LEARNING TEXT-BASED PROGRAMMING WITH HEDY

Moving from a block- to text-based programming language can be a huge leap for children. That's why **Felienne Hermans** created Hedy, a language that gradually introduces the rules of syntax to young coders

Block-based languages like Scratch are a great way to help students take their first steps in programming, because they do not have syntax, the precise way in which commands have to be formulated. Students can simply click blocks together and create programs that work straight away.

FELIENNE HERMANS
Felienne Hermans is an associate professor at Leiden University and also teaches ninth grade computer science. She is the creator of the Hedy programming language, which helps children take their first steps in textual programming (**@felienne**).

Blocks are for kids

Blocks are certainly great, but when teaching students in seventh grade and up, I find that they start to pull away from Scratch. They begin to see it as a toy; as a thing for elementary school kids. This isn't necessarily true, but around that age students tend to get excited to learn text-based languages, like Python or JavaScript, because they see them as more mature, more powerful, and more real.

Textual languages are hard

The step from Scratch to Python, though, is quite big. Something relatively simple, such as counting to 10, requires a child to type:

```
for i in range(1, 11):
    print(i)
```

That is a lot of syntax to remember, and a lot of ways to mess up. Misplace a space and Python will crash with a cryptic message called IndentationError. I found that although kids were motivated to learn textual languages, this could cause frustration. And that is not surprising! Recent research on university-level students learning programming showed that even good students make syntax errors in 50 percent of programs. Knowing that, it is not surprising that our middle-schoolers have trouble getting their Python programs to work correctly. I knew I had to come up with a better solution for them.

English is taught gradually

When I saw the struggles of middle-schoolers learning Python syntax, I started to dive into how children learn to read and write, because when learning a language, children also have to learn syntax rules.

When children learn the syntax rules of natural languages such as English, learning is done gradually. Initially, children just use lower-case letters to make sentences, and that is totally OK! Writing itself is hard enough. New rules are then added step by step: first, children learn that sentences start with upper-case letters, followed by the rule that sentences end in a full stop. Each syntax addition is simple enough, so each step is small, and each new addition gives them new powers. For example, once you know that the full stop separates sentences, you can start to make sentences that span multiple lines!

Hedy: a gradual programming language

Inspired by the way we teach English in elementary school, I created Hedy, a gradual programming language. Hedy consists of a number of levels, each with a few new commands. In the first level, for example, you can simply print something to the screen with 'print':

```
print hello everyone!
```

Hedy's print statement at level one is a lot simpler than Python, which would require students to enclose the text in both round brackets and quotation marks. Step by step, programming concepts are introduced. Level four, for example, introduces loops, but with a simpler syntax using 'repeat':

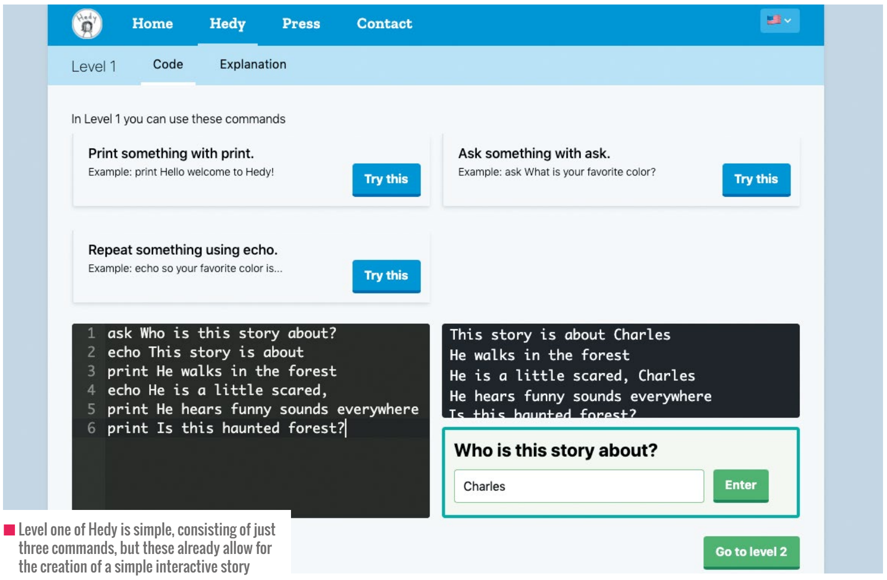
```
repeat 10 times print hello!
```

Each level is a lesson

Hedy is available for free at **hedycode.com** and can be used in a browser, like Scratch. Currently, Hedy has seven levels available, each lesson having new commands and lesson materials with instructions and assignments built in. Each level should keep children occupied for about one 45-minute lesson. Children can optionally login and save their programs.

Interaction at each level

Even though Hedy starts simple, each level allows students to make real and engaging projects. For example, level one not only has a 'print' statement, it also supports an 'ask' command, allowing the user to provide input, and an 'echo' command that can repeat the input back to the screen. These three commands enable students to create a simple, interactive story, as shown in the image above. Being able to make something real and fun straight away also helps students to get a sense of where programming could take them. This is especially relevant for children who did not use Scratch before Hedy, and thus might not know why programming is fun!



In Level 1 you can use these commands

Print something with print.
Example: print Hello welcome to Hedy!

Ask something with ask.
Example: ask What is your favorite color?

Repeat something using echo.
Example: echo so your favorite color is...

```
ask Who is this story about?
echo This story is about
print He walks in the forest
echo He is a little scared,
print He hears funny sounds everywhere
print Is this haunted forest?
```

```
This story is about Charles
He walks in the forest
He is a little scared, Charles
He hears funny sounds everywhere
Is this haunted forest?
```

Who is this story about?
Charles

Level one of Hedy is simple, consisting of just three commands, but these already allow for the creation of a simple interactive story

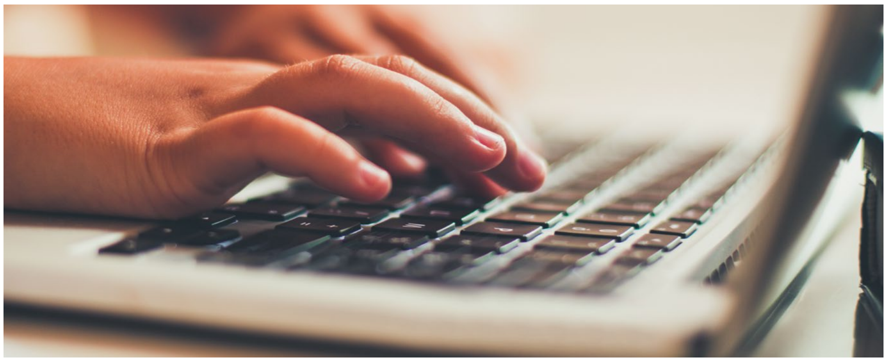
> BEING ABLE TO MAKE SOMETHING REAL AND FUN STRAIGHT AWAY HELPS STUDENTS SEE WHERE PROGRAMMING CAN TAKE THEM

Explaining why we have syntax

The small steps of Hedy are helpful, but another benefit is that Hedy helps to explain to kids why syntax is even in programming. When I taught Python to middle-schoolers, they would often ask me why, for example, the quotation marks are needed. Usually at that level I would say: "They just have to be there", which I found to be a weak answer, but of course students lack the knowledge to really understand. In Hedy, we first show learners the language without quotation marks, helping them so see the value. In level two, for example, variables are printed automatically, like this:

```
name is Hedy
print hello Hedy
```

Being allowed to print variables and text together is easy, but introduces a problem! If you want to print "your name is Hedy" you cannot do it. Typing `print your name is name` will result in "your Hedy is Hedy". This problem is solved in level three, when quotation marks are introduced. This step helps students see why quotation marks are helpful, which is a powerful realisation.



Better error messages

One of the big struggles when using a textual language is the error messages. When you make an error in a print statement in Python, you might be confronted with cryptic messages like 'SyntaxError: unexpected EOL'. Such messages can be discouraging for new programmers, because they are hard to

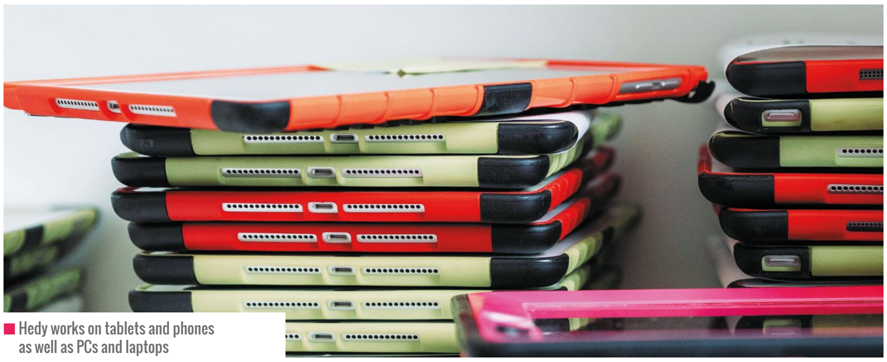
understand, and also not actionable — how should we go about removing this unexpected EOL? As Hedy has a simple syntax, it is also easier to guess what the child meant. For example, if you misspell 'print' in Hedy, the error message will read:

```
pinrt is not a command in Hedy
level 1, did you mean print?
```

The future of Hedy

Hedy is free to use, and over the last nine months more than 200,000 programs have been created. Hedy can be used for a diverse range of programs, and we have seen children who really go wild with creating 100-line programs in a level before advancing to the next. We see this as a strength, because it can take a while before you really get the hang of syntax. So a lot of practice at a lower level will contribute to your programming skill later on.

There is, of course, more work to be done. We are now available in English, Dutch, French, Portuguese, and Spanish, but we'd love to add more natural languages so that we can reach more children. We also want to explore adaptive levelling. In the current version, children can choose to move on a level, which they sometimes do after just one program, because they are curious or ambitious. It would be better to move them on automatically after they have shown enough skill, or maybe even move them back when we see them struggle. We are excited to see what young people create next with Hedy at **hedycode.com**! (HW)


Hedy works on tablets and phones as well as PCs and laptops

A TEACHER'S EXPERIENCE WITH HEDY

Ramon Moorlag, a computer science teacher in the Netherlands, explains how his students have used Hedy:

"Hedy is fun to use. Taking small steps, focusing on a single concept and not being distracted by conventions and/or typos helps a lot.

"My seventh graders enjoyed Hedy. After explaining the lesson concept and practising it, I ask students how to use the concept in a free form. They have built all sorts of things because it helps students create interaction from the start. For example, I have seen students build knock knock jokes, jinxes, riddles, and interactive Christmas wish lists.

"So it's not only a way of bridging the gap; it's also a great way to engage students in textual programming.

"After running the Hedy course for the first time, I found that they had a lot more confidence when we started with some simple HTML and web tutorials."