Async in a nutshell

- A library for writing concurrent programs
  - Less error-prone than traditional threading models

- The types show which operations might block:

  ```haskell
  val count_lines : filename:string -> int Deferred.t
  ```

- The returned “deferred” is like a box:
  - Initially it is empty
  - The box is filled with the result of the computation

- The deferred *does not* contain the computation itself
Combining computations

• Functions of deferred type may be sequenced using >>=:

```haskell
count_lines "myfile"
>>= fun n ->
count_lines "anotherfile"
>>= fun n' ->
Deferred.return (sprintf "%d lines" (n + n'))
```

• The blocks between binds are **uninterruptible**
  • More computations may be scheduled, but the current one is never pre-empted
  • Dramatically simplifies concurrent programming
An example

- Time for some audience participation...

http://oud.janestreet.com:8080/fred/12345
Go get it

opam install async