



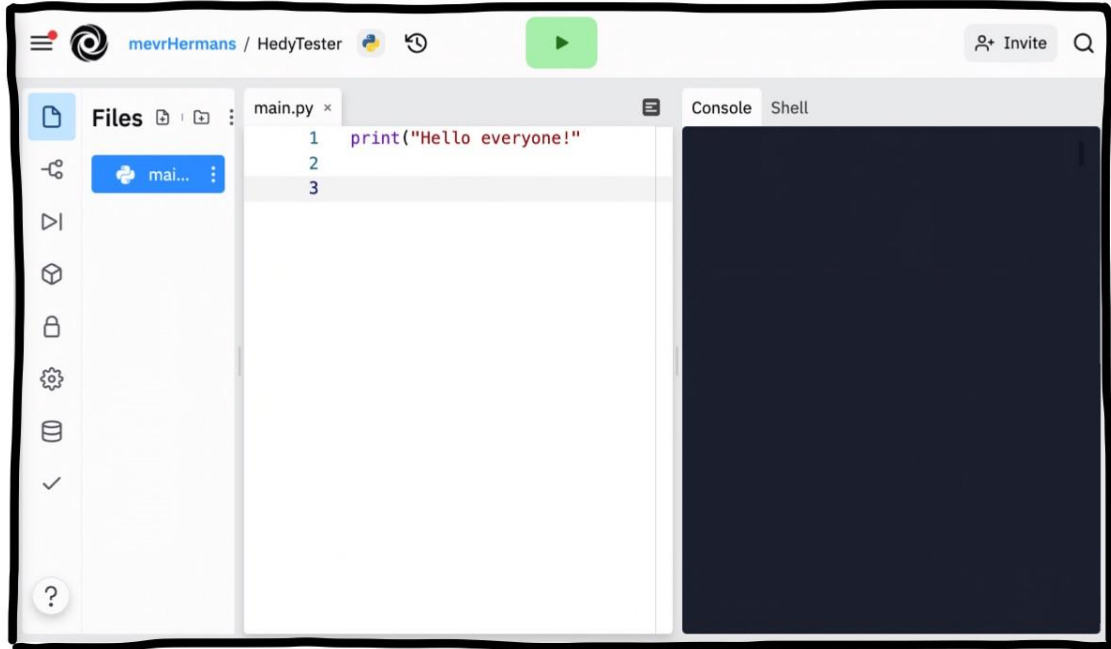
**Open Universiteit**

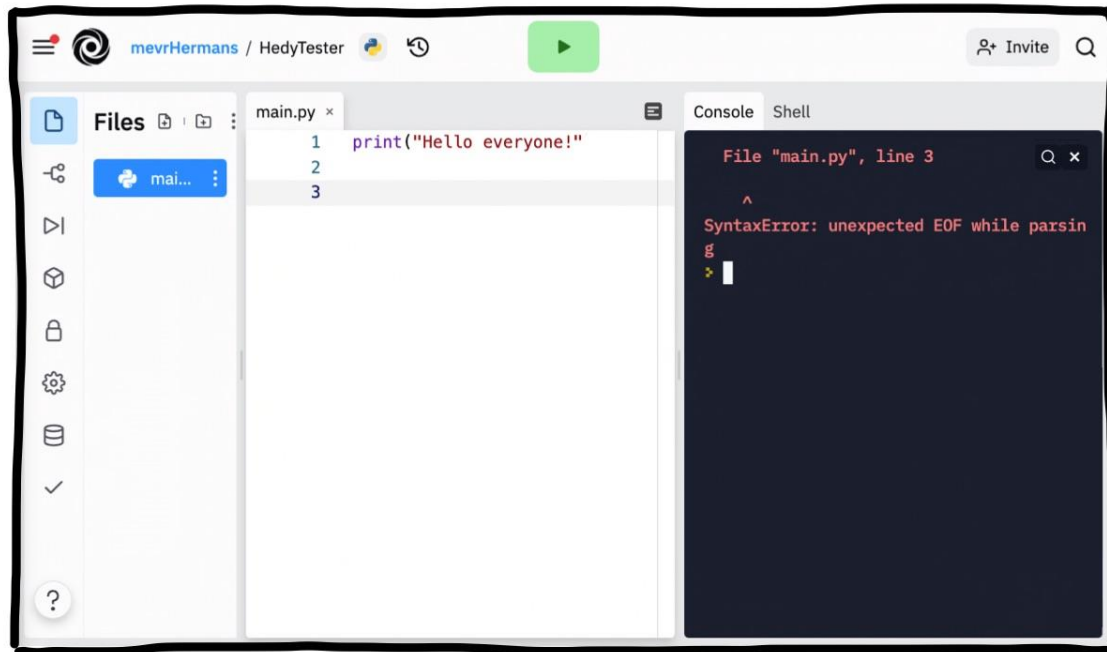
# **Hedy in Arabic: Towards a localized programming platform**

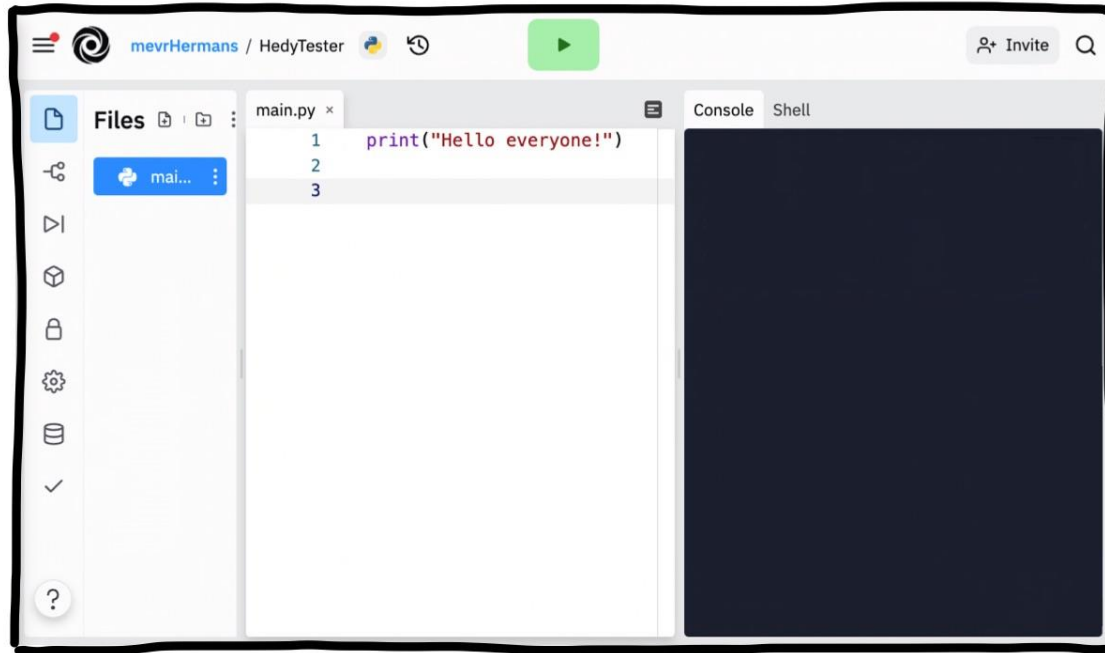


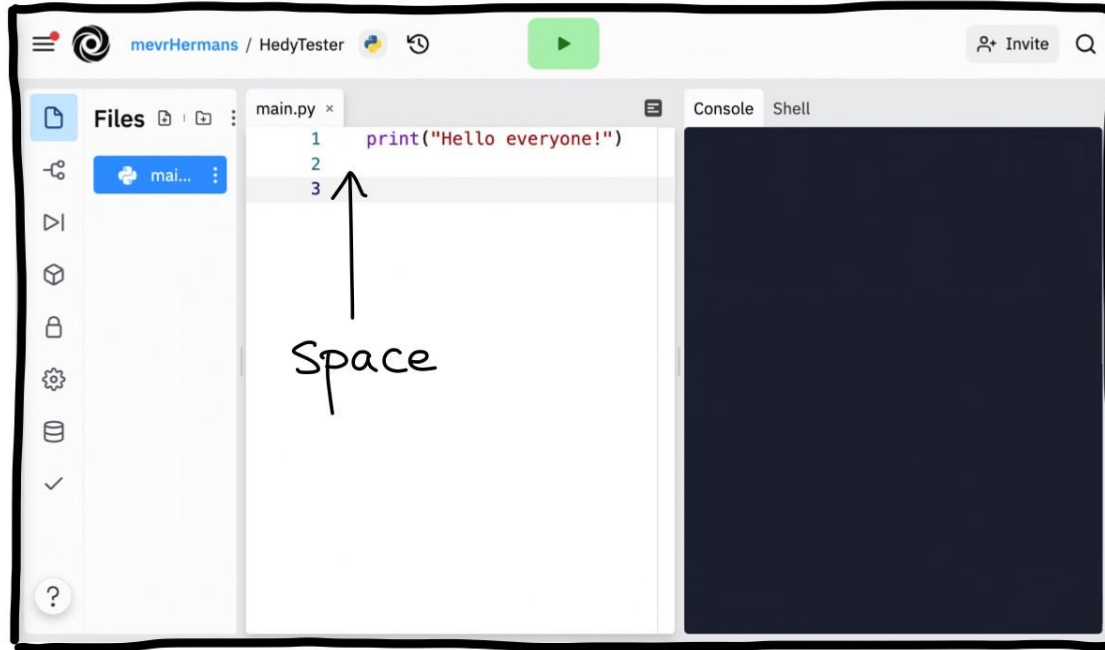
# Hedy: A gradual programming language

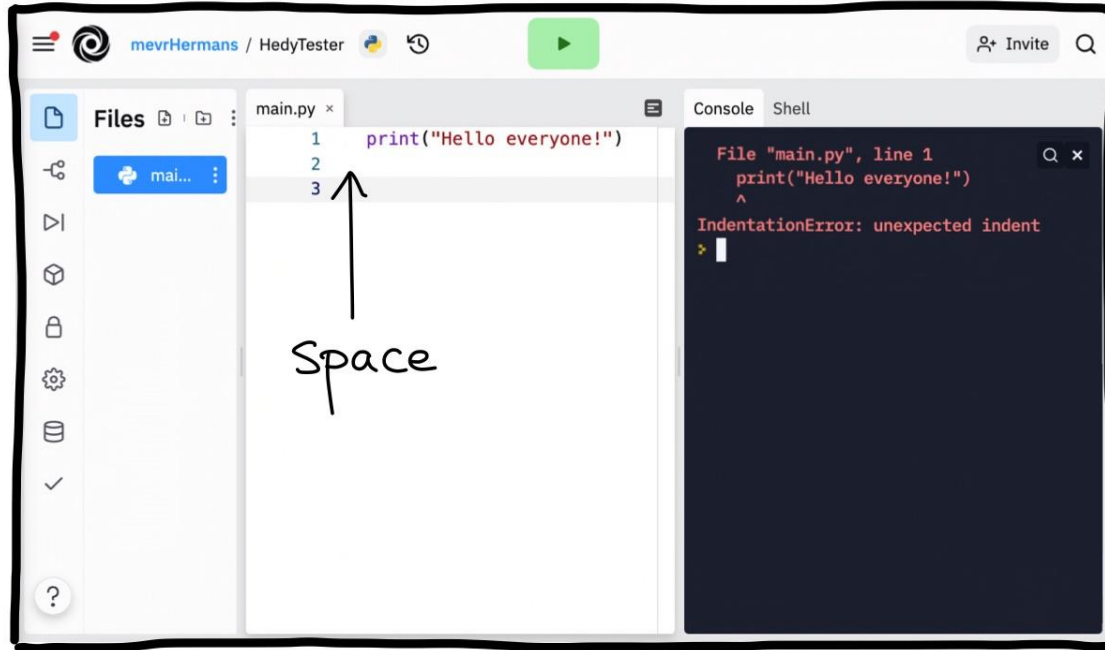
- A textual programming language.
  - Well, just another one?
- Motivation:
  - Syntax rule and development of concepts at the same time: a struggle
  - Error messages











# So conclusion is..

We need a textual programming language that is

- Simpler in syntax especially at the beginning
- Focus on concepts and expand gradually
- Better error messages
- Localized (?)



# Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities

Philip J. Guo  
UC San Diego  
La Jolla, CA, USA  
pg@ucsd.edu

## ABSTRACT

People from nearly every country are now learning computer programming, yet the majority of programming languages, libraries, documentation, and instructional materials are in English. What barriers do non-native English speakers face when learning from English-based resources? What desires do they have for improving instructional materials? We investigate these questions by deploying a survey to a programming education website and analyzing 840 responses spanning 86 countries and 74 native languages. We found that non-native English speakers faced barriers with reading instructional materials, technical communication, reading and writing code, and simultaneously learning English and programming. They wanted instructional materials to use simplified English without culturally-specific slang, to use more visuals and multimedia, to use more culturally-agnostic code examples, and to embed inline dictionaries. Programming also motivated some to learn English better and helped clarify logical thinking about natural languages. Based on these findings, we recommend learner-centered design improvements to programming-related instructional resources and tools to make them more accessible to people around the world.

## Author Keywords

non-native English speakers; learning programming

## ACM Classification Keywords

K.3.2 Computers and Education: Computer and Information Science Education – Literacy

```
con = sqlite3.connect(":memory:")
con.execute("CREATE TABLE person ... <omitted for space>")
with con:
    con.execute("INSERT INTO person ... ", ("Joe",))
# con.rollback() is called after the with block finishes
# with exception, exception still raised and must be caught
try:
    with con:
        con.execute("INSERT INTO person ... ", ("Joe",))
except sqlite3.IntegrityError:
    print("couldn't add Joe twice")
```

Figure 1. English is ubiquitous in source code, as shown in this example adapted from the official Python language docs for SQL database management [55]. English appears in comments, variable names, sqlite3 standard library API identifiers, and in both Python and SQL keywords.

discover barriers faced by non-native English speakers when trying to acquire technological expertise in complex domains. This knowledge can help us to design more inclusive learning technologies that further broaden access to digital literacy and job opportunities for people around the world.

In this paper, we focus on an exemplar form of technical training that is now in high demand: computer programming. Millions of people from over 190 countries are now learning programming online [11, 76]. Yet the most popular programming languages are all designed in English [39]. Their official documentation pages and code examples (e.g., Figure 1) are written in English, their ecosystems of libraries and accompanying-APIs are English-based [51], and the most popular textbooks, MOOCs [11, 76], and discussion sites (e.g., Stack Overflow [8]) are primarily in English. Although translations

# Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities

Philip J. Guo  
UC San Diego

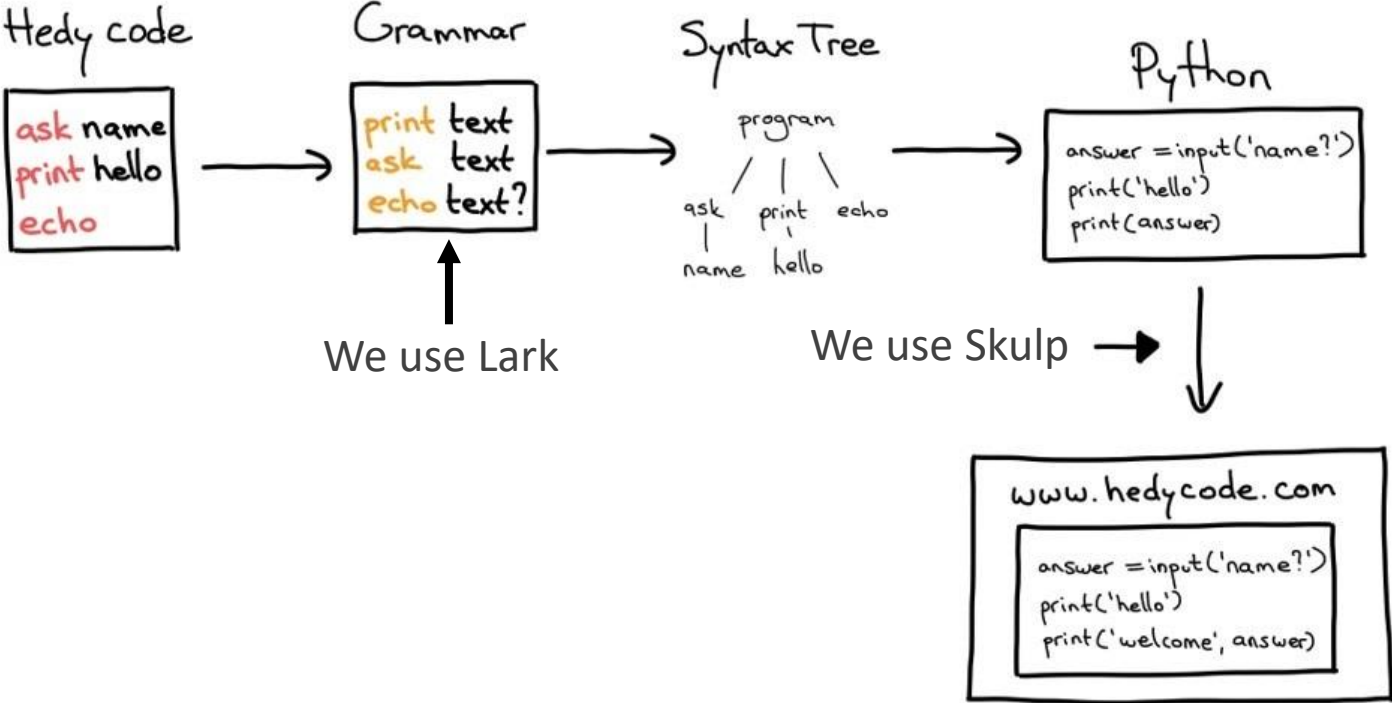
*out of date.” And an Arabic speaker wrote: “Most of the resources I use and the ones that are available are in English (online courses, videos, notes, books, etc.). Options in Arabic would be quite limited and not as good (translated books, university classes if you’re a computer sciences student,..)”*

...ing education needs and analyzing 670 responses spanning 86 countries and 74 native languages. We found that non-native English speakers faced barriers with reading instructional materials, technical communication, reading and writing code, and simultaneously learning English and programming. They wanted instructional materials to use sim-

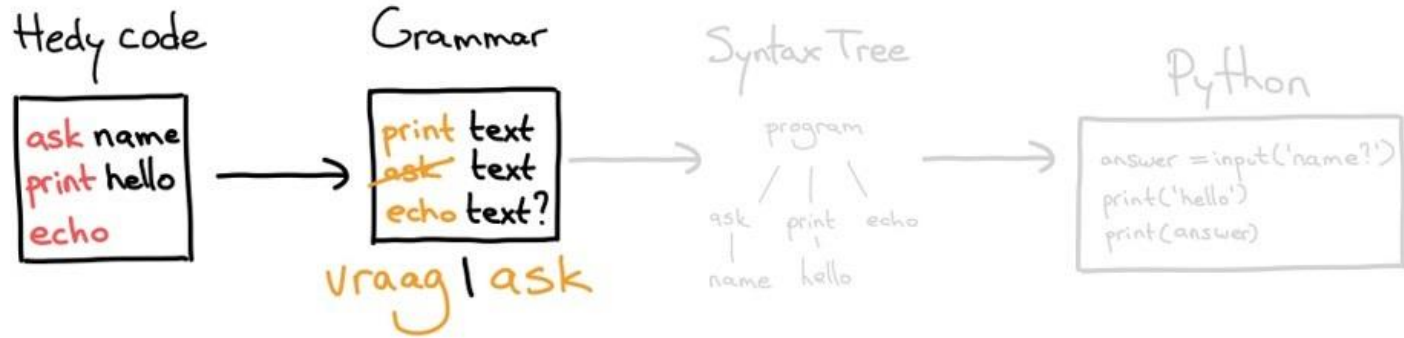
Figure 1. English is ubiquitous in source code, as shown in this example adapted from the official Python language docs for SQL database management [55]. English appears in comments, variable names, sqlite3 standard library API identifiers, and in both Python and SQL keywords.

*For instance, a native Arabic speaker wrote: “It’s really hard I expect 90% will quit because I have to learn English and coding at the same time, and it’s disappointing because I’m making slow progress in both topics.”* A Bulgarian speaker

# The architecture



# The architecture.. but bilingual



Bilingual people  
want to mix!



Title: PILATE: Programming In Languages that Aren'T English

## Organizers

Patrick Wang (France/French) & Felienne Hermans (Netherlands/Dutch) & Alaaeddin Swidan (Netherlands/Arabic) & Yizhou Qian (China/Mandarin)

# Arabic Hedy

<<< (أمثلة)  
سهل: مرحبا با عالم  
متوسط: عدد فيبوناتشي  
متقدم: لعبة الحياة لكونواي  
<<< (حدد فيبوناتشي (لامدا ن)  
... (إذا (أصغر؟ ن ٢)  
... ن  
... (جمع (فيبوناتشي (طرح ن ١)  
... (فيبوناتشي (طرح ن ٢)  
... (قول (فيبوناتشي ١٠)



قلب

Ramsey Nasser

# Arabic Hedy

- Right to left
- Variable names
- Numbers
- Extra's: Cultural aspects? → The ethnoprogramming model

## Right to left

- For the web interface: Tailwind solves this for us.
- For parsing.. Things get a bit more complicated.

print hello world  
قول مرحباً أيتها العالم!



# Right to left

مرحبا أيها العالم!

```
1 قول مرحبا أيها العالم!  
2
```

## Right to left

```
مرحبا أيها العالم!
```

```
1 قول مرحبا أيها العالم!  
2
```

```
قول مرحبا أيها العالم!
```

```
← print(input_string)
```

```
← print(input_string[0])
```

## Right to left

```
مرحبا أيها العالم!
```

```
1 قول مرحبا أيها العالم!  
2
```

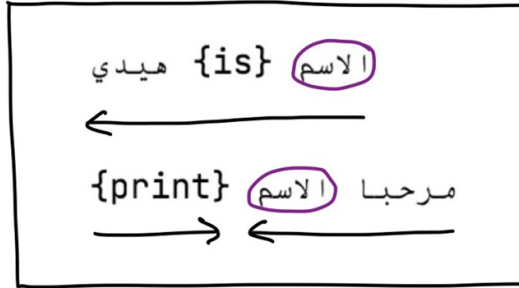
قول مرحبا أيها العالم! ← `print(input_string)`  
ق ← `print(input_string[0])`



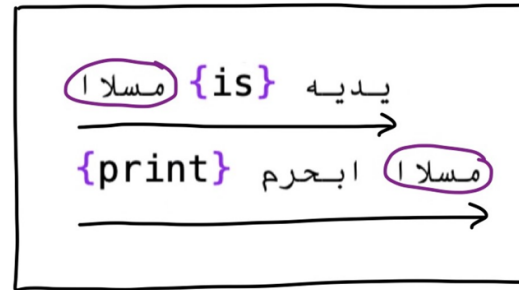
Microsoft Word interface showing a document titled "مستند 1 - Word". The ribbon includes "التخطيط الرئيسي", "إدراج", "رسم", "تصميم", "تخطيط", "مراجع", "مراسلات", "مراجعة", "عرض", "تعليمات", and "مشاركة". The main text area contains two lines of code: `{if} ضفدع {frog} {print} "ممتاز!"` and `{if} ضفدع {is} "frog" {print} "ممتاز!"`. A yellow circle with the letter 'I' is positioned below the second line of code.

# Presentation is up to programs

الاسم هو هيدي  
قول مرحبا الاسم

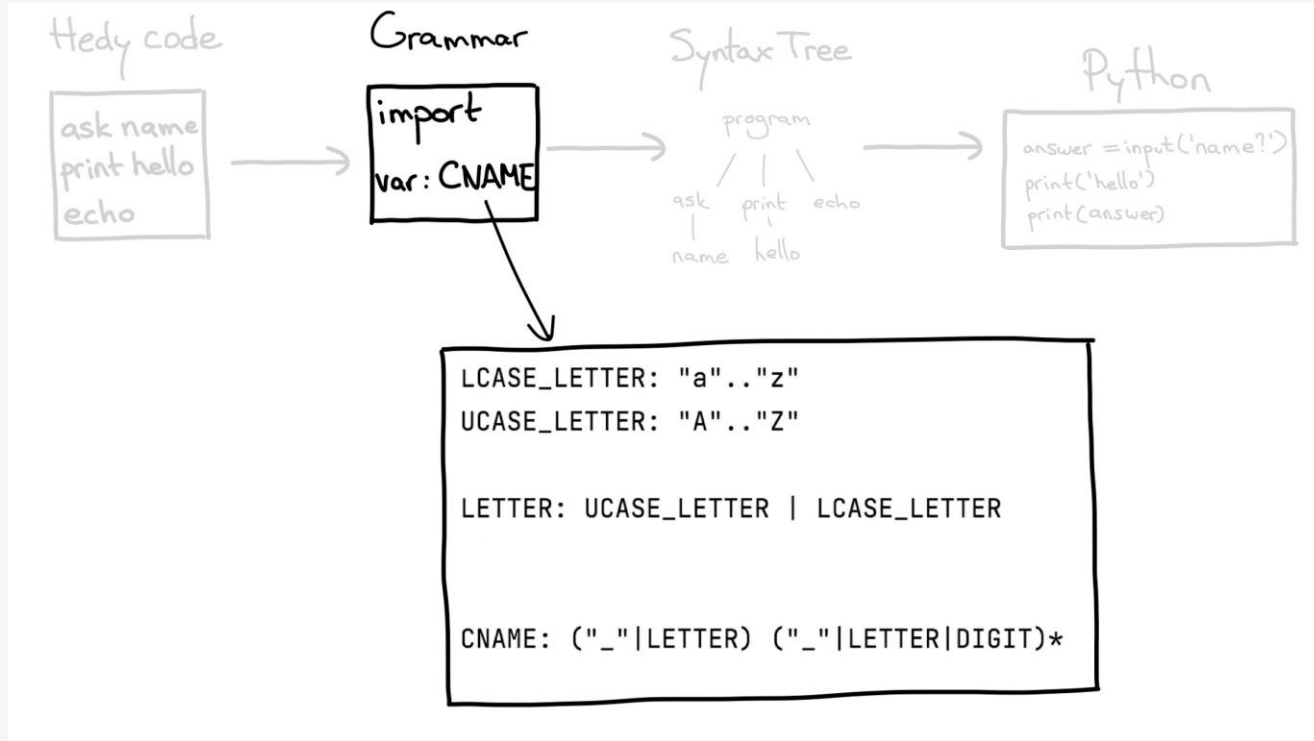


pycharm

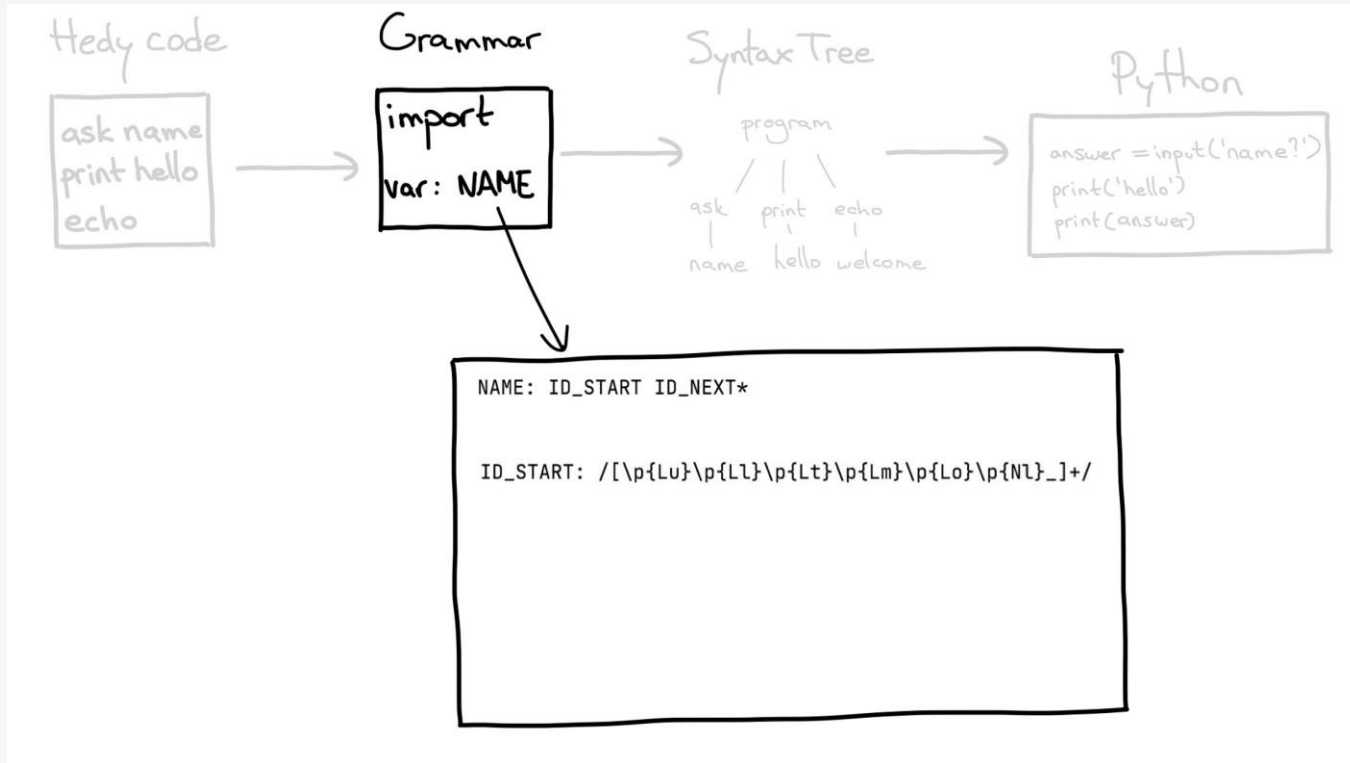


Vim

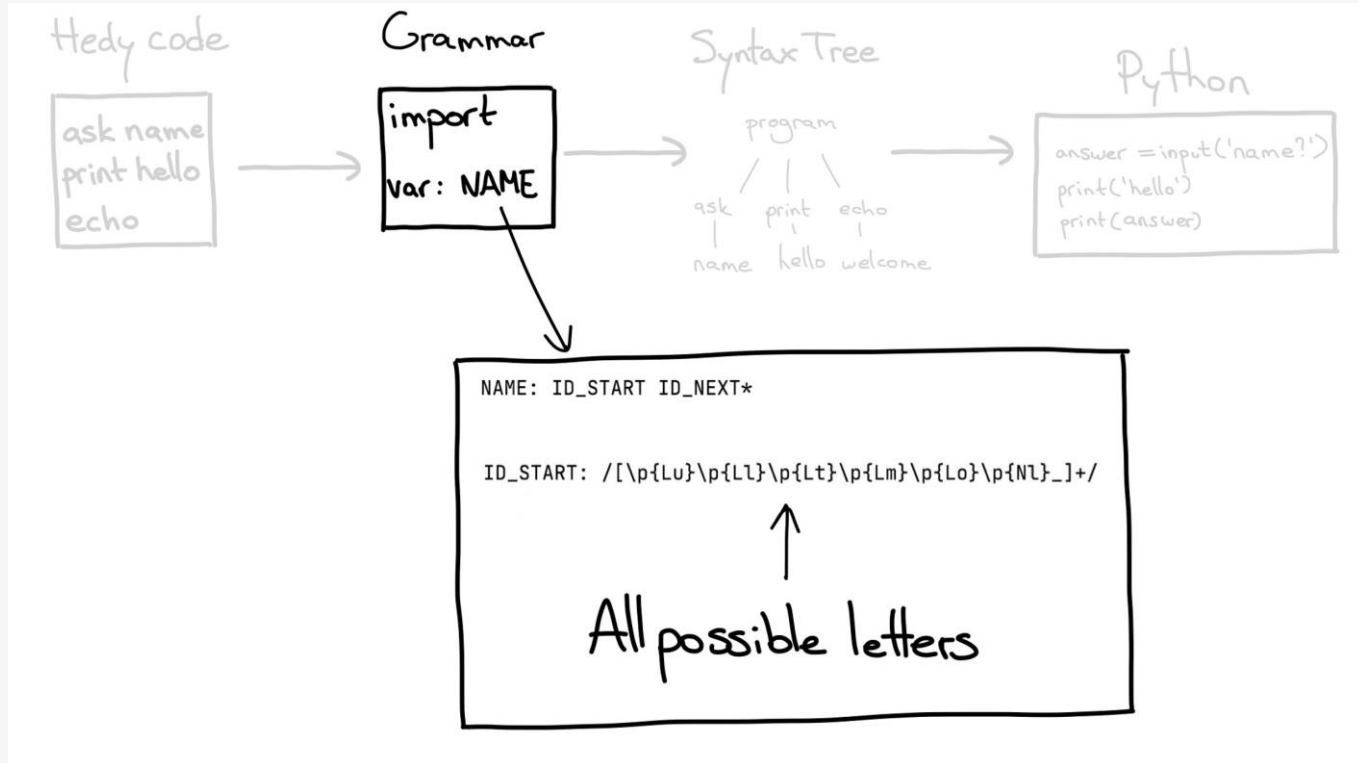
# Variable names



# Variable names



# Variable names









# Variable names

```
1 répète is ask Combien?  
2 print On y va répète fois!
```



**We can't run your program.**



It looks like you forgot to use a command on line 1.

# Numbers

```
هيدي ممتعة!  
هيدي ممتعة!  
هيدي ممتعة!
```

```
1  
2  
كرر 3 مرة قول "هيدي ممتعة!"  
print times repeat
```

```
for i in range(0, 3):  
    print("هيدي ممتعة!")
```

# Numbers

```
هيدي ممتعة!  
هيدي ممتعة!  
هيدي ممتعة!
```

```
1  
2  
كرر ٣ مرة قول "هيدي ممتعة!"  
/  
/  
/
```

```
for i in range(0, int('٣')):  
    print("هيدي ممتعة!")
```

# Numbers

```
for i in range(0, ٣):  
SyntaxError: invalid character  
in identifier
```

```
1  
2  
كرر ٣ مرة قول "هيدي ممتعة!"  
Arabic 3
```

```
for i in range(0, 3):  
print("هيدي ممتعة!")
```



# Numbers

```
i = 3  
print(i)
```



3

```
i = ٣  
print(i)
```



×

```
i = int('٣')  
print(i)
```



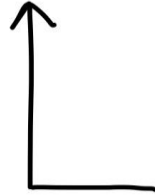
3

# Numbers

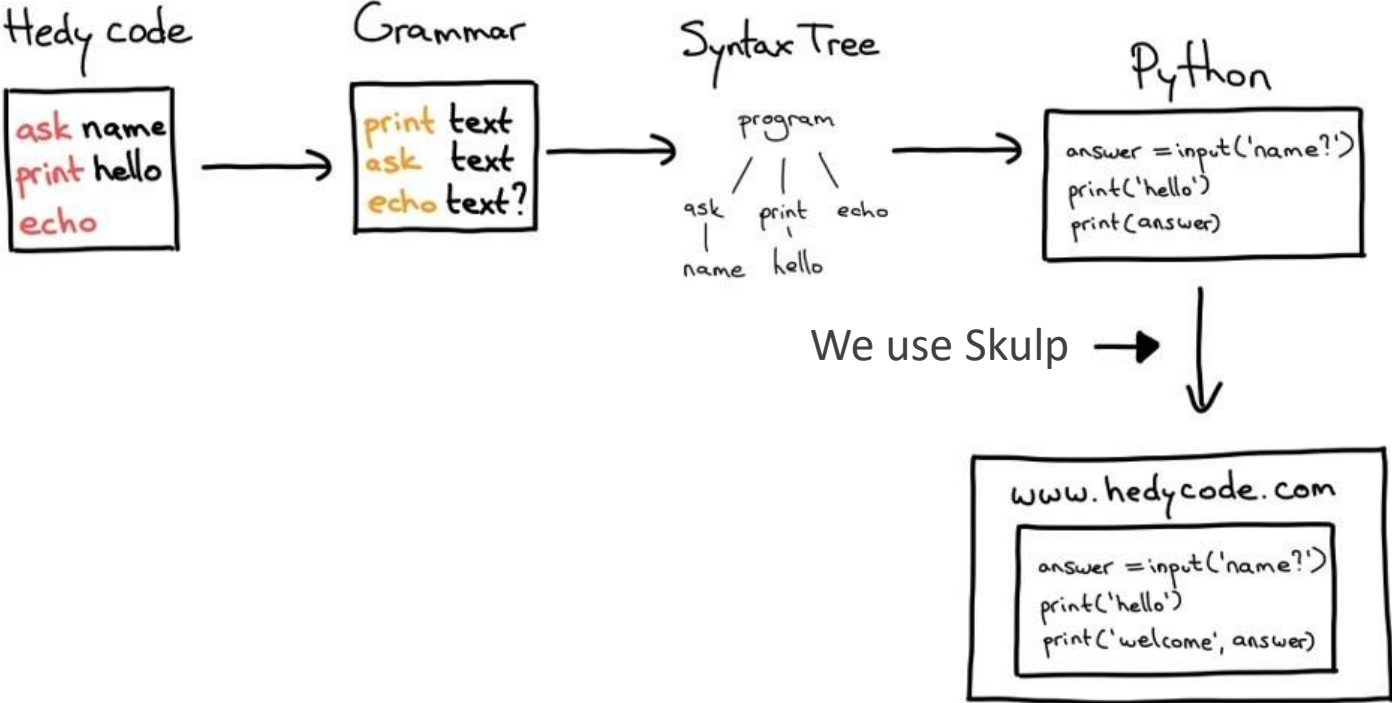
```
هيدي ممتعة!  
هيدي ممتعة!  
هيدي ممتعة!
```

```
1  
2  
/ / /
```

```
for i in range(0, int('٣')):  
    print("هيدي ممتعة!")
```



# The architecture





# Numbers

```
1 كرر ٣ مرة قول "هيدي ممتعة!"  
2
```

ValueError: invalid literal for int() with base 10: '٣'  
on line 1 in main.py

```
for i in range(0, int('٣')):  
    print("هيدى ممتعة!")
```



# Numbers

```
هيدي ممتعة!  
هيدي ممتعة!  
هيدي ممتعة!
```

```
1  
2  
كرر ٣ مرة قول "هيدي ممتعة!"
```



```
int_saver = int
def int(s):
    if isinstance(s, str):
        numerals_dict = {
            '٠': '0', '١': '1', '٢': '2', '٣': '3', '٤': '4',
            '٥': '5', '٦': '6', '٧': '7', '٨': '8', '٩': '9'}
        latin_numerals = ''.join([numerals_dict.get(l, l) for l in s])
        return int_saver(latin_numerals)
    return(int_saver(s))
```

Hardware

Software

Terminology

Culture

Let's talk about

gendered words

Hedy code

```
ask name  
print hello  
echo
```



Grammar

```
var is  
text
```



Syntax Tree

```
program  
 / | \  
ask print echo  
 |   |  
name hello
```



Python

```
answer = input('name?')  
print('hello')  
print(answer)
```

الاسم هو هيدي  
قول مرحبا الاسم

Hedy code

```
ask name
print hello
echo
```



Grammar

```
var is
text
```



Syntax Tree

```
program
 / | \
ask print echo
 |   |
name hello
```



Python

```
answer = input('name?')
print('hello')
print(answer)
```

male is



الاسم هو هيدي  
قول مرحبا الاسم

Hedy code

```
ask name  
print hello  
echo
```



Grammar

```
var is  
text
```



Syntax Tree

```
program  
 / | \  
ask print echo  
 |   |  
name hello
```



Python

```
answer = input('name?')  
print('hello')  
print(answer)
```

male is



الاسم هو هيدي  
قول مرحبا الاسم

الزاوية هي ٩٠  
استدر الزاوية  
تقدم ١٠٠

Hedy code

```
ask name
print hello
echo
```

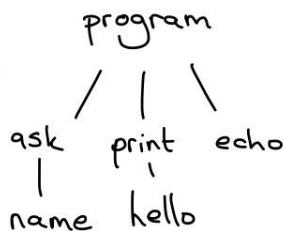


Grammar

```
var is
text
```



Syntax Tree



Python

```
answer = input('name?')
print('hello')
print(answer)
```

male is



الاسم هو هيدي  
قول مرحبا الاسم

female is



الزاوية هي ٩٠  
استدر الزاوية  
تقدم ١٠٠



Hedy code

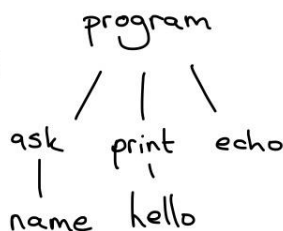
```
ask name
print hello
echo
```

Grammar

```
var is
text
```

هي اهو

Syntax Tree



Python

```
answer = input('name?')
print('hello')
print(answer)
```

male is



الاسم هو هيدي  
قول مرحبا الاسم

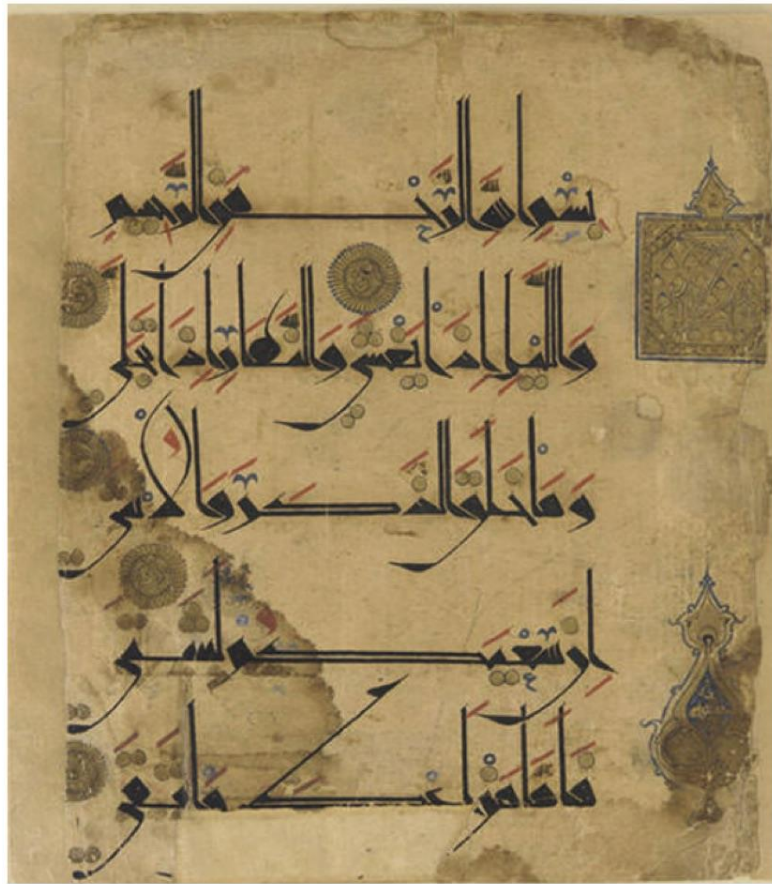
female is



الزاوية هي ٩٠  
استدر الزاوية  
تقدم ١٠٠

Let's talk about

Calligraphy



*Kufic script from the Holy Qur'an, 11th century (Image source: Smithsonian's Museums of Asian Art)*

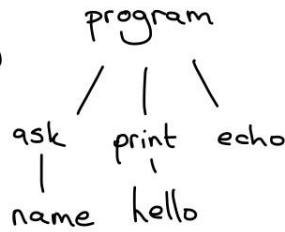
Hedy code

```
ask name
print hello
echo
```

Grammar

```
قول text
اسأل text
ردد text?
```

Syntax Tree



Python

```
answer = input('name?')
print('hello')
print(answer)
```

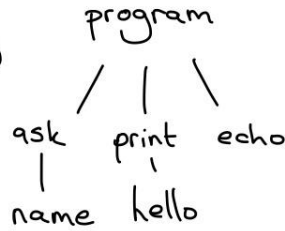
Hedy code

```
ask name
print hello
echo
```

Grammar

```
تقول text
اسأل text
ردد text?
```

Syntax Tree



Python

```
answer = input('name?')
print('hello')
print(answer)
```

\*" \_ " " ل " \* " \_ " " و " \* " \_ " " ق " \* " \_ " "

# Translation



Hosted Weblate

Dashboard

Projects ▾

Languages ▾

Checks ▾



Hedy

Components

Languages

Info

Search












Insights ▾

Files ▾

Tools ▾

Manage ▾

Share ▾

Language	Translated	Unfinished	Unfinished words
 Arabic 	46%	1,959	25,733
 English 			
 Albanian 	26%	2,714	24,586
 Bengali 	22%	2,844	35,441
 Bulgarian 	29%	2,586	33,848

# Translation

- In progress.
- It is not easy to take (some) translation decisions, more people the better.
- Outreach and community building, therefore, is vital.
- We have now a small community of Arabic translators (native Arabic speakers, computer science students/professionals so far)
-

# What's next?

- We want to try out in the real world: with Arabic speaking children, but also with or language
  - Evaluation studies: design choices, learning effect
- Improve the language and the platform
  - Opportunities for OU students (?): automatic detection of certain language elements (gendered is for example)
- Writing about the experiences: a paper and a grant proposal are in the pipeline.



# Demo time

- <https://hedycode.com/>
  - Try it out 😊