Liceo Scientifico P .Ruggieri Marsala

STEM 2020 - Mettiamoci le mani, usiamo la testa

Coding Class - Lezione 1

> Enrico La Sala

> 5 Febbraio 2021

Indice

- Storia del Coding ed il pensiero computazionale
- I linguaggi di programmazione: Scratch e Python
- **Dev Toolkit** - Visual Studio Code
- Concetti e Pratiche computazionali
- Le strutture condizionali
- Costruiamo una calcolatrice in Python
- **Dev Toolkit** - Github

Breve storia del Coding

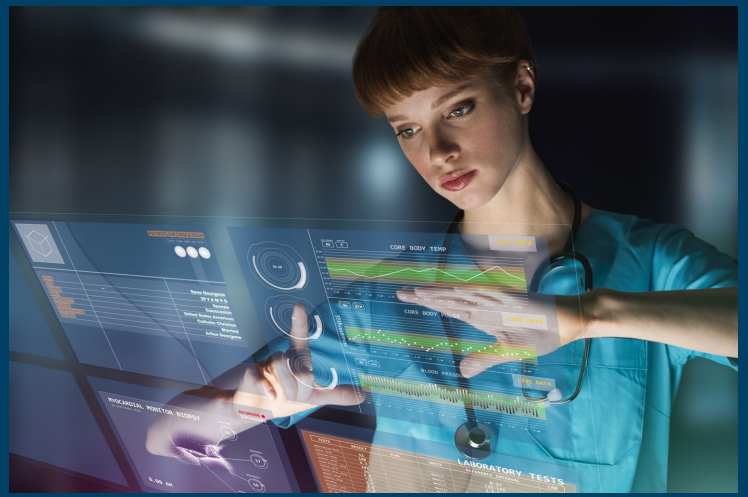
L’Informatica e il pensiero computazionale

> 2021, le macchine teorizzate da Babbage e Turing sono diventate reali e hanno rivoluzionato il mondo e l’idea stessa di innovazione!

> L’esercizio di programmazione fatto da Ada pensando ad una macchina “inesistente” è diventato la professionalità più ricercata sul mercato del lavoro per far funzionare i miliardi di oggetti programmabili esistenti che usiamo ogni giorno.

L’Informatica e il pensiero computazionale

> Per essere adeguatamente preparato a lavori del futuro (o meglio del presente!) è indispensabile avere una buona comprensione dei **concetti di base dell'informatica**. Esattamente com'è accaduto nel passato per le altre materie scientifiche.



> I benefici del “*pensiero computazionale*”: In tutti i lavori moderni ogni giorno si devono affrontare **problemi complessi**; ipotizzare **soluzioni** che prevedono **più fasi** e la **COLLABORAZIONE** con altre persone; formulare una descrizione chiara di cosa fare e quando farlo e quali debbano essere gli **output**.

CHE COS'È?

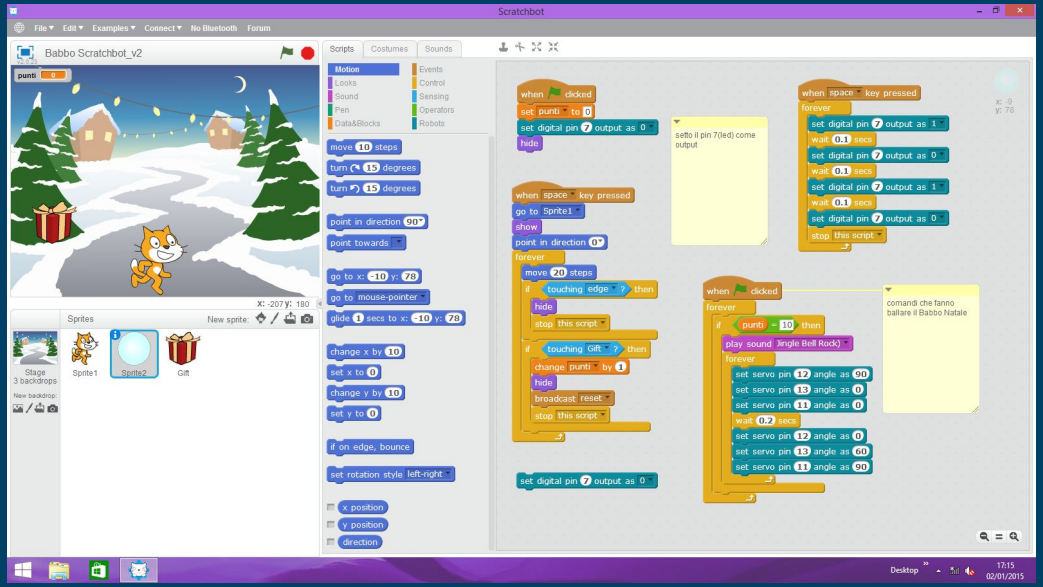
Linguaggio di programmazione ispirato alla **teoria costruzionista dell'apprendimento** e progettato per l'insegnamento della programmazione tramite primitive visive.

scratch.mit.edu

scratch.mit.edu/projects/editor/

Sviluppato dal MIT Media Lab a partire dal 2003, consente di **elaborare storie interattive, giochi, animazioni, arte e musica.**

Inoltre permette di **condividere i progetti con altri utenti del web!**



python.org

repl.it/languages/python3

Ideato da Guido van Rossum all'inizio degli anni novanta, il nome fu scelto per la passione dello stesso inventore verso i Monty Python.

CHE COS'È?

Linguaggio di programmazione dinamico **orientato agli oggetti** utilizzabile per molti tipi di sviluppo software.

Offre un forte supporto all'integrazione con altri linguaggi e programmi.

Python supporta la maggioranza dei sistemi operativi (Windows, Mac, Linux, Mobile etc.) e può essere utilizzato nella maggior parte degli ambiti professionali.

Negli ultimi anni è stato il linguaggio di programmazione con il **più alto tasso di adozione**.

Dev Toolkit - Visual Studio Code

Visual studio code è un "editor di testo" avanzato, compatibile anche con Python, che aiuta la scrittura e lo sviluppo del codice.

È possibile scaricarlo gratuitamente su https://code.visualstudio.com/download

È suggerito, una volta installato, di installare il plugin https://marketplace.visualstudio.com/items?itemName=ms-python.python seguendo la guida nella pagina

Concetti computazionali

Concetto	Descrizione
SEQUENZA	Una serie di passaggi in un azione
LOOPS	Eseguire la stessa sequenza più volte
PARALLELISMO	Far accadere le cose contemporaneamente
EVENTI	Una causa determina un effetto
CONDIZIONI	Prendere decisioni in base alle condizioni
OPERATORI	Espressioni matematiche e logiche
DATI	Memorizzare, recuperare ed aggiornare valori

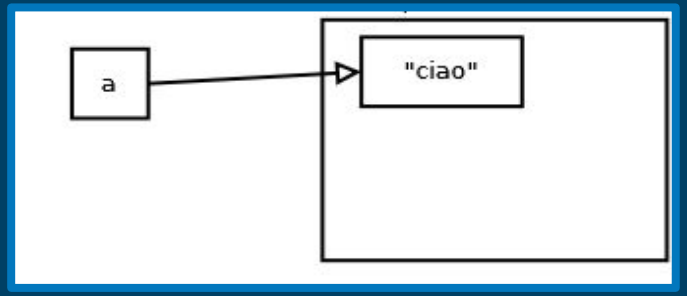
Tratto da https://www.slideshare.net/fpassantino/game-coding-for-kids

Pratiche computazionali

Pratica	Descrizione
PRODURRE PER ITERAZIONI ED INCREMENTI	Sviluppare una parte per volta, provare e svilupparne ancora
TESTARE E RIMUOVERE GLI ERRORI	Essere sicuri che le cose funzionino, cercare e risolvere gli errori
RICICLARE E MESCOLARE	Creare qualcosa partendo da materiale creato da altri
ASTRARRE E RENDERE MODULARE	Creare qualcosa di grande mettendo insieme una collezione di piccole parti

Tratto da https://www.slideshare.net/fpassantino/game-coding-for-kids

Alcuni concetti base

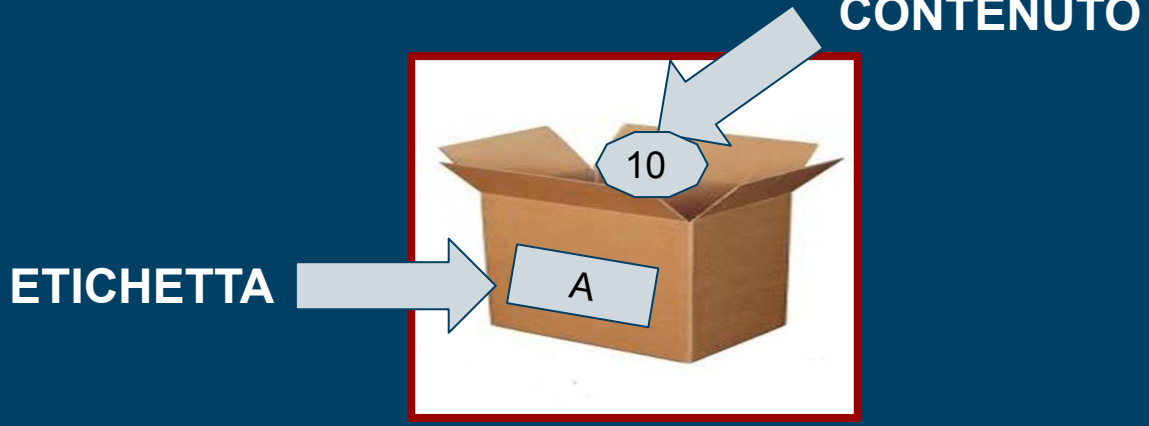


"Ciao" è un testo o una sequenza di caratteri.
In informatica una sequenza di caratteri si chiama **STRINGA**

I numeri rappresentano dei **VALORI**

576

Una **VARIABILE** è un contenitore (scatola) che ha:
1. Un'etichetta , che è il nome della variabile
2. Un contenuto, che può cambiare nel tempo



I concetti di ingresso e uscita

Azioni di ingresso (INPUT):

Azioni con cui il programma richiede in ingresso un dato. (Ex. Python - input())

Possibilità dell'utente di inserire i dati per risolvere il problema.

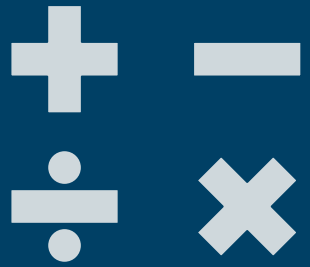
"Leggono" da una unità di ingresso e "scrivono" su una variabile, il cui stato viene pertanto alterato.

Azioni di uscita(output):

Azioni con cui il programma mostra un dato in uscita. (Ex. Python - print(), return)

Mostrano il valore finale di una variabile o un messaggio, sulla base del comportamento del programma e dei dati di ingresso inseriti.

Gli operatori

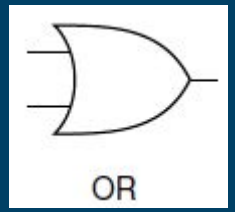


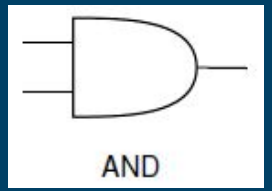
Operatori Matematici:

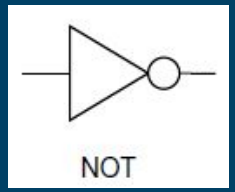
“+” , “-” ,“ x” e “/ “, vengono detti operatori matematici

Un' ESPRESSIONE aritmetica, in matematica, è un insieme di numeri legati da operatori

66 - 44 + 1







Operatori Logici

somigliano alle congiunzioni della lingua italiana

|| **OR**: condizione1 ||(or) condizione2, verifica che almeno una delle condizioni sia verificata

&& **AND** : condizione1 &&(and) condizione2, verifica che entrambe le condizioni siano verificate

| **NOT**: !condizione, indica l’opposto della condizione

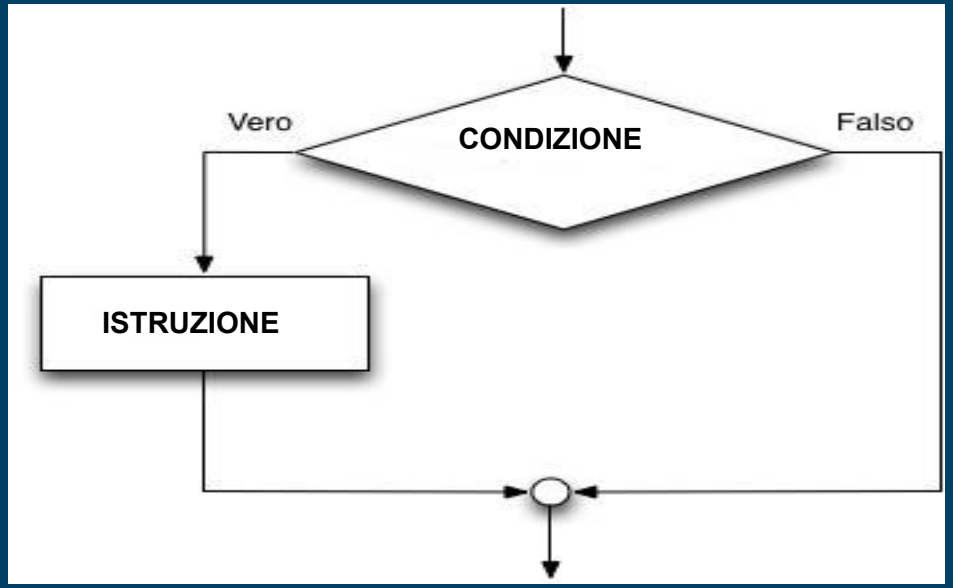
Le strutture condizionali - if/else

Le strutture di controllo condizionali consentono di specificare che una data istruzione o un dato blocco di istruzioni devono essere eseguiti **"(solo) se"** vale una certa condizione.

L'alternativa **if-then** (*se-allora*) è la più semplice forma di alternativa.

Il suo significato può essere parafrasato con la frase

SE
vale la condizione *C*,
ALLORA
esegui l'istruzione (blocco) I.



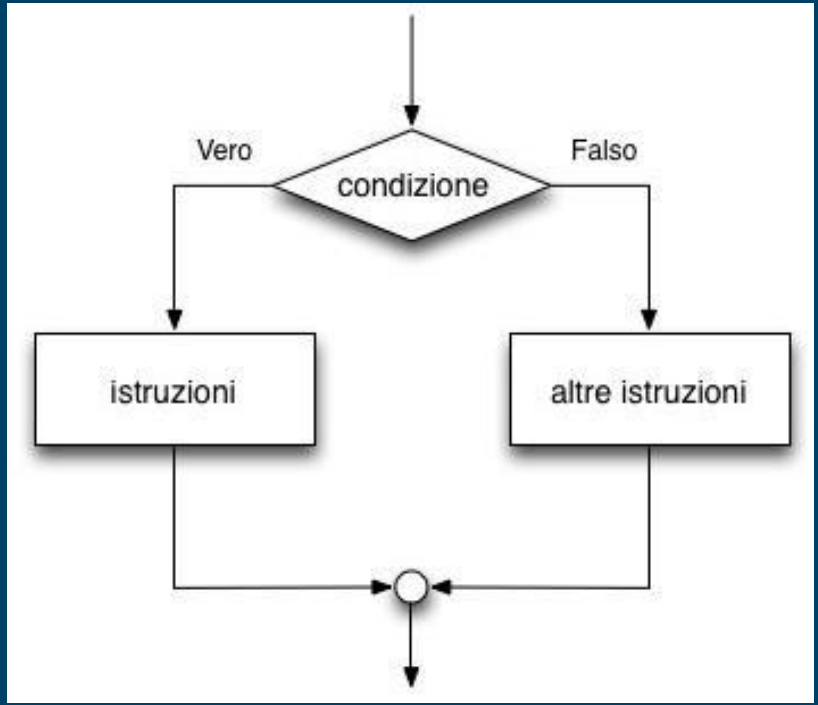
Le strutture condizionali - if/else

La maggior parte dei linguaggi di programmazione ammette anche (come variante) la forma più articolata **if-then-else** (**"se-allora-altrimenti"**), che si può parafrasare come:

SE
vale la condizione *C*

ALLORA
esegui l'istruzione (blocco) *I1*

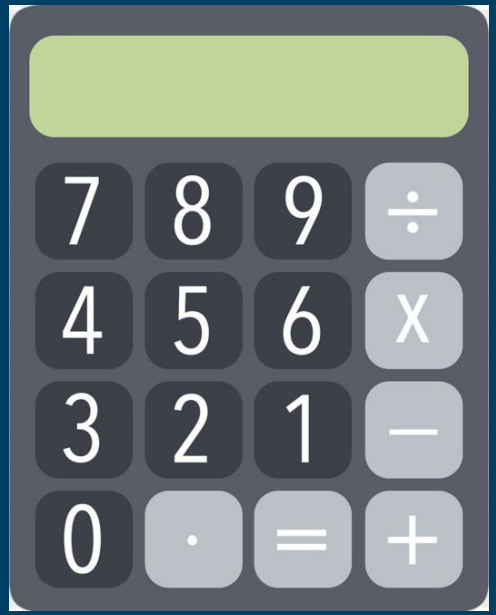
ALTRIMENTI
esegui l'istruzione (blocco) *I2*



Costruiamo una calcolatrice

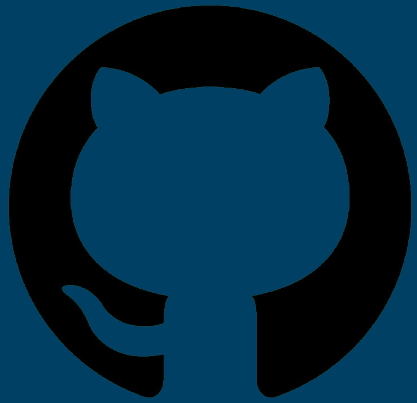
Per costruire un programma che simula una calcolatrice utilizzeremo i seguenti strumenti visti oggi:

- Operatori numerici e variabili
- Funzione input() e output (print())
- Istruzioni di controllo if e else
- Funzioni di conversione di tipo (str(), float())



Dev Toolkit - Git e Github

- Git è un software di controllo di versione distribuito che permette la condivisione di file e cartelle tra diversi utenti
- Github è una piattaforma di condivisione file basata su git
- Gli esempi visti oggi sono ospitati su github all'indirizzo: https://github.com/enryls/coding-class-stem2020-scientifico-marsala
- Potete provare a scaricare gli esercizi ed eseguirli nel vostro computer seguendo questo file con le guide: https://drive.google.com/file/d/1SOjqtjfcdd_M5PKf2CRDnAch8g92rVdQ/view?usp=sharing



Domande?