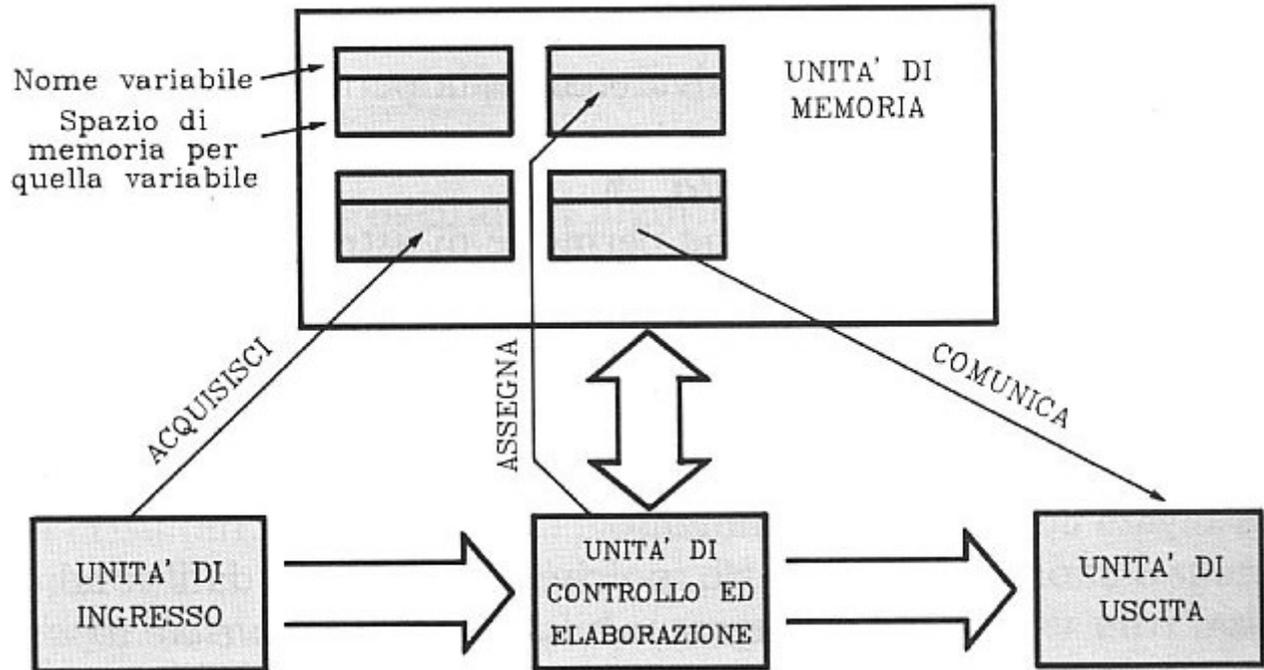


Programmazione: variabili e operatori

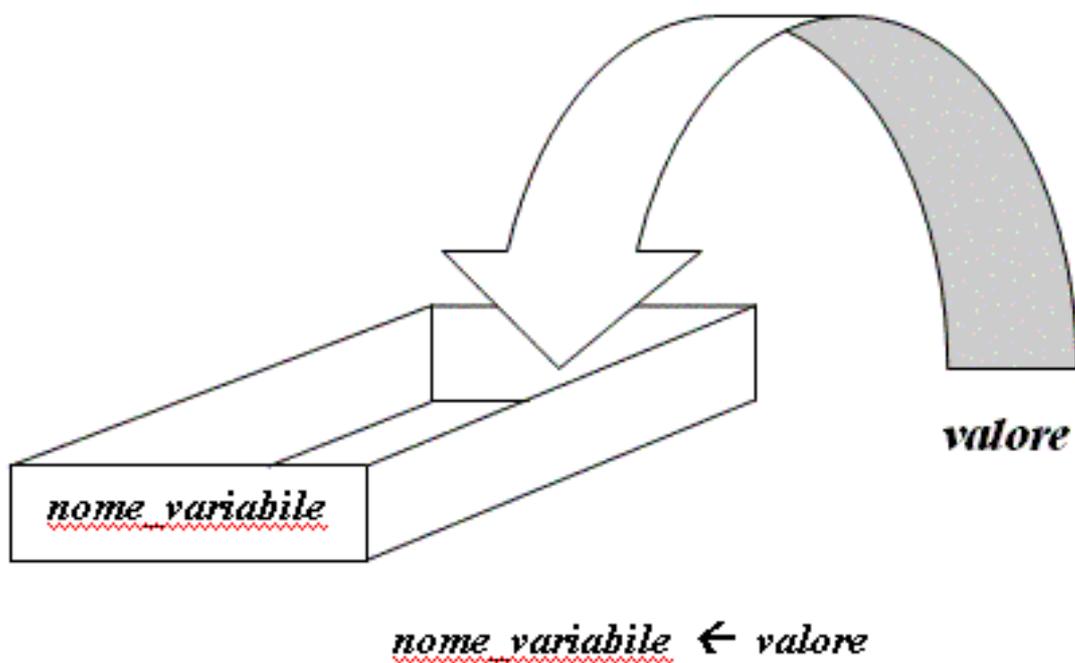
Variabili

Una **variabile** è un'area di memoria (RAM) individuata da un **nome** detto identificativo. Tale nome esprime l'*indirizzo* dove sarà posto il valore che si **assegnerà** alla variabile.



Una variabile si può pensare come una scatola aperta dove si possono introdurre dati.

Operazione di *assegnamento* o *assegnazione*



Variabili in Scratch

Nel *frame* di sinistra, nella categoria **Variabili** si seleziona **Nuova variabile** e si sceglie il nome

Si potrà definire la variabile globale o locale cioè visibile da tutti gli sprites o solo dallo sprite attuale:

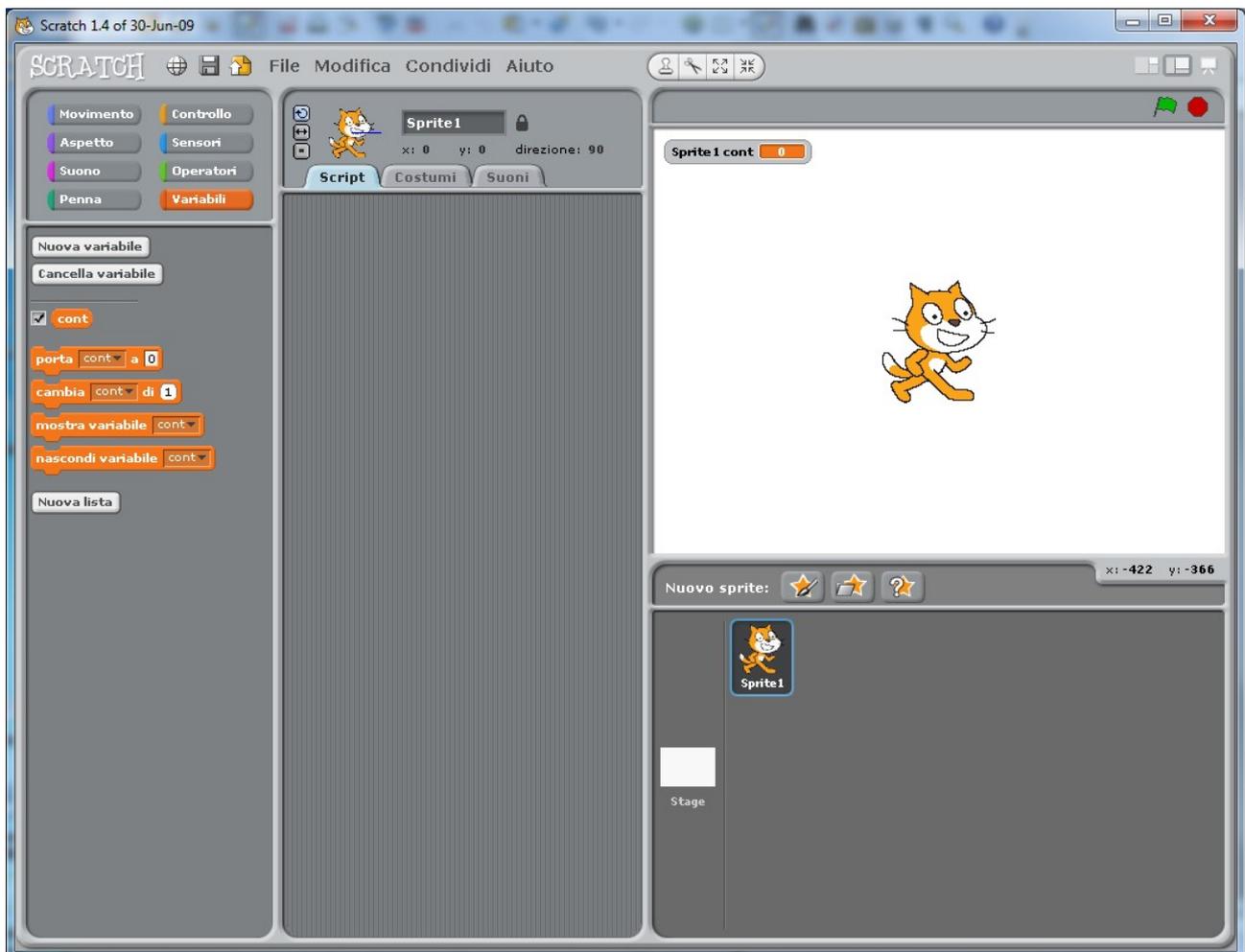


Si consiglia la scelta di variabili **locali**.

Scelto il nome - ad esempio **cont** - e premuto OK



l'interfaccia si presenta come in figura:

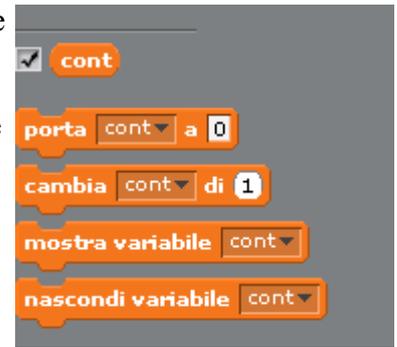


dove, nel *frame* di sinistra è comparso, smarcato, il nome della variabile

e tessere per modificarne il valore

o la visualizzazione

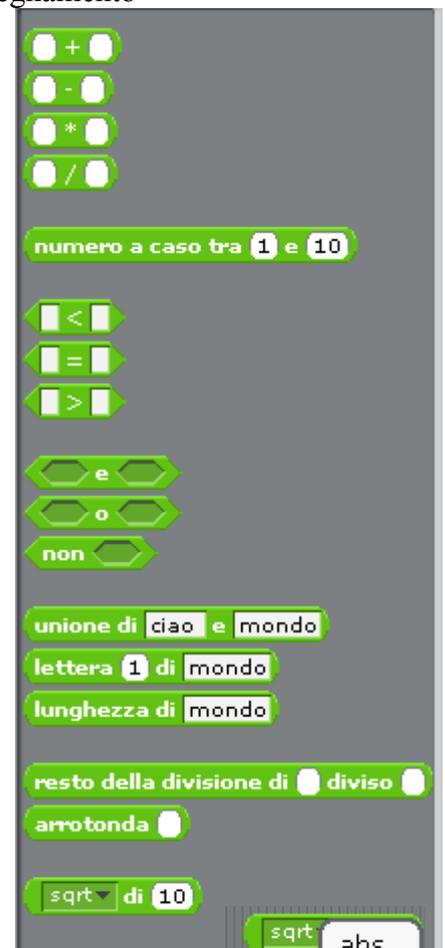
Di default è mostrata nello stage:



Operatori in Scratch

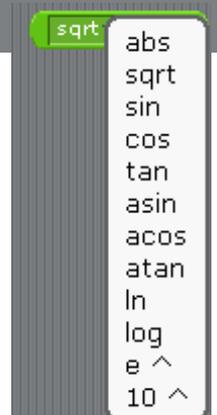
Una volta definita una variabile, si può usare per impostare formule generali in espressioni che coinvolgono diversi **Operatori** oltre alla fondale operazione di assegnamento

- operatori aritmetici
- funzione di generazione di numeri *casuali*
- operatori relazionali
- operatori logici
- funzioni per concatenare, estrarre singoli caratteri da una frase, calcolare la lunghezza (numero di caratteri) di una frase
- operatore resto
- operazioni di arrotondamento
- funzioni matematiche e trigonometriche

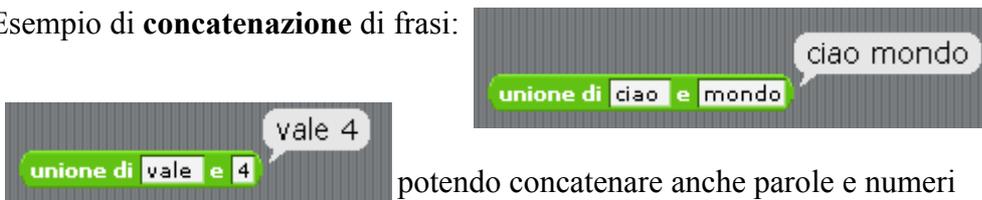


Esempio di **operatore relazionale**:  impostato trascinando nell'area centrale Script l'operatore relazionale minore, trascinando all'interno del primo operando la variabile 

spostandosi tra i comandi della categoria **Operatori** e **Variabili** e digitando come secondo operando il valore 4



Esempio di **concatenazione** di frasi:



Esempio di **estrazione pseudocasuale** di un numero tra 1 e 2.8:



Esempio di **estrazione del terzo carattere** dalla parola mondo:



Esempio di **calcolo della lunghezza** (numero di caratteri) della parola mondo:



Esempio di operatore **resto** tra numeri **non divisibili**



tra numeri **divisibili**



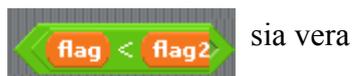
Esempio di **arrotondamento** per eccesso
oppure per difetto



Nell'uso di **operatori logici** l'assenza di operando è per default interpretata come **falso**
infatti la **negazione** di falso è vero



ed impostando le due variabili flag e flag2 in modo che sia
vera la condizione



l'operazione **AND** fornisce risultato falso
(vero solo se entrambi sono veri)

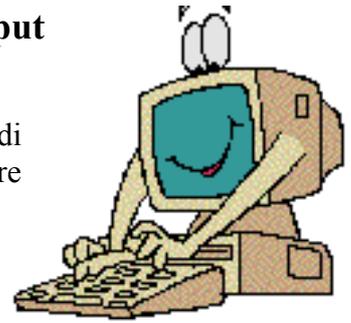
l'operazione **OR** fornisce risultato vero
(è sufficiente che uno sia vero)



Programmazione: operazioni di input e output

È possibile far interagire lo sprite con l'utente; abbiamo già visto esempi di comunicazione (**operazioni di uscita**) da parte dello sprite con uso di tessere **dire** potendo concatenare frasi e risultati.

Per poter consentire di chiedere informazioni all'utente ed usarle per fornire risposte, si considerino i controlli di tipo 

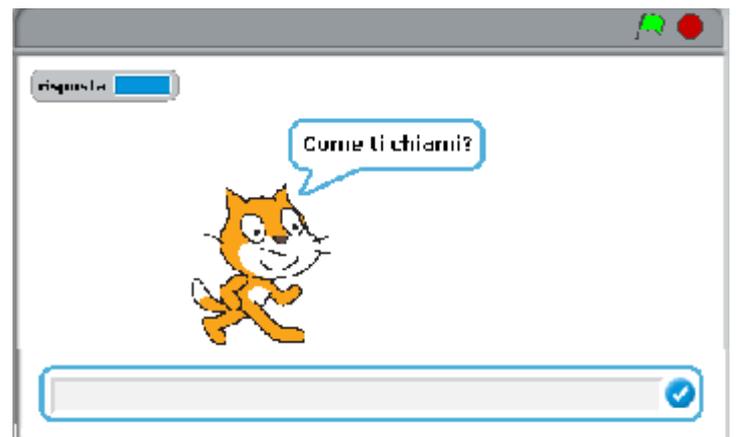


Esiste la tessera  personalizzabile nella richiesta all'utente

e la tessera  che può essere smarcata  volendone anche direttamente nello stage la visualizzazione ed è block di tipo **reporter** cioè può essere inserito in altro come nello **script** esemplificato:



Al click sulla bandiera verde, l'effetto è quello visualizzato:



inserito nella casella di testo un nome o frase

il **click sul pulsante di spunta** interrompe l'attesa programmata, e concatenandola ad un saluto



la risposta dell'utente viene sia visualizzata nello stage sia usata per personalizzare il saluto – la comunicazione dello sprite.



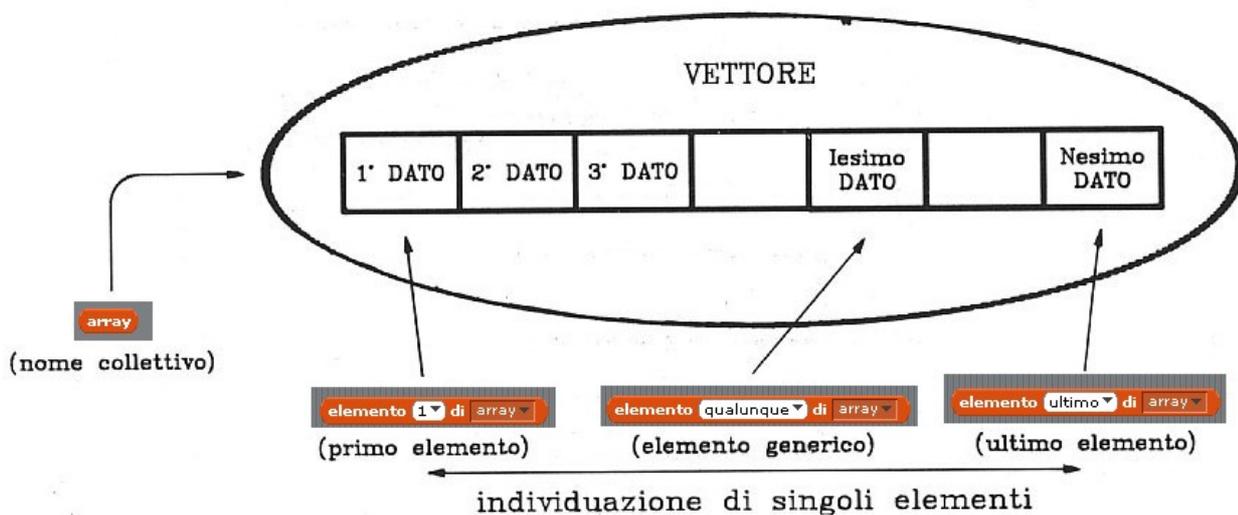
Variabili strutturate: array

Una matrice (**array**) è un *insieme ordinato e omogeneo* di dati: è una variabile che contiene uno o più valori in sequenza ordinata. Questi valori si chiamano "elementi" e devono avere lo stesso tipo di dato. I singoli elementi sono individuati tramite indici.

Variabile di tipo strutturato, dunque, in quanto l'indice introduce delle relazioni all'interno dell'insieme, stabilendo un nuovo ordine che non è necessariamente quello dei valori degli elementi componenti. Il tipo dell'indice deve essere scalare, non reale perché, per ogni elemento, deve essere possibile definire "il precedente" ed "il seguente".

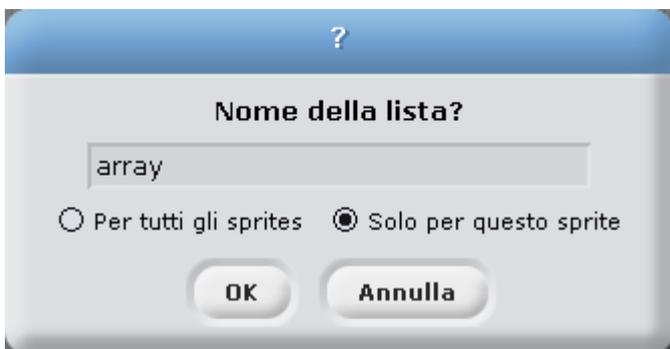
Servono dunque per usare un **unico nome collettivo** per riferirsi a molti elementi.

Il *nome* dell'insieme è l'*indirizzo* della locazione di memoria dove è memorizzato il valore del primo elemento; gli altri elementi della lista occuperanno locazioni di memoria consecutive.



Array in Scratch

Nel *frame* di sinistra, nella categoria **Variabili** si selezioni **Nuova lista** e si scelga il nome



Si potrà definire, come nel caso di variabile elementare, globale o locale cioè **lista** visibile da tutti gli sprites o solo dallo sprite attuale. Si consiglia la scelta di **liste locali**.

Nel *frame* di sinistra è comparso, smarcato, il nome della lista e le tessere relative.



Operazioni su array in Scratch

Tipiche operazioni su array i cui elementi sono valori numerici (per dimensioni dell'array non nulle)

- **Resettare** cioè porre a zero il valore di tutti gli elementi



- **Inizializzazione casuale** dei valori



- **Inizializzazione** dei valori scelti dall'utente



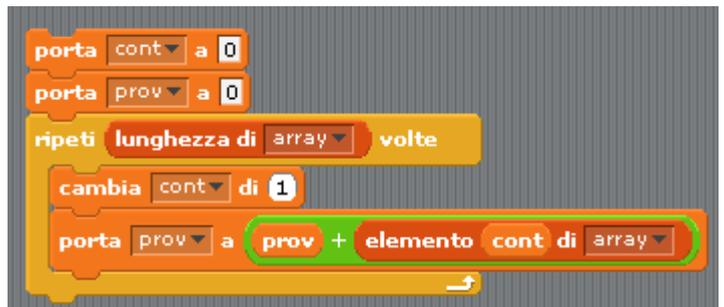
Con uso di **variabili elementari**



- **Sommare (iterativamente)** il valore di tutti gli elementi:

```
{
  cont ← 0
  prov ← 0
  mentre (ci sono elementi)
    cont ← cont + 1
    prov ← prov + el_attuale
}
```

ora nella variabile **prov** c'è la **somma**
e nella variabile **cont** il numero di elementi



- **Calcolare la media** dopo averne calcolato la somma :
 $prov_valore\ medio \leftarrow prov_somma / cont$



- Ricercare il valore **massimo** tra i valori dell'array in modo **efficiente**

```
{
  prov ← el_primo
  cont ← 1
  mentre (Nvolte <= lunghezza array - 1)
    cont ← cont + 1
    se (el_attuale > prov)
      prov ← el_attuale
}
```

ora nella variabile **prov** c'è il **massimo**
e nella variabile **cont** il numero di elementi



- Ricerca il valore **minimo** tra i valori dell'array in modo *efficiente*

```

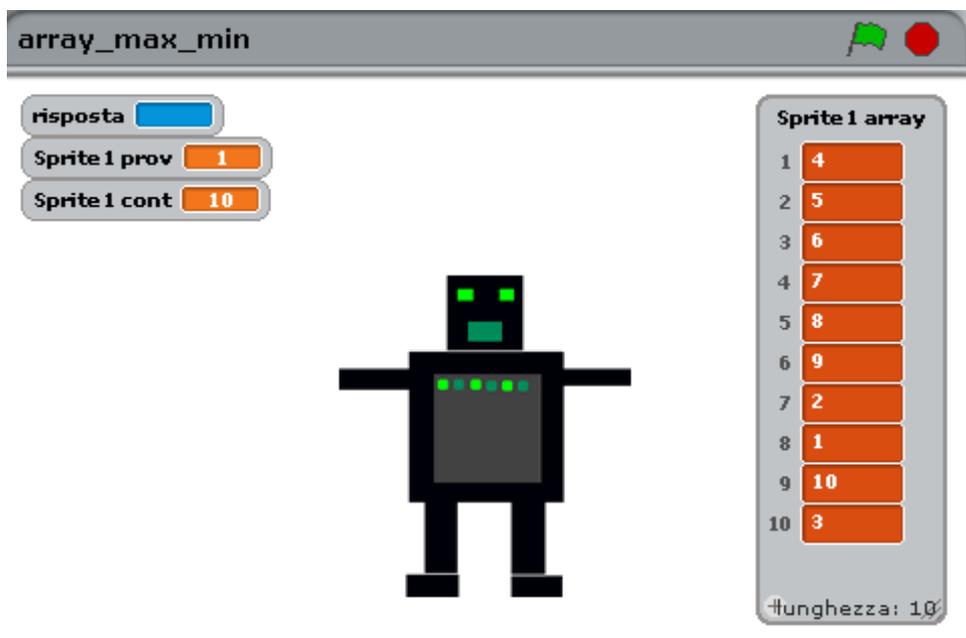
{
  prov ← el_primo
  cont ← 1
  mentre (Nvolte ≤ lunghezza array - 1)
    cont ← cont + 1
    se (el_attuale < prov)
      prov ← el_attuale
}

```

ora nella variabile **prov** c'è il **minimo**
e nella variabile **cont** il numero di elementi



Effetto dopo la creazione di un array di 10 elementi, l'inizializzazione con numeri interi digitati dall'utente ed il calcolo del **minimo**

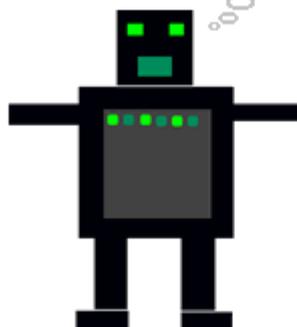


Di seguito un'implementazione **errata** nella ricerca del massimo se i numeri sono tutti negativi

fornisce come **massimo** il **valore zero**



Uhm.. almeno un numero positivo



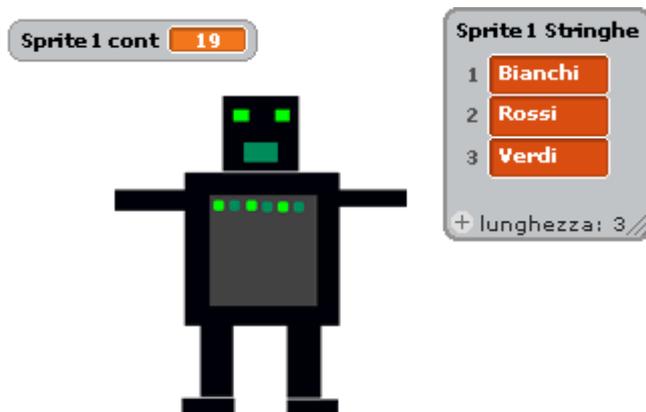
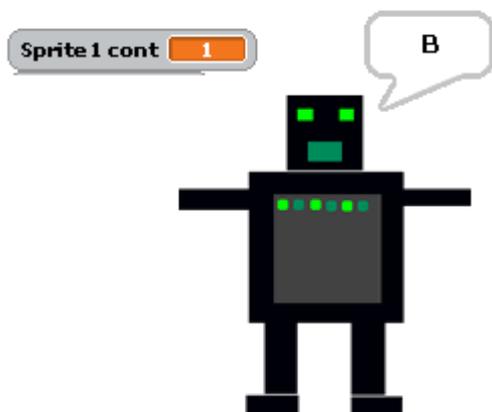
si noti anche la *minor efficienza*:
si compiono tanti confronti quanti sono gli elementi dell'array

Operazioni su stringhe in Scratch

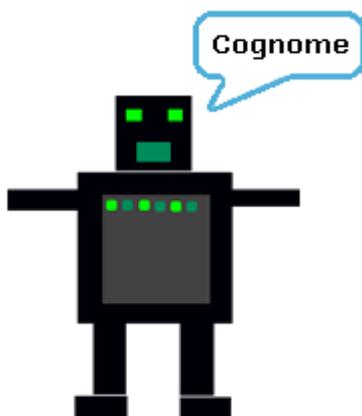
- Inizializzare l'array con *oggetti stringhe*



- Poter elaborare il singolo carattere verificando che le stringhe¹ sono accodate inserendo tra l'una e la successiva un carattere di separazione che non viene visualizzato nella comunicazione dello sprite ma risulta dal conteggio dei singoli caratteri

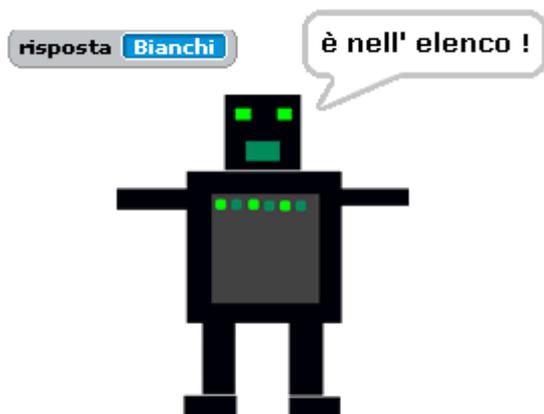


- Poter elaborare il singolo *oggetto stringa*



esempio di *ricerca e inserimento*

possibile *cancellazione*



¹ Stringa: sequenza finita di elementi sintatticamente concatenati in una frase.