarin leerlingen achtereenvolgens een interactief verhaal, een kunstwerk en een datavisualisatie maken.

Klinkt dit leuk? Lees dan [een uitgebreidere handleiding](#).

Module Nederlands

Slides

- [Module Nederlands, les 1](#)
- [Module Nederlands, les 2](#)
- [Module Nederlands, les 3](#)
- [Module Nederlands, les 4](#)
- [Module Nederlands, les 5](#)
- [Module Nederlands, les 6](#)

How to teach programming?

← → ↻ vhto.nl/projecten/computer-science-certificate/ ☆ 👤 ⋮



[OVER VHTO](#) [NIEUWS](#) [AANMELDEN NIEUWSBRIEF](#) [ENGLISH](#)

Typ hier uw zoekopdracht...



[Onderwijs](#) [Professionals & Bedrijven](#) [Meisjes & Ouders](#) [Cijfers & Onderzoek](#) [Projecten](#)



How to teach programming?

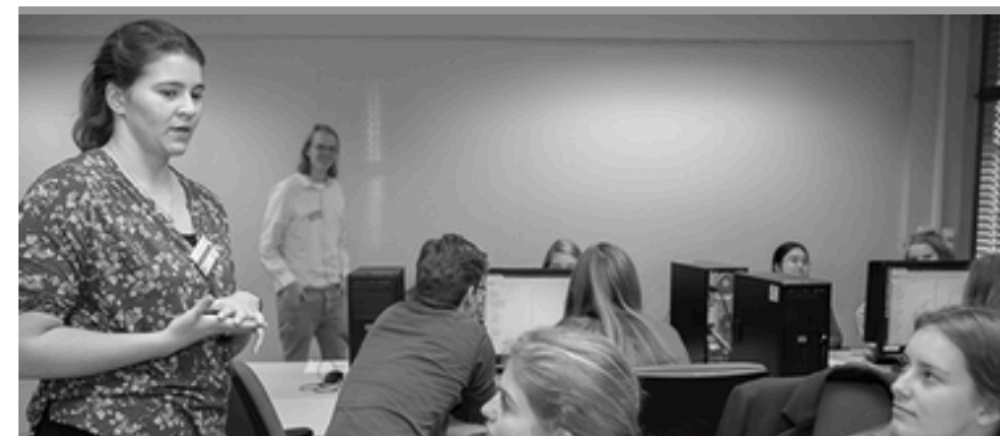
← → ↻ vhto.nl/projecten/computer-science-certificate/ ☆ 🌐 ⋮



Onderwijs Profes



HOME > PROJECTEN > COMPUTER SCIENCE CERTIFICATE



Programma i&i conferentie - donderdag 7 november 2019					
	BBG-020	BBG-069	BBG-071	BBG-075	BBG-077
09.00 - 10.00	Foyer: Inloop en registratie - koffie/thee en bezoek informatiemarkt				
10.00 - 10.15	Opening in Auditorium Cosmos, Welkom door i&i bestuur				
	Toelichting landelijke nascholing keuzedomeinen door Bestuur en Martin Bruggink				
	Toelichting programma i&i - conferentie				
10.15 - 11.00	Keynote "Effectief leren programmeren" door Felienne Hermans				
11.00 - 11.45	Keuzedomein H: Databases - Eelco Dijkstra & Frans Peeters	Security - Erik Poll	Toetsing van Grondslagen (domein B) en programmeren (domein D - Renske Weeda & Martin Bruggink	Toelichting Digitale Geletterdheid uit Curriculum.nu - Koen Buiter	Codasium; programmeeronderwijs in onder- én bovenbouw - Luc van Leeuwe
11.45 - 12.30			Praktische opdracht bij domein R computational science - Nataša Grgurina	Digitaal geletterd met DIGIT! - Hafida Makioul (Instruct)	Lessen en lesmaterialen over eindige automaten (domein B: grondslagen) - Philip Braas
12.30 - 13.30	Lunch en bezoek informatiemarkt				
13.30 - 14.15	Keuzedomein R: Computational Science - Nataša Grgurina & René van der Veen	Leren programmeren door muziek te maken - Christian Köppe	IOT Netwerken en slimme oplossingen - Willem-Jan Derks (Cisco)	Computer Science Certificate - Basisexamen programmeren VO - VHTO	Studententeam Project MARCH (TU Delft) maakt exoskelet
14.15 - 15.00				De didactiek van genderinclusief programmeren - VHTO	Virtuele race met zelfrijdende auto's- Karel Bruneel"
15.00 - 15.15	koffie/thee en bezoek informatiemarkt				
15.15 - 16.00	Keuzedomein J: Programmeerparadigma's / Functioneel programmeren - Kees Hulzing & Hans van den Burg	Rekenen met de Quantumcomputer - CWI - Christian Schaffner	Beauty and Joy of Computing - Casper Kole	Fundament Informatica, docenten aan het woord - Adriaan Gijssen (Instruct)	Raspberry Pi Competities- Ernst Brand (PA Consulting) & Krijn Hoogendorp
16.00 - 16.45					Computational Thinking in de Klas - Eljakim Schrijvers
16.45-17.30	Foyer: Napraten met hapje en drankje - bezoek informatiemarkt				

How to teach programming?

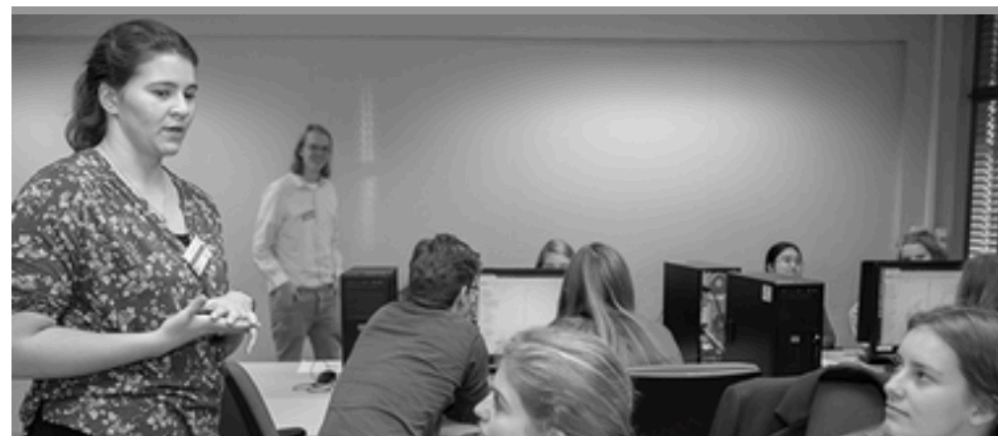
← → ↻ vhto.nl/projecten/computer-science-certificate/ ☆ 🌐 ⋮



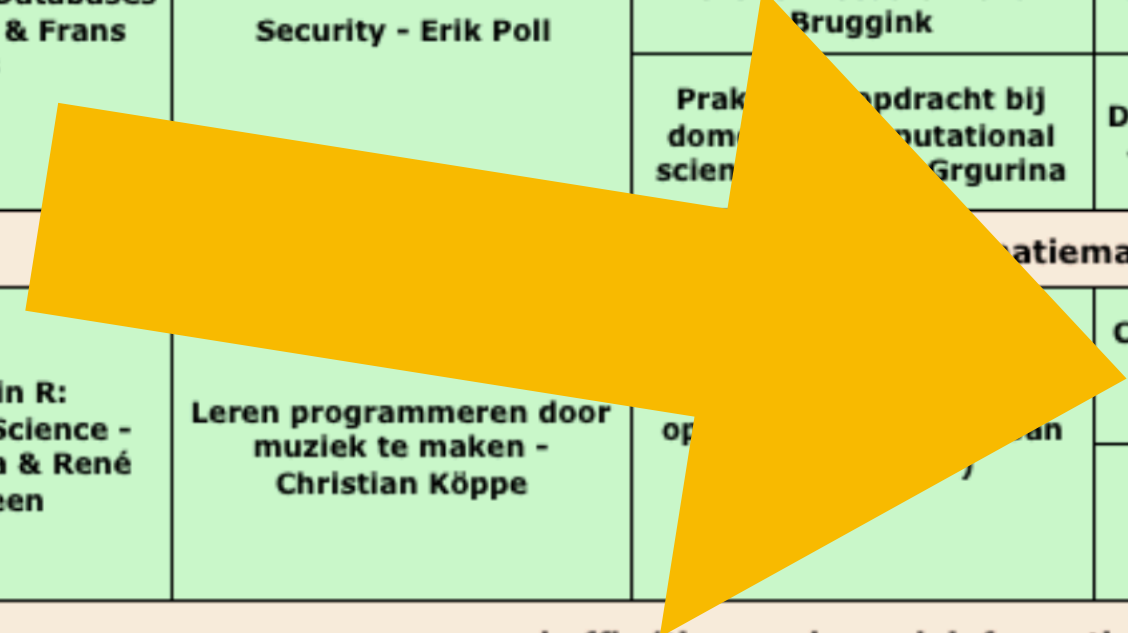
Onderwijs Profes



HOME > PROJECTEN > COMPUTER SCIENCE CERTIFICATE



Programma i&i conferentie - donderdag 7 november 2019					
	BBG-020	BBG-069	BBG-071	BBG-075	BBG-077
09.00 - 10.00	Foyer: Inloop en registratie - koffie/thee en bezoek informatiemarkt				
10.00 - 10.15	Opening in Auditorium Cosmos, Welkom door i&i bestuur				
	Toelichting landelijke nascholing keuzedomeinen door Bestuur en Martin Bruggink				
	Toelichting programma i&i - conferentie				
10.15 - 11.00	Keynote "Effectief leren programmeren" door Felienne Hermans				
11.00 - 11.45	Keuzedomein H: Databases - Eelco Dijkstra & Frans Peeters	Security - Erik Poll	Toetsing van Grondslagen (domein B) en programmeren (domein D - Renske Weeda & Martin Bruggink	Toelichting Digitale Geletterdheid uit Curriculum.nu - Koen Bulters	Codasium; programmeeronderwijs in onder- én bovenbouw - Luc van Leeuwe
11.45 - 12.30			Praktijkopdracht bij domein Computational science - Nataša Grgurina	Digitaal geletterd met DIGIT! - Hafida Makioul (Instruct)	Lessen en lesmaterialen over eindige automaten (domein B: grondslagen) - Philip Braas
12.30 - 13.30	Informatiemarkt				
13.30 - 14.15	Keuzedomein R: Computational Science - Nataša Grgurina & René van der Veen	Leren programmeren door muziek te maken - Christian Köppe	opdracht bij domein Computational science - Nataša Grgurina	Computer Science Certificate - Basisexamen programmeren VO - VHTO	Studententeam Project MARCH (TU Delft) maakt exoskelet
14.15 - 15.00				De didactiek van genderinclusief programmeren - VHTO	Virtuele race met zelfrijdende auto's- Karel Bruneel"
15.00 - 15.15	koffie/thee en bezoek informatiemarkt				
15.15 - 16.00	Keuzedomein J: Programmeerparadigma's / Functioneel programmeren - Kees Hulzing & Hans van den Burg	Rekenen met de Quantumcomputer - CWI - Christian Schaffner	Beauty and Joy of Computing - Casper Kole	Fundament Informatica, docenten aan het woord - Adriaan Gijssens (Instruct)	Raspberry Pi Competities- Ernst Brand (PA Consulting) & Krijn Hoogendorp
16.00 - 16.45					Computational Thinking in de Klas - Eljakim Schrijvers
16.45-17.30	Foyer: Napraten met hapje en drankje - bezoek informatiemarkt				



How to teach programming?

← → ↻ ⓘ Not Secure | pythonindeklas.nl



Waardes zoeken

1) In Python kun je woorden zonder aanhalingstekens gebruiken. Dat zijn variabelen. Als Python die tegenkomt gaat hij omhoog zoeken in het programma naar de *definitie* van de variabele. Dat betekent waar de variabele wordt ingesteld met een is-teken:

```
naam = 'Feliene'
```

Schrijf deze codes over. Zet een pijltje tussen het gebruik van een variabele en zijn definitie.

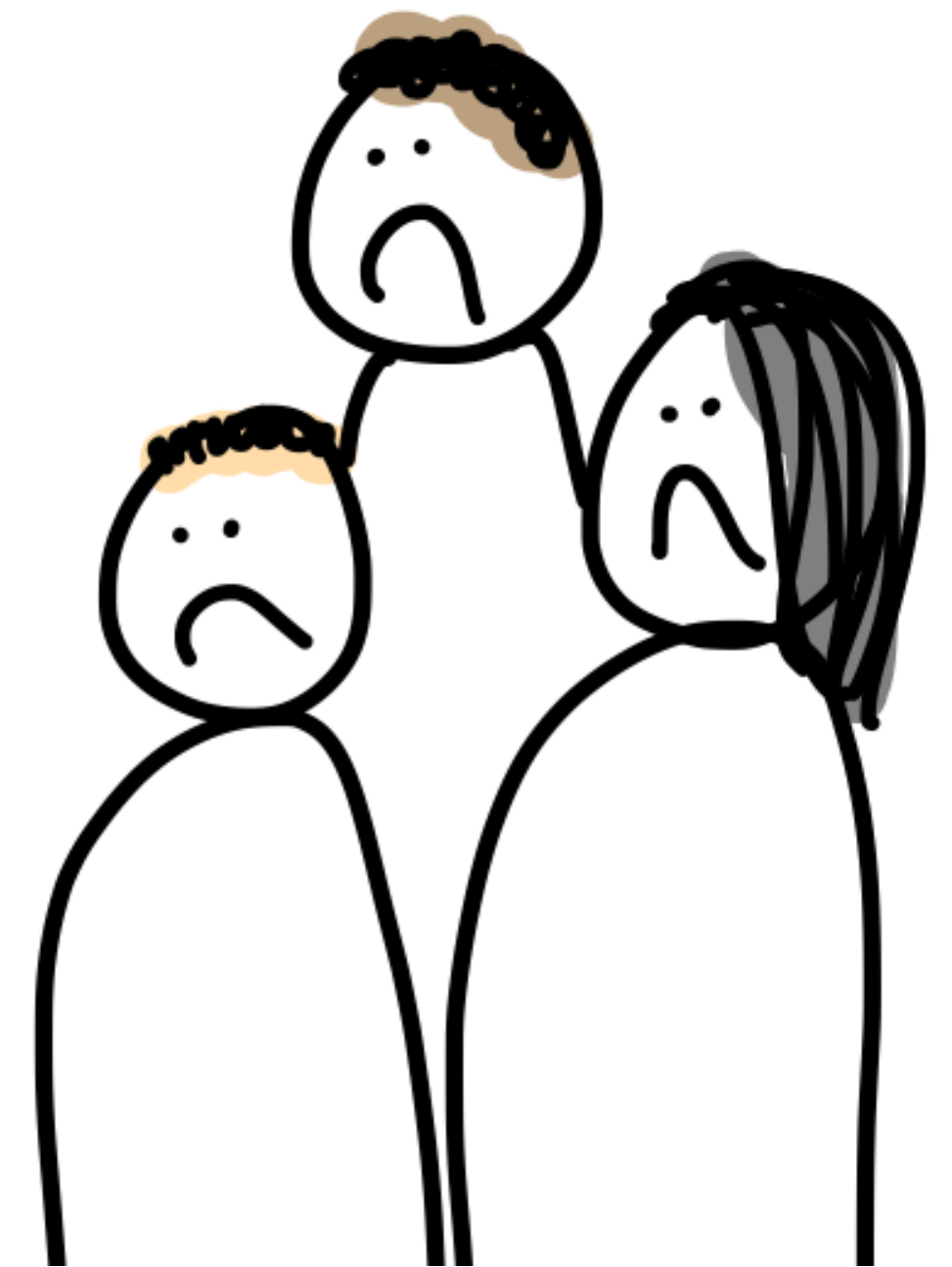
Voorbeeld: `klas = 'c'`
`print('Hallo', klas)`

```
1. les = 'Coderen'  
   print('welkom', 'bij', les)
```

```
2. klas = '1c'  
   print('Hallo', klas)
```

How to teach programming?

That doesn't
seem fun!

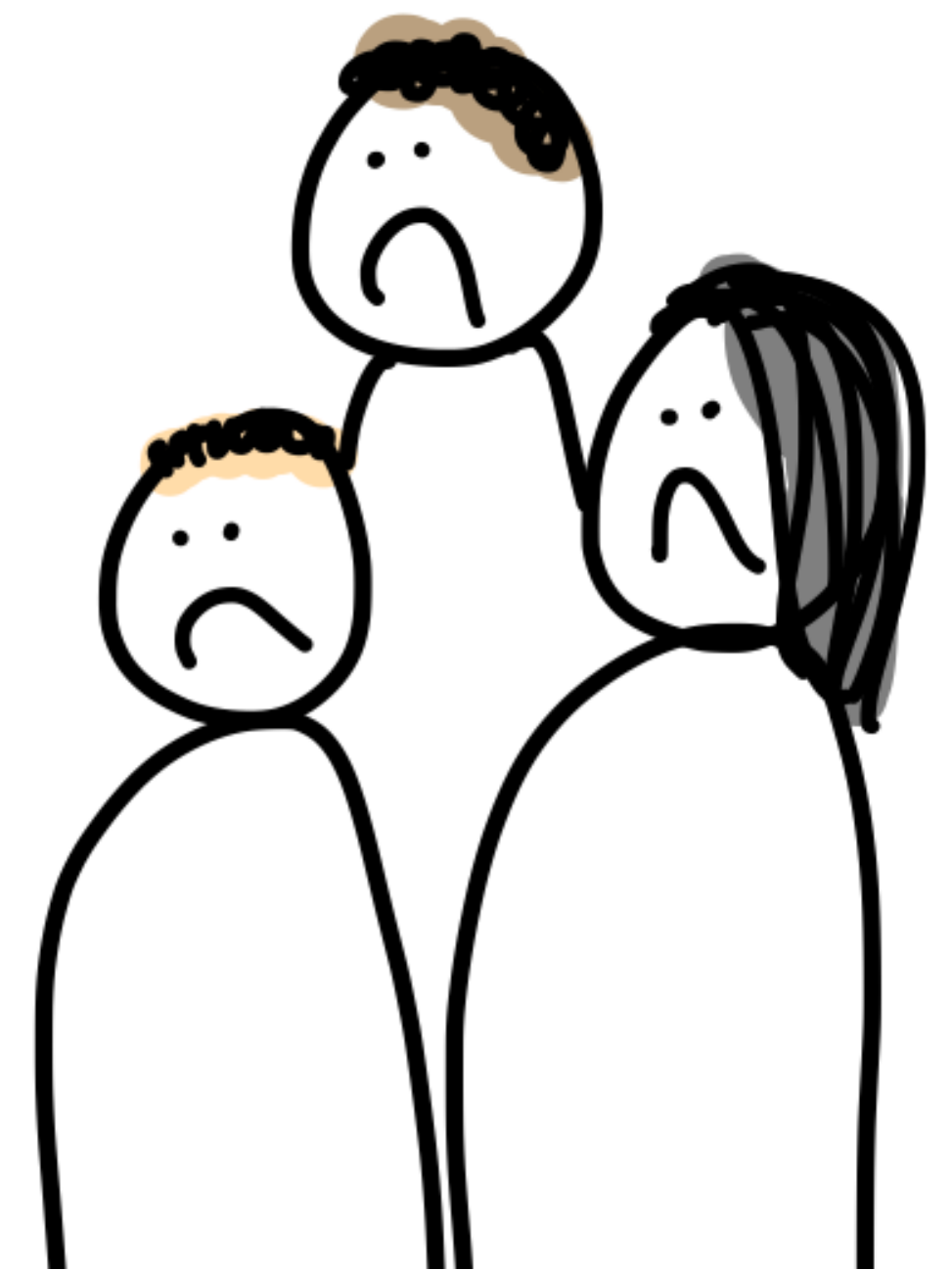


How to teach programming?

That doesn't
matter!



That doesn't
seem fun!



How to teach programming?

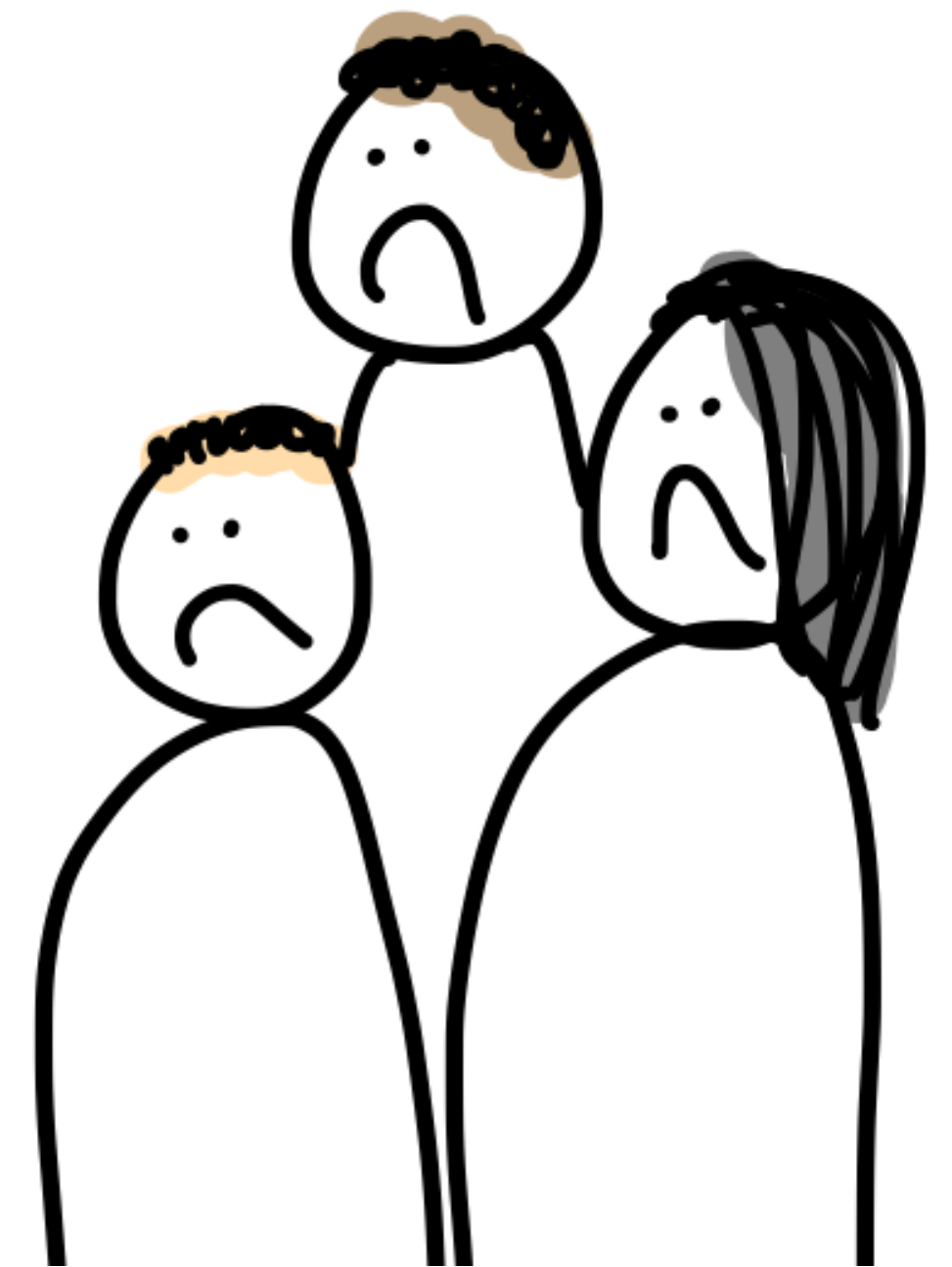
That doesn't matter!

That doesn't seem fun!



Interest

⇒ skill



How to teach programming?

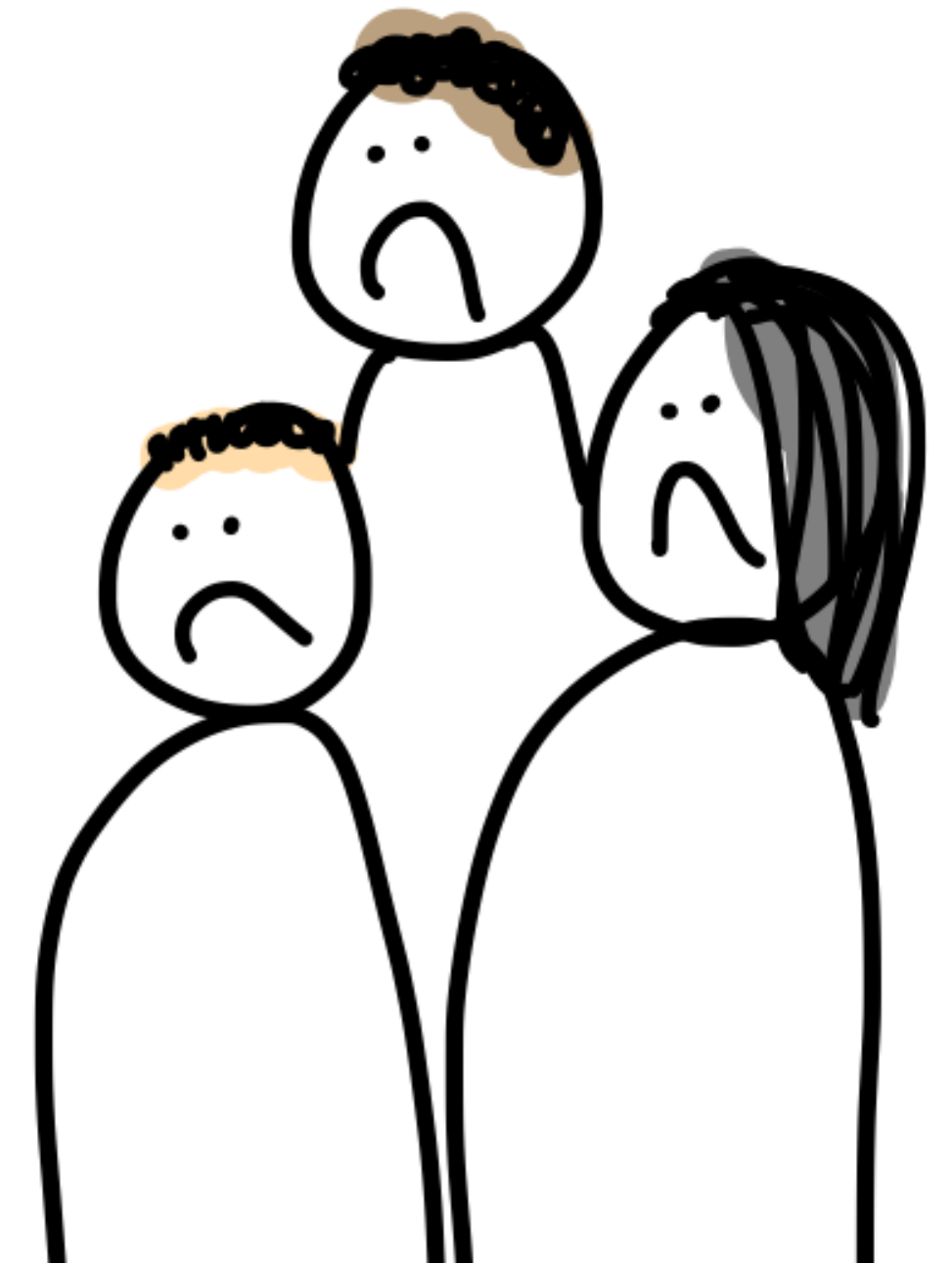
That doesn't matter!

That doesn't seem fun!



Interest

~~⇒~~ skill



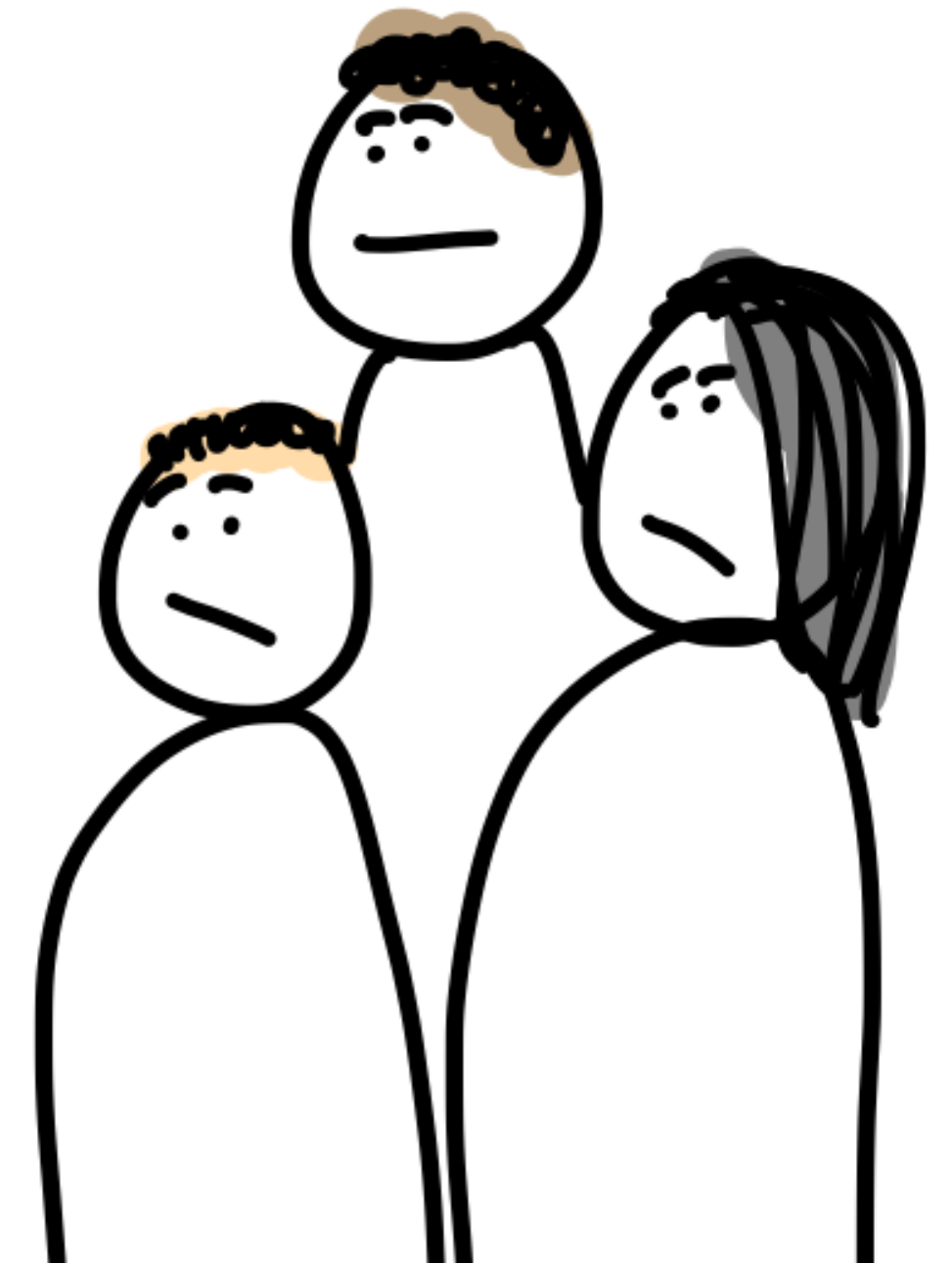
How to teach programming?

That doesn't matter!

That doesn't seem fun!



Interest \Leftarrow skill

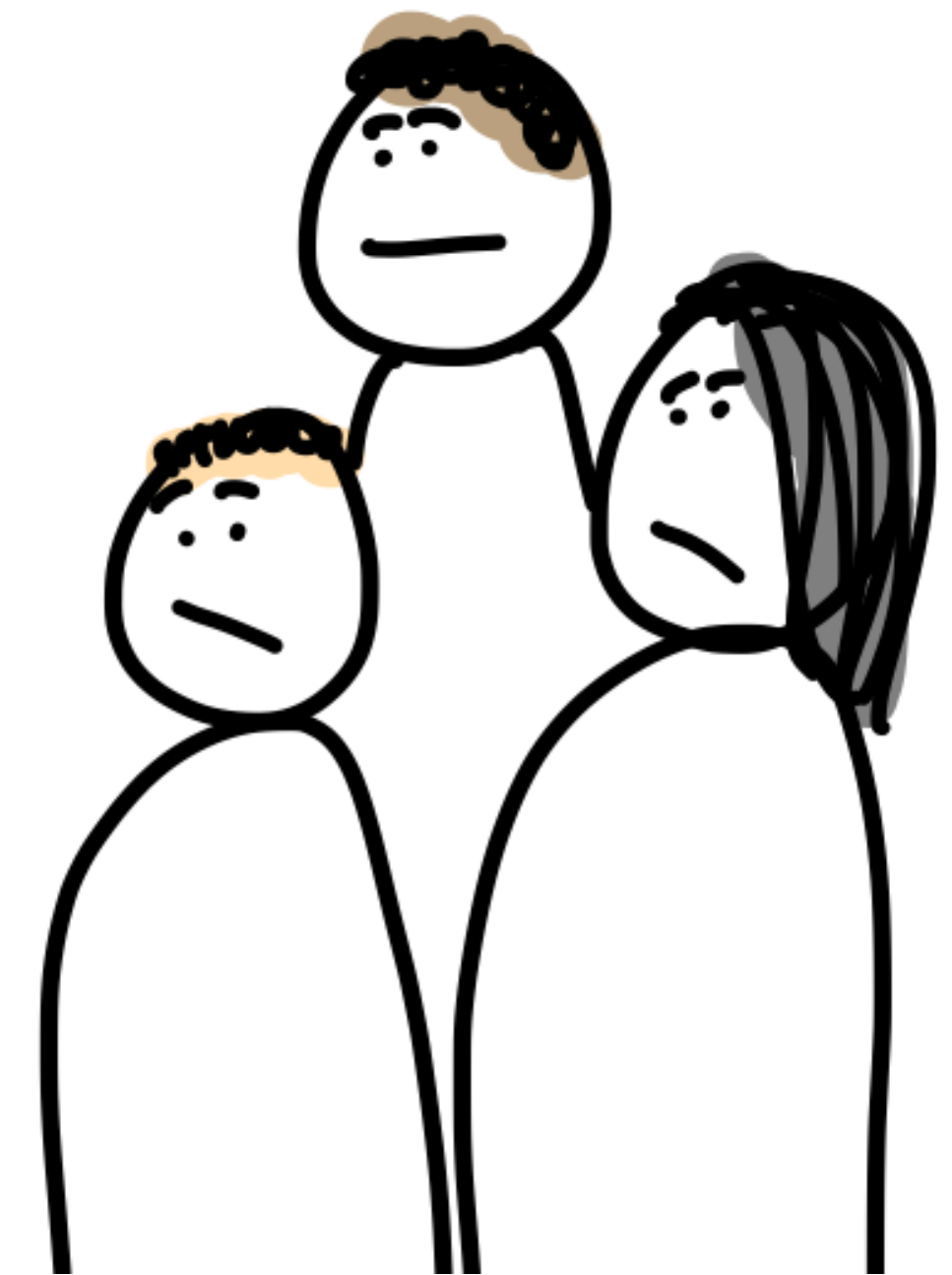


How to teach programming?

The role of interest
in learning

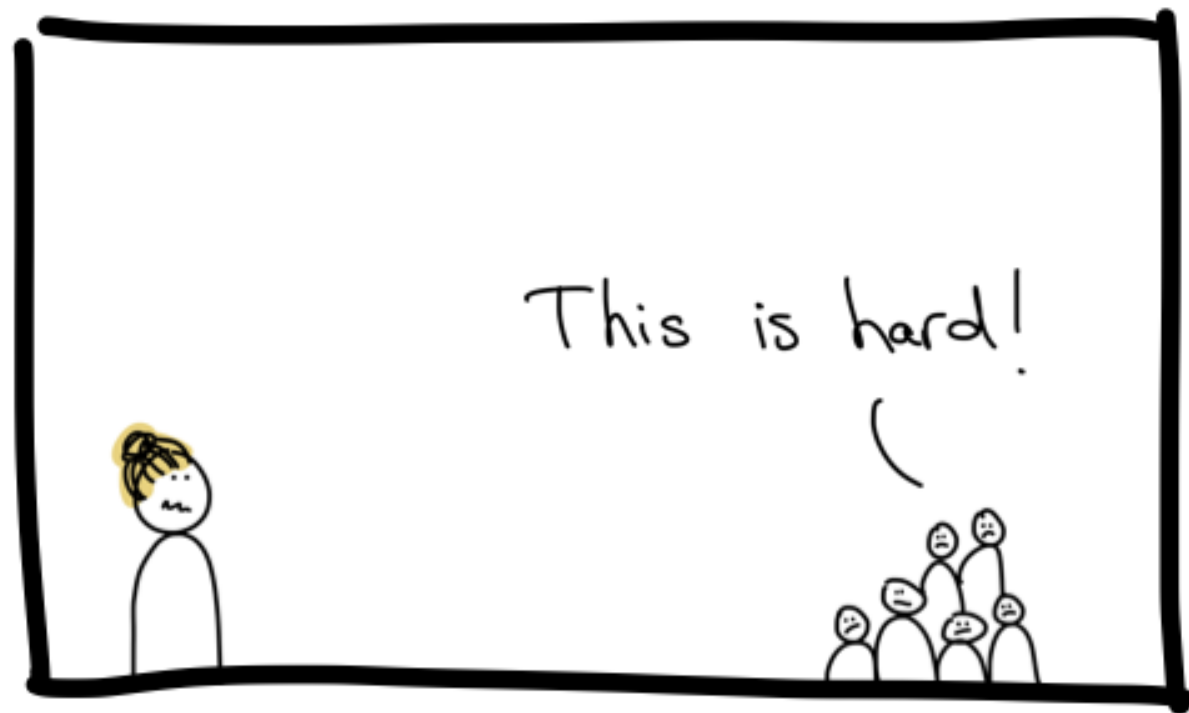
Rotgans & Schmidt '17

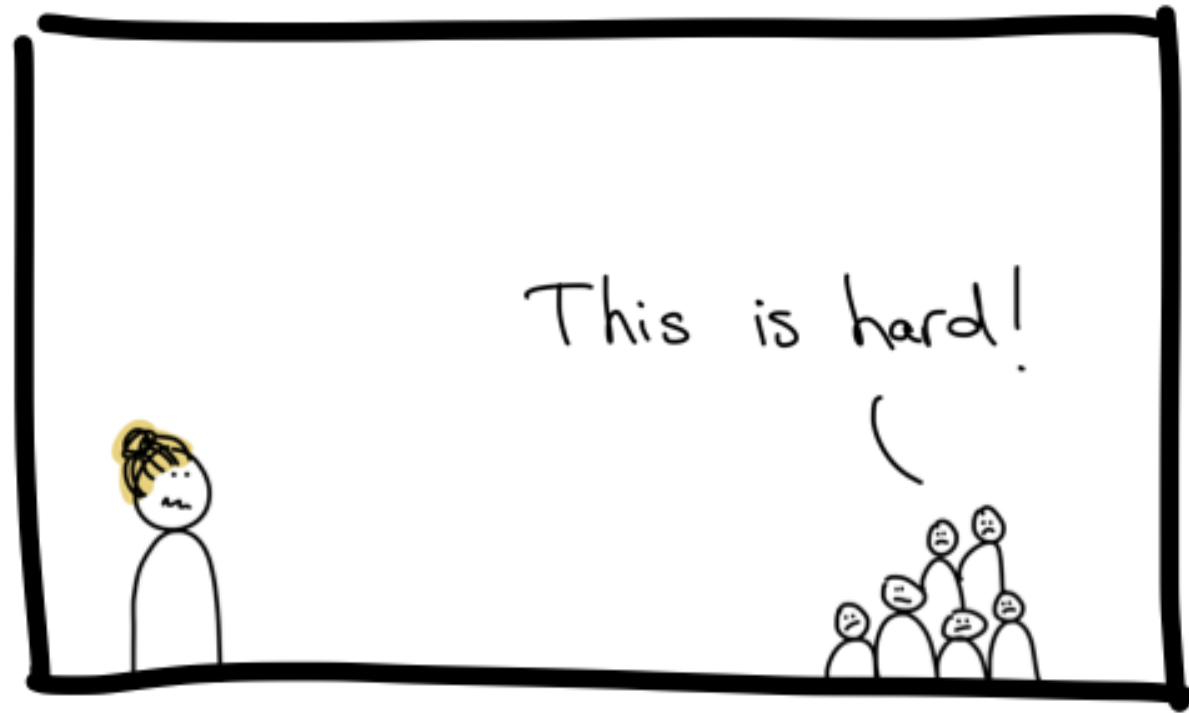
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

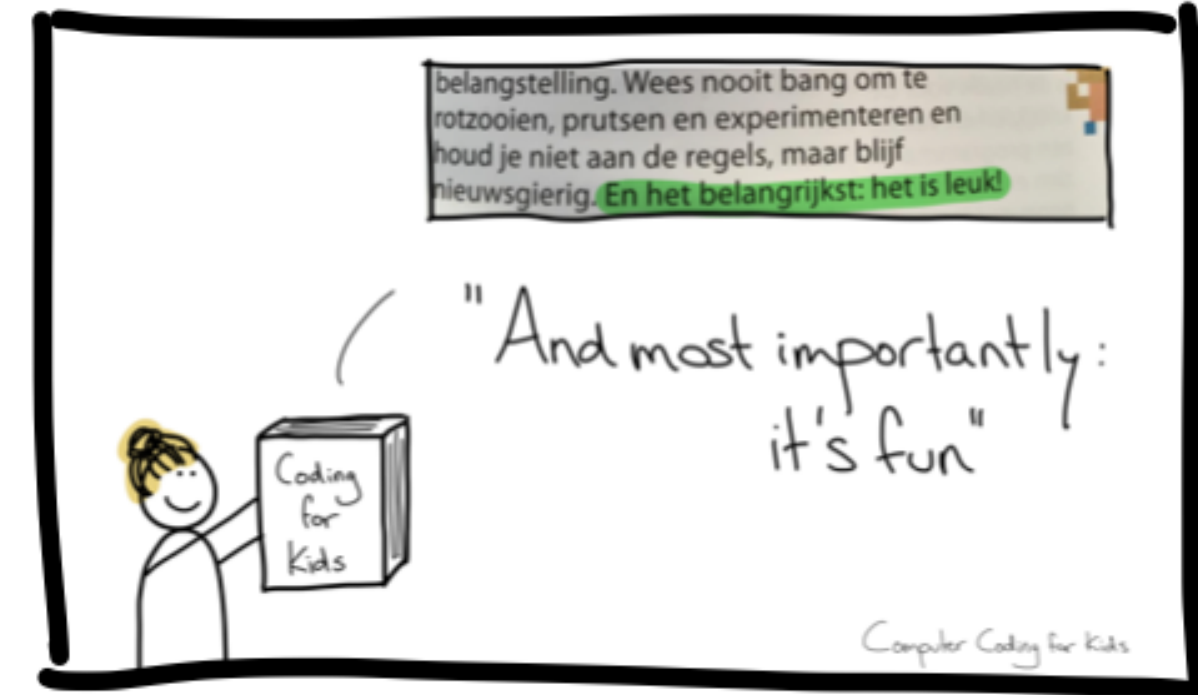
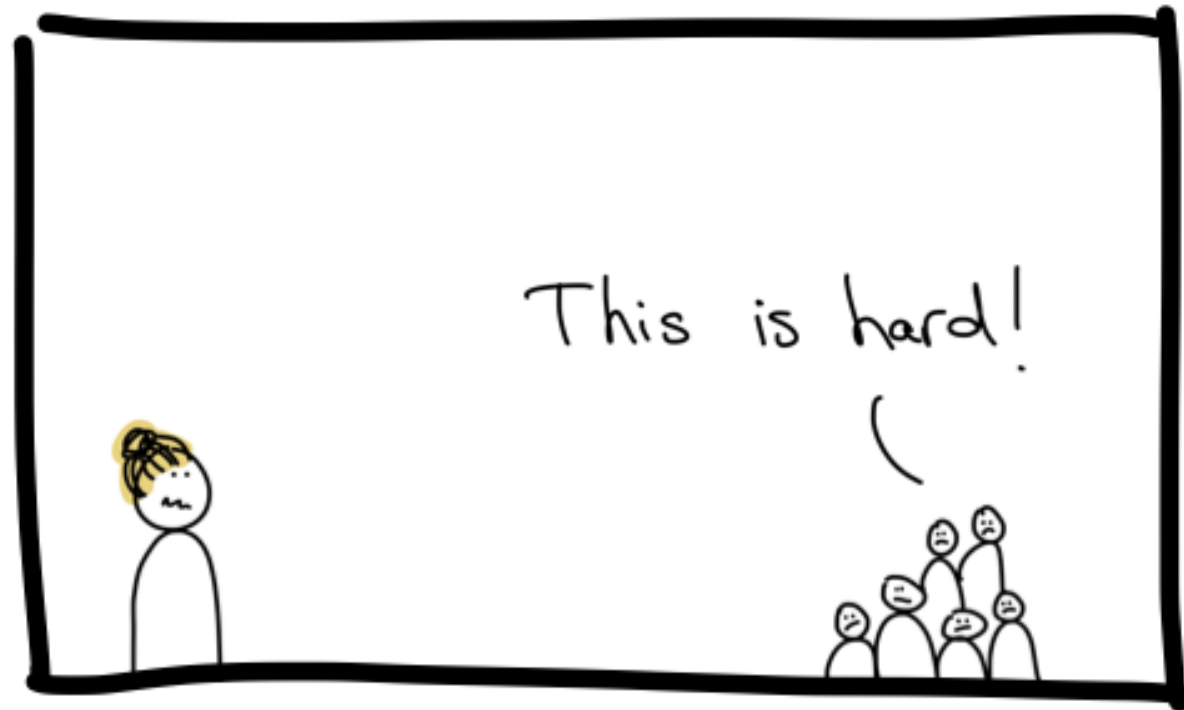


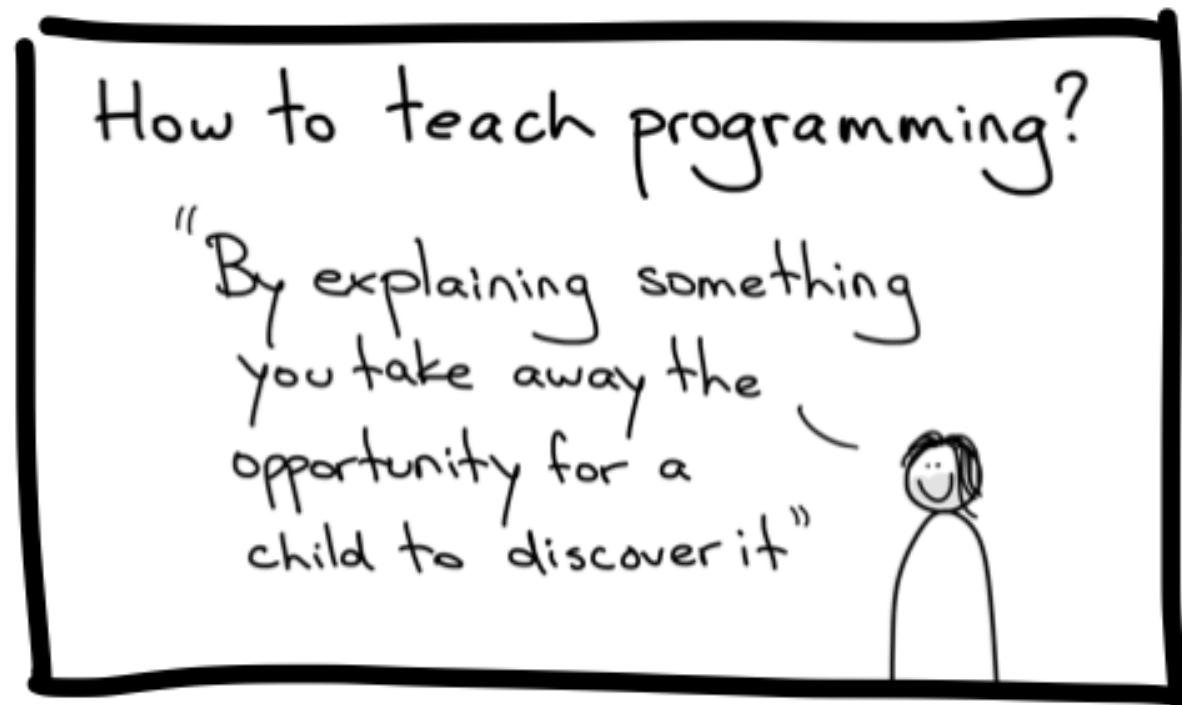
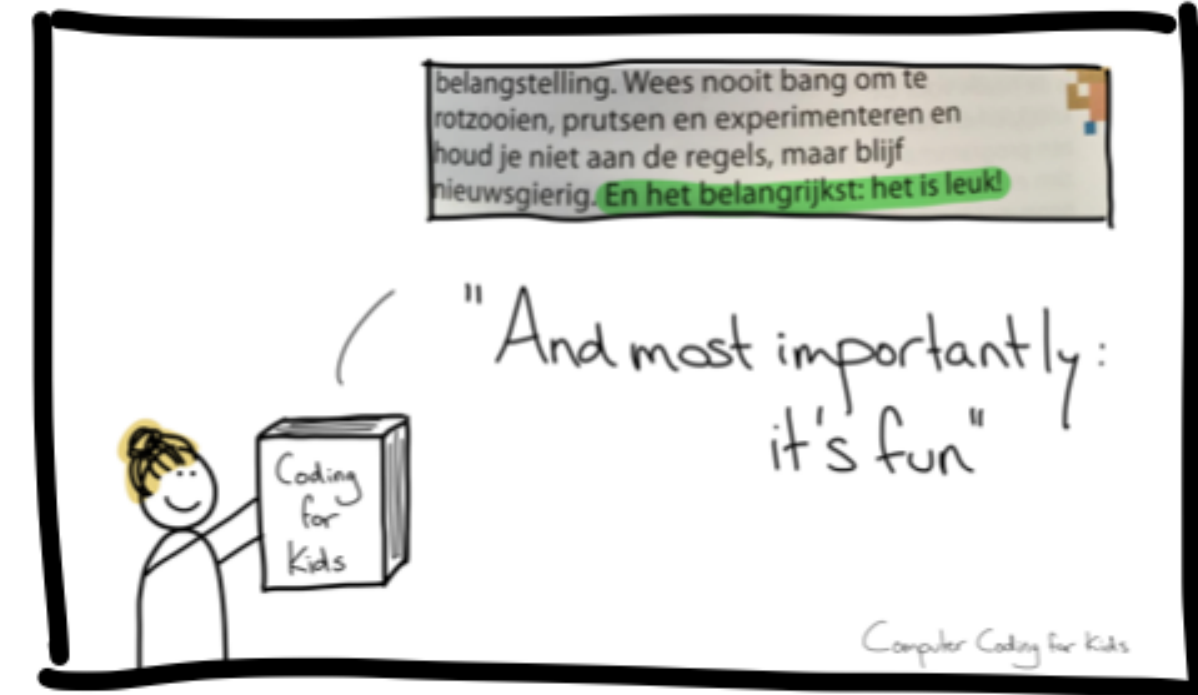
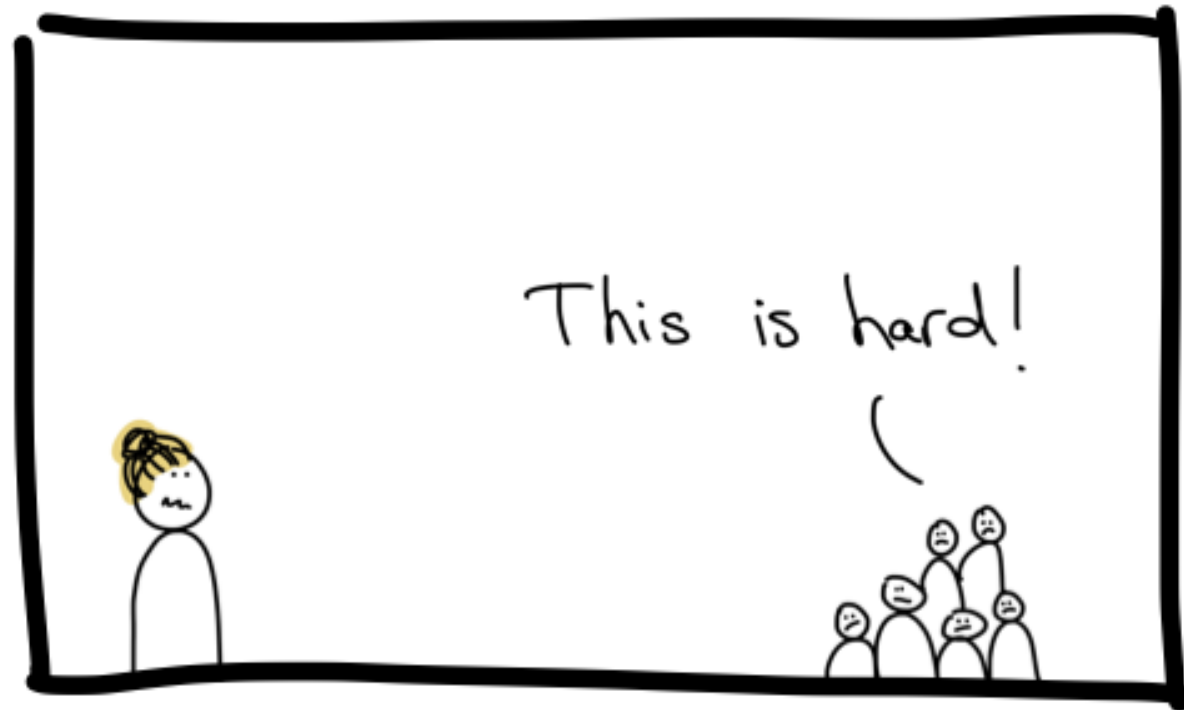
THE END

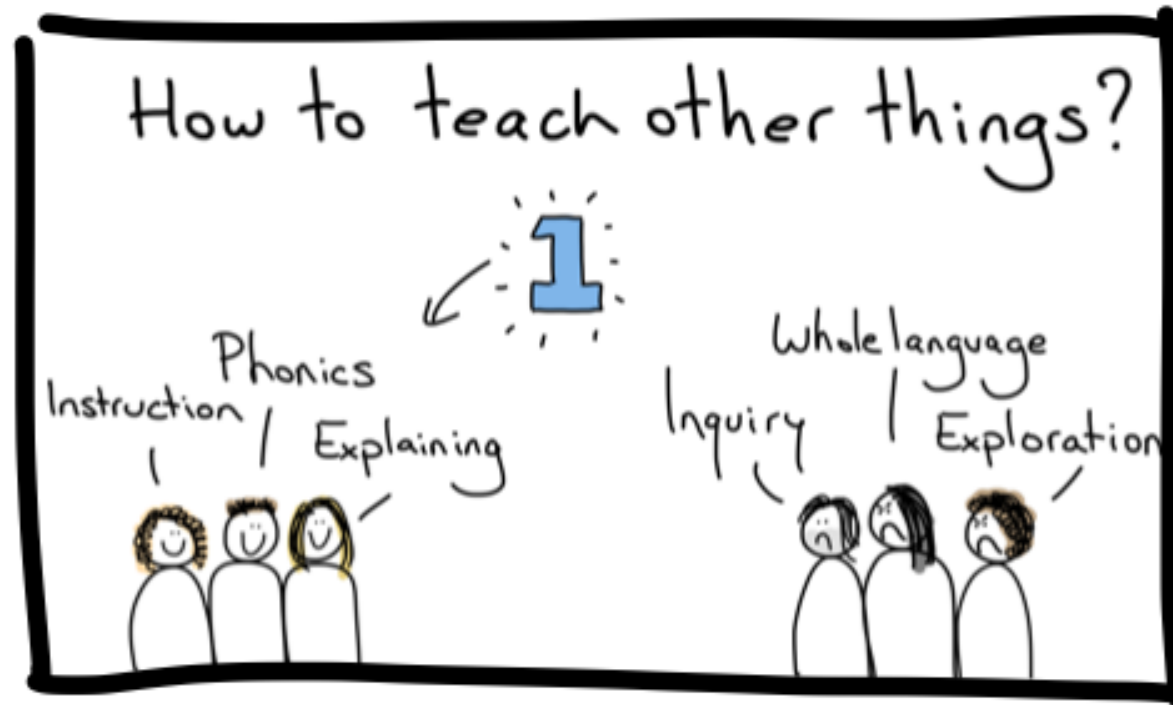
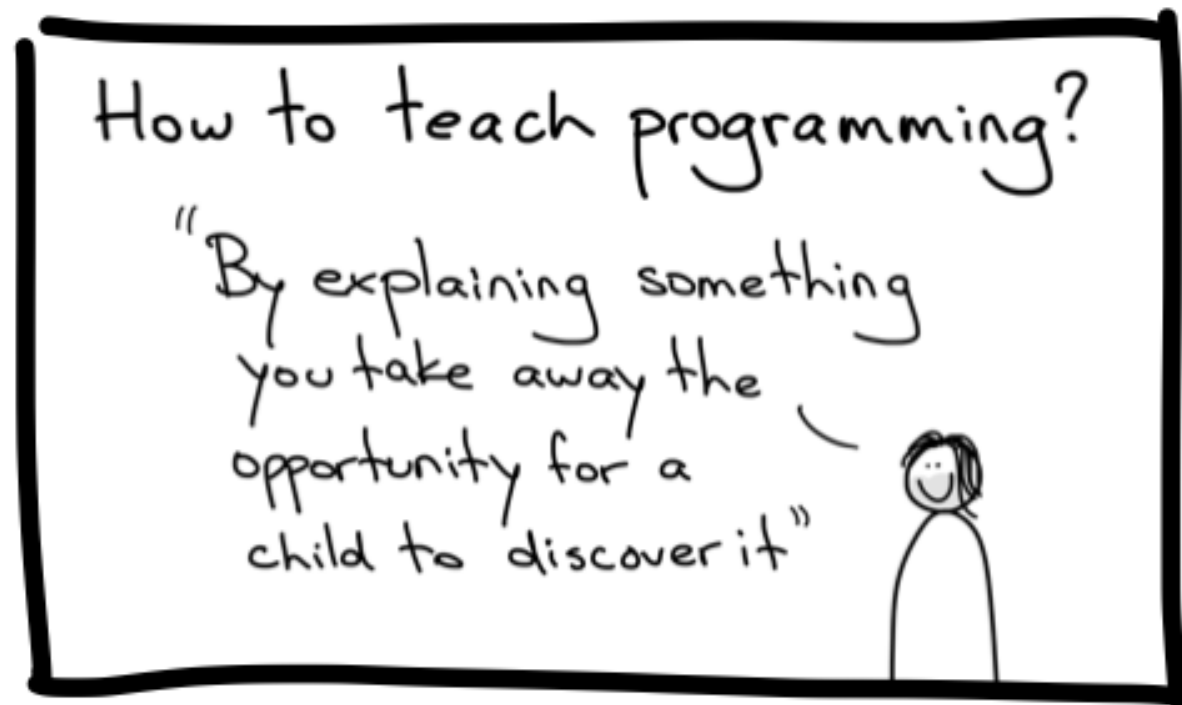
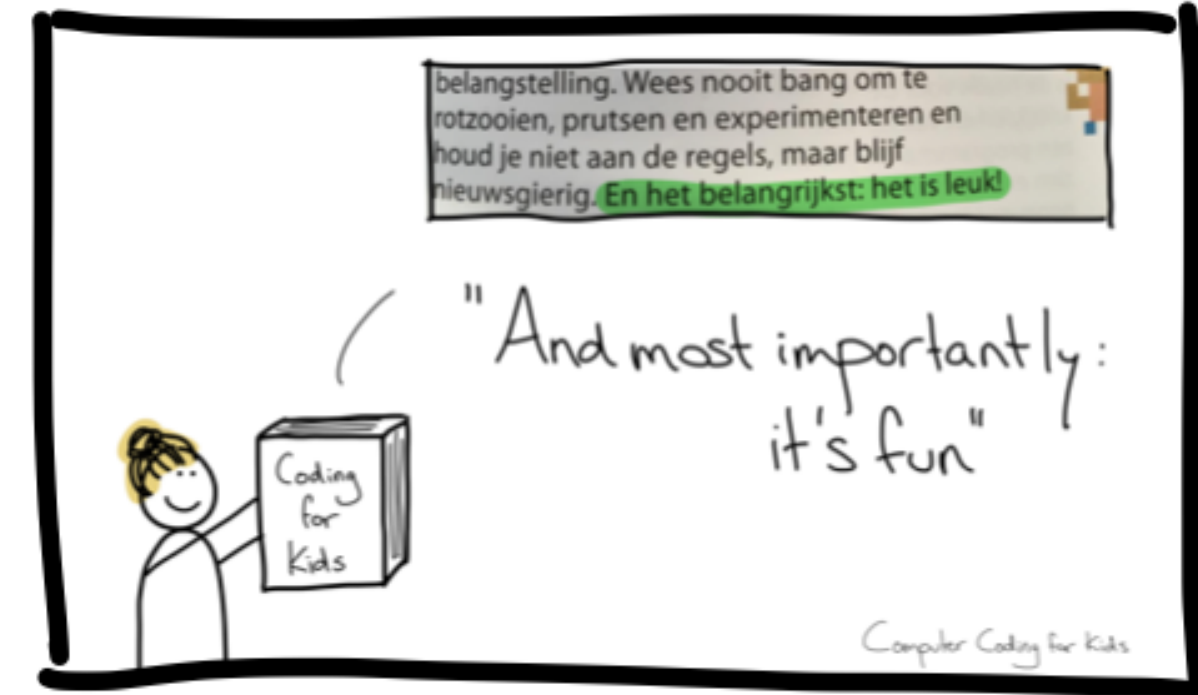
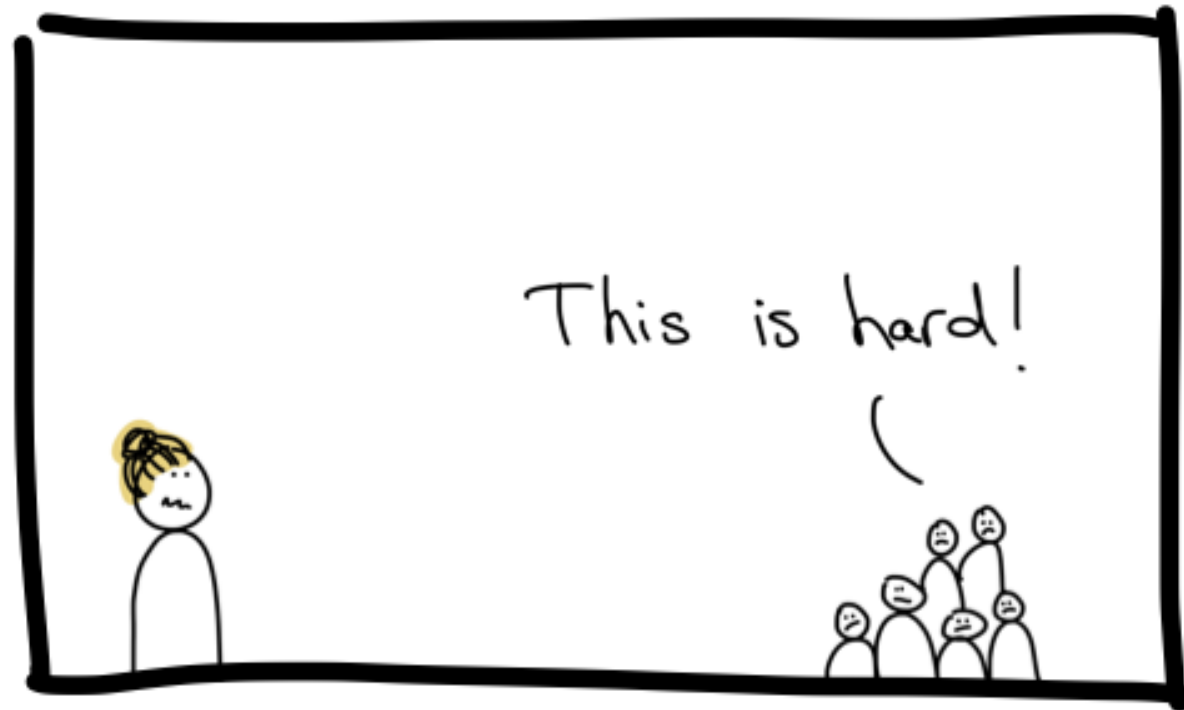


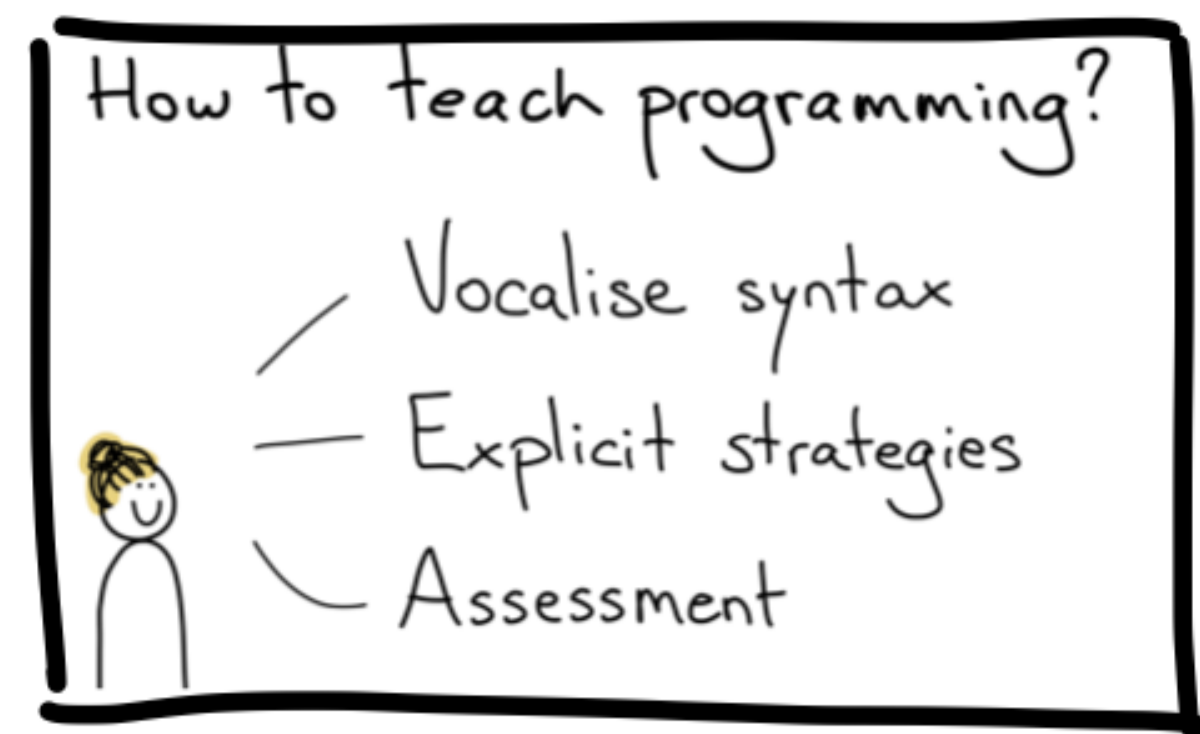
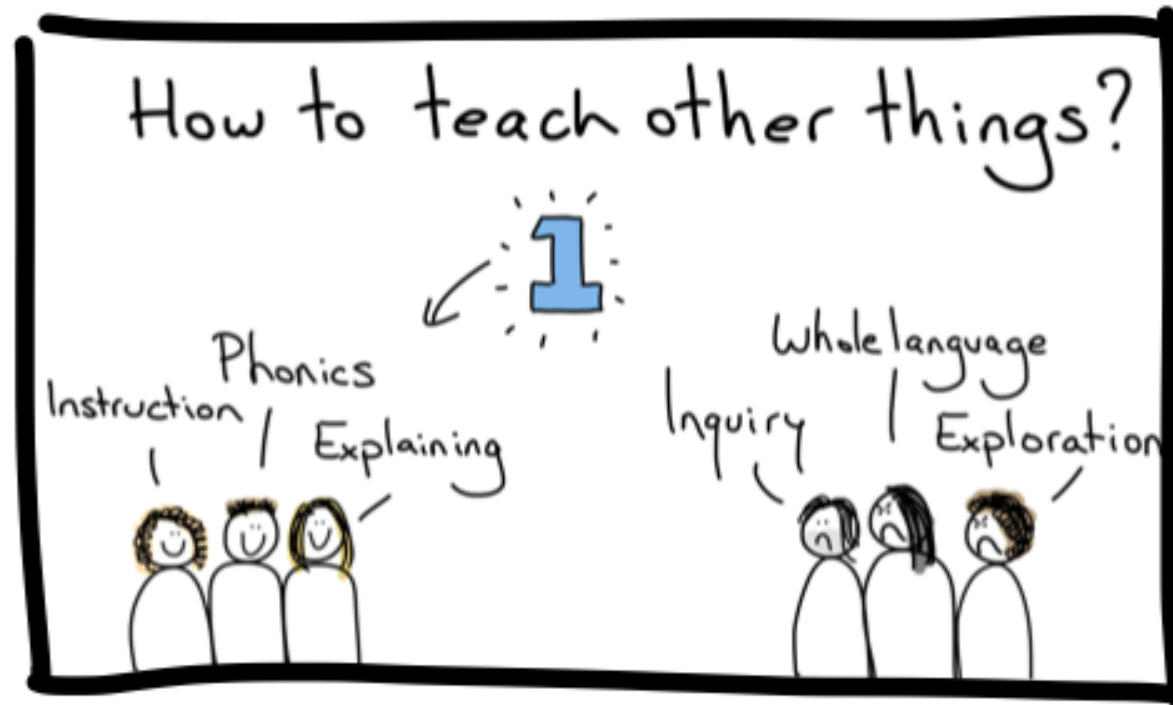
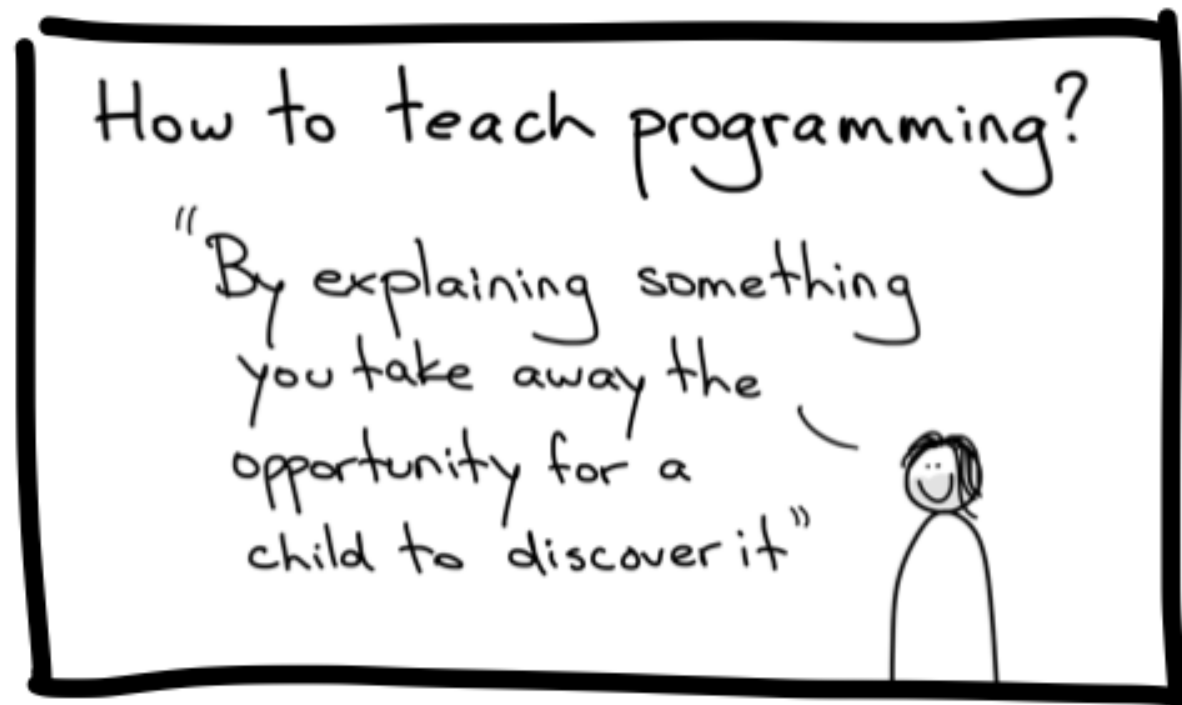
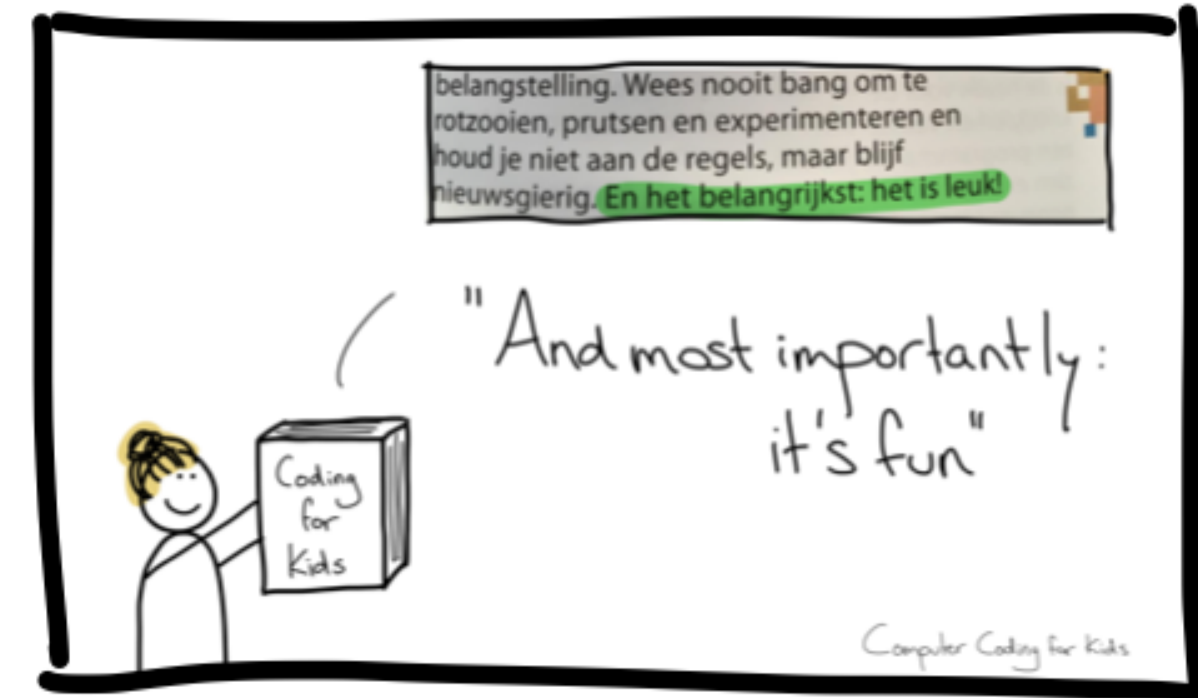
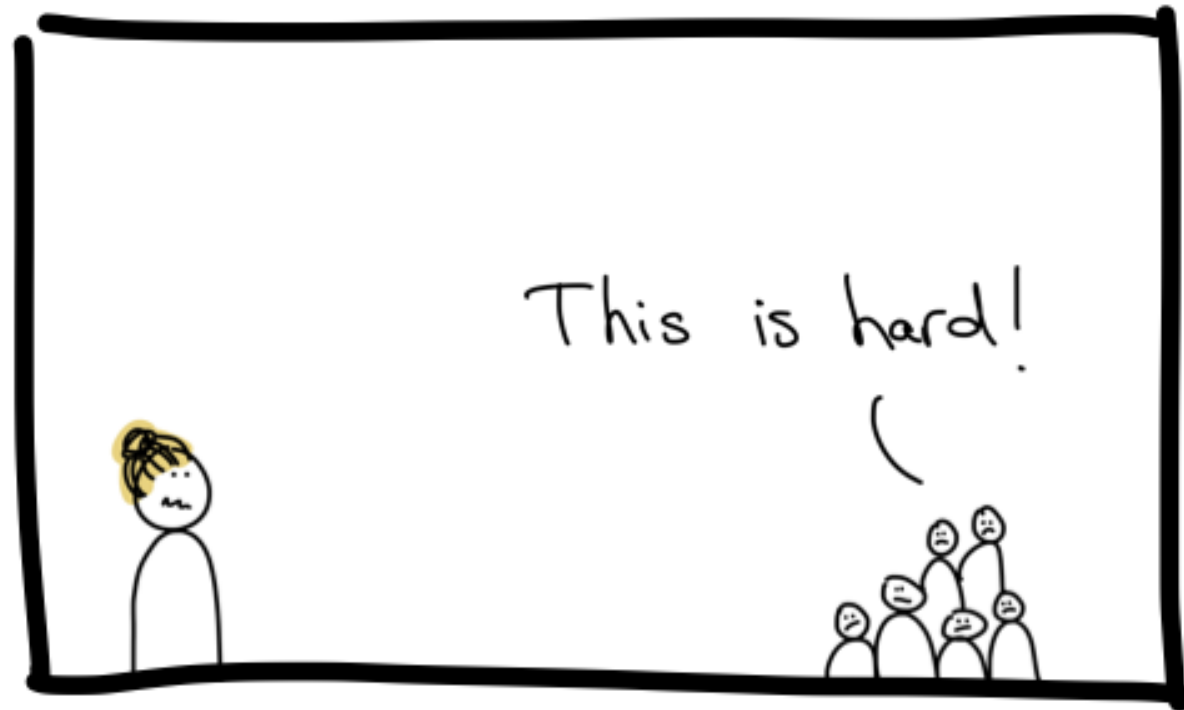




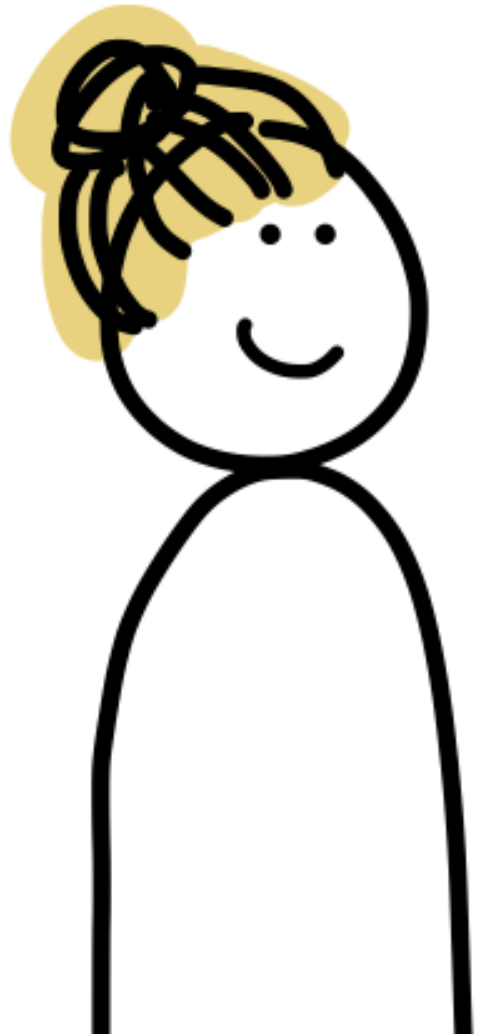








THE END



THE END

@Felicienne 📄
felicienne.com




THE END

@Felienne 📄
felienne.com

pythonindeklas.nl



THE END


I host  SE radio

@Felienne 
felienne.com

pythonindeklas.nl



THE END

I host 
SE radio


bit.ly/LizzysAfstuderen


@Feliene 
feliene.com

pythonindeklas.nl



THE END

I host  SE radio

Made with  Goodnotes

@Feliene 
feliene.com

bit.ly/LizzysAfstuderen

pythonindeklas.nl

