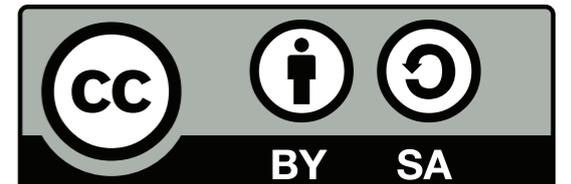


Pensiero computazionale

Lezione 10

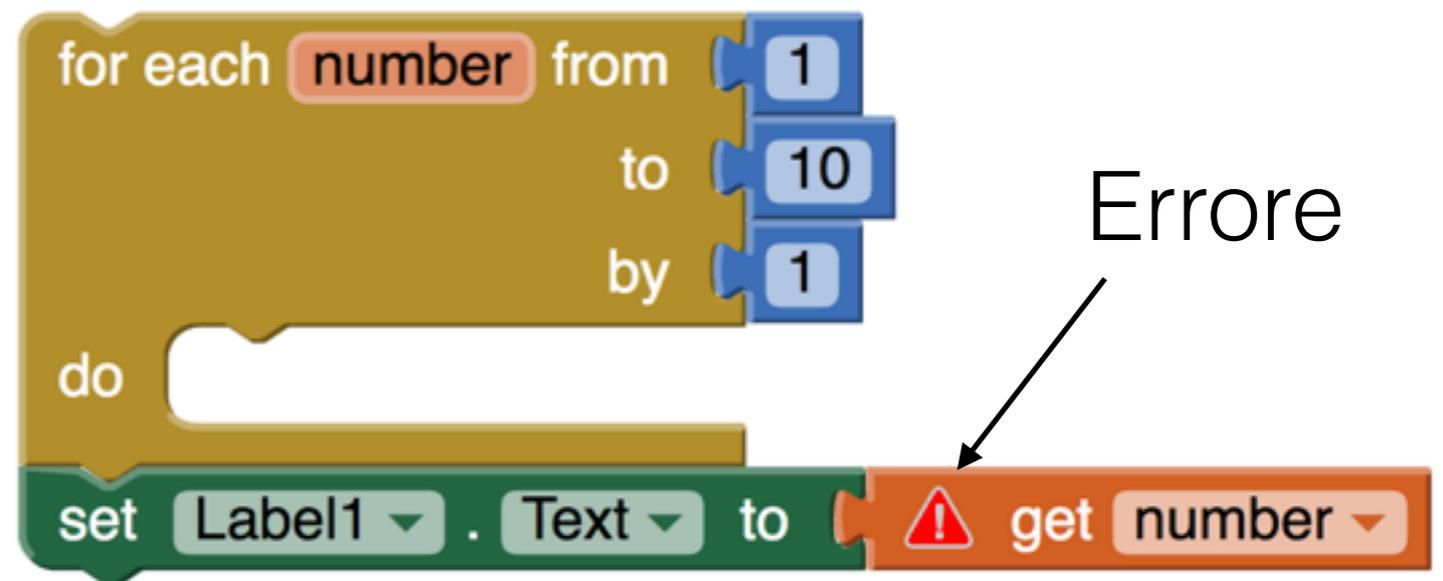
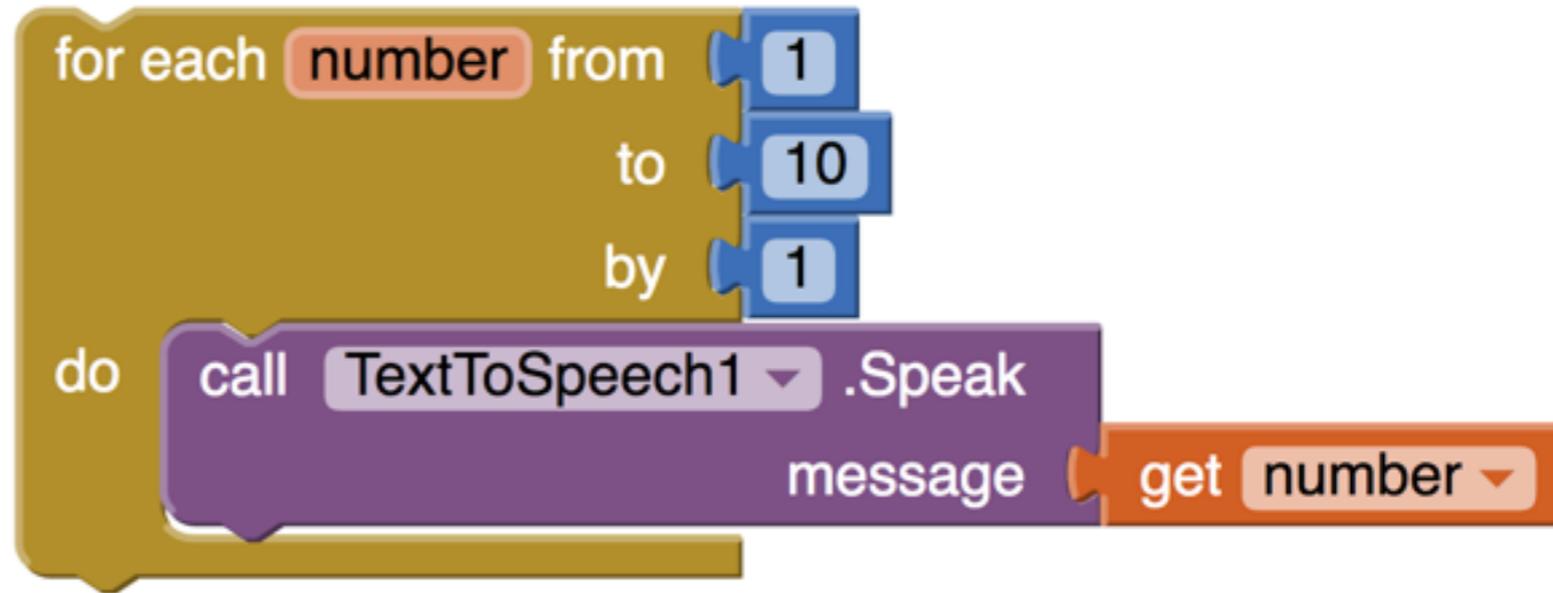
MIT AppInventor

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



Ripetizioni

- Dov'è “Ripeti 10 volte”?
- Cosa succede:
 - La variabile *number* assume i valori 1,2,.....,9,10
 - “by” è l'incremento
 - A quel punto il ciclo termina
 - Potete utilizzarla oppure no
- Nota:
 - La variabile *number* esiste solo all'interno del ciclo



Codice equivalente

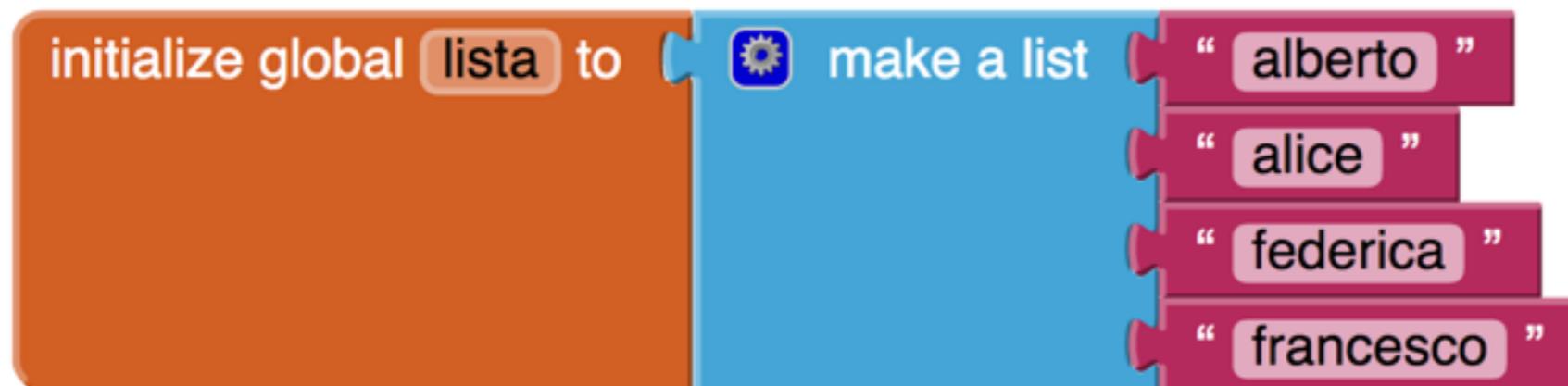
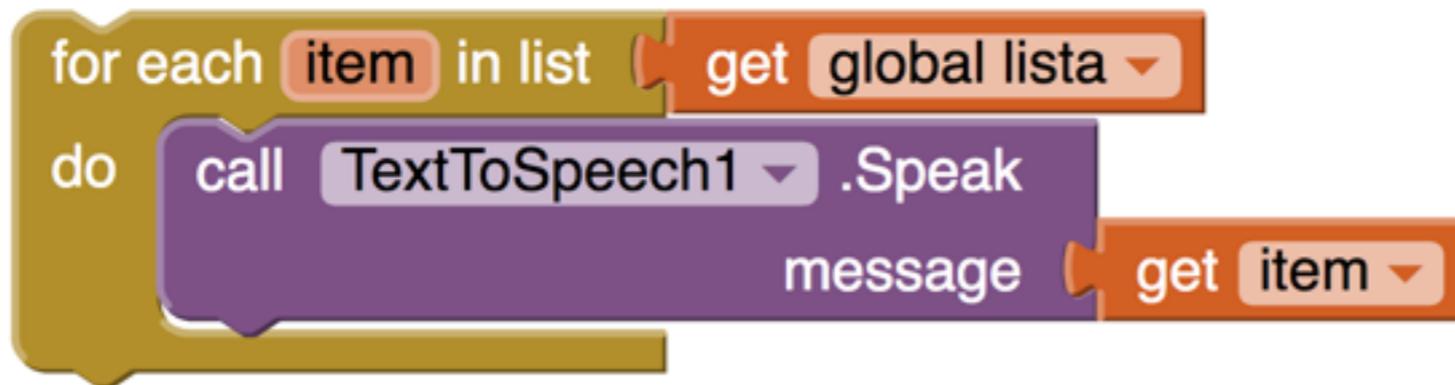
- Esempio: **python**

```
for number in range(1,11):  
    print(number)  
print("Finito!")
```

- Alcuni commenti:
 - range(1,11) restituisce i numeri da 1 a 10
 - utilizziamo l'*indentazione* per stabilire quali comandi fanno parte del ciclo e quali no
 - In altri linguaggi si fa uso delle parentesi graffe

Ripetizioni

- E' possibile scorrere tutti gli elementi di una lista
 - La variabile *item* assume i valori "alberto", "alice", "federica", "francesco"
 - Come prima, esiste solo all'interno del ciclo



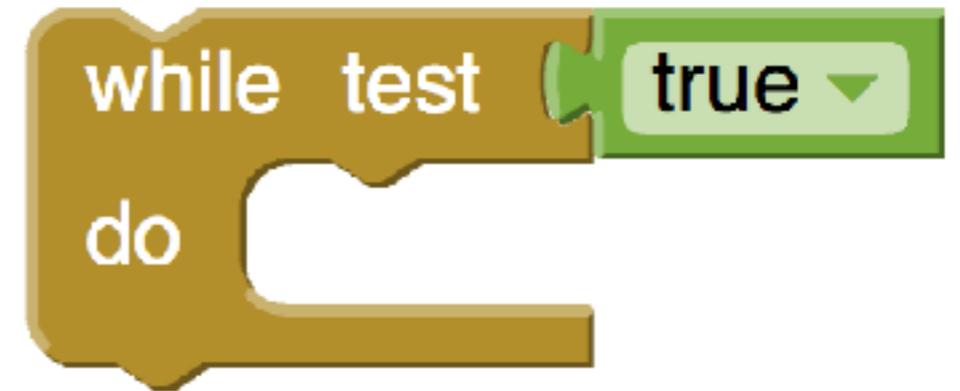
Codice equivalente

- Esempio: **python**

```
lista=["alberto", "alice", "federica", "francesco"]  
for item in lista:  
    print(item)  
print("Finito!")
```

Ripetizioni

- Il ciclo while viene eseguito fintanto che la condizione associata è vera
- Ripeti per sempre:
 - Attenzione, probabile consumo batteria
- Cerca un elemento nella lista



Codice equivalente

- Esempio: **python**

```
lista=["alberto", "alice", "federica", "francesco"]  
indice = 0  
while lista[indice] != "federica":  
    indice = indice+1  
print(indice)
```

Stampa:

2

Liste

- E' possibile creare liste vuote

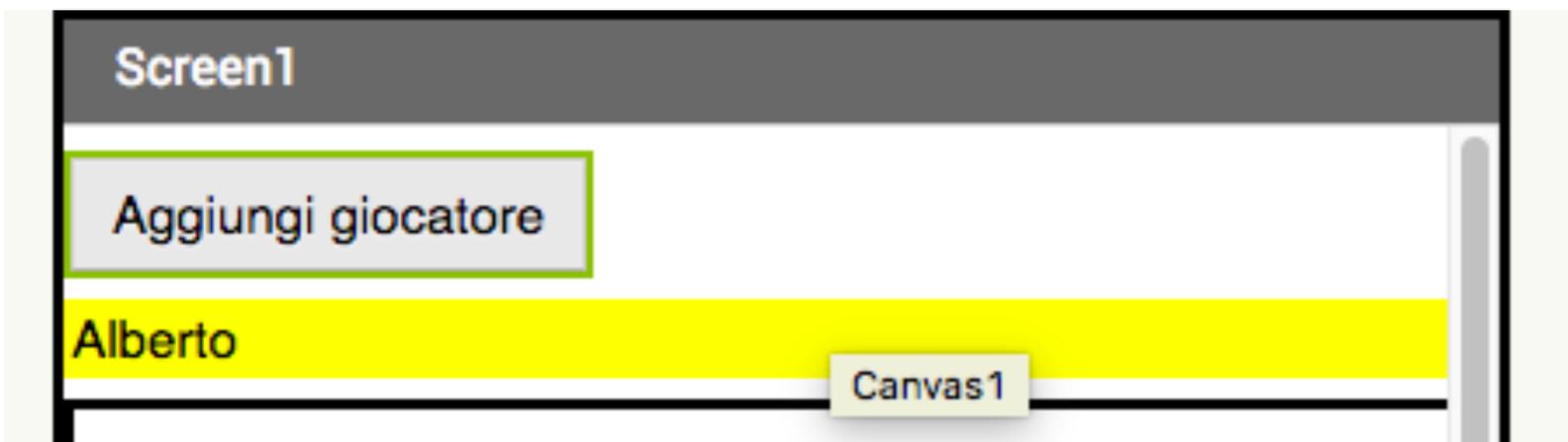


initialize global `giocatori` to `create empty list`

- Aggiungere valori dinamicamente



```
when Button1 .Click
do
  add items to list list
  item
  get global giocatori
  Label1 . Text
```

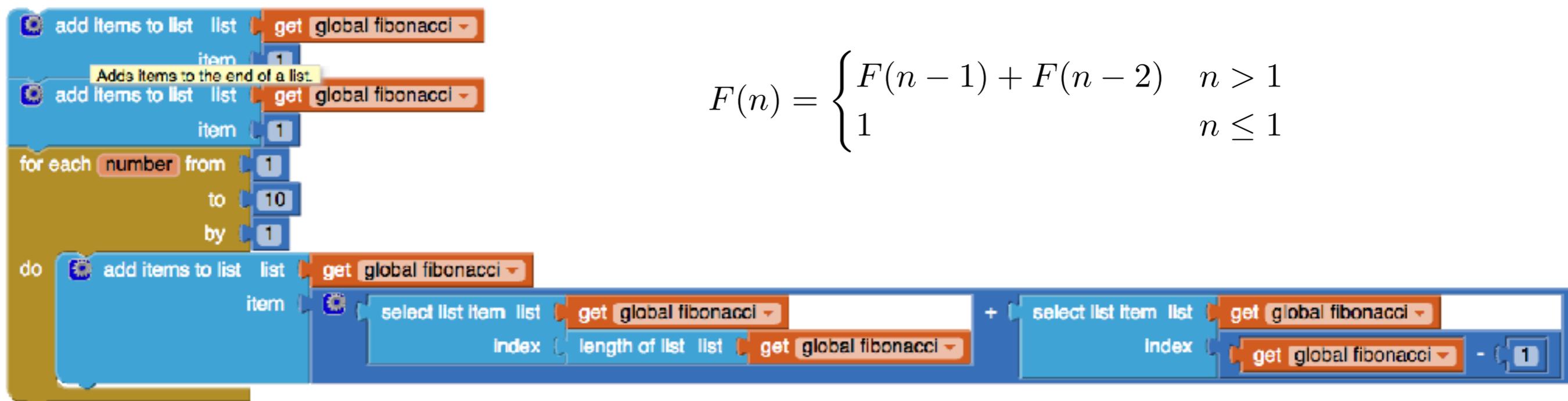


Liste

- E' possibile creare liste vuote

initialize global `fibonacci` to 

- Aggiungere valori dinamicamente



The image shows a Scratch script for calculating the Fibonacci sequence. It starts with two 'add items to list' blocks for the initial values 1 and 1. A 'for each' loop from 1 to 10 by 1 contains a 'do' block. Inside the 'do' block, an 'add items to list' block uses a 'select list item' block to get the value at the current index from the 'global fibonacci' list. The 'select list item' block is configured with 'list' as 'global fibonacci' and 'index' as 'length of list - 1'. The 'add items to list' block then adds the sum of the two previous items to the list.

$$F(n) = \begin{cases} F(n-1) + F(n-2) & n > 1 \\ 1 & n \leq 1 \end{cases}$$

Codice equivalente

- Esempio: **python**

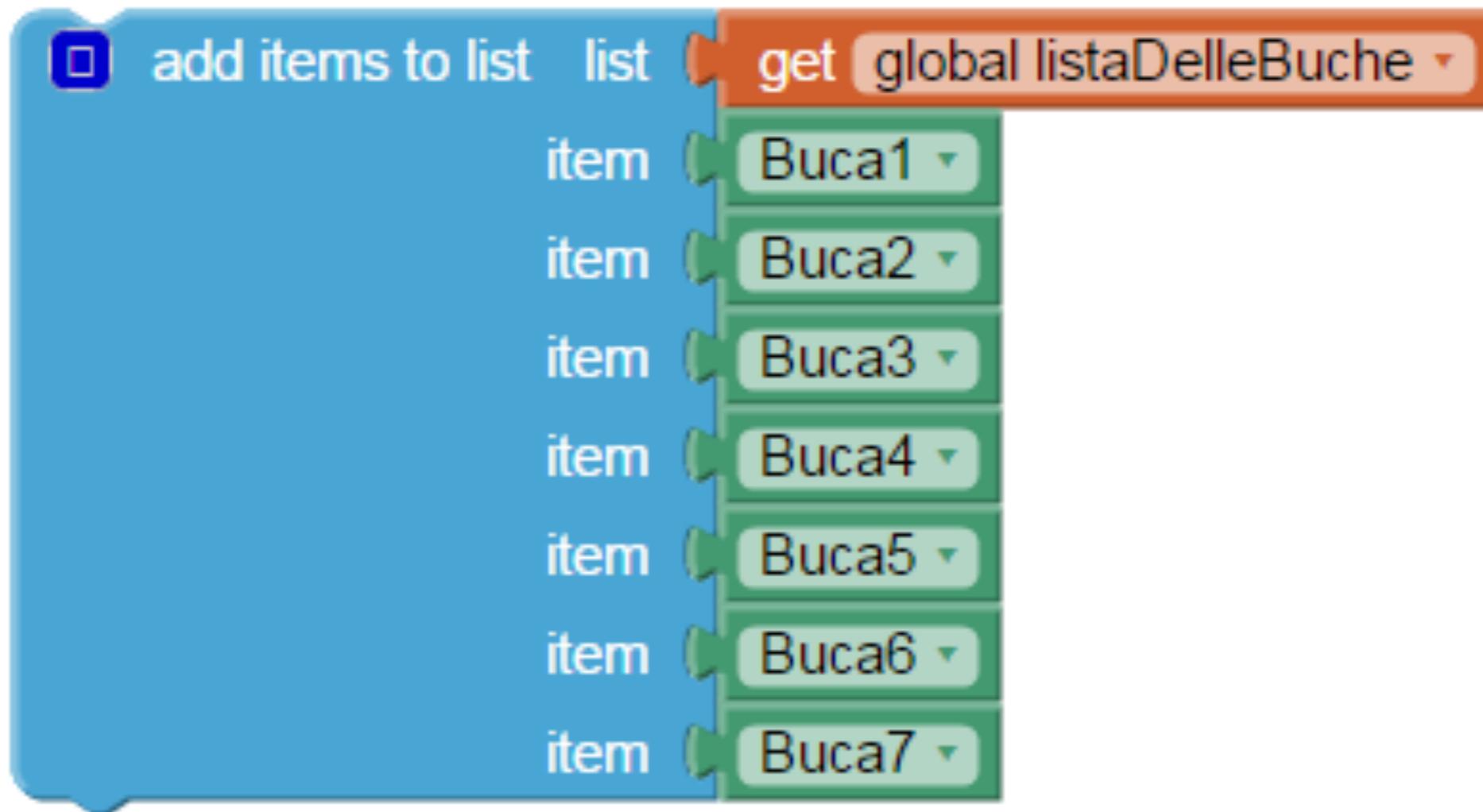
```
L=[1,1]
for i in range(10):
    L.append(L[len(L)-1] + L[len(L)-2])
print(L)
```

Stampa:

```
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144]
```

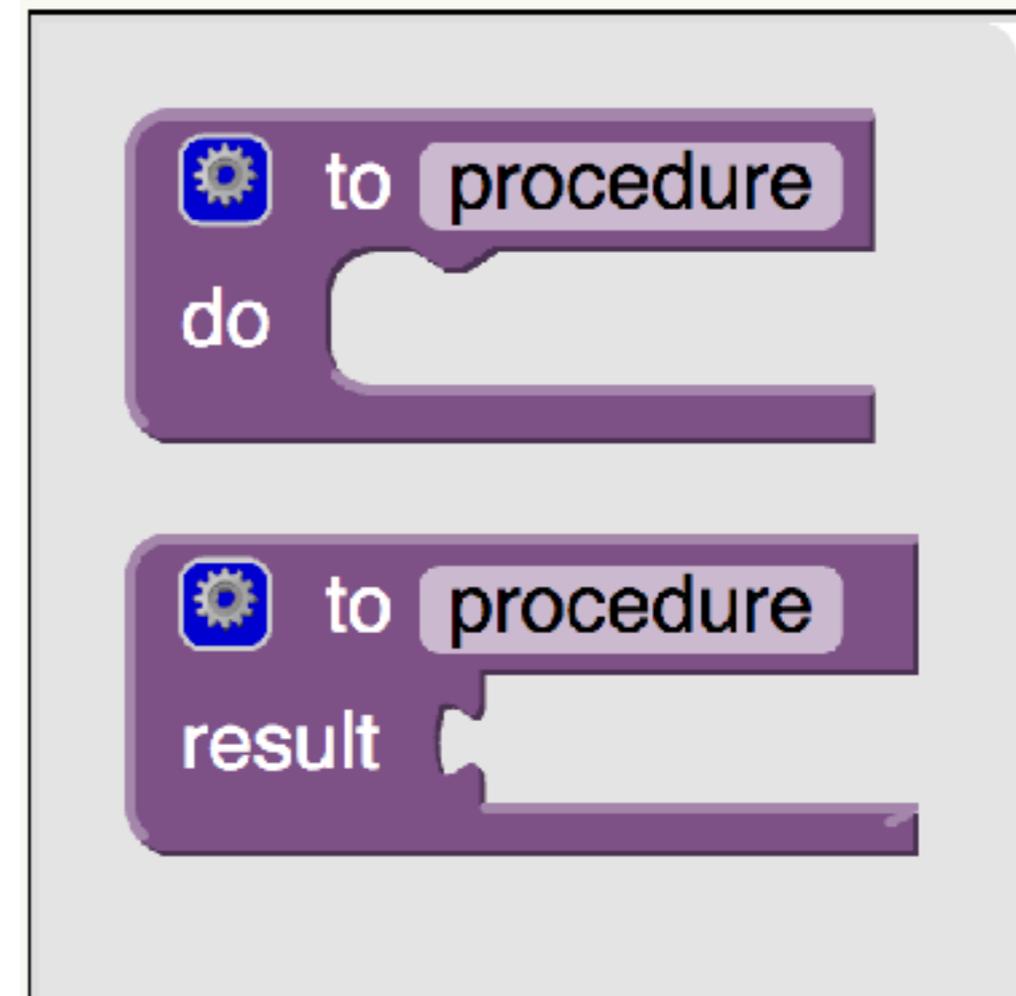
Contenuto delle liste

- Le liste possono contenere qualunque oggetto



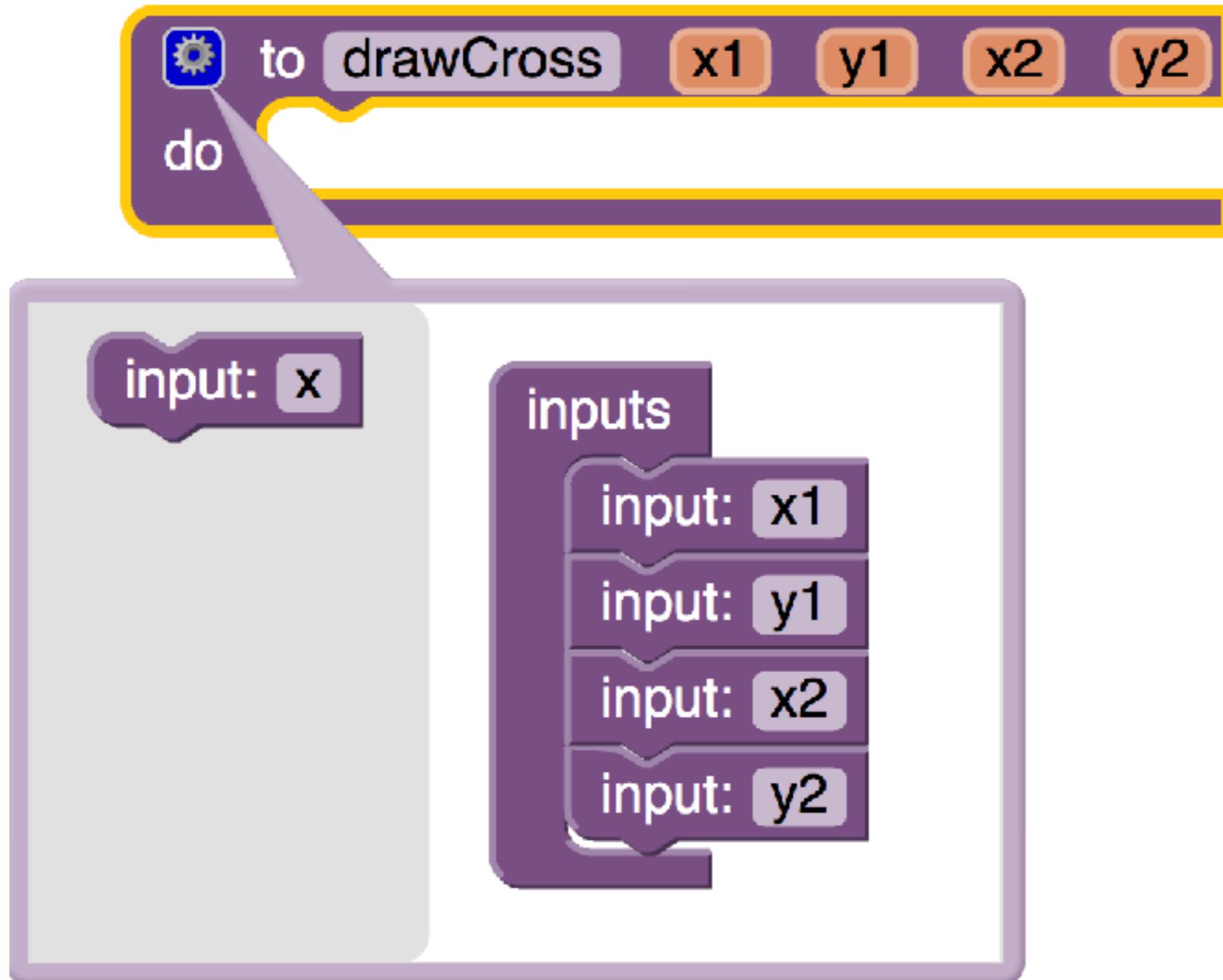
Procedure

- Ci sono due tipi di procedure
 - Quelle che eseguono delle azioni, senza restituire un risultato
 - Quelle che eseguono delle azioni e restituiscono un risultato (**funzioni**)
- Simile alla funzionalità “Nuovi blocchi” di Scratch



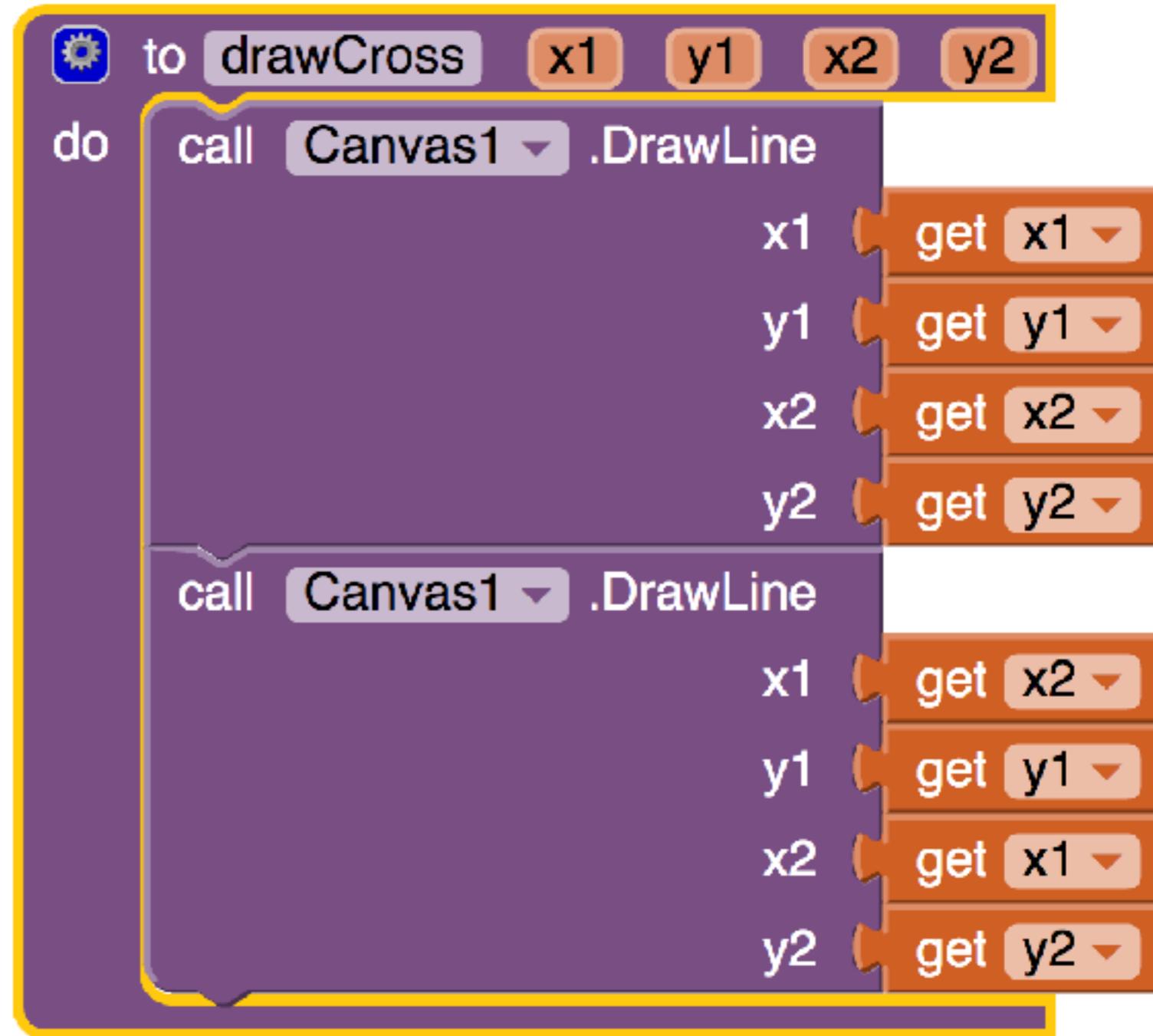
Parametri

- E' possibile aggiungere parametri cliccando sul simbolo *



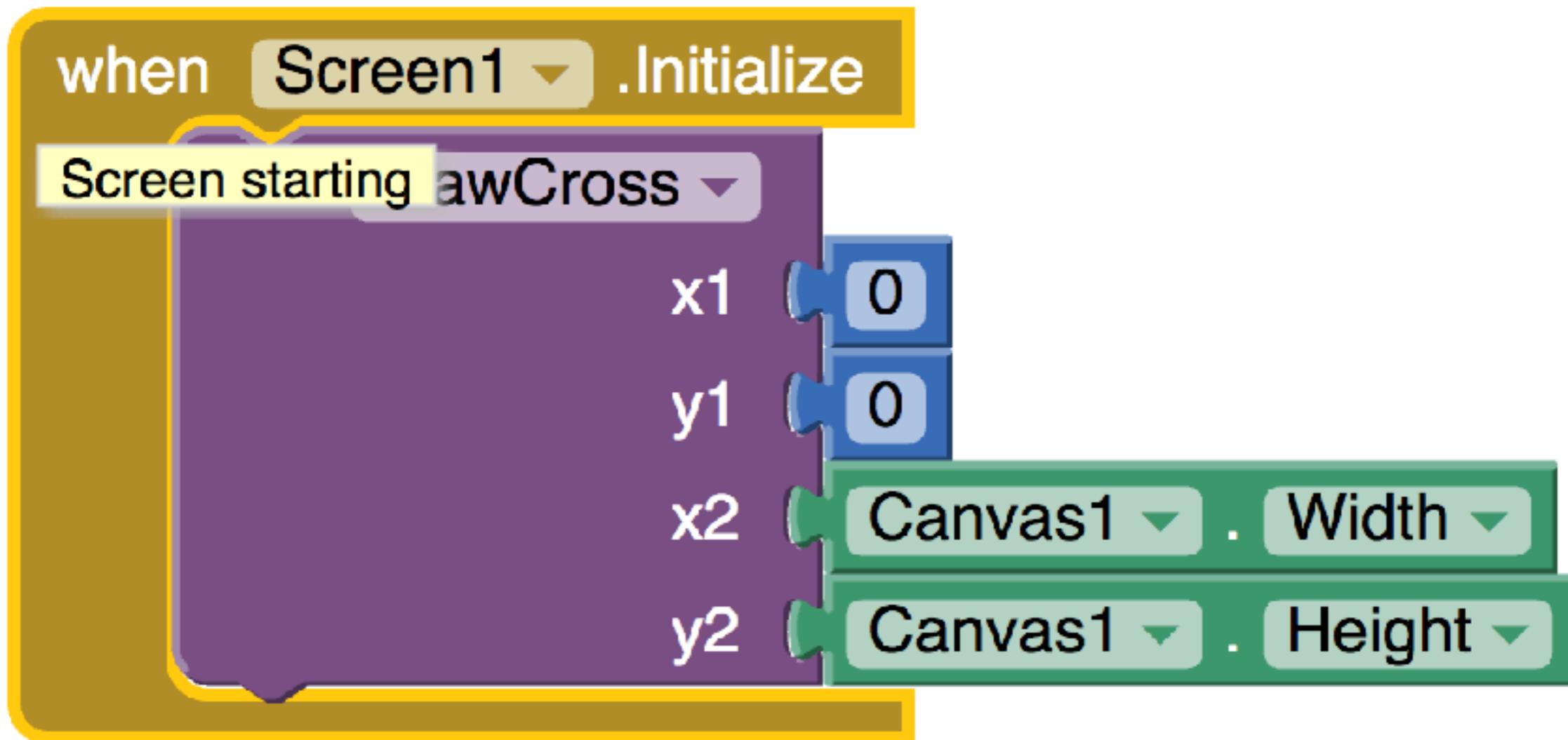
Parametri

- Una volta creata una procedura, è possibile inserire blocchi all'interno
- Questa procedura disegna una croce su un canvas



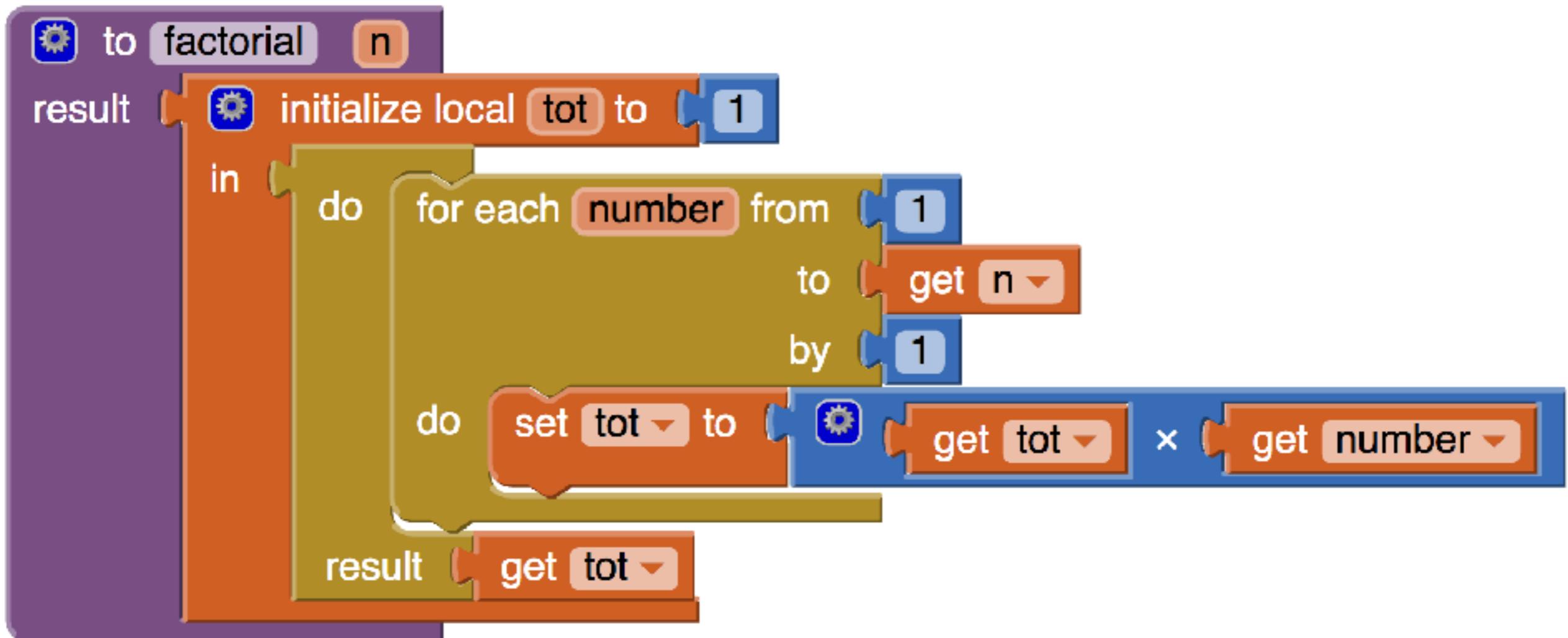
Parametri

- Una volta completata una procedura, è possibile invocarla
- Questi blocchi disegnano una croce su tutto il canvas



Funzioni

- Una funzione può restituire un valore
- Questa funzione calcola il fattoriale di un numero
- Sono presenti due blocchi particolari:
 - variabile locale (initialize local)
 - un blocco che restituisce un valore



Codice equivalente

- Esempio: **python**

```
def factorial(n):  
    tot = 1  
    for i in range(2,n+1):  
        tot = tot*i  
    return tot  
  
print(factorial(10))
```

Stampa:

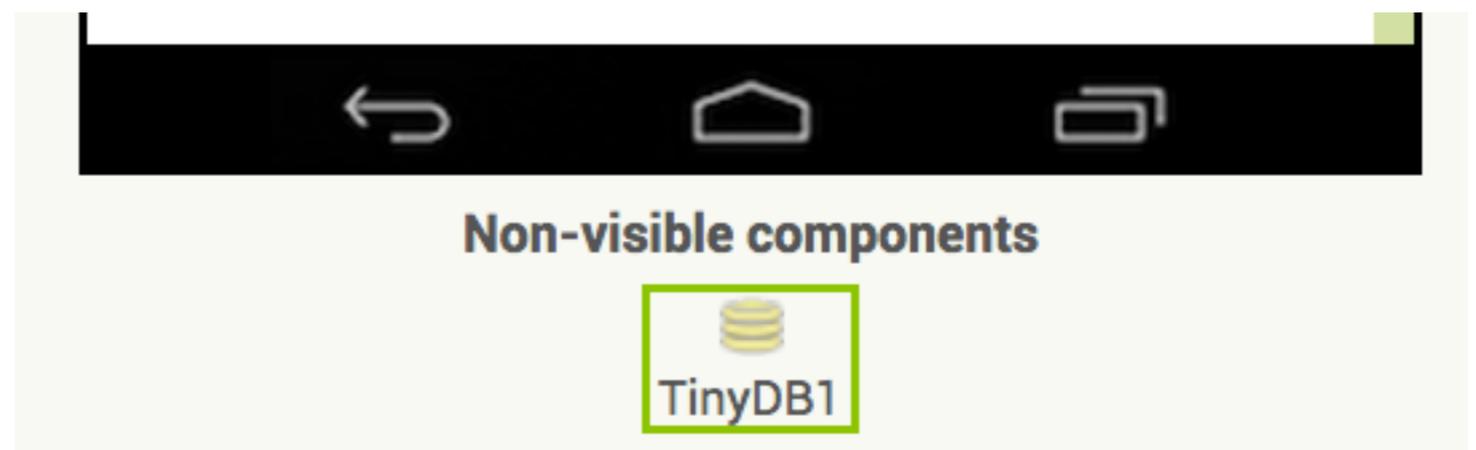
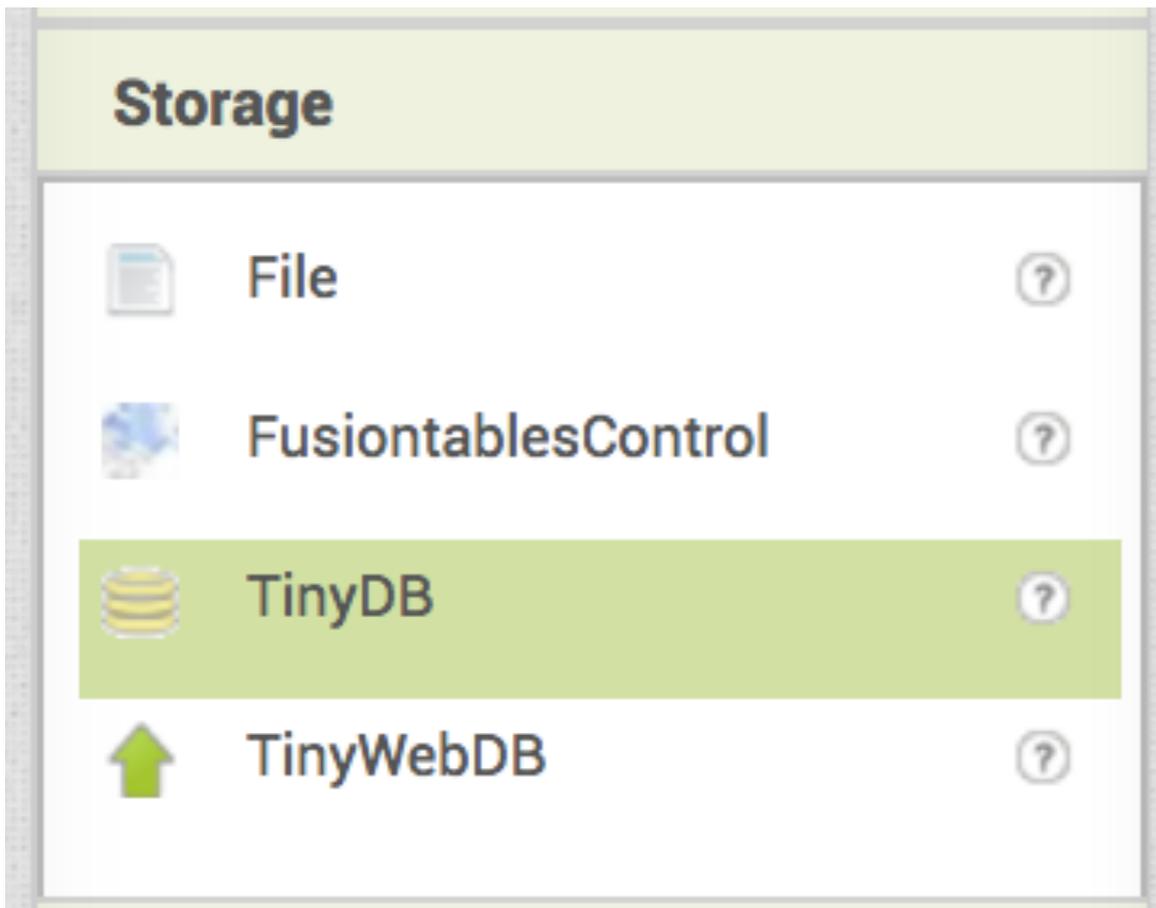
3628800

Database

- Database
 - E' possibile memorizzare informazioni in un database associato all'applicazione
- TinyDB - <http://ai2.appinventor.mit.edu/reference/components/storage.html>
 - **ClearAll():**
Clear the entire data store in the TinyDB.
 - **ClearTag(text tag):**
Clear the entry with the given tag.
 - **any GetTags():**
Return a list of all the tags in the TinyDB.
 - **any GetValue(text tag, any valueIfTagNotThere):**
Retrieve the value stored under the given tag. If there's no such tag, then return valueIfTagNotThere.
 - **StoreValue(text tag, any valueToStore):**
Store the value under the given tag. The storage persists on the phone when the app is restarted.

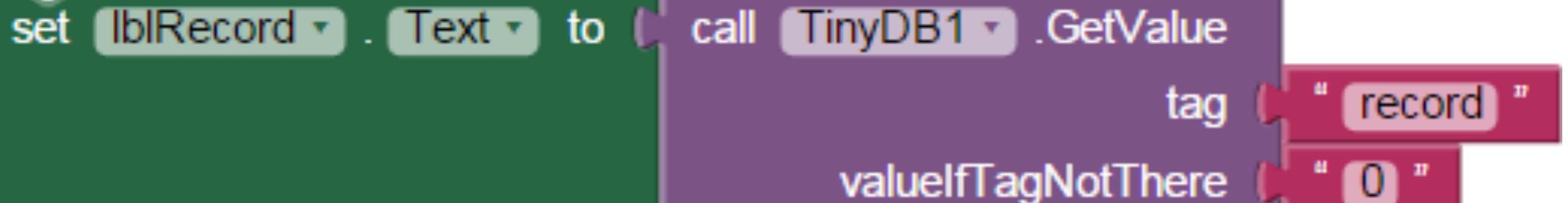
Creare il database

- Aggiungerlo è semplice
 - Basta trascinare TinyDB nell'area del Viewer
 - TinyDB non ha proprietà



Esempio: gestione punteggio massimo (record)

- Per aggiungere un elemento, si aggiunge semplicemente una coppia (tag, valueToStore)
 - Il nome del tag lo decidete voi
 - Il valore viene associato al tag specificato
- Esempio di gestione del punteggio massimo:
 - Questo blocco scrivere nella label “lblRecord” il valore registrato nel tag “record”, oppure 0 se non c’è nessun valore (è la prima volta che si gioca)

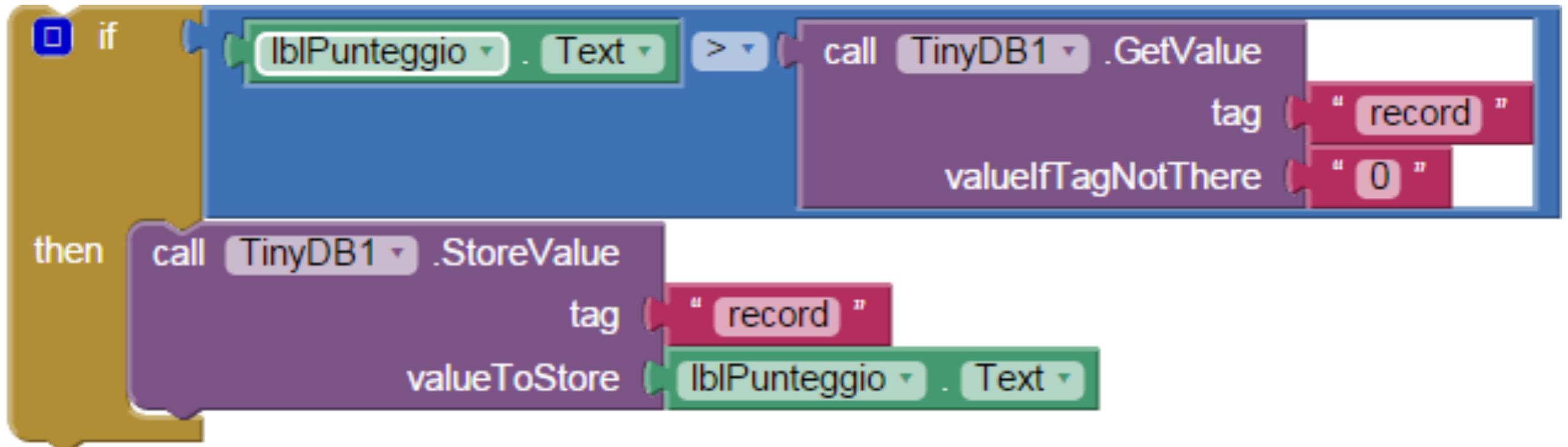


```
set lblRecord . Text to call TinyDB1 .GetValue  
tag "record"  
valueIfTagNotThere "0"
```

The image shows a Scratch code block. The first part is a 'set' block with 'lblRecord' as the object and 'Text' as the property, followed by 'to'. The second part is a 'call' block for 'TinyDB1' with the method '.GetValue'. It has two arguments: 'tag' with the value '"record"' and 'valueIfTagNotThere' with the value '"0"'. The code block is connected to a green flag icon on the left.

Esempio: gestione punteggio massimo (record)

- Per aggiungere un elemento, si aggiunge semplicemente una coppia (tag, valueToStore)
 - Il nome del tag lo decidete voi
 - Il valore viene associato al tag specificato
- Esempio di gestione del punteggio massimo:
 - Se il valore del punteggio appena ottenuto è maggiore del record, si memorizza il nuovo punteggio nel tag "record"



Componenti:

- **Esercizi per oggi:**

- <http://www.bbuio.it/didattica/appinventor/>
 - Provate il numero 5 (liste, database)
- <https://www.dropbox.com/sh/t9ambiw4neoudq/5dkbmYr5JR?n=146248414>
 - Provate il numero 14 (liste)
- Hour of Python: <https://hourofpython.com/>

- **Risorse in Italiano:**

- <http://www.mrwebmaster.it/android/guide/guida-app-inventor/>
- <http://seblogapps.blogspot.it/p/elenco-tutorial.html>

- **Risorse in Inglese**

- <http://www.appinventor.org/book2>
- <http://explore.appinventor.mit.edu/tutorials>