

1. Indica la salida de cada una de las siguientes expresiones y verifica el resultado en DrRacket.

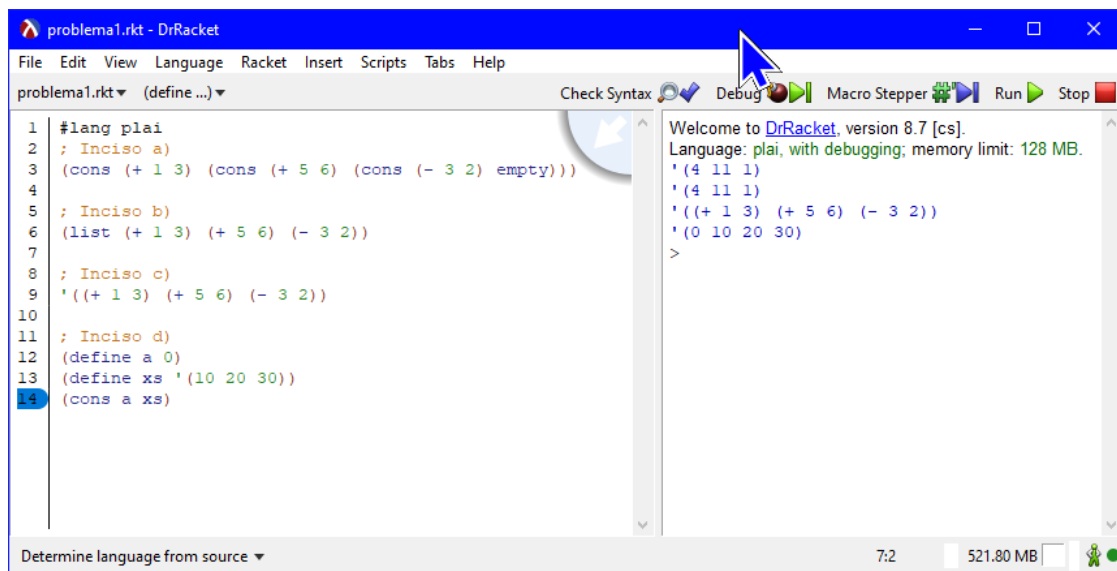
- `(cons (+ 1 3) (cons (+ 5 6) (cons (- 3 2) empty)))`
- `(list (+ 1 3) (+ 5 6) (- 3 2))`
- `'((+ 1 3) (+ 5 6) (- 3 2))`
- `(define a 0)`  
`(define xs '(10 20 30))`  
`(cons a xs)`

### Solución:

- `(cons (+ 1 3) (cons (+ 5 6) (cons (- 3 2) empty)))`, la salida es: `'(4 11 1)`
- `(list (+ 1 3) (+ 5 6) (- 3 2))`, la salida es: `'(4 11 1)`
- `'((+ 1 3) (+ 5 6) (- 3 2))`, la salida es: `'((+ 1 3) (+ 5 6) (- 3 2))`
- `(define a 0)`  
`(define xs '(10 20 30))`  
`(cons a xs)`

la salida es: `'(0 10 20 30)`

La comprobacion aqui esta:



2. Realiza los siguientes ejercicios en el orden indicado. El objetivo de este ejercicio es aprender a usar la documentacion de Racket. En este caso, debes consultar la documentacion de Pares y Lista.

1. Construye una lista con el nombre de tus canciones favoritas (de preferencia mas de 5) usando `cons` o `list`, el que mas te guste. Dale un nombre con `define`.

2. Investiga como verificar si una lista esta vacía, anota el nombre de la función y úsala para verificar que en efecto tu lista no esta vacía.
3. Investiga como obtener el numero de elementos de una lista, anota el nombre de la función y úsala para conocer el numero de elementos de tu lista.
4. Investiga como obtener el primer elemento de una lista, anota el nombre de la función y úsala con tu lista.
5. Crea otra lista, ahora con el nombre de tus películas favoritas (igual mas de 5). Dale un nombre con define. Debe ser distinta de la primera.
6. Investiga como concatenar dos listas, anota el nombre de la función y úsala para crear una nueva lista que sea la concatenación de las dos anteriores. Dale un nombre con define. Debe ser distinto a la primera y a la segunda.
7. Investiga como eliminar un elemento de una lista (a partir de un indice), anota su nombre y úsala para quitar el segundo elemento de la primera lista, el tercero de la segunda y el cuarto de la tercera.

**Solución:**

1. Para el este inciso, la respuesta se vería asi:

```
(define miMusica (list "Information High" "Sin Frenos" "Tu y yo" "Asi soy" "Muchacha"))
```

2. Para el este inciso, la respuesta se vería asi: `(empty? miMusica)`

3. Para el este inciso, la respuesta se vería asi: `(length miMusica)`

4. Para el este inciso, la respuesta se vería asi: `(car miMusica)`

5. Para el este inciso, la respuesta se vería asi:

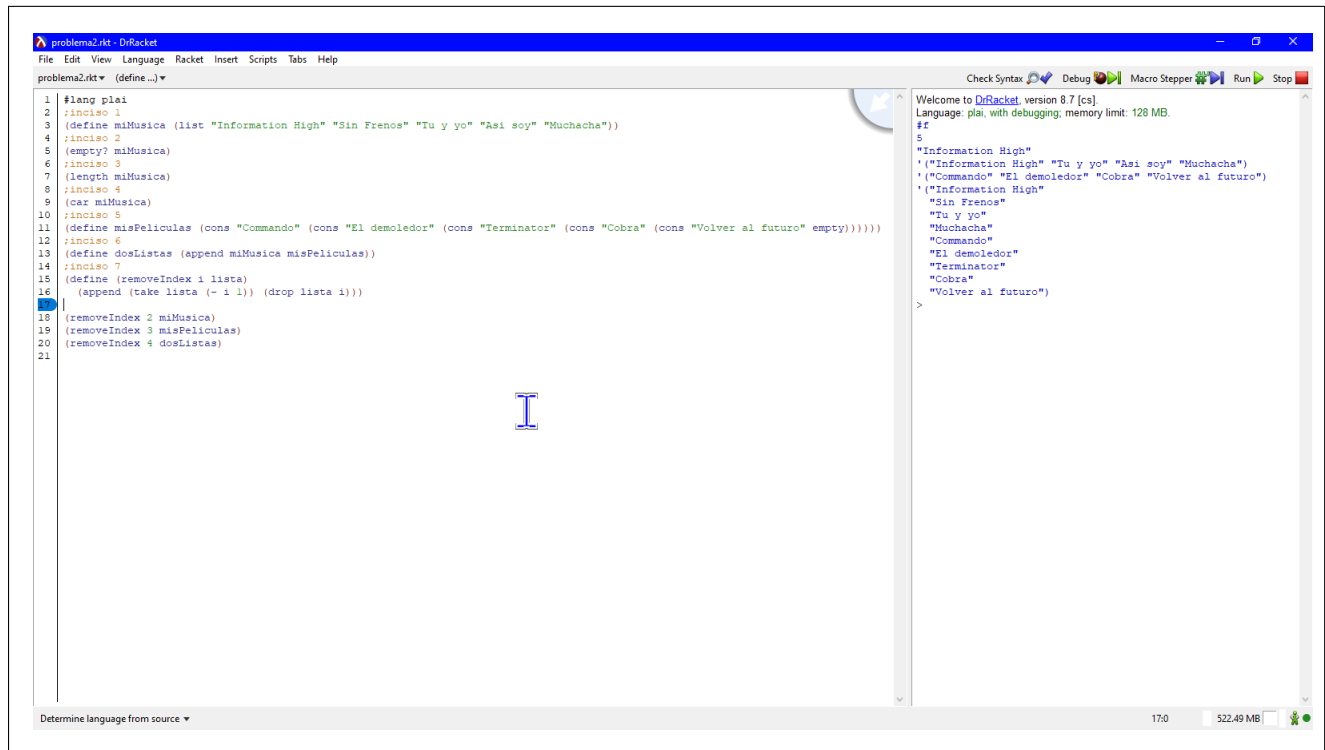
```
(define misPelículas  
  (cons "Commando"  
    (cons "El demoledor"  
      (cons "Terminator"  
        (cons "Cobra"  
          (cons "Volver al futuro" empty)))))))
```

6. Para el este inciso, la respuesta se vería asi: `(define dosListas (append miMusica misPelículas))`

7. Para el este inciso, la respuesta se vería asi:

```
(define (removeIndex i lista)  
  (append (take lista (- i 1)) (drop lista i)))  
  
(removeIndex 2 miMusica)  
(removeIndex 3 misPelículas)  
(removeIndex 4 dosListas)
```

Aqui esta la comprobacion en DrRacket



The screenshot shows the DrRacket IDE interface. The main editor window displays a Racket script with the following code:

```
1 #lang plai
2 ;inciso 1
3 (define miMusica (list "Information High" "Sin Frenos" "Tu y yo" "Asi soy" "Muchacha"))
4 ;inciso 2
5 (empty? miMusica)
6 ;inciso 3
7 (length miMusica)
8 ;inciso 4
9 (car miMusica)
10 ;inciso 5
11 (define misPelículas (cons "Commando" (cons "El demoedor" (cons "Terminator" (cons "Cobra" (cons "Volver al futuro" empty))))))
12 ;inciso 6
13 (define dosListas (append miMusica misPelículas))
14 ;inciso 7
15 (define (removeIndex i lista)
16   (append (take lista (- i 1)) (drop lista i)))
17
18 (removeIndex 2 miMusica)
19 (removeIndex 3 misPelículas)
20 (removeIndex 4 dosListas)
21
```

The right-hand pane shows the output of the script, which is a list of strings:

```
Welcome to DrRacket, version 8.7 [cs].
Language: plai, with debugging, memory limit: 128 MB.
#f
5
"Information High"
'("Information High" "Tu y yo" "Asi soy" "Muchacha")
'("Commando" "El demoedor" "Cobra" "Volver al futuro")
'("Information High"
  "Sin Frenos"
  "Tu y yo"
  "Muchacha"
  "Commando"
  "El demoedor"
  "Terminator"
  "Cobra"
  "Volver al futuro")
>
```

The status bar at the bottom indicates the language is determined from the source, and the memory usage is 522.49 MB.