



# Modelos de Linguagem de Programação

## Unidade 7 – Programação Funcional com a Linguagem Racket

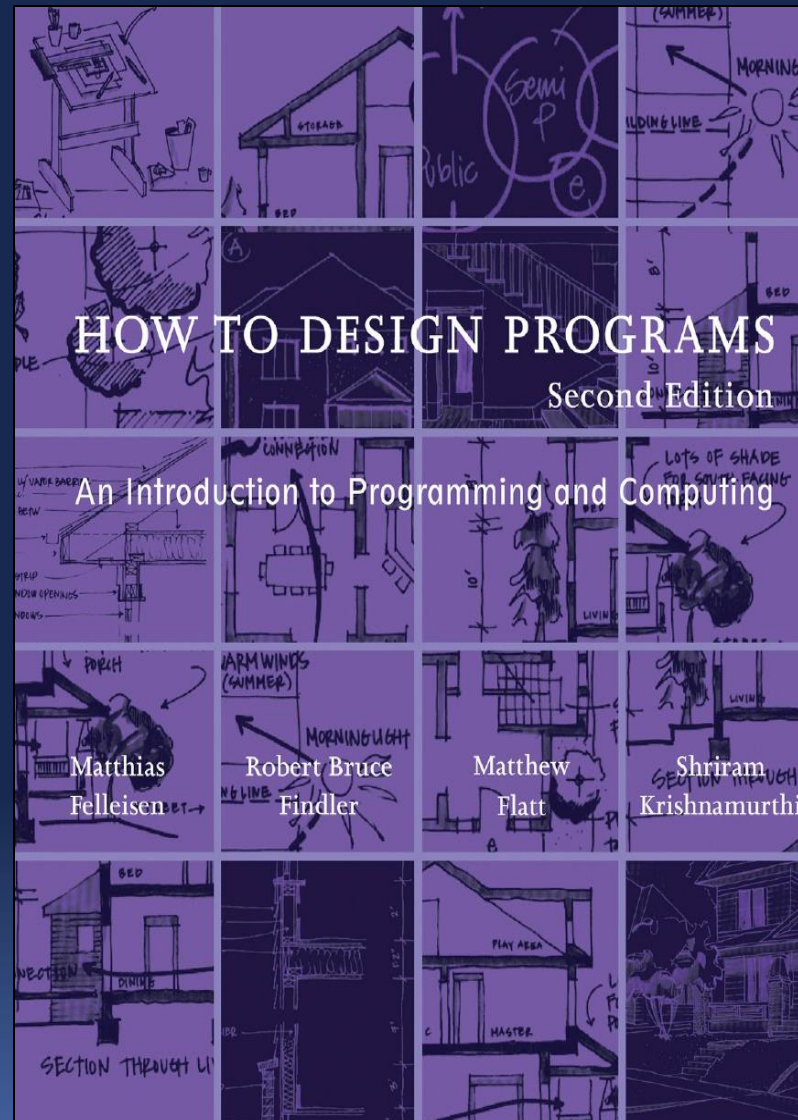


Prof. Aparecido V. de Freitas  
Doutor em Engenharia  
da Computação pela EPUSP  
[aparecidovfreitas@gmail.com](mailto:aparecidovfreitas@gmail.com)





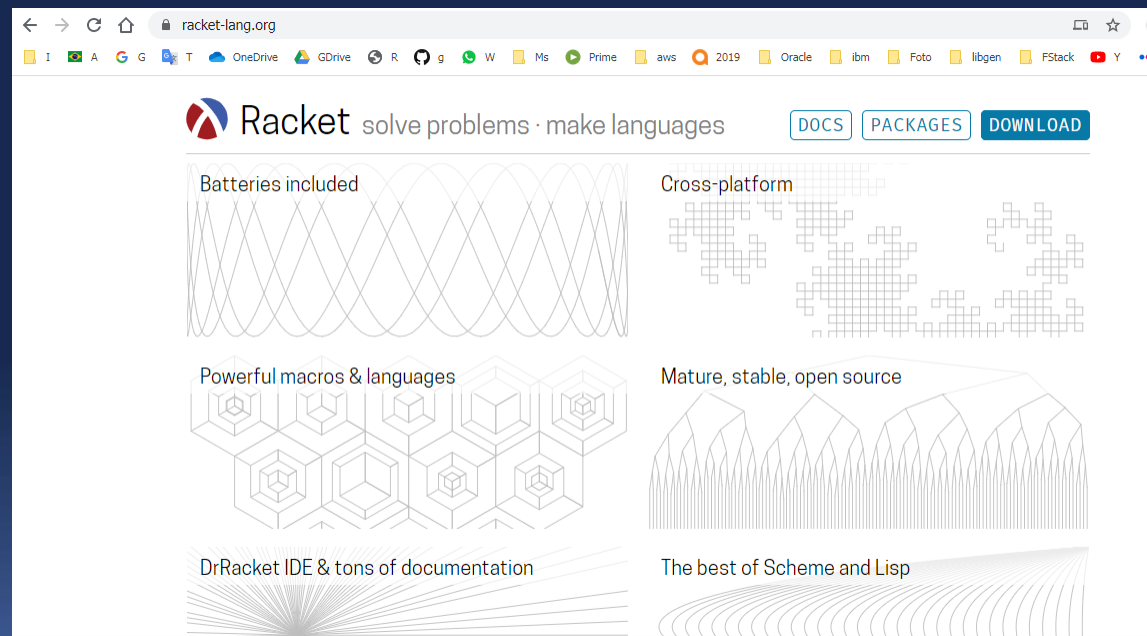
# Bibliografia





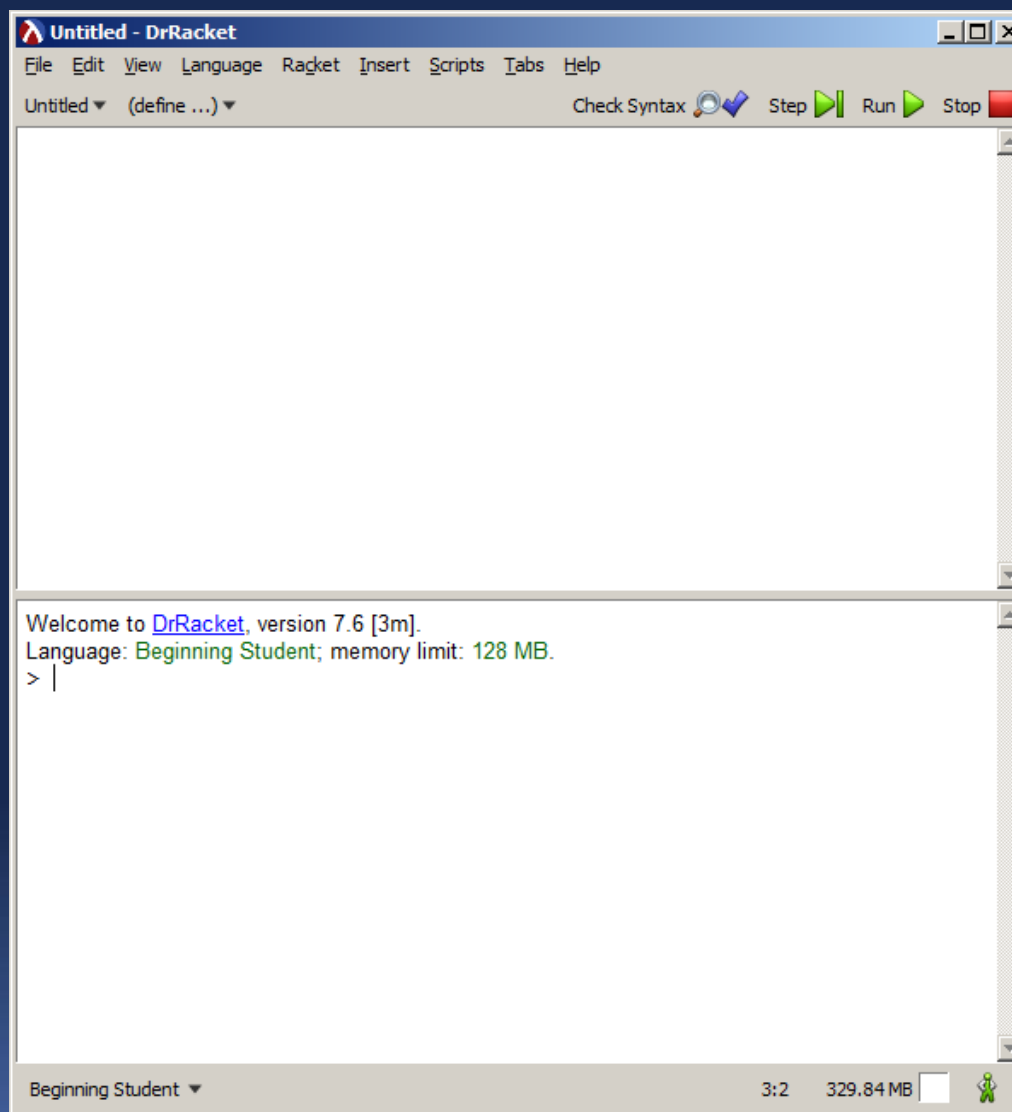
# Introdução

- ✓ 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/>





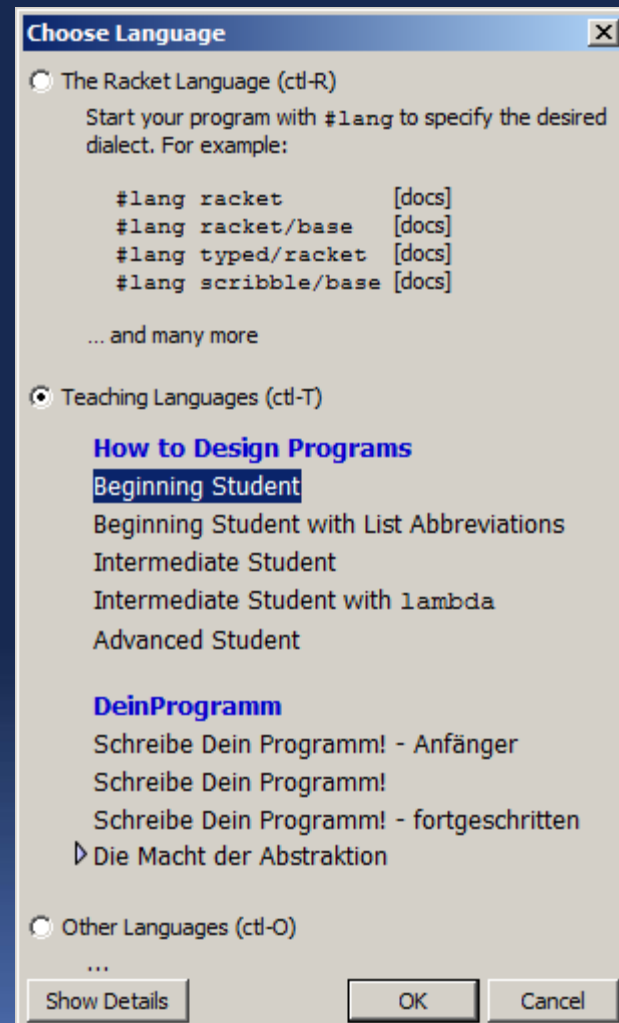
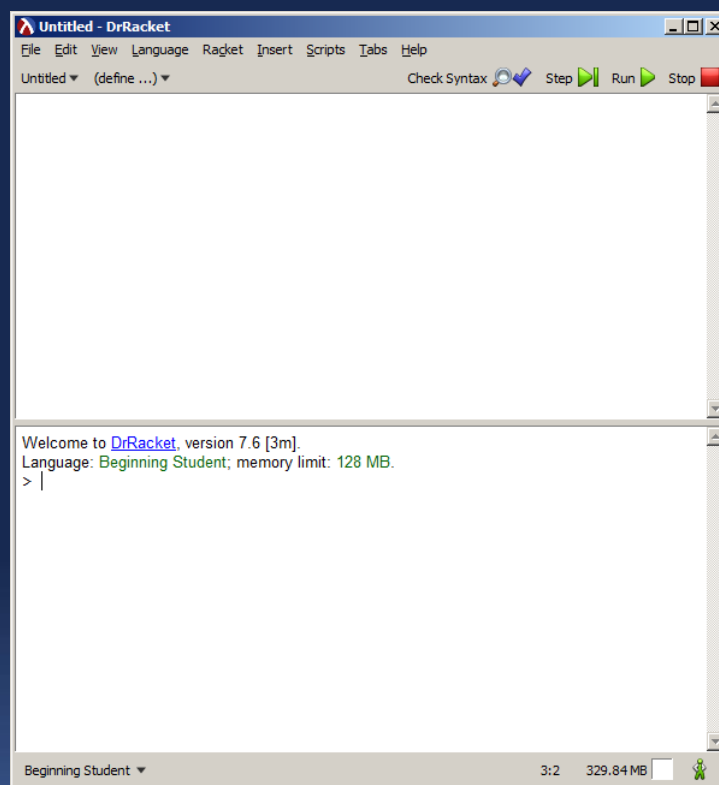
# DrRacket





# DrRacket

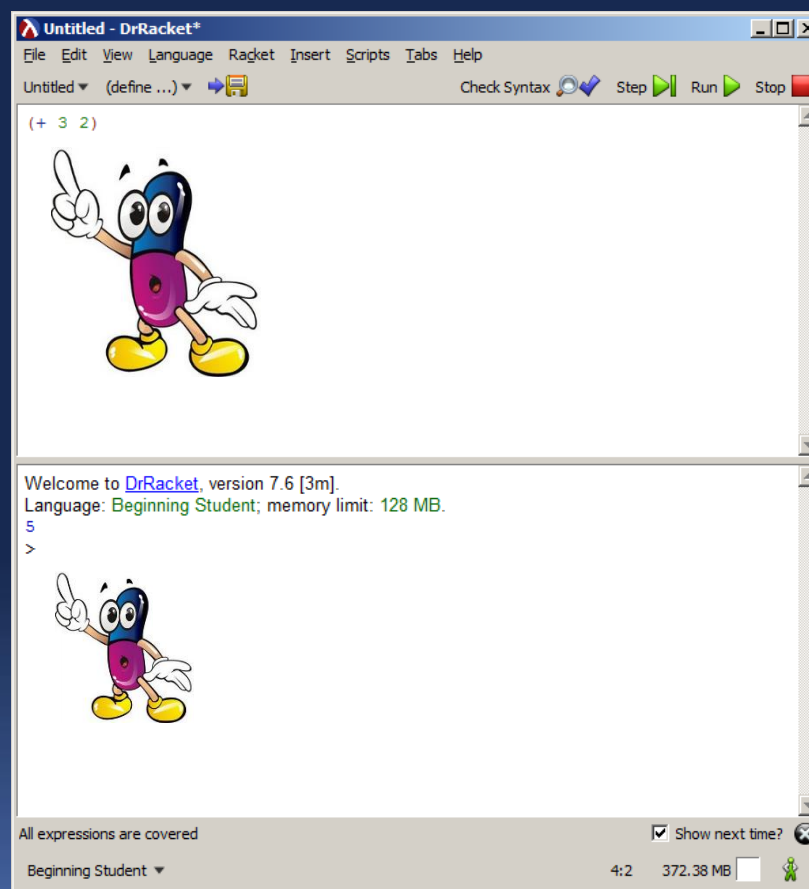
- ✓ Após iniciar o **DrRacket**, selecione "**Choose Language**" a partir do menu "**Language**", o qual abre um diálogo.
- ✓ Definir em **Teaching Languages** a opção "**Beginning Student**";
- ✓ Tecle **OK** após a definição da Teaching Languages.





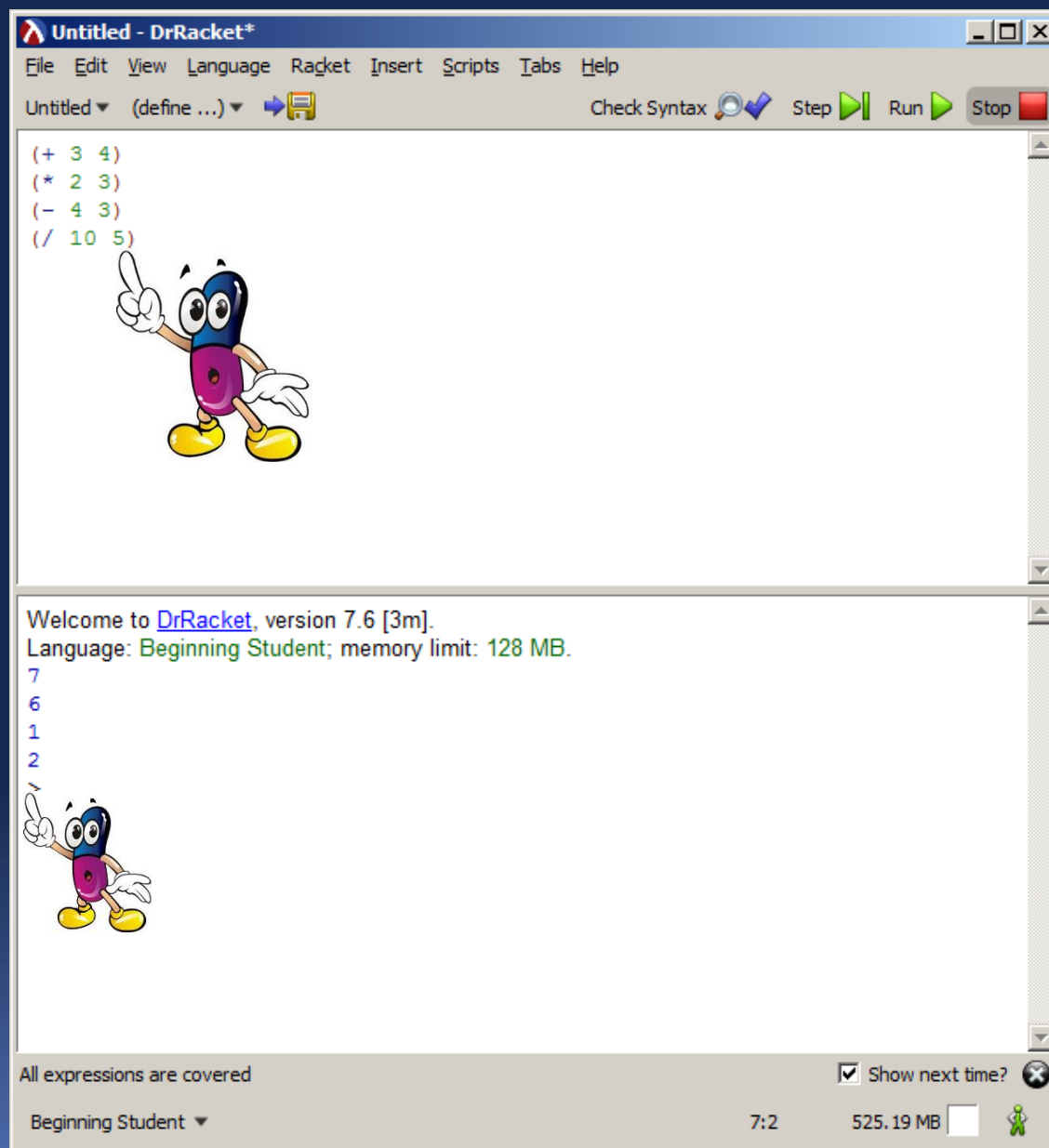
# DrRacket

- ✓ 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.





# Efetuating diversos cálculos



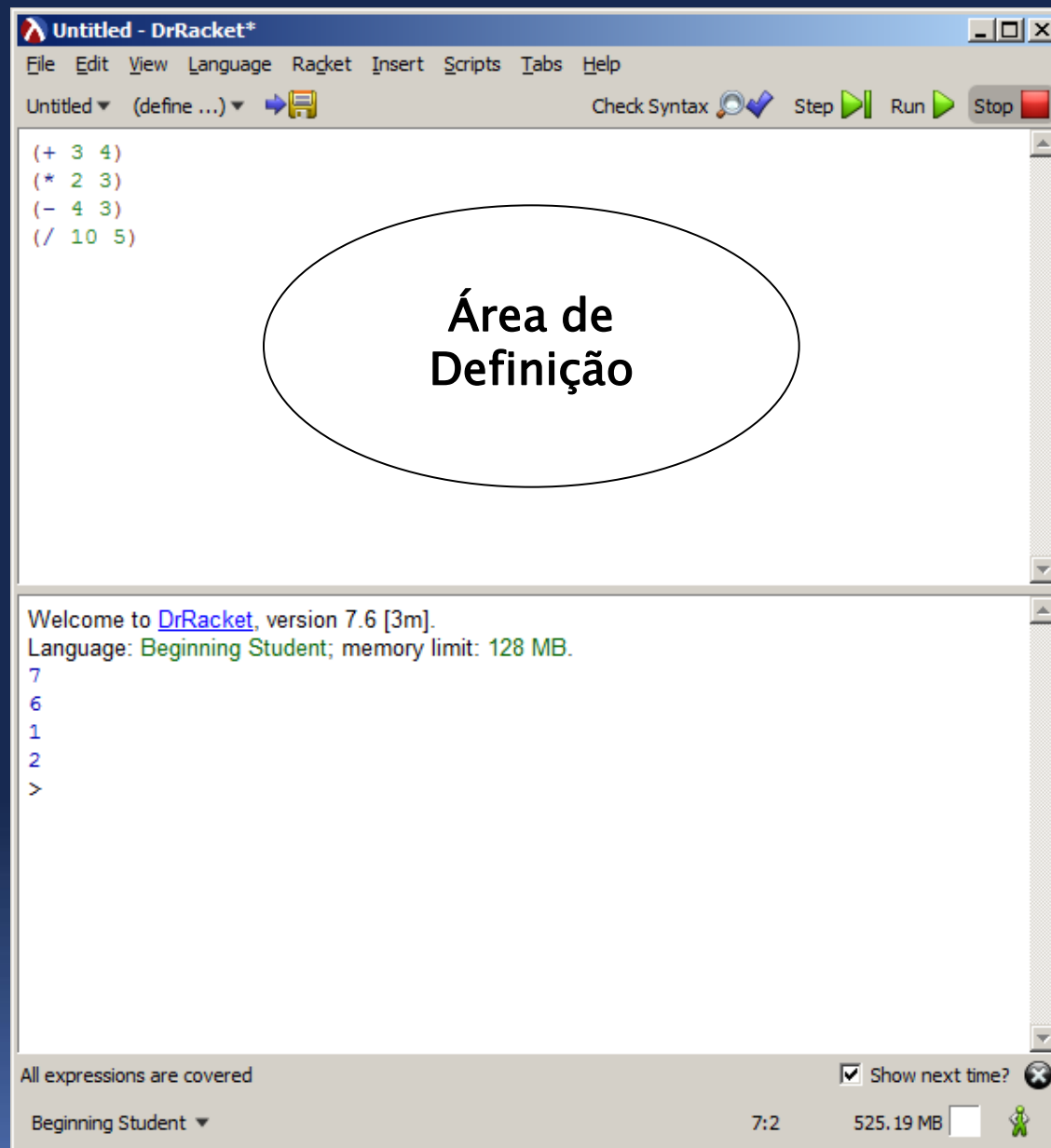




# DrRacket – Área de Definição



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





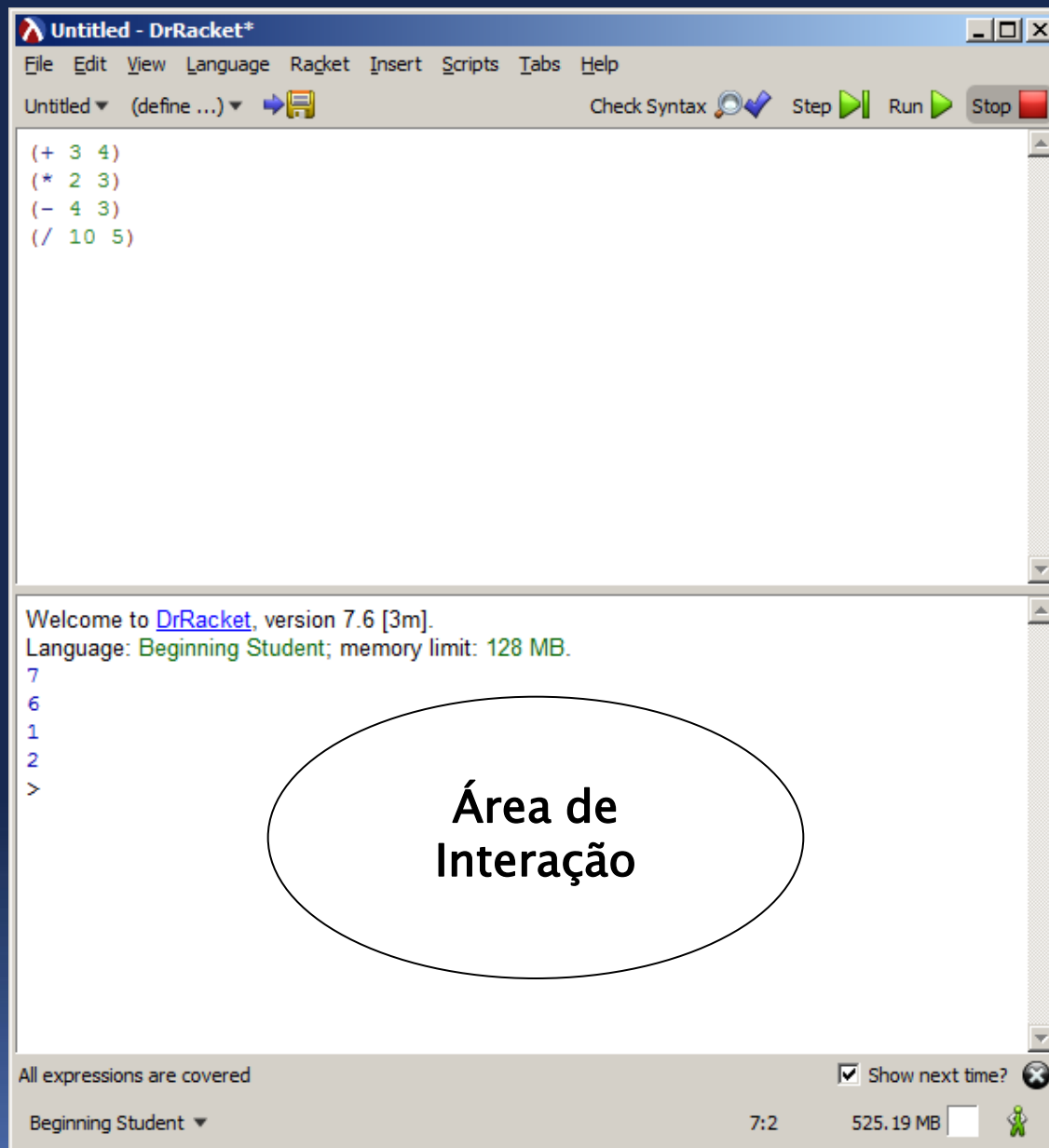


# DrRacket – Área de Interação

- ✓ A parte inferior é chamada de área de **Interação**;
- ✓ Nesta área, pode-se entrar com **expressões** para serem avaliadas de forma interativa (**REPL**)

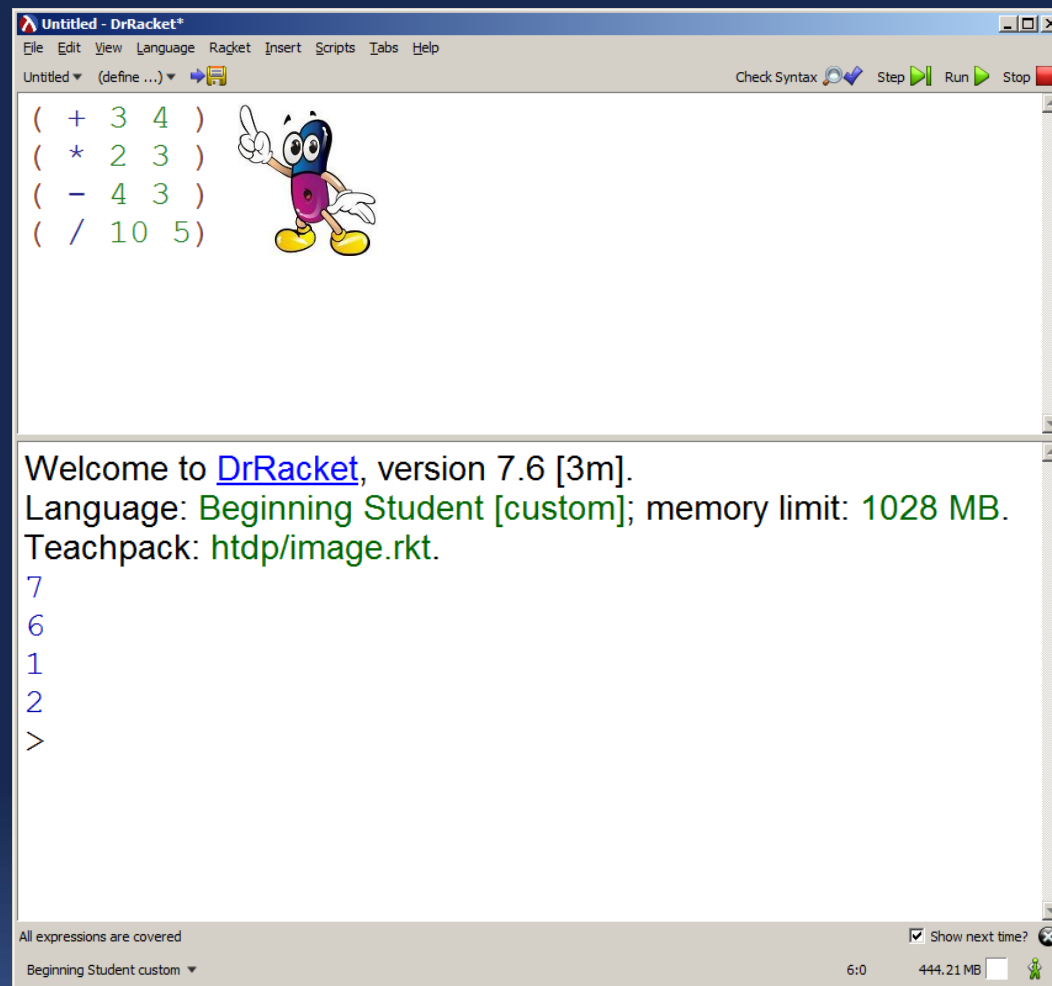


- ✓ **REPL** = Read-Eval-Print Loop





# DrRacket – Botão Save



- ✓ Quando se clica no botão **Save** pela primeira vez, Dr Racket solicita o nome do arquivo para ser salvo;

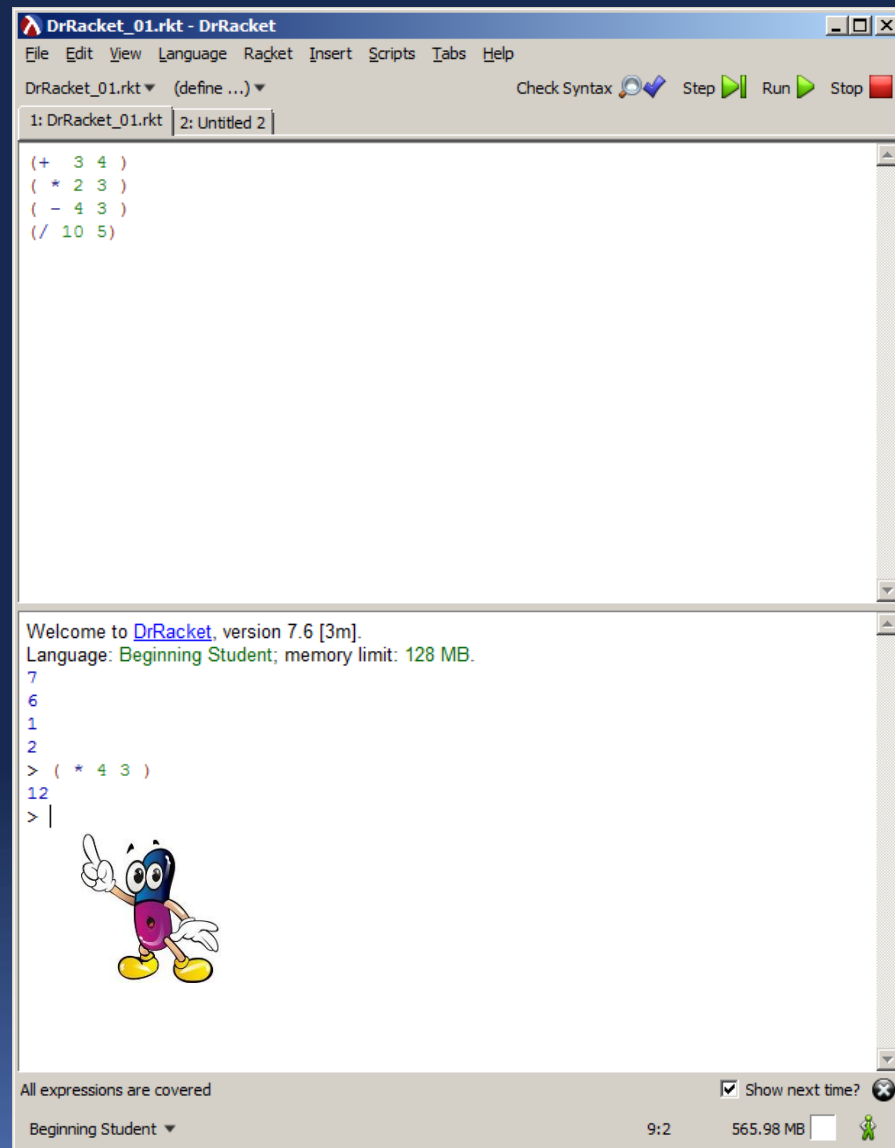




# Programas



- ✓ 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.





# Exemplo



- ✓ Entre com as expressões abaixo, na área de interações, em seguida, tecla **<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
> (sqrt 3)
1.7320508075688772
> (expt 2 3 )
8
> (sin 0)
0
> (cos pi)
-1.0
#i-1.0
> |
```

All expressions are covered ☒ Show next time?

Beginning Student 19:2 321.29 MB



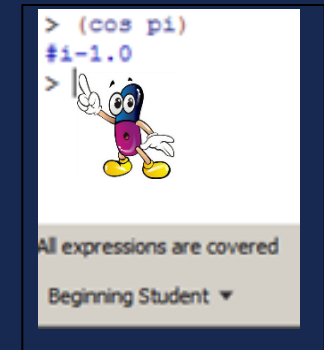


# Observação

- ✓ 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.

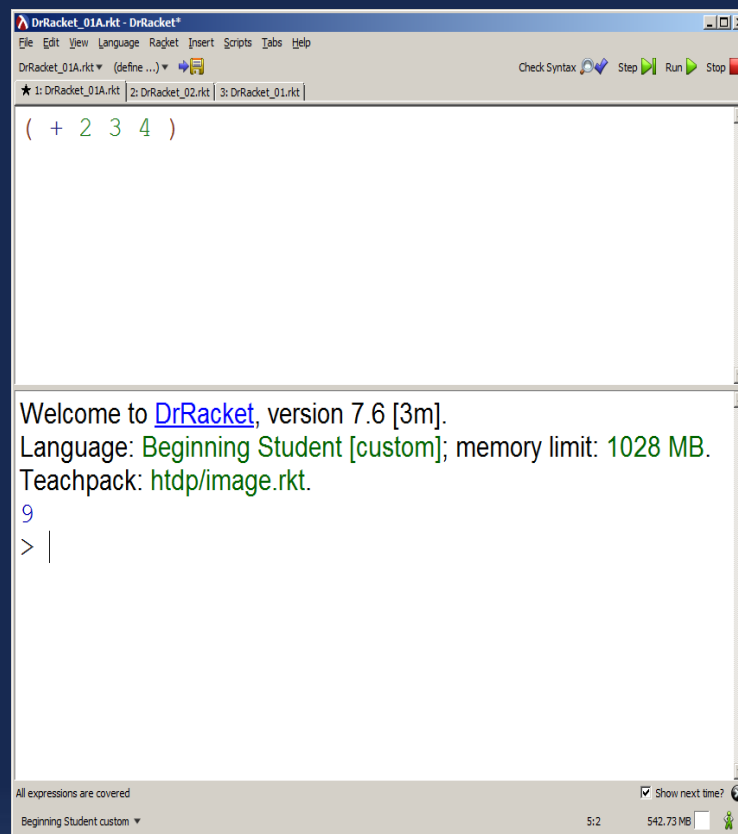


DrRacket é honesto, quando o valor não é exato ele sinaliza um **warning**!!!!





# 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** = **B**egin **S**tudent **L**anguage.







# Encadeando operações (Nested Arithmetic)

The screenshot shows the DrRacket IDE interface. The title bar reads "DrRacket\_01A.rkt - DrRacket". The menu bar includes "File", "Edit", "View", "Language", "Racket", "Insert", "Scripts", "Tools", and "Help". The toolbar contains buttons for "Check Syntax", "Step", "Run", and "Stop". The editor window displays the code `( + 2 ( + 3 4 ) )`. The bottom panel shows the welcome message: "Welcome to [DrRacket](#), version 7.6 [3m]. Language: [Beginning Student](#) [custom]; memory limit: 1028 MB. Teachpack: [http/image.rkt](#)." Below this is a prompt `>`. The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "5:2", and "471.54 MB".

- ✓ 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



DrRacket\_01A.rkt - DrRacket\*

File Edit View Language Racket Insert Scripts Tabs Help

DrRacket\_01A.rkt (define ...) Check Syntax Step Run Stop

★ 1: DrRacket\_01A.rkt 2: DrRacket\_02.rkt 3: DrRacket\_01.rkt

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

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

All expressions are covered ☒ Show next time?

Beginning Student custom 5:2 479.45 MB





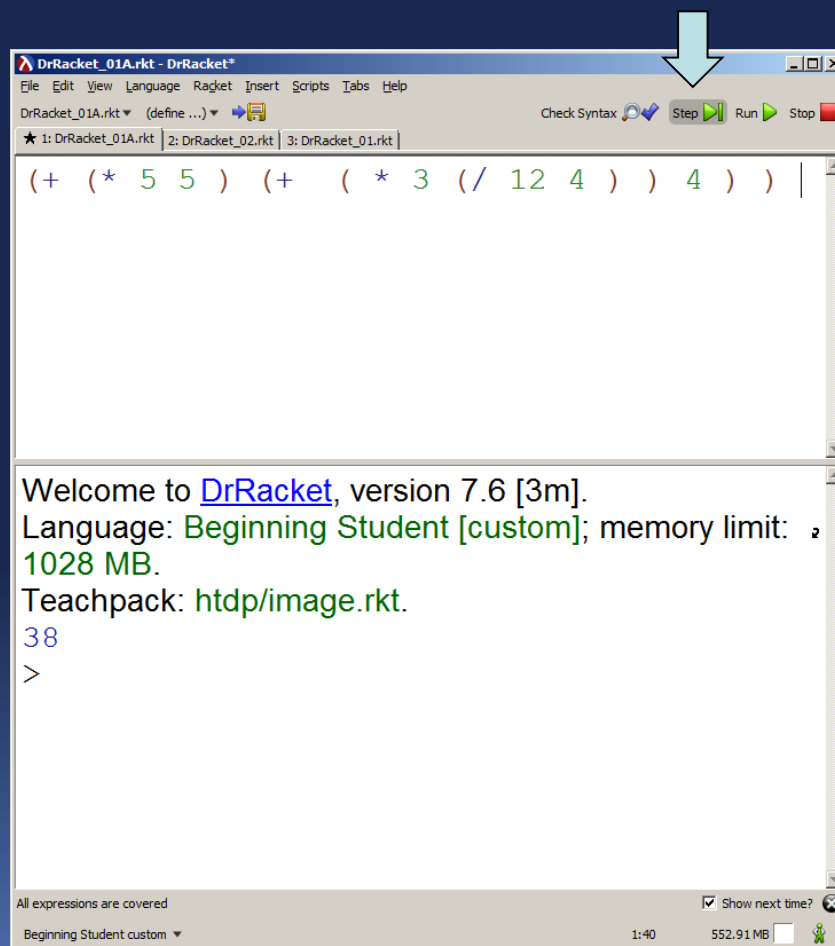
# Observações

- ✓ 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**).





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





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





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





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

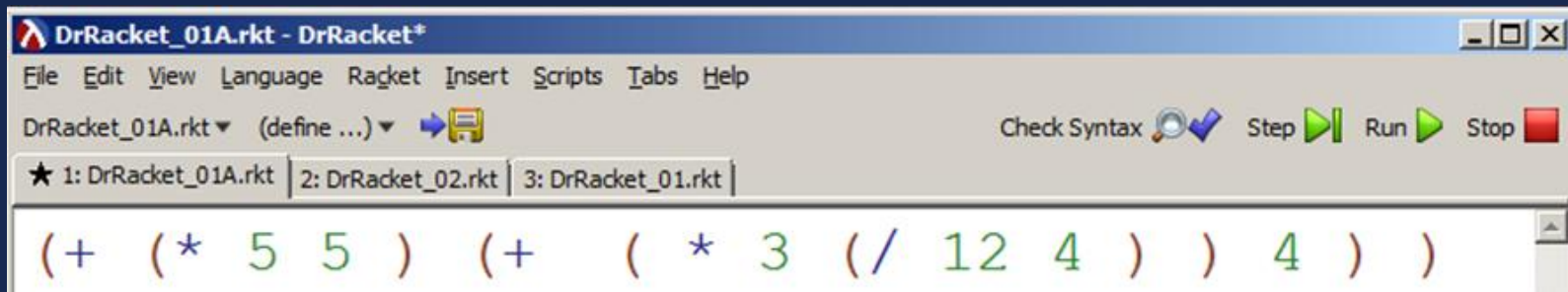




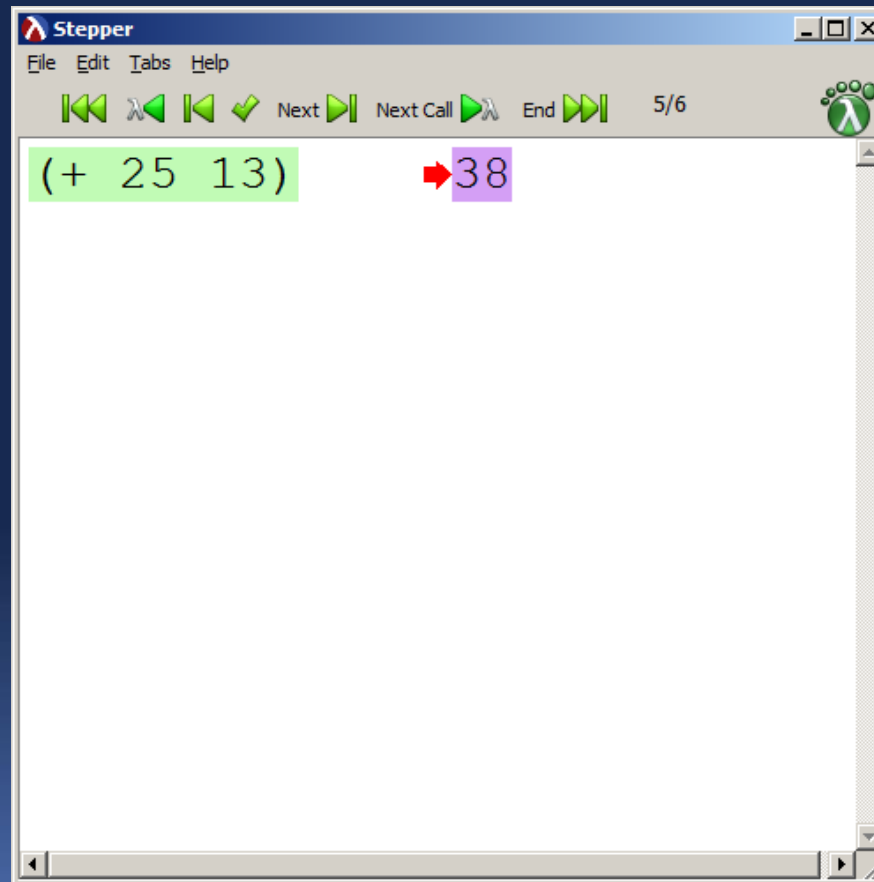


- ✓ A expressão  $(+ 9 4)$  é avaliada na sequência;
- ✓ Clique novamente no botão **Next**;





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





DrRacket\_01A.rkt - DrRacket+

File Edit View Language Racket Insert Scripts Tabs Help

DrRacket\_01A.rkt (define ...) [Save]

Check Syntax [Icon] Step [Icon] Run [Icon] Stop [Icon]

★ 1: DrRacket\_01A.rkt | 2: DrRacket\_02.rkt | 3: DrRacket\_01.rkt

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

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

Stepper

File Edit Tabs Help

Beginning [Icon] Previous Call [Icon] Previous [Icon] Selected [Icon] Next [Icon] Next Call [Icon] End [Icon]

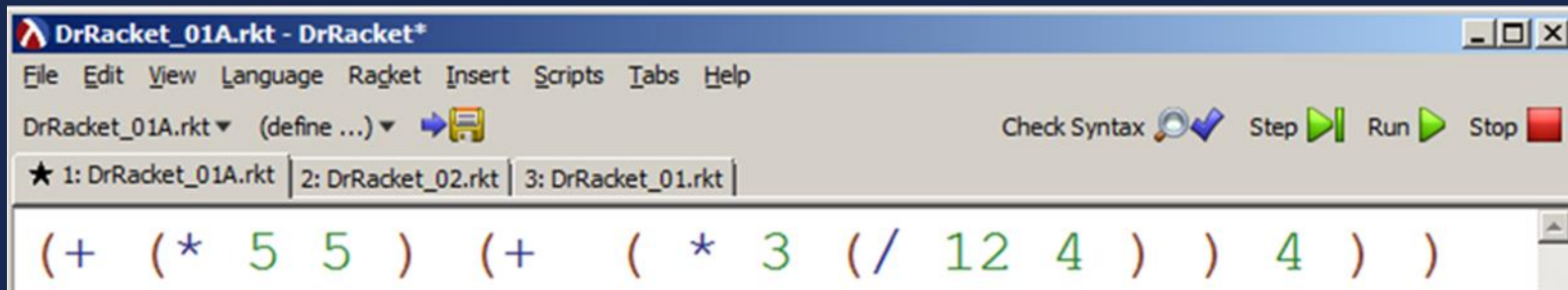
6/6 [Icon]

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.

```
DrRacket_01A.rkt - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
DrRacket_01A.rkt (define ...)
★ 1: DrRacket_01A.rkt 2: DrRacket_02.rkt 3: DrRacket_01.rkt

( + (1) (2) )

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: http/image.rkt.
function call: expected a function
after the open parenthesis, but found a
number
>
```





# XXXXXX Manipulando Strings

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



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





# Manuseando Strings



- ✓ 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: http://image.rkt.
> "Hello DrRacket"
"Hello DrRacket"
> |

All expressions are covered
Beginning Student custom 6:2 325.44 MB Show next time?
```







# 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: http://image.rkt.

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

All expressions are covered
Beginning Student custom 6:2 370.89 MB
```

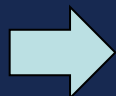




# Retornando tamanho do string

xxxxx

- ✓ A função **string-length** retorna um valor numérico que corresponde ao tamanho do 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-append "Hello" " World...")
(string-length (string-append "Hello" " World..."))

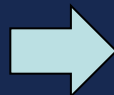
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: htdp/image.rkt.
"Hello World..."
14
>
```





# Convertendo strings para números

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



The screenshot shows the DrRacket IDE window titled "Untitled 3 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar has icons for Check Syntax, Step, Run, and Stop. The code editor contains the Racket expression `( + (string->number "20") (string->number "30") )`. The output window displays the following text: "Welcome to [DrRacket](#), version 7.6 [3m].", "Language: **Beginning Student [custom]**; memory limit: 1028 MB.", "Teachpack: [http://image.rkt](#).", followed by the result `50` and a prompt `>`. The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", and "5:2 467.44 MB".





# Convertendo strings para números

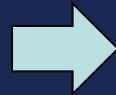
- ✓ E se na função **string->number** for passado um argumento que não corresponde a um número ?





# 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**.



The screenshot shows the DrRacket IDE window titled "Untitled 3 - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows buttons for Check Syntax, Step, Run, and Stop. The code editor contains the expression `(string->number "AA")`. The output area displays the following text: "Welcome to [DrRacket](#), version 7.6 [3m].", "Language: **Beginning Student** [custom];", "memory limit: 1028 MB.", "Teachpack: [htdp/image.rkt](#).", and the result `#false`. The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "5:2", and "602.38 MB".





# Trabalhando com valores Boolean



The screenshot shows the DrRacket IDE window titled "Untitled 5 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu bar, there's a toolbar with "Check Syntax", "Step", "Run", and "Stop" buttons. The main text area displays the following text:

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

At the bottom of the window, there's a status bar with the text "All expressions are covered", a checkbox for "Show next time?" which is checked, and a small icon. Below this, it says "Beginning Student custom", the time "14:2", and the memory usage "360.32 MB".





# Comparando valores numéricos

The screenshot shows the DrRacket IDE window titled "Untitled 6 - 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 main text area contains the following text:

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

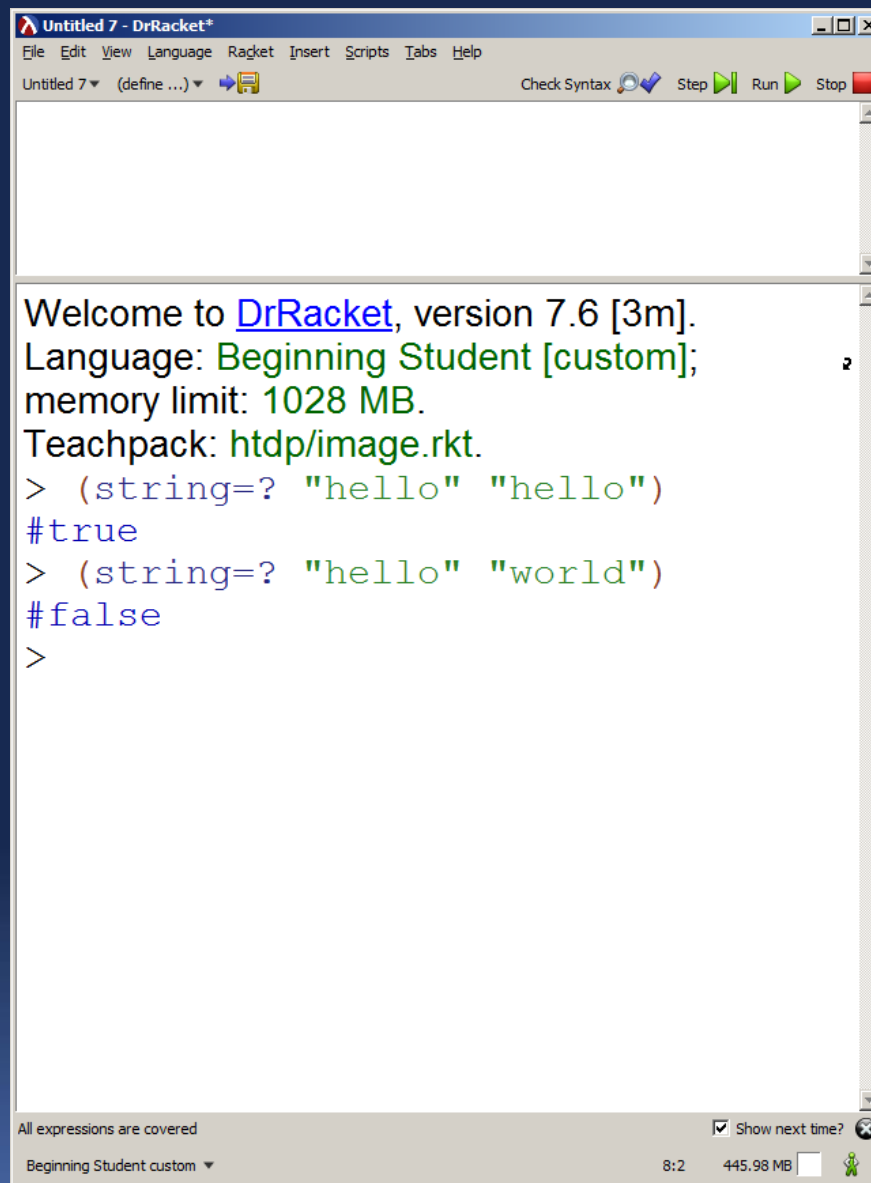
The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "10:2", and "407.61 MB". There is also a checkbox for "Show next time?" and a small icon.







# Comparando strings



The screenshot shows the DrRacket IDE window titled "Untitled 7 - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. The main text area 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  
>
```

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "8:2", and "445.98 MB".





✓ Qual o resultado da avaliação da expressão abaixo:

```
Untitled 9 - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled 9 (define ...) Check Syntax Step Run Stop
(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 window titled "Untitled 9 - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains icons for 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 expression is evaluated, and the result is displayed in the bottom panel:

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

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





## Definindo Funções

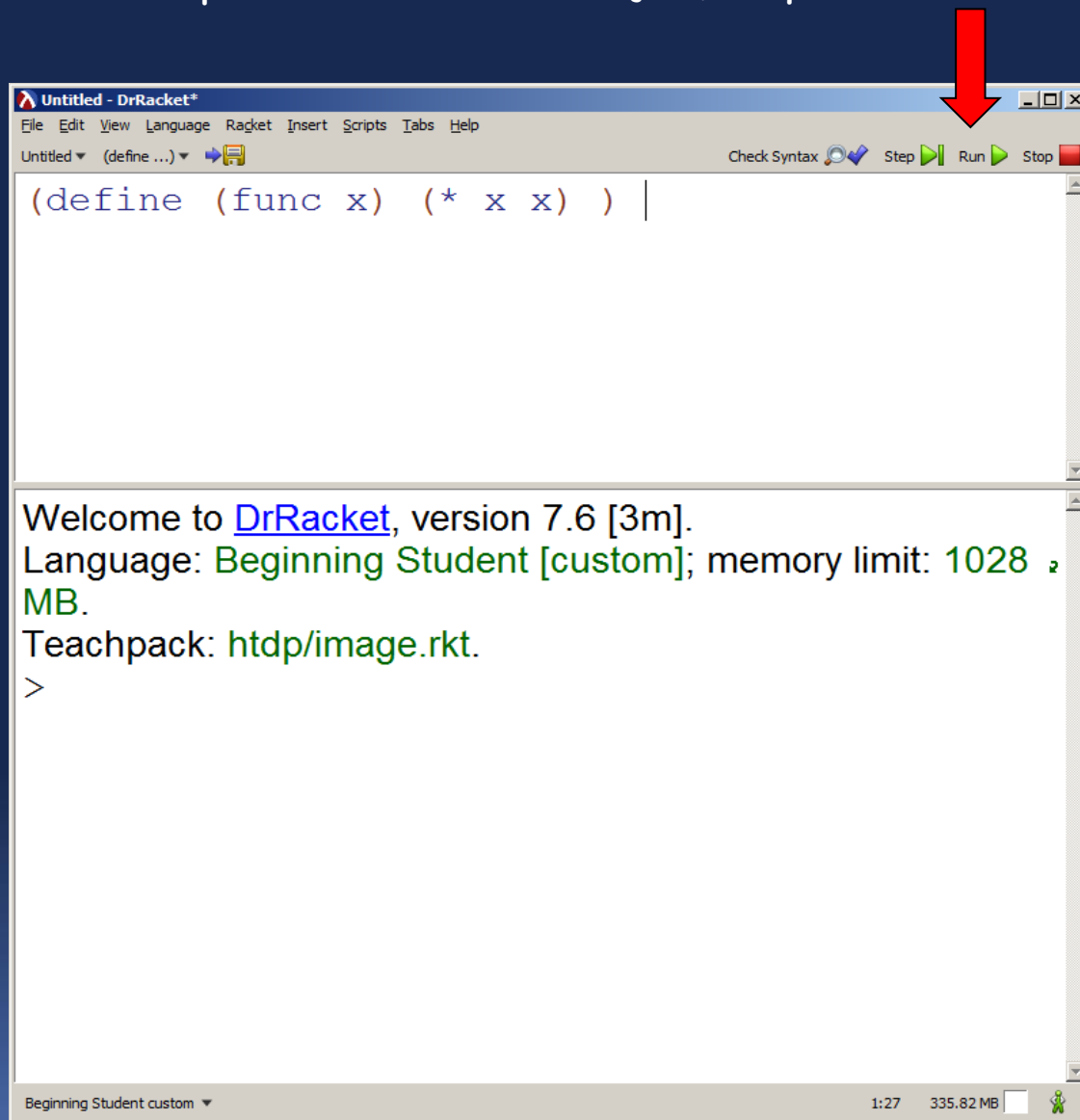
- ✓ Funções são definidas por outra função chamada **define**;
- ✓ Por exemplo: `(define (func x) (* x x))`
- ✓ **define** significa: “**considere func uma função**”, a qual, como uma expressão, computará um valor;
- ✓ O valor de uma função, depende, dos valores entrados (input), os quais são expressos 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.





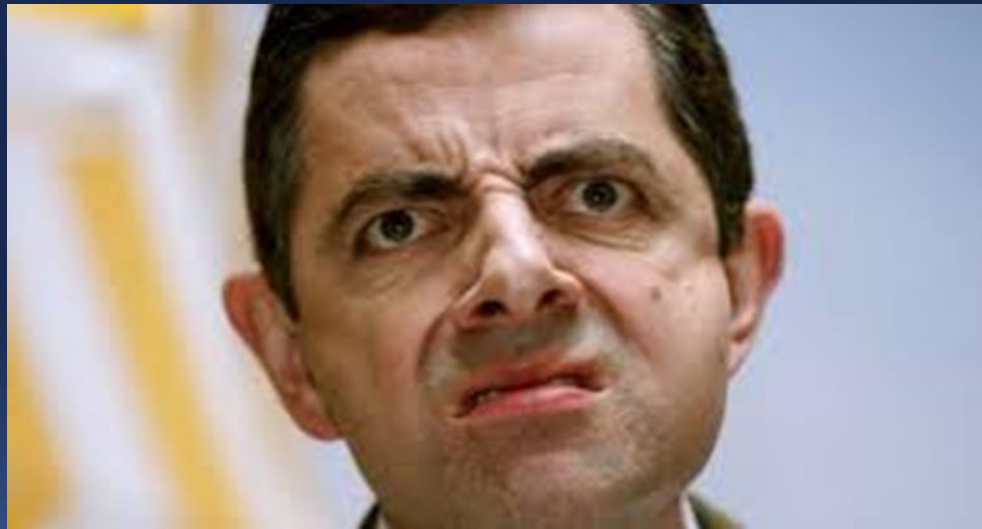
## Definindo Funções

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





Vê ... 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.

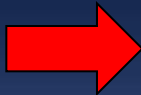
The screenshot shows the DrRacket IDE window titled "Untitled - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar has buttons for Check Syntax, Step, Run, and Stop. The editor area contains the Racket code: `(define (func x) (* x x) )`. The bottom pane displays the welcome message: "Welcome to [DrRacket](#), version 7.6 [3m]. Language: [Beginning Student](#) [custom]; memory limit: 1028 MB. Teachpack: [http/image.rkt](#)." followed by a prompt `>`. The status bar at the bottom shows "Beginning Student custom", "4:2", and "405.72 MB".





## 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.



```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [Check Syntax] [Step] [Run] [Stop]

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

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

Beginning Student custom 6:2 404.67 MB
```

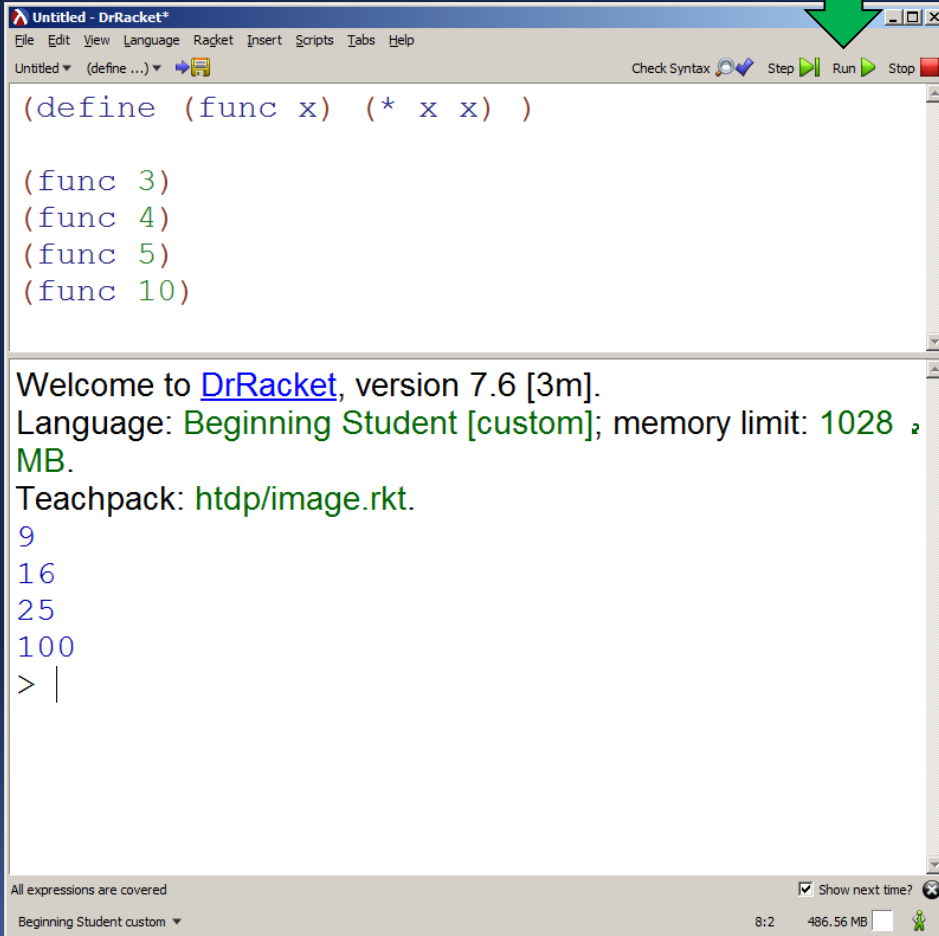






## Aplicação da Função

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



The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains icons for Check Syntax, Step, Run, and Stop. The main text area contains the following Racket code:

```
(define (func x) (* x x) )  
  
(func 3)  
(func 4)  
(func 5)  
(func 10)
```

Below the code area, the output window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/image.rkt.  
9  
16  
25  
100  
> |
```

Two green arrows point to the Run button in the toolbar and the output window. A third green arrow points to the code area.





- ✓ 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 o ou 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:

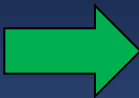
```
(FunctionName ArgumentExpression)
```

- ✓ 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 ...) 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
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 window titled "Untitled 2 - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. The main text area contains the following Racket code:

```
(define (func x)

  (cond

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

Below the code area, the output window displays the following text:

```
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 shows "Beginning Student custom", "6:2", and "462.51 MB".





## Exemplo 3

The screenshot shows the DrRacket IDE with a Racket program defined as a function `x`. The program uses a `cond` expression to evaluate several conditions and return corresponding strings. The output shows that all conditions are false, resulting in the message *cond: all question results were false*.

```
(define (func x)

  (cond

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

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





# Definindo constantes

- ✓ Vimos nos exemplos anteriores que funções são definidas pela função **define**;
- ✓ Mas, **define** também pode ser usado para se definir **constantes**;
- ✓ 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 window titled "Untitled 2 - 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 contains the following Racket code:

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

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

The bottom panel shows the Welcome message and the interactive prompt:

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

The status bar at the bottom indicates "Beginning Student custom", "6:2", and "860.61 MB".





# Definindo constantes

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

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

(define (func Y )

  (+ X Y) )
```

The bottom panel shows the welcome message and the results of an evaluation:

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

The status bar at the bottom indicates "Beginning Student custom", "6:2", and "967.18 MB".





# Definindo constantes

The screenshot shows the DrRacket IDE interface. The main editor window 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 ) )
```

Below the editor, the Welcome message and the execution output are visible:

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

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

The status bar at the bottom indicates the language is 'Beginning Student custom', the current line is 6:2, and the memory usage is 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 DrRacket:

```
(define x 3)  
(define y 4)
```

- ✓ Agora, imagine que  $x$  e  $y$  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 window titled "Untitled 2 - DrRacket". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar has icons for Check Syntax, Step, Run, and Stop. The main text area 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 panel shows the welcome message and the execution of the `(distancia a b)` function:

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

The status bar at the bottom indicates "Beginning Student custom", "6:2", and "1724.76 MB".





# 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 *i*String localizado na *i*ésima 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 window, titled 'Untitled 2 - DrRacket\*', 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 window shows the REPL output:

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

The status bar at the bottom indicates 'Beginning Student custom', '6:2', and '506.91 MB'.





# Aritmética de Strings – Exemplo

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

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 shows "Beginning Student custom", "6:2", and "347.98 MB".





## Exercício 2 – Aritmética de Strings

- ✓ Adicione no ambiente de programação DrRacket:

```
(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 window titled "Untitled 2 - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. 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 output window displays the following text:

```
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 shows "Beginning Student custom", "6:2", and "427.29 MB".





## Exercício 3 – Aritmética de Strings

- ✓ Adicione no ambiente de programação DrRacket:

```
(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 DrRacket:

```
(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** **consume** um string **e** produz um número;
  - ❑ **string-ith** **consume** um string **s** junto com um número **i** e extrai o 1String localizado na iésima posição (contando de 0)
  - ❑ **number->string** **consume** um número **e** produz um string;
  - ❑ **string-append** **concatena** uma sequência de strings.





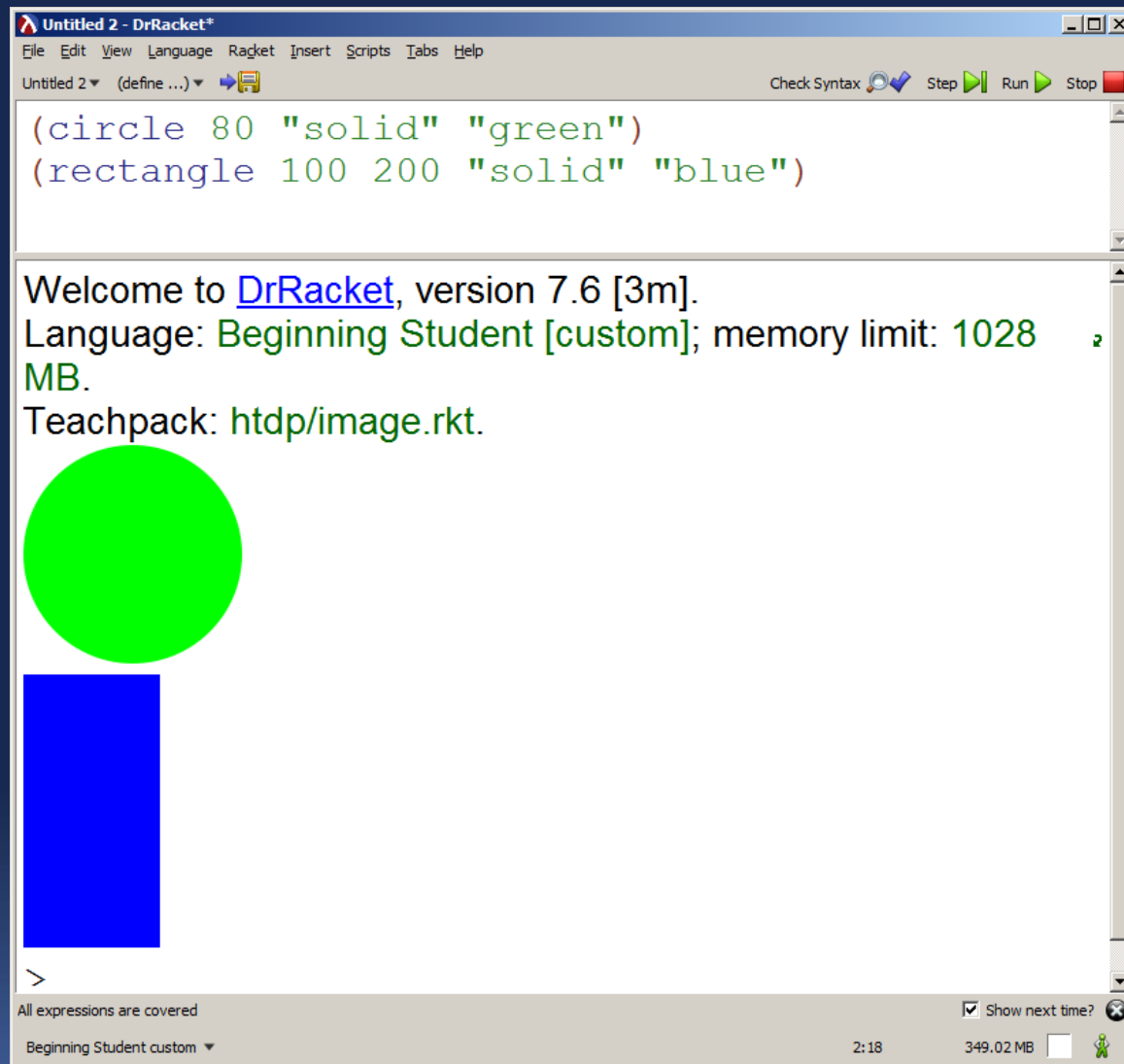
# Aritmética de Imagens

- ✓ 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





# 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 IDE window titled "Untitled 2 - 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 contains the following Racket code:

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

The bottom panel shows the REPL output:

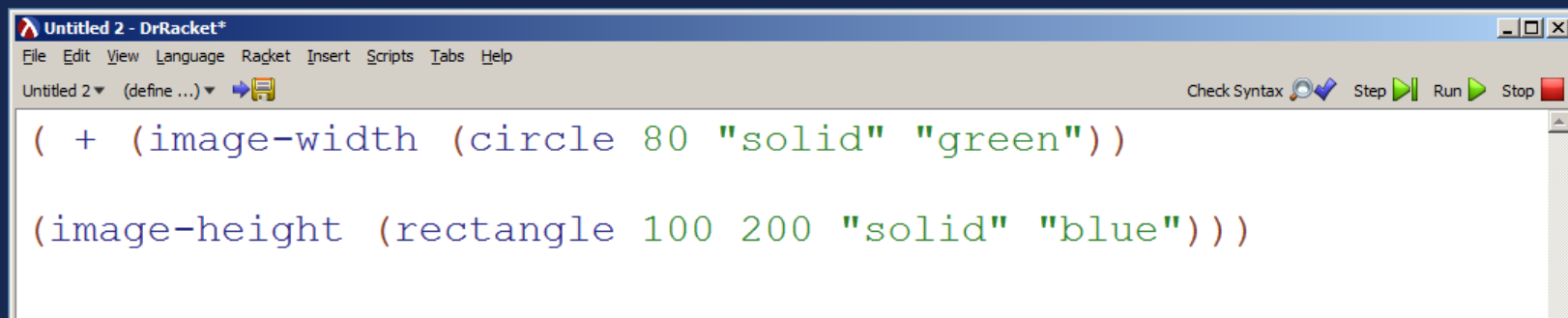
```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
160  
200  
> |
```

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "6:2", and "464.23 MB".





Qual o valor dessa expressão ?



```
Untitled 2 - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled 2 (define ...) Check Syntax Step Run Stop

( + (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 menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains icons for Check Syntax, Step, Run, and Stop. The main text area contains the following Racket expression:

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

Below the code area, the Welcome message is displayed:

Welcome to [DrRacket](#), version 7.6 [3m].  
Language: **Beginning Student [custom]**; memory limit: 1028 MB.  
Teachpack: [http/image.rkt](http://image.rkt).  
360  
>

The bottom status bar shows "All expressions are covered", "Beginning Student custom", "5:2", and "522.55 MB".





# 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.

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

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

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

Beginning Student custom 8:2 554.35 MB





# Comparando valores numéricos

The screenshot shows the DrRacket IDE window titled "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 editor contains the following Racket code:

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

The bottom panel displays the welcome message and the results of two test cases:

```
Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: http/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 shows "Beginning Student custom", "8:2", and "384.85 MB".





# Comparando valores numéricos

The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. Below the menu bar, there's a toolbar with buttons for 'Check Syntax', 'Step', 'Run', and 'Stop'. The main editor window 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") )
```

Below the code editor, the 'Welcome to DrRacket, version 7.6 [3m].' message is displayed, followed by 'Language: Beginning Student [custom]; memory limit: 1028 MB.' and 'Teachpack: [http/image.rkt](http://image.rkt).' The console shows the results of three function calls:

```
> (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', the time '5:10', and the memory usage '537.98 MB'.





# Comparando strings

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

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

Below the code editor, the Welcome message reads: "Welcome to [DrRacket](#), version 7.6 [3m]. Language: **Beginning Student [custom]**; memory limit: 1028 MB. Teachpack: [http://image.rkt](#)." The interaction area shows the following commands and results:

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

The status bar at the bottom indicates the language is "Beginning Student custom", the mode is "8:2", and the memory usage is "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.

```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [Icons] Check Syntax [Icons] Step [Icons] Run [Icons] Stop [Icons]

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
>

All expressions are covered [X] Show next time?
Beginning Student custom 1:0 480.33 MB [Icons]
```







# Predicados



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

```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) Check Syntax Step Run Stop

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

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: http://image.rkt.
> (tamanho_string "Hello")
5
> (tamanho_string 34543)
"Argumento entrado inválido!"
>

Beginning Student custom 8:2 749.07 MB
```





# Predicados

✓ Predicado integer?

```
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: http/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 14:2 1061.57 MB
```





# Predicados

✓ Predicado **rational?**

```
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.
> pi
#i3.141592653589793
> (rational? pi)
#true
> (rational? 2)
#true
> (rational? 1/3)
#true
>

All expressions are covered
Beginning Student custom 12:2 1035.91 MB
```





# Predicados

✓ Predicado real?

```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [icons] Check Syntax [icons] Step [icon] Run [icon] Stop [icon]

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
> (real? (/ 1 3))
#true
> |

All expressions are covered
Beginning Student custom 14:2 1096.46 MB [checkbox] [icon]
```





# Predicados

✓ Predicado complex?

DrRacket, version 7.6 [3m].', 'Language: Beginning Student [custom]; memory limit: 1028 MB.', 'Teachpack: [http/image.rkt](#).', and two REPL interactions: '> (complex? 1+2i)' followed by '#true', and '> (complex? 5-9i)' followed by '#true'. The bottom status bar shows 'All expressions are covered', 'Beginning Student custom', '8:2', '1395.77 MB', and a 'Show next time?' checkbox which is checked."/>

```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [Save] Check Syntax Step Run Stop

Welcome to DrRacket, version 7.6 [3m].
Language: Beginning Student [custom]; memory limit: 1028 MB.
Teachpack: http/image.rkt.
> (complex? 1+2i)
#true
> (complex? 5-9i)
#true
>

All expressions are covered
Beginning Student custom 8:2 1395.77 MB [Show next time?]
```





# 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 window titled "Untitled - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. The editor contains the code `(define (f x) 1)`. The interactions window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/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", the time "14:2", and the memory usage "600.59 MB".





# Exemplos de Definição de Funções

The screenshot shows the DrRacket IDE window titled "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 editor area contains the following Racket code:

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

The output area displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/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", "10:2", and "420.79 MB".





# Exemplos de Definição de Funções

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

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

Below the editor, the console displays the following output:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/image.rkt.  
> (g 1 2 )  
11  
> (g 999 555)  
11  
> (g 44 (+ 4 5) )  
11  
>
```

The status bar at the bottom shows "Beginning Student custom", "10:2", and "384.72 MB".





# Exemplos de Definição de Funções

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

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

The output window displays the following text:

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

The interactive prompt shows the following sequence of inputs and outputs:

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

The status bar at the bottom shows "Beginning Student custom", "10:2", and "521.80 MB".





# 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 window titled "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 contains the following Racket code:

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

The output window displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (func 3)  
"Hello World...."  
> (func 10)  
"Hello World...."  
> (func "USCS")  
"Hello World...."  
>
```

The status bar at the bottom shows "Beginning Student custom", "10:2", and "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 window titled "Untitled - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. The editor area contains the following code:

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

Below the editor, the output area displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: http/image.rkt.  
> (func "Hello World...")  
999  
> (func (or true false) )  
999  
> (func 1234)  
999  
> (func (+ 1 2 ) )  
999  
> |
```

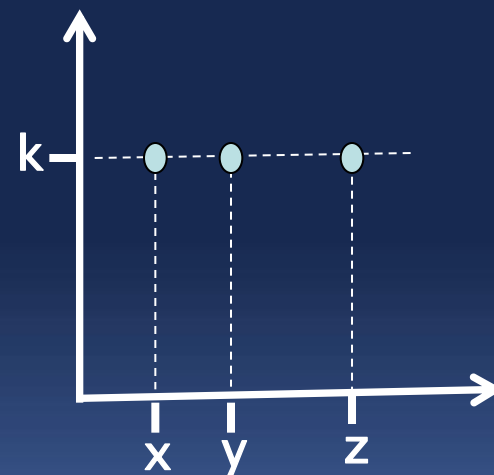
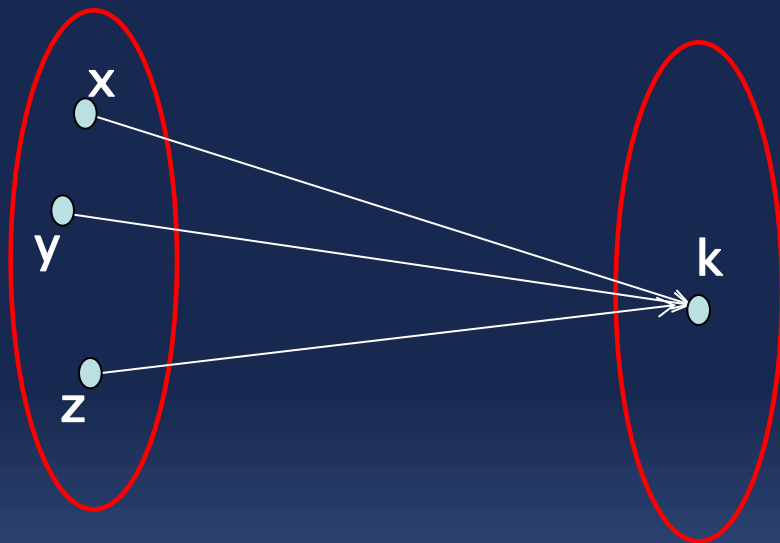
The status bar at the bottom shows "Beginning Student custom", "12:2", and "738.04 MB".





# 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**;

```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) Check Syntax Step Run Stop

(define x 99)

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
>
```

All expressions are covered ☒ Show next time?

Beginning Student custom 8:2 980.11 MB





# 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

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

Welcome to [DrRacket](#), version 7.6 [3m].  
Language: **Beginning Student [custom]**; memory limit: 1028 MB.  
Teachpack: [http/image.rkt](http://image.rkt).  
> (func "Hello World...")  
1  
>





# Aplicações nested

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

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

The output area displays the following text:

```
Welcome to DrRacket, version 7.6 [3m].  
Language: Beginning Student [custom]; memory limit: 1028 MB.  
Teachpack: htdp/image.rkt.  
> (+ (func (func 2) ) 100)  
104  
> |
```

The status bar at the bottom shows "Beginning Student custom", "6:2", and "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 window titled "Untitled - DrRacket\*". The menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar shows "Check Syntax", "Step", "Run", and "Stop" buttons. The editor contains the following Racket code:

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

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

The output window displays the following text:

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

The status bar at the bottom shows "Beginning Student custom", "4:2", and "514.23 MB".





# Exercício 1

Resposta:

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

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

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

Below the editor, the Welcome message is displayed:

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

The prompt shows the number 5 and a greater-than sign (>).

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "5:2", and "605.98 MB".





# Exercício 2

Qual o resultado ?

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

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

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

The output window displays the following text:

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

The status bar at the bottom shows "Beginning Student custom", "4:2", and "343.15 MB".







# Exercício 2

Resposta:

The screenshot shows the DrRacket IDE interface. The top menu bar includes File, Edit, View, Language, Racket, Insert, Scripts, Tabs, and Help. The toolbar contains icons for opening a file, saving, and running. The main editor window displays the following Racket code:

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

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

Below the editor, the output window shows the following text:

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

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", and "5:2 417.83 MB".



# Exercício 3

Qual o resultado?

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

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

The output window displays the following text:

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

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





# Exercício 3

Qual o resultado?

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

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

The output window displays the following text:

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

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





# Exercício 3

Qual o resultado?

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

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

The output window displays the following text:

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

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "5:2", and "326.09 MB".





# Exercício 4

Qual o resultado?

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

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

The status bar at the bottom shows "Beginning Student custom", "4:2", and "399.08 MB".





# Exercício 4

Resposta:

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

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

(string-insert "Hello-world" 5)
```

The output window shows the following text:

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

The status bar at the bottom indicates "All expressions are covered", "Beginning Student custom", "5:2", "450.03 MB", and a "Show next time?" checkbox.





# 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



```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [Icons]
Check Syntax [Icons] Step [Icons] Run [Icons] Stop [Icons]

(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: http/image.rkt.
> (carta "Aparecido" "Freitas" "assinatura")
"Caro Aparecido,\n\nOlá, tudo bem?\n Aparecido de Freitas\n\nUm grande abraço! \n\nassinatura\n\n"
>

Beginning Student custom 12:13 511.68 MB [Icons]
```







# Composição de Funções – Exemplo



```
Untitled - DrRacket*
File Edit View Language Racket Insert Scripts Tabs Help
Untitled (define ...) [Icons]
Check Syntax [Icons] Step [Icons] Run [Icons] Stop [Icons]

(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: http/image.rkt.
> (carta "Aparecido" "Freitas" "assinatura")
"Caro Aparecido,\n\nOlá, tudo bem?\n Aparecido de Freitas\n\nUm grande abraço! \n\nassinatura\n\n"
>

Beginning Student custom 12:13 511.68 MB [Icons]
```





# 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! "
    assinatura) )

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





# Adicionando para console

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

The screenshot shows the DrRacket IDE interface. The top pane contains Racket code defining functions for generating a letter. The bottom pane shows the output of the code, which is a letter from 'Aparecido' to 'Freitas'.

```
(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 "
    "Aparecido de Freitas\n "
    "Um grande abraço!\n "
    assinatura
    "\n\n"
    "'stdout"
    "\n"
    ">"))
```

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

> (write-file 'stdout (carta "Aparecido" "Freitas" "assinatura") )  
Caro Aparecido,  
  
Olá, tudo bem?  
Aparecido de Freitas  
  
Um grande abraço!  
  
assinatura  
  
'stdout  
>





# 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!!!)

