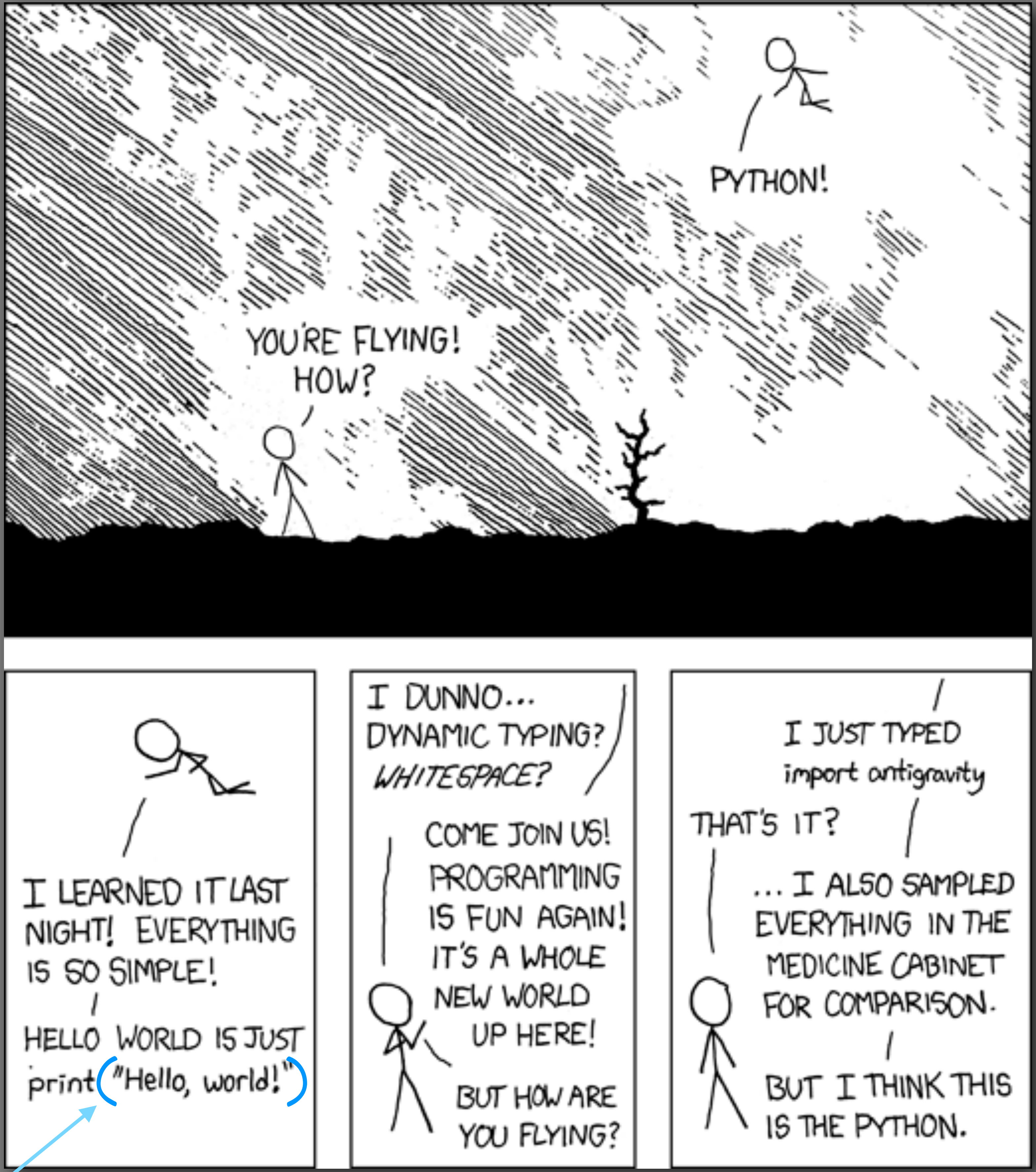

PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print("Hello, world!")
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Python 3

xkcd.com/353

Take-home midterm #2

Available: this Sunday night (10/28)

Must return by: the following Sunday (11/4) at 5pm

Time-limit: one sitting (with small breaks)

Covers: from the end of Hmmm to before break
functional programming, higher-order functions, recursion, analysis (recurrence and Big O), use-it-or-lose-it, trees

Resources: one, 8½ x 11 sheet of notes (double-sided)

Honor code: don’t discuss exam questions

What’s still unclear about FP?

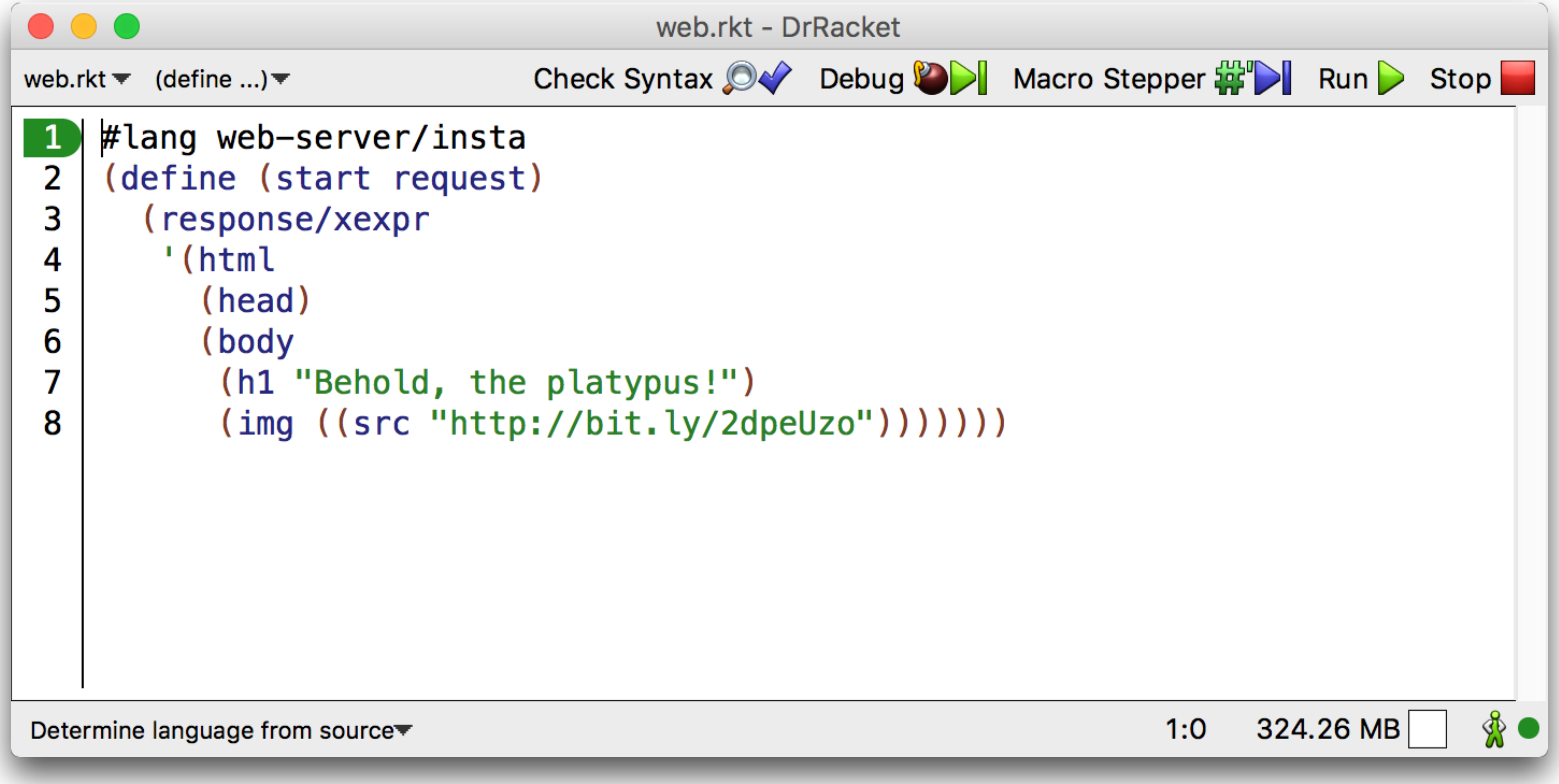
In the larger space, write down any lingering questions you have about functional programming, recursion, “use it or lose it”, analysis (e.g., recurrence relations), Racket, etc.

Use the “lingering questions” portion for questions about *today’s* material.

Firstname Lastname Th. 10 / 25

(Your response)

There’s so much more to Racket



```
#lang web-server/insta
(define (start request)
  (response/xexpr
   '(html
     (head)
     (body
      (h1 "Behold, the platypus!")
      (img ((src "http://bit.ly/2dpeUzo")))))))
```

racket-lang.org

clojure.org

Racket is
syntactic sugar
for the λ-calculus.

```
x ∈ Identifiers
expr ::= x
       |  (lambda (x)  expr)
       |  (expr expr)
```

Preface:

What follows will never be on a CS 42 assignment or exam.

```
(define (□ x)
  (if x false true))
```

```
(define (□ x y)
  (if x y false))
```

```
(define (□ x y)
  (if x true y))
```

```
(define (NOT x)
  (if x false true))

(define (AND x y)
  (if x y false))

(define (OR  x y)
  (if x true y))
```

```
(define NOT (λ (x)
  (if x false true)))

(define AND (λ (x y)
  (if x y false)))

(define OR  (λ (x y)
  (if x true y)))
```

```
(define NOT (λ (x)
  (if x FALSE TRUE)))
```

```
(define AND (λ (x y)
  (if x y FALSE)))
```

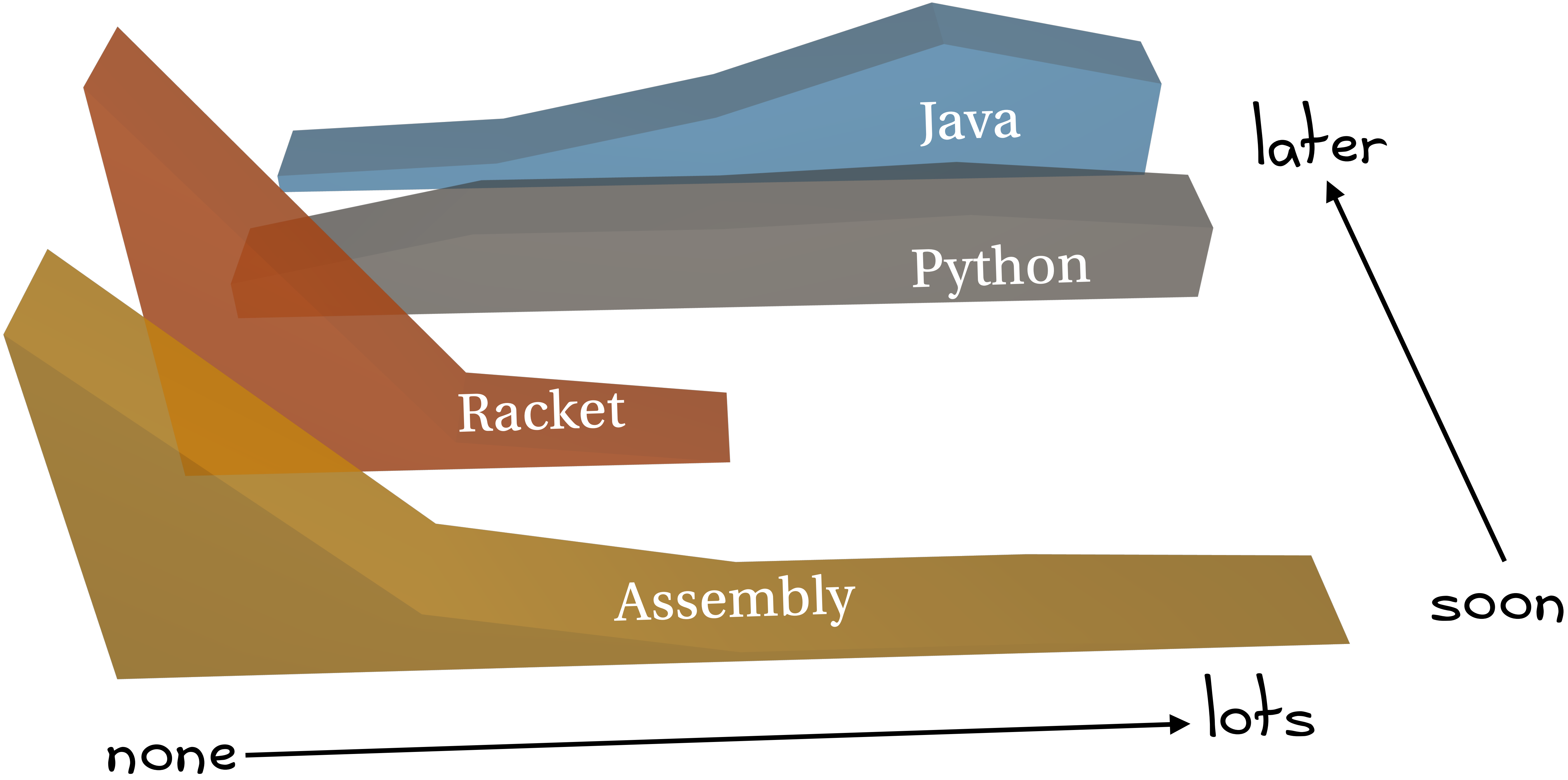
```
(define OR  (λ (x y)
  (if x TRUE y)))
```

```
(define TRUE  (λ (x y) x))
(define FALSE (λ (x y) y))
```

```
  (define NOT (λ (x)
    (x FALSE TRUE)))


  (define AND (λ (x y)
    (x y FALSE)))


  (define OR  (λ (x y)
    (x TRUE y)))

(define TRUE  (λ (x y) x))
(define FALSE (λ (x y) y))
```

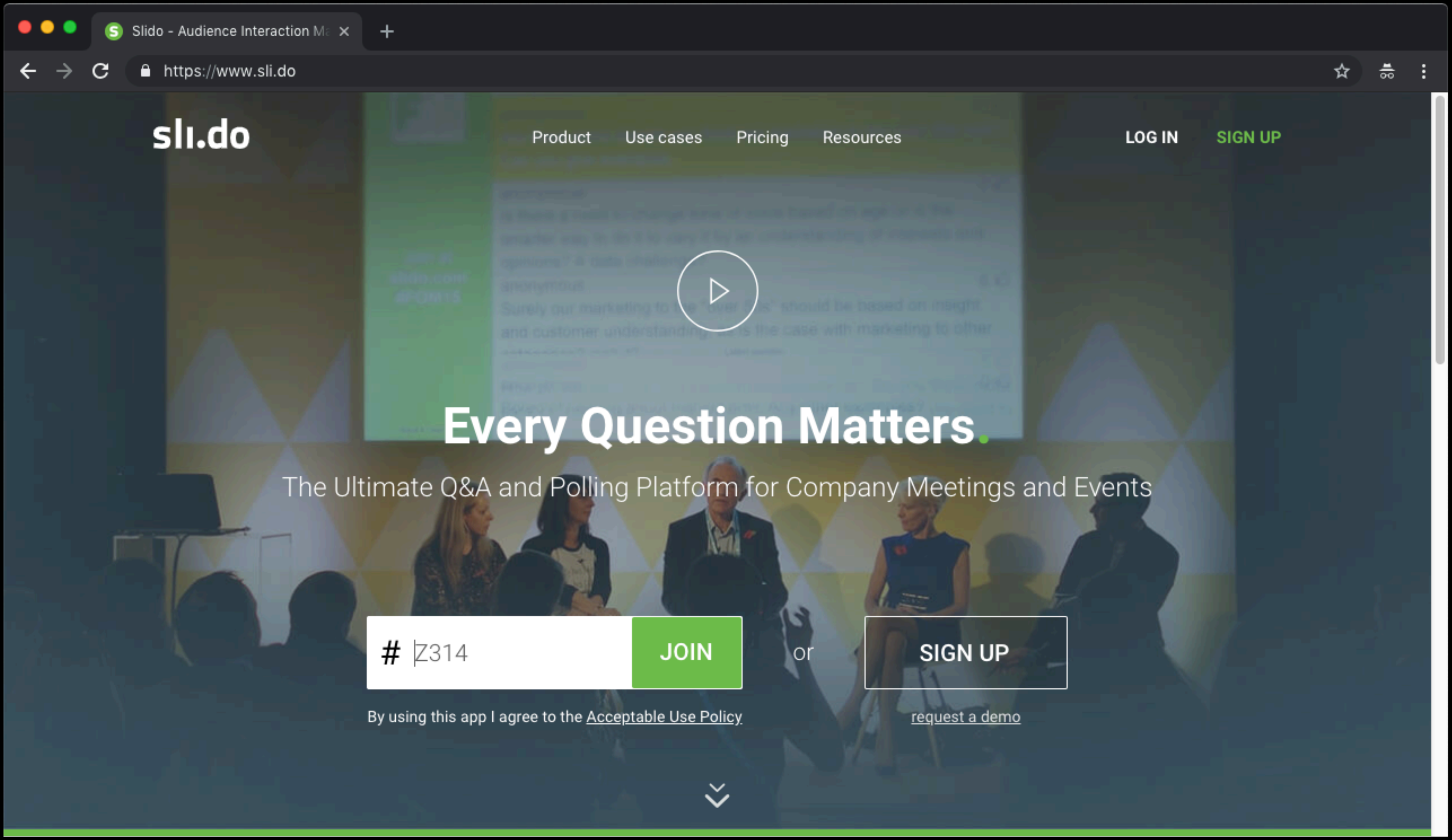
Python Overview

Prior experience: programming languages



repl.it/languages/python3

slido.com (event code: Z314)



Every Question Matters.

The Ultimate Q&A and Polling Platform for Company Meetings and Events

By using this app I agree to the Acceptable Use Policy

The essence of Python

Everything is an object.*

Every object has

- a **value**.
- a **type**.
- an **identity**.
- a **namespace**.

```
>>> 1                       # value
1
>>> type(1)                 # type
<type 'int'>
>>> id(1)                   # identity
140686900921016
>>> dir(1)                  # namespace
['__abs__', ..., 'real']
```

*but some objects (e.g., numeric and boolean literals) are “special”.