



1



2

Listas

- Tipo especial que armazena qualquer tipo de dado (heterogêneos)
 - Valores atômicos
 - Vetores
 - Matrizes
 - Listas
 - Funções
 - Outros objetos

Prof. Dr. Razer A N R Montão

SEPT/UFPR

3

3

Listas: Criação

- Cria-se com a função `list()`
- Exemplo

```
> x <- list(1:5, "Z", TRUE, c("a", "b"))
> x
[[1]]
[1] 1 2 3 4 5

[[2]]
[1] "Z"

[[3]]
[1] TRUE

[[4]]
[1] "a" "b"
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

4

4

Listas: Criação

- Pode-se dar um nome para os elementos da lista, com `names()`

```
> x <- list(1:5, "Z", TRUE, c("a", "b"))
> names(x) <- c("numeros", "letra", "lógico", "vetor")
> x

$numeros
[1] 1 2 3 4 5

$letra
[1] "Z"

$lógico
[1] TRUE

$vetor
[1] "a" "b"

> x$lógico
[1] TRUE
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

5

5

Listas: Criação.

- Pode-se criar listas já com os nomes nos elementos
- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432",
endereco="Rua X")
> razer

$nome
[1] "Razer"

$cpf
[1] "1010"

$celular
[1] "9876-5432"

$endereco
[1] "Rua X"
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

6

6

Listas: Manipulação

- Para acessar os elementos da lista, usa-se o nome (\$) ou índice ([[índice]])

- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432", endereco="Rua X")
```

```
> razer$cpf
```

```
[1] "1010"
```

```
> razer[[3]]
```

```
[1] "9876-5432"
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

7

7

Listas: Manipulação

- Ao acessar com somente um colchete ([x])
 - Fatia a lista e retorna uma lista
 - Flexível, pois pode-se usar qualquer operador de lista

- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432", endereco="Rua X")
```

```
> razer[1]
```

```
$nome
```

```
[1] "Razer"
```

```
> razer["cpf"]
```

```
$cpf
```

```
[1] "1010"
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

8

8

Listas: Manipulação

- Pode-se fatiar a lista em mais de um elemento

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432", endereco="Rua X")
```

```
> razer[c("nome", "celular")]
```

```
$nome
```

```
[1] "Razer"
```

```
$celular
```

```
[1] "9876-5432"
```

Listas: Manipulação

- Diferença entre `[[i]]` e `[i]`

- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432", endereco="Rua X")
```

```
> class(razer[1])
```

```
[1] "list"
```

```
> class(razer[[1]])
```

```
[1] "character"
```

Listas: Manipulação

- Acessar elementos da lista

- Seja a lista

```
> x <- list(1:5, "Z", TRUE, c("a", "b"))
> names(x) <- c("numeros", "letra", "lógico", "vetor")
> x
$numeros
[1] 1 2 3 4 5

$letra
[1] "Z"

$lógico
[1] TRUE

$vetor
[1] "a" "b"
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

11

11

Listas: Manipulação

- Acessar elementos da lista

```
> x[1]
$numeros
[1] 1 2 3 4 5

> x[[1]]
[1] 1 2 3 4 5

> x[1]$numeros
[1] 1 2 3 4 5

> x[[1]]$numeros
Error in x[[1]]$numeros : $ operator is invalid for atomic
vectors

> x[1][1]
$numeros
[1] 1 2 3 4 5

> x[[1]][1]
[1] 1
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

12

12

Listas: Manipulação

- Usa-se atribuição indexada para **alterar um elemento da lista** ([] ou [[]])

- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432",
endereco="Rua X")
> razer[[1]] <- "Razer Anthom"
> razer
$nome
[1] "Razer Anthom"

$cpf
[1] "1010"

$celular
[1] "9876-5432"

$endereco
[1] "Rua X"
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

13

13

Listas: Manipulação

- Para **adicionar um elemento** basta atribuir com um rótulo inexistente

- Exemplo

```
> razer <- list(nome="Razer", cpf="1010", celular="9876-5432",
endereco="Rua X")
> razer[["numero"]] <- 500
> razer
$nome
[1] "Razer"

$cpf
[1] "1010"

$celular
[1] "9876-5432"

$endereco
[1] "Rua X"

$numero
[1] 500
```

Prof. Dr. Razer A N R Montão

SEPT/UFR

14

14

Listas: Manipulação

- Pode-se usar o operador `$` para acessar um elemento
 - Usa-se o nome do elemento e não o índice
- Exemplo

```
> razer$numero <- 200
```

- Se o nome do elemento possui um caractere especial, usa-se ```, `"` ou `'`
- Exemplo

```
> razer$`numero` <- 200
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

15

15

Listas: Manipulação

- Para **remover elementos da lista**, basta setar para `NULL`
 - Ou com operador `$` ou com o operador `[x]` ou `[[x]]`
- Exemplo

```
> razer[["cpf"]] <- NULL
```

```
> razer
```

```
$nome
```

```
[1] "Razer Anthom"
```

```
$celular
```

```
[1] "9876-5432"
```

```
$endereco
```

```
[1] "Rua X"
```

```
$numero
```

```
[1] 500
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

16

16

Listas: Manipulação

- Após a remoção, os elementos são rearranjados
- Exemplo

```
> razer$cpf <- NULL
> razer[[1]]
[1] "Razer Anthom"
> razer[[2]]
[1] "9876-5432"
> razer[[3]]
[1] "Rua X"
> razer[[4]]
[1] 500
> razer[[5]]
Error in razer[[5]] : índice fora de limites
```

Prof. Dr. Razer A N R Montaña

SEPT/UFR

17

17

Listas: Manipulação.

- Pode-ser concatenar listas colocando-as juntas com `c()`
- Exemplo

```
> x <- list(1:5, "Z", TRUE, c("a", "b"))
> names(x) <- c("numeros", "letra", "lógico", "vetor")
> y <- list(nome="Razer", sobrenome="Montaña")
> minha_lista <- c(x, y)
> minha_lista
$numeros
[1] 1 2 3 4 5

$letra
[1] "Z"

$lógico
[1] TRUE

$vetor
[1] "a" "b"

$nome
[1] "Razer"

$sobrenome
[1] "Montaña"
```

Prof. Dr. Razer A N R Montaña

SEPT/UFR

18

18

Listas: Conversão

- Pode-se converter uma **lista** para **vetor** com `unlist()`

- Exemplo

```
> x <- list(1:5, "Z", TRUE, c("a", "b"))
> names(x) <- c("numeros", "letra", "lógico", "vetor")
> unlist(x)
numeros1 numeros2 numeros3 numeros4 numeros5  letra  lógico  vetor1  vetor2
      "1"      "2"      "3"      "4"      "5"    "Z"   "TRUE"    "a"    "b"
```

Listas: Conversão.

- Pode-se converter um **vetor** para **lista** com `as.list()`

- Exemplo

```
> vet <- 1:3
> vet
[1] 1 2 3
> l <- as.list(vet)
> l
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3
```



Exercícios

- Sejam os seguintes três vetores

```
v1 <- c(2005:2016)
v2 <- c(1:12)
v3 <- c(1:31)
```

- Defina uma lista chamada *datas* que, ao ser impressa, seja:

```
$anos
[1] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016

$meses
[1] 1 2 3 4 5 6 7 8 9 10 11 12

$dias
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

21

21



Exercícios

- Sejam o seguinte vetor:

```
v <- c(1,3,4,7,11,18,29)
```

- Crie uma lista com os dados x^2 , $x/2$ e \sqrt{x}

```
$`x*2`
[1] 2 6 8 14 22 36 58

$x/2`
[1] 0.5 1.5 2.0 3.5 5.5 9.0 14.5

$`sqrt(x)`
[1] 1.000000 1.732051 2.000000 2.645751 3.316625 4.242641 5.385165
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

22

22



Exercícios.

- Dada lista anterior, apresente o seguinte vetor

2.000000 2.645751 3.316625

Procure a posição dos dados no vetor e acesse-os. Não é necessário buscar pelos números decimais.