JavaScript Core Language

Prof. Francesco Accarino
IIS Atiero Spinelli
Sesto San Giovanni via leopardi 132

Condizioni

- L’utilizzo di operatori relazionali e logici consente di formulare delle condizioni che possono essere utilizzate per controllare l’esecuzione del programma

(Pagamento==“contrassegno”)&&(Italiano)

Controllo dll'esecuzione

- L'esecuzione di un programma è generalmente sequenziale
- Tuttavia in determinate "condizioni" può essere necessario eseguire solo alcune istruzioni, ma non altre, oppure ripetere più volte un'operazione

Sintassi della struttura di Selezione

La struttura di selezione si codifica così

```
if (condizione)
    istruzione1;
else
    istruzione2 ;
```

Se sono presenti più istruzioni in ciascun ramo

```
if (condizione)
        {
          istruzione1;
          istruzione2;
        }
else
        {
          istruzione1;
          istruzione2;
        }
```

RICORDA:
Le istruzioni che fanno parte della sequenza sono racchiuse tra le parentesi graffe.

LA SELEZIONE A UNA VIA

In alcuni casi è possibile scegliere di compiere o non compiere un'azione;
in tal caso l'azione da eseguire **deve essere** posta sul ramo del SI.

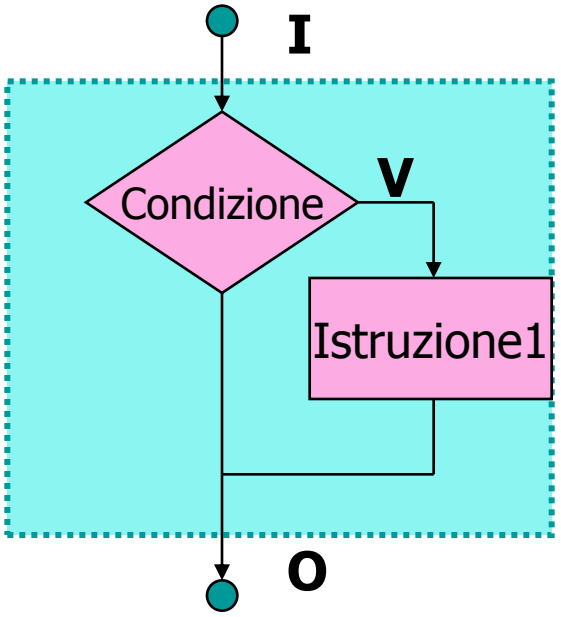
In questo caso la codifica in C++ è:

```
if (condizione)
        istruzione1;
```

o nel caso di più istruzioni:

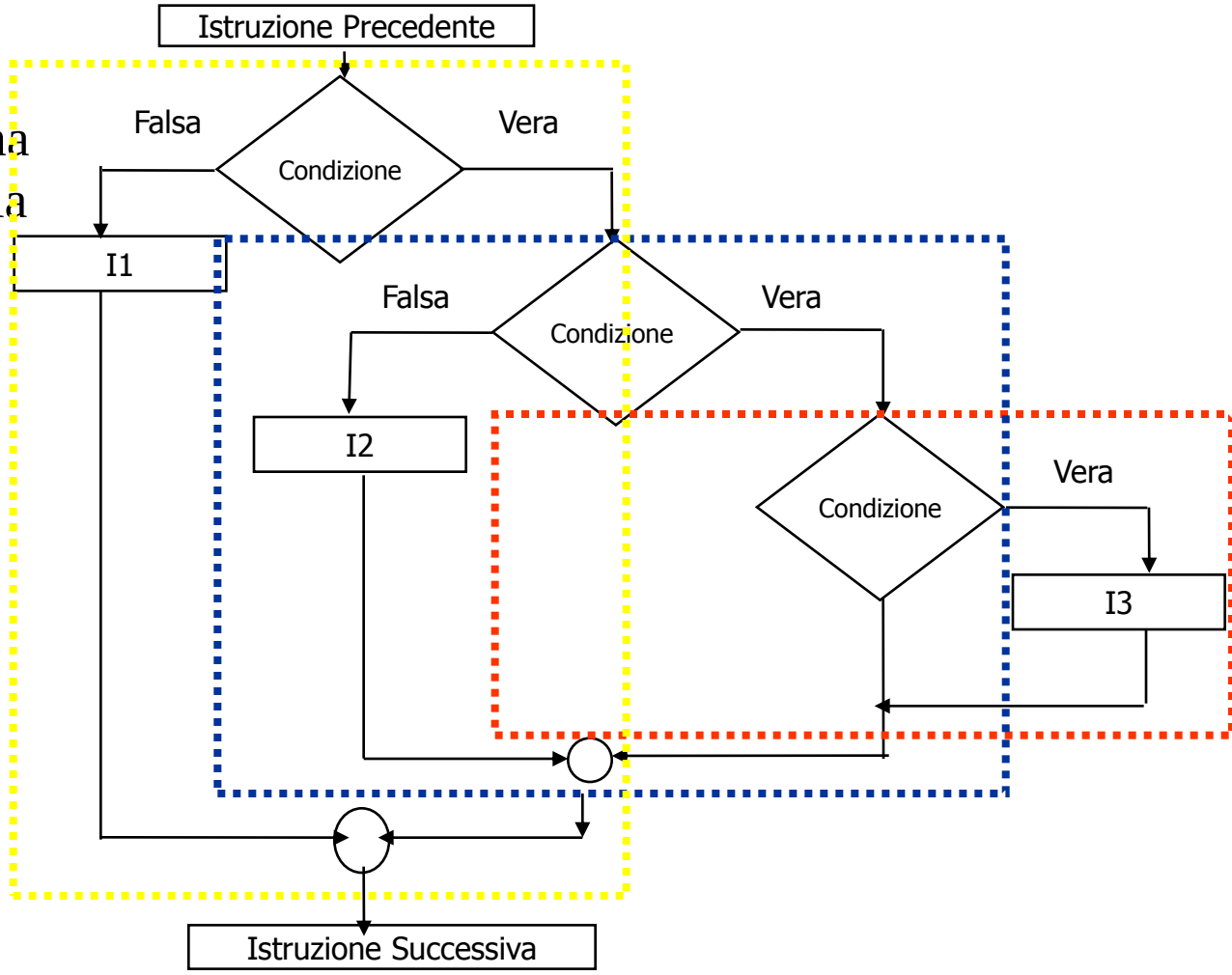
```
if (condizione)
{
    istruzione1;
    istruzione2;
    ……;
}
```

La clausola **else** (altrimenti) in queste situazioni non viene codificata



Strutture di Controllo if-else Nidificati

A volte si possono usare istruzioni di selezione (a una o a due vie) **nidificate** , ossia le istruzioni da eseguire al verificarsi della condizione sono a loro volta istruzioni condizionali



Occorre ricordare che ogni blocco deve avere un inizio e una fine

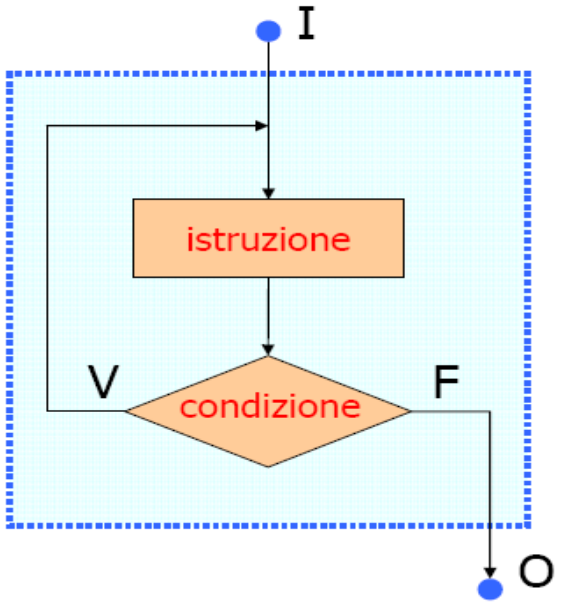
ITERAZIONE CON CONTROLLO IN CODA (do…while)

Per **iterazione** si intende la **ripetizione** di **una o più azioni** sotto il **controllo di una condizione**.

Il gruppo di azioni da ripetere è detto **corpo del ciclo**

```
Do{
    ……
   Istruzioni
   …..
}While(condizione);
```

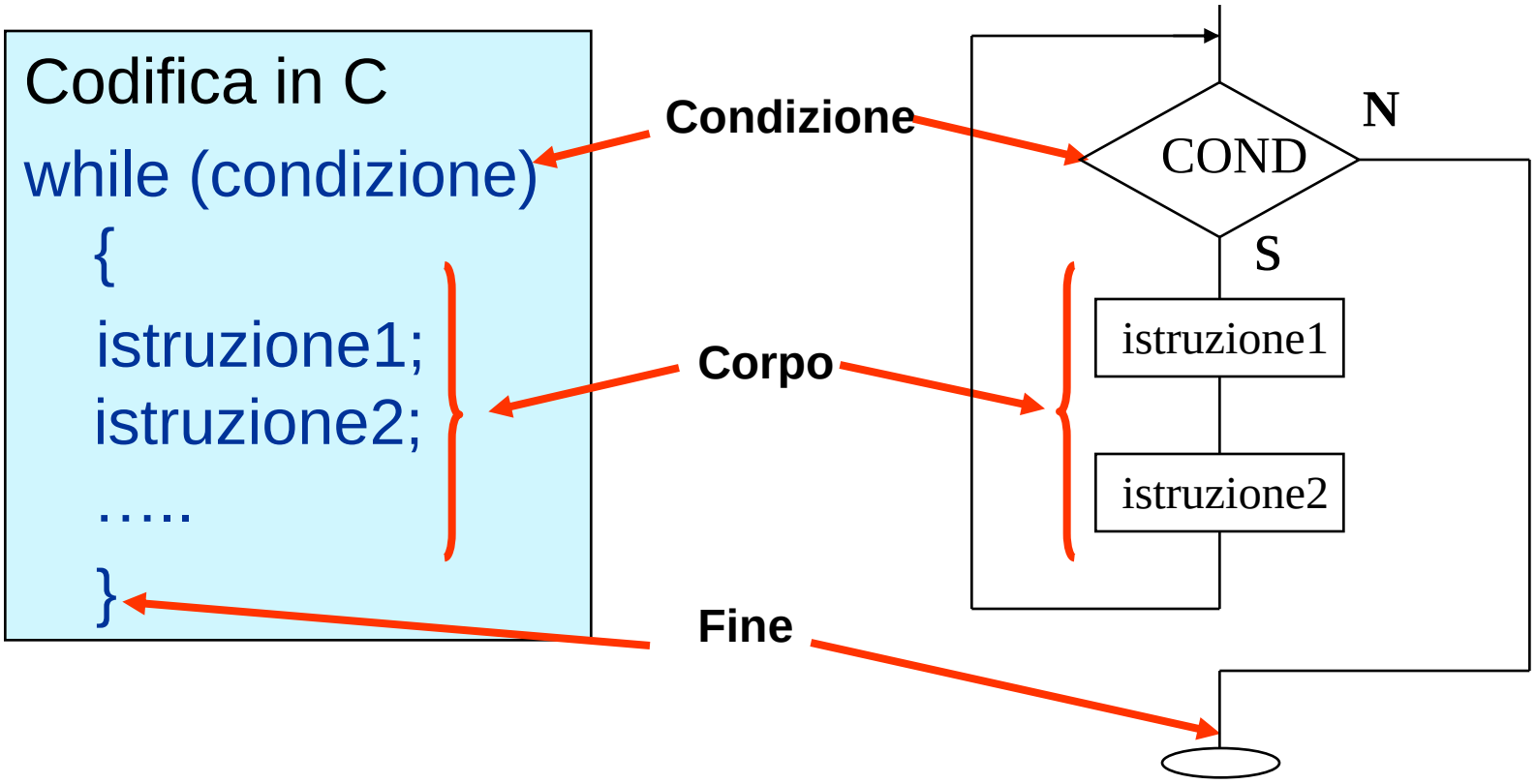
Diagramma a blocchi



E' necessario che all'interno del corpo vi sia una istruzione che modifichi il valore della condizione, altrimenti il ciclo "va in loop"

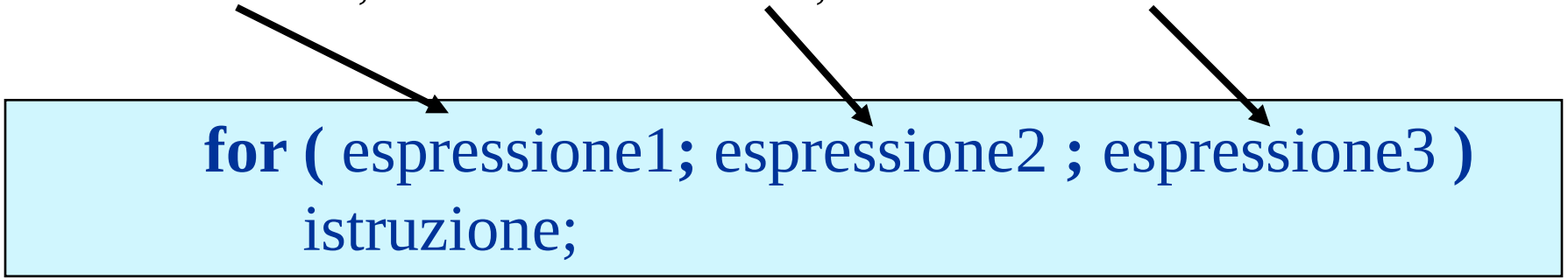
ITERAZIONE CON CONTROLLO IN TESTA (while …)

Il while verifica la condizione all'inizio del ciclo per cui **se inizialmente la condizione risulta falsa il corpo del ciclo non viene mai eseguito**



Iterazione Predefinita

L'istruzione for è composta da tre espressioni separate dal '**;**' la prima è di **inizializzazione**, la seconda di **controllo**, la terza di **incremento**.



L'ordine di esecuzione è il seguente:

1) esecuzione dell'INIZIALIZZAZIONE espressione1 (una sola volta)

2) controllo della CONDIZIONE espressione2 (ripetuta dopo ogni incremento)

3) esecuzione del CICLO istruzione (se la condizione è vera)

4) esecuzione dell'INCREMENTO espressione3 (ritorno al punto 2)

Selezione multipla

La **Selezione Multipla** è rappresentata dalla istruzione **switch-case**

```
switch (variabile){
        case <costante1>: <sequenza di istruzioni1>;    break;
        case <costante2>: <sequenza di istruzioni2>;    break;
             ....
        case <costanten>: <sequenza di istruzionin>;    break;
        default   : <sequenza di istruzioni>;           break;
          }
```

- le espressioni costanti **costante1, costante2, ..., costante*n*** devono essere numeri o stringhe
- l'istruzione break serve per terminare lo **switch** dopo aver eseguito solo la sequenza associata ad una particolare scelta (**case)** e proseguire con l'istruzione successiva allo switch-case;
- nel costrutto può comparire la clausola **default** per individuare la sequenza di operazioni da compiere quando il valore esaminato non coincide con alcuno di quelli specificati

Funzioni

- Una funzione è un elemento di un programma che calcola un valore che dipende “funzionalmente” da altri

  $Y=f(x)$

  $y=log10(x)$

- L’utilizzo delle funzioni nella programmazione strutturata aumenta la modularità e favorisce il riutilizzo

Definizione di unzioni

- In JavaScript è possibile definire una o più funzioni all'interno di un programma

  function **nome**(**arg0** , **arg1** , …, **argn-1**) {

  …

  }
- La funzione definita è identificata da **nome** e dipende dagli argomenti **arg0**, **arg1**, …,**argn-1**

Definizione di unzioni

- Somma di due numeri

```
function somma(a, b) {
return a+b;
}
```

- La funzione viene “invocata” all’interno di un’espressione :

```
var s = somma(1, 2);
```

Chiamata di funzioni

- Quando una funzione viene invocata gli argomenti sono inizializzati con i valori specificati
- Quindi si procede con l'esecuzione delle istruzioni costituenti la funzione
- L'istruzione return restituisce il valore calcolato al chiamante

Chiamata di funzioni

- La chiamata
  x = somma(1, 2)
  inizializza gli argomenti a e b
  rispettivamente ai valori 1 e 2
- L'istruzione
  return a+b;
  valuta l'espressione e restituisce il risultato
  (3) che viene assegnato alla variabile x

Variabili locali e globali

- All'interno di una funzione è possibile definire delle variabili "confinate" all'interno della funzione stessa
- Tali variabili, dette "locali", sono create all'atto dell'invocazione della funzione e sono distrutte al termine dell'esecuzione
- Il loro valore non è accessibile dall'esterno della funzione
- Ogni argomento di una funzione è una variabile locale definita implicitamente

Variabili locali e globali

- Le variabili definite all'esterno delle funzione sono denominate, invece, "globali". Si dichiarano prima delle funzioni

- A differenza delle variabili locali, le variabili globali sono accessibili da qualsiasi punto del programma, anche dall'interno di una funzione, sempre che in quest'ultima non sia stata definita una variabile locale con lo stesso nome

Funzioni predefinite

- In JavaScript sono presenti alcune funzioni predefinite

– isNaN(v) verifica se v non è un numero

– parseFloat(str) converte str in un numero decimale

–parseInt(str) converte str in un numero intero

–String(n) converte n in una stringa