TAREA 1 LENGUAJES DE PROGRAMACIÓN

NAVA BANDA JUAN ADAN

Revisado por Gael Balderrama Dominguez

* Si la variable es declarada en el bloque actual, usamos esa declaración.

1. (define (ejemplo1)

(define x 10) ; Declaración de variable local en el bloque actual

(display x))

(ejemplo1) ; Llamada a la función

1. (define x 20) ; Declaración de variable global

(define (ejemplo2)

(display x)) ; Usamos la declaración global en el bloque actual

(ejemplo2) ; Llamada a la función

1. (define x 30) ; Declaración de variable global

(define (ejemplo3)

(define x 40) ; Declaración de variable local en el bloque actual

(display x)) ; Usamos la declaración local en el bloque actual

(display x) ; Mostramos la declaración global fuera de la función

(newline) ; Salto de linea (Espaciador entre respuestas

(ejemplo3) ; Llamada a la función

* Programa 1: Uso de variable local en el bloque actual
* Se define una función llamada ejemplo1.
* Dentro de la función, se declara una variable local x con el valor 10.
* Se llama a la función ejemplo1.
* Dentro de la función, se muestra el valor de la variable local x, que es 10.

Revisión: Consola muestra correctamente el valor de X qué es 10 ya qué se define el valor de la variable dentro de el bloque de la función ejemplo

* Programa 2: Uso de variable global en el bloque actual
* Se declara una variable global x con el valor 20.
* Se define una función llamada ejemplo2.
* Dentro de la función, se muestra el valor de la variable global x, que es 20.
* Se llama a la función ejemplo2.

Revisión: Consola muestra el valor de X como 20 ya qué dentro de la función ejemplo2 no existe una definición de X de manera local así que subí al bloque superior y usa la variable global

* Programa 3: Uso de variable local y global en el bloque actual
* Se declara una variable global x con el valor 30.
* Se define una función llamada ejemplo3.
* Dentro de la función, se declara una variable local x con el valor 40, que prevalece dentro del bloque de la función.
* Se muestra el valor de la variable global x, que es 30, fuera de la función.
* Se llama a la función ejemplo3.
* Dentro de la función, se muestra el valor de la variable local x, que es 40.

Revisión: Consola muestra 2 valores de X, primero el que se encuentra dentro del bloque de la función ejemplo3 que sería 40, luego se imprime X fuera de cualquier función haciendo qué agarre el valor de la variable global siendo 30

* De lo contrario, buscamos en el bloque donde aparece el bloque actual, continuando recursivamente.

1. (define x 10) ; Declaración de variable global

(define (ejemplo1)

(define x 20) ; Declaración de variable local en el bloque de la función

(display x))

(ejemplo1) ; Llamada a la función

1. (define x 30) ; Declaración de variable global

(define (ejemplo2)

(display x)) ; Usamos la declaración global en el bloque de la función

(ejemplo2) ; Llamada a la función

* Programa 4: Búsqueda de variable local en el bloque actual
* Se declara una variable global x con el valor 10.
* Se define una función llamada ejemplo1.
* Dentro de la función, se declara una variable local x con el valor 20 en el mismo bloque.
* Se llama a la función ejemplo1.
* Dentro de la función, se muestra el valor de la variable local x, que es 20.

Revisión: Se declara la variable x con valor de 10 de manera global, luego se define la función ejemplo1, en esta se declara una variable local de x con valor de 20, a la hora de llamar a ejemplo1 se imprime la x con valor de 20

* Programa 5: Búsqueda de variable global en el bloque actual
* Se declara una variable global x con el valor 30.
* Se define una función llamada ejemplo2.
* Dentro de la función, se muestra el valor de la variable global x, que es 30.
* Se llama a la función ejemplo2.

Revisión: se declara una variable global con valor de 30, luego se define la función ejemplo2, dentro de esta se hace referencia a x qué no esta declarada dentro del bloque así que sube 1 nivel del bloque y utiliza el valor de 30.

* Si el bloque actual ya es el bloque primordial y aún no hemos encontrado una declaración correspondiente, la variable no está ligada.

1. (define x 10) ; Declaración de variable global en el bloque primordial

(define (ejemplo1)

(display x)) ; Variable ligada en el bloque primordial

(ejemplo1) ; Llamada a la función

1. (define (ejemplo2)

(display y)) ; Intento de uso de variable no declarada

(ejemplo2) ; Llamada a la función

* Programa 1: Variable ligada en bloque primordial
* Se declara una variable global x con el valor 10 en el bloque primordial.
* Se define una función llamada ejemplo1.
* Dentro de la función, se muestra el valor de la variable global x, que está ligada en el bloque primordial.

Revisión: se define variable x de manera global, se define la función ejemplo1 qué no contiene la definición local de X así que sube 1 nivel de bloque y agarra la definición global

* Programa 2: Variable no ligada en bloque primordial
* Se define una función llamada ejemplo2.
* Dentro de la función, se intenta mostrar el valor de la variable y, que no ha sido declarada en ningún lugar.
* Al ejecutar este programa, obtendrás un error indicando que la variable y no está ligada en ningún bloque.

Revisión: Se busca la variable x dentro de la función ejemplo2, no la encuentra en este bloque y sube 1 nivel de bloque, no la encuentra dentro del bloque primordial así que regresa un error

* Cada defvar evalua la expresión inmediatamente y vincula la varialbe al valor, incluso si la variable no es usada en el resto del programa.

1. (defvar x (+ 2 3)) ; defvar vincula x al valor de (+ 2 3)

(display "Programa ejecutado") ; Mensaje para indicar la ejecución

1. (define (ejemplo)

(defvar y (\* 4 5)) ; defvar vincula y al valor de (\* 4 5)

(display "Función ejecutada")) ; Mensaje para indicar la ejecución

(ejemplo) ; Llamada a la función

* Programa 8: Uso de defvar con variable no utilizada
* Se utiliza defvar para vincular la variable x al valor de la expresión (+ 2 3), que es 5.
* Aunque la variable x no se utiliza en el resto del programa, la expresión se evalúa inmediatamente y la vinculación se realiza.

Revisión: Aunque no se imprime el valor de X, dentro del programa si se evalúa la variable X y se vincula dentro del codigo

* Programa 9: Uso de defvar con variable no utilizada en una función
* Se define una función llamada ejemplo.
* Dentro de la función, se utiliza defvar para vincular la variable y al valor de la expresión (\* 4 5), que es 20.
* Aunque la variable y no se utiliza en el resto de la función, la expresión se evalúa inmediatamente y la vinculación se realiza cuando se llama a la función.

Revisión: No se imprime el valor de Y pero se evalúa y se le vincula el valor de 20 (\*4 5) , luego se usa una función qué regresa una cadena de texto que muestra qué se recorrió el código

* Cada llamada a función evalúa los argumentos inmediatamente y vincula los valores a los parámetros formales, incluso cuando los parámetros formales no son usados en la función.

1. (define (ejemplo1 a b)

(display "Función ejecutada"))

(ejemplo1 (+ 2 3) (\* 4 5)) ; Llamada a la función con argumentos evaluados

1. (define (funcion-interna x)

(display "Función interna ejecutada"))

(define (ejemplo2 a b)

(display "Función ejecutada")

(funcion-interna (\* a b)))

(ejemplo2 2 3) ; Llamada a la función con argumentos evaluados

* Programa 1: Parámetros formales no utilizados en la función
* Se define una función llamada ejemplo1 que tiene dos parámetros formales a y b, pero no los utiliza.
* Se realiza una llamada a la función ejemplo1 con los argumentos evaluados (+ 2 3) y (\* 4 5). Aunque los parámetros a y b no se utilizan en la función, los valores de los argumentos se vinculan a estos parámetros.

Revision: se evalúan a y b, luego se le vinculan los valores aunque no se utilizen estas variables dentro de la función

* Programa 2: Parámetros formales no utilizados en una función anidada
* Se define una función llamada funcion-interna que toma un parámetro formal x, pero no lo utiliza.
* Se define una función llamada ejemplo2 con los parámetros formales a y b, y dentro de esta función se llama a funcion-interna con el argumento (\* a b).
* Al llamar a ejemplo2 con los argumentos evaluados 2 y 3, la expresión (\* a b) se evalúa y el valor se vincula al parámetro x de la función funcion-interna, aunque no se utilice en ella

Revision: la función funcion-interna recibe un parámetro x, este nunca se emplea en su interior. El valor obtenido de la expresión (\* a b) se enlaza con el parámetro x, aunque no se aproveche más adelante en el cuerpo de `funcion-interna".