

Structuur van Computerprogramma's I

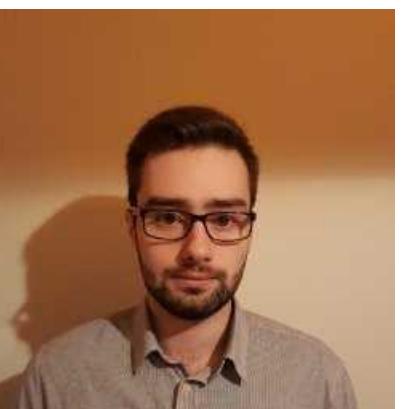
Het onderwijsteam



Viviane Jonckers



Wolfgang De Meuter



Sander Huyghebaert



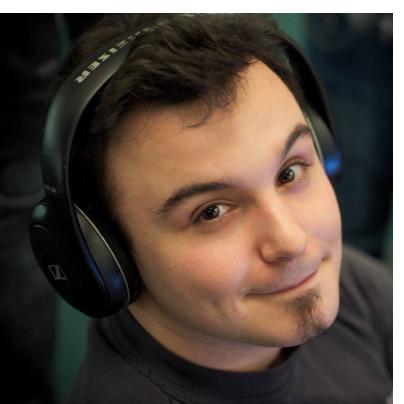
Mathijs Saey



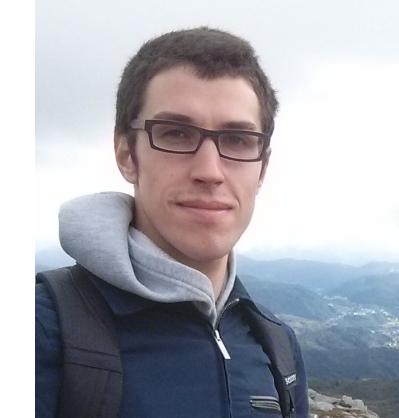
Jef Jacobs



Lin de Huybrecht



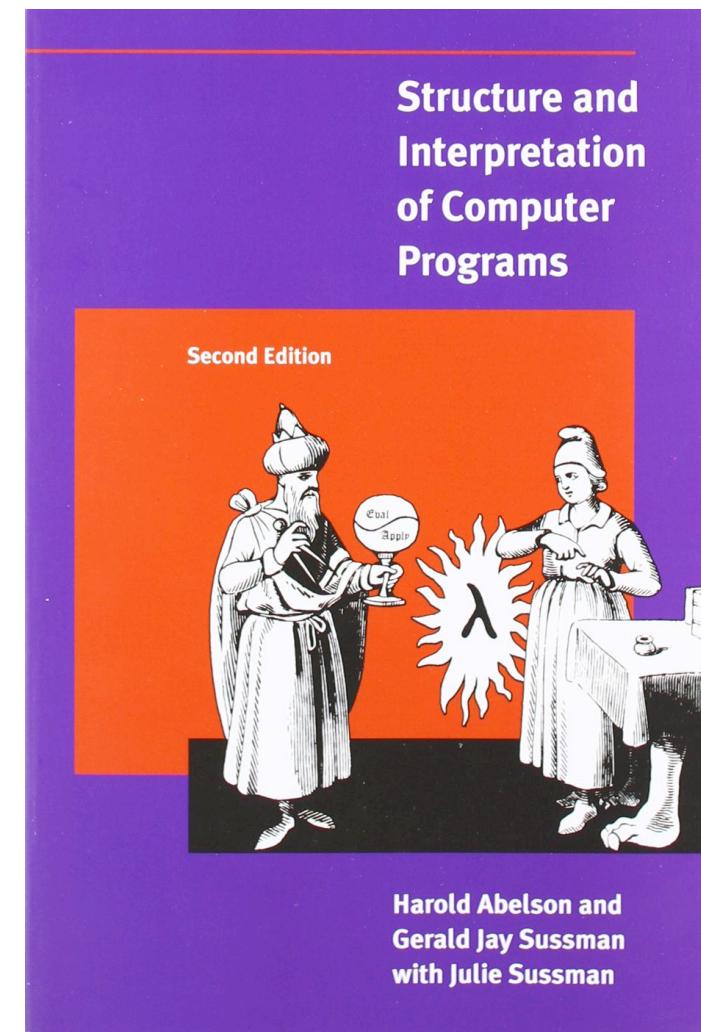
Youri Coppens



Ward Muylaert

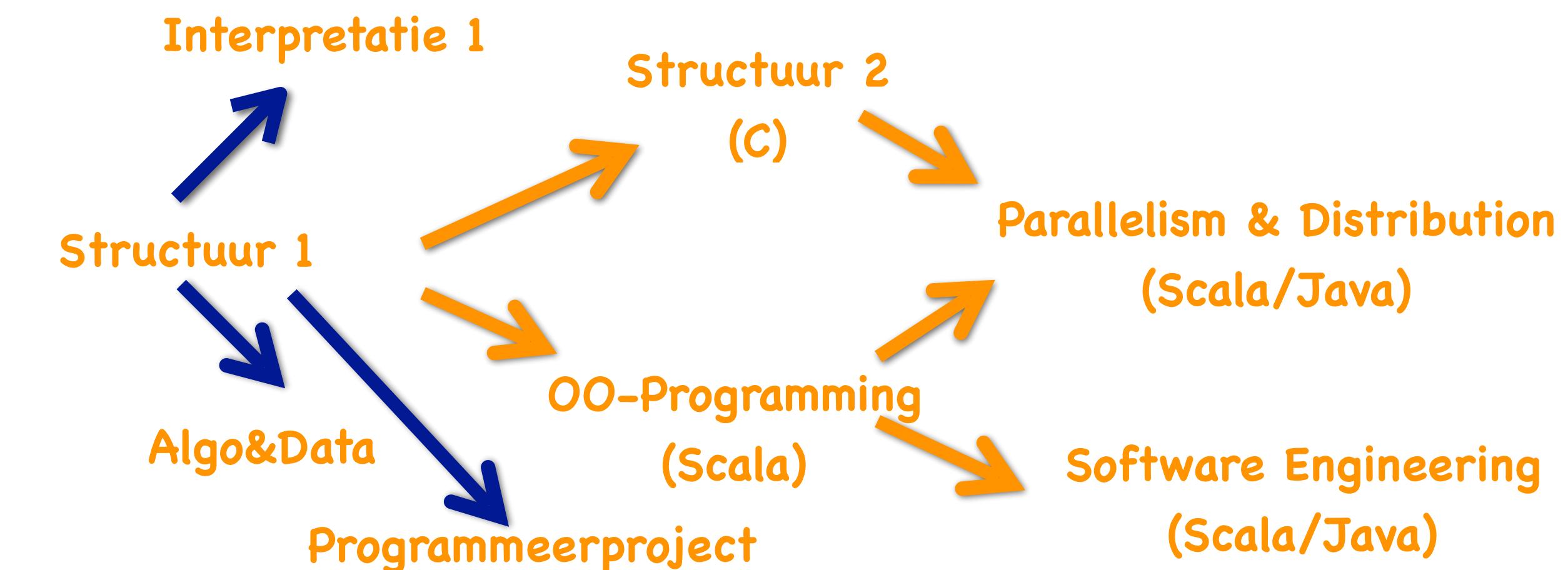
De cursusinhoud

- Waarom die titel?
 - ✿ inleiding programmeren
 - ✿ inleiding programmeertalen
 - ✿ complementair aan Interpretatie van Computerprogramma's 1



- Waarom Scheme?
 - ✿ nieuw voor (bijna) iedereen
 - ✿ programmeren in veel verschillende stijlen
 - ✿ beheersbaar volume

- Plaats in het curriculum



Vragen? Problemen?

scpi@dinf.vub.ac.be

questions@dinf.vub.ac.be

De HOCs

- Uitleg van basiskennis en concepten
- Zelfstudie
 - slides
 - video's op panopto (canvas)

De WPOs

- Oefeningen op concepten gezien tijdens HOCs
- Zelf maken
 - Opgave online beschikbaar
 - Extra uitleg van oefeningen op panopto
 - Evaluatie van oplossingen via Dodona
- Oplossingen
 - Online beschikbaar
 - Extra uitleg over oplossingen op panopto

De Clinics

- Online Q&A Sessies
- Gepland voor de tussentijdse evaluatie en voor het examen
 - Dinsdag 24/10
 - Dinsdag 12/12

Taken

- 1 taak
- Optioneel voor werkstudenten
- Feedback

Tussentijdse Evaluatie

- Schriftelijk
 - Oefeningen + Theorie
- Optioneel voor werkstudenten
- Maandag 30/10, 09:00 - 12:00 (tentatief)
- 20% van eindresultaat als eindresultaat < tentamen
- Voorbeeld later beschikbaar op canvas

Examen

- Schriftelijk examen
 - Oefeningen + Theorie
- Voorbeeldexamen later beschikbaar op Canvas

De HOCs

- Uitleg van basiskennis en concepten
- Zelfstudie
 - slides
 - video's op panopto (canvas)

De HOCs: Slides

The screenshot shows the VUB Canvas LMS interface for the course "Structuur van computerprogramma's 1 - 000086". The left sidebar contains links for Account, Dashboard, Cursussen, Kalender, Inbox, Geschiedenis, and Help. The main content area displays course modules:

- Week 2 HOC maandag:**
 - Les00-Intro.pdf
 - Les01-Basiselementen.pdf
 - Opname Les0-Intro
 - Opname Les1-Basiselementen-Sessie1
 - Opname Les1-Basiselementen-Sessie2
- Week 2 HOC dinsdag:**
 - Les02-LokaleProcedures.pdf
 - Opname Les2-LokaleProcedures-Sessie1
 - Opname Les2-LokaleProcedures-Sessie2
- Week 2 HOC donderdag:**
 - Les03-Procedures&Processen.pdf (tot slide 21)
 - Opname Les3-Procedures&Processen-Sessie1
 - Opname Les3-Procedures&Processen-Sessie2
- Week 3 HOC maandag:**

A large orange box highlights a hand-drawn diagram illustrating a linear iterative process for calculating factorial using a helper function `fac-bis`:

Een lineair iteratief proces voor faculteit

The diagram shows the recursive calls for $fac-bis(5, 1)$:

- $5 * fac-bis(4, 5)$
- $4 * fac-bis(3, 4)$
- $3 * fac-bis(2, 3)$
- $2 * fac-bis(1, 2)$
- $1 * fac-bis(0, 1)$

Annotations explain the process:

- "bij elke oproep wordt een tussenresultaat opgebouwd"
- "de eindconditie wordt bereikt"
- "het eindresultaat staat klaar"
- "na terugkeer wordt er 'NIETS' meer gedaan met het resultaat (= **staartrecursie**)"

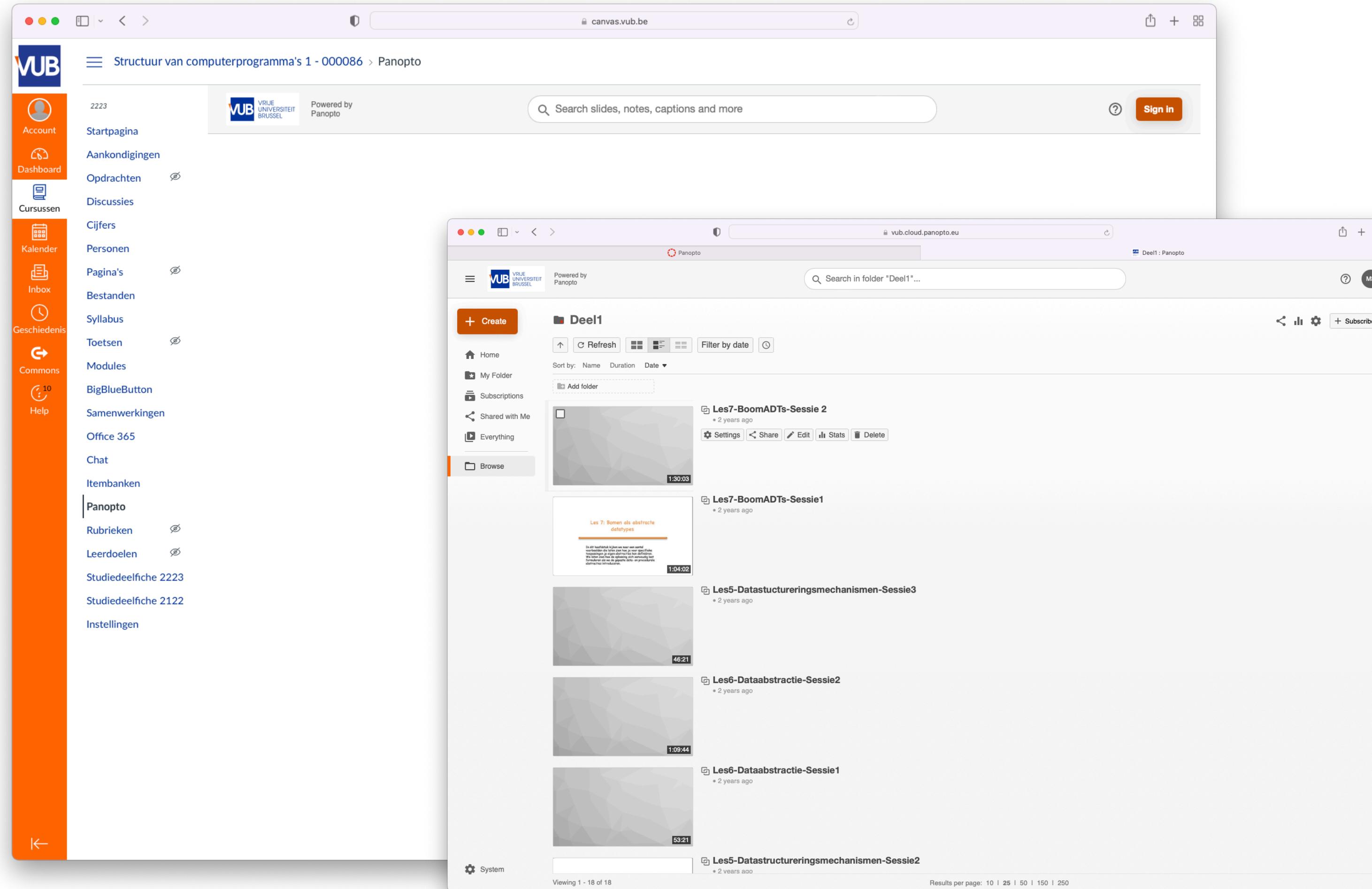
A blue arrow labeled "CODE" points to the following Scheme code:

```
(define (fac n)
  (define (fac-iter counter result)
    (if (= counter 0)
        result
        (fac-iter (- counter 1) (* counter result))))
  (fac-iter n 1))
```

Output: > (fac 5)
120

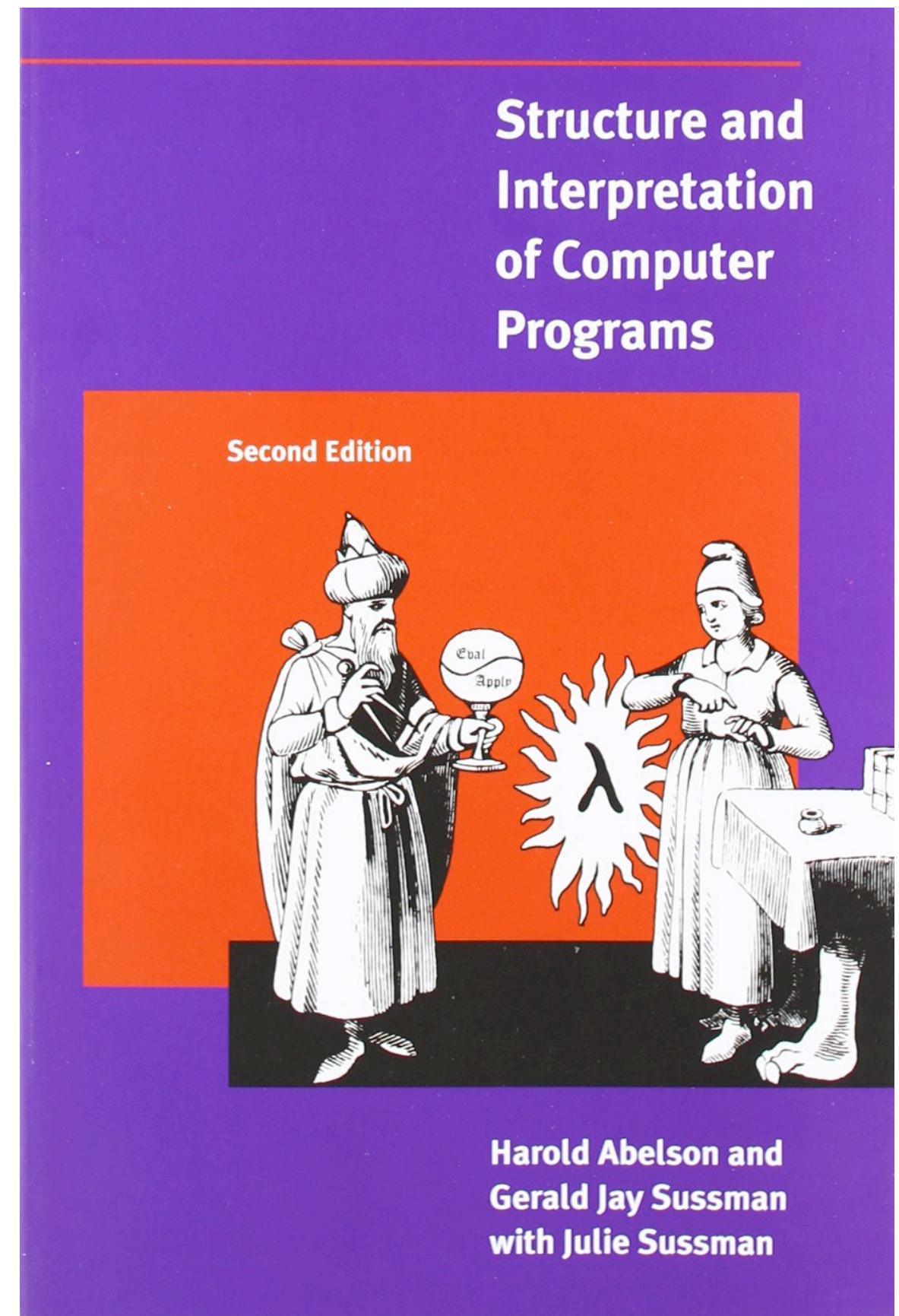
Page number: 8

De HOCs: Video's



Het boek

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



De WPOs

- Oefeningen op concepten gezien tijdens HOCs
- Zelf maken
 - Opgave online beschikbaar
 - Extra uitleg van oefeningen op panopto
 - Evaluatie van oplossingen via Dodona
- Oplossingen
 - Online beschikbaar
 - Extra uitleg over oplossingen op panopto

De WPOs

The screenshot shows the Dodona platform interface. At the top, it displays the course title 'Structuur van Computerprogramma's 1 (2022–2023)' and the authors 'Viviane Jonckers, Wolfgang De Meuter - Vrije Universiteit Brussel'. Below this, there are statistics: 174 registered users, 46 assignments, and 6 490 submitted solutions. A sidebar on the right lists assignment categories like 'Oefeningenreeksen', 'Taal 1', 'Expressies en Procedures...', etc. The main content area shows a specific assignment titled 'Taak 1' due on 10 oktober 2022 08:00. It contains instructions and a text area for students to enter their code.

The screenshot shows the SCPI website's 'Oefeningenbundel' page. It features a sidebar with a list of topics: Expressies en Procedures, Procedures, condities en blockstructuren, Processen Recursie versus Iteratie, Hogere Orde Procedures, Lijsten, Abstrakte Data Types, Bomen, Objecten, Destructieve operaties, and Stromen. The main content area includes a note about solutions becoming available after all groups complete the assignments, and links to the book 'Structure and Interpretation of Computer Programs' and its Scheme implementation. It also provides information on the DrRacket plugin and PDF versions of the material.

dodona.ugent.be

- Opgaven die we tijdens de les bekijken
- Testen van jullie oplossingen
- Taken

soft.vub.ac.be/SCPI

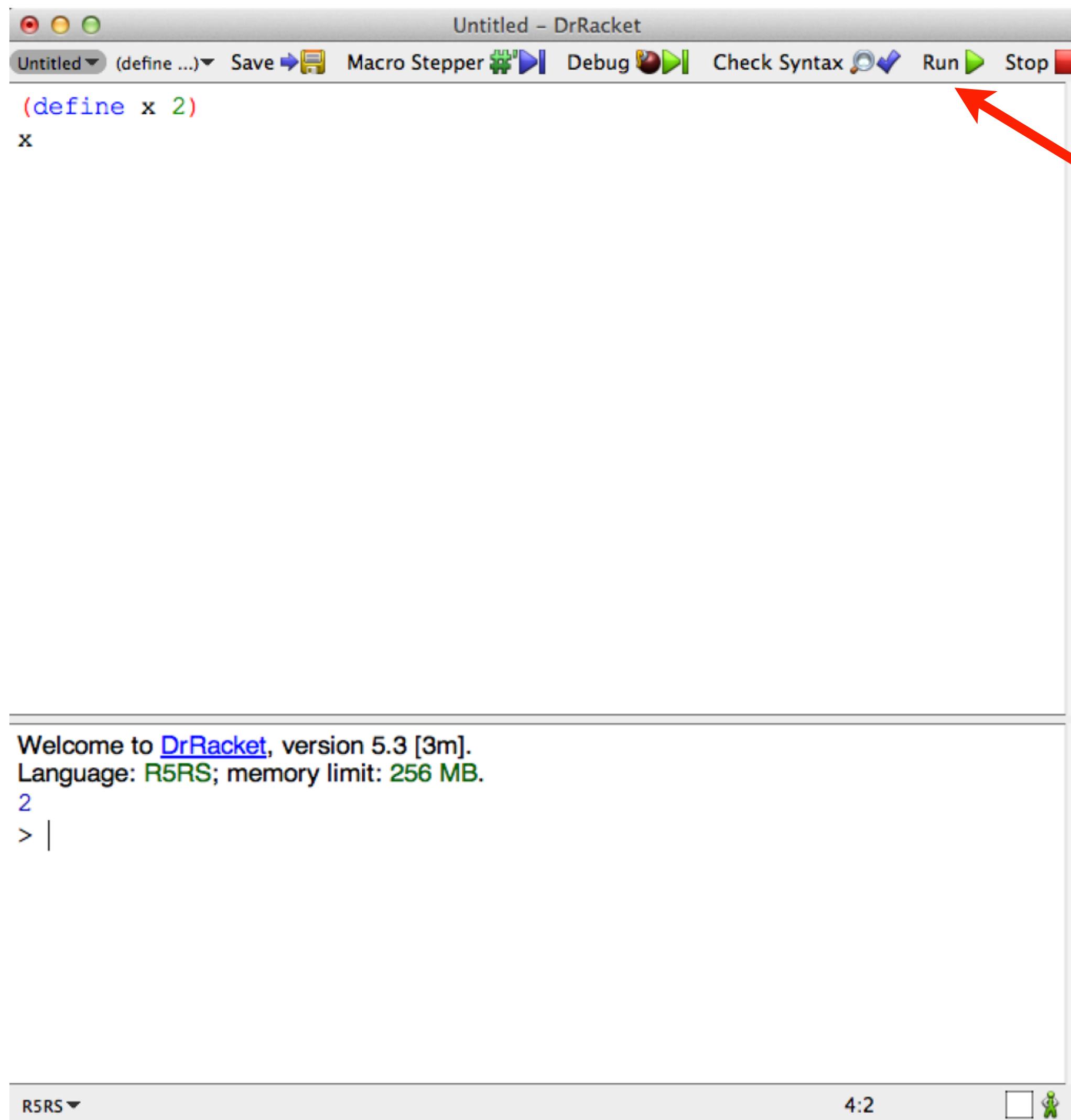
- Alle opgaven
- Oplossingen

DrRacket

<https://www.racket-lang.org>

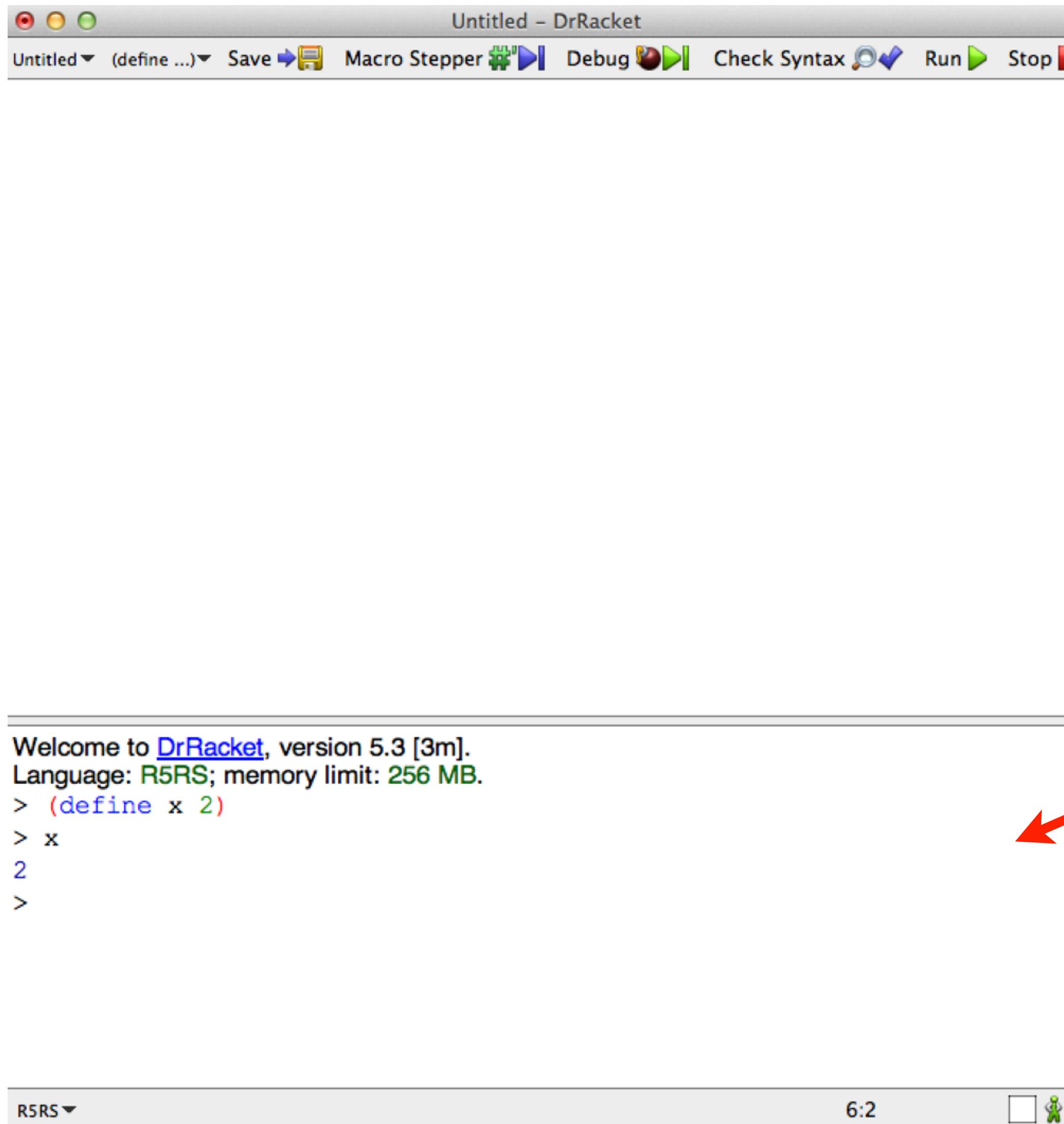


Definition window



Opslaan van je code

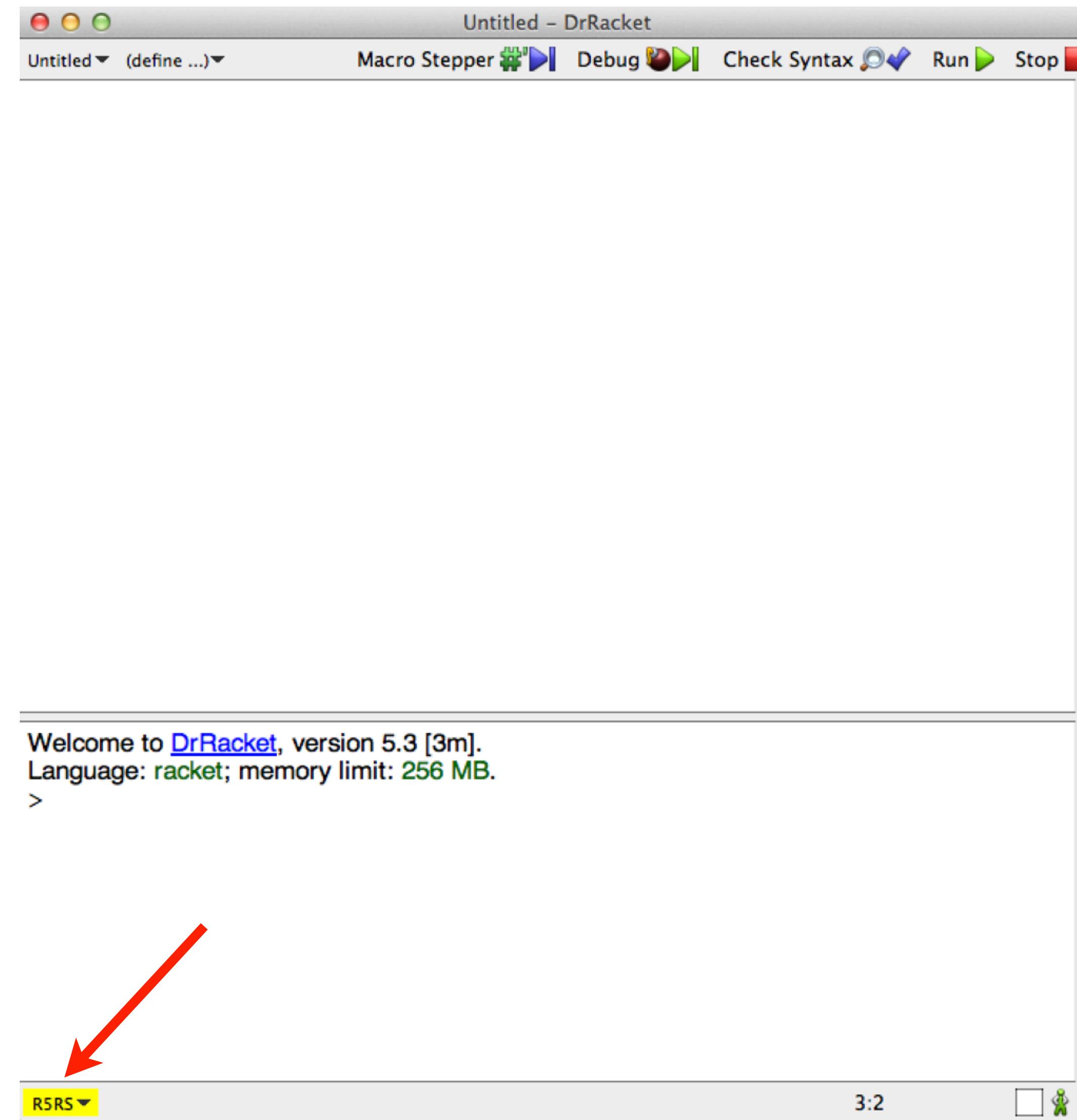
Interaction window

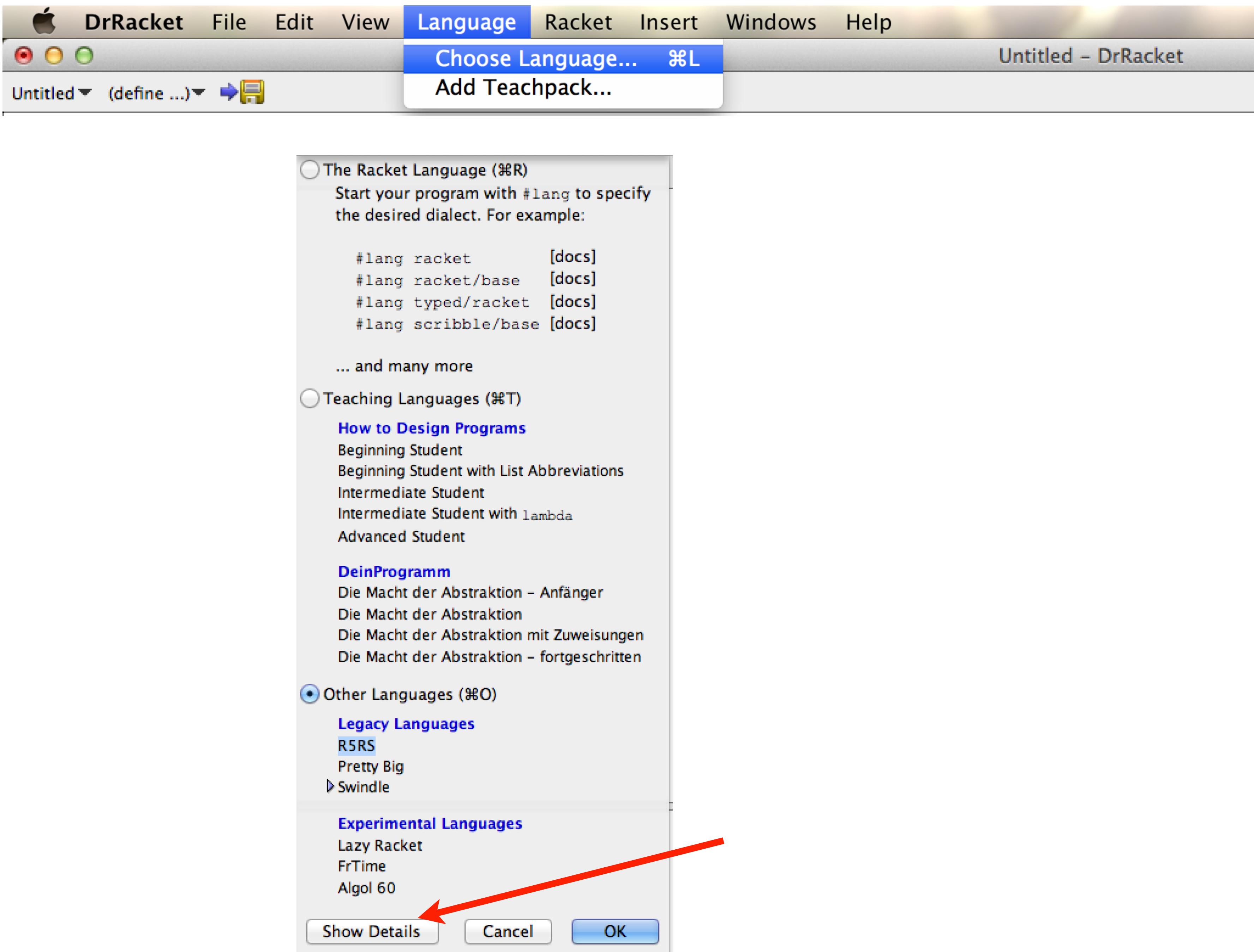


Read-Eval-Print loop

zo werkt Viviane in de les

DrRacket





- 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

Macro Debugger  

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
 Insert newlines in printed values
 Use decimal notation for rationals

Initial Bindings
 Disallow redefinition of initial bindings



Dodona

The image displays two side-by-side screenshots of the Dodona platform, a web-based tool for teaching and assessing computer programming assignments.

Screenshot 1 (Left): Shows a student's submission for a recursive addition function. The code provided is:

```
> (rec-add 4 5)
9
```

The student has also included a more complex factorial function:

```
1 (define (1- x) (- x 1))
2 (define (1+ x) (+ 1 x))
3
4 (define (fac n)
5   (if (= n 0)
6       1
7       (* (fac (- n 1))))))
8
9 ;(define (fac n)
10 ;  (define (iter ctr res)
11 ;    (if (= ctr 0)
12 ;        res
13 ;        (iter (- ctr 1) (* res ctr)))))
14 ;  (iter n 1))
15
16 (define (rec-add a b)
17   (if (= b 0)
18       a
19       (rec-add (+ a 1) (- b)))))
```

A note at the bottom of the code editor says: "Je kunt zo vaak indienen als je wenst. Er wordt enkel rekening gehouden met je laatste indiening."

Screenshot 2 (Right): Shows the same assignment page with a different student's submission. The code is identical to the one in Screenshot 1.

The feedback section shows a red error message: "Fout · Er zitten nog een paar fouten in je code". Below it, a note says: "2 dagen geleden #2 Fout · Er zitten nog een paar fouten in je code".

The test results show two successful tests:

- nul:** (rec-add 0 0)
0
- positieve getallen:** (rec-add 3 2)
5
(rec-add 3 10)
13

In the "is rec-add recursief?" section, the student's answer is marked as "incorrect". The expected output was "#1", but the student's output was "#0".

Both screenshots include a sidebar with a navigation menu:

- Structuur van Computerprogramma's 1 > Procesken: Recursie versus Iteratie > 3.1.1 Recursief
- Indienen
- Oplossingen
- Feedback
- Procesken: Recursie versus Iteratie
 - 3.1 Som Herbekken
 - 3.1.1 Recursief
 - 3.1.2 Iteratief
 - 3.2 Multiply
 - 3.2.1 Fast Multiply
 - 3.3 Het getal e

Dodona Plugin



Submission Report

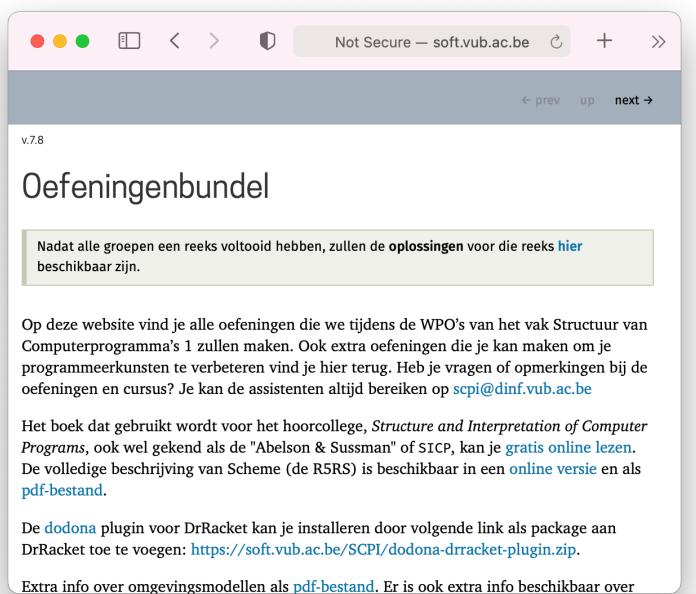
Alle testen zijn correct uitgevoerd!

Test **Code**

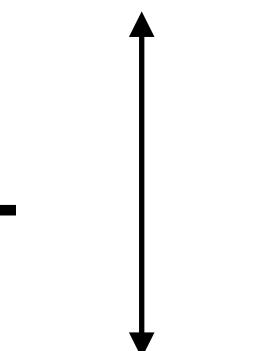
tests	expected-result	correct?	
rec-multiply			
(rec-multiply 10 0)	0	✓	Copy to REPL
(rec-multiply 0 10)	0	✓	Copy to REPL
(rec-multiply 3 2)	6	✓	Copy to REPL
is rec-multiply recursief?			
'rec-multiply' is recursive	#t	✓	Copy to REPL
iter-multiply			
(iter-multiply 10 0)	0	✓	Copy to REPL
(iter-multiply 0 10)	0	✓	Copy to REPL
(iter-multiply 3 2)	6	✓	Copy to REPL
is iter-multiply iteratief?			

[Open in Dodona](#) [Ok](#)

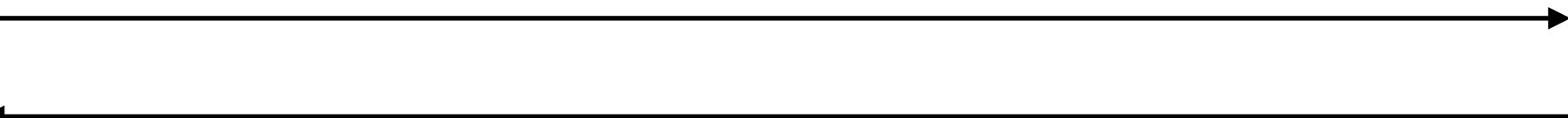
Dodona



REPL



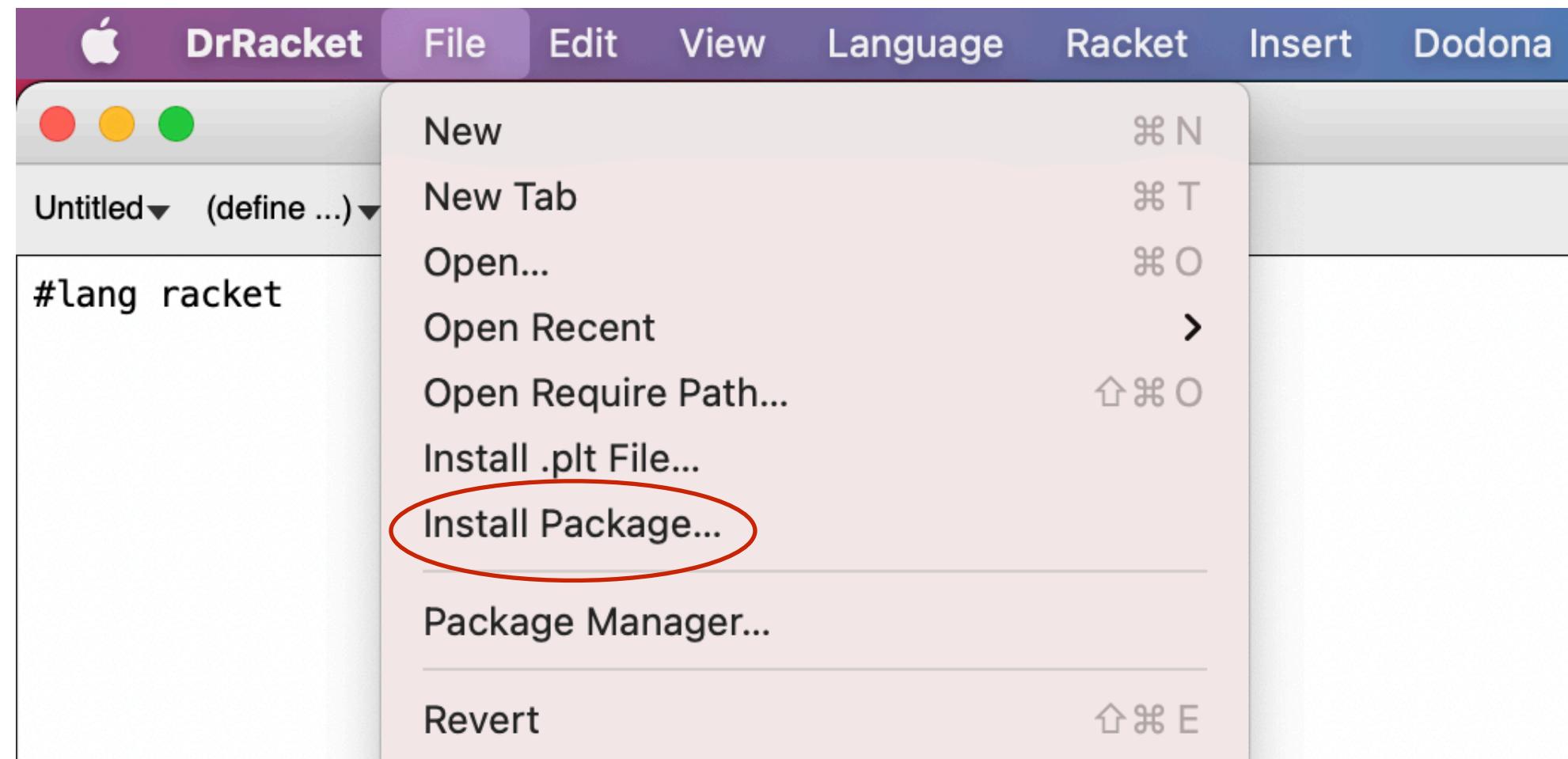
Is mijn code correct?



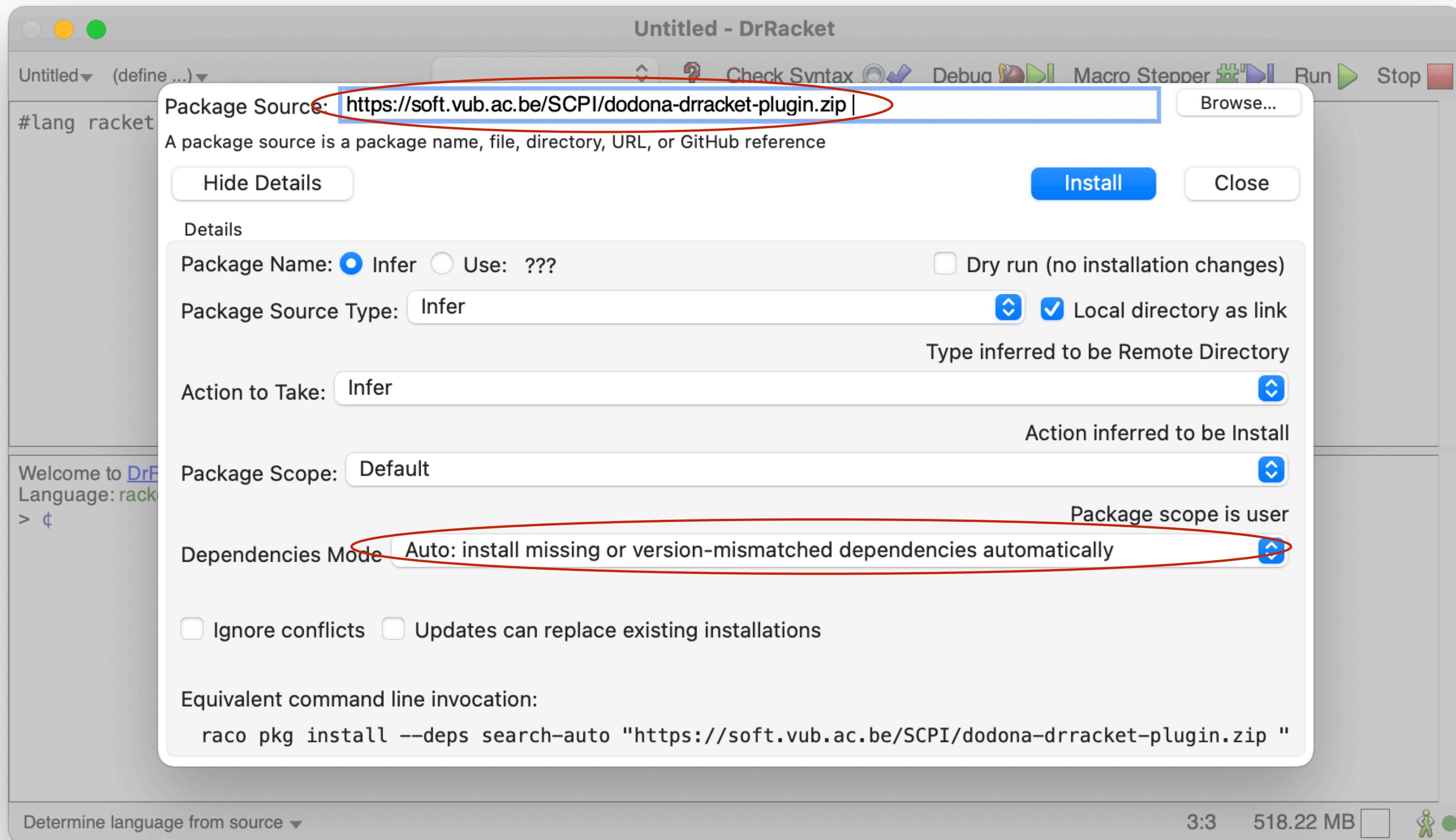
Ja/Nee + Testinvoer



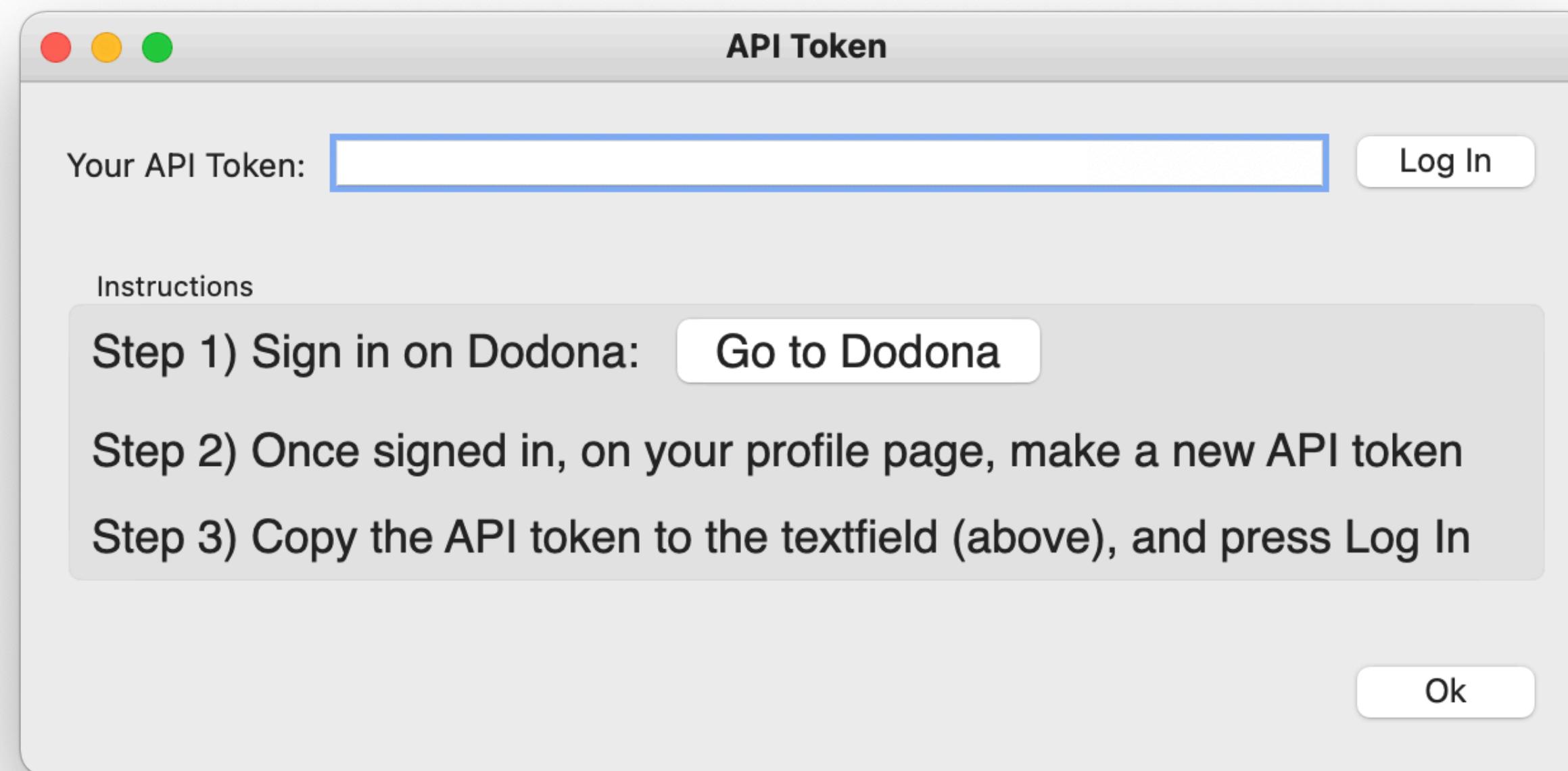
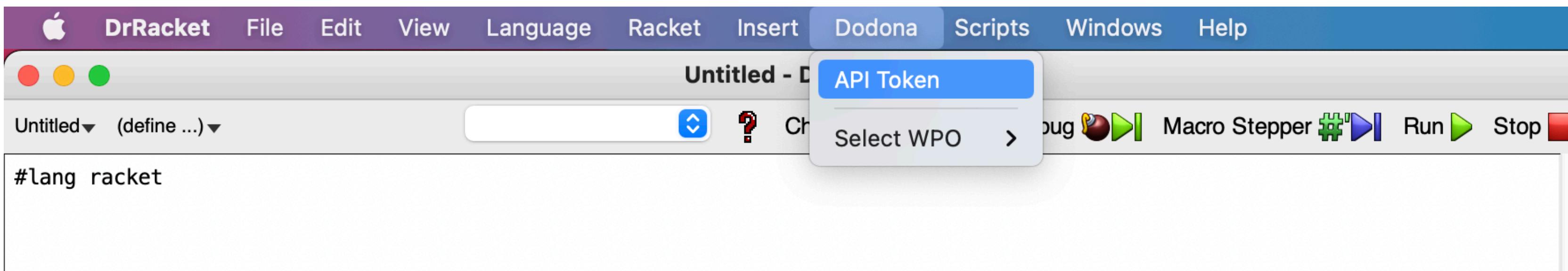
Dodona Plugin



Dodona Plugin



Dodona Plugin



Structuur van Computerprogramma's 1

- Theorie
 - Canvas (slides)
 - Panopto (videos)
- **Oefeningen**
 - Dodona (opgave, testen)
 - Oefeningenbundel (extra opgaven, antwoorden)
 - Panopto (videos met extra uitleg)
- Vragen
 - Clinics
 - scpi@dinf.vub.ac.be
 - Discord deputies