



# Javascript: il browser

---

- Un browser è un programma
- Un browser interpreta il linguaggio HTML e visualizza le pagine sullo schermo
- Per la visualizzazione delle pagine un browser si appoggia al software di base
- Il software di base si appoggia alla macchina fisica per realizzare effettivamente il "rendering" della pagina su video



## Javascript: il browser

---

- noi vedremo solo il "core-language" di Javascript per imparare i principi della programmazione
- in JavaScript è possibile programmare in maniera più semplici quelle che sono le funzionalità offerte da un browser e quindi il controllo del "rendering" di documenti HTML
- approfondirete questi aspetti, il modello ad eventi ed il DOM di Javascript nel corso di IUM



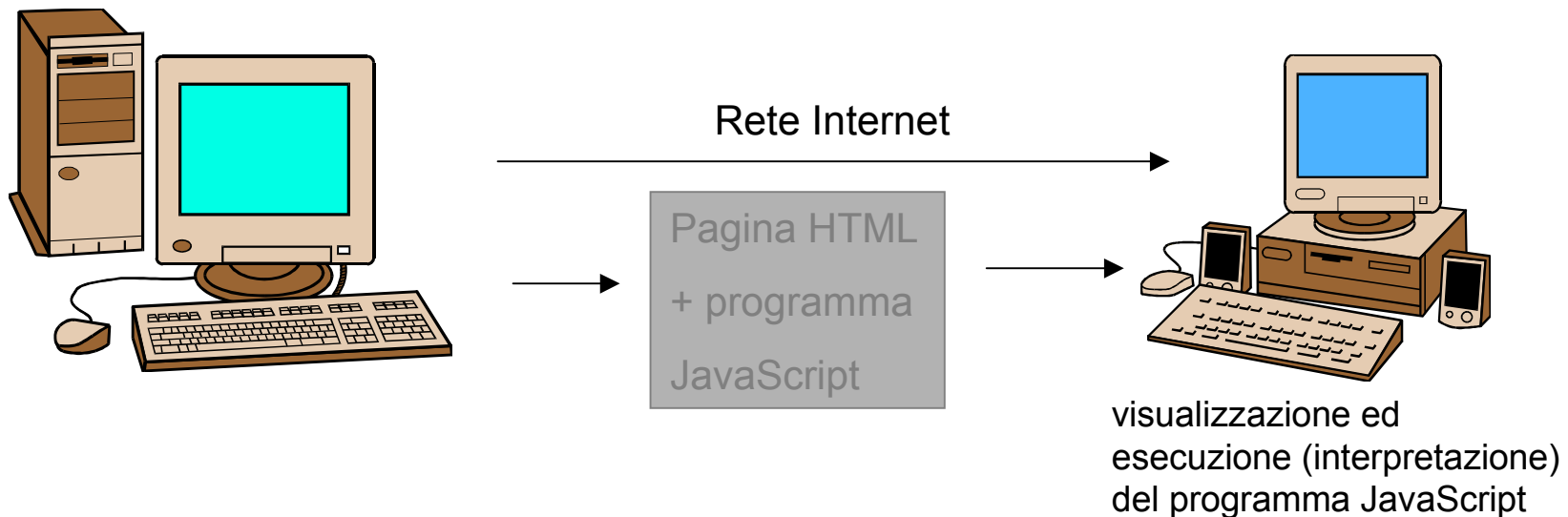
## Interpretazione di Javascript

---

- JavaScript è interpretato ed il suo interprete è il browser (es. Netscape, Internet Explorer, Opera)
- Non tutti i browser sanno interpretare JavaScript e non tutti lo fanno nella stessa maniera sebbene il linguaggio JavaScript sia uno (standard)
- Ogni comando è immediatamente tradotto in un insieme di comandi che il browser può eseguire grazie l'ausilio del software di base della macchina

# Programmi Javascript

- I programmi JavaScript (gli script) possono essere inclusi ("embedded") in una pagina HTML
- Gli script in JavaScript vengono scaricati insieme ad una pagina HTML e quindi interpretati dal browser del client





# Struttura di una pagina HTML + Javascript

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- Il codice JavaScript è incluso in una pagina HTML mediante il tag SCRIPT
- È buona norma includere il codice JavaScript all'interno di un tag di commento HTML per visualizzare la pagina HTML correttamente anche dai browser che non supportano JavaScript (ovviamente non potranno fare uso delle funzionalità dello script!)



## Specifica di inizio e di fine

---

- Ogni algoritmo deve avere un inizio ed una fine

start



end

**start**

.....

.....

.....

.....

.....

.....

**end**



# Identificatori

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- JavaScript è case-sensitive, cioè MCD e mcd sono due identificatori diversi
- Ogni istruzione termina con un punto e virgola ";"
- Gli spazi, le tabulazioni e gli a capo sono ignorati dall'interprete (quindi usarli senza timore per rendere il programma più leggibile possibile)
- Le stringhe di caratteri sono rappresentate con la notazione "..."

# Commenti

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- È possibile inserire dei commenti su di una singola linea facendo precedere la linea di commento con il simbolo "//"
- Commenti su più linee possono essere introdotti racchiudendoli tra i simboli "/\*" e "\*/"



# Convenzioni nomi variabili

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- L'identificatore assegnato ad una variabile non deve mai coincidere con quello di una parola chiave (o riservata)

Se un identificatore è ottenuto per composizione da più parole non si devono lasciare spazi tra queste, si può rimpiazzarli con il simbolo "\_" o unendo le parole e scrivendo in maiuscolo la prima lettera, ad esempio

lunghezza\_sequenza

lunghezzaSequenza



# Dichiarazione di variabili

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- Per utilizzare una variabile è necessario dichiararla
- La dichiarazione di una variabile viene effettuata tramite la parola chiave var
- È possibile combinare una dichiarazione con una inizializzazione



# Tipi di variabili

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

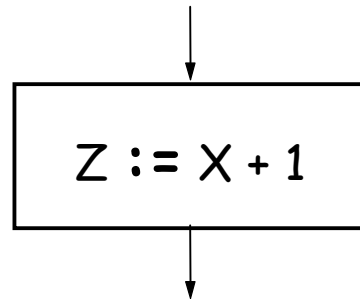
- Numeri
- Stringhe
- Booleani
- Array
- Oggetti
- Le variabili in JavaScript non hanno però alcun tipo associato alla dichiarazione:  
var x = 10;  
x = "pippo";



## Specifica delle azioni

---

- Ogni algoritmo specifica azioni che l'esecutore deve compiere del tipo descritto in precedenza



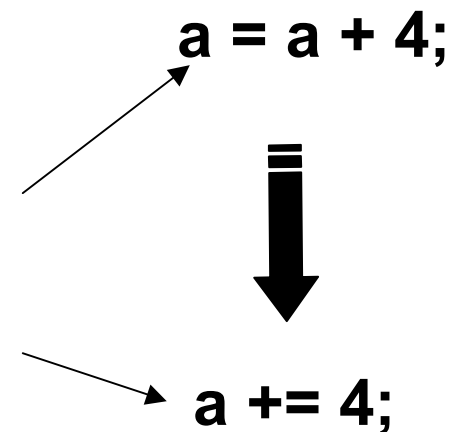
.....

$Z := X + 1$

.....

## Assegnamento a variabili ed operatori

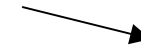
- = indica l'assegnamento
- Operatori aritmetici: + (somma), - (sottrazione), \* (prodotto), / (divisione non intera), % (resto della divisione).
- per comodità, la somma di un valore ad una variabile e l'assegnamento del totale alla variabile stessa si può indicare con +=
- analogamente si possono utilizzare -=, \*=, /=, %= ed altri ancora



## Assegnamento a variabili ed operatori

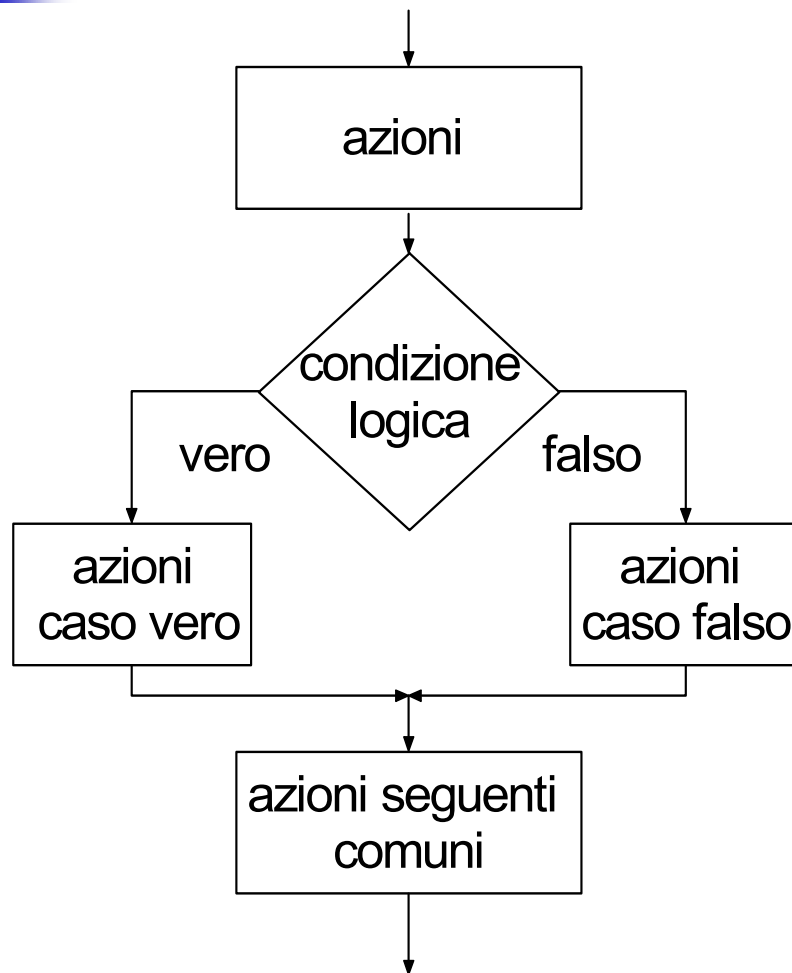
- Operatori di incremento (in posizione di prefisso o di suffisso di una variabile numerica): ++ e -- (es. i++; a=--i;)
- Operatori relazionali e booleani: ==, !=, <, >, <=, >=, &&, ||, !
- Operatori sui bit: &, |, ^, ~, >>, <<
- Parentesi: ( )

**a = a + 1;**



**a++;**

# Specifica delle condizioni logiche



.....

azioni

**if** (condizione logica) **then**

azioni caso vero

**else**

azioni caso falso

**end if**

azioni seguenti comuni

.....



## Blocchi condizionali

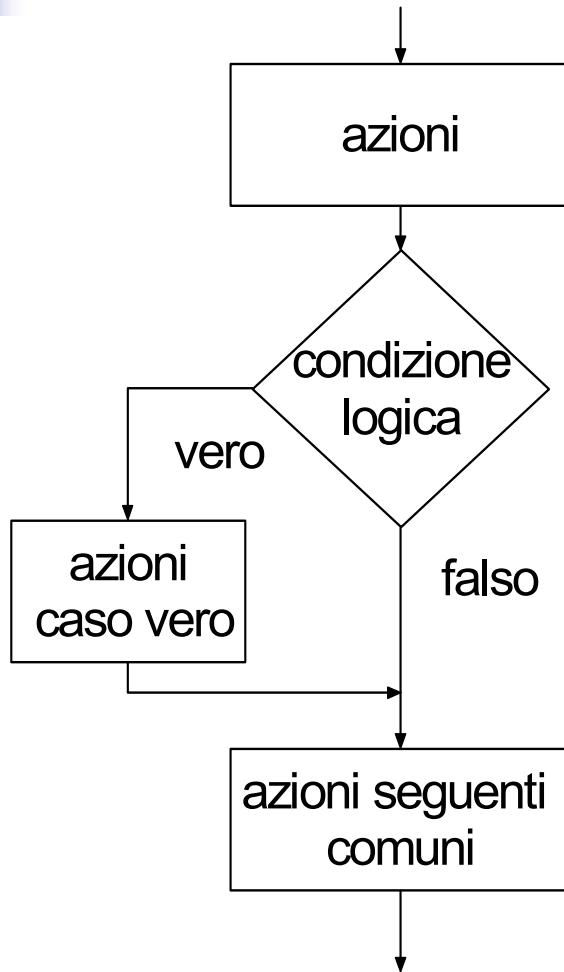
---

- Istruzione "if ... then ... else ..."

```
azioni comuni
if (condizione logica) {
    azioni caso vero
}
else {
    ... azioni caso falso
}
azioni comuni seguenti
```

- al posto di un blocco di istruzione è possibile anche avere una sola istruzione
- queste vanno sempre terminate da un punto e virgola, anche tra l'if e l'else

# Specifica delle condizioni logiche



.....  
azioni  
**if** (condizione logica) **then**  
    azioni caso vero  
**end if**  
azioni seguenti comuni  
.....



## Blocchi condizionali

---

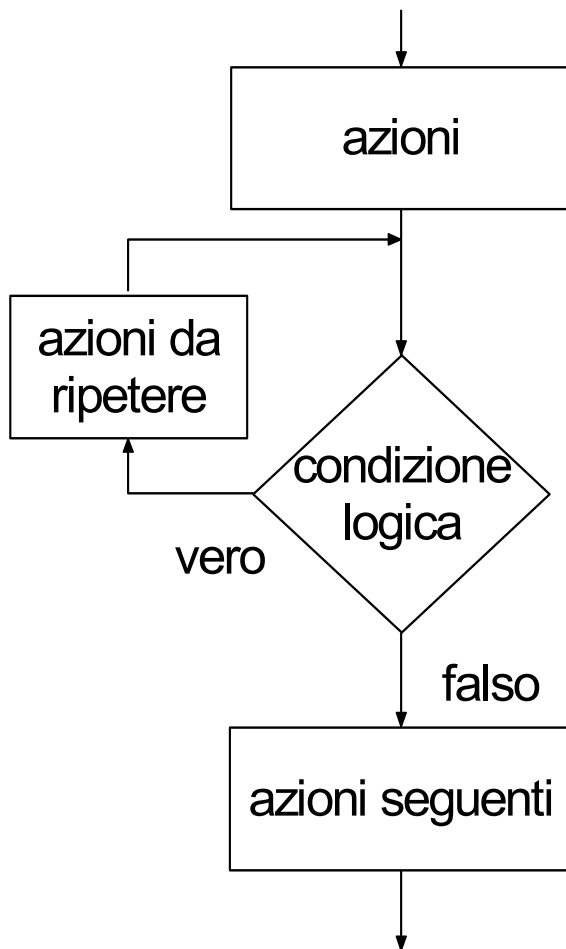
- Istruzione "if ... then"

```
if (condizione logica) {  
    azioni caso vero  
}
```

```
azioni seguenti comuni
```

- al posto di un blocco di istruzione è possibile anche avere una sola istruzione
- queste vanno sempre terminate da un punto e virgola

# Strutture di controllo: iterazione



.....

azioni

**while** (condizione logica)

azioni da ripetere

**end while**

azioni seguenti

.....



## Iterazioni while

---

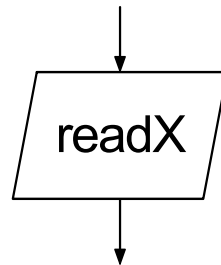
- Istruzione "while"

```
while (condizione logica)
{
    azioni da ripetere
}
```

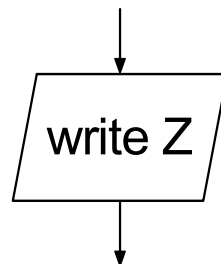
- Se il blocco di istruzioni contengono una sola istruzione le parentesi graffe non sono necessarie (ma voi usatele comunque!)
- Come al solito le istruzioni vanno sempre terminate con un punto e virgola

# Input ed output di dati

- Ogni algoritmo parte da dati in ingresso per produrre dati in uscita (problema computazionale)



.....  
read X  
.....

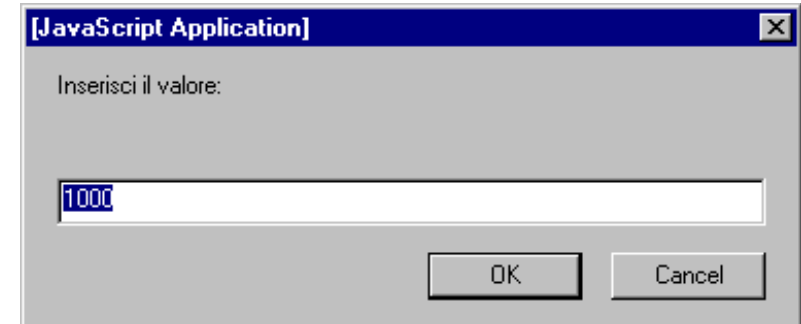


.....  
write Z  
.....

# Controllo dell'input e dell'output

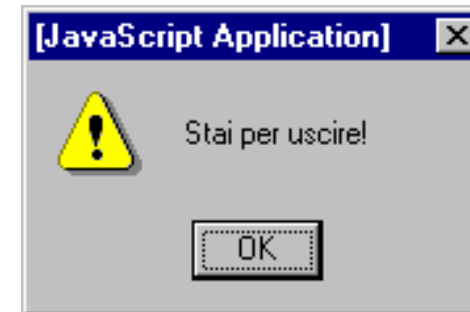
- `v = window.prompt("Inserisci il valore");`

restituisce nella variabile v la stringa inserita nel campo



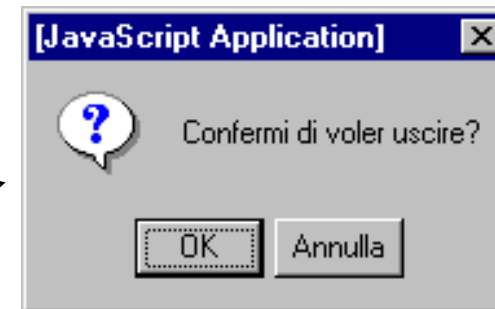
- `window.alert("Stai per uscire!");`

non restituisce nulla



- `c = window.confirm("Confermi di voler uscire?");`

restituisce in c true o false



# Input ed output: tipi di variabili

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript
/* Dichiarazione variabili per termini e somma */
var X;
var Y;
var Z = 0;

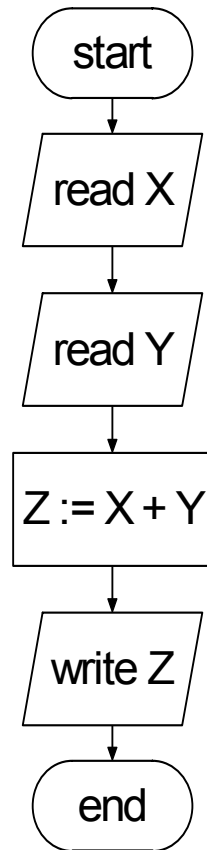
// Leggo il valore del primo termine
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
/*
In questa istruzione calcolo il risultato
*/
Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

- Le funzioni per la gestione dell'input restituiscono sempre stringhe di caratteri
- Per questo, per ottenere un numero ho moltiplicato per 1 (javascript converte automaticamente)
- Le variabili numeriche sono sempre rappresentate come numeri reali
  - Se Z valesse 7 ed io effettuassi  $Z/2$  otterrei 3.5
- L'operatore + sulle stringhe indica la concatenazione

## Esempio: somma di due numeri

- Scrivere l'algoritmo che esegue la somma di due numeri



**start**

read X

read Y

Z := X + Y

write Z

**end**



## Esempio: somma di due numeri

---

- Scrivere l'algoritmo che esegue la somma di due numeri

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: somma di due numeri</TITLE>
</HEAD>
<BODY>

<SCRIPT>
<!-- Inizio script JavaScript

var X;
var Y;
var Z;

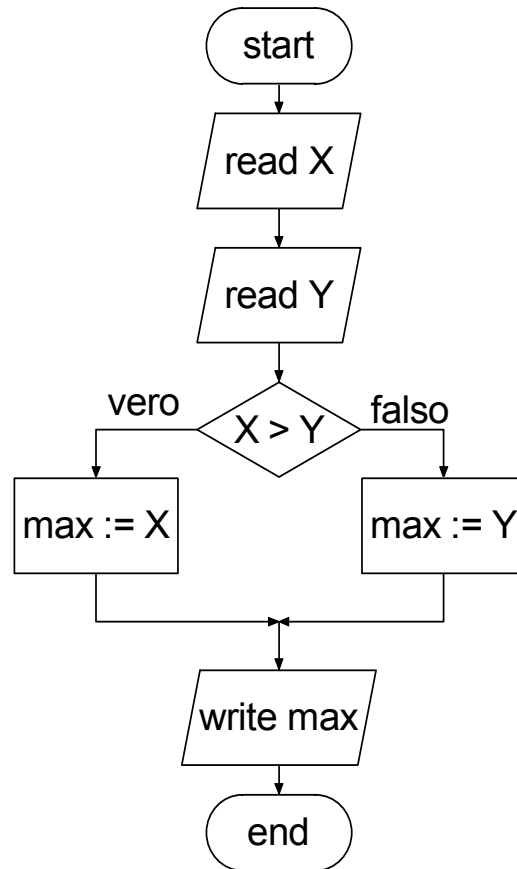
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;

Z = X + Y;

window.alert("Somma = " + Z);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

## Esempio: massimo tra due numeri

Dati due numeri, dire qual è il massimo tra i due.



**start**

read X

read Y

**if** (X > Y) **then**

max := X

**else**

max := Y

**end if**

write max

**end**

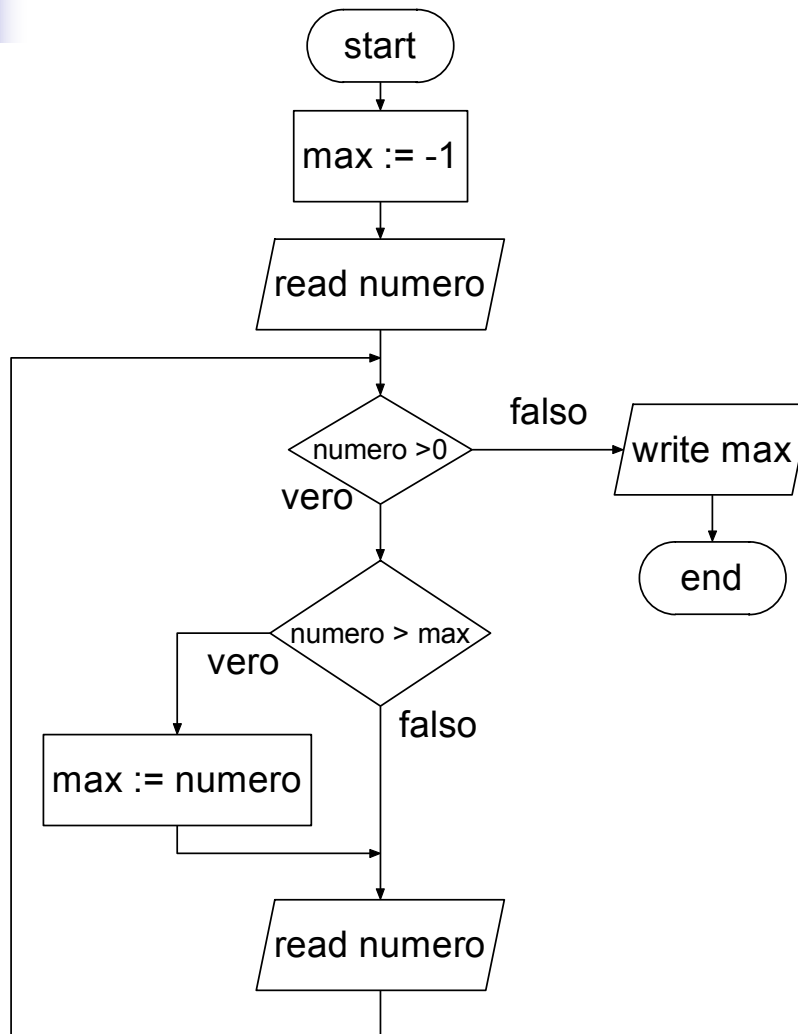


# Esempio: massimo tra due numeri

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: massimo tra due numeri</TITLE>
</HEAD>
<BODY>
<SCRIPT>
<!-- Inizio script JavaScript
var X;
var Y;
var max;
X = window.prompt("Inserisci il primo numero");
Y = window.prompt("Inserisci il secondo numero");
X = X*1; Y=Y*1;
if (X>Y)
{
    max = X;
}
else
{
    max = Y;
}
window.alert("Massimo = " + max);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

# Esercizio: massimo di una sequenza



**start**

max := -1

read numero

**while** (numero > 0)

**if** (numero > max) **then**

    max := numero

**end if**

  read numero

**end while**

write max

**end**



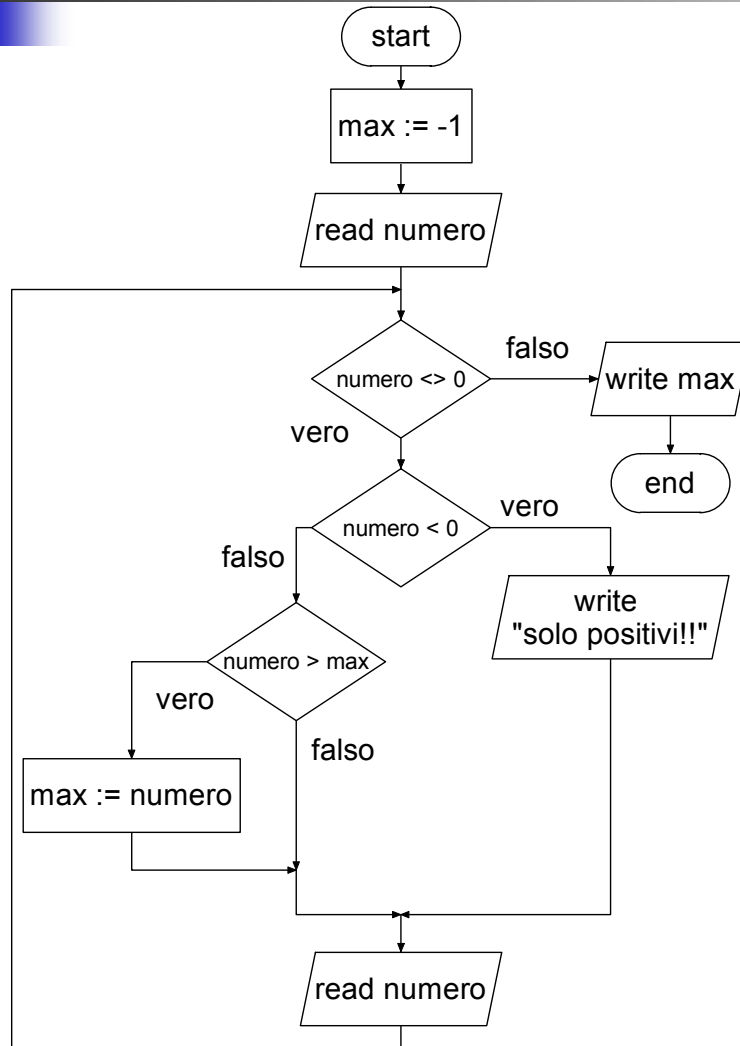
# Esercizio: massimo di una sequenza

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: massimo tra due numeri</TITLE>
</HEAD>
<BODY>
<SCRIPT>
<!-- Inizio script JavaScript
var numero;
var max;

max = -1;
numero = window.prompt("Inserisci un numero");
numero = numero * 1;
while (numero > 0)
{
  if (numero > max)
  {
    max = numero;
  }
  numero = window.prompt("Inserisci un numero");
  numero = numero * 1;
}
window.alert("Massimo = " + max);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

# Esercizio: massimo di una sequenza



**start**

max := -1

read numero

**while** (numero <> 0)

**if** (numero < 0) **then**

write "solo positivi!!"

**else**

**if** (numero > max) **then**

max := numero

**end if**

**end if**

read numero

**end while**

write max

**end**

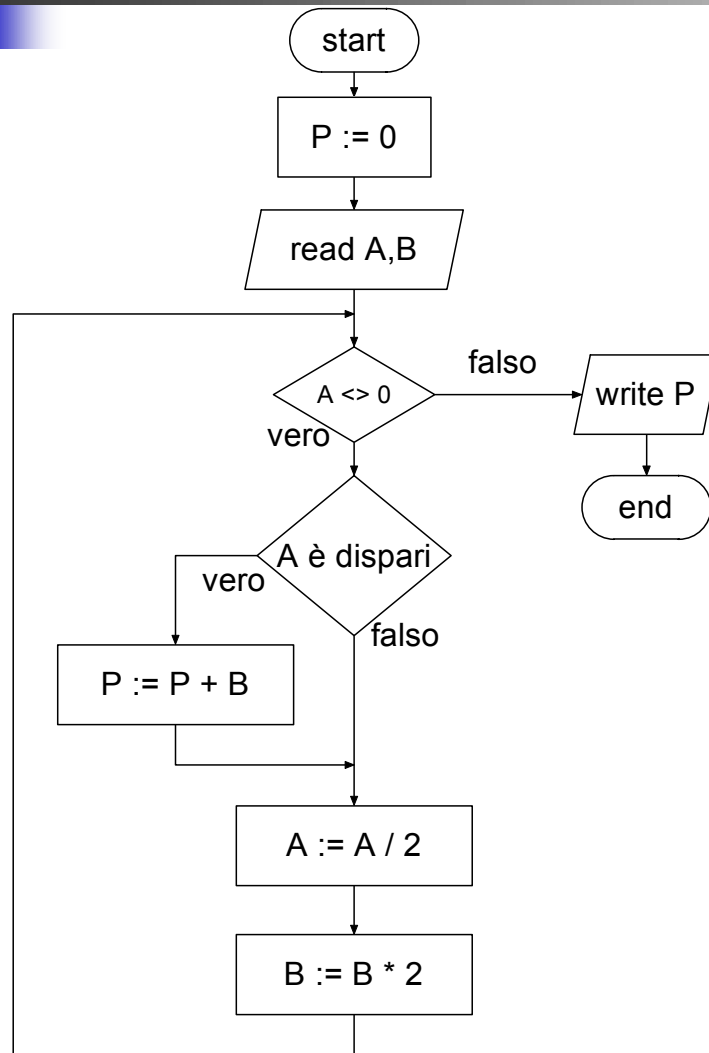


# Esercizio: massimo di una sequenza

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: massimo di una sequenza</TITLE>
</HEAD>
<BODY>
<SCRIPT>
<!-- Inizio script JavaScript
var numero;
var max;
max = -1;
numero = window.prompt("Inserisci un numero");
numero = numero * 1;
while (numero != 0)
{
  if (numero < 0)
  {
    window.alert("Solo numeri positivi!");
  }
  else
  {
    if (numero > max)
    {
      max = numero;
    }
  }
  numero = window.prompt("Inserisci un numero");
  numero = numero * 1;
}
window.alert("Massimo = " + max);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```

# Esercizio: moltiplicazione



**start**

$P := 0$

read  $A, B$

**while** ( $A \neq 0$ )

**if** ( $A$  è dispari) **then**

$P := P + B$

**end if**

$A := A / 2$

$B := B * 2$

**end while**

write  $P$

**end**



# Esercizio: moltiplicazione

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: moltiplicazione</TITLE>
</HEAD>
<BODY>
<SCRIPT>
<!-- Inizio script JavaScript
var P;
var A;
var B;
A = window.prompt("Inserisci il primo fattore");
A = A * 1;
B = window.prompt("Inserisci il secondo fattore");
B = B * 1;
while (A != 0)
{
  if (A%2 == 1)
  {
    P = P + B;
  }
  A = A / 2;
  B = B * 2;
}
window.alert("Prodotto = " + P);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```



# Esercizio: moltiplicazione

---

```
<HTML>
<HEAD>
<TITLE>ESEMPIO: moltiplicazione</TITLE>
</HEAD>
<BODY>
<SCRIPT>
<!-- Inizio script JavaScript
var P;
var A;
var B;
P = 0;
A = window.prompt("Inserisci il primo fattore");
A = A * 1;
B = window.prompt("Inserisci il secondo fattore");
B = B * 1;
while (A != 0)
{
  if (A%2 == 1)
  {
    P = P + B;
  }
  A = Math.floor(A / 2);
  B = B * 2;
}
window.alert("Prodotto = " + P);
// Fine script -->
</SCRIPT>
</BODY>
</HTML>
```