

Fun and Games 2

Ben Greenman
2021 11 07



identifier already defined

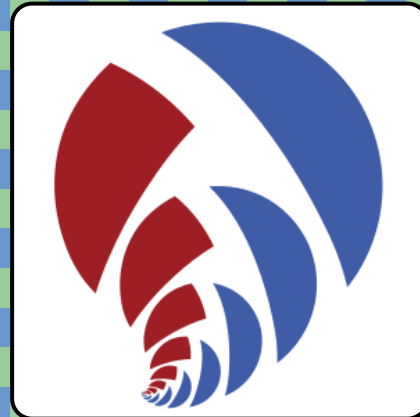
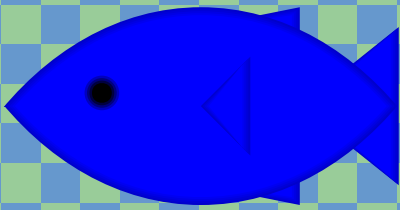
contract error

namespace mismatch

bad syntax



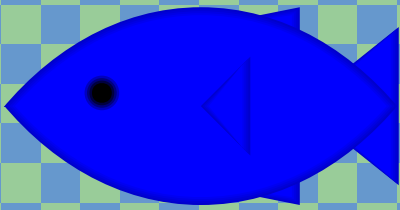
Reminder: Programming is fun!



Standard Fish
2019

Quicksript
2020

Reminder: Programming is fun!



Standard Fish
2019



Quicksript
2020



Syntax Parse
2021

Extensible Functions

```
(interp0 '(+ 1 (+ 5 6)))  
(interp1 '(+ 1 (* 5 6)))
```

Extensible Functions

```
(interp0 '(+ 1 (+ 5 6)))  
(interp1 '(+ 1 (* 5 6)))
```

```
(define (fix f)  
  ((λ (x) (f (λ (g) ((x x) g))))  
   (λ (x) (f (λ (g) ((x x) g))))))
```

```
(define ((interp0 recur) e)  
  (match e  
    ....))
```

```
(define ((interp1 recur) e)  
  (match e  
    ....))
```

Extensible Functions

```
((fix interp0) '(+ 1 (+ 5 6)))  
((fix interp1) '(+ 1 (* 5 6)))
```

```
(interp0 '(+ 1 (+ 5 6)))  
(interp1 '(+ 1 (* 5 6)))
```

```
(define (fix f)  
  ((λ (x) (f (λ (g) ((x x) g))))  
   (λ (x) (f (λ (g) ((x x) g))))))
```

```
(define ((interp0 recur) e)  
  (match e  
    ....))
```

```
(define ((interp1 recur) e)  
  (match e  
    ....))
```


Extensible Functions

```
((fix interp0) '(+ 1 (+ 5 6)))  
((fix interp1) '(+ 1 (* 5 6)))
```

```
(interp0 '(+ 1 (+ 5 6)))  
(interp1 '(+ 1 (* 5 6)))
```

```
(define (fix f)  
  ((λ (x) (f (λ (g) ((x x) g))))  
   (λ (x) (f (λ (g) ((x x) g))))))
```

```
(define ((interp0 recur) e)  
  (match e  
    ....))
```

```
(define ((interp1 recur) e)  
  (match e  
    ....))
```

```
(define-extend (interp0 e)  
  (match e  
    [ `(+ ,x ,y)  
      (+ (interp0 x) (interp0 y))]  
    [(? number?) e]))
```

```
(define-extend (interp1 e)  
  #:extend interp0  
  (match e  
    [ `(* ,x ,y)  
      (* (interp1 x) (interp1 y))]  
    [_ (interp0 e)]))
```

Dot and Underscore Notation

```
(define h
  (new Horse [breed "Danish Pony"]))

(send h run "Bella")

(define fibs (vector 42 1 2 3 5 8))

(vector-ref fibs 0)

(vector-set! fibs
  0
  (+ 1
    (vector-ref fibs 0)))
```

Dot and Underscore Notation

```
(define h
  (new Horse [breed "Danish Pony"]))

(send h run "Bella")

(define fibs (vector 42 1 2 3 5 8))

(vector-ref fibs 0)

(vector-set! fibs
  0
  (+ 1
    (vector-ref fibs 0)))
```

```
(define h
  (new Horse [breed "Danish Pony"]))

(h.run "Bella")

(define fibs (vector 42 1 2 3 5 8))

fibs_0

(:= fibs_0 (+ 1 fibs_0))
```

JS Dict Notation

```
(define tbl (hash 'a 1 'b 2 'c 3))
```

```
(define a (hash-ref tbl 'a))
```

```
(define b (hash-ref tbl 'b))
```

```
(define tbl2  
  (hash-remove  
    (hash-remove tbl 'a)  
    'b))
```

JS Dict Notation

```
(define tbl (hash 'a 1 'b 2 'c 3))  
  
(define a (hash-ref tbl 'a))  
  
(define b (hash-ref tbl 'b))  
  
(define tbl2  
  (hash-remove  
    (hash-remove tbl 'a)  
    'b))
```

```
(define tbl (hash 'a 1 'b 2 'c 3))  
  
(js-extract {a b #:rest tbl2} tbl)  
  
;; a = 1  
;; b = 2  
;; tbl2 = (hash 'c 3)
```

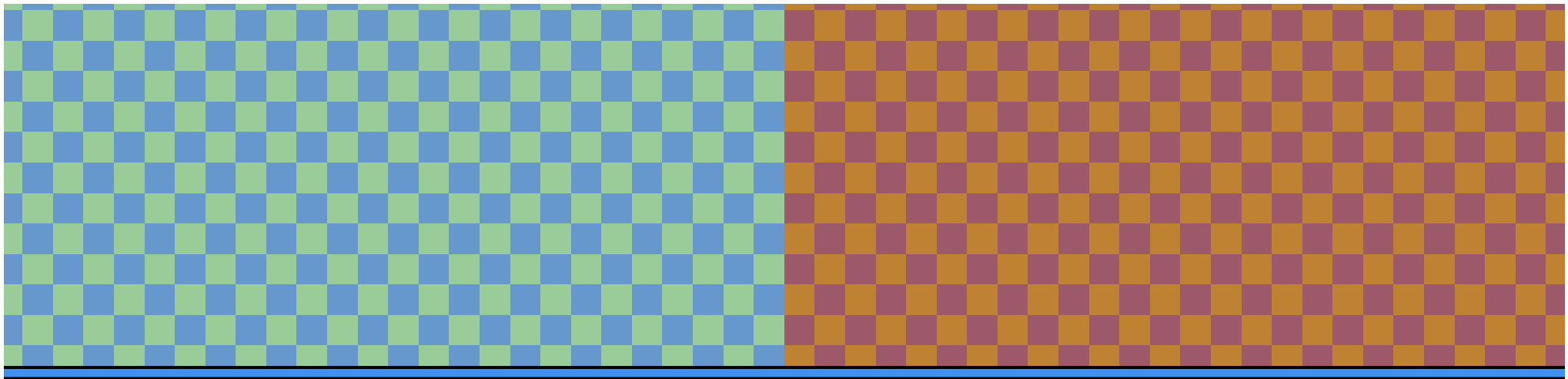
Pyret for

```
(andmap (λ (thing quantity)
  ....)
  things
  quantities)
```

Pyret for

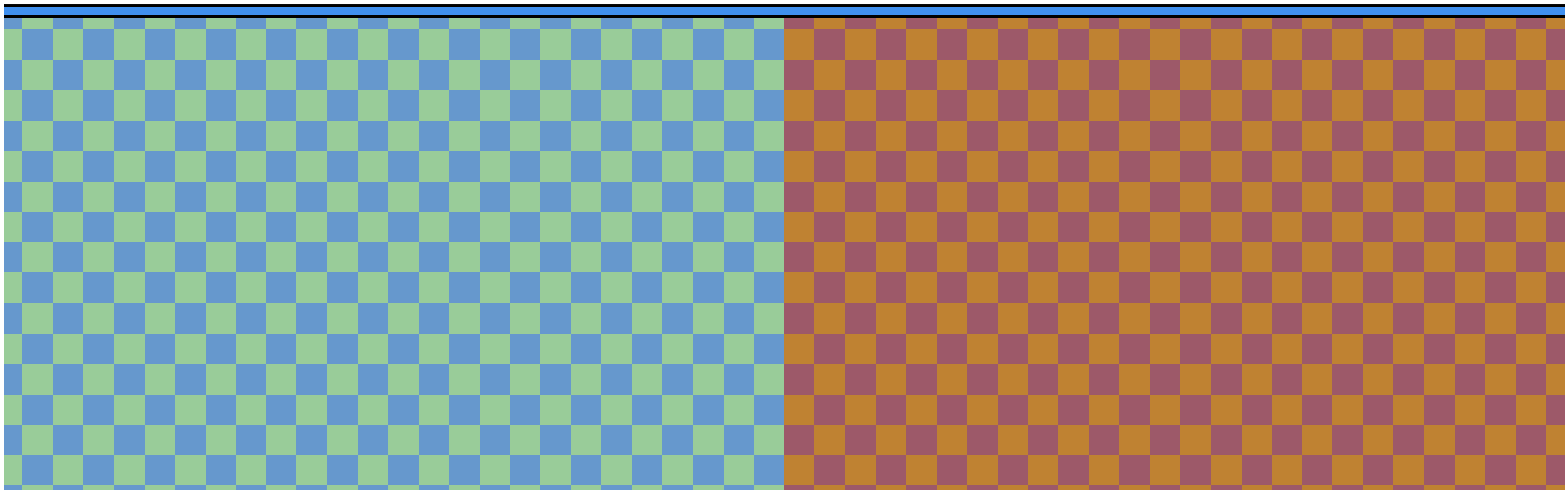
```
(andmap (λ (thing quantity)
  ....)
  things
  quantities)
```

```
(pyret-for andmap
  ([thing things]
   [quantity quantities])
  ....)
```



(~? A B)

(~@ A ...)



Sequencing definitions with ~?

```
(define x 4)
(define y 18)
(define-values (quot rem)
  (quotient/remainder x y))
```

```
(define*
  [x 4]
  [y 18]
  [(quot rem)
   (quotient/remainder x y)])
```

Sequencing definitions with ~?

```
(define x 4)
(define y 18)
(define-values (quot rem)
  (quotient/remainder x y))
```

```
(define*
  [x 4]
  [y 18]
  [(quot rem)
   (quotient/remainder x y)])
```

```
(define-syntax-parser define*
  [(_ (~or [id:id expr:expr]
           [(idv:id ...+) expr:expr])
      ...+)
   #'(begin
       (~? (define id expr)
           (define-values (idv ...) expr)) ...))])
```

Splicing keyword args with ~@

```
(string-trim " abc "  
  #:left? #t  
  #:right? #t)
```

```
(string-trim " abc "  
  #:left?  
  #:right?)
```

Splicing keyword args with ~@

```
(string-trim " abc "  
  #:left? #t  
  #:right? #t)
```

```
(string-trim " abc "  
  #:left?  
  #:right?)
```

```
(define-syntax-parser #%app/flag  
  [(_ f no-kw:expr ...  
     kw:arg/kw ...)  
   #'(%app f no-kw ... {~@ kw.k kw.v} ...)])
```

try

define-extend

define/datum

displayln*

marc-matcher

define/curry

define*

try

dot underscore

make-variable

kw-ctc

try

cmdline/help

define/freevar

js-dict

fnarg

#!/app/flag

while/break

FreshVar

log-once

pyret-for

Submissions

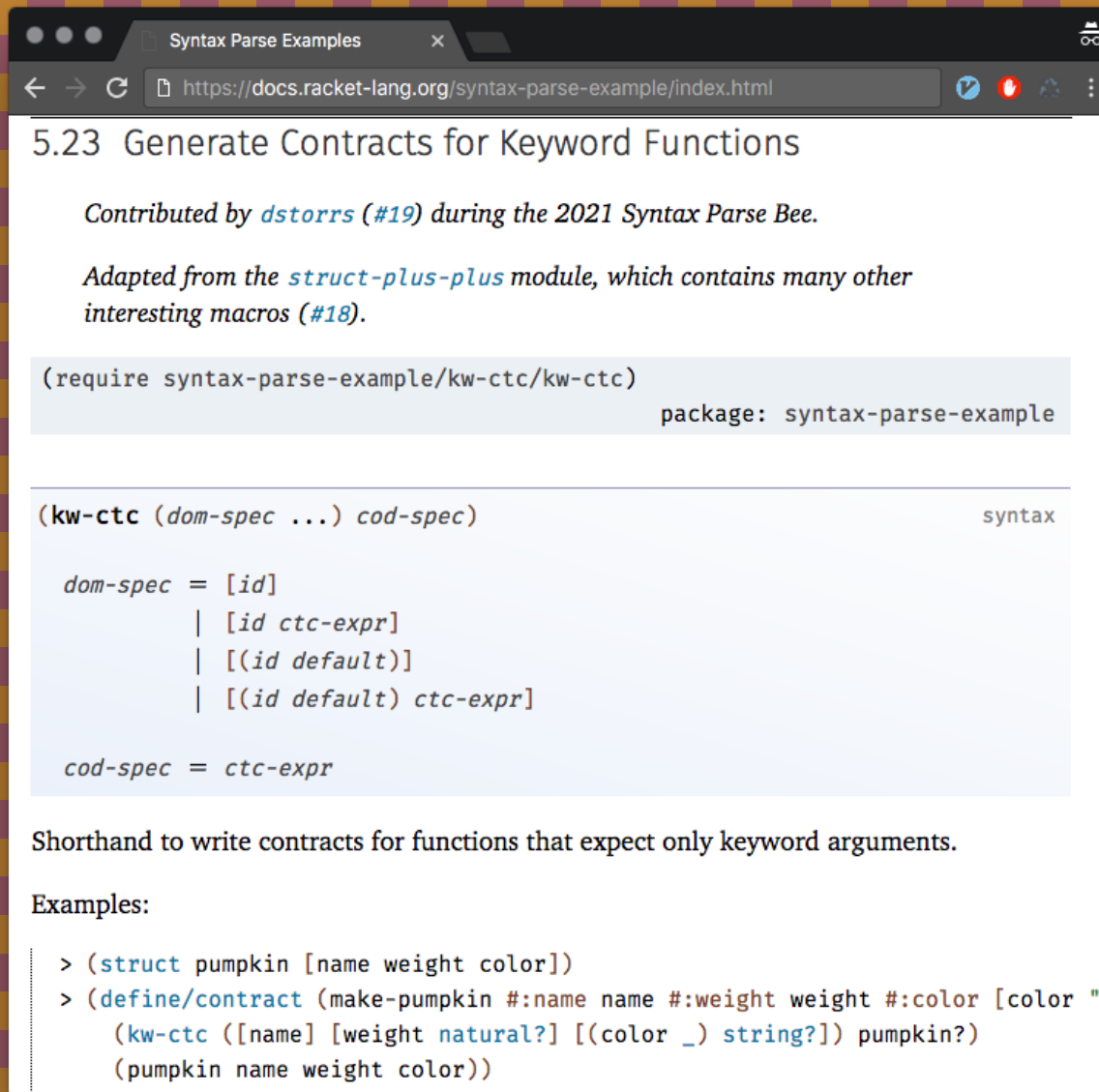
github.com/syntax-objects/Summer2021

The screenshot shows the GitHub interface for the repository 'syntax-objects/Summer2021'. The page title is 'syntax-objects / Summer2021' with a 'Public' badge. It features navigation tabs for 'Code', 'Issues 23', 'Pull requests', 'Actions', 'Security', and 'Insights'. A search bar contains the query 'is:issue is:open'. Below the search bar, there are filters for 'Labels 10' and 'Milestones 0', along with a 'New issue' button. The main content area displays a list of 23 open issues, with the following details visible:

Issue Title	Author	Label	Projects	Milestones	Assignee	Sort
Thank You Participants	benn					1
Functions With Free Variables: Poorman's Implicit Parameters	shhyou					1
Abstracting Syntax Patterns With Parameterized Syntax Classes	shhyou					
Generate Temporay Identifiers On-the-fly	shhyou					2
define/with-datum -- defining pattern variables for datums	shhyou					

Community Resource

docs.racket-lang.org/syntax-parse-example



Syntax Parse Examples

https://docs.racket-lang.org/syntax-parse-example/index.html

5.23 Generate Contracts for Keyword Functions

Contributed by [dstorrs](#) (#19) during the 2021 Syntax Parse Bee.

Adapted from the [struct-plus-plus](#) module, which contains many other interesting macros (#18).

```
(require syntax-parse-example/kw-ctc/kw-ctc)
```

package: syntax-parse-example

```
(kw-ctc (dom-spec ...) cod-spec) syntax
```

```
dom-spec = [id]
           | [id ctc-expr]
           | [(id default)]
           | [(id default) ctc-expr]

cod-spec = ctc-expr
```

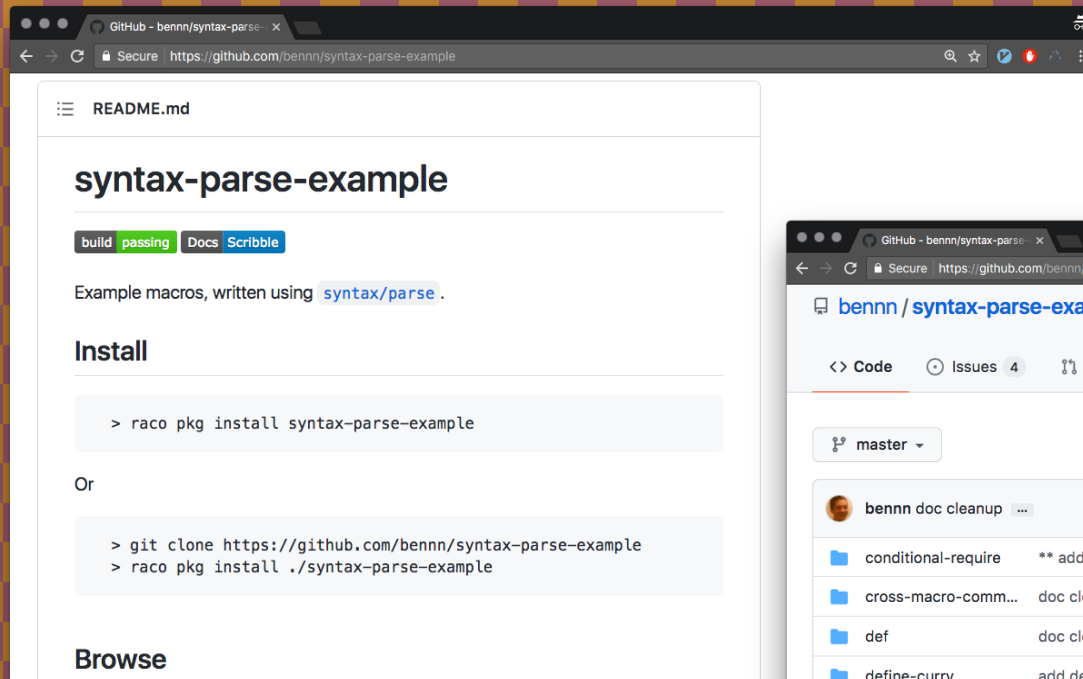
Shorthand to write contracts for functions that expect only keyword arguments.

Examples:

```
> (struct pumpkin [name weight color])
> (define/contract (make-pumpkin #:name name #:weight weight #:color [color "D
  (kw-ctc ([name] [weight natural?] [(color _) string?]) pumpkin?)
  (pumpkin name weight color))
```

Contribute!

github.com/syntax-objects/syntax-parse-example



GitHub - bennn/syntax-parse

Secure <https://github.com/bennn/syntax-parse-example>

☰ README.md

syntax-parse-example

build passing Docs Scribble

Example macros, written using [syntax/parse](#).

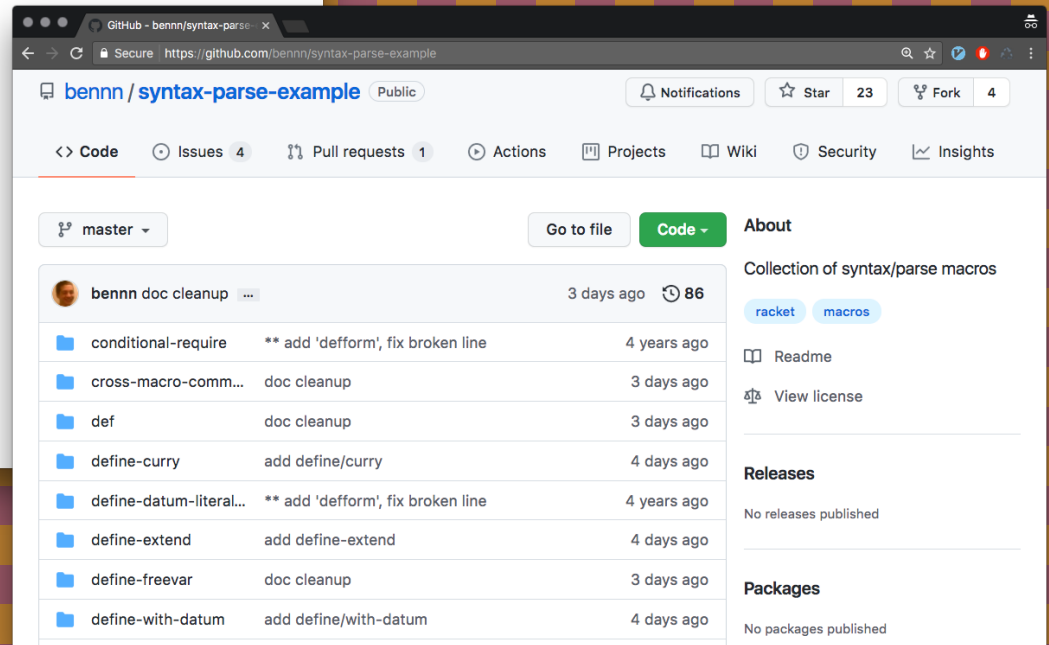
Install

```
> raco pkg install syntax-parse-example
```

Or

```
> git clone https://github.com/bennn/syntax-parse-example
> raco pkg install ./syntax-parse-example
```

Browse



GitHub - bennn/syntax-parse

Secure <https://github.com/bennn/syntax-parse-example>

bennn / [syntax-parse-example](#) Public

Notifications Star 23 Fork 4

<> Code Issues 4 Pull requests 1 Actions Projects Wiki Security Insights

master

Go to file Code

About

Collection of syntax/parse macros

racket macros

Readme

View license

Releases

No releases published

Packages

No packages published

bennn doc cleanup ...	3 days ago	86
conditional-require	** add 'defform', fix broken line	4 years ago
cross-macro-comm...	doc cleanup	3 days ago
def	doc cleanup	3 days ago
define-curry	add define/curry	4 days ago
define-datum-literal...	** add 'defform', fix broken line	4 years ago
define-extend	add define-extend	4 days ago
define-freevar	doc cleanup	3 days ago
define-with-datum	add define/with-datum	4 days ago

Legend of the Macro Pillow







ccs.neu.edu/home/pnkfelix/prl-tmp/gallery/index.html



Community Resource

docs.racket-lang.org/syntax-parse-example

Submissions

github.com/syntax-objects/Summer2021

Contribute!

github.com/syntax-objects/syntax-parse-example

Thank You!



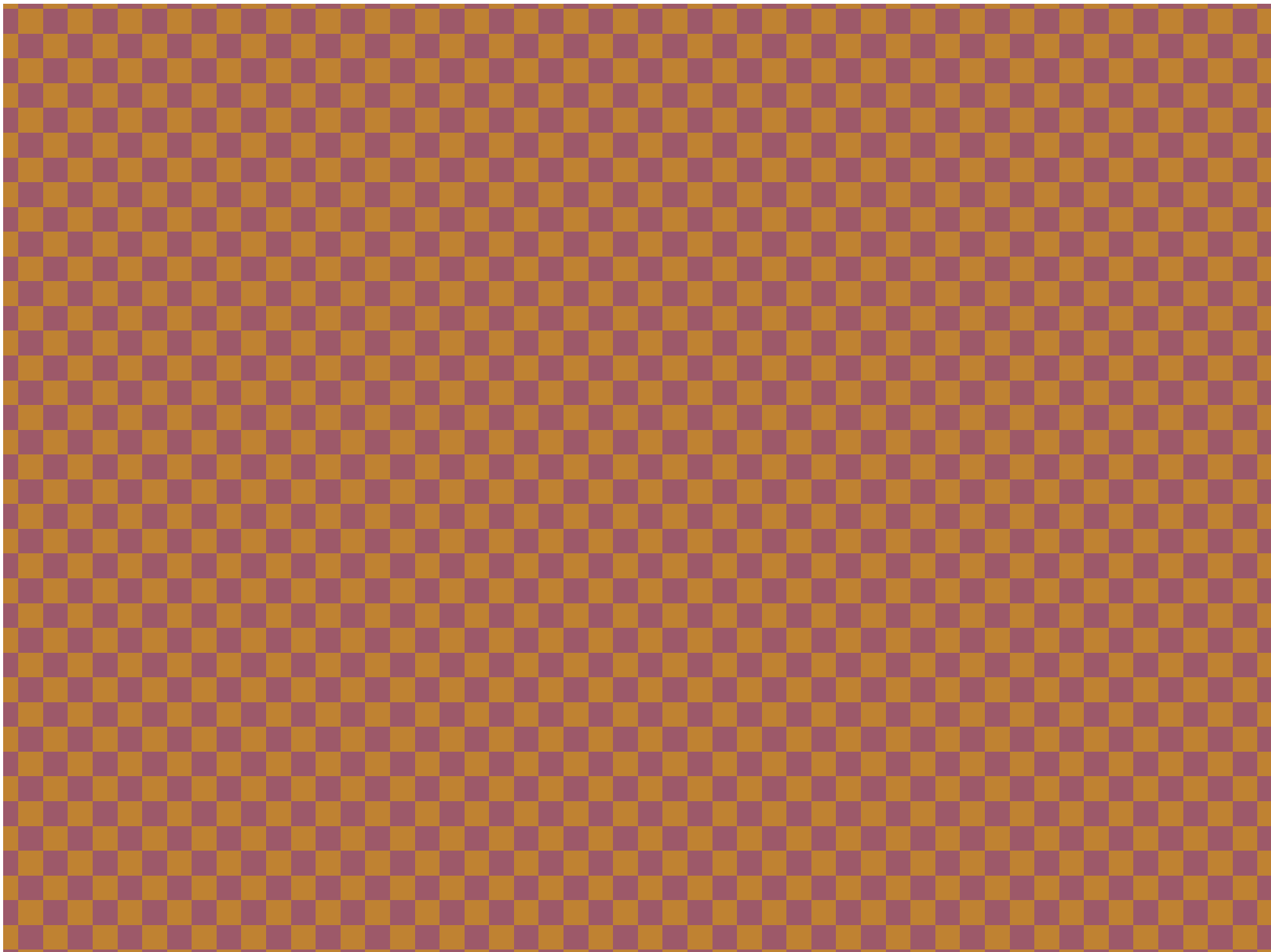
Stephen



Lazerbeak12345 hzafar agj Fictitious-Rotor xgqt benknoble AlexKnauth

sorawee eutro camoy soegaard Metaxal dstorrs countvajhula shhyou

Contributors





Community Resource

docs.racket-lang.org/syntax-parse-example

Submissions

github.com/syntax-objects/Summer2021

Contribute!

github.com/syntax-objects/syntax-parse-example