derp

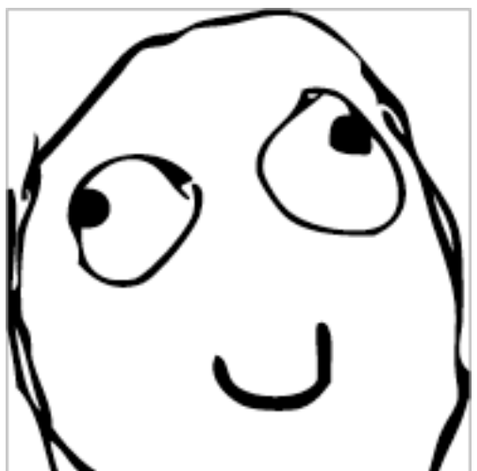
Matt Might

University of Utah

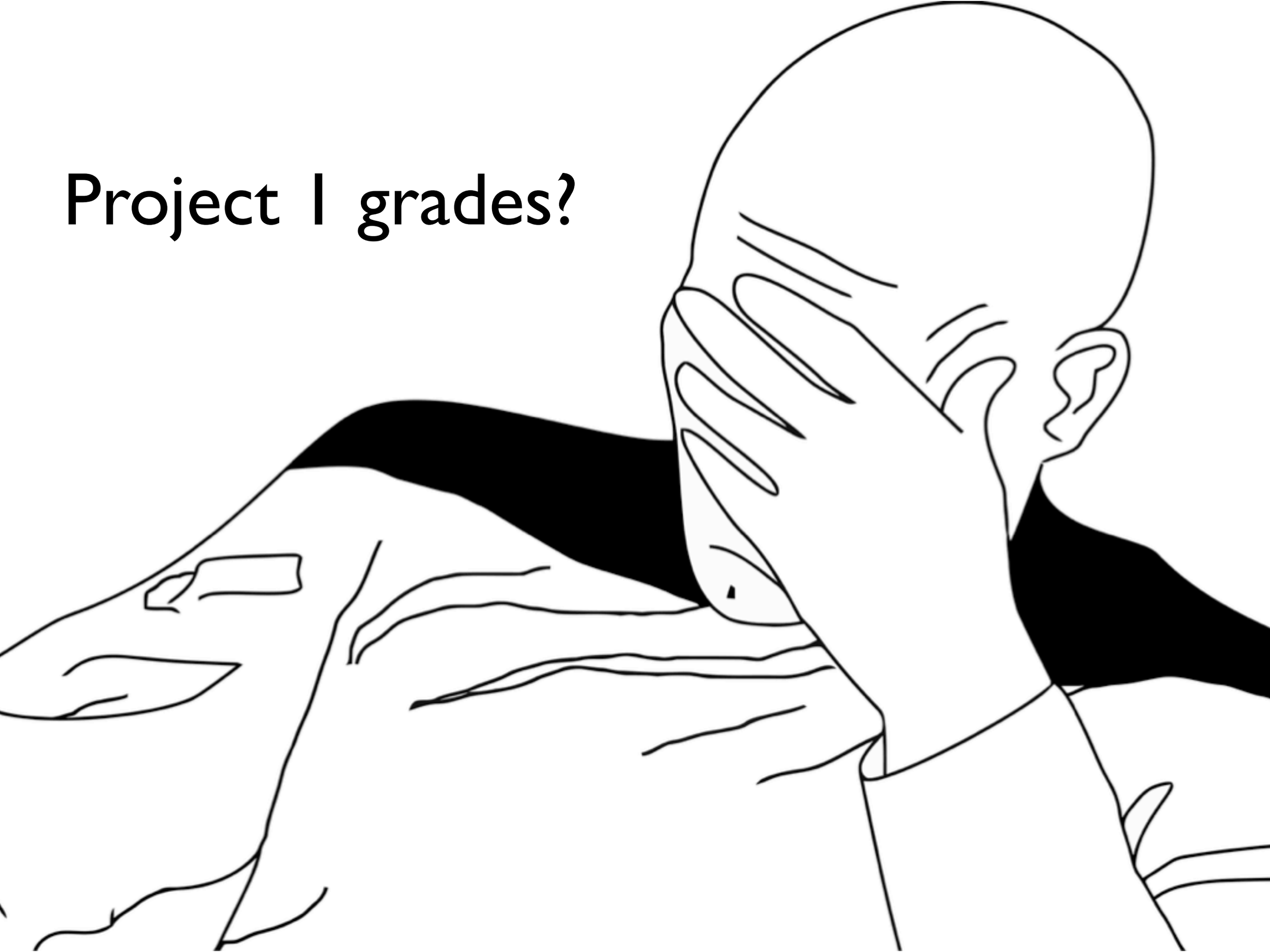
matt.might.net

derivative parsing

derivative parsing



Project 1 grades?



Project 2

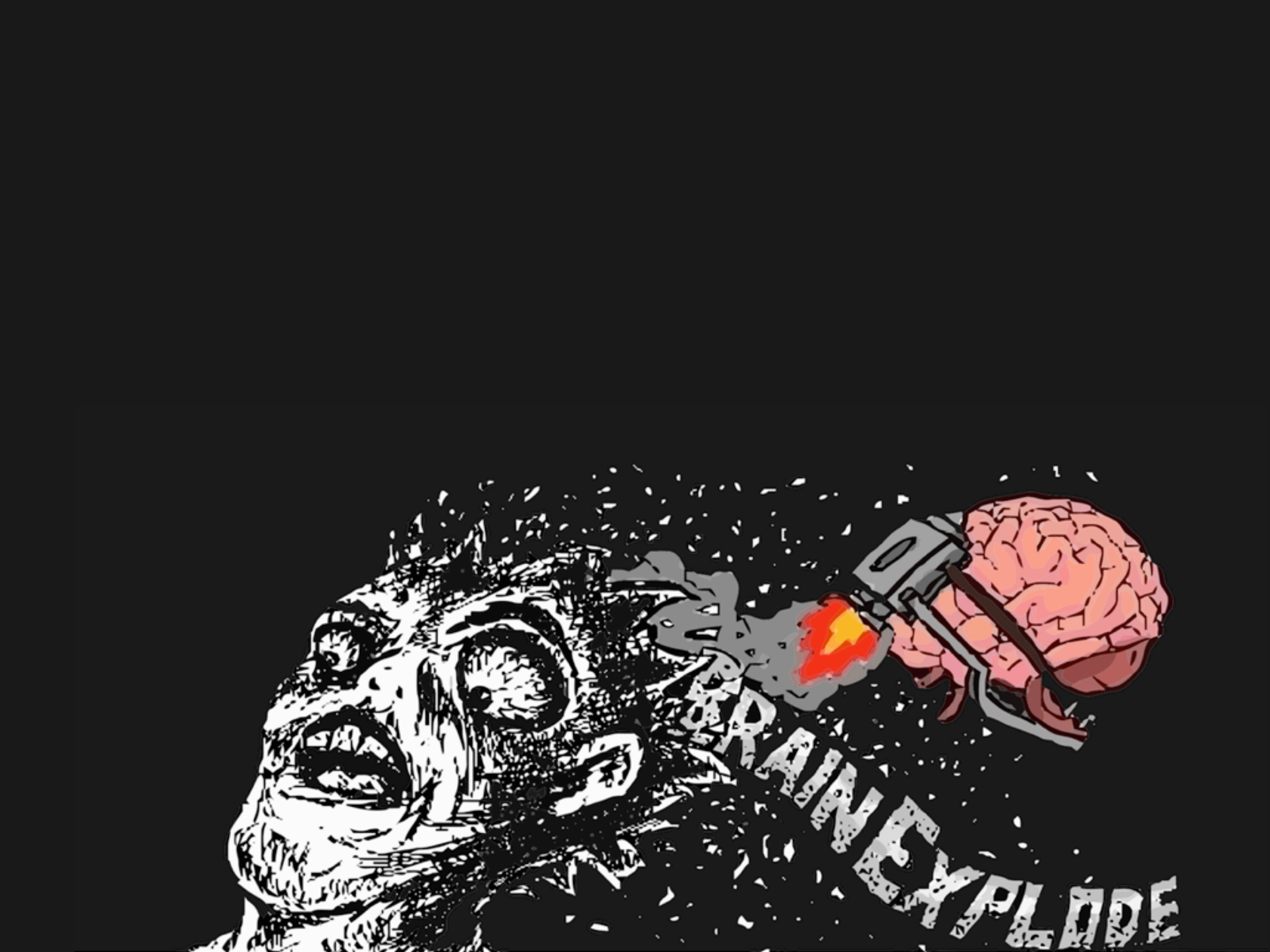
Project 2

How to start?

How to start?

```
$ emacs pyparse.rkt
```

```
$ emacs derivative-parsers.rkt
```


BRAINSPLODE

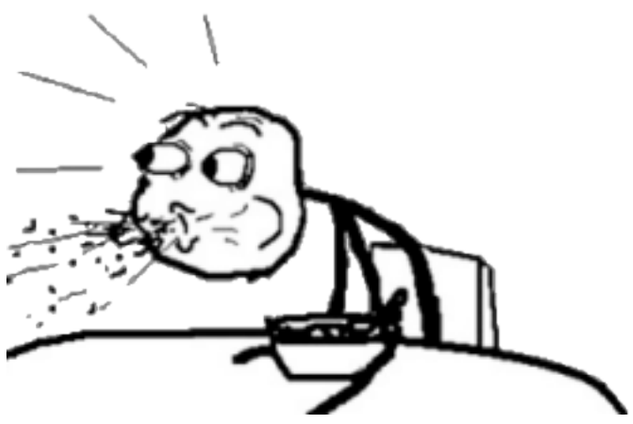
“What is this *sourcery*!?”

Step 1: Read the assignment

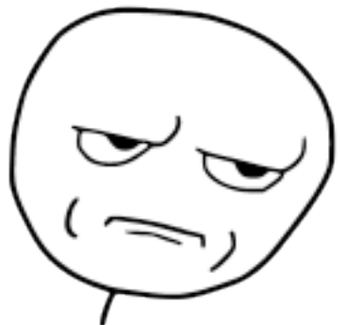
Step 1: Read the assignment

Step 2: Download stub

Step 3: `make; make parse`



Step 4: “Just add -->”



Defining languages

(lang *lang*)

(parse *parser stream*)

Language forms

(empty)

(eps)

(eps *value*)

(? *predicate class*)

‘c

string

number

boolean

lit

(literal->language *lit*)

Examples

Regular forms

(or *lang* ...)

```
(seq lang ...)
```

(seq* *lang* ...)

```
(seq! qqlang ...)
```

Examples

(rep *lang*)

(rep+ *lang*)

Examples

(opt *lang*)

(opt *lang def*)

Reductions

(red *lang proc*)

(--> lang proc)

```
(@--> lang proc)
```

```
(red lang (λ t
(apply proc t)))
```

($--> *lang exp* ...)

```
(begin
 (define $$ ...)
 (define ($ n) ...)
  exp ...)
```

(>--> *lang case* ...)

```
(--> lang (λ (t)
  (match t
    case ...))
```

(car *lang*)

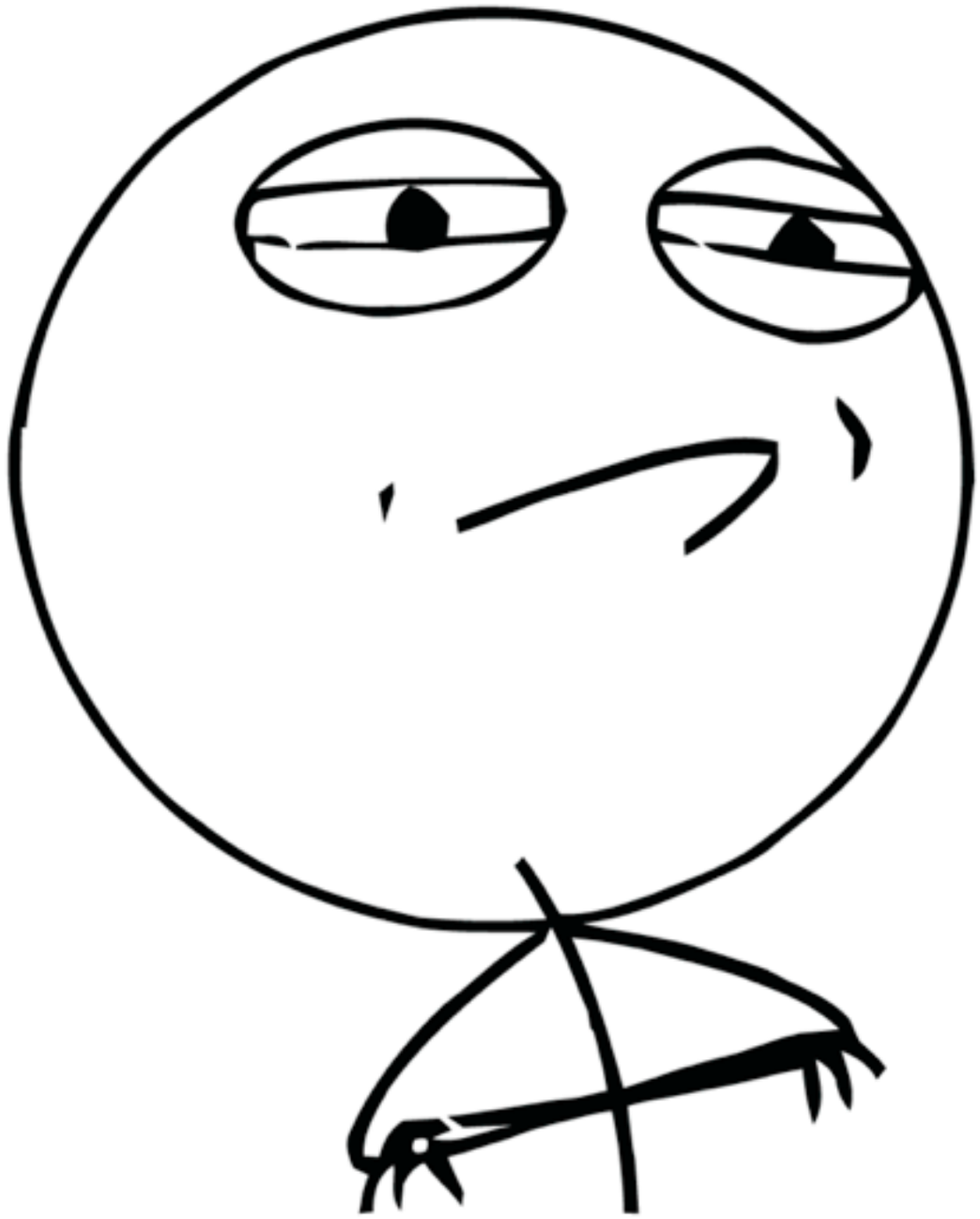
```
(--> lang (λ (t)
(car t))
```

Grammars

```
(grammar
  [nonterm lang]
  ...
  start)
```

(grammar-from-file
  start
  filename)

Python in derp



Questions