

# Il linguaggio Javascript

---

---

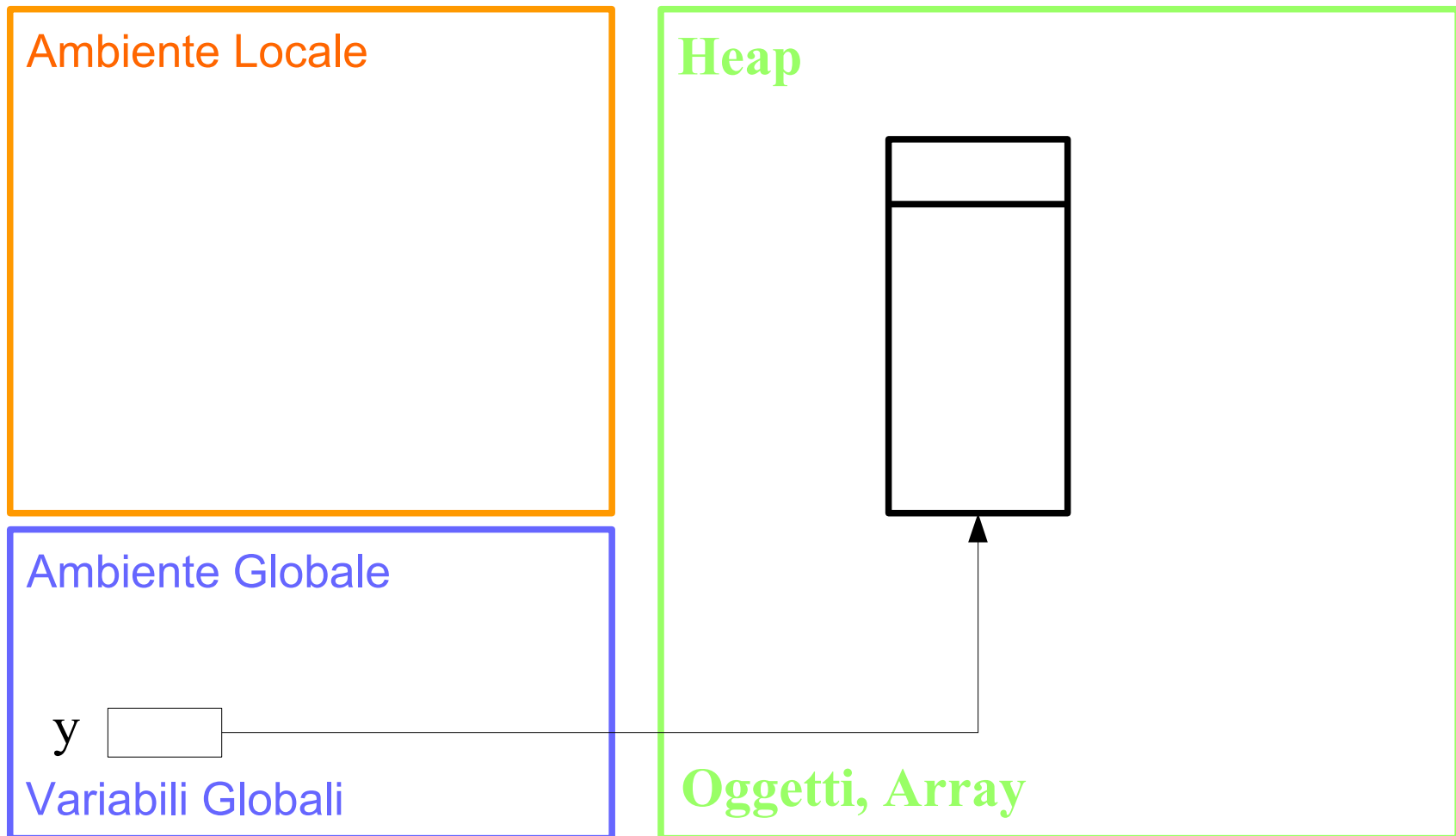
Modello di Memoria

Scope delle variabili e passaggio dei parametri

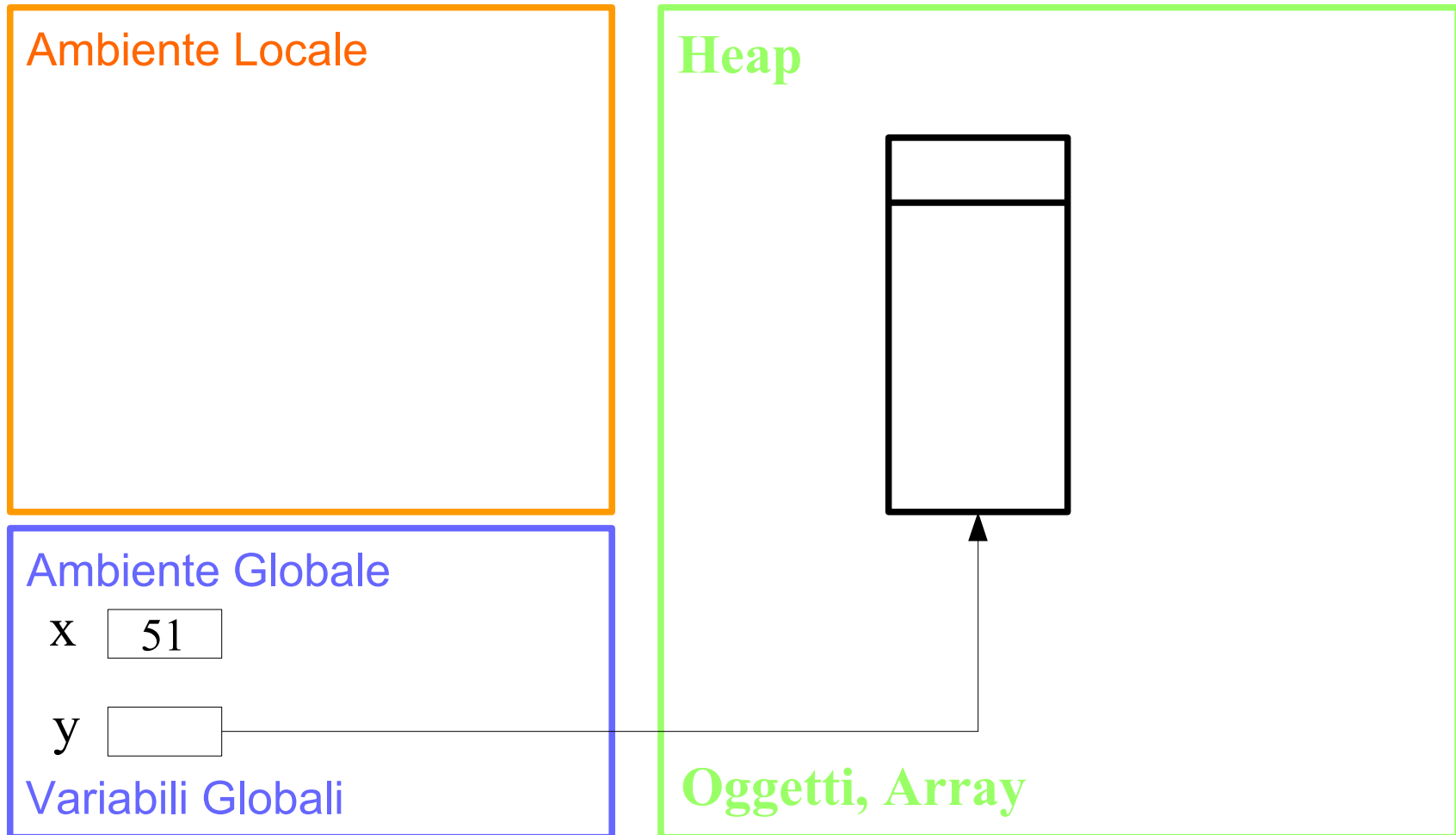
# Modello di Memoria per Javascript

---

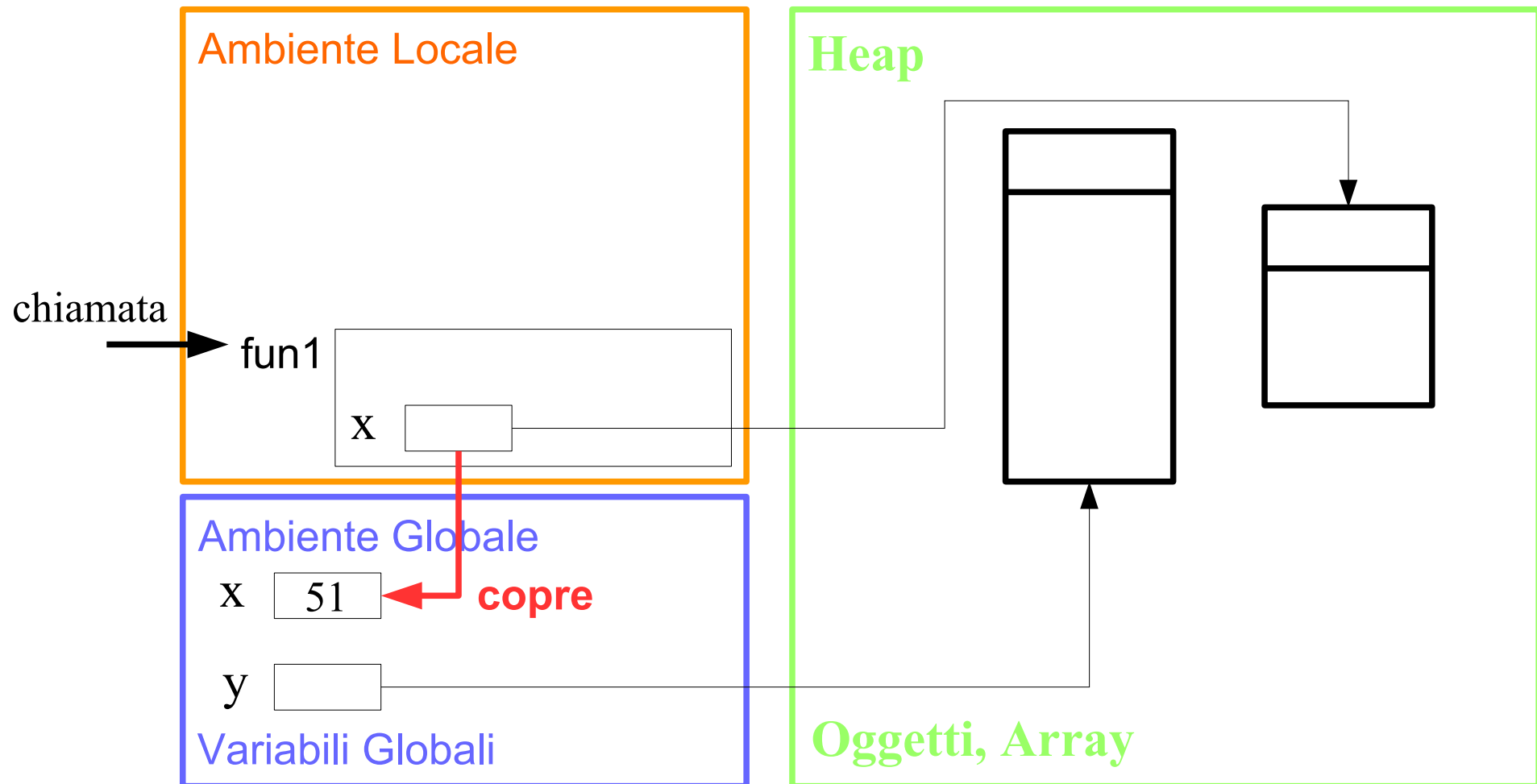
---



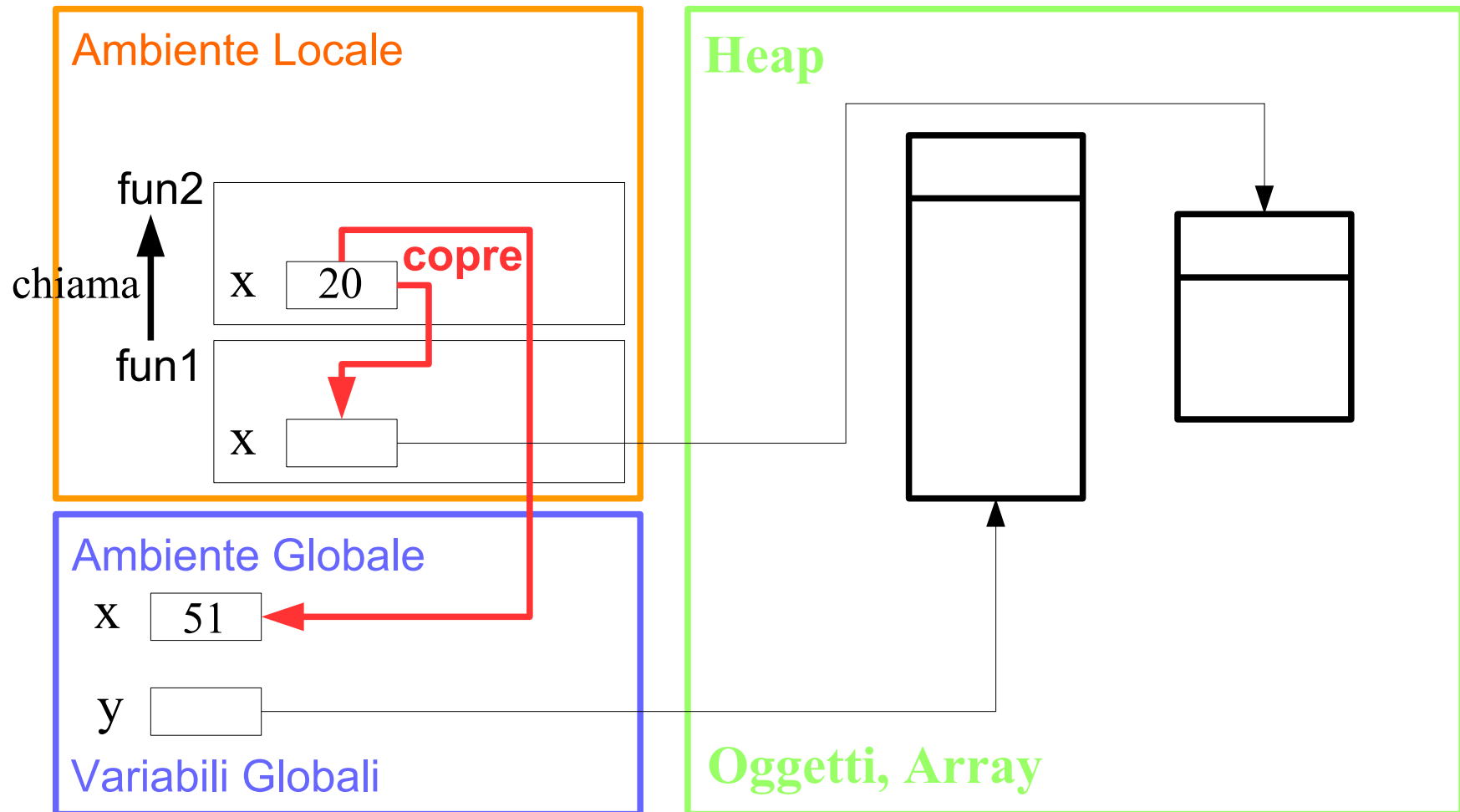
# Modello di Memoria per Javascript



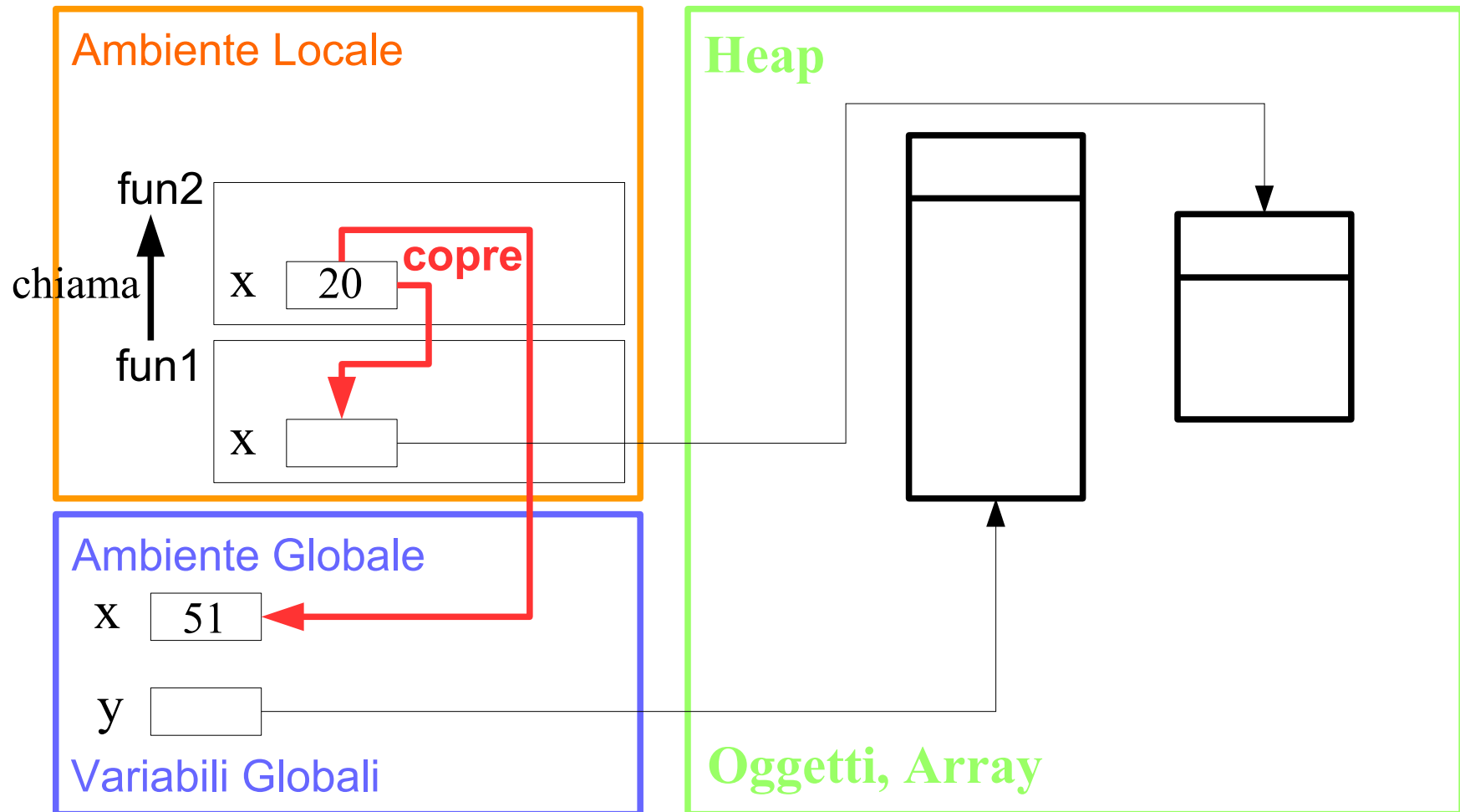
# Modello di Memoria per Javascript



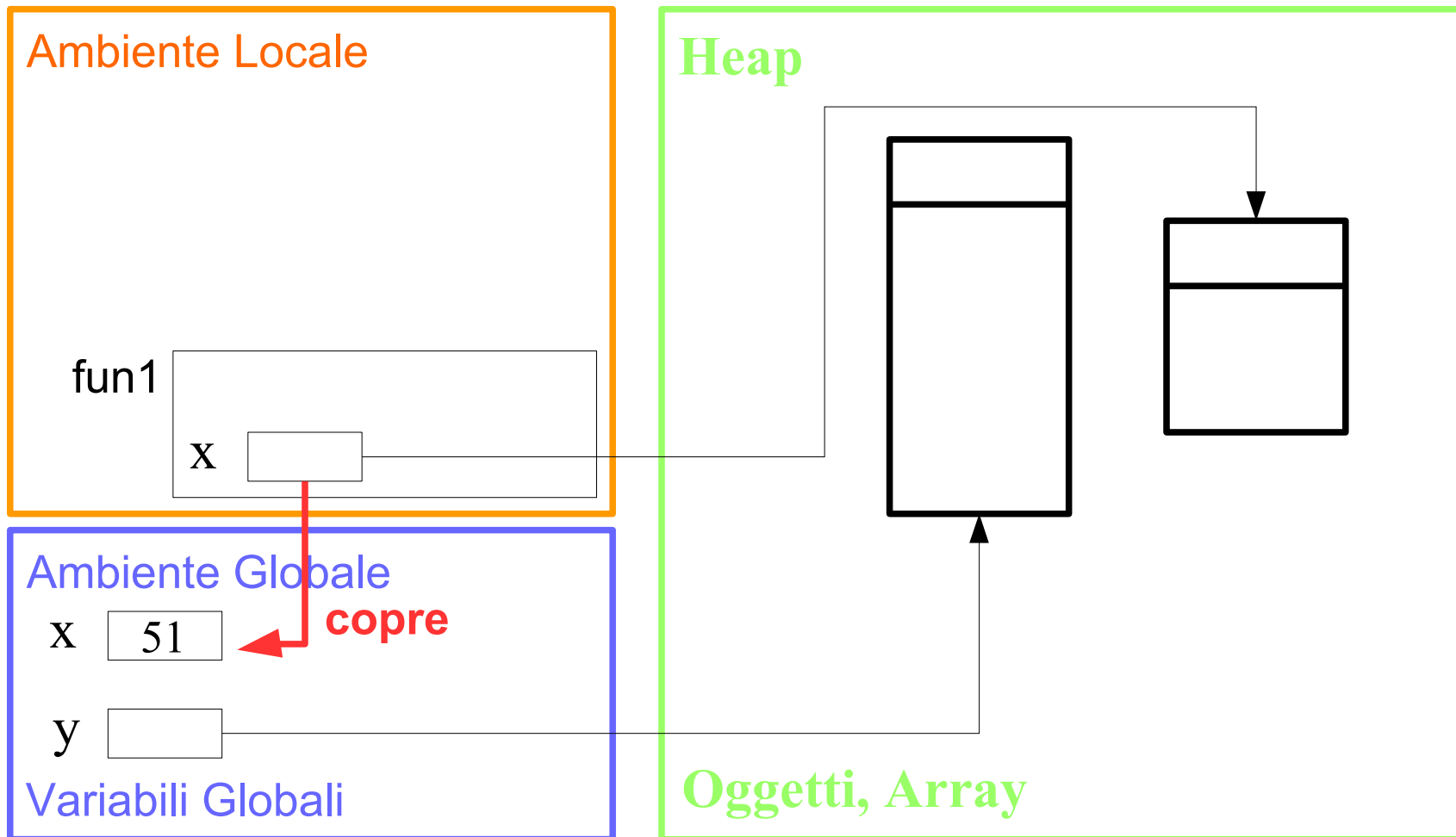
# Modello di Memoria per Javascript



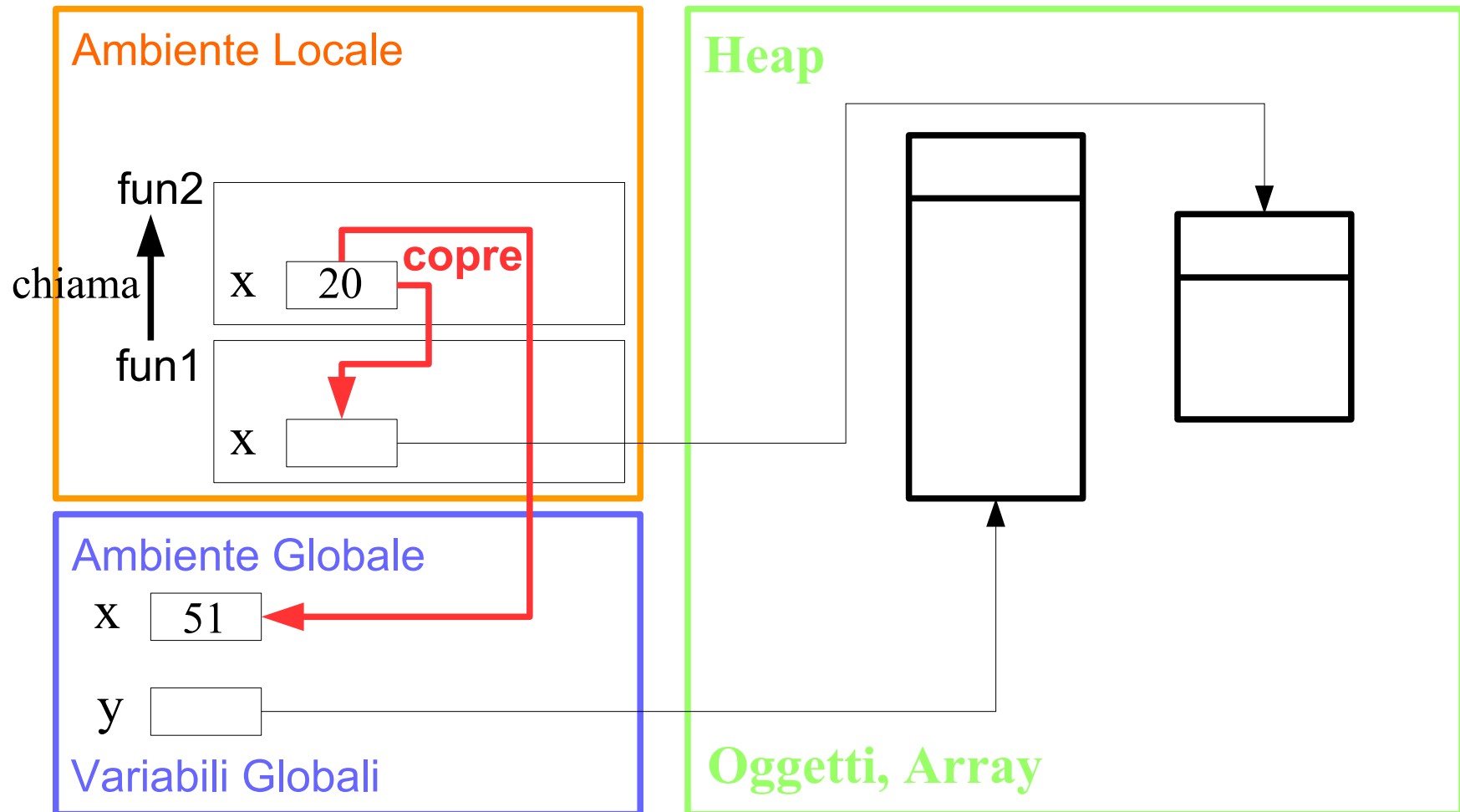
# Modello di Memoria per Javascript



# Modello di Memoria per Javascript

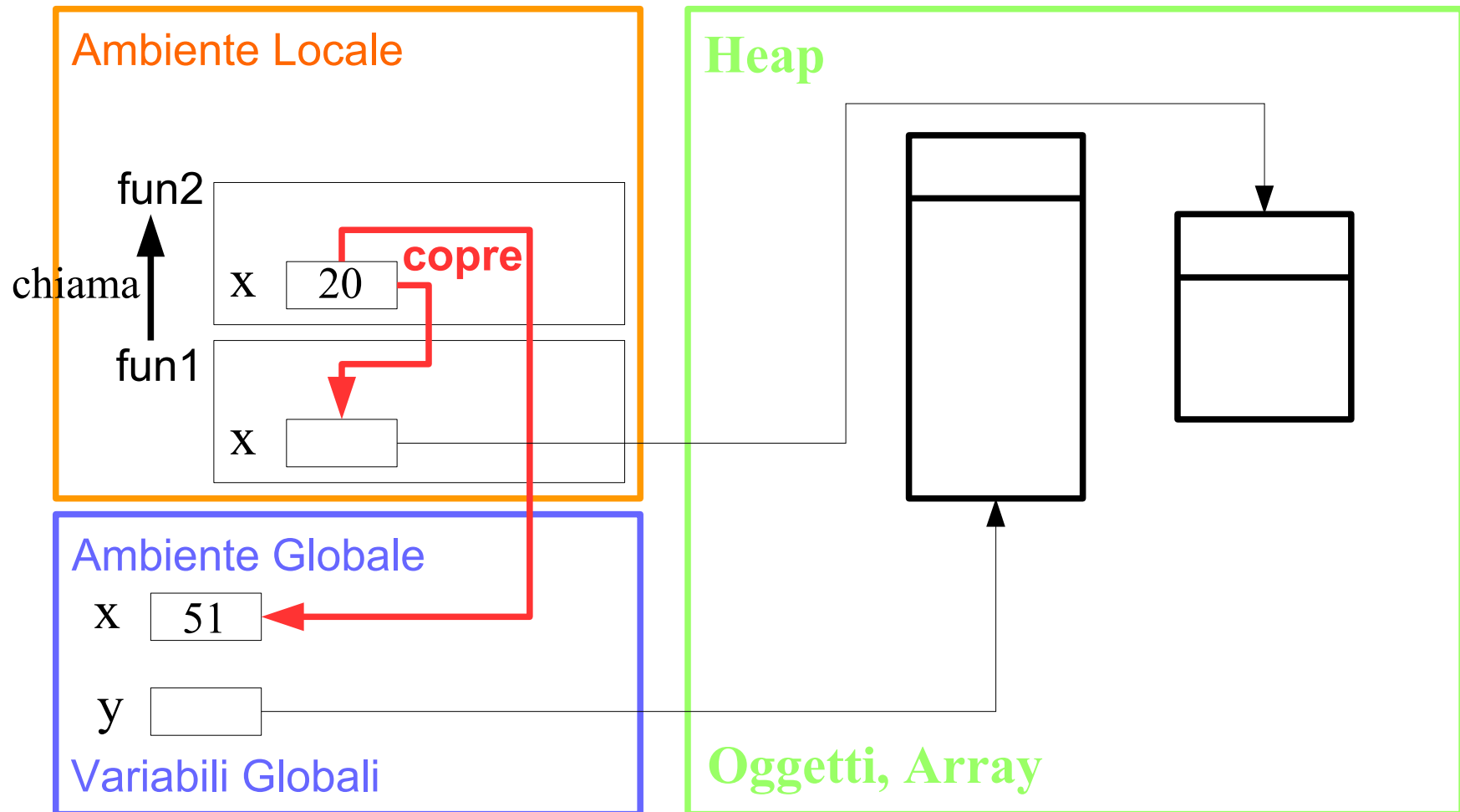


# Modello di Memoria per Javascript

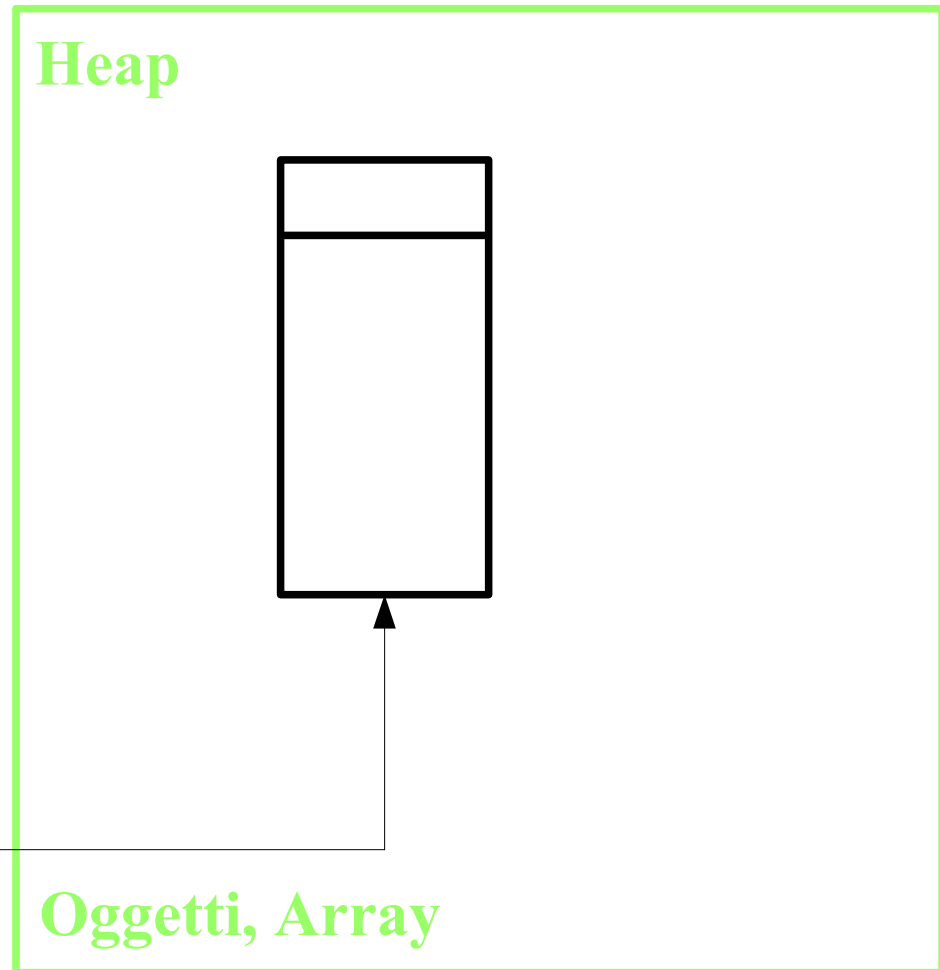
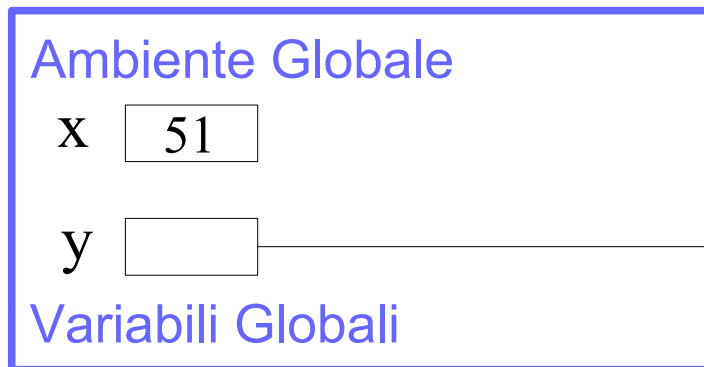




# Modello di Memoria per Javascript



# Modello di Memoria per Javascript



# Scope delle variabili

---

---

- Le variabili globali sono visibili ovunque tranne quando è definita una variabile locale con lo stesso nome
- Le variabili locali sono visibili solo dentro la funzione in cui sono dichiarate

# Ciclo di vita delle variabili

---

---

- Le variabili globali appaiono nell'ambiente globale quando vengono dichiarate (o usate la prima volta)
- Esse persistono fino alla chiusura della pagina web nel browser
- Le variabili locali (inclusi i parametri formali) appaiono nell'attivazione della funzione che le dichiara all'interno dell'ambiente locale
- Esse persistono fino al ritorno della funzione

# Passaggio dei parametri

---

---

- I parametri dei tipi di base:
  - number, string, boolean
- Sono passati per valore
- I parametri di tipo Array o Object vengono passati sempre per valore ma come accade in Java, tale valore è un riferimento!
- Quindi la funzione può modificare gli array o gli oggetti passati come parametri
- Vedi codice allegato variablesJs.html

# W3C Schools Javascript Tutorial

---

---

- JS Scope
- JS Events
- JS Arrays
- JS Array Methods
- JS Conditions / Switch / Cycles
- JS Type Conversion
- JS Errors and Debugging

# Esercitazione

---

---

- Realizzare una pagina web interattiva che permette a due giocatori di giocare a Tris
- Usare immagini diverse per i diversi stati della scacchiera
- Usare un colore e un simbolo per ogni giocatore
- Dichiarare la vittoria o la parità alla fine di una partita
- Permettere sempre di ricominciare una nuova partita