Structuur van Computerprogramma's I

Assistenten

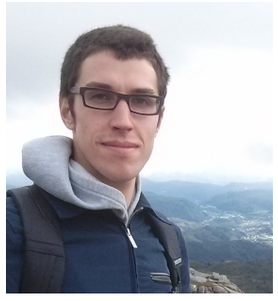
Sander Huyghebaert

Mathijs Saey

Jef Jacobs

Lin de Huybrecht

Youri Coppens



Ward Muylaert

Vragen? Problemen?

scpi@dinf.vub.ac.be

De WPOs

- ~ 3x per week, nadien ~1x per week
- Oefeningen op concepten gezien tijdens HOCs
- Naar de WPOs komen = opletten en meewerken

Taken

- 1 taak
- Verplicht voor studenten 1ste bachelor
- Optioneel voor schakel- en werkstudenten
- Feedback

Tussentijdse Evaluatie

- Verplicht voor 1ste bachelorstudenten
- Optioneel voor schakel- en werkstudenten
- Maandag 30/10, 09:00 - 12:00 (tentatief)
- 20% van eindresultaat als eindresultaat < tentamen

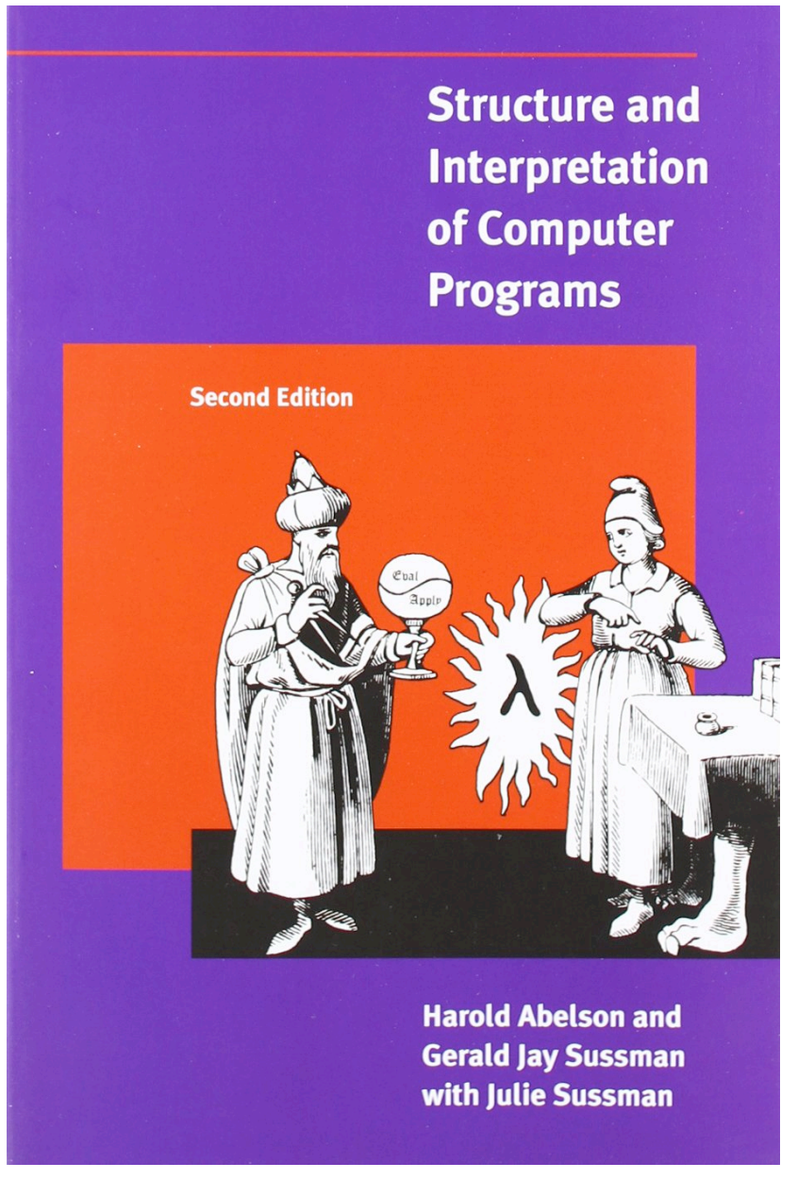
Examen

- Schriftelijk examen

  - Oefeningen + Theorie

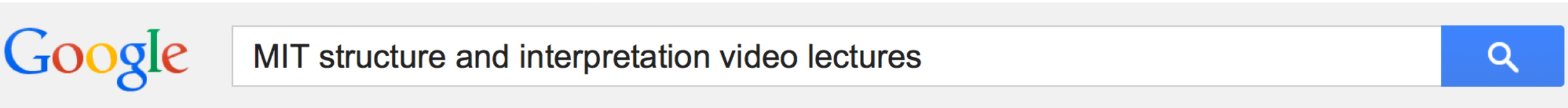
Het boek

- Gratis op de MIT website: https://mitpress.mit.edu/sicp/



Andere nuttige weetjes

- SBC: Vincent Luyten
- Infogroep (peter/meter)
- Opnames HOC
  - http://studio.infogroep.be
- Voorbeeldexamens
  - Beschikbaar op Canvas



Zaal PCs

- Activeer eigen account op https://wendy.vub.ac.be/accounts/
- Log in met guest/guest

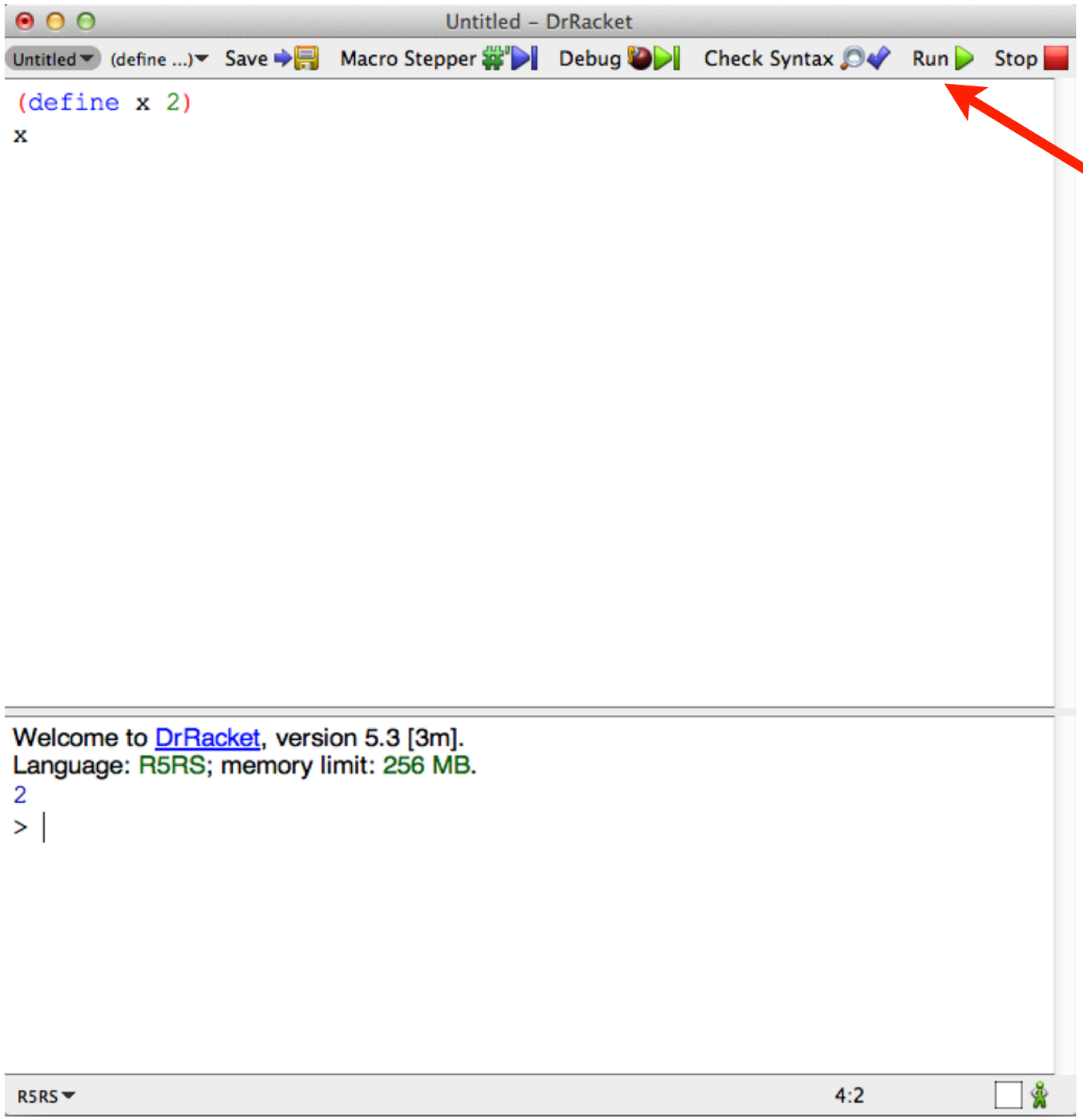
Eigen PC

- Download & installeer racket https://racket-lang.org

DrRacket

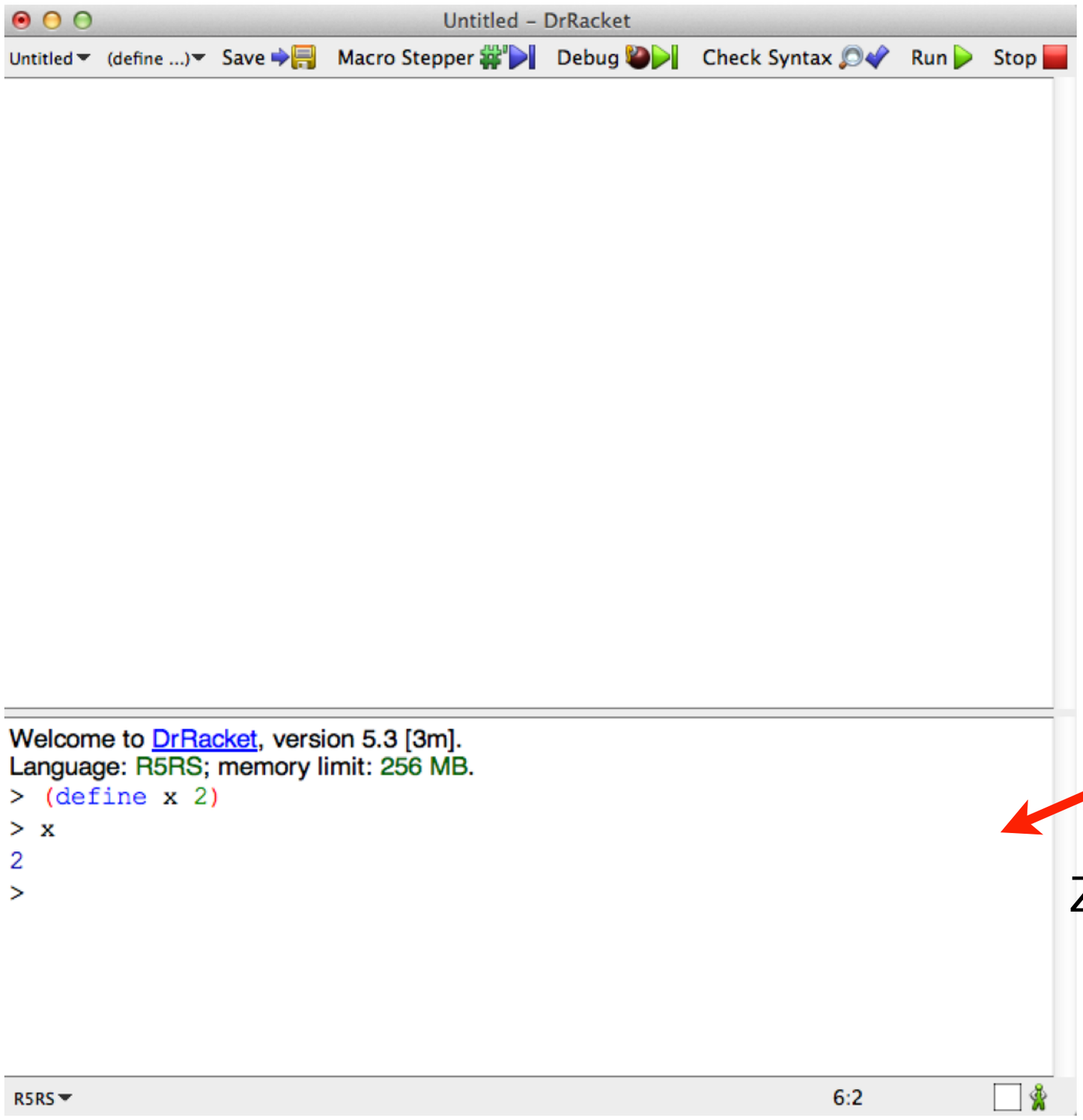
https://www.racket-lang.org

Definition window



Opslaan van je code

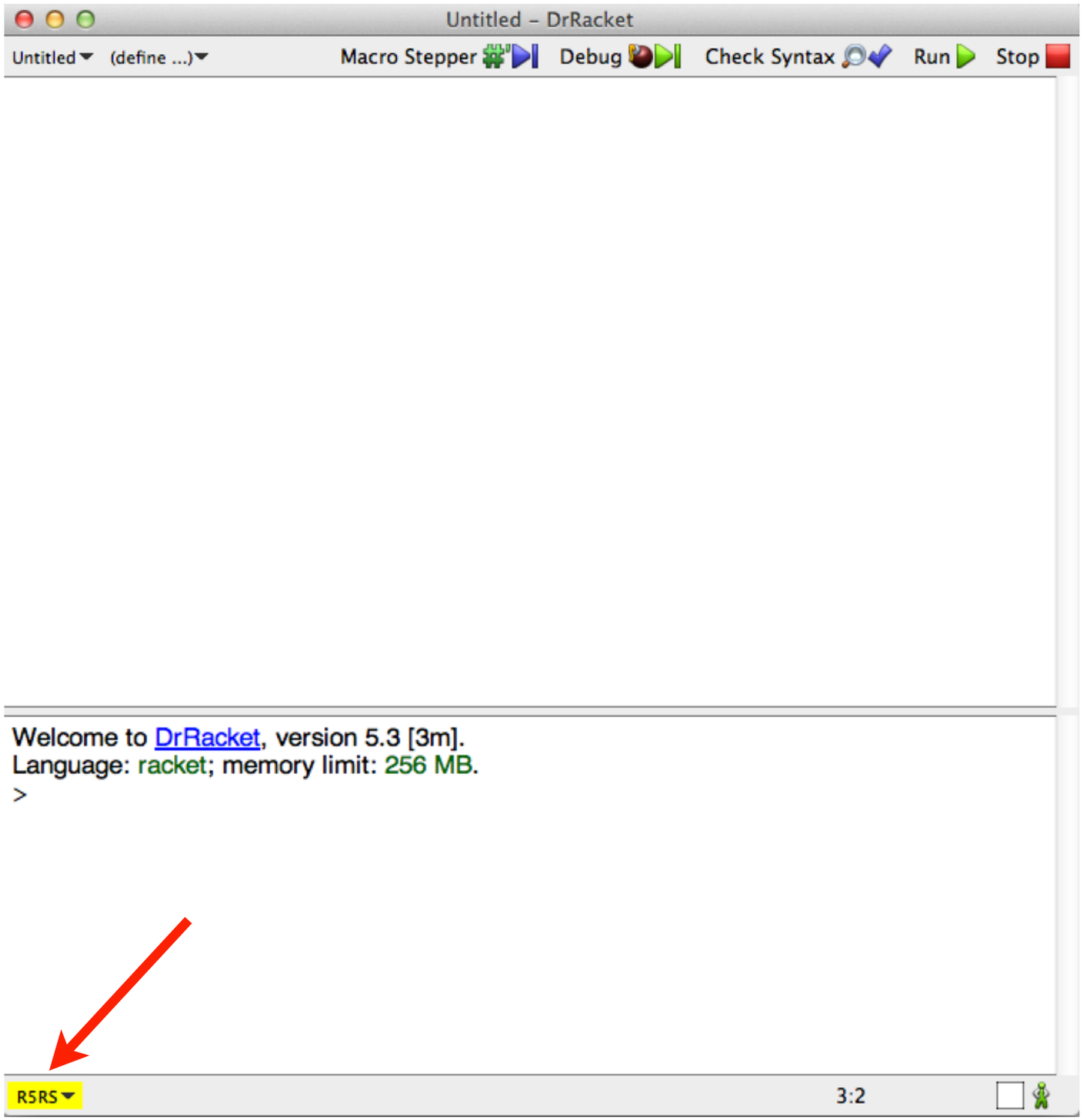
Interaction window

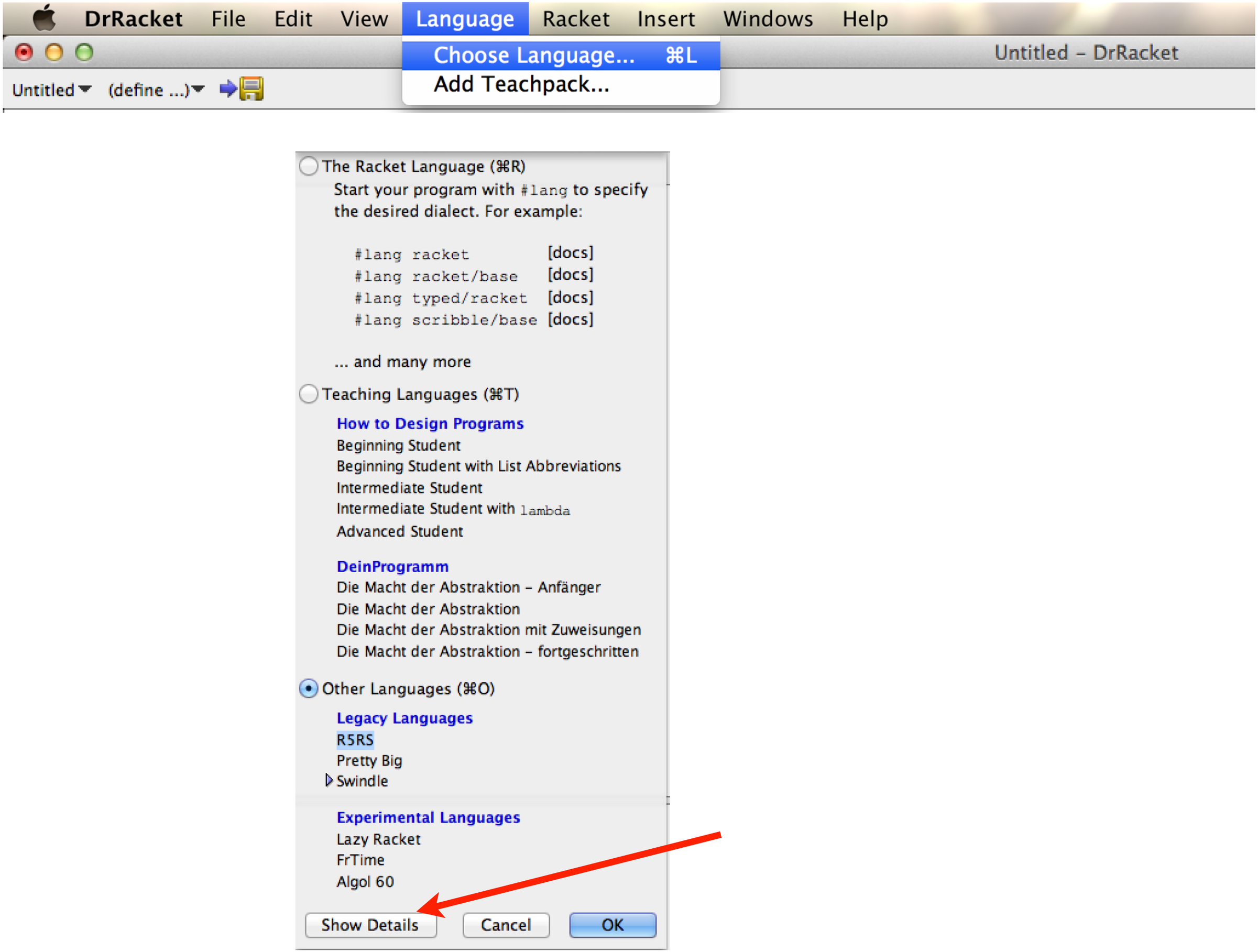


Read-Eval-Print loop

zo werkt Viviane in de les

DrRacket


Untitled – DrRacket
Untitled
(define ...)
Macro Stepper
Debug
Check Syntax
Run
Stop
Welcome to DrRacket, version 5.3 [3m].
Language: racket; memory limit: 256 MB.
>
R5RS
3:2


DrRacket File Edit View Language Racket Insert Windows Help
Choose Language... ⌘L
Add Teachpack...
Untitled – DrRacket
Untitled (define ...)
The Racket Language (⌘R)
Start your program with #lang to specify the desired dialect. For example:
#lang racket [docs]
#lang racket/base [docs]
#lang typed/racket [docs]
#lang scribble/base [docs]
... and many more
Teaching Languages (⌘T)
How to Design Programs
Beginning Student
Beginning Student with List Abbreviations
Intermediate Student
Intermediate Student with lambda
Advanced Student
DeinProgramm
Die Macht der Abstraktion – Anfänger
Die Macht der Abstraktion
Die Macht der Abstraktion mit Zuweisungen
Die Macht der Abstraktion – fortgeschritten
Other Languages (⌘O)
Legacy Languages
R5RS
Pretty Big
Swindle
Experimental Languages
Lazy Racket
FrTime
Algol 60
Show Details
Cancel
OK

The Racket Language (⌘R)

Start your program with `#lang` to specify the desired dialect. For example:

```
#lang racket          [docs]
#lang racket/base     [docs]
#lang typed/racket    [docs]
#lang scribble/base   [docs]
```

... and many more

Teaching Languages (⌘T)

How to Design Programs
Beginning Student
Beginning Student with List Abbreviations
Intermediate Student
Intermediate Student with `lambda`
Advanced Student

DeinProgramm
Die Macht der Abstraktion – Anfänger
Die Macht der Abstraktion
Die Macht der Abstraktion mit Zuweisungen
Die Macht der Abstraktion – fortgeschritten

Other Languages (⌘O)

Legacy Languages
R5RS
Pretty Big
Swindle

Experimental Languages
Lazy Racket
FrTime
Algol 60

Input Syntax
- [ ] Case sensitive

Dynamic Properties
- ( ) No debugging or profiling
- ( ) Debugging and profiling
- (•) Debugging
- ( ) Syntactic test suite coverage

Output Syntax

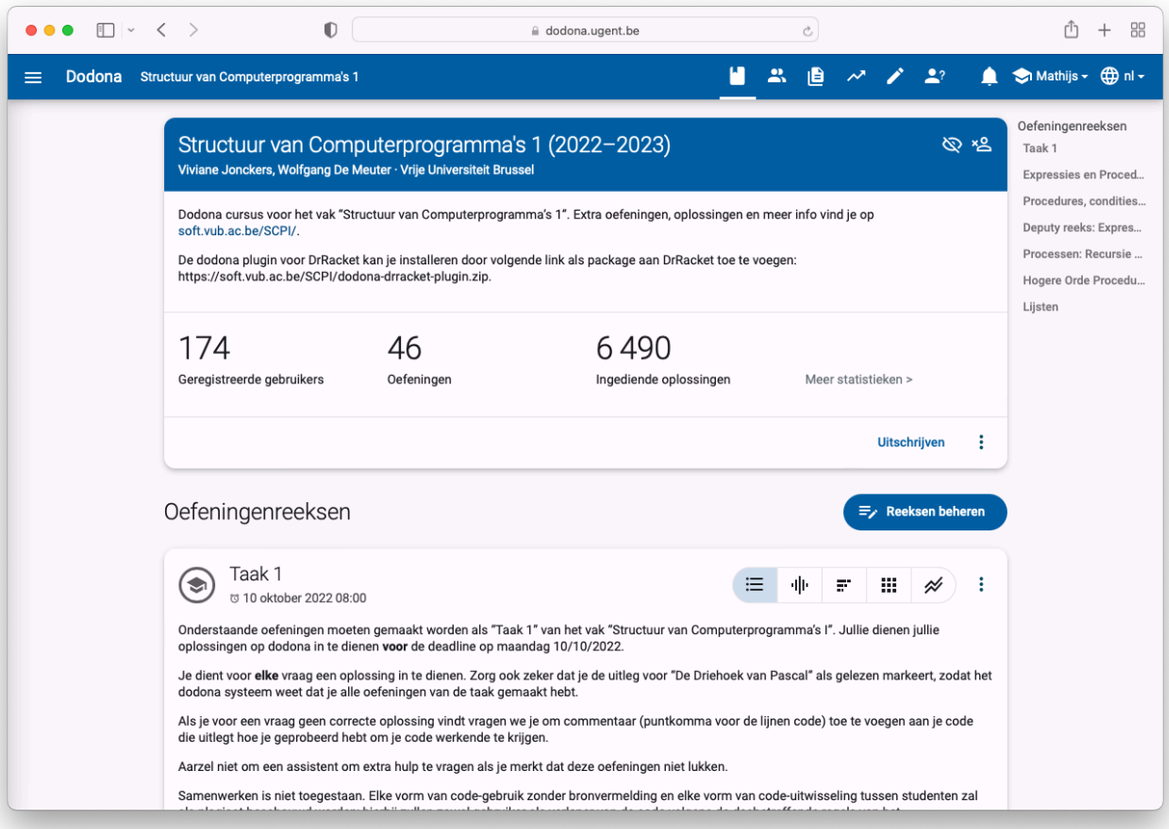
Output Style
- ( ) Constructor
- ( ) Quasiquote
- (•) write
- ( ) print

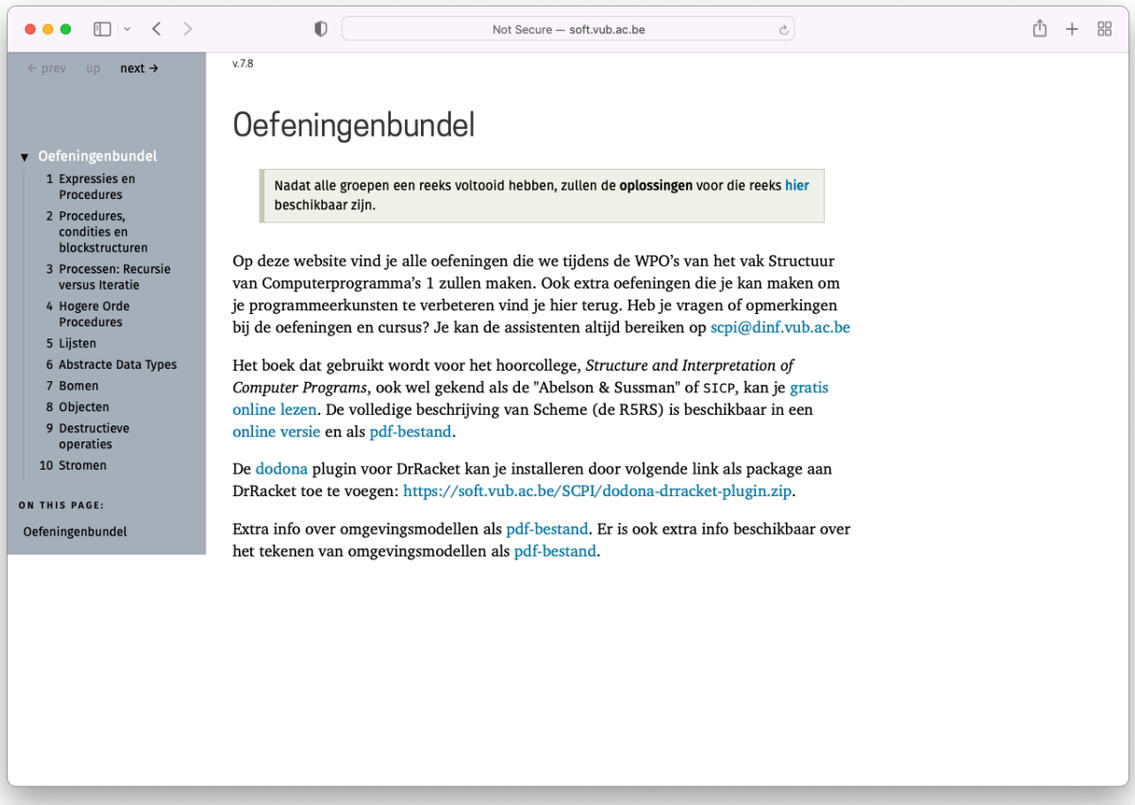
- [ ] Show sharing in values
- [x] Insert newlines in printed values
- [ ] Use decimal notation for rationals

Initial Bindings
- [ ] Disallow redefinition of initial bindings

Hide Details | Revert to Language Defaults | Cancel | OK

Opgaven





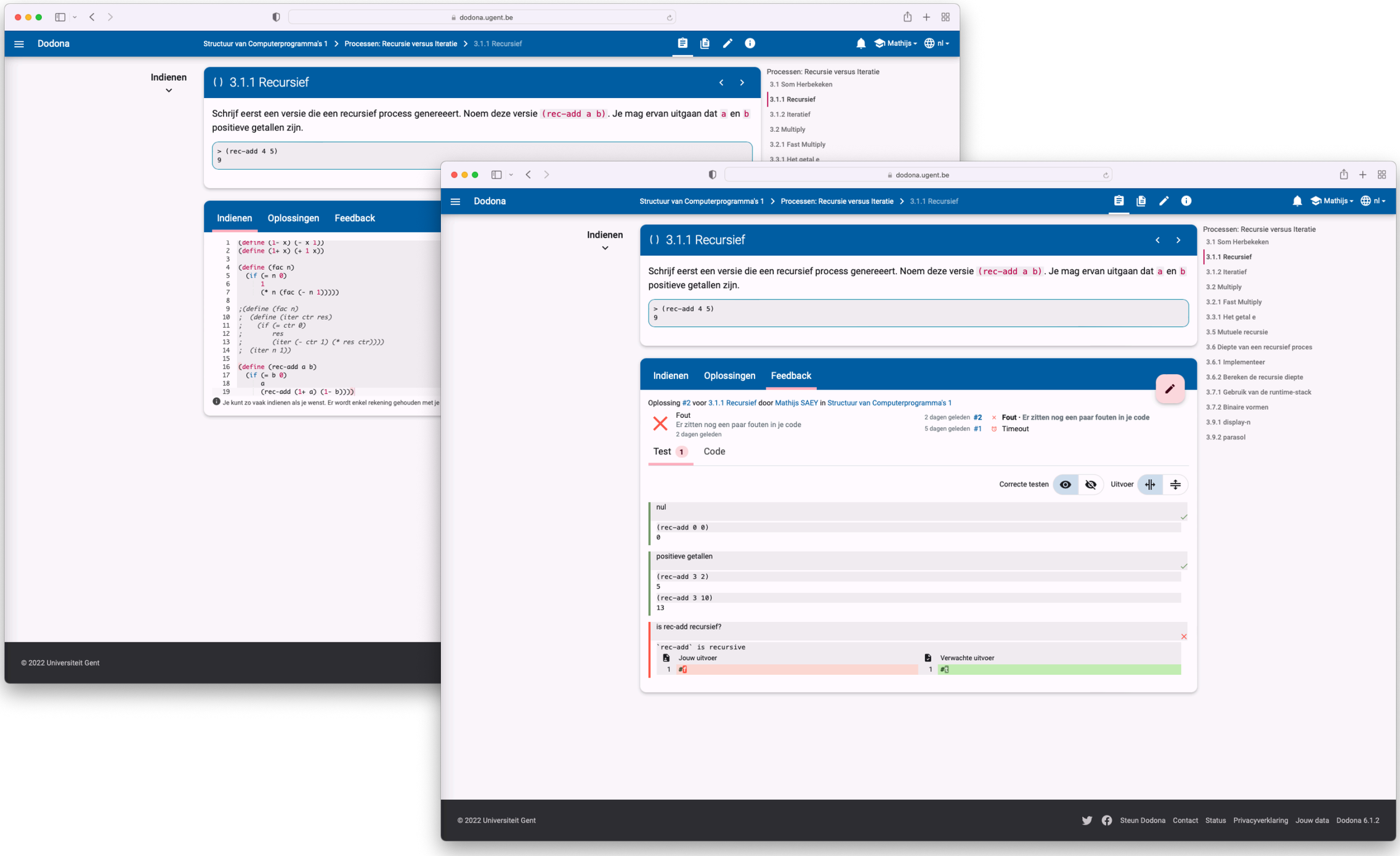
dodona.ugent.be

- Opgaven die we tijdens de les bekijken
- Taken

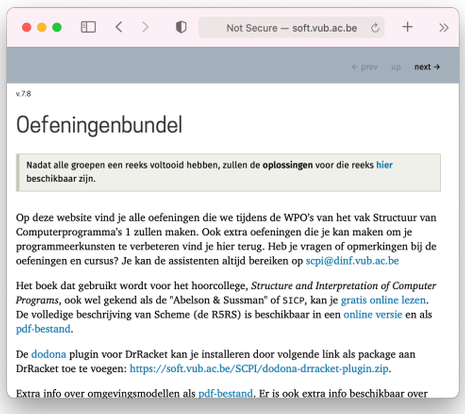
soft.vub.ac.be/SCPI

- Alle opgaven
- Oplossingen

Dodona


dodona.ugent.be
Dodona
Structuur van Computerprogramma's 1 > Processen: Recursie versus Iteratie > 3.1.1 Recursief
Mathijs
nl
Indienen
3.1.1 Recursief
Schrijf eerst een versie die een recursief process genereeert. Noem deze versie (rec-add a b). Je mag ervan uitgaan dat a en b positieve getallen zijn.
> (rec-add 4 5)
9
Indienen
Oplossingen
Feedback
(define (1- x) (- x 1))
(define (1+ x) (+ 1 x))
(define (fac n)
(if (= n 0)
1
(* n (fac (- n 1)))))
;(define (fac n)
; (define (iter ctr res)
; (if (= ctr 0)
; res
; (iter (- ctr 1) (* res ctr))))
; (iter n 1))
(define (rec-add a b)
(if (= b 0)
a
(rec-add (1+ a) (1- b))))
Je kunt zo vaak indienen als je wenst. Er wordt enkel rekening gehouden met je
Processen: Recursie versus Iteratie
3.1 Som Herbekeken
3.1.1 Recursief
3.1.2 Iteratief
3.2 Multiply
3.2.1 Fast Multiply
© 2022 Universiteit Gent
dodona.ugent.be
Dodona
Structuur van Computerprogramma's 1 > Processen: Recursie versus Iteratie > 3.1.1 Recursief
Mathijs
nl
Indienen
3.1.1 Recursief
Schrijf eerst een versie die een recursief process genereeert. Noem deze versie (rec-add a b). Je mag ervan uitgaan dat a en b positieve getallen zijn.
> (rec-add 4 5)
9
Indienen
Oplossingen
Feedback
Oplossing #2 voor 3.1.1 Recursief door Mathijs SAEY in Structuur van Computerprogramma's 1
Fout
Er zitten nog een paar fouten in je code
2 dagen geleden
2 dagen geleden #2 Fout · Er zitten nog een paar fouten in je code
5 dagen geleden #1 Timeout
Test 1
Code
Correcte testen
Uitvoer
nul
(rec-add 0 0)
0
positieve getallen
(rec-add 3 2)
5
(rec-add 3 10)
13
is rec-add recursief?
`rec-add` is recursive
Jouw uitvoer
Verwachte uitvoer
Processen: Recursie versus Iteratie
3.1 Som Herbekeken
3.1.1 Recursief
3.1.2 Iteratief
3.2 Multiply
3.2.1 Fast Multiply
3.3.1 Het getal e
3.5 Mutuele recursie
3.6 Diepte van een recursief proces
3.6.1 Implementeer
3.6.2 Bereken de recursie diepte
3.7.1 Gebruik van de runtime-stack
3.7.2 Binaire vormen
3.9.1 display-n
3.9.2 parasol
© 2022 Universiteit Gent
Steun Dodona
Contact
Status
Privacyverklaring
Jouw data
Dodona 6.1.2

Dodona

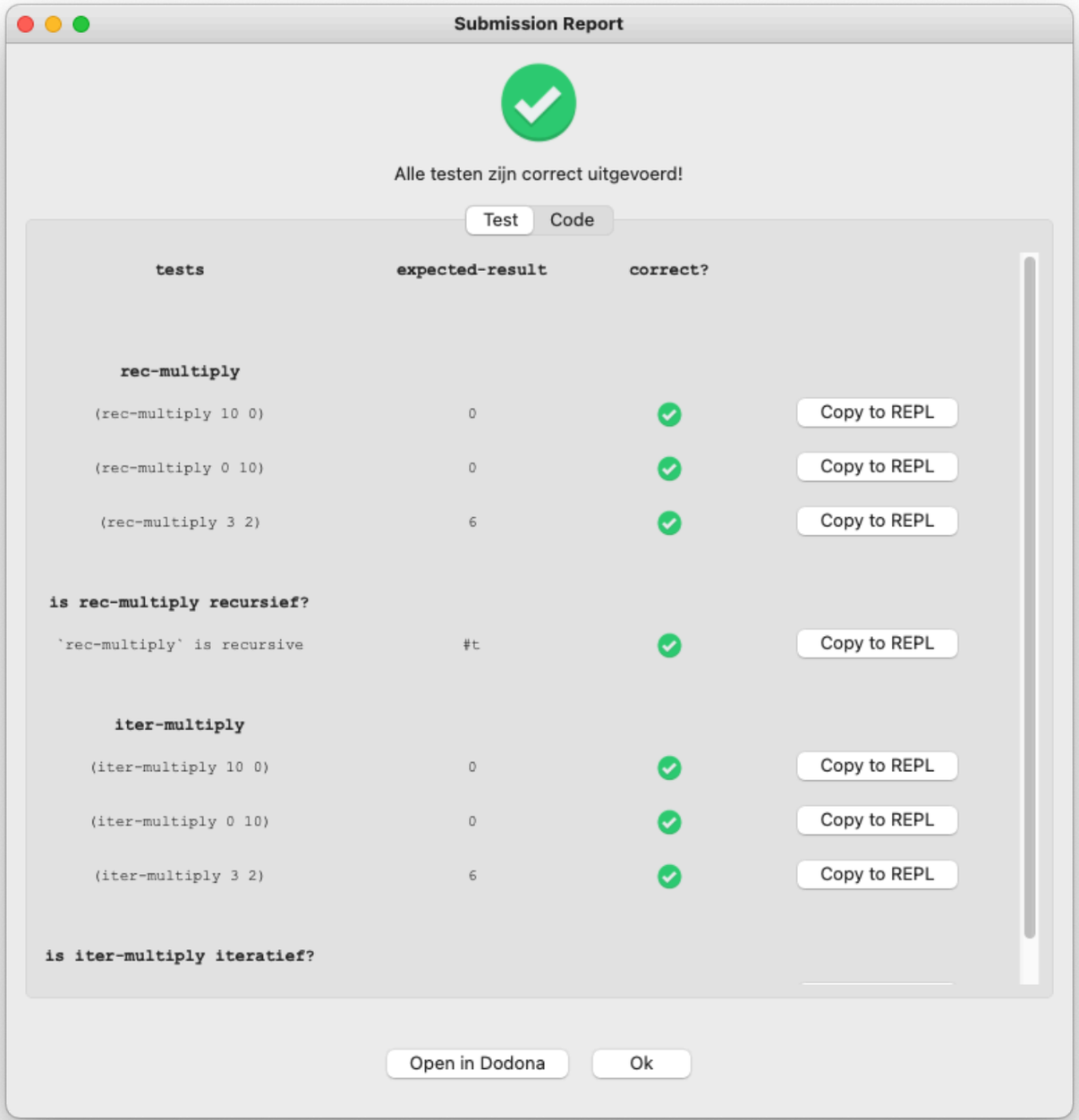


REPL

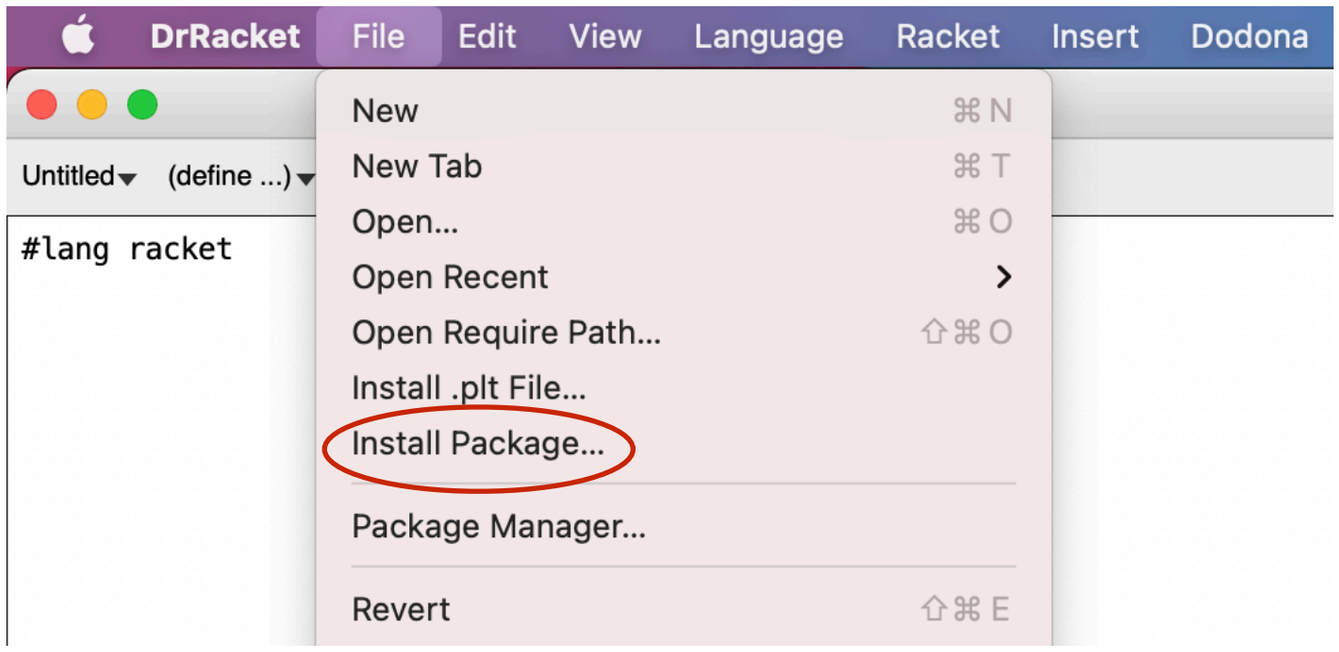
Is mijn code correct?

Ja/Nee + Testinvoer

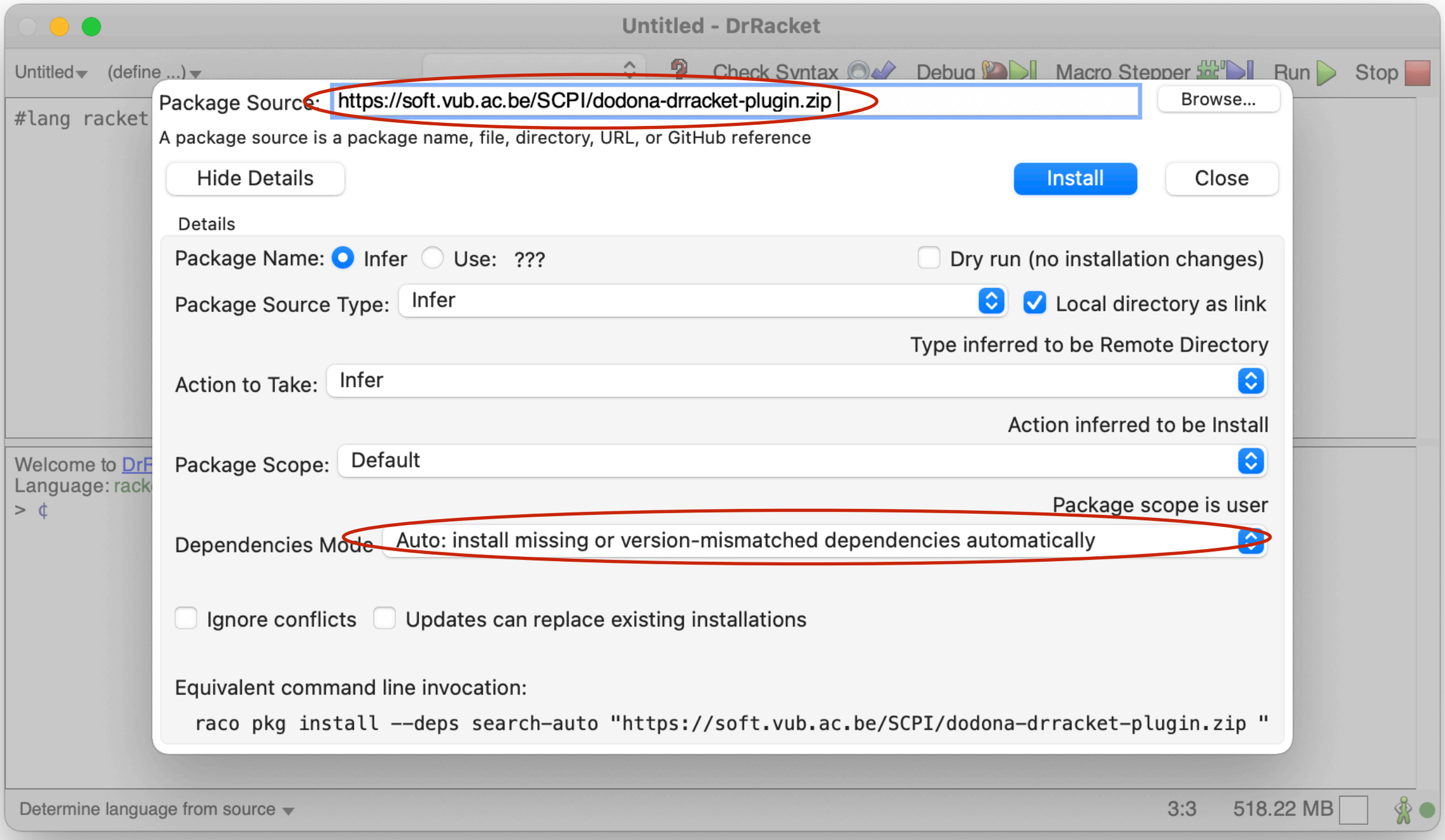
Dodona Plugin



Dodona Plugin



Dodona Plugin



Dodona Plugin