

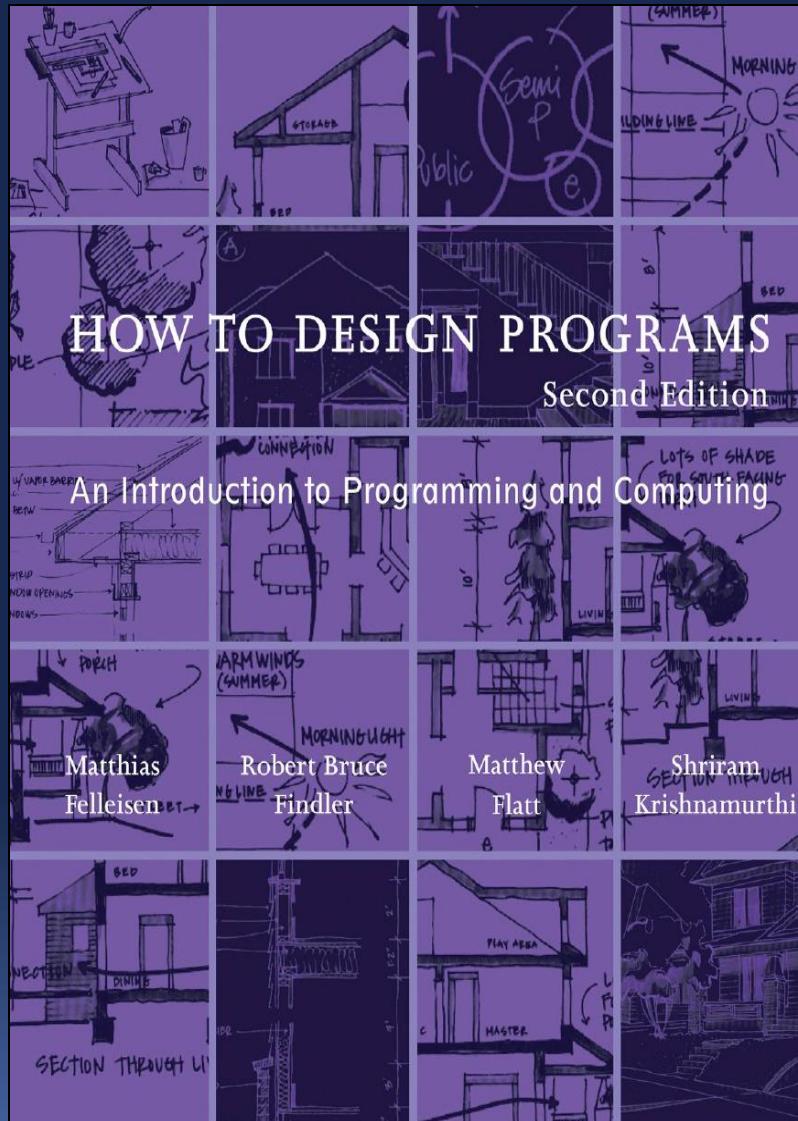
Programação Funcional

Unidade 2 – Programação Funcional com a Linguagem Racket



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecidovfreitas@gmail.com

Bibliografia





Racket

[DONATE](#)[DOCS](#)[PACKAGES](#)[DOWNLOAD](#)

Racket version 7.8 is available.

tenth RacketCon will be held online in October.

Racket, the Programming Language

Mature

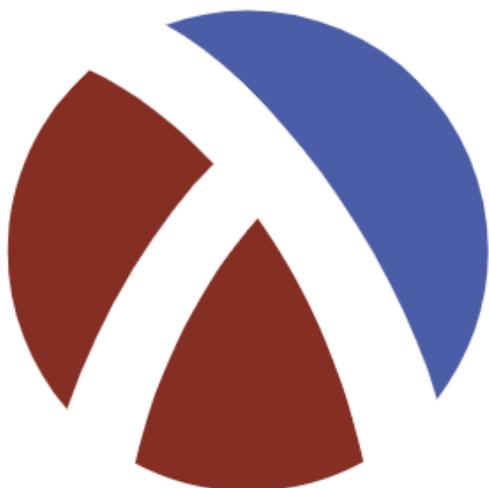
Jet Fueled

Extensible

Robust

Polished

Vibrant Community



```
#lang racket/gui

(define my-language 'English)

(define translations
  #hash([English . "Hello world"]
        [French . "Bonjour le monde"]
        [German . "Hallo Welt"]
        [Greek . "Γειά σου, κόσμε"]
        [Portuguese . "Olá mundo"]
        [Spanish . "Hola mundo"]
        [Thai . "สวัสดีชาวโลก"]))

(define my-hello-world
  (hash-ref translations my-language
            "hello world"))

(message-box "" my-hello-world)
```

Racket, the Ecosystem

[Software](#)[Tutorials & Documentation](#)[Community](#)[Books](#)[Education](#)[Swag](#)

docs.racket-lang.org

v.7.8

...search manuals...

Racket Documentation

Getting Started
Racket Cheat Sheet

Tutorials
[Quick: An Introduction to Racket with Pictures](#)
[Continue: Web Applications in Racket](#)
[More: Systems Programming with Racket](#)

Racket Language and Core Libraries
[The Racket Guide](#)
[The Racket Reference](#)

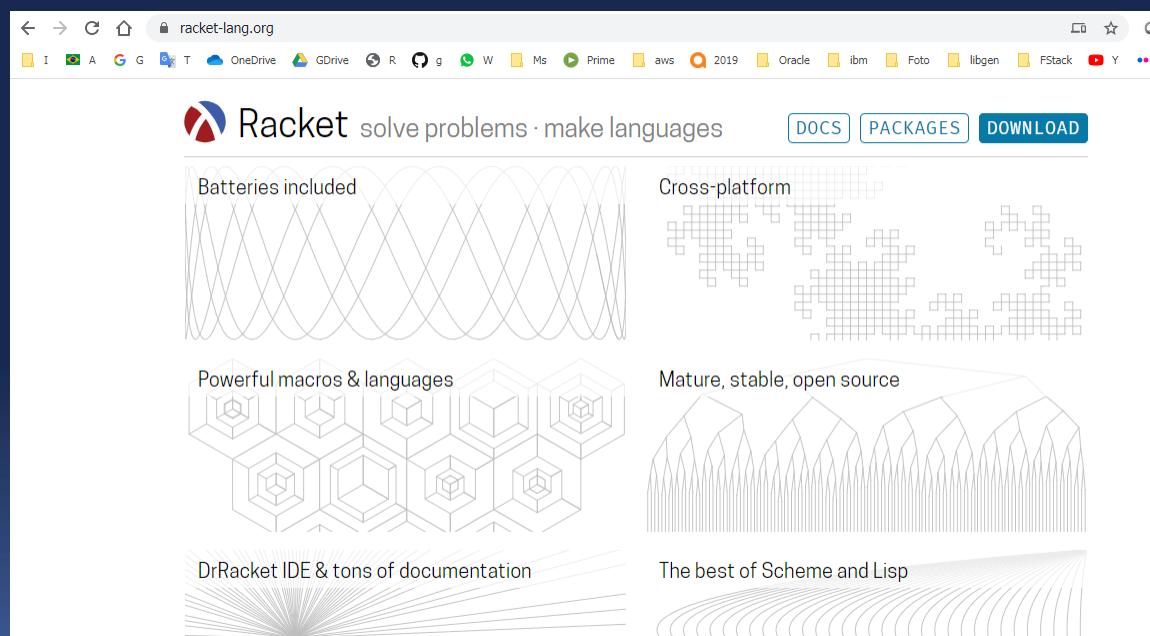
Package Management in Racket
[The Racket Drawing Toolkit](#)
[The Racket Graphical Interface Toolkit](#)
[The Racket Foreign Interface](#)
[Scribble: The Racket Documentation Tool](#)
[DrRacket: The Racket Programming Environment](#)
[raco: Racket Command-Line Tools](#)

Building, Distributing, and Contributing to Racket
[How to Program Racket: a Style Guide](#)

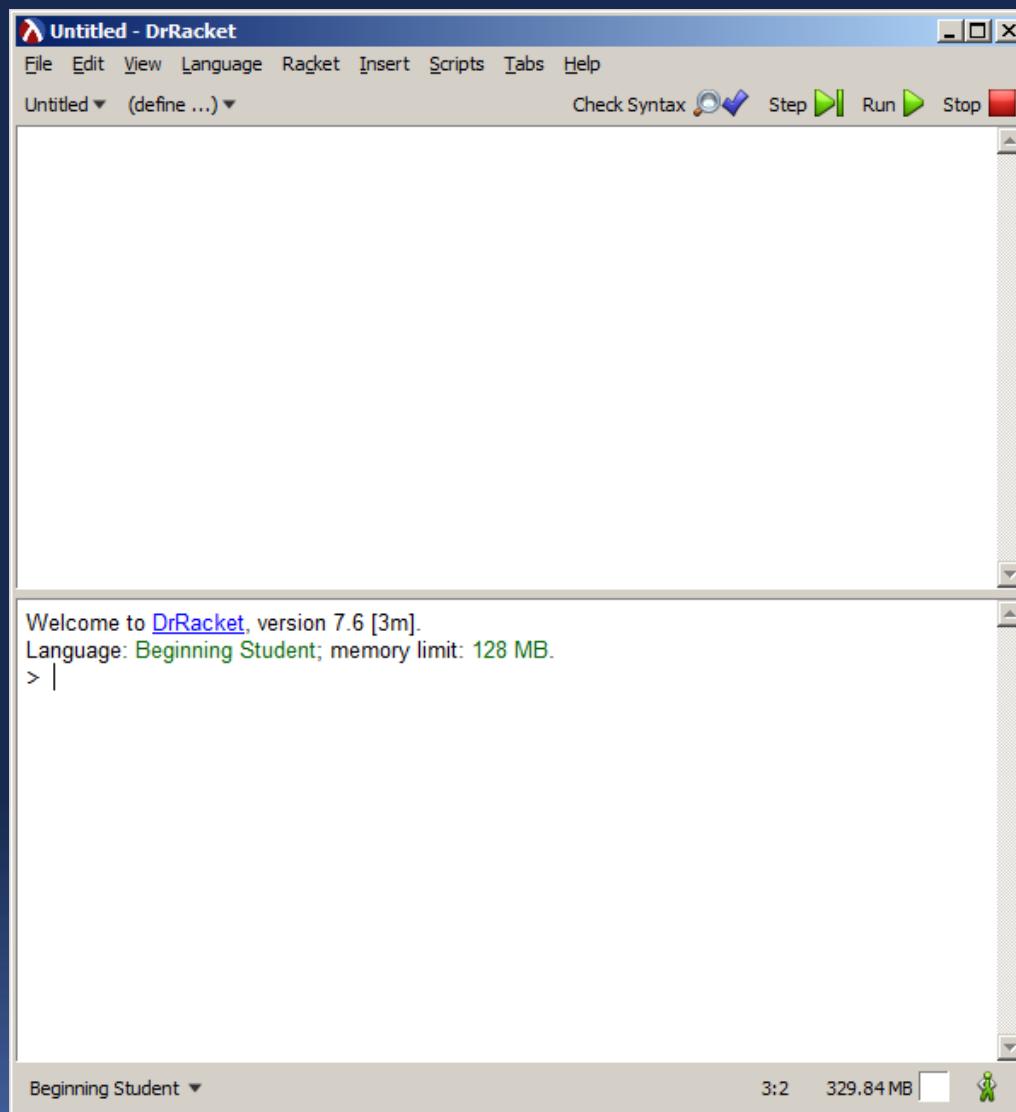
 Racket

Introdução

- ✓ Nesta unidade trataremos de conceitos da **Programação Funcional**, um paradigma onde o foco da computação é realizado por meio de **funções**;
- ✓ Para **exercitarmos** os conceitos da Programação Funcional, empregaremos um ambiente de programação, desenvolvido no **MIT**, chamado **DrRacket**;
- ✓ Assim, será necessário baixar o ambiente no endereço: <https://racket-lang.org/>

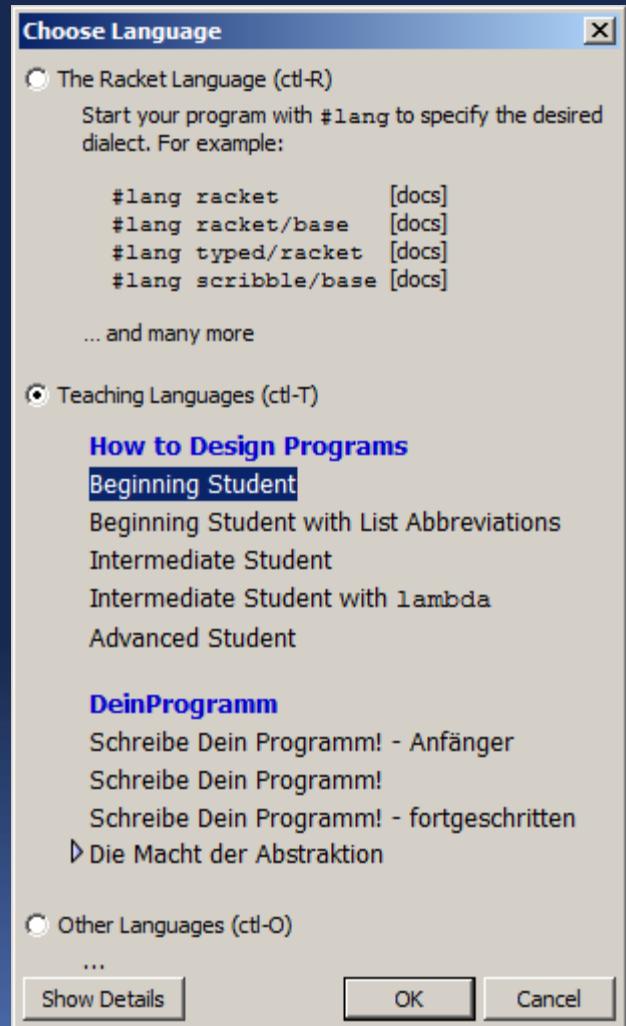
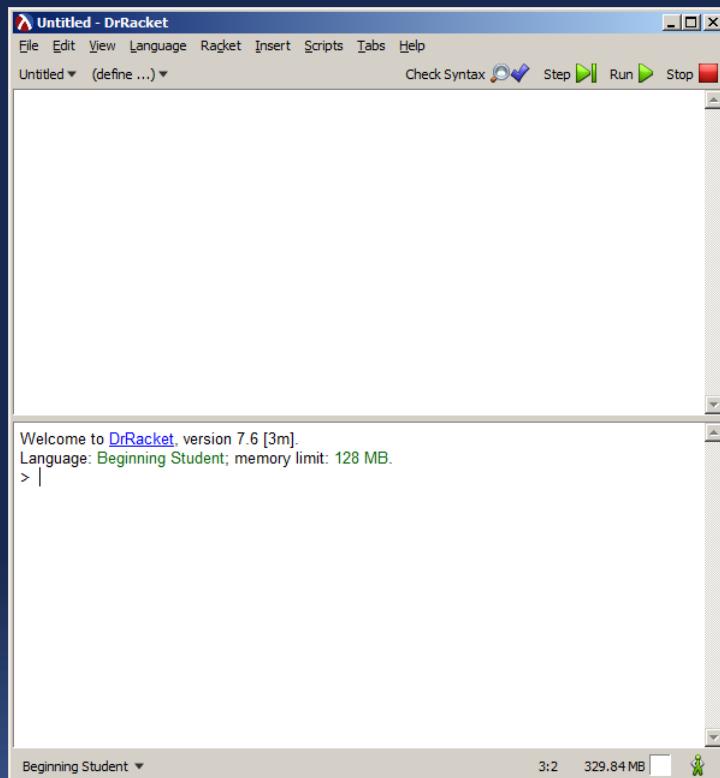


DrRacket



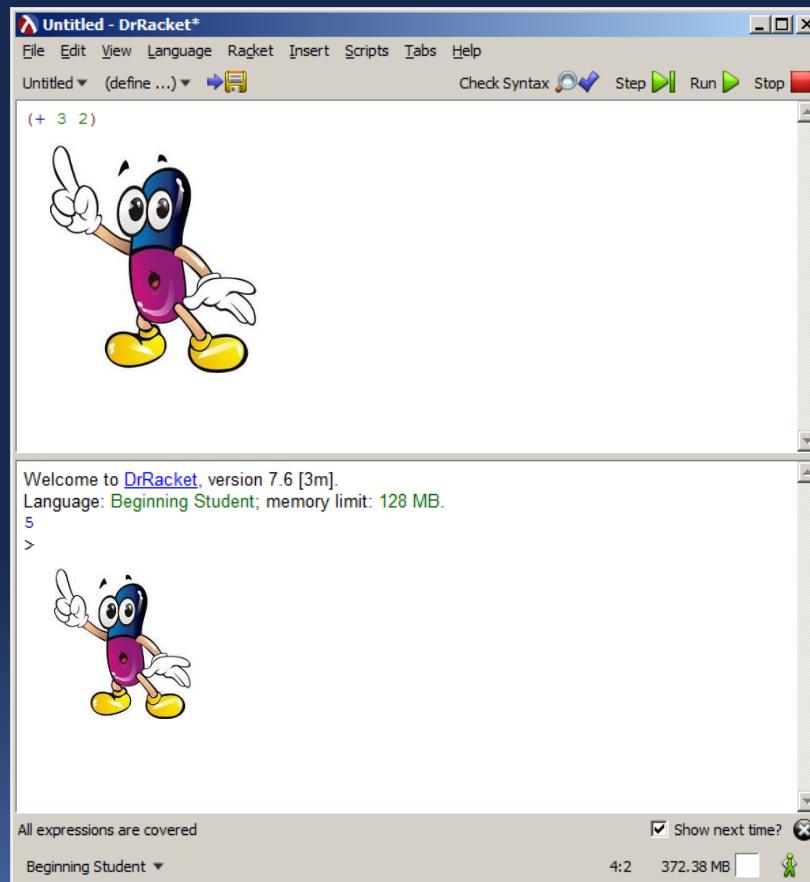
DrRacket

- ✓ Após iniciar o DrRacket, selecione “Choose Language” a partir do menu “Language”, o qual abre um diálogo.
- ✓ Definir em Teaching Languages a opção “Beginning Student”;
- ✓ Tecle OK após a definição da Teaching Languages.



DrRacket

- ✓ Após essa definição, podemos iniciar a programação com o DrRacket;
- ✓ Iniciaremos com um simples cálculo: $(+ 3 2)$
- ✓ Em seguida clique em **Run**;
- ✓ O resultado será exibido na parte inferior da janela principal do ambiente.



Efetuando diversos cálculos

Untitled - DrRacket*

File Edit View Language Racket Insert Scripts Tabs Help

Untitled ▾ (define ...) ▾ ➔

Check Syntax Step Run Stop

```
(+ 3 4)
(* 2 3)
(- 4 3)
(/ 10 5)
```



Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student; memory limit: 128 MB.

```
7
6
1
2
>
```



All expressions are covered Show next time?

Beginning Student ▾

7:2 525.19 MB

DrRacket – Área de Definição

- ✓ A parte superior é chamada de Área de Definição;
- ✓ Nesta área, pode-se criar programas;
- ✓ Esta área também é chamada de área de edição.



Untitled - DrRacket*

File Edit View Language Racket Insert Scripts Tabs Help

Untitled (define ...) Check Syntax Step Run Stop

```
(+ 3 4)
(* 2 3)
(- 4 3)
(/ 10 5)
```

Área de Definição

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student; memory limit: 128 MB.

```
7
6
5
4
3
2
>
```

All expressions are covered Show next time?

Beginning Student 7:2 525.19 MB

DrRacket – Área de Interação

✓ A parte inferior é chamada de área de **Interação**;

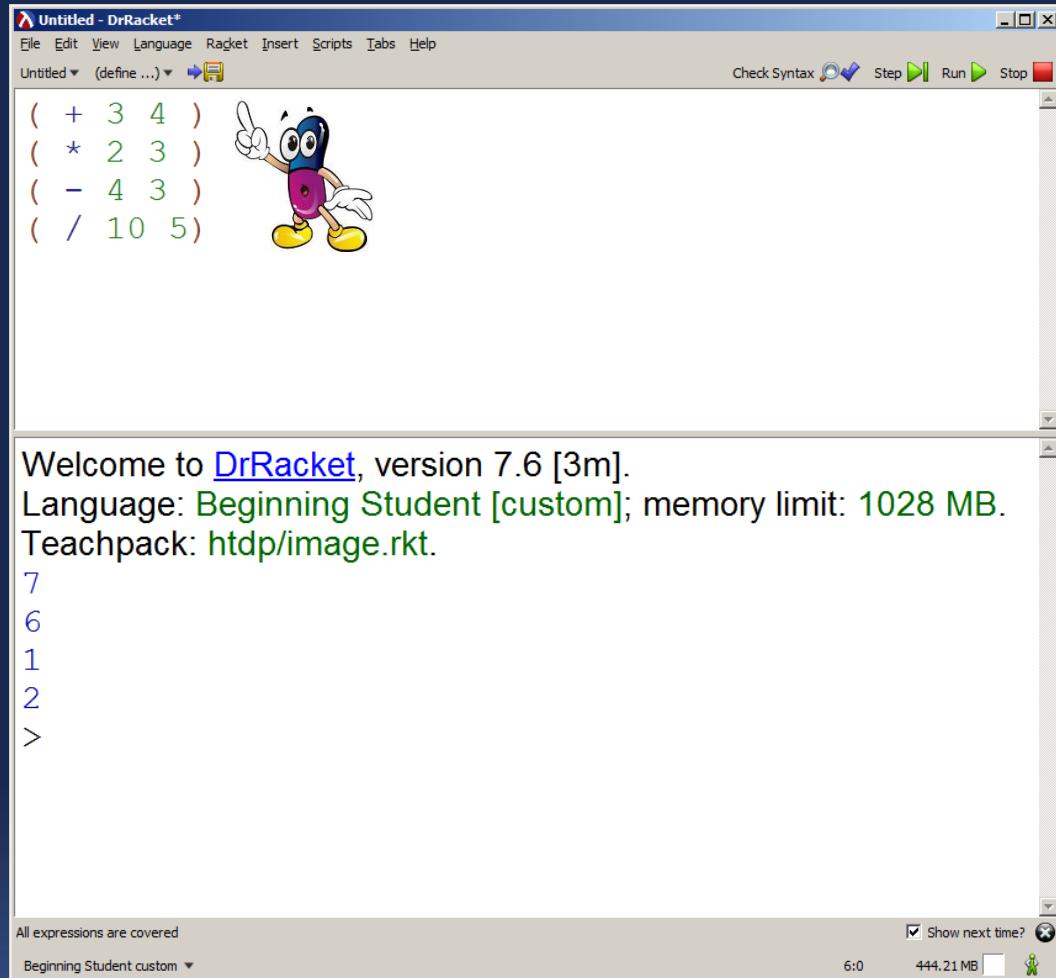
✓ Nesta área, pode-se entrar com **expressões** para serem avaliadas de forma interativa (**REPL**)



✓ **REPL** = Read-Eval-Print Loop

The screenshot shows the DrRacket interface. At the top is a menu bar with File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu is a toolbar with buttons for Untitled, (define ...), Check Syntax, Step, Run, and Stop. The main area contains some Racket code:
(+ 3 4)
(* 2 3)
(- 4 3)
(/ 10 5)
Below this is a welcome message:
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student; memory limit: 128 MB.
The REPL prompt shows:
7

DrRacket – Botão Save



Programas

- ✓ Consistem de expressões;
- ✓ Uma expressão consiste de termos que são iniciados por "(" e encerrados por ")";
- ✓ Ao se clicar no botão **RUN**, **DrRacket** avalia as expressões existentes na área de definição e mostra o resultado na **área de interações**. (janela inferior do ambiente)
- ✓ Na **área de interações**, o prompt ">" indica que o **DrRacket** está esperando novas expressões para serem avaliadas.

The screenshot shows the DrRacket environment. The top window is titled "DrRacket_01.rkt - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Check Syntax, Step, Run, and Stop. The code editor (tab 1) contains the following Racket code:

```
(+ 3 4)
( * 2 3 )
( - 4 3 )
(/ 10 5)
```

The bottom window is titled "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student; memory limit: 128 MB." It shows a history of evaluated expressions:

```
7
6
1
2
> ( * 4 3 )
12
> |
```

A cartoon character with a blue head, purple pants, and a white shirt is pointing upwards between the two windows. A status bar at the bottom indicates "All expressions are covered", "Beginning Student", "9:2", "565.98 MB", and a checked "Show next time?" checkbox.

Exemplo

- ✓ Entre com as expressões abaixo, na área de interações, em seguida, tecle <enter> e aguarde a resposta do DrRacket com os resultados;

```
> (+ 2 2)
4
> (* 3 3)
9
> (- 4 2)
2
> (/ 6 2)
3
> (sqr 3)
9
> (expt 2 3)
8
> (sin 0)
0
> (cos pi)
#i-1.0
```

Exemplo

Untitled 3 - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

Untitled 3 ▾ (define ...) ▾

Check Syntax Step Run Stop

1: DrRacket_01.rkt | 2: Untitled 2 | 3: Untitled 3

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student; memory limit: 128 MB.

```
> (+ 2 2 ) 4
> (* 3 3 ) 9
> (- 4 2 ) 2
> (/ 6 2 ) 3
> (sqr 3) 9
> (expt 2 3) 8
> (sin 0) 0
> (cos pi)
#i-1.0
> |
```

All expressions are covered Show next time?

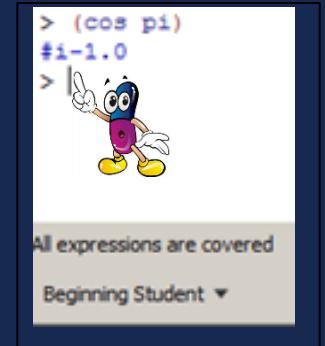
Beginning Student ▾ 19:2 321.29 MB



Observação

- ✓ Observe o resultado da avaliação da última expressão;

→ > (cos pi)
#i -1.0
>

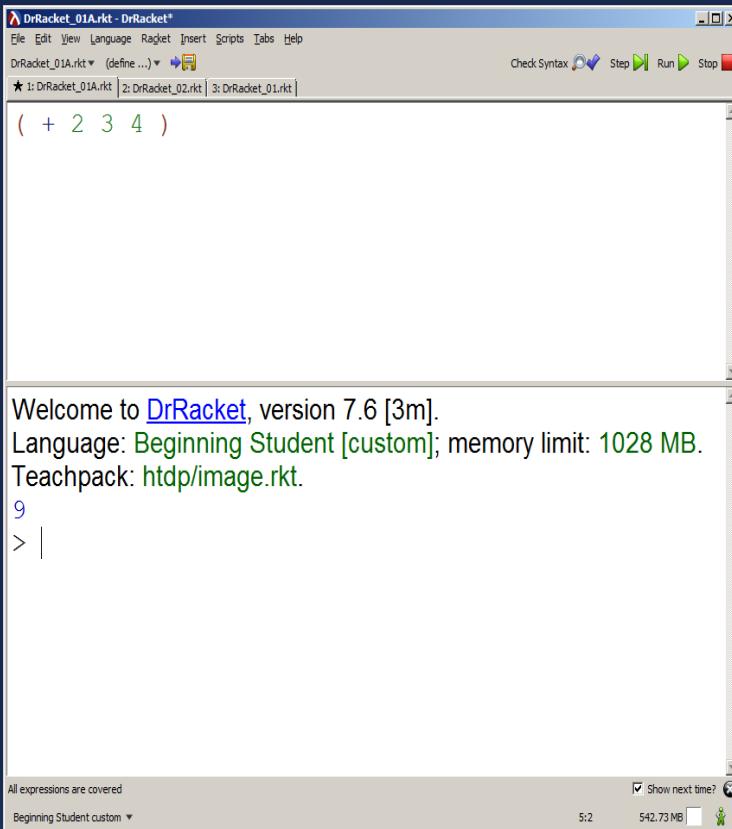


- ✓ No resultado da avaliação, consta o prefixo “**#i**” que significa “**inexact number**”;
- ✓ Nesse caso o prefixo “**#i**” é acrescentado ao resultado para indicar (**warning**) que o valor **não** é exato.



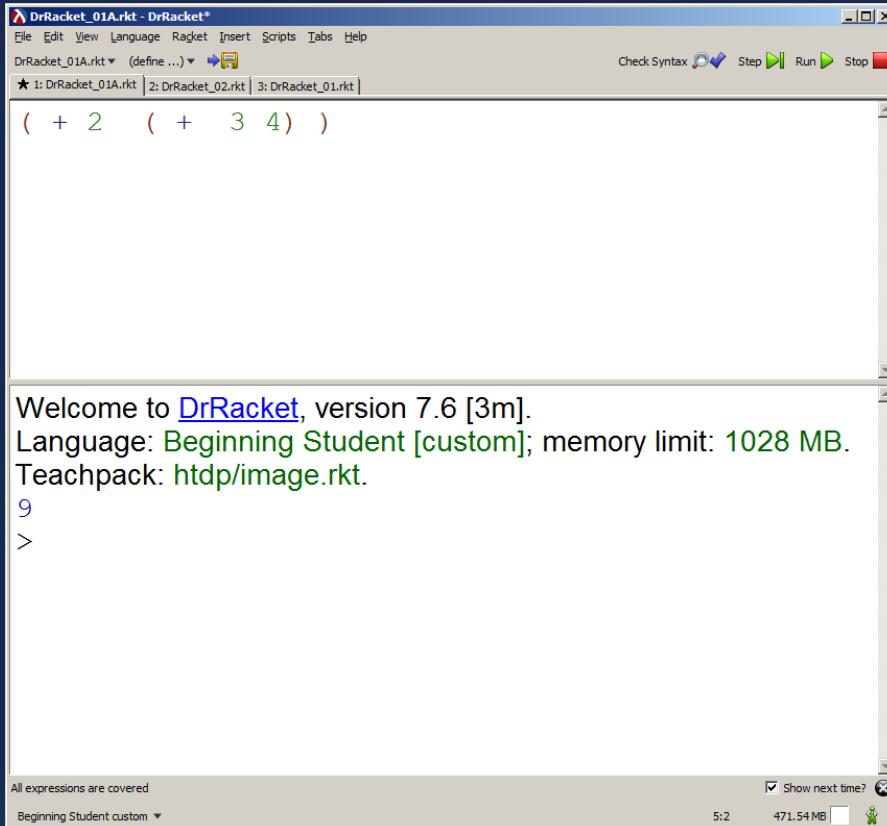
DrRacket é honesto, quando o valor não é exato ele sinaliza um **warning**!!!!

Encadeando operações (BSL Arithmetic)



- ✓ Em **BSL**, deve-se **abrir** parênteses, em seguida definir a operação a ser executada, os valores a serem operados (**operandos**) e por fim um **fecha** parênteses.
- ✓ **BSL** = Begin Student Language.

Encadeando operações (Nested Arithmetic)



- ✓ Em **Nested Arithmetic**, deve-se **abrir** parênteses, em seguida definir a operação a ser executada, os **operandos** a serem operados e por fim um **fecha** parênteses.
- ✓ Mas, nessa aritmética, um operando pode ser substituído por uma expressão, como é o caso do exemplo acima.

Encadeando funções – Exemplo

The screenshot shows the DrRacket IDE interface. The top window title is "DrRacket_01A.rkt - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu is a toolbar with "Check Syntax" (blue checkmark), "Step" (green arrow), "Run" (green arrow), and "Stop" (red square). The tabs at the bottom of the toolbar show 1: DrRacket_01A.rkt (selected), 2: DrRacket_02.rkt, and 3: DrRacket_01.rkt. The code editor contains the following Racket expression:

```
(+ (* 5 5) (+ (* 3 (/ 12 4)) ) 4) )
```

The DrRacket welcome message is displayed in the main window:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.
```

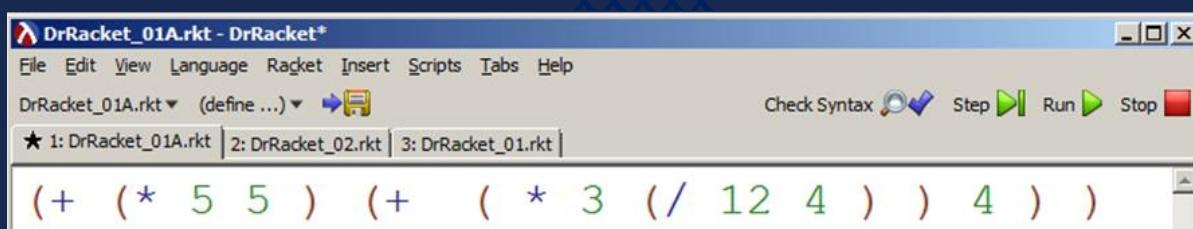
The REPL prompt shows the number 38 followed by a greater than sign (>).

At the bottom of the window, status messages include "All expressions are covered", "Beginning Student custom", "Show next time?", "5:2", "479.45 MB", and a small green figure icon.

XXXXX

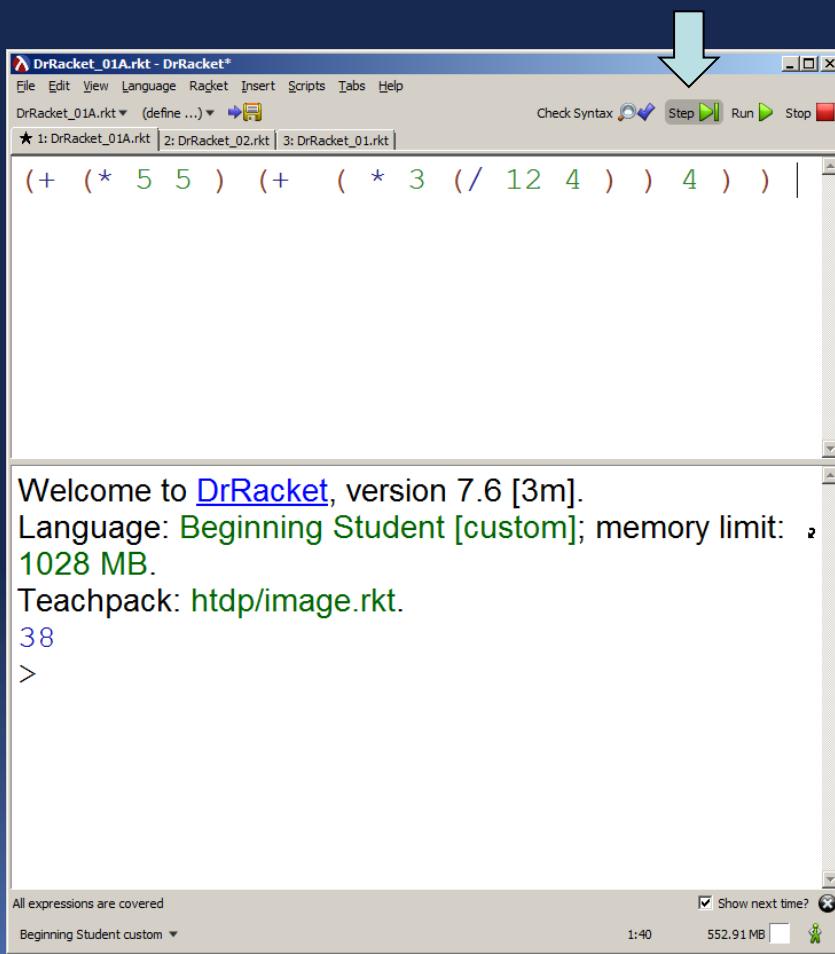
Observações

- ✓ A **avaliação** de uma expressão é feita por meio de **regras**;
- ✓ As **regras** no caso de operações aritméticas são as usuais da **Matemática**;
- ✓ Assim, o resultado de uma operação somente será obtido **após** a avaliação de seus **operandos**;
- ✓ Sempre que um **abre** parênteses ocorre, será determinado o resultado da primeira expressão encadeada (**nested**).



A screenshot of the DrRacket IDE. The title bar says "DrRacket_01A.rkt - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has "Check Syntax" and buttons for Step (green arrow), Run (yellow arrow), and Stop (red square). The tabs show 1: DrRacket_01A.rkt, 2: DrRacket_02.rkt, and 3: DrRacket_01.rkt. The code editor contains a complex Racket expression: $(+ (* 5 5) (+ (* 3 (/ 12 4))) 4))$.

✓ Clique no botão **Step** para rastrear a execução da expressão acima;



A screenshot of the DrRacket IDE showing the result of clicking the Step button. A large blue arrow points from the first screenshot to this one. The interface is identical to the first, but the code editor now shows the expression partially evaluated: $(+ (* 5 5) (+ (* 3 (/ 12 4))) | 4))$. The bottom pane displays the Racket welcome message and a command-line interface:

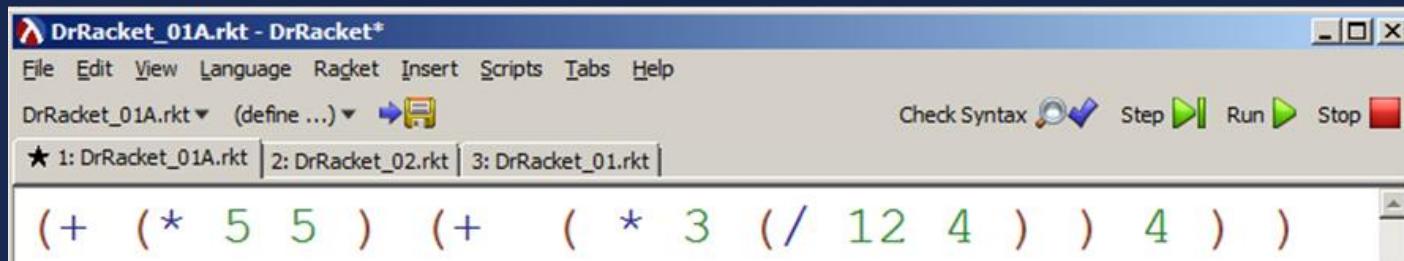
```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/image.rkt.  
38  
>
```

The status bar at the bottom indicates "All expressions are covered" and "Beginning Student custom".

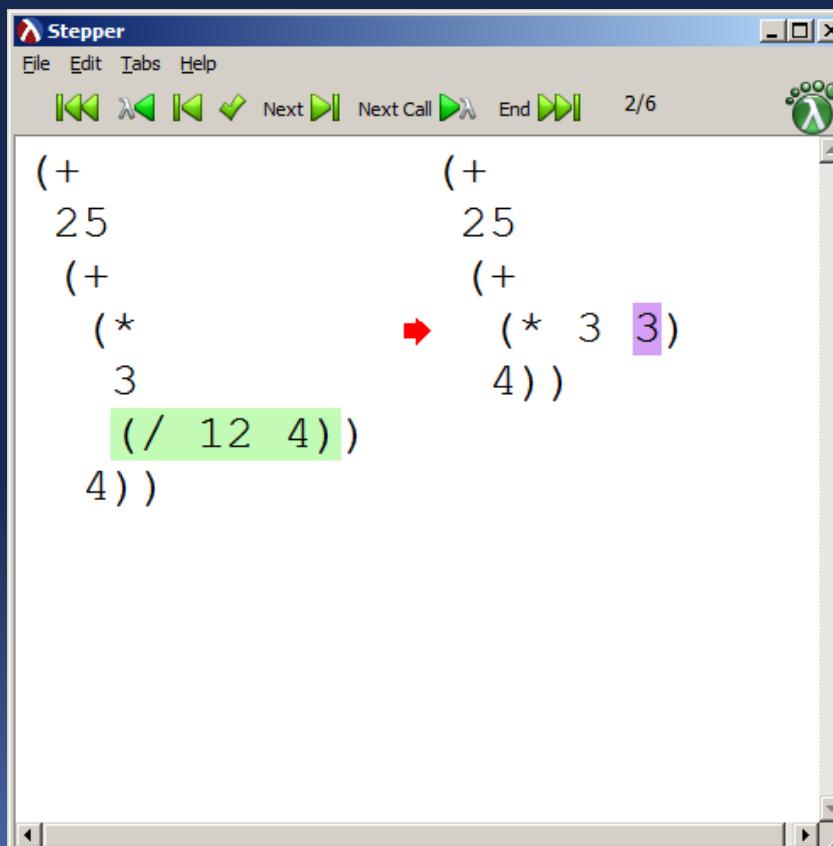
The screenshot shows the DrRacket interface with the title bar "DrRacket_01A.rkt - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. A toolbar with buttons for Check Syntax, Step, Run, and Stop is visible. The tabs at the bottom show "1: DrRacket_01A.rkt", "2: DrRacket_02.rkt", and "3: DrRacket_01.rkt". The main window displays a complex Racket expression:
$$(+ (* 5 5) (+ (* 3 (/ 12 4))) 4))$$

- ✓ A expressão $(* 5 5)$ é primeiramente avaliada;
- ✓ Clique agora no botão **Next** para continuar o rastreamento.

The screenshot shows the Stepper tool with the title bar "Stepper". The menu bar includes File, Edit, Tabs, and Help. The toolbar features navigation buttons: Back, Previous, Next, Next Call, End, and a magnifying glass icon. It also shows "1/6" and a lambda logo. The main window displays the same Racket expression as the DrRacket screenshot, but it is annotated with evaluation steps:
1. The term $(* 5 5)$ is highlighted in green.
2. The result of the multiplication, 25 , is highlighted in purple.
3. The term $(+ (* 3 (/ 12 4)))$ is shown below the first step.
4. A red arrow points from the result 25 to the term $(*$.
5. The term $(/ 12 4))$ is shown below the second step.
6. The final result $(/ 12 4))$ is shown below the third step.



- ✓ A expressão $(/ 12 4)$ é avaliada na sequência;
- ✓ Clique novamente no botão Next;



The screenshot shows the DrRacket interface with the title "DrRacket_01A.rkt - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar features "Check Syntax" with a blue checkmark icon, "Step" with a green right-pointing arrow icon, "Run" with a green right-pointing arrow icon, and "Stop" with a red square icon. A tab bar at the bottom shows three tabs: 1: DrRacket_01A.rkt (marked with a star), 2: DrRacket_02.rkt, and 3: DrRacket_01.rkt. The main code editor displays the following Racket expression:

```
(+ (* 5 5) (+ (* 3 (/ 12 4)) ) 4) )
```

- ✓ A expressão $(+ 3 3)$ é avaliada na sequência;
- ✓ Clique novamente no botão **Next**;

The screenshot shows the Stepper tool window titled "Stepper". The menu bar includes File, Edit, Tabs, and Help. The toolbar includes navigation icons: back, forward, next, next call, end, and a progress indicator showing 3/6. The main area displays the step-by-step evaluation of the Racket expression. The expression is shown in two columns. The left column contains the first part of the expression: $(+$, 25 , $(+$, $(* 3 3)$, $4)$, $)$. The right column contains the second part: $(+$, 25 , \Rightarrow , $(+ 9 4)$, $)$. The term $(* 3 3)$ is highlighted in green, and the term 9 is highlighted in purple, indicating the current step of the evaluation.

The screenshot shows the DrRacket interface with the title "DrRacket_01A.rkt - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The tabs at the top show "1: DrRacket_01A.rkt", "2: DrRacket_02.rkt", and "3: DrRacket_01.rkt". Below the tabs is a code editor containing the following Racket expression:

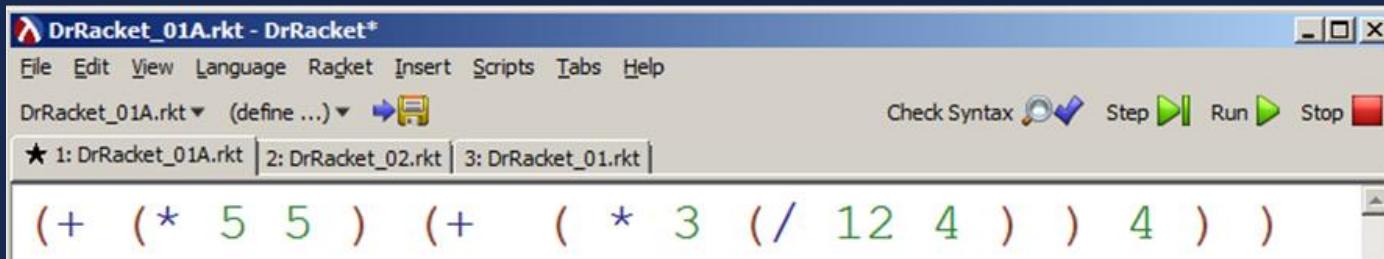
```
(+ (* 5 5) (+ (* 3 (/ 12 4)) ) 4) )
```

- ✓ A expressão $(+ 9 4)$ é avaliada na sequência;
- ✓ Clique novamente no botão **Next**;

The screenshot shows the Stepper tool window with the title "Stepper". The menu bar includes File, Edit, Tabs, and Help. The toolbar has buttons for Back, Lambda Back, Left, Right, Next, Next Call, End, and Step. The status bar shows "4/6". The main area displays the Racket expression and its state during evaluation:

```
(+ 25 ( + 9 4 ))
```

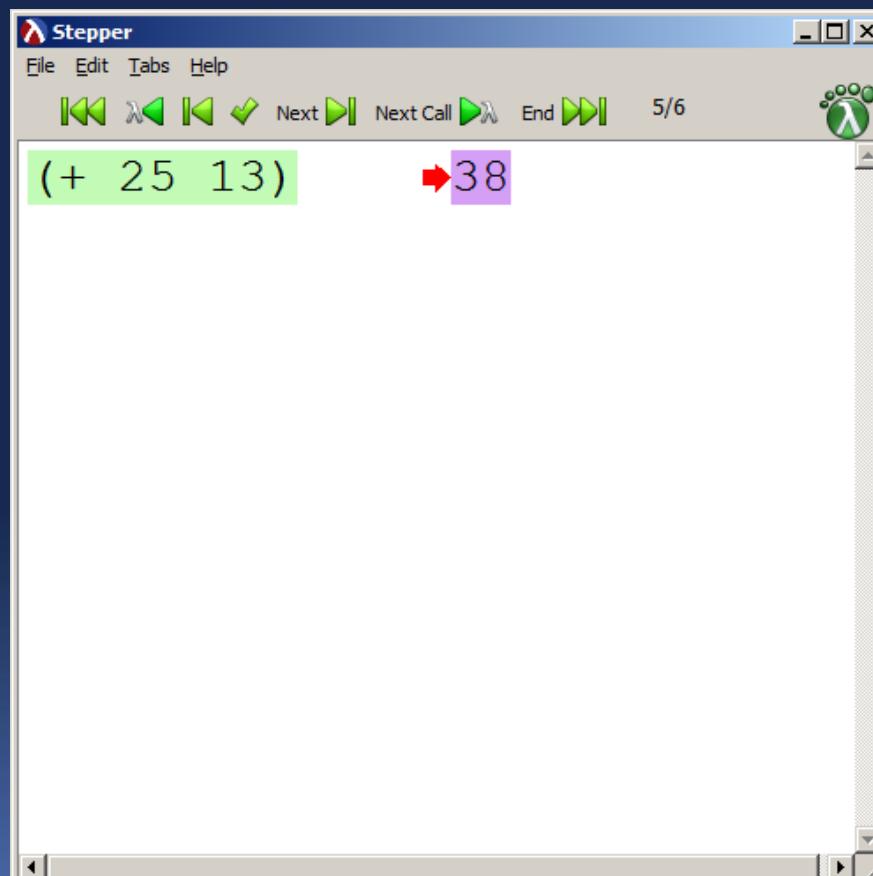
The term $(+ 9 4)$ is highlighted in green, indicating it is the current target for evaluation. A red arrow points to the right, indicating the direction of the next step.

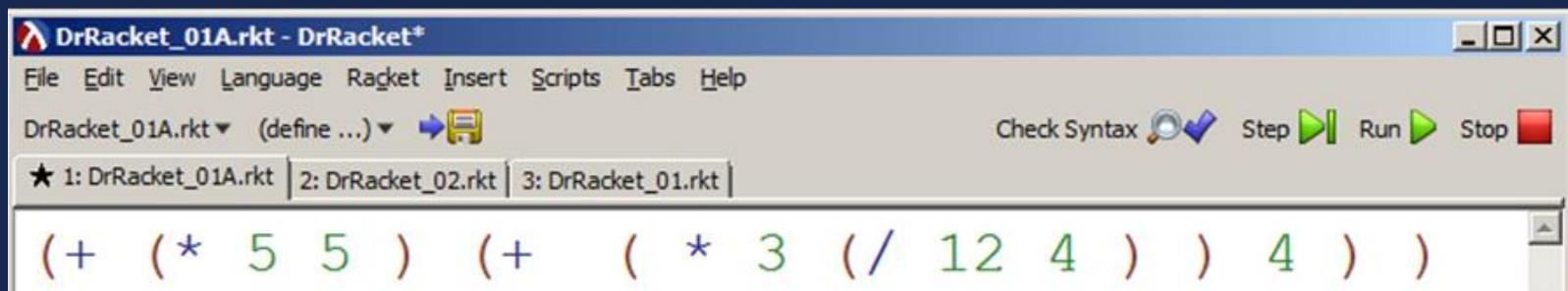


The screenshot shows the DrRacket interface with a complex Racket expression in the editor:

```
(+ (* 5 5) (+ (* 3 (/ 12 4)) ) 4) )
```

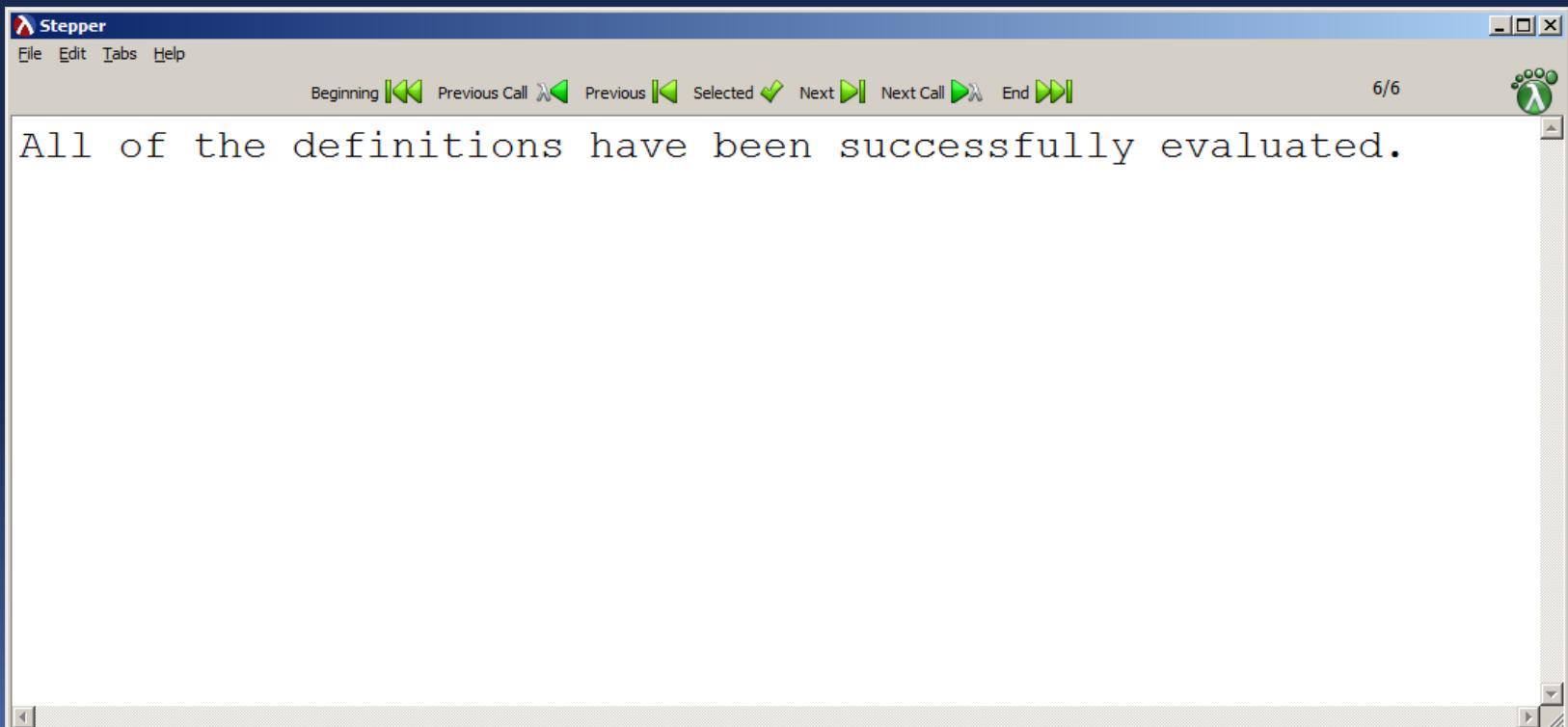
- ✓ A expressão $(+ 25 13)$ é avaliada na sequência;
- ✓ Clique novamente no botão **Next**;





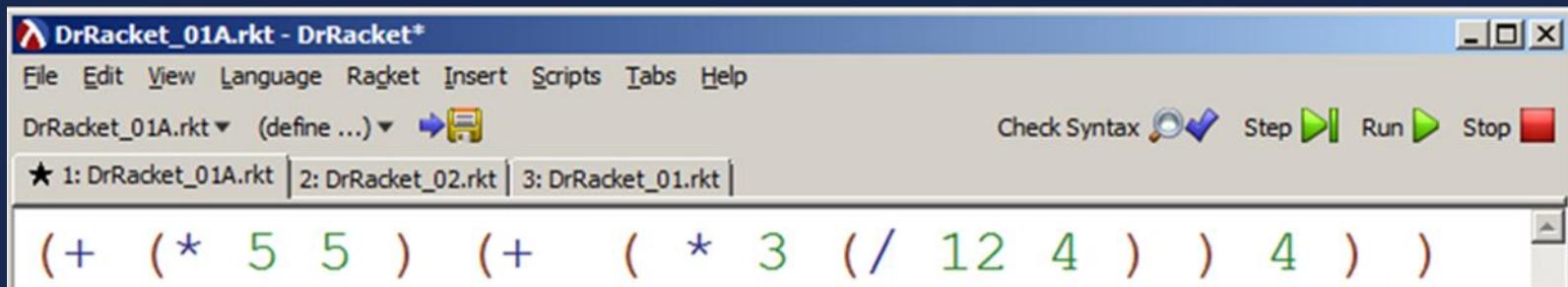
The screenshot shows the DrRacket interface with a menu bar (File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help) and a toolbar with buttons for Check Syntax, Step, Run, and Stop. There are three tabs open: 1: DrRacket_01A.rkt, 2: DrRacket_02.rkt, and 3: DrRacket_01.rkt. The main window displays a complex Racket expression:
$$(+ (* 5 5) (+ (* 3 (/ 12 4)) 4))$$

✓ O DrRacket indica que o **rastreamento** chegou ao fim e com sucesso !



The screenshot shows the Stepper tool interface with a menu bar (File, Edit, Tabs, Help) and a toolbar with buttons for Beginning, Previous Call, Previous, Selected, Next, Next Call, and End. A status bar indicates 6/6. The main window displays the message: "All of the definitions have been successfully evaluated."

Observação Importante



- ✓ Observe que na avaliação da expressão acima, o DrRacket nunca necessitou ponderar qual regra deveria ser aplicada!
- ✓ Por uma simples razão ! **SÓ EXISTE UMA REGRA A SER APLICADA!**



Parênteses têm significado

- ✓ Deve-se entrar a quantidade **correta** de parênteses no aninhamento;
- ✓ Nem **mais**, nem **menos** !
- ✓ Por exemplo, matematicamente a expressão $(+ (1) (2))$ pode estar correta, **mas** na **semântica** das linguagens funcionais, após um abre parênteses deve haver uma função, o que causaria **erro** em tempo de interpretação.

The screenshot shows the DrRacket IDE interface. In the top-left window, there is a code editor with the following text:

```
( + (|1| (2) )
```

A large blue arrow points upwards from the bottom of the slide towards the opening parenthesis in the code. In the bottom-right window, the Racket REPL shows the following output:Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).
function call: expected a function
after the open parenthesis, but found a number
>

The error message is displayed in red text. At the bottom of the DrRacket window, there is a status bar with the text "All expressions are covered" and "Show next time?".

Manuseando Strings

- ✓ Em BSL, texto é uma sequência de caracteres delimitados por " e são denominados **strings**:

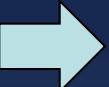


- ✓ Assim, "Hello DrRacket" é um perfeito string;

Manuseando Strings

xxxxx

- ✓ Quando **DrRacket** avalia um string, ele ecoa o próprio string na área de interação, como um número;



Untitled - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

Untitled (define ...) ▾ Check Syntax Step Run Stop

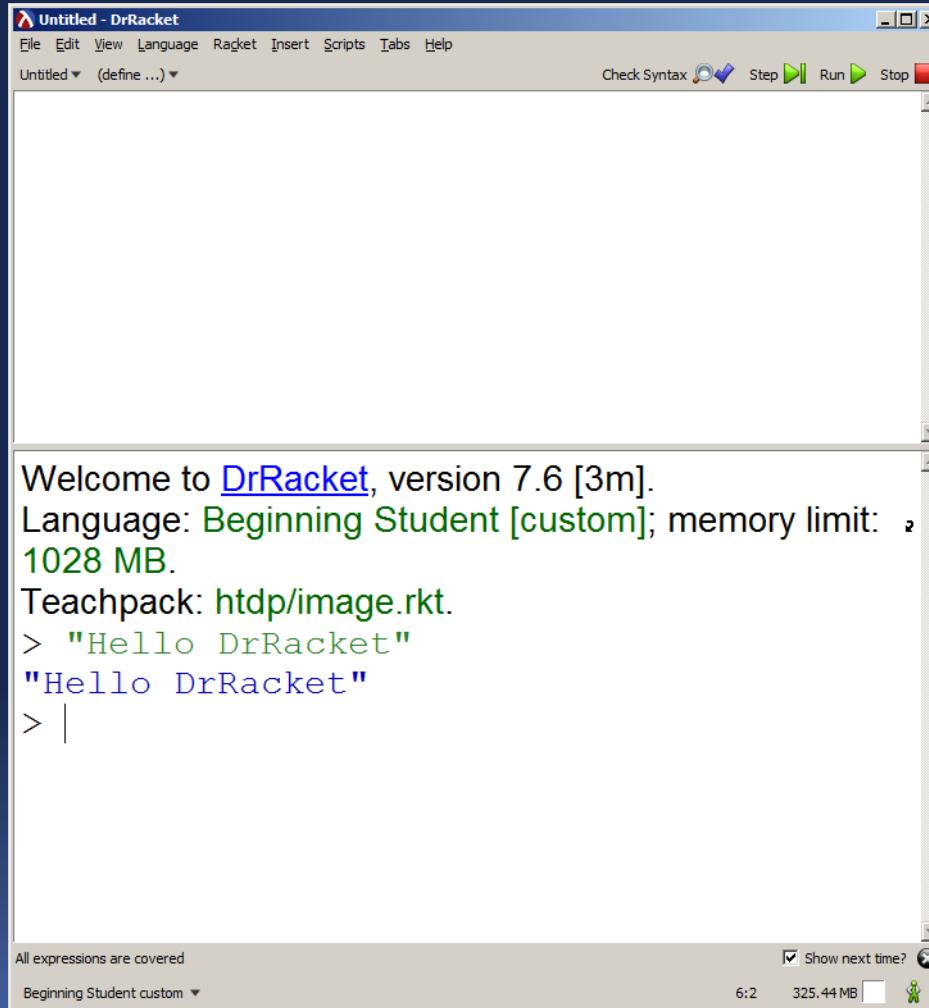
Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

```
> "Hello DrRacket"
"Hello DrRacket"
> |
```

All expressions are covered

Show next time?

Beginning Student custom 6:2 325.44 MB



Concatenando strings

xxxxx

- ✓ A função **string-append** retorna um string que corresponde à concatenação de strings passados como argumentos;
- ✓ O segundo string é adicionado ao final do primeiro ;



Untitled 2 - DrRacket

File Edit View Language Racket Insert Scripts Tabs Help

Untitled 2 ▾ (define ...) ▾

Check Syntax Step Run Stop

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

```
> (string-append "Hello" " World...")  
"Hello World..."  
> |
```

All expressions are covered

Beginning Student custom ▾

Show next time?

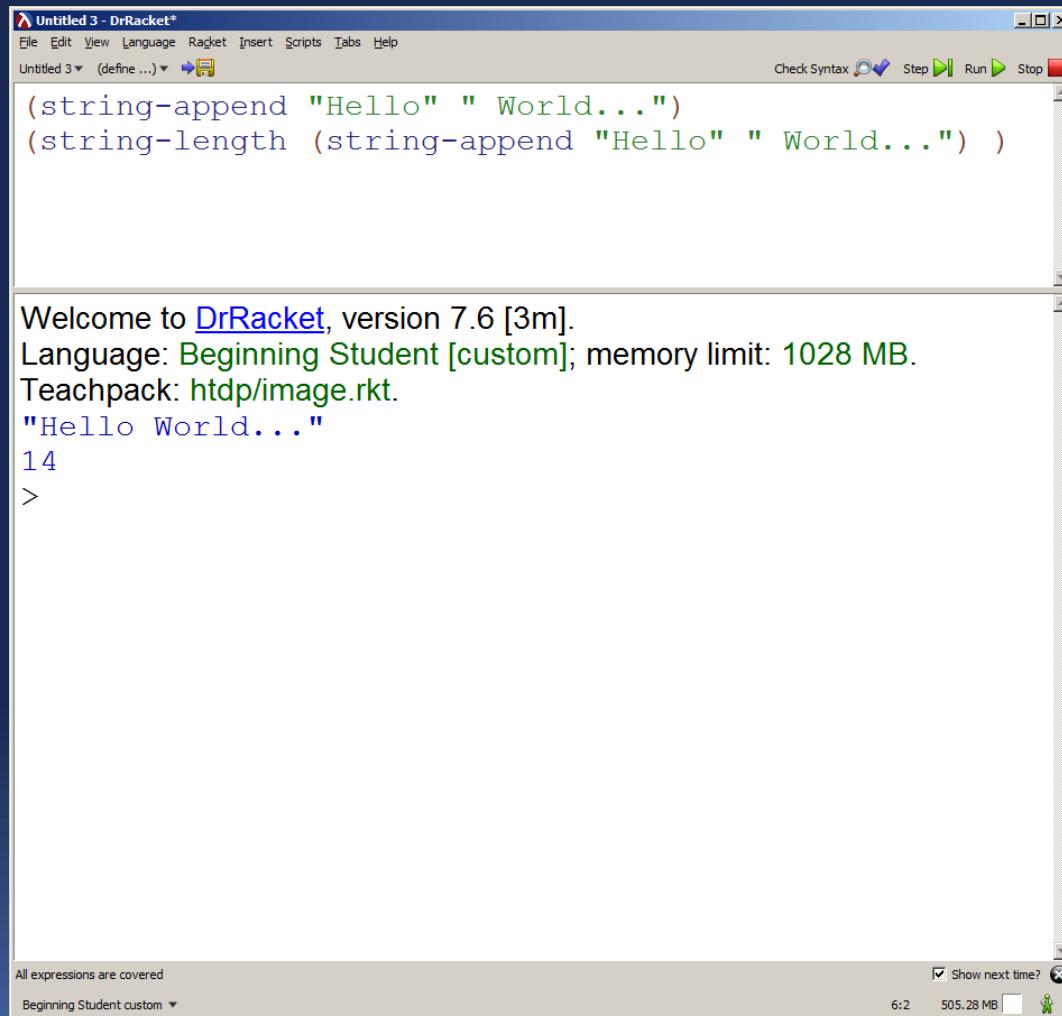
6:2 370.89 MB

LAMBDA CALCULUS

Retornando tamanho do string

xxxxx

- ✓ A função **string-length** retorna um valor numérico que corresponde ao tamanho do string passado como argumento ;



The screenshot shows the DrRacket interface. In the top editor window, the code is:

```
(string-append "Hello" " World...")  
(string-length (string-append "Hello" " World..."))
```

In the bottom interaction window, the output is:

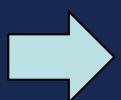
```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
"Hello World..."  
14  
>
```

A large blue arrow points from the left towards the interaction window.

Convertendo strings para números

xxxxx

- ✓ A função **string->number** retorna o **valor numérico** que corresponde ao **string** passado como argumento.



```
Untitled 3 - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled 3 (define ...) Check Syntax Step Run Stop
(+ (string->number "20") (string->number "30"))

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: httpd/image.rkt.
50
>

All expressions are covered
Beginning Student custom ▾
Show next time? 5:2 467.44 MB
```

xxxxx

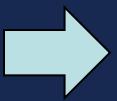
Convertendo strings para números

- ✓ E se na função **string->number** for passado um argumento que não corresponde a um número ?



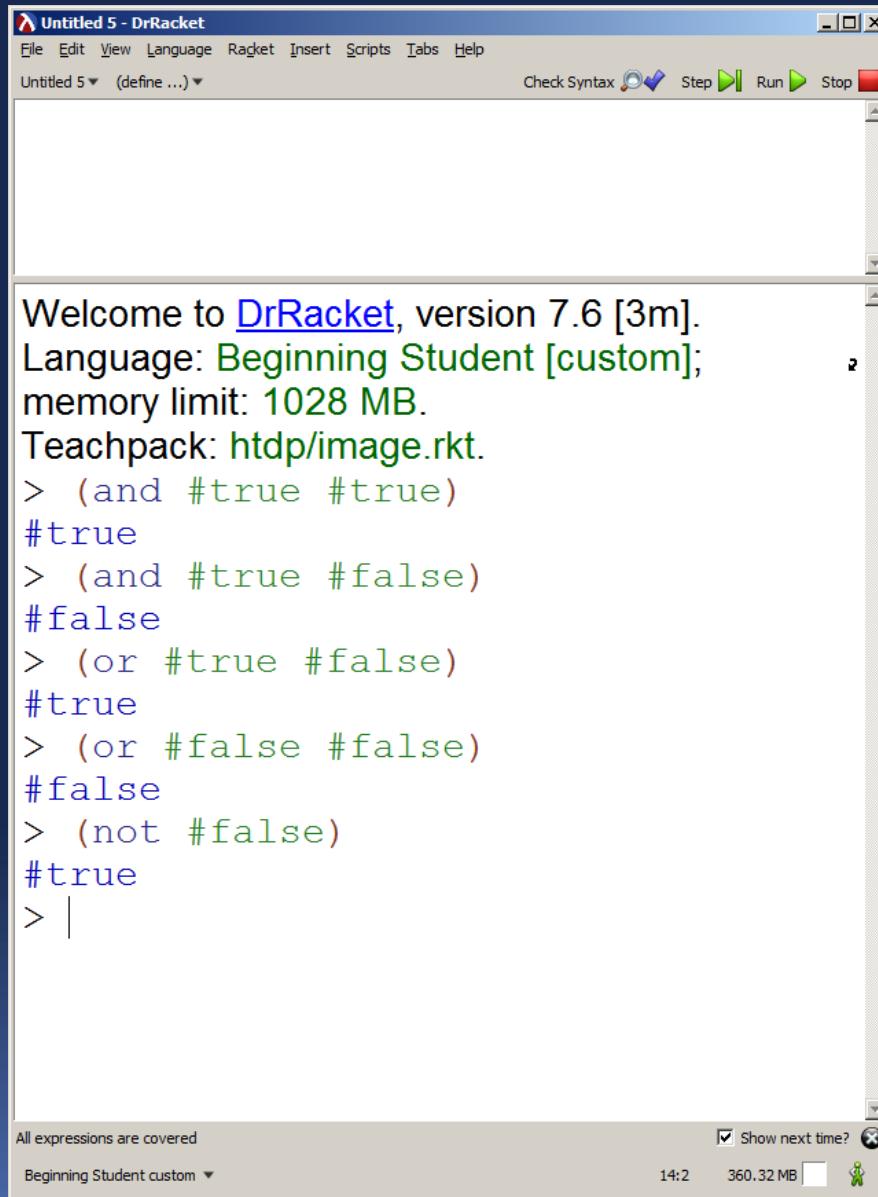
Convertendo strings para números

- ✓ Se o argumento passado para `string->number` não for um número apropriado, a função retornará `#false`, um tipo Boolean;
- ✓ Valores Boolean podem assumir dois valores: `#true` ou `#false`.



A screenshot of the DrRacket IDE. The title bar says "Untitled 3 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The top-level expression input field contains "(string->number \"AA\")". The interaction area shows the welcome message: "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB. Teachpack: htdp/image.rkt." followed by the result "#false". The status bar at the bottom indicates "All expressions are covered", "Show next time?", "Beginning Student custom", "5:2", "602.38 MB", and a battery icon.

Trabalhando com valores Boolean

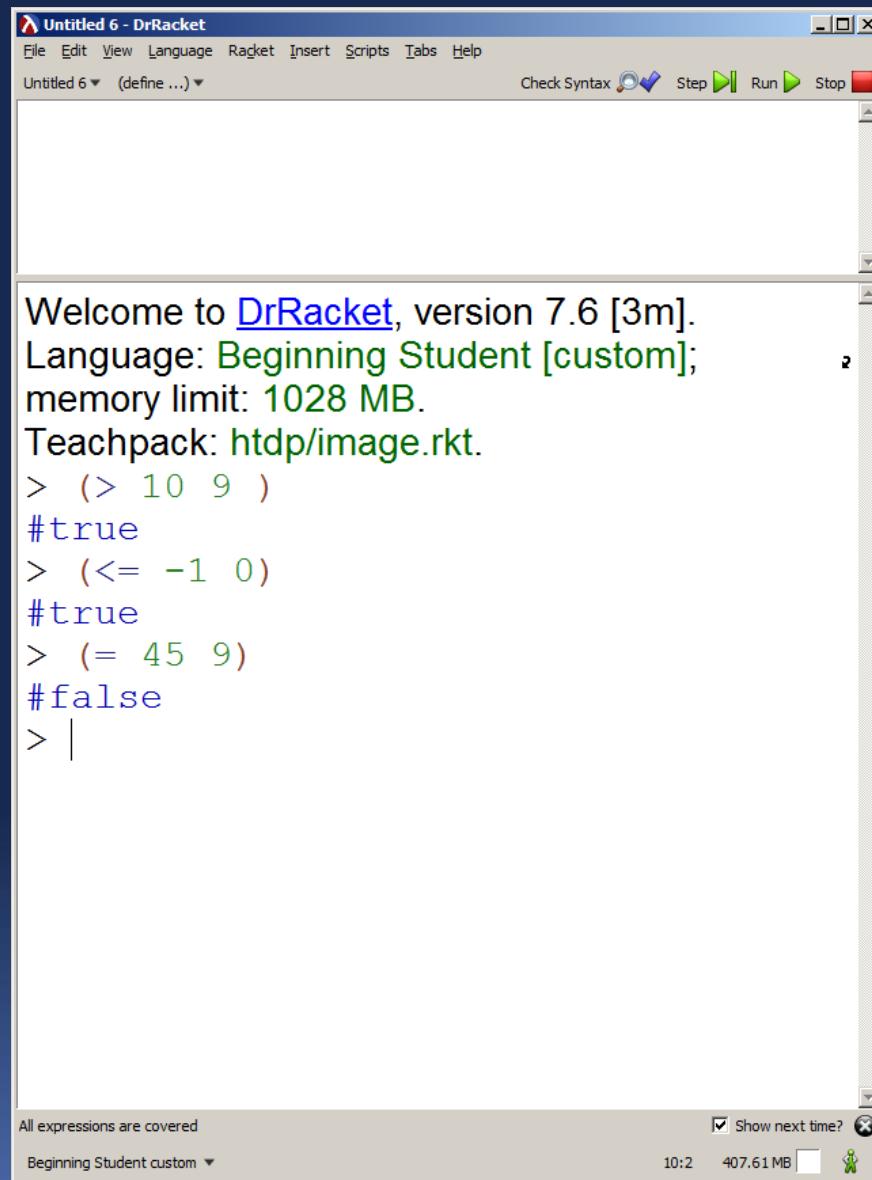


The screenshot shows the DrRacket IDE interface. The title bar reads "Untitled 5 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The code editor window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom];  
memory limit: 1028 MB.  
Teachpack: httpd/image.rkt.  
> (and #true #true)  
#true  
> (and #true #false)  
#false  
> (or #true #false)  
#true  
> (or #false #false)  
#false  
> (not #false)  
#true  
> |
```

The status bar at the bottom shows "All expressions are covered", "Show next time?", "Beginning Student custom", "14:2", "360.32 MB", and a small green person icon.

Comparando valores numéricos

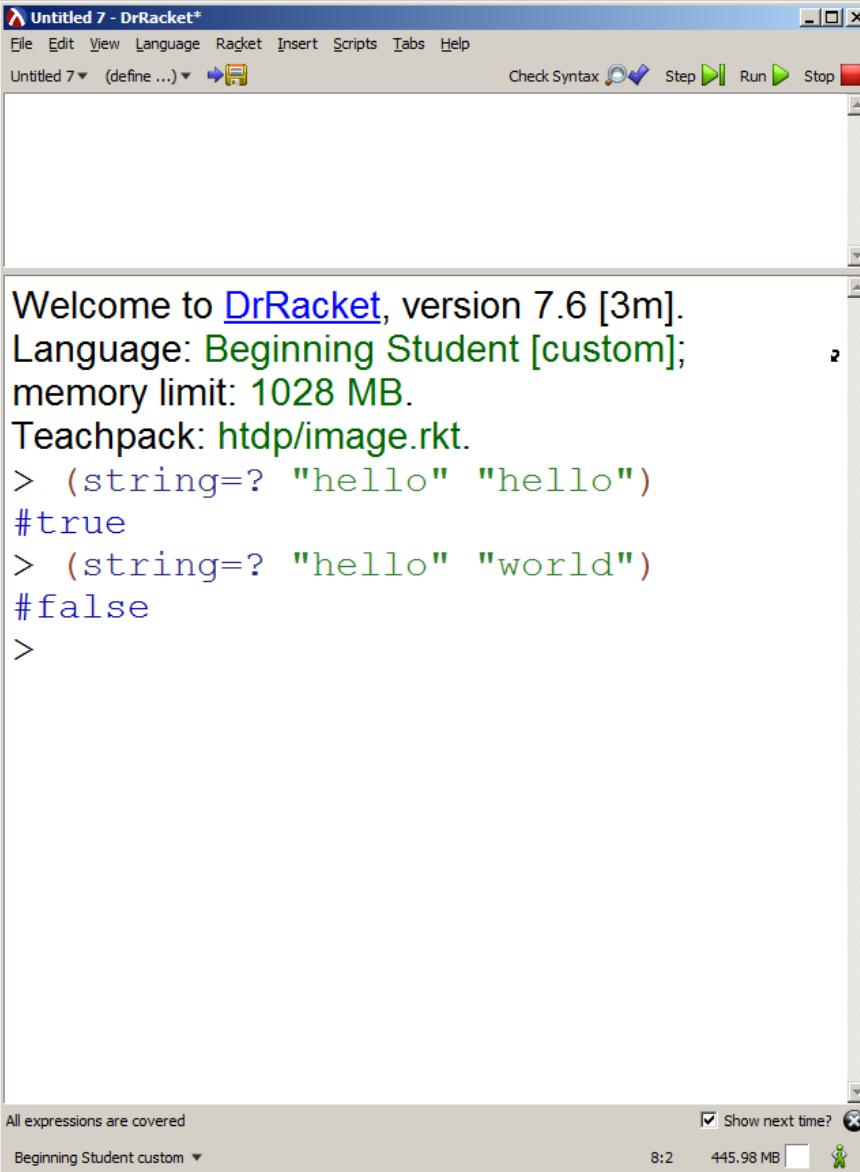


The screenshot shows the DrRacket IDE interface. The title bar reads "Untitled 6 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu is a toolbar with "Check Syntax" and buttons for Step, Run, and Stop. The main window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom];  
memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (> 10 9)  
#true  
> (≤ -1 0)  
#true  
> (= 45 9)  
#false  
> |
```

At the bottom of the window, status messages include "All expressions are covered" and "Show next time?". The footer shows "Beginning Student custom" and system information: 10:2, 407.61 MB, and a battery icon.

Comparando strings



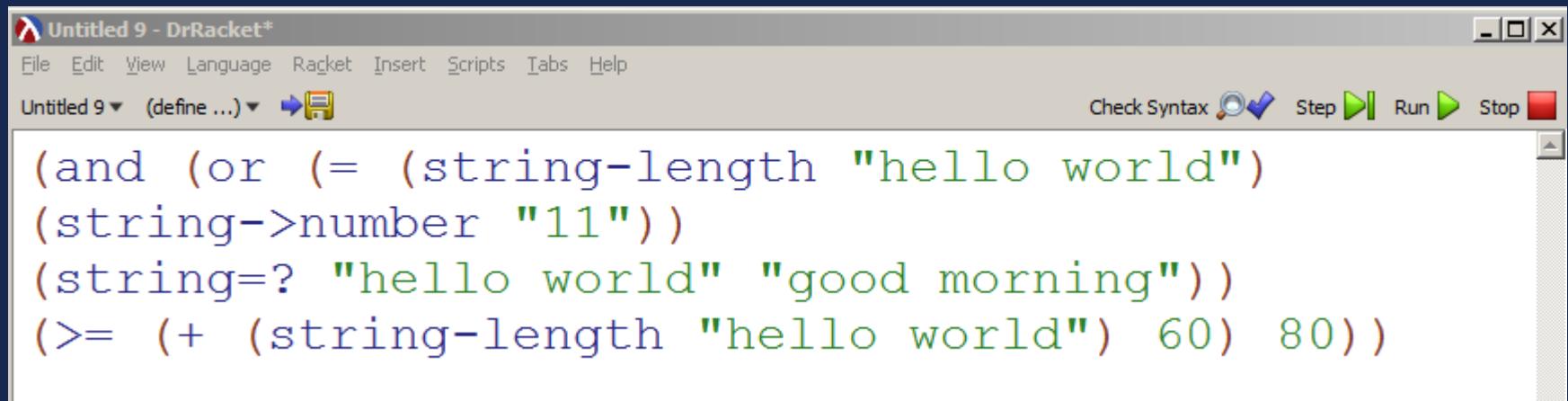
The screenshot shows the DrRacket IDE interface. The title bar reads "Untitled 7 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Untitled 7, (define ...), and a toolbar with Check Syntax, Step, Run, and Stop buttons.

The main window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom];  
memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (string=? "hello" "hello")  
#true  
> (string=? "hello" "world")  
#false  
>
```

At the bottom of the window, status messages include "All expressions are covered", "Beginning Student custom", "8:2", "445.98 MB", and a checkbox for "Show next time?".

✓ Qual o resultado da avaliação da expressão abaixo:



The screenshot shows the DrRacket interface with the title "Untitled 9 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar features "Untitled 9", "(define ...)", "Check Syntax", "Step", "Run", and "Stop". The code area contains the following Racket expression:

```
(and (or (= (string-length "hello world")
(string->number "11")))
(string=? "hello world" "good morning"))
(>= (+ (string-length "hello world") 60) 80))
```



✓ Qual o resultado da avaliação da expressão abaixo:

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 9 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Check Syntax, Step, Run, and Stop. The code editor contains the following Racket expression:

```
(and (or (= (string-length "hello world")  
          (string->number "11"))  
        (string=? "hello world" "good morning"))  
     (>= (+ (string-length "hello world") 60) 80))
```

The bottom window displays the evaluation results:

```
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
#false  
> |
```

The status bar at the bottom shows "Beginning Student custom" and "5:2 364.68 MB".

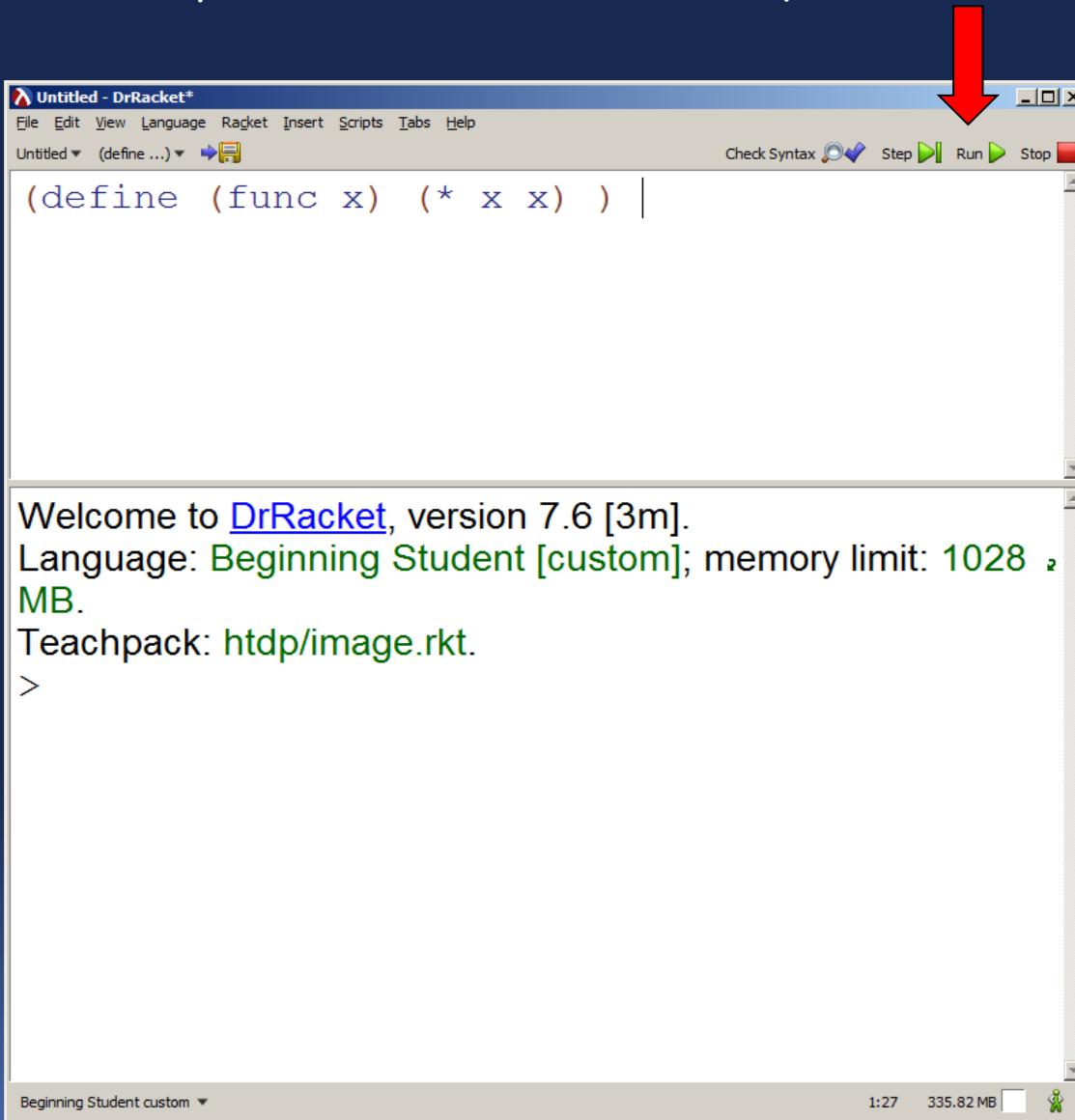
Definindo Funções

- ✓ Funções são definidas por outra função chamada **define**;
- ✓ Por exemplo: **(define (func x) (* x x))**

- ✓ **define** significa: “**considere func uma função**”, a qual, como uma expressão, computará um valor;
- ✓ O valor de uma função, depende, dos valores entrados (**input**), os quais são expressos por **(func x)**, sendo **func** o nome da função e **x** o **parâmetro** que será consumido por ela;
- ✓ Na segunda parte da definição, define-se a expressão na forma de **lista**, cuja avaliação retornará o valor computado.

Definindo Funções

✓ Após a escrita da função, clique em >Run.



The screenshot shows the DrRacket interface. In the top-left pane, there is a code editor with the following Racket code:

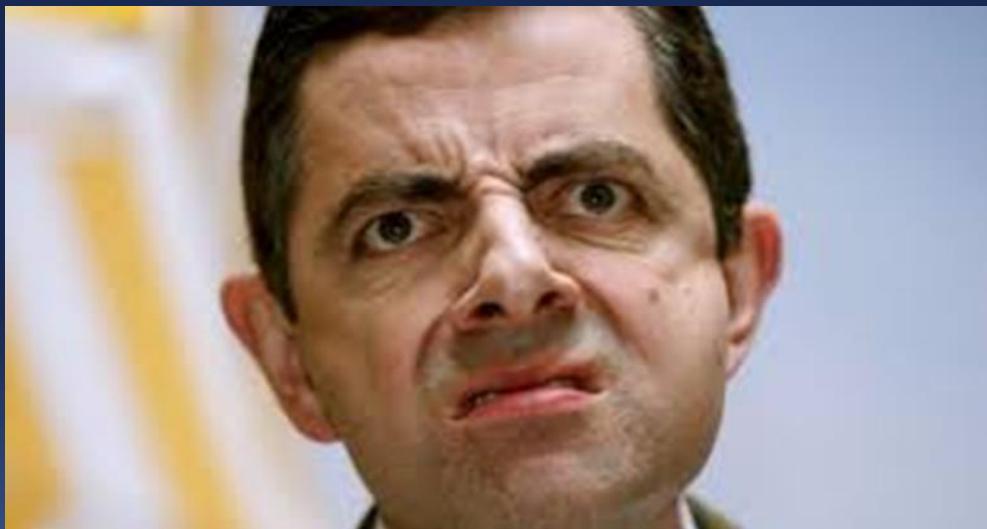
```
(define (func x) (* x x))
```

A large red arrow points from the text "Após a escrita da função, clique em >Run." down to the "Run" button in the toolbar. The toolbar also includes "Check Syntax", "Step", and "Stop" buttons. In the bottom-left pane, the Racket environment displays the following startup message:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
>

The status bar at the bottom shows "Beginning Student custom" and "1:27 335.82 MB".

Vé ... Mas, ao clicar no Botão nada
acontece ????!



Definindo Funções

- ✓ Aparentemente nada foi feito, mas DrRacket armazenou no ambiente a definição da função, podendo ser executada mais tarde.

The screenshot shows the DrRacket interface. The top window is titled "Untitled - DrRacket*". In the code editor, the following Racket code is shown:

```
(define (func x) (* x x))
```

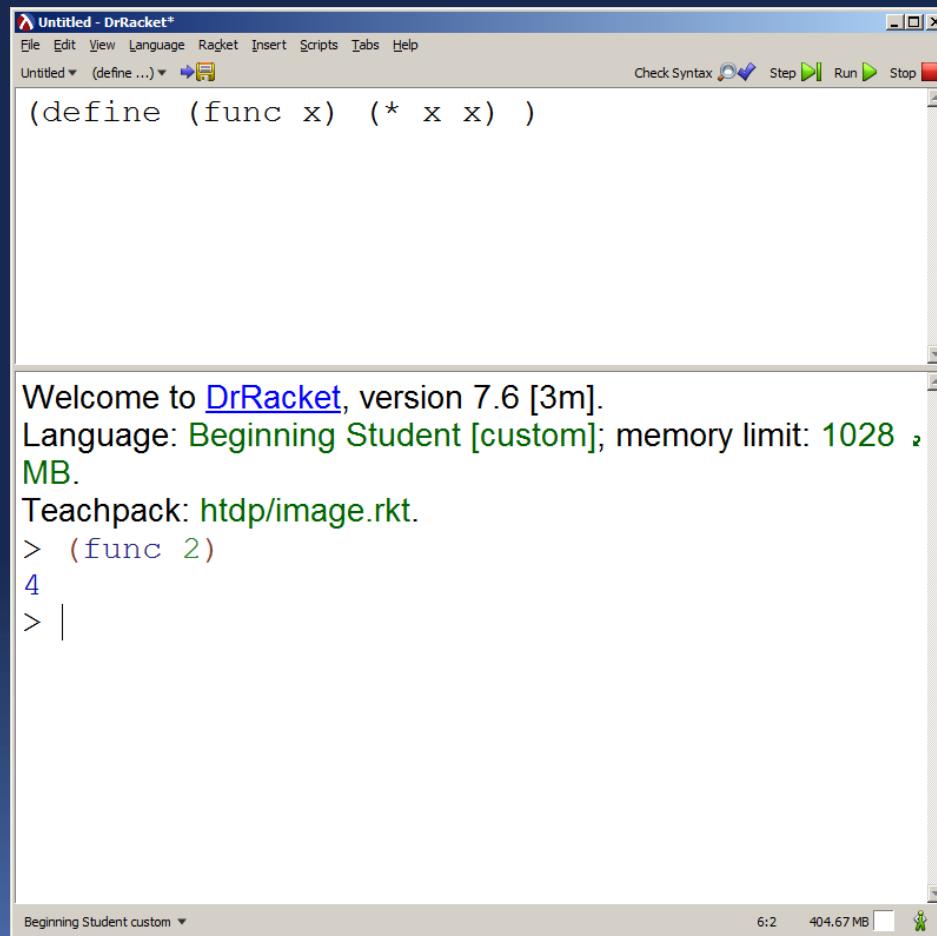
The bottom window displays the welcome message:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

The command prompt shows a greater than sign (>).

Aplicação da Função

- ✓ Após a definição da função, pode-se executá-la da seguinte forma, na área de interação: **(func 2)**
- ✓ **(func 2)** é chamado de Aplicação da Função.



The screenshot shows the DrRacket interface. In the top editor pane, the code `(define (func x) (* x x))` is written. In the bottom interaction pane, the text "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB." is displayed. Below this, the command `> (func 2)` is entered, followed by the result `4`. A red arrow points from the left towards the interaction pane, highlighting the application of the function.

```
(define (func x) (* x x))

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: httpd/image.rkt.
> (func 2)
4
> |
```

Aplicação da Função

- ✓ Pode-se definir a função e suas respectivas aplicações na área de definição e em seguida clicar em >Run.

The screenshot shows the DrRacket interface. In the top-left code editor pane, there is a green arrow pointing right towards the code area. The code defines a function `(func x) (* x x)` and applies it to values 3, 4, 5, and 10. In the bottom-right interactive window pane, there is another green arrow pointing right towards the output. The output shows the welcome message for DrRacket version 7.6, the language settings, and the results of applying the function to the specified values: 9, 16, 25, and 100. The status bar at the bottom indicates "All expressions are covered".

```
(define (func x) (* x x))  
(func 3)  
(func 4)  
(func 5)  
(func 10)  
  
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
9  
16  
25  
100  
> |
```

Funções em DrRacket

- ✓ No Paradigma Funcional **programas** são **funções**;
- ✓ Assim, a compreensão do conceito de função é vital para a Programação Funcional;
- ✓ Genericamente, definição da função é:

```
(define (FunctionName InputName) BodyExpression)
```

- ✓ “**define**” é uma keyword usada para definição de Função;
- ✓ A definição é feita em 2 partes: Na primeira parte, define-se o **nome** da função e os **parâmetros** que são desconhecidos até o momento de se efetuar a aplicação da função;
- ✓ Na segunda parte, define-se a **expressão** que representa o corpo da função usada para se **computar** o resultado.

Aplicação da Função

- ✓ Escrita por:

$$(FunctionName\ ArgumentExpression)$$

- ✓ Na primeira parte escreve-se o nome da função a ser aplicada;
- ✓ Na segunda parte escrevem-se os argumentos (inputs) que serão empregados para a aplicação da função.

Aplicação da Função - Exemplo



Untitled - DrRacket*

File Edit View Language Racket Insert Scripts Tabs Help

Untitled (define ...) ▾ Run Stop

Check Syntax Step Run Stop

```
(define (tamanho-string s)
  (string-length s) )

(define (dobro num)
  (+ num num) )

(dobro (tamanho-string "USCS") )
```

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

8
>

All expressions are covered Show next time?

Beginning Student custom 6:0 448.85 MB 

Descontinuidade de Funções

- ✓ Considere a seguinte função matemática:

$$\text{sign}(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

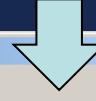
- ✓ Essa função tem três diferentes tipos de entrada: os valores que são maiores que zero, aqueles que são iguais a zero ou ainda aqueles que são menores que zero.
- ✓ Dependendo do argumento, o resultado da função poderá ser +1, 0 ou -1.

$$\text{sign}(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

- ✓ No DrRacket, essa função pode ser definida por:

```
(define ( sign x)
  (cond
    [ (> x 0) 1]
    [ (= x 0) 0]
    [ (< x 0) -1] ) )
```

$$\text{sign}(x) = \begin{cases} +1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$



Untitled 2 - DrRacket*

File Edit View Language Racket Insert Scripts Tabs Help

Untitled 2 (define ...) ▾

Check Syntax Step Run Stop

```
(define (sign x)
  (cond
    [(> x 0) 1]
    [= (x 0) 0]
    [(< x 0) -1]))
```

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

```
> (sign 2)
1
> (sign 0)
0
> (sign -1)
-1
> |
```

Beginning Student custom ▾

10:2 426.70 MB

Expressões Condicionais - Sintaxe

```
(cond  
  [ConditionExpression1 ResultExpression1]  
  [ConditionExpression2 ResultExpression2]  
  ...  
  [ConditionExpressionN ResultExpressionN])
```

- ✓ Em geral, **expressões condicionais** consistem de tantas linhas condicionais quantas forem necessárias;
- ✓ Cada linha contém duas expressões: a da **esquerda** define a **condição** e a da **direita** define o **resultado**;

Expressões Semântica

```
(cond  
  [ConditionExpression1 ResultExpression1]  
  [ConditionExpression2 ResultExpression2]  
  ...  
  [ConditionExpressionN ResultExpressionN] )
```

- ✓ DrRacket avalia a **primeira expressão condicional**. Se esta resultar **#true**, DrRacket usa o valor dessa linha para o resultado inteiro da expressão cond e **encerra a avaliação**;
- ✓ Se a avaliação for **falsa**, Dr. Racket salta essa linha e passa a avaliar a **próxima**, e assim, sucessivamente.
- ✓ Caso todas as condições forem **falsas**, DrRacket sinaliza **erro**.

Exemplo 1

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*" and contains the following Racket code:

```
(define (func x)
  (cond
    [ (< 2 0) 10]
    [ (= 1 0) "Olá"]
    [ (> 3 0) "Hello"]
    [ (= 2 1) "World..."]
    [ (= x 1) "Bingo..."] ))
```

The bottom window is a REPL (Read-Eval-Print Loop) window displaying the welcome message and a sample run:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

> (func 3)
"Hello"
>
```

The status bar at the bottom indicates "Beginning Student custom" and "6:2 462.51 MB".

Exemplo 3

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*". The code editor contains the following Racket code:

```
(define (func x)
  (cond
    [(< 2 0) 10]
    [(= 1 0) "Olá"]
    [(= 3 0) "Hello..."]
    [(= x 3) "World..."]
    [(= x 1) "Bingo..."])))
```

The bottom window displays the welcome message and the results of the function call:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (func 99)
cond: all question results were false
>
```

The status bar at the bottom indicates "Beginning Student custom" and "3:2 355.86 MB".

Definindo constantes

- ✓ Vimos nos exemplos anteriores que funções são definidas pela função **define**;
- ✓ Mas, **define** também pode ser usado para se definir **constants**;
- ✓ Ou seja, nesse caso atribuímos um nome a uma constante;
- ✓ A sintaxe geral para essa situação é:

```
(define Name Expression)
```

Definindo constantes

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*" and contains the following Racket code:

```
(define A 10)
(define B 20)
(define X 30)

(define (func a)
  (+ A B X a))
```

The bottom window is a REPL (Read-Eval-Print Loop) window displaying the welcome message and a response to a function call:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (func 1)
61
>
```

The status bar at the bottom indicates "Beginning Student custom" and "860.61 MB".

Definindo constantes

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*" and contains the following Racket code:

```
(define A 10)
(define B 20)
(define X (+ A B) )

(define (func Y)
  (+ X Y))
```

The bottom window displays the welcome message and a command-line interaction:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (func 5)
35
>
```

The status bar at the bottom indicates "Beginning Student custom" and shows system information: 6:2, 967.18 MB, and a battery icon.

Definindo constantes

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*". It contains the following Racket code:

```
; constantes

(define A 5)
(define B 20)
(define X (+ A B) )
(define Y (- B A) )

; funções
(define (func a b)

  (+ (+ a b) X Y) )
```

The bottom window displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

```
> (func 1 2)
43
>
```

At the bottom of the screen, there is a toolbar with icons for file operations, and a status bar showing "Beginning Student custom" and "6:2 1045.22 MB".

Operações com dados de Tamanho Fixo

- ✓ Toda linguagem de programação vem com uma **linguagem de dados** e uma outra **linguagem de operações** sobre os dados;
- ✓ A primeira linguagem sempre provê algumas **formas de dados atômicos**;
- ✓ Para representar informações do mundo real, o programador deve **aprender a compor** esses dados básicos e descrever tais **composições**;
- ✓ Similarmente, a **segunda linguagem** provê algumas **operações básicas** sobre os dados atômicos;
- ✓ Cabe ao programador **compor** essas operações em programas que executem as tarefas necessárias.

Tipos Atômicos de Dados

- ✓ Números
- ✓ Strings
- ✓ Imagens
- ✓ Valores Booleanos

Aritmética de Números

- ✓ Para aritmética de números utiliza-se a notação **prefixada**;
- ✓ Exemplo: (+ 3 4)
- ✓ Algumas operações primitivas disponíveis na linguagem:
 - ❖ +, -, *, /
 - ❖ abs, add1, ceiling, denominator, exact->inexact, expt, floor, gcd
 - ❖ log, max, numerator, quotient,
 - ❖ random, remainder, sqr, tan.

Exercício 1 – Aritmética de Números

- ✓ Adicione no ambiente de programação Racket:

```
(define a 3)  
(define b 4)
```

- ✓ Agora, imagine que **a** e **b** sejam as coordenadas de um ponto Cartesiano.
- ✓ Escreva uma **função** que retorne a **distância** desse ponto até a origem (0,0)

Exercício 1

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*" and contains the following Racket code:

```
(define a 3)
(define b 4)

(define (quadrado x)
  (* x x))

(define (raizquadrada x)
  (sqrt x))

(define (somaquadrados x y)
  (+
    (quadrado x)
    (quadrado y) ))

(define (distancia a b)
  (raizquadrada (somaquadrados a b) ) )
```

The bottom window is a terminal window displaying the welcome message and a few commands:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (distancia a b)
5
>
```

At the bottom left, there is a small window titled "Beginning Student custom" showing a grid of icons.

Aritmética de Strings

- ✓ *String* é uma sequência de caracteres que se pode entrar pelo teclado e são delimitadas entre apóstrofes ('');
- ✓ Algumas operações primitivas estão disponíveis sobre strings:
 - **string-length** consome um string e produz um número;
 - **string-ith** consome um string s junto com um número i e extrai o 1String localizado na iésima posição (contando de 0)
 - **number->string** consome um número e produz um string;
 - **string-append** concatena uma sequência de strings.

Aritmética de Strings – Exemplo

The screenshot shows the DrRacket IDE interface. The top part is the code editor with the title "Untitled 2 - DrRacket*". It contains the following Racket code:

```
(define prefix "hello")
(define suffix "world")
(define espaco " ")

(define (concatena a b c)
  (string-append a b c))
```

The bottom part is the interaction area, which displays the welcome message and a command-line session:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (concatena prefix espaco suffix)
"hello world"
>
```

The status bar at the bottom shows "Beginning Student custom" and "6:2 506.91 MB".

Aritmética de Strings – Exemplo

The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. A toolbar below the menu contains buttons for Check Syntax, Step, Run, and Stop. The code editor window displays the following Racket code:

```
(define prefix "hello")
(define suffix "world")
(define espaco " ")

(define (tamanhotexto a b c)
  (+
    (string-length a)
    (string-length b)
    (string-length c) ) )
```

The interaction window below shows the welcome message and the result of running the function:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (tamanhotexto prefix suffix espaco)
11
>
```

The status bar at the bottom indicates "Beginning Student custom" and "347.98 MB".

Exercício 2 – Aritmética de Strings

- ✓ Adicione no ambiente de programação Racket:

```
(define prefix "hello")
(define suffix "world")
```

- ✓ Escreva uma função chamada `concat` que concatena `prefix` com `suffix` e adiciona “-” entre eles.

Exercício 2 – Aritmética de Strings

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains buttons for Check Syntax, Step, Run, and Stop. The code editor contains the following Racket code:

```
(define prefix "hello")
(define suffix "world")
(define espaco " ")

(define (concat a b)
  (string-append a "-" b) )
```

The bottom window displays the Racket REPL output:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (concat prefix suffix)
"hello-world"
>
```

The status bar at the bottom indicates "Beginning Student custom" and "427.29 MB".

Exercício 3 – Aritmética de Strings

- ✓ Adicione no ambiente de programação Racket:

```
(define str "helloworld")
(define i 3)
```

- ✓ Escreva uma função usando operações primitivas chamada incluec que **adiciona** “_” na posição **i**.

Exercício 4 – Aritmética de Strings

- ✓ Adicione no ambiente de programação Racket:

```
(define str "helloworld")
(define i 3)
```

- ✓ Escreva uma função usando operações primitivas chamada **deletec** que **deleta** a posição **ith** a partir de **str**. (Que valores para **i** são legítimos?)

Aritmética de Strings

- ✓ String é uma **sequência** de caracteres que se pode entrar pelo teclado e são delimitadas entre apóstrofes ("');
- ✓ Algumas **operações primitivas** estão disponíveis sobre strings:
 - **string-length** consome um string e produz um número;
 - **string-ith** consome um string s junto com um número i e extrai o 1String localizado na iésima posição (contando de 0)
 - **number->string** consome um número e produz um string;
 - **string-append** concatena uma sequência de strings.

Aritmética de Imagens

- ✓ Uma **imagem** representa dados visuais, por exemplo, uma **foto**, uma **figura**, etc;
- ✓ Pode-se inserir imagens em DrRacket e manipulá-las tal como números e strings;
- ✓ Um programa DrRacket pode manipular imagens como operações primitivas;
- ✓ Operações básicas primitivas para criação de imagens: **circle**, **ellipse**, **line**, **rectangle**, **text**, **triangle**.

Operações de Criação de Imagens

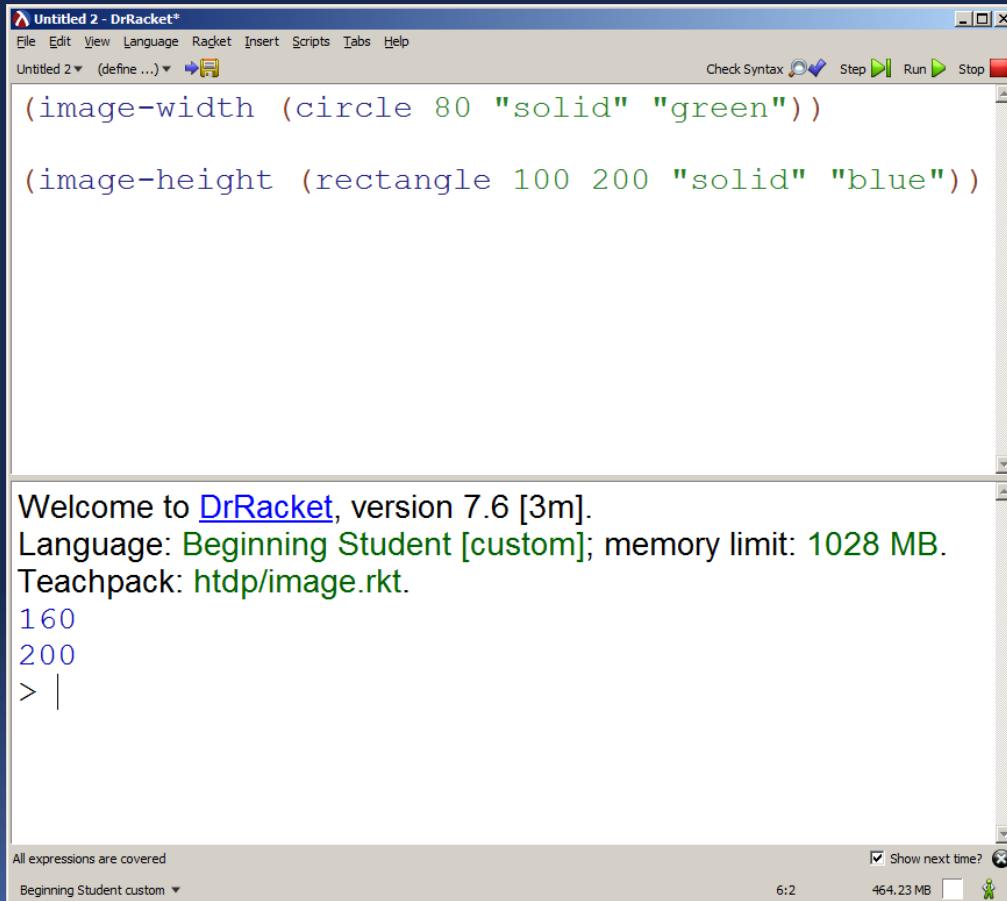
The screenshot shows the DrRacket interface with the following details:

- Title Bar:** Untitled 2 - DrRacket*
- Menu Bar:** File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help.
- Toolbar:** Check Syntax, Step, Run, Stop.
- Code Editor:** Contains the following racket code:

```
(circle 80 "solid" "green")
(rectangle 100 200 "solid" "blue")
```
- Welcome Message:** Welcome to [DrRacket](#), version 7.6 [3m].
- Language Information:** Language: Beginning Student [custom]; memory limit: 1028 MB.
- Teachpack:** Teachpack: `htdp/image.rkt`.
- Image Preview:** A green circle and a blue rectangle are displayed in the preview area.
- Status Bar:** All expressions are covered, Beginning Student custom, 2:18, 349.02 MB, Show next time? checkbox checked.

Funções em propriedades de Imagens

- ✓ `image-width` determina a **largura** de uma imagem em pixels
- ✓ `image-height` determina a **altura** de uma imagem;



The screenshot shows the DrRacket interface with two code snippets in the top editor window:

```
(image-width (circle 80 "solid" "green"))

(image-height (rectangle 100 200 "solid" "blue"))
```

In the bottom interaction window, the output is:

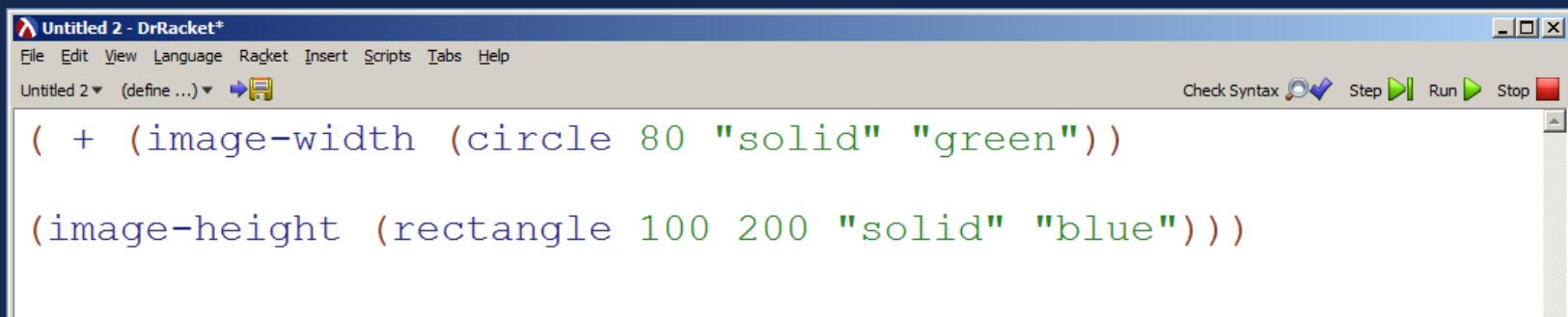
```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

160
200
> |
```

At the bottom of the interaction window, status information includes:

- All expressions are covered
- Show next time?
- Beginning Student custom
- 6:2
- 464.23 MB
- A small green figure icon

Qual o valor dessa expressão ?



The screenshot shows the DrRacket IDE interface with the title bar "Untitled 2 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar features "Untitled 2", "(define ...)", and icons for Check Syntax, Step, Run, and Stop. The code editor contains the following Racket code:

```
( + (image-width (circle 80 "solid" "green")))  
(image-height (rectangle 100 200 "solid" "blue")) )
```

Qual o valor dessa expressão ?

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled 2 - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains buttons for Check Syntax, Step, Run, and Stop. The code area contains the following Racket expression:

```
( + (image-width (circle 80 "solid" "green"))  
     (image-height (rectangle 100 200 "solid" "blue")) )
```

The bottom window displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

360
>

At the bottom of the DrRacket window, there are status messages: "All expressions are covered", "Beginning Student custom", "Show next time?", "5:2", "522.55 MB", and a small green figure icon.

Aritmética de Booleans

- ✓ Tipos primitivos booleanos são úteis no projeto de programas;
- ✓ Há dois tipos de valores booleanos: **#true** e **#false**.
- ✓ Programas usam os tipos booleanos para implementar **decisões**;
- ✓ Há três operações primitivas sobre booleans: **or**, **and** e **not**.

Aritmética de Booleans

```
> (or #true #true)
#true
> (or #true #false)
#true
> (or #false #true)
#true
> (or #false #false)
#false
```

```
> (and #true #true)
#true
> (and #true #false)
#false
> (and #false #true)
#false
> (and #false #false)
#false
```

```
> (not #true)
#false
```

Misturando operações

- ✓ Operações com tipos **booleanos** podem manusear diferentes tipos de dados;
- ✓ No exemplo, a função “**if**” é seguida por 3 expressões separadas por espaços em branco, que são chamadas de **sub-expressões**.
- ✓ Se a avaliação da primeira expressão é **#true**, então a segunda expressão é avaliada, **caso contrário** a terceira expressão será avaliada.

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". It contains the following Racket code:

```
(define (inverso x)
  (if (= x 0)
      0
      (/ 1 x) ) )
```

The bottom window is titled "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB. Teachpack: htdp/image.rkt." It shows the execution of the function:

```
> (inverso 2)
0.5
> (inverso 0)
0
>
```

Comparando valores numéricos

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (compara_numeros x y)
  (if (= x y)
      "Valores entrados são iguais"
      "Valores entrados diferentes"))
```

The bottom window is the DrRacket console, displaying the welcome message and some sample interactions:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (compara_numeros 1 1.1)
"Valores entrados diferentes"
> (compara_numeros 2.0 2.0)
"Valores entrados são iguais"
> |
```

The status bar at the bottom of the DrRacket window indicates "Beginning Student custom" and "8:2 384.85 MB".

Comparando valores numéricos

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (compara_numeros x y)
  (if (<= x y)
      (if (= x y)
          "Valores entrados iguais"
          "Primeiro valor menor que o segundo")
      "Primeiro valor maior que o segundo"))
```

The bottom window is titled "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB. Teachpack: htdp/image.rkt." and shows the results of running the function:

```
> (compara_numeros 2 2)
"Valores entrados iguais"
> (compara_numeros 1 2)
"Primeiro valor menor que o segundo"
> (compara_numeros 5 1)
"Primeiro valor maior que o segundo"
>
```

The status bar at the bottom indicates "Beginning Student custom" and "537.98 MB".

Comparando strings

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (compara_strings x y)
  (if (string=? x y)
      "Strings entrados iguais"
      "Strings entrados diferentes..."))
```

The bottom window displays the welcome message and environment details:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

```
> (compara_strings "USCS" "Uscs")
"Strings entrados diferentes..."
> (compara_strings "uscs" "uscs")
"Strings entrados iguais"
> |
```

At the bottom of the DrRacket window, it says "Beginning Student custom". The status bar at the bottom right shows "8:2" and "342.97 MB".

Predicados

- ✓ Um **predicado** é uma função que **consome** um valor e determina se ele **pertence** ou **não** à uma classe de dados. Por exemplo, o predicado **number?** determina se o argumento passado é ou não um número.

The screenshot shows the DrRacket IDE interface. The title bar says "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The main window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom];  
memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (number? 4)  
#true  
> (number? pi)  
#true  
> (number? "Hello")  
#false  
>
```

At the bottom, a status bar shows "All expressions are covered" and "Beginning Student custom". There are also icons for "Show next time?", a progress bar at 1:0, and memory usage at 480.33 MB.

Predicados

- ✓ O uso de **predicados** auxilia no tratamento de erros em funções.

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (tamanho_string in)
  (if (string? in)
      (string-length in)
      "Argumento entrado inválido!"))
```

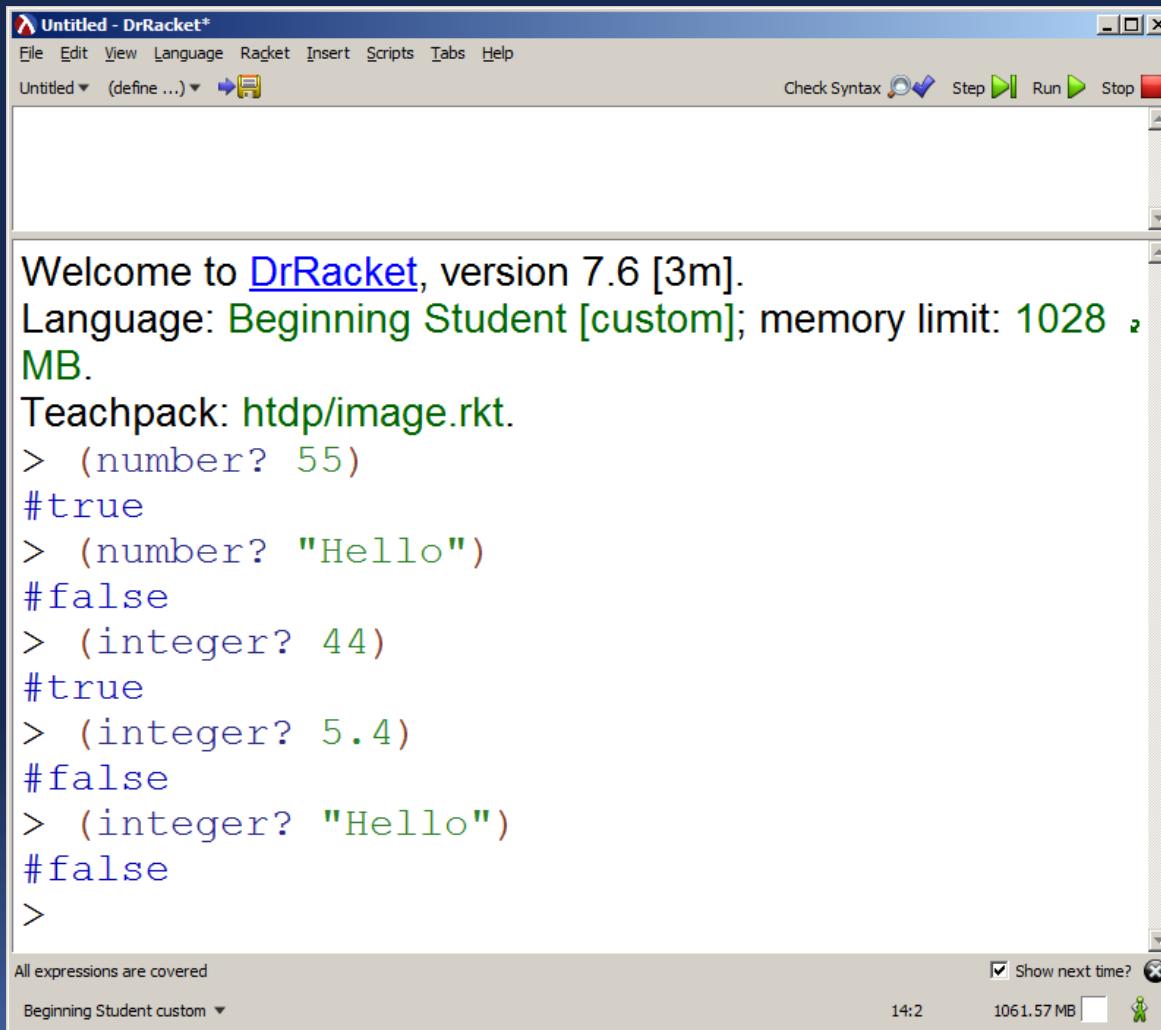
The bottom window is titled "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB." It shows the execution of the function:

```
> (tamanho_string "Hello")
5
> (tamanho_string 34543)
"Argumento entrado inválido!"
```

The status bar at the bottom indicates "Beginning Student custom" and "8:2 749.07 MB".

Predicados

✓ Predicado **integer?**



The screenshot shows the DrRacket IDE interface with the title bar "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar features "Check Syntax" (blue checkmark), "Step" (green arrow), "Run" (yellow arrow), and "Stop" (red square). The main window displays the following Racket session:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (number? 55)  
#true  
> (number? "Hello")  
#false  
> (integer? 44)  
#true  
> (integer? 5.4)  
#false  
> (integer? "Hello")  
#false  
>  
All expressions are covered  
Beginning Student custom ▾
```

The status bar at the bottom right shows the time as 14:2 and memory usage as 1061.57 MB. A checkbox for "Show next time?" is checked.

Predicados

✓ Predicado rational?

The screenshot shows the DrRacket IDE interface. The title bar says "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The code editor window displays the following session:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> pi  
#i3.141592653589793  
> (rational? pi)  
#true  
> (rational? 2)  
#true  
> (rational? 1/3)  
#true  
>  
  
All expressions are covered  
Beginning Student custom ▾
```

The status bar at the bottom right shows the time as 12:2 and memory usage as 1035.91 MB. There is also a "Show next time?" checkbox.

Predicados

✓ Predicado real?

The screenshot shows the DrRacket IDE interface. The title bar says "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Untitled, (define ...), Check Syntax, Step, Run, and Stop. The main window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> pi  
#i3.141592653589793  
> (real? pi)  
#true  
> (real? 5)  
#true  
> (real? 4/5)  
#true  

```

At the bottom, it says "All expressions are covered" and "Beginning Student custom". There are checkboxes for "Show next time?" and "Beginning Student custom". The status bar shows the time as 14:2 and memory usage as 1096.46 MB.

Predicados

✓ Predicado complex?

The screenshot shows the DrRacket IDE interface. The title bar says "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar has buttons for Untitled, (define ...), Check Syntax, Step, Run, and Stop. The main window displays the following text:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

```
> (complex? 1+2i)
#true
> (complex? 5-9i)
#true
>
```

At the bottom, it says "All expressions are covered" and "Beginning Student custom". There are checkboxes for "Show next time?" and "Run at startup". The status bar shows "8:2" and "1395.77 MB".

Funções e Programas em Racket

- ✓ No Paradigma **Funcional** de Programação **funções** são **programas**;
- ✓ Como **funções**, **programas** consomem **entradas** e produzem **saídas**;
- ✓ Programas trabalham com diferentes tipos de dados;
- ✓ Programas em Racket consistem de diversas definições, geralmente seguidas por uma expressão que envolve essas definições. Existem 2 tipos de definições:
 - ➡❖ Definições de Constantes
 - ➡❖ Definições de Funções

Definição de Constantes

- ✓ Variáveis **NÃO** são dados; elas representam dados;
- ✓ Por exemplo:

➡ (define x 3)

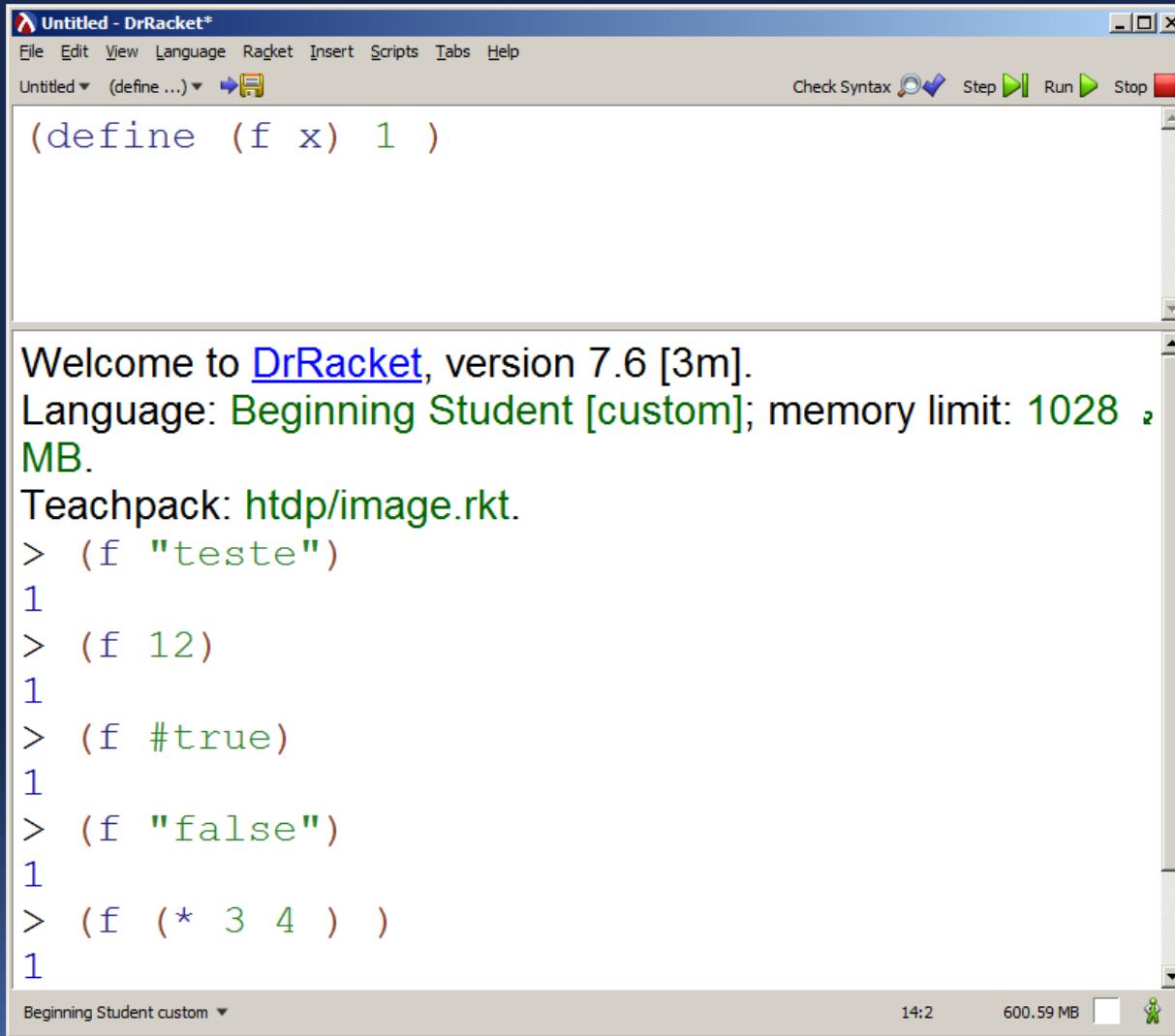
- ✓ **Essa definição diz que x sempre estará associado ao valor numérico 3;**

Definição de Funções

✓ Para se **definir** uma **função**, escrevemos:

- ❖ " **(define** ("
- ❖ O **nome** da Função
- ❖ Seguido por diversas **variáveis**, separadas por **espaço** e terminadas com ")".
- ❖ E uma expressão seguida por ")".

Exemplos de Definição de Funções



The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu is a toolbar with "Untitled" and "(define ...)" dropdowns, and buttons for Check Syntax, Step, Run, and Stop. The main code editor contains the following Racket code:

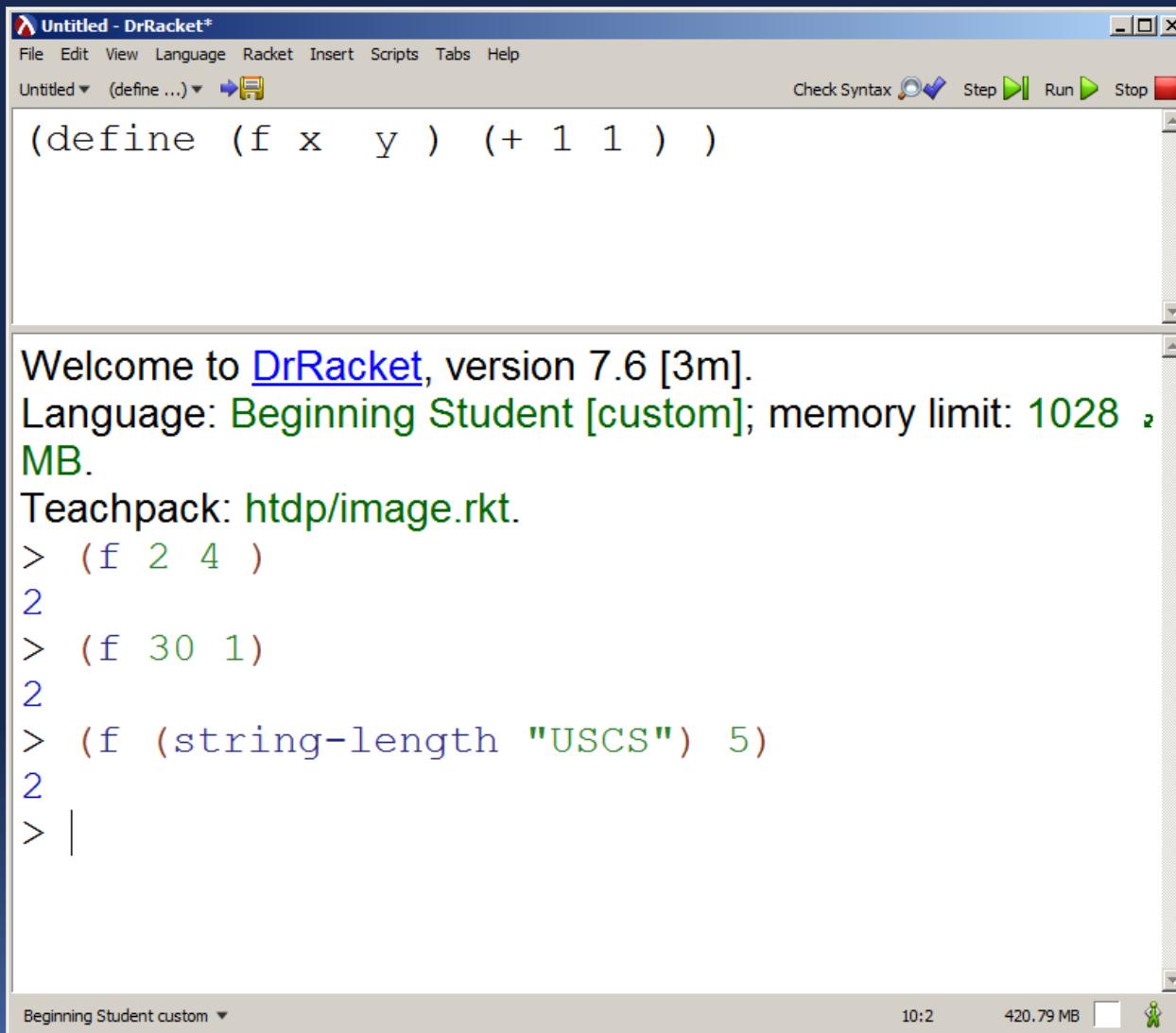
```
(define (f x) 1)
```

The output pane below the editor displays the welcome message and a series of interactions:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdd/image.rkt.  
> (f "teste")  
1  
> (f 12)  
1  
> (f #true)  
1  
> (f "false")  
1  
> (f (* 3 4))  
1
```

The status bar at the bottom shows "Beginning Student custom" and system information like time (14:2) and memory usage (600.59 MB).

Exemplos de Definição de Funções



The screenshot shows the DrRacket IDE interface. The top window title bar reads "Untitled - DrRacket®". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu is a toolbar with "Check Syntax" and "Run" buttons. The code editor contains the following Racket code:

```
(define (f x y) (+ 1 1))
```

The DrRacket console window displays the following output:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (f 2 4)  
2  
> (f 30 1)  
2  
> (f (string-length "USCS") 5)  
2  
> |
```

The status bar at the bottom shows "Beginning Student custom" and system information like "10:2", "420.79 MB", and a battery icon.

Exemplos de Definição de Funções

The screenshot shows the DrRacket IDE interface. The top window title is "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains buttons for Check Syntax, Step, Run, and Stop. The code editor window displays the following Racket code:

```
(define (g x y) (+ (* 2 3) 5))
```

The DrRacket welcome message is displayed below the code editor:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

The interactive window shows the results of evaluating the function g:

```
> (g 1 2)
11
> (g 999 555)
11
> (g 44 (+ 4 5))
11
>
```

The status bar at the bottom indicates "Beginning Student custom" and system information like "10:2" and "384.72 MB".

Exemplos de Definição de Funções

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the Racket code:

```
(define (h a b c) (+ (+ 10 5) 2 c))
```

The bottom window displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

Interaction history:

```
> (h 1 1 1)
18
> (h 999 555 1)
18
> (h 1 2 3)
20
> |
```

At the bottom of the interface, there is a status bar with the text "Beginning Student custom ▾" and system information like "10:2", "521.80 MB", and a battery icon.

Funções - Observações

- ✓ Uma **definição** de uma **nova função** introduz no ambiente uma **nova operação** nos dados;
- ✓ Como uma **função primitiva**, uma **nova** função **consome** entradas;
- ✓ O **número** de variáveis definidas na função determina quantas **entradas** (**argumentos** ou **parâmetros**) a função **consome**;
- ✓ Se **f** é uma função de **um-argumento**, dizemos que ela é **unária**;
- ✓ Em contraste, uma função com definida com **2 argumentos**, é **binária**, e assim, por diante;
- ✓ A expressão definida na função determina o seu **corpo (body)**

Funções - Observações

- ✓ O corpo da função **pode** não envolver variáveis. Nesse caso, a função **sempre retorna o mesmo valor**, resultado da avaliação.

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". It contains the Racket code:

```
(define (func x) "Hello World....")
```

Below this, the DrRacket welcome message is displayed:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

The interaction window shows the following session:

```
> (func 3)  
"Hello World...."  
> (func 10)  
"Hello World...."  
> (func "USCS")  
"Hello World...."  
>
```

The status bar at the bottom indicates "Beginning Student custom" and "10:2 423.73 MB".

Funções - Observações

- ✓ O **corpo da função pode** não envolver variáveis. Nesse caso, a função **sempre retorna o mesmo valor**, resultado da avaliação.

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". It contains the following code:

```
(define (func x) 999 )
```

The bottom window is titled "Welcome to DrRacket, version 7.6 [3m]. Language: Beginning Student [custom]; memory limit: 1028 MB." It displays the following interaction:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.

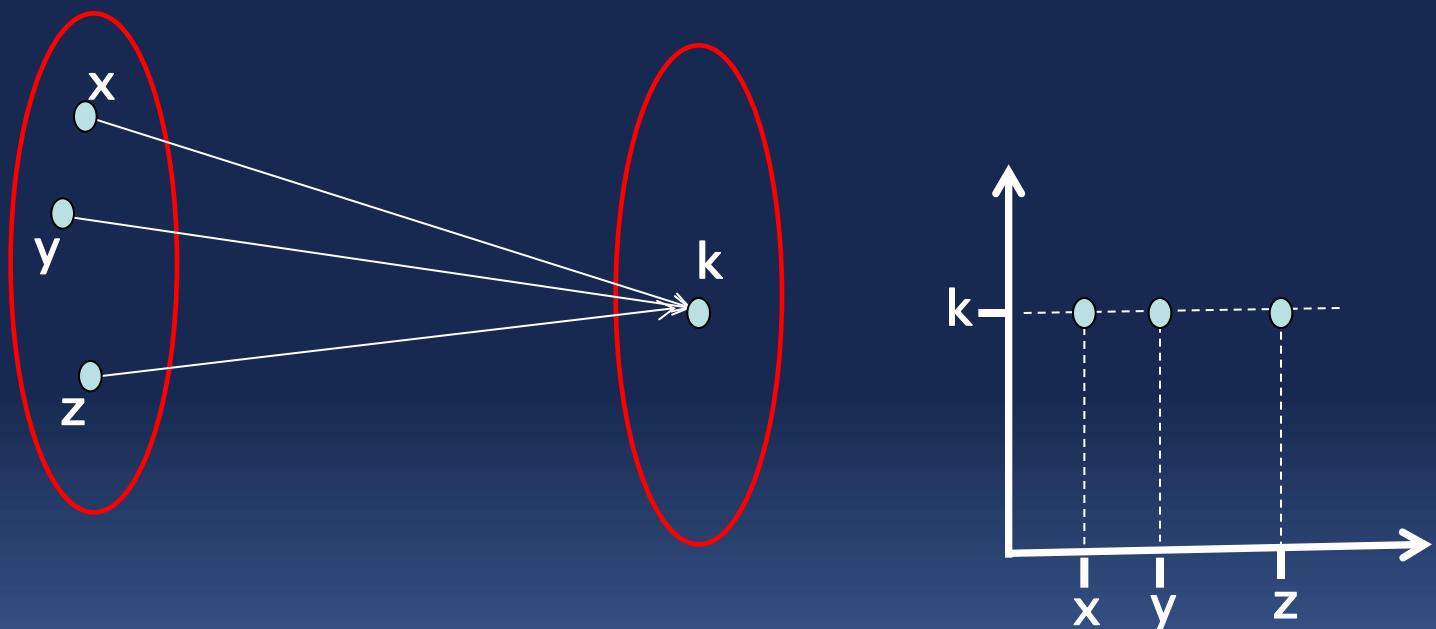
Teachpack: htdp/image.rkt.

> (func "Hello World...")
999
> (func (or true false))
999
> (func 1234)
999
> (func (+ 1 2))
999
> |
```

The status bar at the bottom shows "Beginning Student custom" and "12:2 738.04 MB".

Funções - Observações

- ✓ Variáveis **não** são dados;
- ✓ **Variáveis** apenas **representam entradas** (inputs);
- ✓ Quando as variáveis **não** são mencionadas no corpo da função (expressão) significa que a saída da função é **independente** das entradas e, portanto a função sempre retornará, nesse caso, o **mesmo valor** (**constante**).



Define para definir constantes

- ✓ **define** pode ser usado para definir **constantes**;
- ✓ Por exemplo: (**define x 99**) diz que **x** representa o valor constante **99**;

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". In the code editor, the expression `(define x 99)` is typed. Below the editor, the DrRacket REPL displays the following interaction:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (+ 3 x)  
102  
> (- (+ x (string-length "USCS")) x) )  
-202  
>
```

The status bar at the bottom indicates "All expressions are covered" and "Beginning Student custom".

Funções - Observações

- ✓ As variáveis na definição da função (header) são peças de dados **desconhecidas** (as entradas da função);
- ✓ Mencionar variáveis no corpo da função é a forma pela qual usamos essas peças de dados no instante de aplicação da função;
- ✓ No instante de aplicação da função, os valores dessas peças de dados (variáveis) são conhecidas
- ✓ Ou seja, em tempo de definição da função os valores das variáveis (inputs) não são conhecidos;
- ✓ Por exemplo, se definimos:

```
(define (ff a)
      (* 10 a))
```

- ✓ Estamos dizendo no corpo da função que a saída da função é 10 vezes a sua entrada.

Aplicação de uma função

- ✓ Uma aplicação de uma função a coloca para trabalhar;
- ✓ A forma como se define a aplicação é semelhante à definição da função:

- ❖ “(“
- ❖ seguido do nome da função f
- ❖ seguido de tantos argumentos quantos f consome, separados por espaços
- ❖ e finalmente, um ")"

- ✓ Exemplo:
 - > (define (func x) 1) => definição da função
 - > (f "Hello World") => aplicação da função
1



The screenshot shows the DrRacket IDE interface. The title bar says "Untitled - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Untitled, (define ...), Check Syntax, Step, Run, and Stop. The code editor window contains the following Racket code:

```
(define (func x) 1)
```

Below the code editor, the DrRacket welcome message is displayed:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

The interaction window shows the following session:

```
> (func "Hello World...")  
1  
>
```

The status bar at the bottom right shows "Beginning Student custom" and "1413.60 MB".

Aplicações nested

The screenshot shows the DrRacket interface. The top window is titled "Untitled - DrRacket*". It contains the Racket code:

```
(define (func x) (+ x 1))
```

Below this is a welcome message:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student
[custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

The DrRacket repl shows the following interaction:

```
> (+ (func (func 2)) 100)  
104  
> |
```

The status bar at the bottom indicates "Beginning Student custom" and "6:2 1692.40 MB".

Computações

- ✓ Definição de função e aplicação de função caminham juntas;
- ✓ A aplicação de uma função envolve 3 etapas:
 1. Racket determina os valores das expressões dos argumentos;
 2. Racket checa se o número de argumentos coincide com o número de parâmetros na definição da função;
 3. Se sim, Racket computa o valor no corpo da função (avaliação da expressão), com todos os parâmetros substituídos pelos correspondentes valores dos argumentos.

Exercício 1

Qual o resultado ?

The screenshot shows the DrRacket IDE interface. The top window title is "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The code editor contains the following Racket code:

```
(define (distancia-a-origem x y)
  (sqrt (+ (sqr x) (sqr y) ) ) )

(distancia-a-origem 3 4 )
```

The interaction window below displays the welcome message and environment details:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

>

At the bottom of the interaction window, there is a status bar with "Beginning Student custom" and a clock/timer.

Exercício 1

Resposta:

The screenshot shows the DrRacket IDE interface. The code editor contains the following Racket code:

```
(define (distancia-a-origem x y)
  (sqrt (+ (sqr x) (sqr y)) ))  
  
(distancia-a-origem 3 4 )
```

The DrRacket welcome message is displayed below the code:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: `htdp/image.rkt`.

The Racket repl shows the following interaction:

```
5  
>
```

At the bottom of the window, status messages indicate:

- All expressions are covered
- Show next time? (checkbox checked)
- Beginning Student custom
- 5:2
- 605.98 MB

Exercício 2

Qual o resultado ?

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". It contains the following Racket code:

```
(define (string-first s)
  (substring s 0 1) )  
  
(string-first "Hello World...")
```

The bottom window displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).
>

The status bar at the bottom indicates "Beginning Student custom" and "343.15 MB".



Exercício 2

Resposta:

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket". It contains the following Racket code:

```
(define (string-first s)
  (substring s 0 1) )

(string-first "Hello World...")
```

The bottom window is a REPL (Read-Eval-Print Loop) window. It displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.

"H"
> |

At the bottom of the DrRacket window, there is a status bar with the following information:

- All expressions are covered
- Show next time?
- Beginning Student custom
- 5:2
- 417.83 MB
- A small green person icon

Exercício 3

Qual o resultado?

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (===> x y)
  (or (not x) y))

(===> true false)
```

The bottom window is a "Welcome" window with the following text:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).
>

At the bottom of the screen, there is a status bar with the text "Beginning Student custom" and system information like "4:2", "553.96 MB", and a battery icon.

Exercício 3

Qual o resultado?

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*" and contains the following Racket code:

```
(define (===> x y)
  (or (not x) y))

(===> true false)
```

The bottom window displays the welcome message for DrRacket version 7.6 [3m]:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

The status bar at the bottom indicates "Beginning Student custom" and "4:2 553.96 MB".

Exercício 3

Qual o resultado?

The screenshot shows the DrRacket IDE interface. The top part is the code editor with the following Racket code:

```
(define (===> x y)
  (or (not x) y))

(===> true false)
```

The bottom part is the interaction area displaying the welcome message and the result of the evaluation:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
#false
>
```

At the bottom of the interaction area, there is a status bar with the following information:

- All expressions are covered
- Show next time? checkbox (unchecked)
- Beginning Student custom dropdown
- 5:2
- 326.09 MB
- A small green figure icon

Exercício 4

Qual o resultado?

The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Untitled, (define ...), Check Syntax, Step, Run, and Stop. The code editor contains the following Racket code:

```
(define (string-insert s i)
  (string-append (substring s 0 i)
                 " "
                 (substring s i) ) )

(string-insert "Helloworld" 5)
```

The interaction window displays the welcome message and environment information:

Welcome to [DrRacket](#), version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: [htdp/image.rkt](#).

>

At the bottom, the status bar shows Beginning Student custom ▾ 4:2 399.08 MB and a small green person icon.

Exercício 4

Resposta:

The screenshot shows the DrRacket IDE interface. The top window is titled "Untitled - DrRacket*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains buttons for Check Syntax, Step, Run, and Stop. The code editor contains the following Racket code:

```
(define (string-insert s i)
  (string-append (substring s 0 i)
                 "—"
                 (substring s i) ) )

(string-insert "Helloworld" 5)
```

The output window below displays the welcome message and the result of running the code:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
"Hello-world"
>
```

At the bottom of the output window, status messages include "All expressions are covered" and "Beginning Student custom". There are also checkboxes for "Show next time?" and "5:2" and "450.03 MB".

Composição de Funções

- ✓ Um programa tipicamente consiste de uma definição principal (**main**) de função e diversas outras funções;
- ✓ A computação tipicamente ocorre no processamento de uma função, no qual a **saída** dela será canalizada (entrada) para a **entrada** de outra função e assim, sucessivamente, até completar-se a computação;
- ✓ Essa forma de computação é **conhecida** por **composição de funções** e as funções adicionais são **conhecidas** por funções **auxiliares** (**helpers**);

Composição de Funções - Exemplo

The screenshot shows the DrRacket IDE interface. The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Untitled, (define ...), and a toolbar with Check Syntax, Step, Run, and Stop buttons.

```
(define (carta first last assinatura)
  (string-append
    (opening first)
    "\n\n"
    (body first last)
    "\n\n"
    (closing assinatura) ) )

(define (opening first)
  (string-append "Caro " first ","))

(define (body first last)
  (string-append
    "Olá, tudo bem?\n "
    first
    " de "
    last) )

(define (closing assinatura)
  (string-append
    "Um grande abraço! "
    "\n\n"
    assinatura
    "\n\n") )

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: httpd/image.rkt.
> (carta "Aparecido" "Freitas" "assinatura")
"Caro Aparecido,\n\nOlá, tudo bem?\n Aparecido de Freitas\n\nUm grande abraço! \n\nassinatura\n\n"
```

The status bar at the bottom shows "Beginning Student custom" and system information like "12:13" and "511.68 MB".

Composição de Funções - Exemplo

The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, Help, Untitled, (define ...), and a toolbar with Check Syntax, Step, Run, and Stop buttons.

The code area contains the following Racket code:

```
(define (carta first last assinatura)
  (string-append
    (opening first)
    "\n\n"
    (body first last)
    "\n\n"
    (closing assinatura) ) )

(define (opening first)
  (string-append "Caro " first ",,") )

(define (body first last)
  (string-append
    "Olá, tudo bem?\n "
    first
    " de "
    last) )

(define (closing assinatura)
  (string-append
    "Um grande abraço! "
    "\n\n"
    assinatura
    "\n\n") )
```

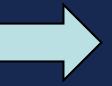
The output window displays the welcome message and the result of running the code with specific arguments:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (carta "Aparecido" "Freitas" "assinatura")
"Caro Aparecido,\n\nOlá, tudo bem?\n Aparecido de Freitas\n\nUm grande abraço! \n\nassinatura\n\n"
```

The status bar at the bottom shows "Beginning Student custom", "12:13", "511.68 MB", and a battery icon.

Adicionando para console

- ✓ No programa anterior, adicione na primeira linha: **(require 2htdp/batch-io)**
- ✓ Para adicionar a função **(write-file)** em nosso repertório de funções.



```
Untitled - DrRacket*  
File Edit View Language Racket Insert Scripts Tabs Help  
Untitled (define ...) Check Syntax Step Run Stop  
(require 2htdp/batch-io)  
  
(define (carta first last assinatura)  
  (string-append  
    (opening first)  
    "\n\n"  
    (body first last)  
    "\n\n"  
    (closing assinatura) ) )  
  
(define (opening first)  
  (string-append "Caro " first ",") )  
  
(define (body first last)  
  (string-append  
    "Olá, tudo bem?\n "  
    first  
    " de "  
    last) )  
  
(define (closing assinatura)  
  (string-append  
    "Um grande abraço! "  
  
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit:  
1028 MB.  
Teachpack: htdp/image.rkt.  
>  
  
Beginning Student custom 24:23 357.40 MB
```

Adicionando para console

- ✓ Agora na área de Interação: (write-file 'stdout (carta "Aparecido" "Freitas")))

The screenshot shows the DrRacket IDE interface. The code area contains the following Racket code:

```
(define (carta first last assinatura)
  (string-append
    (opening first)
    "\n\n"
    (body first last)
    "\n\n"
    (closing assinatura) ) )

(define (opening first)
  (string-append "Caro " first ",") )

(define (body first last)
  (string-append
    "Olá, tudo bem?\n "
    
```

The interaction area shows the welcome message and the result of running the code:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
> (write-file 'stdout (carta "Aparecido" "Freitas" "assinatura"))
Caro Aparecido,
```

Output:

```
Olá, tudo bem?
Aparecido de Freitas

Um grande abraço!
```

```
assinatura
'standard
>
```

Boa prática



- ✓ Defina uma função para cada **tarefa** (task);
- ✓ Em geral, quando um problema se refere à distintas tarefas de computação, um programa deveria consistir de **uma função para cada tarefa** e **uma função principal** (**main**) que efetua a composição;
- ✓ Essa estratégia favorece a compreensão do programa, uma vez que funções razoavelmente pequenas são mais fáceis de serem entendidas e, por consequência, a composição também será mais fácil de ser compreendida;
- ✓ Além disso, esta estratégia favorece a manutenção (**Software muda!!!**)