I SUPER REPORT DI

PROGRAMMA DA ZERO

Manuale di sopravvivenza per

ASPIRANTI PROGRAMMATORI

di Luca Giusti

INDICE

Introduzione

ovvero “Chi è Luca Giusti?”

Caro amico/a, prima di inoltrarti nella lettura di questo lavoro, è giusto che ti dica due parole su di me.

Il mio nome è **Luca Giusti** e sono un insegnante di informatica e reti di calcolatori presso un istituto tecnico tecnologico. Parallelamente all’attività di insegnamento, svolgo la professione di programmatore freelance, con una particolare predilezione per lo sviluppo di videogames, applicazioni web e mobile. Sono, da sempre, profondamente appassionato da tutto quello che riguarda la tecnologia, i computer e la programmazione e cerco di infondere questa mia passione nel lavoro che faccio quotidianamente con i miei studenti.

La mia personalissima *mission* come formatore ed educatore è quella di rendere semplici e facilmente assimilabili concetti tradizionalmente ritenuti complessi, complicati, difficili.

Il motivo per cui ritengo questo aspetto molto importante per un insegnante è dovuto, devo confessarlo, al fatto di non essere mai stato nel corso della mia carriera scolastica uno studente molto brillante. O almeno questa **era la mia convinzione**.

Sin dai tempi delle superiori, ho sempre dovuto studiare e lavorare parecchio per riuscire laddove alcuni dei miei compagni ‘più dotati’ andavano avanti con poco sforzo. Questa situazione risultava particolarmente accentuata nelle materie scientifiche e tecniche: passavo un sacco di tempo a rielaborare gli appunti delle lezioni, cercando di integrare il materiale usando fonti e libri diversi (al tempo internet era ancora alla portata di pochi, ahimè). Il risultato era che ricostruivo letteralmente le lezioni, impostandole in modo tale che fosse **facile e veloce** per me ripassare e prepararmi per le verifiche.

Fortunatamente, andando avanti con gli studi cominciai a sospettare che le mie ‘difficoltà’ non dipendessero solo da me: esistevano infatti pochi, rarissimi docenti che avevano il dono di rendere tutto molto più facile e comprensibile. Un docente di fisica all’università in particolare, con **un semplice schema ed una metafora** riuscì a chiarirmi un concetto che, per quanto mi fossi sforzato, non avevo mai compreso veramente durante cinque anni di superiori.

Le prime esperienze come insegnante in seguito confermarono in me la convizione che **non esistevano cattivi allievi, ma solo pessimi maestri**. Iniziai quindi a considerare sotto un altro punto di vista l’insuccesso di uno studente, **vedendolo più come un mio insuccesso**. Questa presa di coscenza ha costituito un passaggio fondamentale nella mia crescita professionale, cambiando letteralmente l’idea che avevo di ‘buon insegnante’.

Adesso svolgo il mio lavoro assumendomi buona parte della responsabilità del successo dell’apprendimento, cercando di mettere i miei studenti nelle condizioni migliori per imparare, proponendo del materiale e delle lezioni che siano chiare e facilmente assimilabili.

Il lavoro che stai per leggere e tutte le risorse di “Programma da zero” nascono proprio con questo obiettivo, aiutare chiunque ad entrare nel mondo della programmazione eliminando la paura di non esserne capace o non essere abbastanza ‘intelligente’.

“Non ho mai insegnato nulla ai miei studenti; ho solo cercato di metterli nelle condizioni migliori per imparare”

Albert Einstein

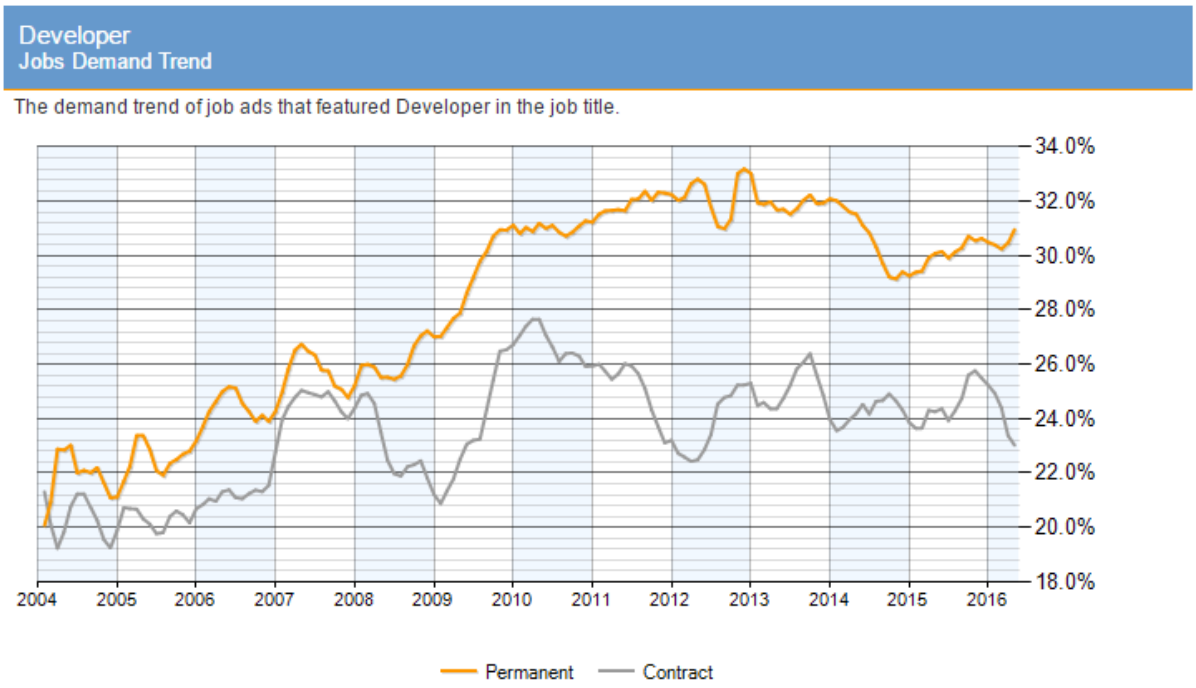
Perchè imparare a programmare?

i 5 motivi per cui non ti pentirai mai di questa decisione!

Se stai leggendo questo report, molto probabilmente hai già un buon motivo per volerti avvicinare alla programmazione. O forse no, ed hai le idee un po' confuse. In ogni caso, permettimi di mostrarti alcune delle ragioni principali per cui imparare a programmare rappresenta, al giorno d'oggi, una delle migliori scelte tu possa fare per il tuo futuro.

Ti offre maggiori opportunità di lavoro e di guadagno

Cercherò di non girarci troppo attorno ed andrò direttamente alla motivazione più importante: **un programmatore non avrà mai problemi a trovare lavoro** . Non c'è bisogno che sia io a dirti come in tempi di crisi globale e generalizzata, avere la certezza di un reddito non sia cosa da poco. Qua sotto puoi osservare alcuni grafici tratti da un famoso sito inglese specializzato nel trovare lavori legati all'Information Technology (http://www.itjobswatch.co.uk):



Percentuale di richieste di lavoro per sviluppatori (Developers) rispetto al totale delle richieste

Il trend è evidente. Negli ultimi anni la richiesta da parte delle aziende (inglesi in questo caso, ma è un fenomeno comune a tutta l'area europea) **è quasi sempre andata crescendo**. Non c'è quasi traccia della tremenda crisi che sta mettendo in ginocchio gran parte del mercato del lavoro.

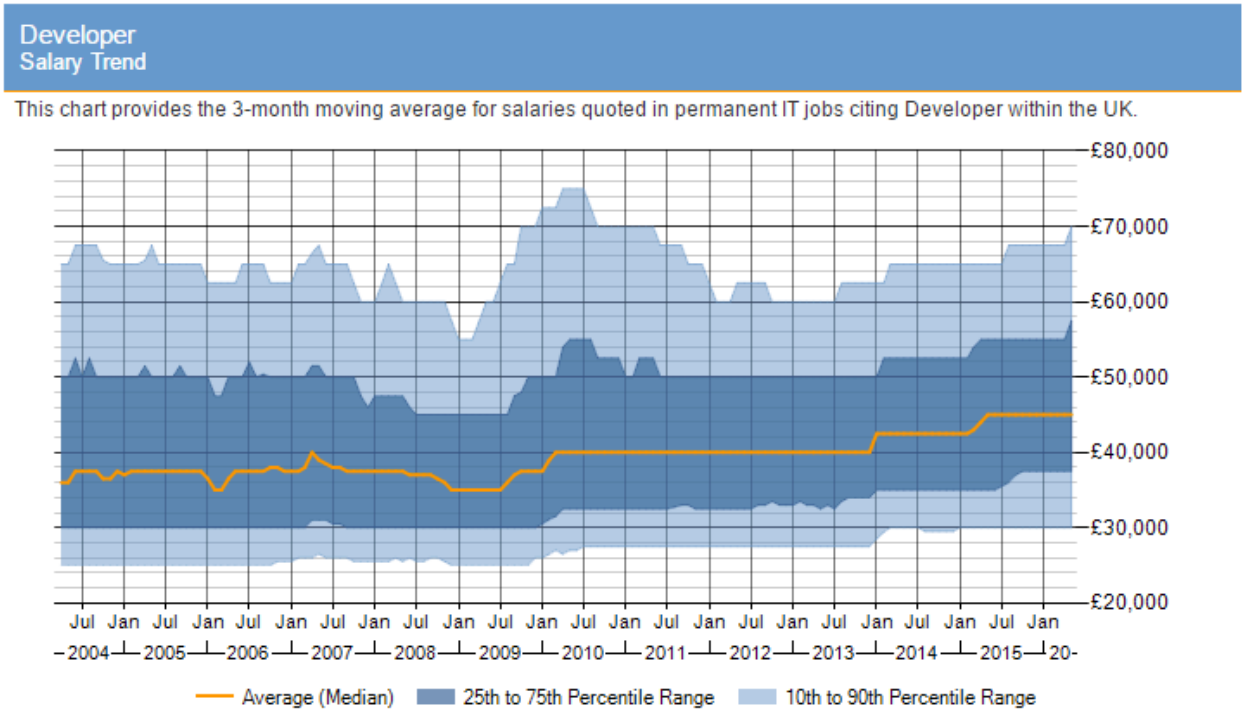
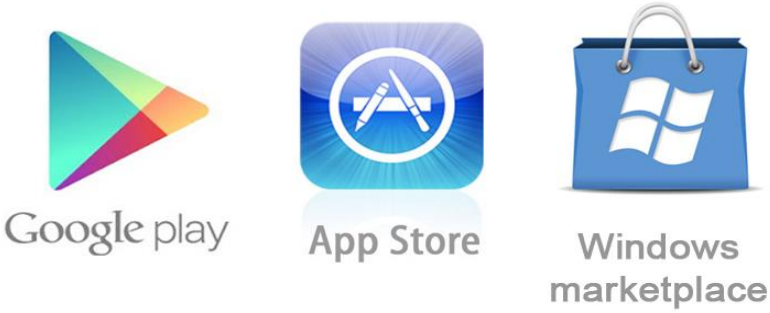


Grafico della media dei salari degli sviluppatori software (Developers) nel regno unito

Anche i salari medi sono di tutto rispetto: parliamo di una media di **45.000 sterline** annue (al cambio attuale circa 58.000 euro) con punte di **quasi 70.000 (**oltre 90000 euro). Ovviamente siamo lontani dai favolosi guadagni milionari dei guru dell'informatica moderna; il punto però è che **imparare a programmare non è come comprare un biglietto della lotteria**, ma piuttosto **un investimento** in vista di una professione sicura e mediamente ben retribuita.

Ad ogni modo, chi si sentisse particolarmente ispirato e creativo (oltre che molto fortunato!) può sempre tentare la strada dello sviluppo indipendente e cercare di vendere direttamente al pubblico le sue applicazioni attraverso i vari negozi virtuali presenti in rete (Apple Store, Google Play, Windows Store, Amazon App Store, Steam, eccetera). Stiamo parlando di un bacino di **miliardi di utenti** potenzialmente raggiungibili, con possibilità di guadagno veramente enormi.

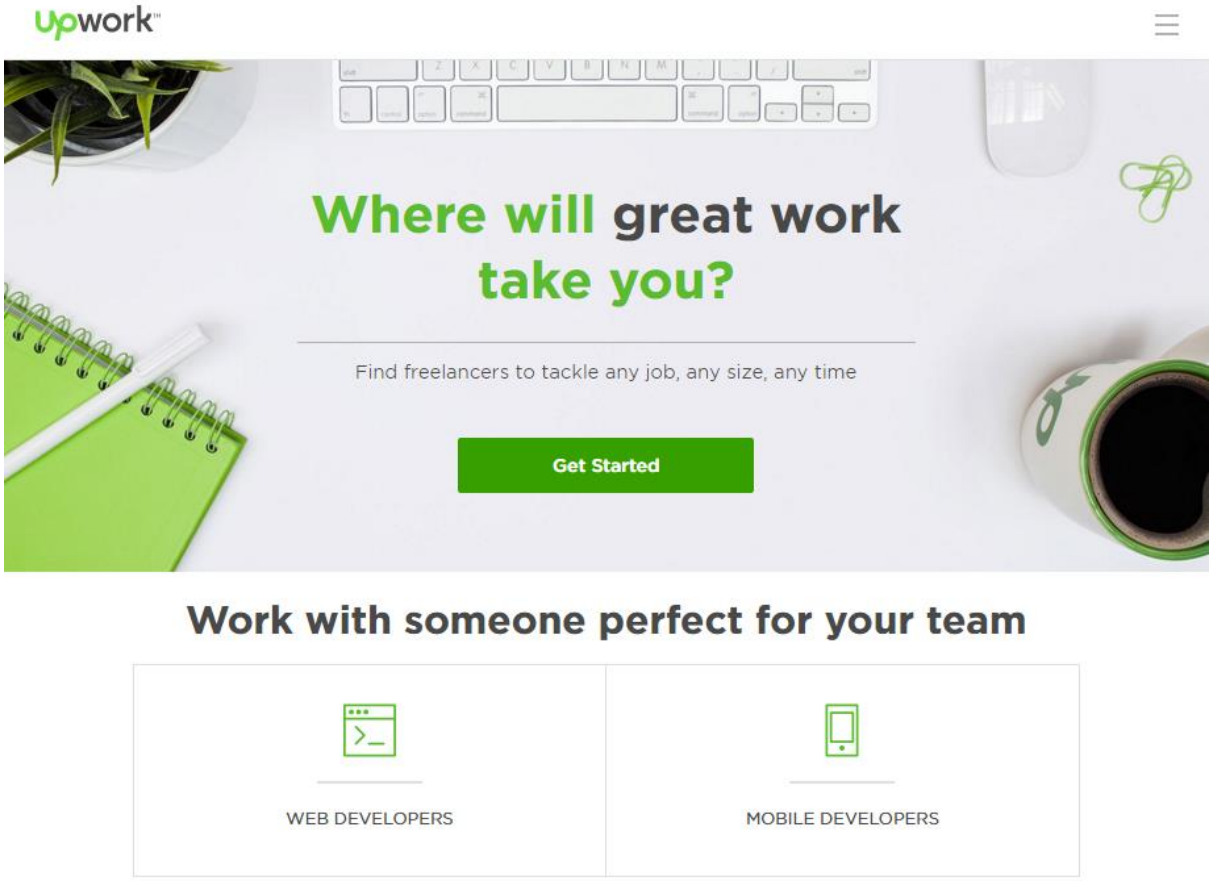


La famosissima applicazione di ritocco e condivisione foto per smartphone Instagram, è un classico esempio di come sia possibile fare (tanti!) soldi sviluppando e vendendo il proprio software mediante questi canali. Ideata e rilasciata nel 2010 da Kevin Systrom e Mike Krieger, conta ad oggi milioni di download, con guadagni stimati superiori al miliardo di dollari. Ovviamente, il successo in questo mercato dipende molto dall'avere l'idea giusta al momento giusto e, soprattutto, promuovere efficacemente la propria applicazione garantendole una buona visibilità (marketing).

Ti regala maggiore libertà

Un ulteriore aspetto positivo del diventare uno sviluppatore è la **liberta**, intesa come la possibilità di svolgere il proprio lavoro **praticamente ovunque**.

A differenza di molte altre professioni infatti sono sufficienti un computer ed una connessione ad internet per poter svolgere la propria attività; un programmatore freelance può scegliere liberamente se **lavorare da casa** ed in quale parte del mondo vivere, gestendo i propri clienti e lavori attraverso internet e/o piattaforme di outsourcing come ad esempio **UpWork.**



Infine, nonostante il grado di libertà di uno sviluppatore freelance sia **quasi totale**, anche come programmatore dipendente le possibilità di accedere a contratti molto flessibili (che contemplino formule miste di telelavoro e lavoro in ufficio) sono sempre più frequenti.

Aumenta la tua cultura tecnologica

La terza motivazione per cui ritengo che imparare a programmare sia un ottimo proposito (anche se svolgi una professione non proprio del settore informatico) consiste nel fatto che avere una mentalità informatica contribuisce parecchio ad **aumentare la propria cultura tecnologica,** e dunque la capacità di **orientarsi in un mondo oramai permeato da dispositivi elettronici programmabili**.

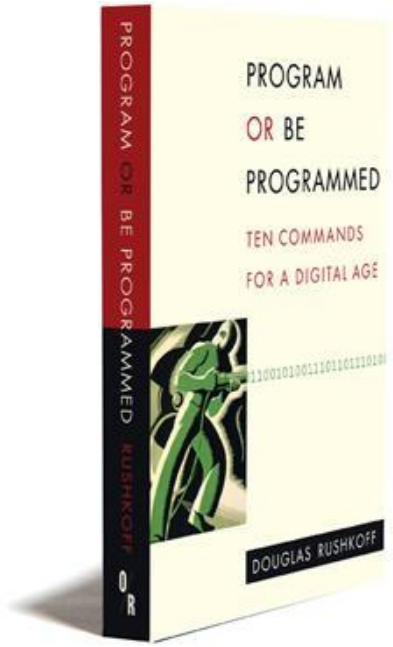
La tendenza (o se vogliamo essere più tragici, il destino) della maggior parte delle persone è quello di diventare **succube della tecnologia**, utenti passivi ed inconsapevoli che saranno sempre meno in grado di capire la natura della realtà che li circonda.

Voglio citarti il terribile ma illuminante libro di Douglas Rushkoff, "Programma o sarai programmato" che sintentizza molto bene i rischi che possono derivare dallo stato di suddito tecnologico**:**

"L'entusiasmo per una tecnologia digitale che comprendiamo poco e su cui abbiamo scarso controllo, ci sta portando verso un livello di consapevolezza più basso. Finiamo per ritrovarci alla mercè di macchine [...] decifrabili solo da chi le ha programmate e la cui neutralità va accettata come un atto di fede. Diventiamo dipendenti da motori di ricerca e smartphone sviluppati da aziende che possiamo solo sperare valutino maggiormente la nostra produttività rispetto ai loro obiettivi commerciali. Impariamo a socializzare e a fare amicizia tramite interfacce e reti che sembrano più interessate ad individuare un valido modello pubblicitario che ad aiutarci a sviluppare nuove relazioni"

ed ancora:

“La programmazione è il punto di forza ideale di ogni società digitale, ma se non impariamo a programmare rischiamo di essere programmati da qualcun’altro. Non è troppo difficile nè troppo tardi per apprendere il codice che si nasconde dietro le cose comuni, o quantomeno capire che esistono dei codici nascosti tra le interfacce di siti e programmi. In caso contrario, restiamo alla mercè dei programmatori, di chi li paga e perfino della tecnologia in quanto tale.”

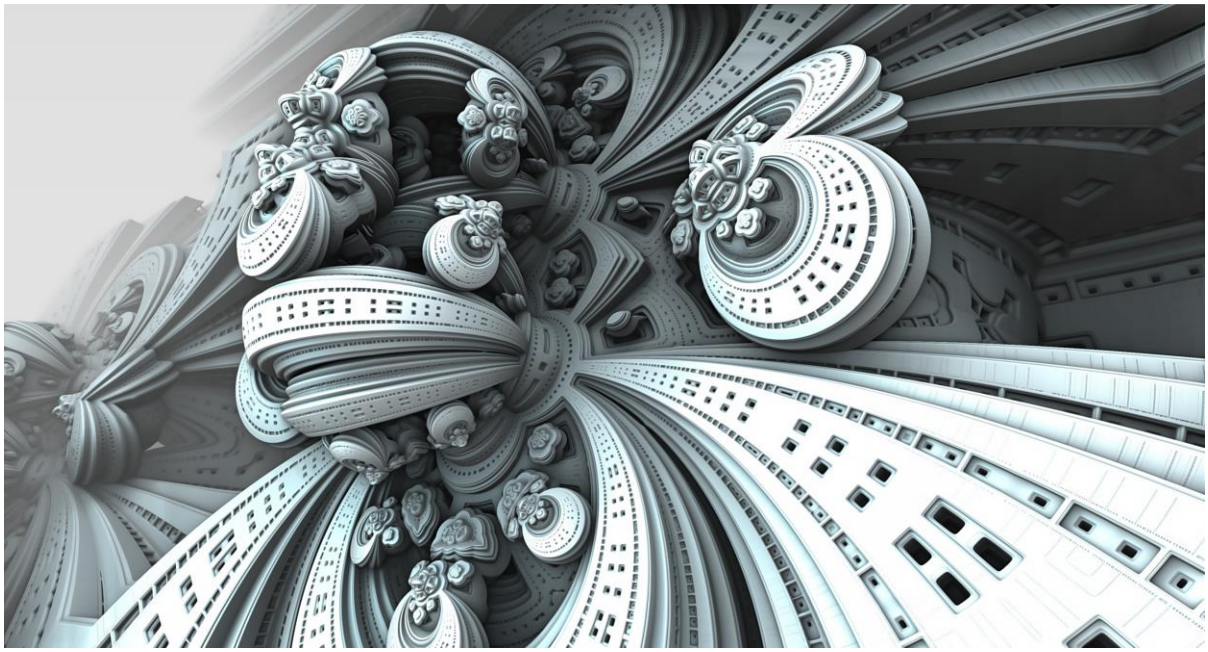


Anche se si possono non condividere i toni di Rushkoff, è innegabile che come semplici utenti saremo sempre costretti (volenti o nolenti) a **subire le scelte** e le decisioni di chi crea il software che utilizziamo quotidianamente sui nostri PC, tablet e smartphone. Imparare a programmare forse non ci renderà completamente indipendenti, ma come minimo **ci renderà più consapevoli** dei rischi e delle limitazioni alla libertà personale che l’uso di un software può comportare.

Espande le tue possibilità creative

Quasi sempre dipinta come abilità prettamente **tecnica**, dare la vita ad un software può rivelarsi in realtà un attività **sorprendentemente creativa**: inoltre la programmazione è sempre più uno **strumento fondamentale** per molte delle professioni più creative e per i cosiddetti **artisti digitali**.

La manifestazione più estrema della fusione tra arte e programmazione è sicuramente l'**Arte Generativa** (http://it.wikipedia.org/wiki/Arte_generativa), una recentissima forma d'arte che sfrutta degli algoritmi per la creazione di immagini, sculture, musica ed animazioni: si programma una macchina e la si istruisce sul come generare l'opera che si ha in mente.



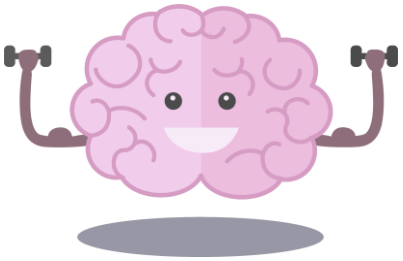
Anche se non sei un 'vero artista', ma svolgi una professione fortemente creativa come il web designer, il grafico, l'animatore, il video maker o simili, conoscere le basi della programmazione può rivelarsi una vera e propria **arma micidiale** per **fare la differenza** rispetto ad una concorrenza che si aggrappa a pratiche e metodi di lavoro più tradizionali.

Migliora la tua mente e potenzia il tuo modo di pensare

Senza scomodare troppo la filosofia, alla base della programmazione c'è l'abilità di **risolvere problemi** e quindi di **pensare**. Imparare a programmare ha dunque l'effetto di allenare la tua mente, disciplinandola, aumentando la memoria e rinforzando capacità analitiche, di sintesi e l'intuito.

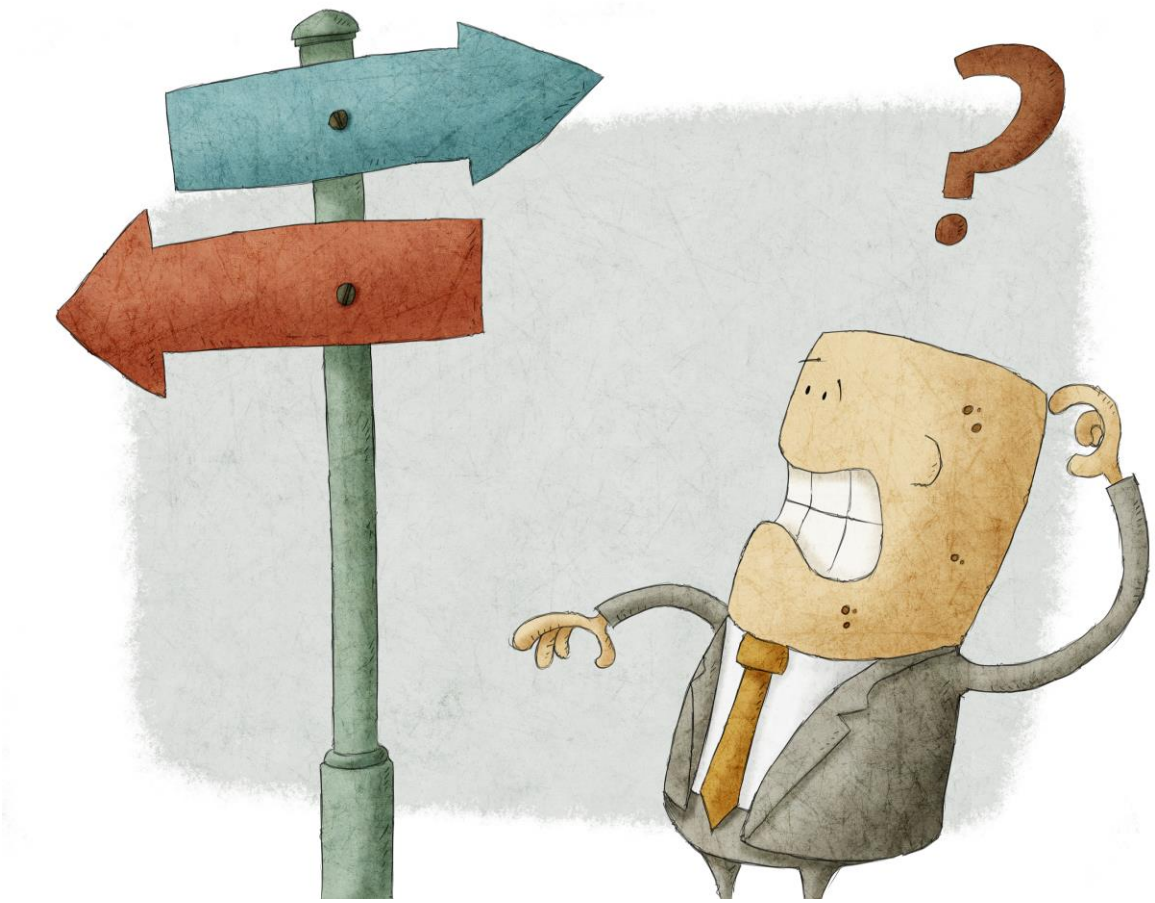
In particolare, ecco alcune delle abilità che vedrai crescere man mano che migliorerai come programmatore:

- Capacità di concentrazione e pazienza
- Capacità di prestare attenzione ai dettagli
- Memoria a breve, medio e lungo termine
- Abilità di ridurre un grande problema in problemi più piccoli e facili da risolvere
- Capacità di astrazione
- Abilità di vedere le cose da diversi punti di vista
- Capacità di visualizzare sistemi e processi



Da dove iniziare?

“Chi ben comincia è a metà dell’opera”



Come recita fin troppo banalmente il noto proverbio, individuare un buon punto di partenza è fondamentale per la buona riuscita di qualsiasi impresa. Partire con il piede giusto infatti, non solo **aumenta le tue probabilità di successo**, ma ti consente di **ridurre il tempo** necessario ad ottenere i primi **risultati**.

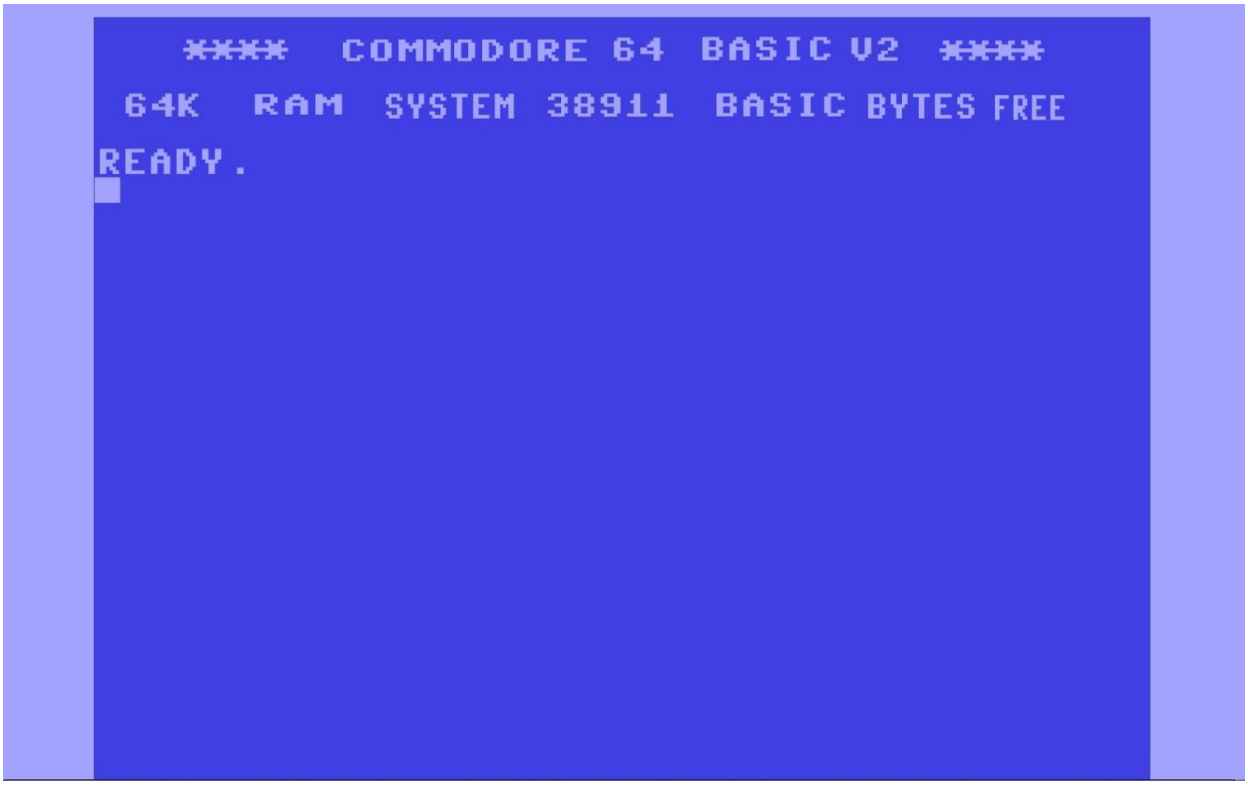
Se sei impaziente di lanciarti in questa nuova avventura, puoi saltare direttamente alla sezione **Spunti per cominciare,** dove troverai informazioni su alcune delle migliori opzioni per iniziare subito ad imparare.

In tal caso però, ti consiglio di trovare del tempo per tornare a leggere i paragrafi che seguono, dove oltre a confessarti quali siano stati i miei personali e goffissimi inizi (!!!) cercherò di metterti in guardia dagli errori più comuni che potresti fare come apprendista programmatore.

Da dove ho iniziato io

Nell'ormai remoto natale del 1984, ricevetti come regalo una misteriosa macchina da scrivere che poteva collegarsi alla televisione e dare vita, magicamente, ad un'infinità di giochi meravigliosi:

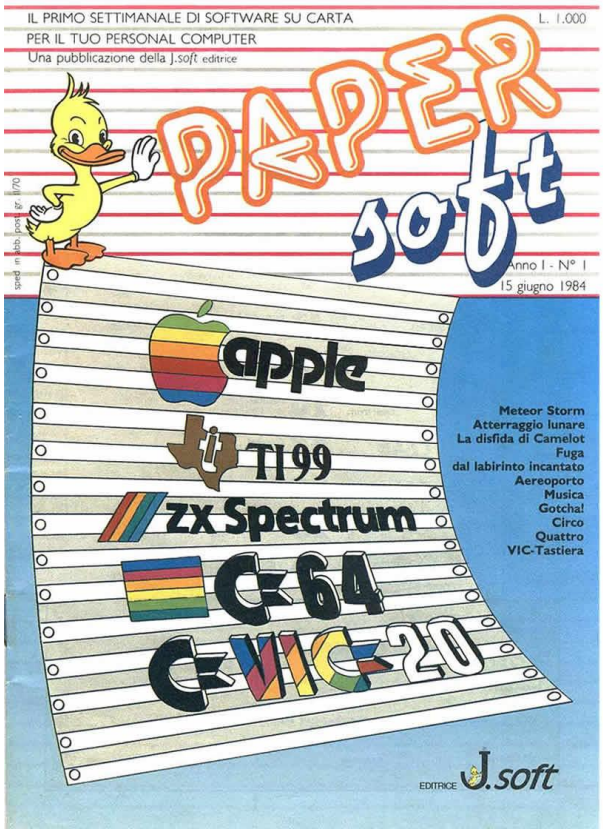
Il glorioso Commodore 64 fu uno dei primi computer ad entrare nelle case della gente portando con se un mondo di divertimento ed **affascinando letteralmente un'intera generazione**. Quello che però rendeva il C64 così interessante (almeno ai miei occhi) rispetto alle console da gioco molto diffuse all'epoca (come l'ATARI 2600 ad esempio) era la possibilità di utilizzare la tastiera per digitare e **creare dei programmi**. All'avvio del computer infatti, compariva sullo schermo della TV una strana schermata blu, con un **curioso quadratino lampeggiante** che invitava a scrivere qualche cosa:

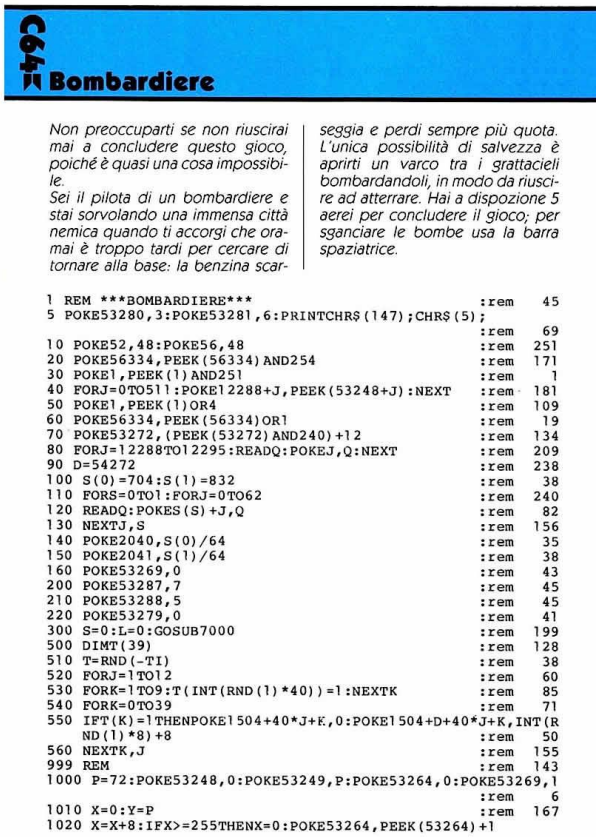


Nonostante il desiderio di creare qualcosa fosse molto forte, mi ritrovavo purtroppo senza la più pallida idea di **come iniziare**. Per di più questo senso di smarrimento era accentuato dal fatto che la macchina **sembrava capire solo l'inglese**, o un linguaggio ad esso molto simile.

Ci vollero anni, prima che potessi anche solo dare un significato alle frasi che leggevo ed a mia volta digitavo sulla tastiera. Termini come PRINT, GOSUB, RETURN e compagnia bella erano solo delle **buffe parole** che, in quanto tali, ero molto bravo a ricordare (e pronunciare in un inglese super-maccheronico!).

Senza una guida e con il solo aiuto di un risicato manuale utente (in inglese per giunta) i miei primi approcci alla programmazione furono **abbastanza ridicoli**. Per quanta buona volontà e passione ci mettessi, il massimo che riuscivo a fare era copiare i listati di programmi pubblicati su alcune riviste specializzate cercando di farli funzionare. Tra le tante che al tempo si potevano trovare in edicola la mia preferita era **Paper Soft**, dedicata esclusivamente ai listati di **videogiochi**:





C64

Bombardiere

Non preoccuparti se non riuscirai mai a concludere questo gioco, poiché è quasi una cosa impossibile.
Sei il pilota di un bombardiere e stai sorvolando una immensa città nemica quando ti accorgi che ormai è troppo tardi per cercare di tornare alla base: la benzina scarseggia e perdi sempre più quota. L'unica possibilità di salvezza è aprirti un varco tra i grattacieli bombardandoli, in modo da riuscire ad atterrare. Hai a dispozione 5 aerei per concludere il gioco; per sganciare le bombe usa la barra spaziatrice.

```
1 REM ***BOMBARDIERE***                                  :rem   45
5 POKE53280,3:POKE53281,6:PRINTCHR$(147);CHR$(5);
                                                         :rem   69
10 POKE52,48:POKE56,48                                   :rem  251
20 POKE56334,PEEK(56334)AND254                           :rem  171
30 POKE1,PEEK(1)AND251                                   :rem    1
40 FORJ=0TO511:POKE12288+J,PEEK(53248+J):NEXT            :rem  181
50 POKE1,PEEK(1)OR4                                      :rem  109
60 POKE56334,PEEK(56334)OR1                              :rem   19
70 POKE53272,(PEEK(53272)AND240)+12                      :rem  134
80 FORJ=12288TO12295:READQ:POKEJ,Q:NEXT                  :rem  209
90 D=54272                                               :rem  238
100 S(0)=704:S(1)=832                                    :rem   38
110 FORS=0TO1:FORJ=0TO62                                 :rem  240
120 READQ:POKES(S)+J,Q                                   :rem   82
130 NEXTJ,S                                              :rem  156
140 POKE2040,S(0)/64                                     :rem   35
150 POKE2041,S(1)/64                                     :rem   38
160 POKE53269,0                                          :rem   43
200 POKE53287,7                                          :rem   45
210 POKE53288,5                                          :rem   45
220 POKE53279,0                                          :rem   41
300 S=0:L=0:GOSUB7000                                    :rem  199
500 DIMT(39)                                             :rem  128
510 T=RND(-TI)                                           :rem   38
520 FORJ=1TO12                                           :rem   60
530 FORK=1TO9:T(INT(RND(1)*40))=1:NEXTK                  :rem   85
540 FORK=0TO39                                           :rem   71
550 IFT(K)=1THENPOKE1504+40*J+K,0:POKE1504+D+40*J+K,INT(R
    ND(1)*8)+8                                           :rem   50
560 NEXTK,J                                              :rem  155
999 REM                                                  :rem  143
1000 P=72:POKE53248,0:POKE53249,P:POKE53264,0:POKE53269,1
                                                         :rem    6
1010 X=0:Y=P                                             :rem  167
1020 X=X+8:IFX>=255THENX=0:POKE53264,PEEK(53264)+1
```

In pratica, quello che si faceva era **digitare con cura certosina** i caratteri riportati dai listati sulla tastiera del computer, facendo molta attenzione a non introdurre errori di battitura.

Non è difficile intuire come il processo potesse diventare facilmente **un vero incubo**: copiare le istruzioni del programma avendo solo una vaga idea di quello che si stava facendo oltre ad essere **estremamente noioso** apriva le porte ad un **enorme frustrazione** qualora il programma non avesse funzionato o lo avesse fatto in modo diverso dalle aspettative. Ancora oggi mi meraviglio di quanto sia stato paziente ed ostinato, molti avrebbero sicuramente abbandonato l'impresa per dedicarsi a qualcosa di più divertente (ed intorno ai 13 anni di alternative divertenti c'era solo l'imbarazzo della scelta!).

Ciò nonostante io continuai in questa pratica masochista e, piano piano, terminai con l'imparare il linguaggio BASIC. E lo imparari così bene, che non solo riuscivo (finalmente!) a capire quello che gli autori dei listati avessero in mente, ma ero persino in grado di **fare delle modifiche e personalizzare i giochi**!

Il ricordo della **soddisfazione** di quei momenti è tuttora indelebile dentro di me: non c'è cosa più gratificante che ottenere un risultato dopo essersi impegnati tanto per raggiungerlo.

Pur tuttavia, il seguito della mio percorso come programmatore (ed insegnante, più avanti) mi mostrò ben presto come nel mio approccio infantile fossero insiti un sacco di **errori** e **scelte sbagliate** (certamente perdonabili data la tenera età!).

Nonostante i tempi siano molto cambiati rispetto agli anni '80 ho potuto constatare come alcuni di questi errori siano **tuttora molto comuni** e pericolosamente in agguato.

I 3 errori più comuni che commette chi vuole iniziare a programmare

(che possono rallentare o addirittura pregiudicare il tuo successo)

Vediamo adesso in dettaglio quali sono **le tre trappole** nelle quali più spesso cadono le persone che per la prima volta si avvicinano al mondo della programmazione.

Errore 1 - Fare tutto da soli

E' vero, alcune persone hanno imparato a programmare con il metodo brevettato **"Impara a nuotare o affoga™"**. La ricetta è semplice: un computer, un bel manuale, una connessione ad internet ed il gioco è fatto, siamo pronti per cominciare. Google ed una bella musica di sottofondo completano questo scenario di studio solitario, "matto e disperatissimo".

Pochi però ci raccontano quello che succede a questi volenterosi autodidatti: parafrasando Highlander, ne resterà soltanto uno (o comunque, pochi). La maggior parte si perde alla prima o alla seconda difficoltà e **scoraggiata abbandona i suoi propositi**.

L’approccio autodidatta, statisticamente parlando, è fallimentare. E non solo per l’elevato numero di **scottati** che difficilmente si avvicinerà alla programmazione una seconda volta. Anche chi per talento, predisposizione e/o ostinatezza riesce nell’impresa ottiene risultati in tempi molto lunghi.

Lo studente deve infatti affrontare e cercare di risolvere **da solo** molti problemi chiave, ovvero:

- Individuare un **punto di partenza** che sia adatto per un principiante
- Definire degli **obiettivi realistici** ed alla sua portata
- Stabilire un **percorso chiaro** da seguire
- Valutare i **progressi ottenuti** mano mano che procede
- Chiarire gli eventuali **dubbi** che sorgono
- Individuare e correggere gli **errori** che inevitabilmente farà

Ci sono oggettivamente **troppe trappole** nelle quali un principiante senza una guida può cadere, alcune purtroppo **mortali** per le buone intenzioni di molti.

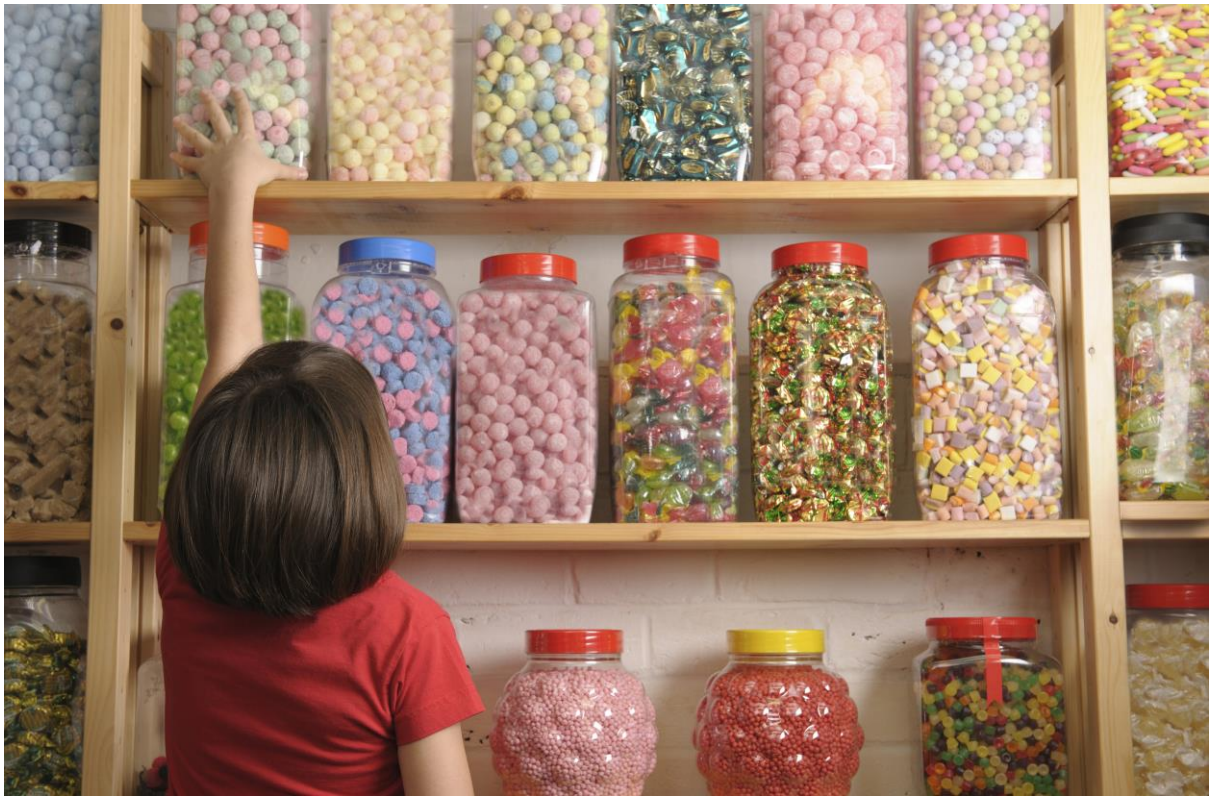
L’avvento di Internet **un po’ ha migliorato** l’esperienza di chi vuole intraprendere questo viaggio da solo: al giorno d’oggi infatti è facilissimo trovare **tonnellate di informazioni**, guide, manuali e video su praticamente qualsiasi argomento. La rete inoltre **pullula di forum** e comunità alle quali rivolgersi per cercare **consigli ed aiuto**.

Il punto però è che siamo passati da uno scenario in cui l’informazione era **scarsa** ad uno nel quale semplicemente **ce n’è troppa**. Il fenomeno prende il nome di **Information overload**, e mette un principiante di fronte a scelte del tipo:

“*Da quale tra le **centinaia di linguaggi di programmazione** esistenti mi conviene iniziare?*”

“*Quali tra le **migliaia di libri e manuali sull’argomento** sono più adatti a me?*”

“*Quali tra le **decine di migliaia di siti, blog e forum sulla programmazione** possono essermi realmente utili?*”



Queste domande per chi inizia possono essere **veramente paralizzanti**: generano un senso di insicurezza, una mancanza di direzione che può essere difficile da superare.

Da qui **l'importanza di una guida**, un mentore, ovvero una persona che:

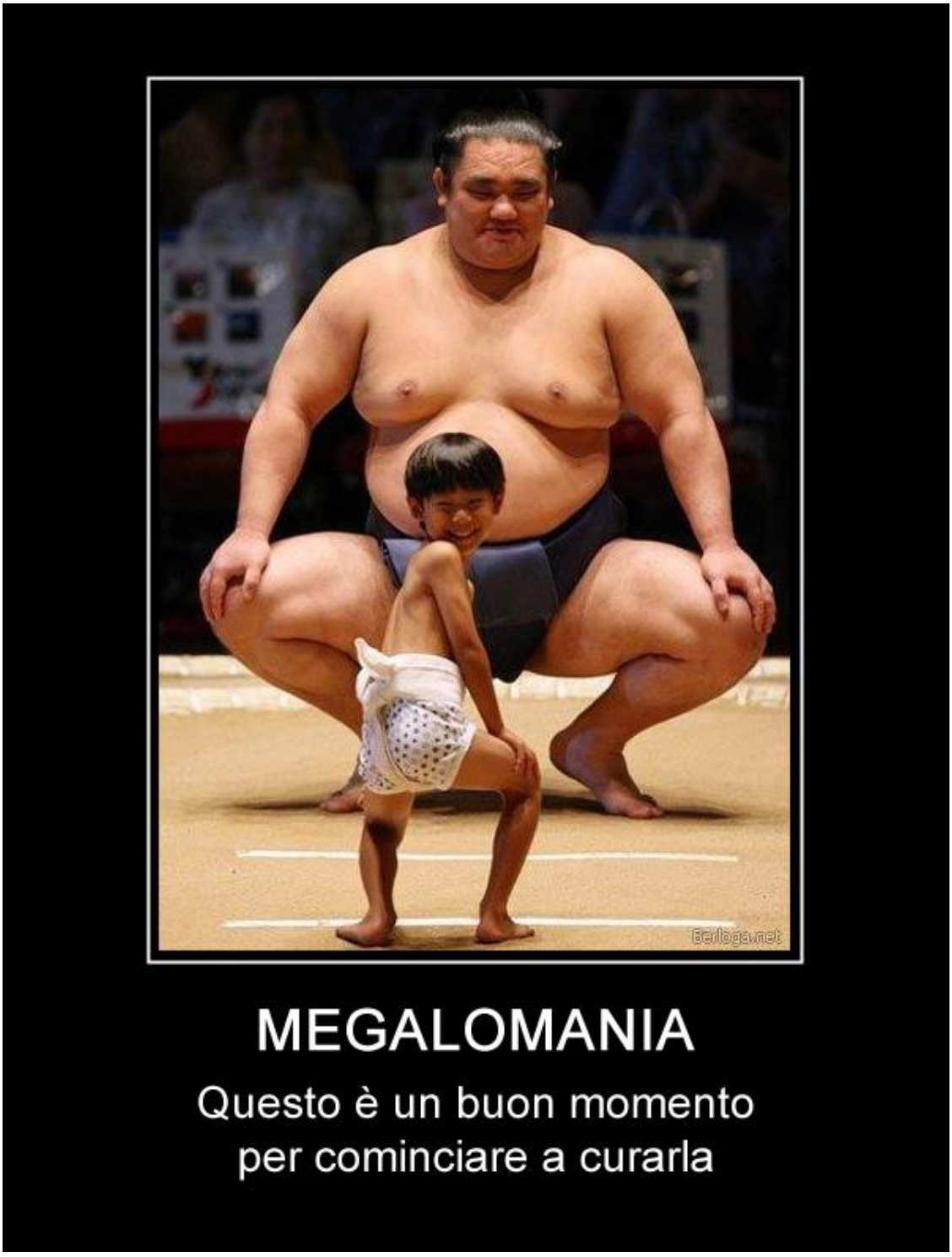
- come te sia partita da zero ed abbia **affrontato tutte le trappole** e le insidie che si incontrano lungo il percorso, sopravvivendo!
- abbia **compreso**, spesso a sue spese, quali siano **gli errori** da evitare e possa **metterti in guardia** da commetterli a tua volta
- abbia **individuato** grazie all'esperienza dei **percorsi più sicuri**, strade alternative adatte a chi comincia
- sia riuscita infine ad **aiutare altre persone** ad ottenere dei risultati

Penso sia abbastanza chiaro ed innegabile il vantaggio di poter contare su qualcuno del genere: in sintesi, **maggiori probabilità di riuscita** in un **tempo minore**. Il problema a questo punto diventa **trovare questo tipo di persona,** in grado e soprattutto **disposta ad aiutarti**.

Ti prometto che più avanti affronteremo il problema in maniera esauriente, esaminando più di una soluzione.

Errore 2 - Non partire dalle basi

Il secondo errore, conseguenza di non avere una guida che ti possa orientare verso un buon punto di partenza, consiste nell'**ignorare le basi della programmazione.** Succede spessissimo che **persone completamente a digiuno** di ogni minimo concetto relativo allo scrivere programmi inizino a studiare linguaggi di programmazione **poco adatti ai principianti** ponendosi **obiettivi troppo ambiziosi**.



Il giardino dei sogni infranti è lastricato di **deliranti propositi**, figli di ragionamenti del tipo:

"Vorrei creare la prossima app milionaria per smartphone"

quindi

"Comincio a studiarmi il linguaggio Objective-C o JAVA"

oppure:

"Il mio sogno è sviluppare un videogioco come Call of Duty"

allora

"Mi metto a studiare il linguaggio C++"

ed ancora:

"Ho un'idea micidiale per una web-application rivoluzionaria"

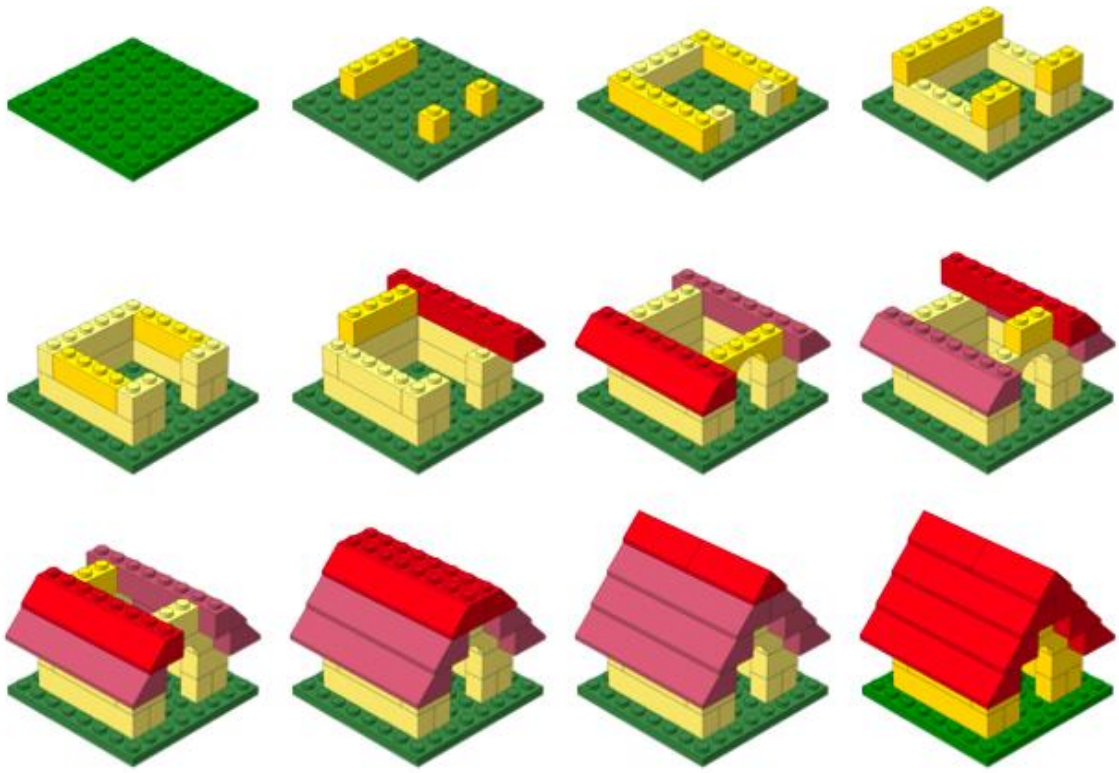
per cui

"Mi imparo per benino il linguaggio PHP"

Lo so, ci sono passato anch'io: abbiamo tutti bisogno di una forte motivazione, **un obiettivo 'scintillante'** che possa darci la carica per metterci a lavorare. Sfortunatamente, **senza delle basi solide** sulle quali poggiare il rischio di cadere e farsi male è elevato. Inutile dire che da più in alto si cade **maggiori sono le probabilità di non rialzarsi**.

La **ricetta** per evitarlo in realtà sarebbe estremamente semplice: partire **apprendendo le basi**, porsi **obiettivi semplici** all'inizio ed **alzare gradualmente l'asticella** mano mano che si migliora e si comprendono i meccanismi che regolano il gioco.
Purtroppo in molti **non realizzano questa banale verità** e si imbarcano subito nello studio di un linguaggio, delle sue librerie ed eventuali framework di contorno: memorizzano un sacco di informazioni su tecnologie e strumenti di sviluppo, e poi iniziano a **copiare e modificare pezzi di codice scaricati dalla rete** pensando che quello voglia dire 'imparare a programmare'.

In realtà stanno tralasciando **competenze fondamentali** come l'abilità di risolvere problemi utilizzando i **mattoncini base** della programmazione. Suona molto come 'tornare alle elementari' lo so, ma è l'unica strada che possa garantire **buoni risultati nel lungo periodo**.



Più avanti ti mostrerò alcuni percorsi che potrai intraprendere per imparare a programmare **partendo dai concetti chiave** e limitando al massimo il 'rumore di fondo tecnologico' che permea questo mondo.

Errore 3 - Prendere la cosa 'troppo seriamente'

Questo è forse **il più subdolo e letale degli errori** che un aspirante programmatore possa fare all'inizio del suo cammino. **Se non c'è divertimento** nell'imparare qualcosa infatti, anche con il miglior maestro e partendo dall'ABC quello che ti aspetta è **un percorso di guerra** dove raggiungere ogni singolo risultato sarà **una vera pena**.

Purtroppo questo è l'approccio mentale di parecchie persone, derivante in parte da convinzioni (in questo caso limitanti) tipo *"la vita non è solo divertimento"* oppure *"è necessaria disciplina, senso del dovere e spirito di sacrificio per ottenere risultati"*.

Ok, chiariamoci bene: non sto affermando che non sia necessaria **disciplina** e che non servano **senso del dovere** e **spirito di sacrificio**. Sono tutte cose importanti. Quello che vorrei ribadire è che se non si trae piacere e non ci si diverte almeno un pochino durante una lezione o un esercizio più che spirito di sacrificio dovremmo avere una **predisposizione al masochismo**.

La domanda che sorge spontanea a questo punto è: **ma come posso imparare a programmare, divertendomi?**

"*E' un grande errore pensare che la gioia della scoperta e della ricerca possano essere promosse da mezzi coercitivi e dal senso del dovere*"

Albert Einstein

Come imparare a programmare divertendosi

Ovvero, la tastiera non è comoda come cuscino!

A rendere piacevole e divertente l'apprendimento contribuiscono in parti uguali lo **studente**, l'**insegnante** e gli **strumenti d'insegnamento:**

Lo studente

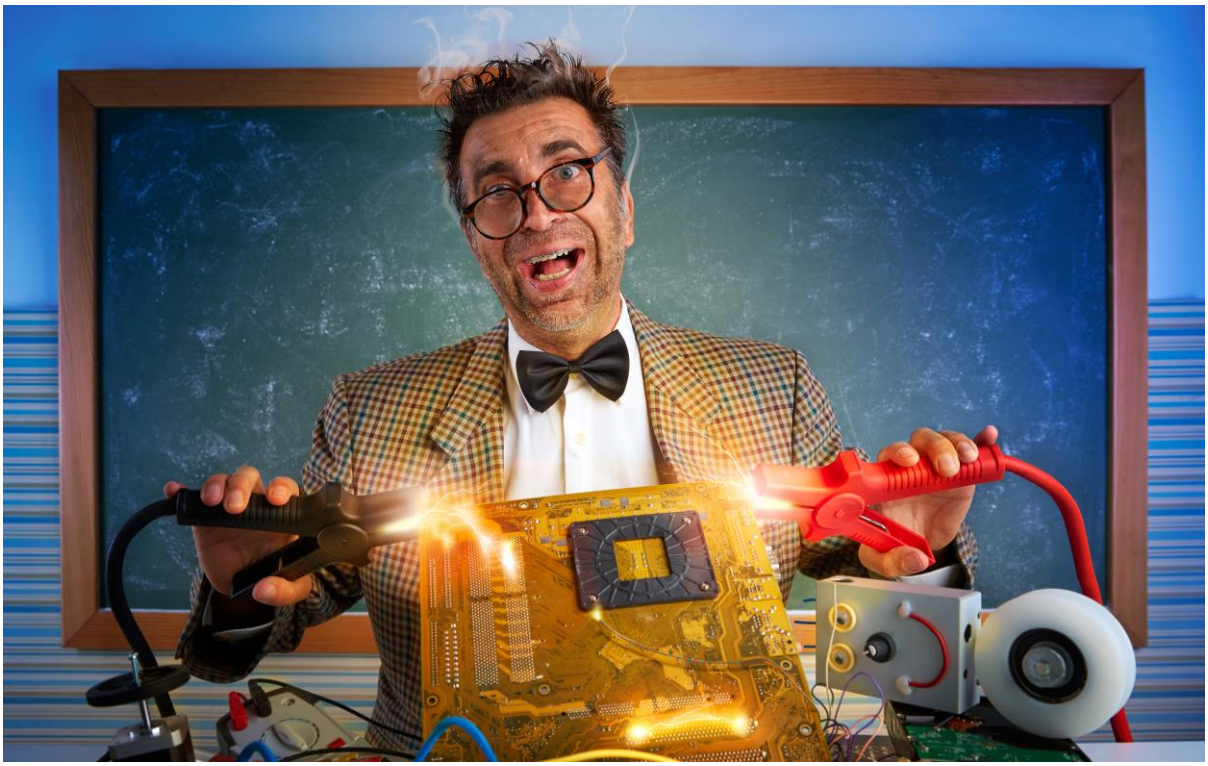
Affinchè imparare possa essere divertente e appassionante è necessario che lo studente sia in uno **stato mentale di apertura, curiosità e fiducia nell'insegnante**. Essere già appassionati della materia può aiutare, ma non è strettamente necessario: se la tua guida è capace e trasmette abbastanza **passione** non trascorrerà molto tempo prima che ti contagi! Se invece ti avvicini alla programmazione con **scarso interesse**, per **obbligo** o peggio ancora nutrendo dei **pregiudizi** nei confronti della materia non sarà tanto facile divertirsi: la responsabilità che l'apprendimento sia un piacere a questo punto cade interamente sul tuo insegnante.

Non voglio contraddirmi: **è sacrosanto essere molto esigenti nei confronti di chi insegna**, troppo spesso si incontrano persone assolutamente non all'altezza di questo delicato compito. E' altrettanto vero che **l'apprendimento non è una strada a senso unico** e per quanto preparato, divertente e carismatico possa essere un docente, se dall'altra parte ci sono troppe resistenze o attitudini negative le probabilità di successo saranno minime.

In sintesi: *un buon punto di partenza per imparare a programmare divertendosi è avere il giusto stato mentale. Sentimenti come curiosità, fiducia, entusiasmo, impazienza di cominciare sono ottimi indicatori che si è sulla buona strada. In caso contrario, pensi di poter fare qualcosa per cambiare questo atteggiamento negativo? Se la risposta fosse NO, è ora di investigare sulle tue reali motivazioni: perchè vuoi imparare a programmare?*

L’insegnante

Dall’altra parte della cattedra, virtuale o reale che sia, l’insegnante ha il potere (e tutta la responsabilità) di rendere le sue lezioni **avvincenti, divertenti ed interessanti** oppure **pesanti, sgradevoli e soporifere.**



Il **principale requisito** che deve avere chi insegna per non risultare di una noia mortale è la **passione per la materia**: se una persona *non ama la programmazione* e non trae il benchè minimo piacere dal creare programmi è impossibile che possa trasmettere emozioni positive ed *è impossibile che possa rendere avvincenti e divertenti le sue lezioni*.

La **passione**, quando si tratta di insegnamento, viene addirittura **prima della competenza**: è inutile essere un *esperto totale, fine conoscitore di ogni particolare e pignolo in ogni dettaglio,* se i contenuti vengono trasmessi in maniera asettica, senza un minimo coinvolgimento emotivo. Google e Wikipedia possono tranquillamente sostituire questo tipo di insegnante.

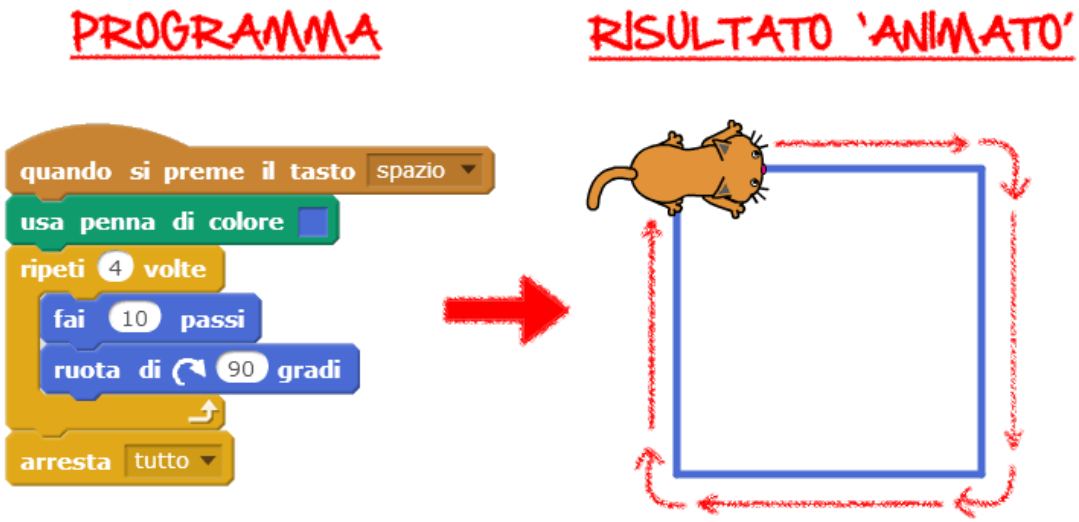
In sintesi: *Sii selettivo. Sincerati che la tua guida sia competente ma soprattutto appassionata. Come fare? Se puoi assisti a qualche lezione prima di scegliere il corso e ‘misura il livello di energia’ che ti viene trasmesso. Parla con altri studenti, cerca su Internet informazioni e verifica che il tuo insegnante abbia effettivamente sviluppato qualcosa. Chi ha una grande passione per la programmazione non potrà mai limitarsi al solo insegnamento, dovrà per forza lavorare o aver lavorato alla creazione di qualche software.*

Gli strumenti d'insegnamento

Anche la scelta degli **strumenti didattici** incide molto su quanto sarà efficace un corso di programmazione. Per strumenti didattici in questo caso intendiamo:

- Ambienti ed i linguaggi di programmazione
- Problemi ed esercizi proposti
- Strumenti per la presentazione dei contenuti

Per quanto riguarda gli **ambienti ed i linguaggi di programmazione**, è importante che in un corso per principianti venga fatta una scelta che privilegi la **semplicità** ed un **feedback immediato** (meglio se visivo, con animazioni e grafica). Giusto per fare un esempio, l'ambiente di sviluppo Scratch consente di creare programmi **costruendoli mediante istruzioni-mattoncino**, osservando l'effetto del programa **in tempo reale** attraverso disegni ed animazioni.



Ad alcuni potrà sembrare infantile e davvero 'troppo poco serio'. In realtà, per veicolare i concetti di base questo approccio è **efficace**, **veloce** ed **estremamente divertente**.

Anche la **tipologia dei problemi** da risolvere può fare la differenza tra un corso divertente ed uno estremamente noioso: assegnare esercizi come ad esempio **disegnare forme, creare animazioni o semplici giochi** è, almeno all'inizio, più stimolante che risolvere problemi di calcolo.

In sintesi: *Se è possibile evita corsi che utilizzino ambienti e linguaggi di programmazione troppo impegnativi. Un tempo non c'erano molte alternative, ci si doveva arrangiare con quel che passava il convento. Oggi però abbiamo a disposizione una vastissima gamma di soluzioni per rendere il nostro primo approccio al coding il più divertente e appassionante possibile. Nel seguito parleremo ampiamente di soluzioni specificatamente studiate per chi parte da zero.*

Alcuni spunti per cominciare

E’ ora di rimboccarsi le maniche!

In questa sezione ti presenterò alcune delle **migliori opzioni** al momento disponibili per imparare a programmare partendo da zero. Per ognuna di queste soluzioni cercherò di evidenziare i **pregi** e gli eventuali **difetti** in modo da aiutarti a scegliere quella **più adatta alle tue esigenze ed i tuoi gusti**.

Dato il **fermento** che caratterizza il settore informatico e l’estrema **velocità** con cui nascono nuove tecnologie, ambienti e linguaggi di programmazione, questo **elenco è lungi dall’essere completo**: se desideri essere aggiornato su tutte le novità relative a questo argomento, ti consiglio di seguirmi sul mio blog, www.programmadazero.com o sulle piattaforme social che preferisci.

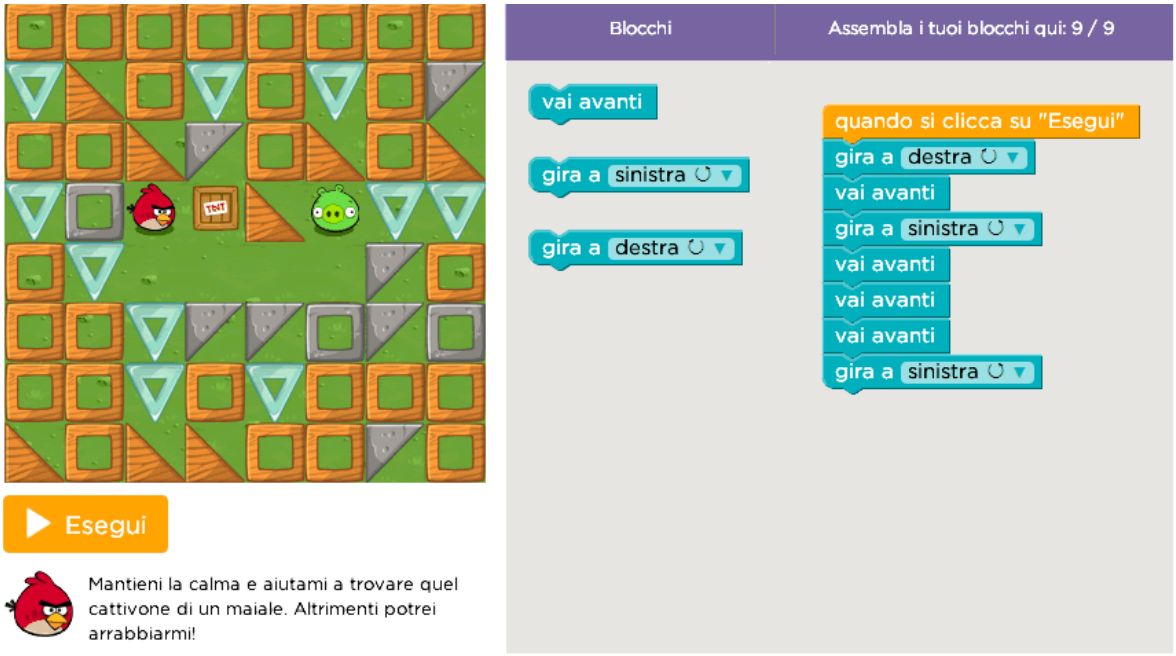
www.code.org

Code.org è un'organizzazione no-profit che mira ad incoraggiare le persone, in particolare gli studenti delle scuole elementari e medie, ad imparare la programmazione (in inglese **coding**). L'iniziativa ha avuto un discreto successo, anche grazie al coinvolgimento di personaggi del calibro di **Bill Gates** (ex-Microsoft), **Mark Zuckerberg** (il creatore di Facebook) e molti altri.

Il sito ufficiale è in continuo aggiornamento, e si arricchisce ogni mese di nuovi esercizi e lezioni interattive. La pagina dalla quale ti consiglio di cominciare per esplorare il sito è: http://studio.code.org

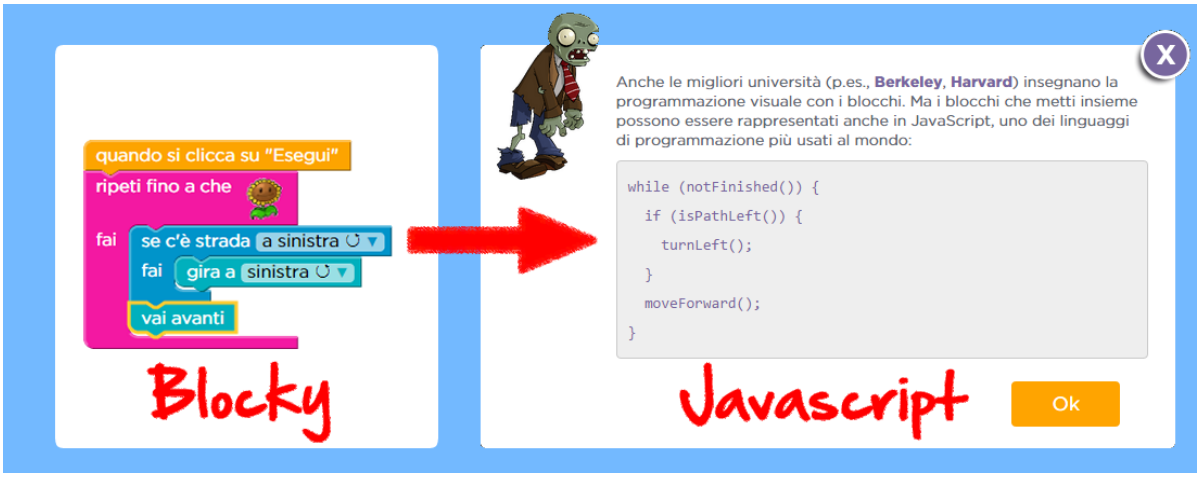
Nel momento in cui sto scrivendo i percorsi disponibili sono parecchi: **quattro corsi base** (destinati principalmente a bambini, di varie fasce d'età) **un corso più completo** adatto a tutti (della durata approssimativa di 20 ore) ed **otto percorsi rapidi** da un'ora per chi avesse veramente poco tempo e volesse farsi un'idea di cosa sia il coding.

Ogni lezione, preceduta da un **breve video** introduttivo (in inglese, ma sottotitolato in varie lingue tra cui l'italiano), ti invita a risolvere dei **divertenti puzzle** utilizzando un semplice linguaggio di programmazione 'a blocchi'. Per coinvolgere maggiormente i più giovani, i protagonisti dei vari puzzle sono presi in prestito da film di animazione famosi o videogame come **Angry Birds**, **Flappy Bird**, **Piante contro Zombie**, **Minecraft**, **Guerre Stellari**, **Frozen**, eccetera.



Chi immaginava che programmare fosse così divertente?

Le lezioni sono **estremamente chiare e curate**, i concetti vengono presentati nel **giusto ordine** e gli esercizi sono appositamente studiati affinchè lo studente li acquisisca con la pratica **divertendosi**. E' presente infine la possibilità di vedere il *'vero codice'* nascosto dietro ai blocchi (per chi non si fidasse che stiamo realmente programmando!). Il linguaggio sottostante si rivela essere **Javascript**, attualmente uno tra i più utilizzati al mondo come primo approccio al coding.



Non solo blocchetti colorati... sotto sotto si nasconde del VERO codice!

PRO

- Tutti i percorsi didattici sono **completamente gratuiti**
- Non si deve installare software sul proprio computer per iniziare, **basta il browser**
- Le lezioni sono tutte **tradotte in italiano**, i video molto ben sottotitolati
- Introduce **tutti i concetti base** della programmazione
- Estremamente **divertente e adatto a tutti**, grandi e piccini
- Prevede un sacco di **mini-corsi** per chi ha veramente poco tempo (**L'ora del codice™**)

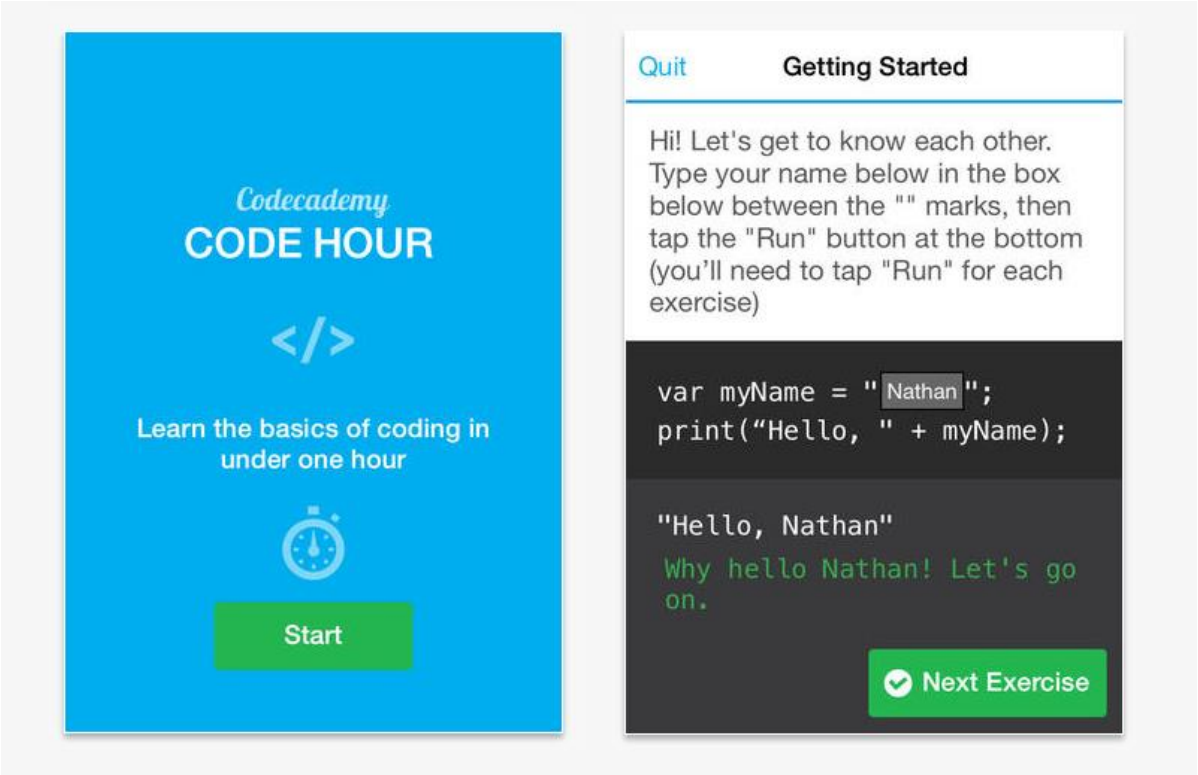
CONTRO

- A qualcuno l'approccio giocoso potrebbe sembrare poco serio
- Non è prevista al momento la possibilità di interagire con un tutor per chiarimenti, domande, etc.

In sintesi: *Code.org è sicuramente uno dei migliori punti di partenza per chi vuole iniziare a programmare. Non fatevi ingannare dall'aspetto giocoso e colorato, dietro uccellini colorati e zombie affamati si nascondono dei veri professionisti dell'insegnamento che hanno creato lezioni ed esercizi perfettamente calibrati su chi parte da zero. Ripeto: anche se vi sembra infantile, seguite il mio consiglio e portate a termine almeno "L'ora del codice™" prima di affrontare qualsiasi altro corso!*

www.codecademy.com

Codecademy, come suggerisce il nome, è una vera e propria 'accademia della programmazione' online. L'offerta di corsi è veramente ampia: **Javascript**, **Python, PHP** e **Ruby** sono solo alcuni tra i principali linguaggi affrontati, e sulla piattaforma trovano spazio anche corsi su **HTML** e **CSS** (per chi fosse orientato più allo sviluppo di pagine web). Al momento attuale più di 25 milioni di studenti in tutto il mondo hanno seguito uno dei corsi di Codecademy, e la cifra è destinata ad aumentare grazie al recente lancio di una App che consente di provare un breve corso introduttivo dal proprio dispositivo mobile.



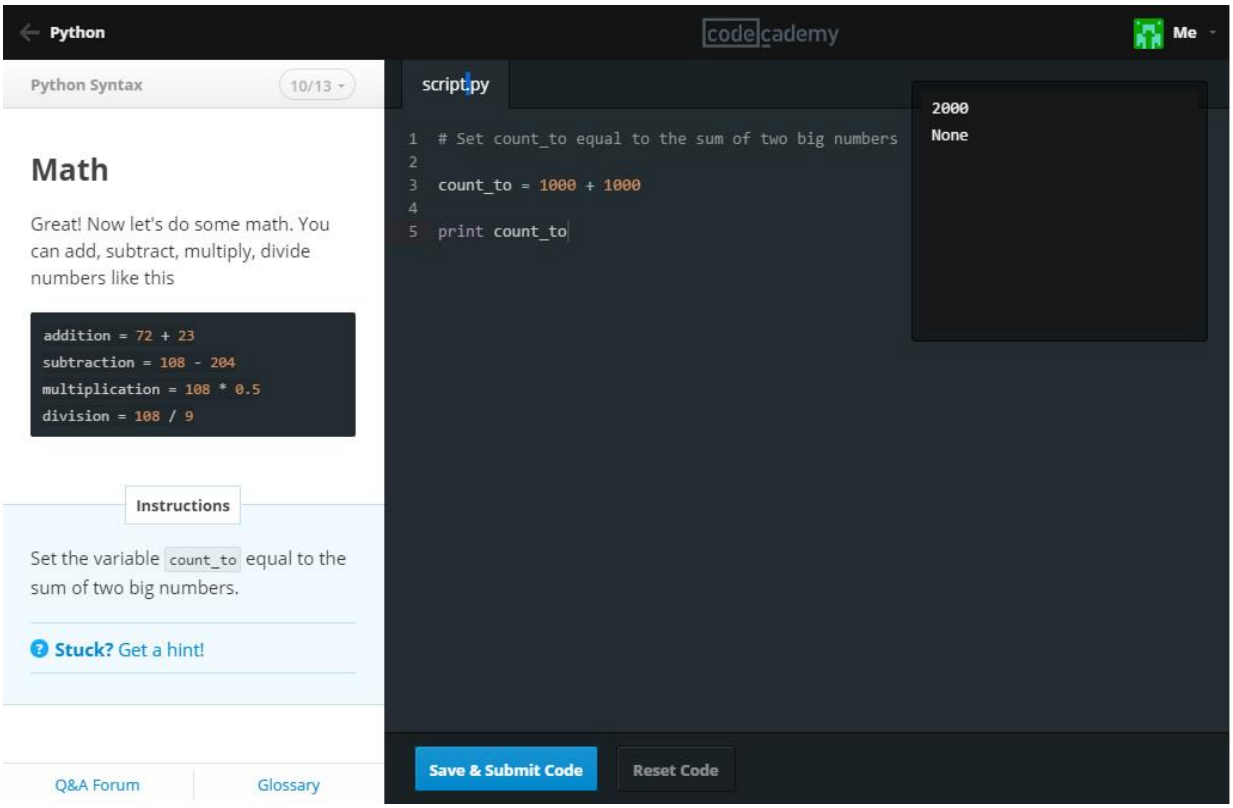
Un assaggio di cos'è la programmazione con l'app per smartphone "CODE HOUR"

I corsi sono al momento **tutti gratuiti** e, a quanto dicono i fondatori del progetto, lo saranno per sempre. L'approccio scelto per insegnare la programmazione è **estremamente pratico:** ogni lezione infatti ci invita a provare immediatamente i concetti esposti scrivendo piccoli **pezzi di codice**, premiandoci in caso di successo attraverso un simpatico **sistema di punteggi e medaglie** (badge).

L’interfaccia dei corsi è molto semplice ed include:

a. Un’area in cui ci viene spiegato un concetto (in genere una brevissima lezione)
b. La descrizione di un esercizio da svolgere
c. Una zona dove poter scrivere la soluzione dell’esercizio
d. Una finestra dove osservare i risultati dei nostri sforzi :)

In caso di dubbi è possibile richiedere un **suggerimento** (hint) alla piattaforma oppure accedere al **forum di discussione** dedicato alla specifica lezione: dato il gran numero di studenti è molto facile ricevere aiuto e spiegazioni aggiuntive (spesso dagli stessi creatori del corso).



L’interfaccia permette di provare in tempo reale il programma, verificandone i risultati

PRO

- I corsi sono **completamente gratuiti**
- Non si deve installare software sul proprio computer, **basta il browser**
- Presenta una vasta gamma di linguaggi, quasi tutti adatti a chi comincia
- Offre la possibilità di **ricevere aiuto** e dialogare con istruttori ed altri studenti
- E’ presente una **app gratuita** con un corso rapido introduttivo

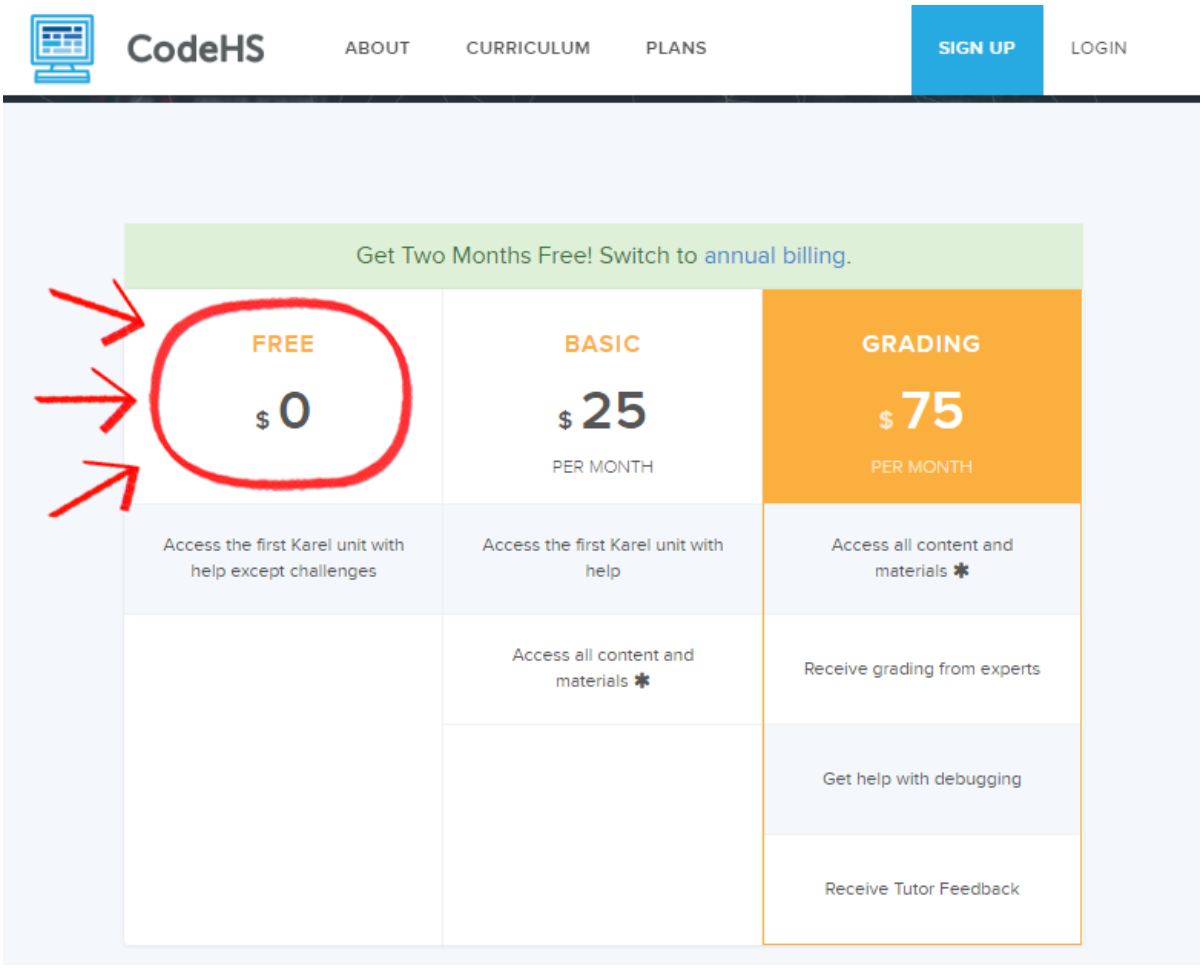
CONTRO

- Per adesso tutti i corsi sono in **inglese**
- Assenza di videolezioni, tutto il materiale didattico è **testo da leggere**
- Look **un po' serioso**, forse non adattissimo ai più giovani

In sintesi: *Codecademy tra le risorse per iniziare a programmare è forse la più professionale. Offre tanti corsi gratuiti su molti linguaggi 'veri' presentando i concetti in modo graduale ma immediatamente applicabile in un contesto reale. Se siete proprio allergici ai blocchi colorati o vi sentite un po' troppo maturi per giocare con le costruzioni, Codecademy è il punto di partenza che mi sento di consigliare… ovviamente a patto che conosciate l'inglese!*

codehs.com

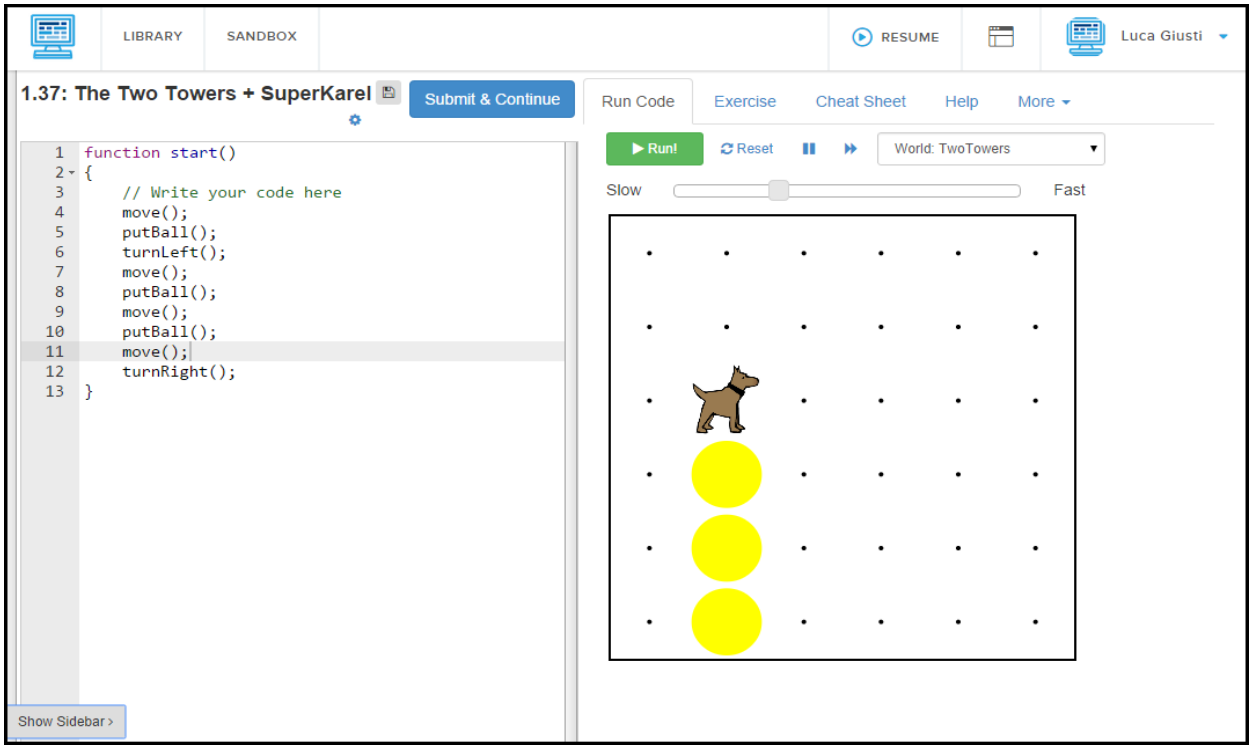
CodeHS è una piattaforma di code learning molto evoluta ideata da due studenti della prestigiosa università di Standford. Orientata principalmente verso i linguaggi **Javascript** e **Java** al momento presenta vari livelli di registrazione destinati principalmente alle scuole **uno gratuito e due a pagamento** ed un sistema di **crediti** per poter accedere a materiale aggiuntivo. I corsi sono suddivisi in moduli di **difficoltà crescente**, il primo dei quali particolarmente adatto a chi parte da zero:



I tre livelli di abbonamento alla piattaforma CodeHS

Il mini-corso **Programming with Karel**, incluso nella sottoscrizione gratuita, presenta le basi della programmazione incoraggiando lo studente ad usare **Javascript** per istruire un simpatico cagnone (Karel) a fare degli esercizi. Il percorso è costituito da brevi **video-lezioni** intervallate da **sessioni pratiche** dove si viene invitati a programmare.

L’interfaccia in questi casi è suddivisa in due aree, una dedicata alla **scrittura del codice** ed un’altra dove è possibile **osservare in tempo reale l’effetto** che le istruzioni hanno sul nostro amico a quattro zampe. L’approccio, giocoso ed immediato, risulta molto efficace e stimolante ed introduce in maniera graduale ai moduli successivi (decisamente più impegnativi, ma ugualmente divertenti).



Karel esegue esattamente gli ordini impartiti dal programma scritto a sinistra

PRO

- I corsi disponibili sono **estremamente ben fatti** ed adatti a chi comincia
- Non si deve installare software sul proprio computer, **basta il browser**
- E’ prevista la possibilità di **ricevere assistenza** ed avere un **tutor** personale

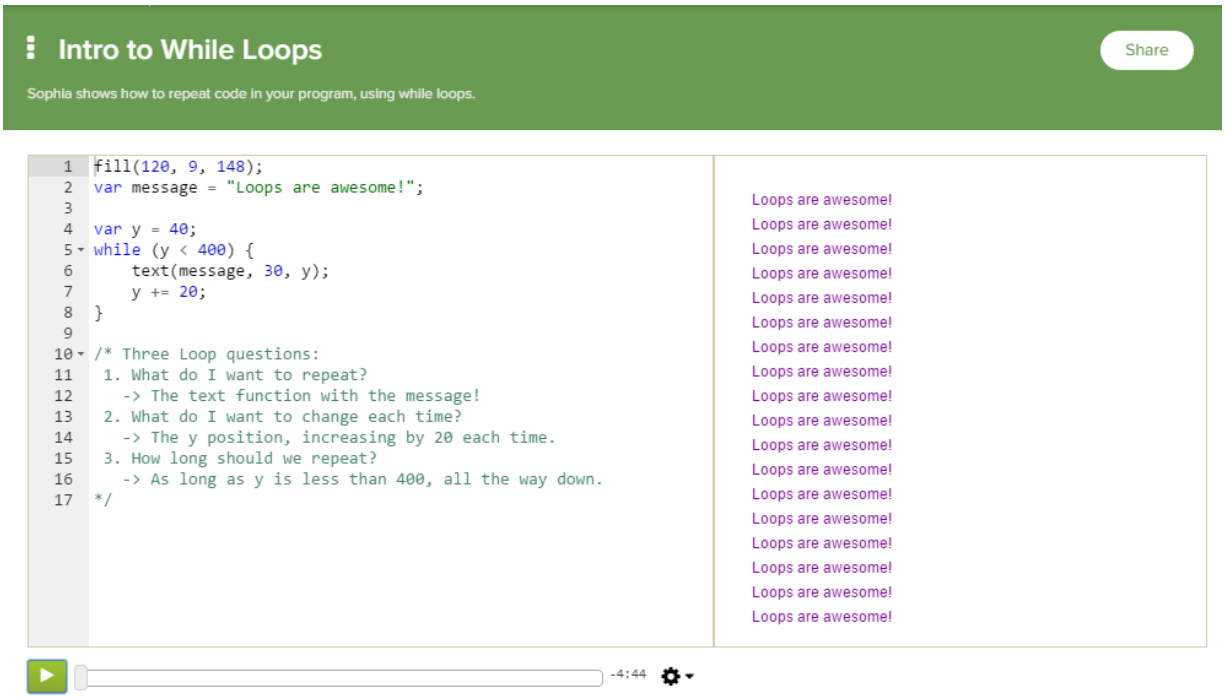
CONTRO

- Per accedere a tutti i contenuti è necessario un **piano a pagamento**
- I servizi di assistenza e tutoraggio sono eccezionali, ma **abbastanza costosi**
- Tutto il materiale è in lingua **inglese**

In sintesi: *CodeHS è una piattaforma particolarmente evoluta che offre ottimi corsi ed un servizio di assistenza/tutoraggio di livello professionale. Se volete avere un esperto al vostro fianco (ed avete un po’ di soldi da spendere) è sicuramente una delle migliori opzioni. In caso contrario, consiglio comunque di seguire il corso gratuito “Programmare con Karel”, estremamente ben fatto ed ottimo come punto di partenza.*

www.khanacademy.org

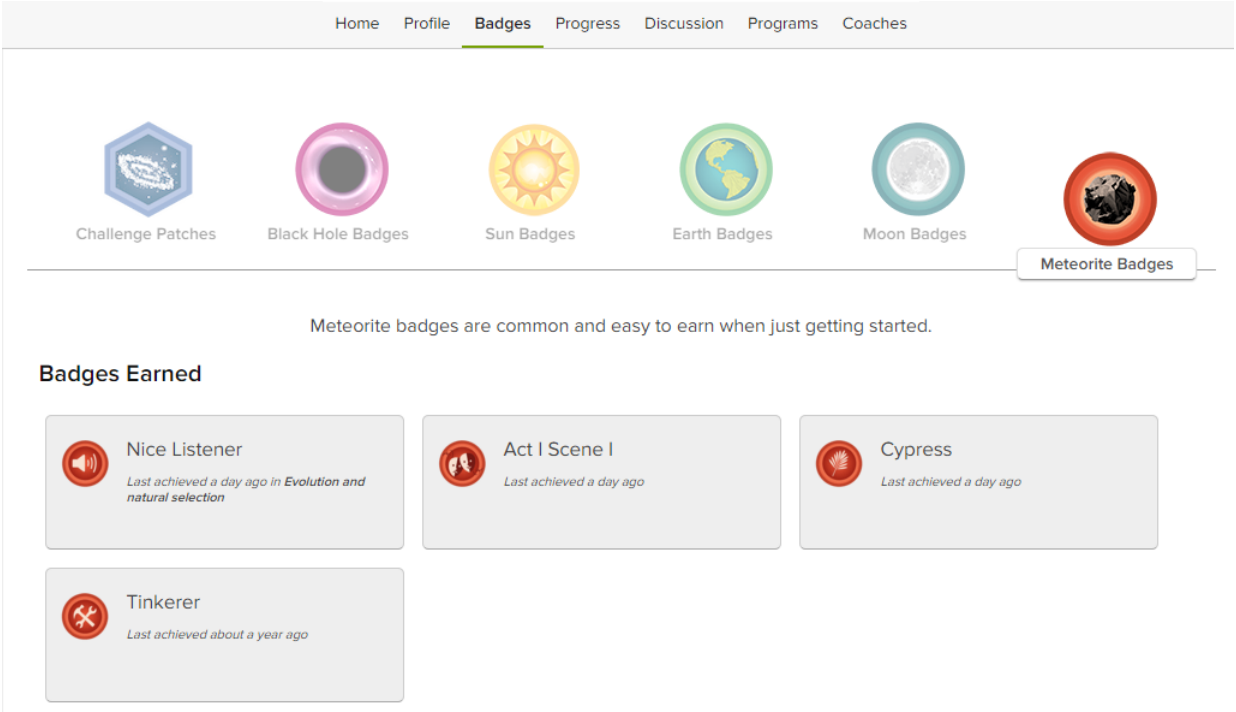
Khan Academy è un'organizzazione educativa senza scopo di lucro che ha lo scopo di offrire servizi e materiali gratuiti per l'istruzione e l'apprendimento a distanza attraverso tecnologie di e-learning. Tra le numerose discipline trattate, spicca un'interessante **serie di video tutorial** sulla programmazione. Il linguaggio utilizzato per introdurre i concetti di base è **Javascript,** le lezioni sono completamente interattive ed è possibile addirittura modificare e fare esperimenti in tempo reale sul codice che digita l'istruttore!



Il codice del programma scritto a sinistra, viene fatto eseguire a destra

Anche se l'approccio è molto giocoso (i compiti che ci verranno affidati sono incentrati principalmente sullo scrivere programmi che creano disegni) vengono forniti tutti i concetti chiave della programmazione oltra a delle robuste basi di Javascript, un linguaggio REALE e che ha numerose applicazioni pratiche e professionali. Sono presenti anche dei corsi introduttivi alle tecnologie Web (HTML e CSS) ed approfondimenti sugli algoritmi e la teoria dell'informazione. Analogamente ad altre piattaforme (come Codecademy ad esempio) presenta tutta una serie di

medaglie ed obiettivi da sbloccare, in pieno stile videogame, che contribuiscono a rendere lo studente più motivato e l'apprendimento divertente.



Premi, medaglie e missioni rendono imparare estremamente avvincente

PRO

- Tutti i corsi sono **estremamente ben fatti** e **completamente gratuiti**
- Non si deve installare software per iniziare, **basta il browser**
- Introduce **tutti i concetti base** della programmazione
- Ad ogni lezione è associato un **canale social** dove gli utenti possono commentare, **fare domande** ed osservazioni agli istruttori

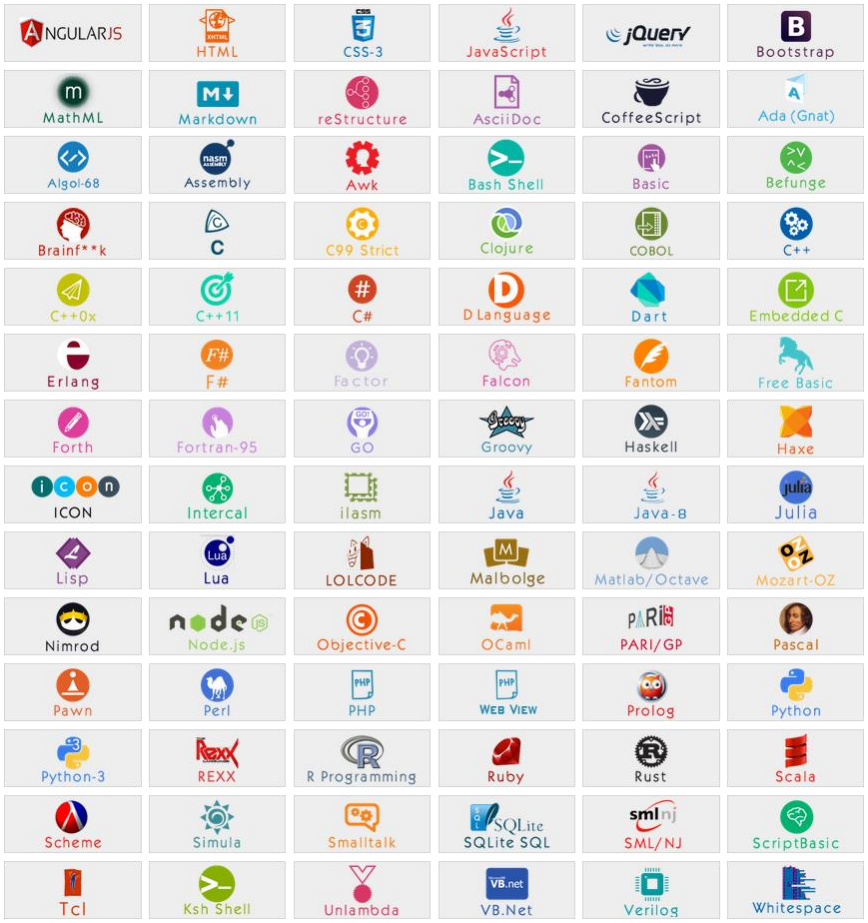
CONTRO

- Solo una parte del materiale è tradotto/sottotitolato in italiano

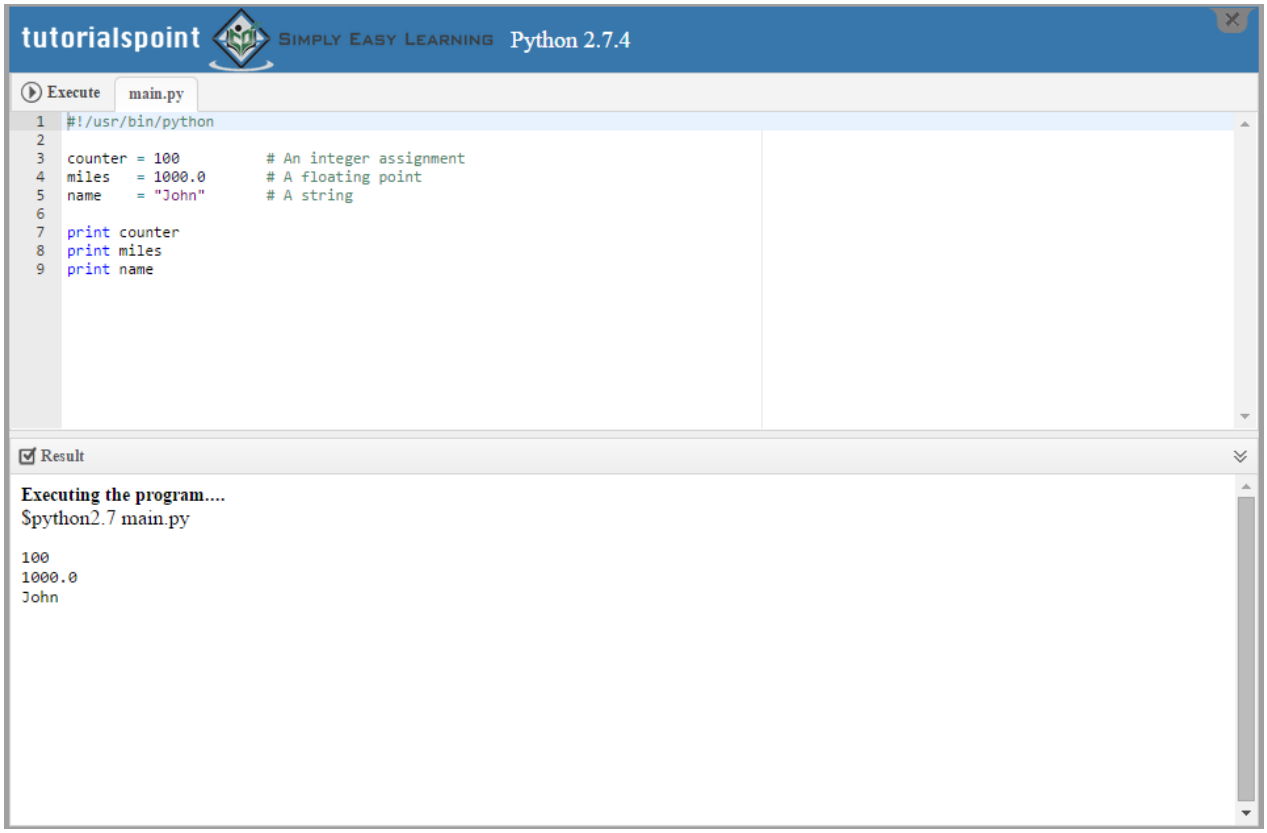
In sintesi: *Khan Academy è una soluzione eccellente per chiunque voglia muovere i primi passi divertendosi ed al contempo imparare un linguaggio molto diffuso. Anche se il target per cui sono pensate le lezioni è quello degli studenti di scuola media e superiore, la solidità e la professionalità con cui vengono esposti i materiali didattici li rendono adatti a chiunque voglia avvicinarsi al coding.*

www.tutorialspoint.com

Tutorialspoint, diversamente dalle piattaforme presentate finora è un portale orientato principalmente a **guide e tutorial da leggere**. Non sono quindi presenti video, animazioni o slideshow ma solo un insieme molto ben organizzato di pagine web sulle più disparate tecnologie e linguaggi di programmazione. L'**enorme mole e varietà di contenuti** è impressionante, così come **la cura e la qualità** di ogni singolo tutorial.



Altra caratteristica molto interessante è l'integrazione di ogni corso con la piattaforma **Coding Ground**, che consente di eseguire i programmi di esempio e fare esperimenti senza lasciare il browser e senza dover installare ambienti di sviluppo sul proprio computer.



Coding Ground consente di scrivere codice e vederne subito l'effetto all'interno del browser

PRO

- Tutorials molto professionali e **completamente gratuiti**
- Esiste un tutorial per **praticamente qualsiasi linguaggio** di programmazione
- La piattaforma Coding Ground consente di svolgere gli esercizi suggeriti dai tutorial **senza dover installare compilatori, interpreti o ambienti di sviluppo**
- Ad ogni linguaggio e argomento è collegato un canale di un **forum di discussione** molto attivo e ben moderato

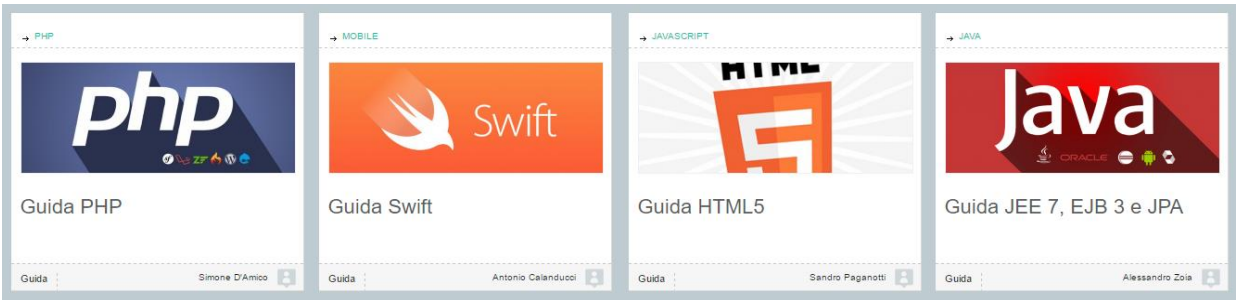
CONTRO

- Materiali esclusivamente **in inglese**
- Può risultare più ostico delle soluzioni viste in precedenza

In sintesi: *Tutorialspoint è una miniera d'oro, ricca di contenuti di qualità. Prima di cominciare ad esplorarla però, è altamente consigliato affrontare almeno uno dei corsi introduttivi di piattaforme come Codecademy, CodeHS o Khan Academy sulle basi della programmazione.*

http://www.html.it/programmazione/guide/

HTML.IT, portale originariamente dedicato all'omonimo linguaggio per la creazione di pagine web (ed a tutte le tecnologie correlate) è al momento quanto di più simile a Tutorialspoint ci possa essere in lingua italiana. Contiene guide (sempre testuali, quindi da leggere sul browser) su tutti i principali linguaggi di programmazione, tra i quali Java, Javascript, Python, Ruby, C/C++ e molti altri. Le guide sono ben organizzate ed i contenuti, complessivamente, di buona qualità.



Alcune delle numerose guide sulla programmazione di HTML.it

PRO

- Guide molto professionali e **completamente gratuite**
- Ogni argomento è collegato ad un canale di un **forum di discussione** dove interagire con gli istruttori e gli altri studenti
- Completamente **in italiano**

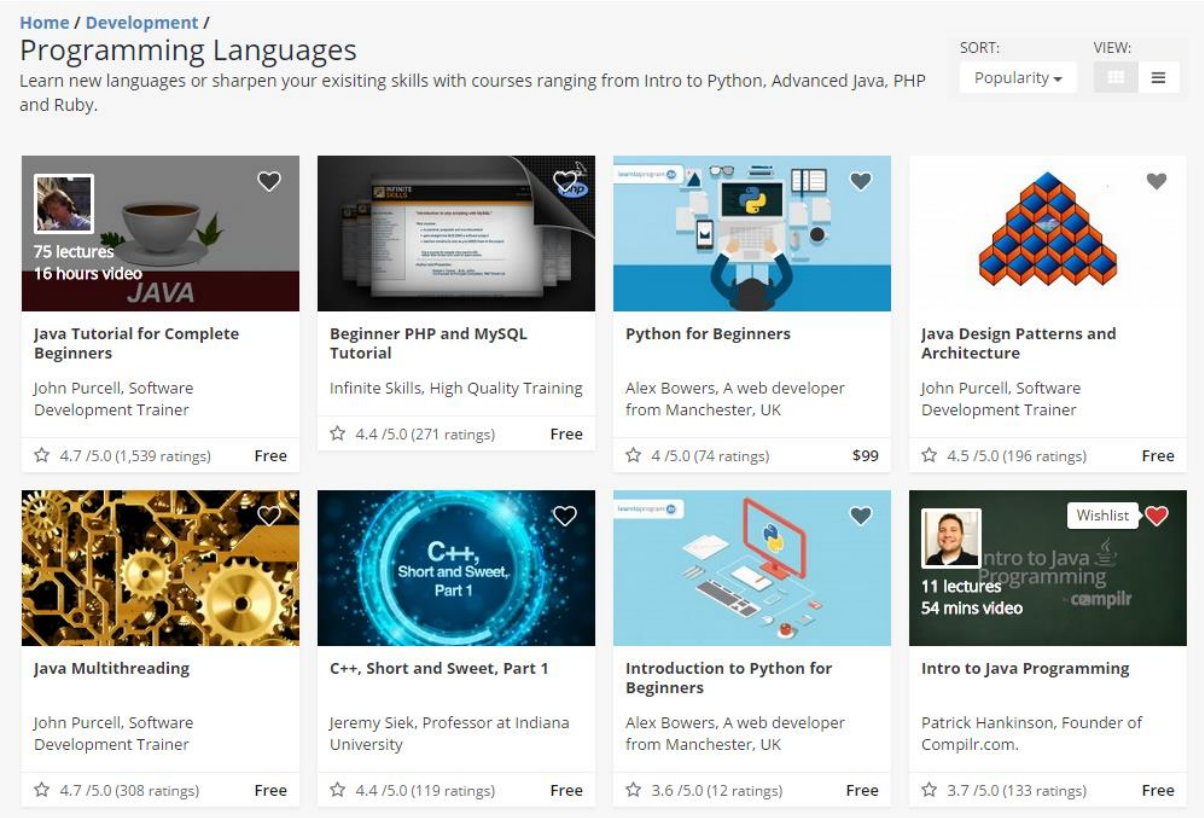
CONTRO

- Più **orientato ai linguaggi** che alle abilità di programmazione
- Difetta di integrazione con una piattaforma per il live coding da browser

In sintesi: *Ottimo compromesso per chi non conosce l'inglese e volesse imparare alcuni tra i più famosi linguaggi di programmazione. La mancanza della piattaforma di live coding può essere compensata aprendo una sessione di Coding Ground in un'altra scheda del browser, in modo da poter fare esercizi senza installare altro software. Valgono le stesse considerazioni fatte per Tutorialspoint in merito al livello dei contenuti non proprio 'da zero': è consigliato affrontare prima un corso introduttivo su piattaforme come Codecademy, CodeHS o Khan Academy.*

www.udemy.com

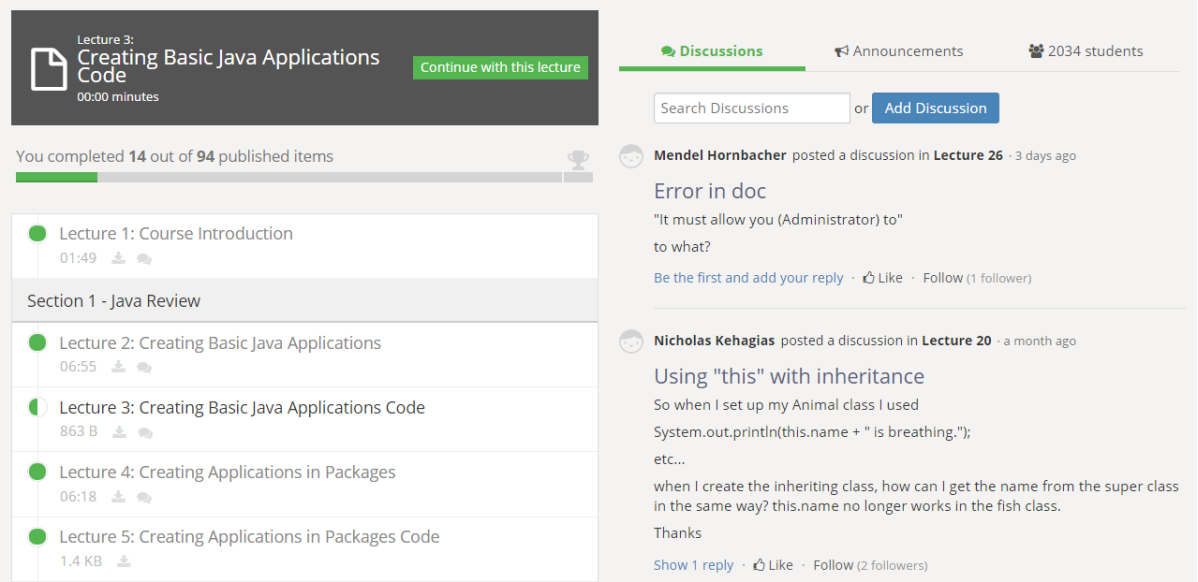
Udemy è una piattaforma di e-learning che ospita una grande quantità di **videocorsi** sugli argomenti più disparati. La sezione dedicata alla programmazione in particolare è **estremamente ricca**.



I corsi possono essere ricercati ed ordinati per popolarità, prezzo o numero di studenti

Udemy è un vero e proprio **market-place**: offre sia corsi **gratuiti** che a **pagamento** ed un meccanismo molto efficace di **rating e recensioni** attraverso il quale gli utenti contribuiscono a mettere in risalto i **corsi migliori** e gli **istruttori più bravi e disponibili**.

Oltre ai video, la piattaforma consente agli istruttori di includere dei **test di verifica**, **esercizi**, codice sorgente delle lezioni e molto altro. Gli studenti possono **fare domande** sulle singole lezioni e ricevere chiarimenti e spiegazioni aggiuntive.



L'interfaccia dei corsi presenta indicatori di progresso per l'intero corso e le varie unità didattiche

Esistono moltissimi **corsi introduttivi** che, anche se dedicati a specifici linguaggi di programmazione, spiegano abbastanza bene le basi della programmazione. Ecco alcuni spunti per cominciare ad esplorare la piattaforma:

Corso completo e gratuito su **Java**: http://www.udemy.com/java-tutorial
Corso completo e gratuito su **PHP**: www.udemy.com/php-mysql-tutorial
Corso gratuito su **C++:** www.udemy.com/cpp-short-and-sweet

PRO

- Presenta un **enorme varietà** di corsi ed una **comunità molto attiva**
- Il processo di revisione dei corsi garantisce **standard qualitativi superiori**
- L'**interazione con gli istruttori** è integrata nella piattaforma
- Sistema di **rating molto efficace**

CONTRO

- Buona parte dei corsi è solo **in inglese**
- Al momento non integra uno strumento per il live coding interattivo

In sintesi: *Se non si è spaventati dall'inglese, Udemy è una preziosa risorsa per chi cerca videocorsi di qualità sui linguaggi di programmazione. Inestimabile è la possibilità di interagire con gli istruttori, facendo domande e discutendo con altri studenti in merito alle varie unità di apprendimento. Il sistema di valutazioni e recensioni garantisce inoltre che solo i corsi migliori ed i docenti più bravi vengano messi in evidenza. Anche se molti dei contenuti sono appositamente creati per i principianti, resta comunque valido il consiglio di seguire prima uno dei percorsi interattivi su Code.org, Codecademy, CodeHS o Khan Academy.*

www.youtube.com

YouTube, la famosa piattaforma di self-broadcast di Google, ospita un sacco di videolezioni dedicate alla programmazione ed ai linguaggi di programmazione. Anche se la maggior parte ha caratteristiche **molto amatoriali**, con un po' di pazienza e ricerche mirate è possibile portare alla luce delle vere e proprie **miniere d'oro** che poco hanno da invidiare a corsi professionali di analoghe piattaforme a pagamento.

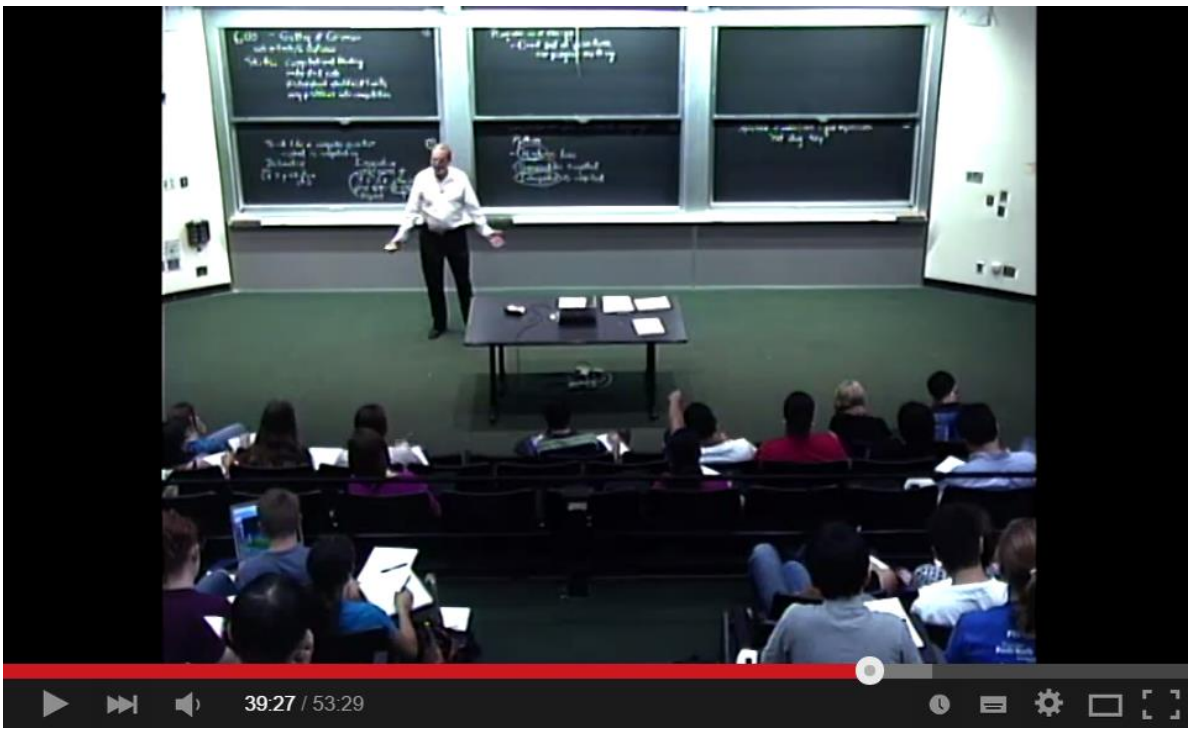
Tra i vari canali disponibili ce n'è uno in particolare, **completamente in italiano,** che mi sento di consigliare vivamente a chi si avvicina per la prima volta alla programmazione.

E' quello del **Prof. Camuso**, un docente di Informatica che ha reso disponibile **gratuitamente** centinaia di videolezioni su linguaggi e tecnologie più disparate: https://www.youtube.com/user/fcamuso. Estremamente preciso nelle spiegazioni, Camuso ha confezionato delle playlist complete sotto ogni aspetto e mai superficiali.

Una delle tante videolezioni sul Java del Professor Camuso

Altre risorse molto interessanti (ma in lingua inglese) sono anche le videolezioni introduttive dei corsi di informatica del rinomato **Massachusetts Institute of Technology (MIT)** e della prestigiosa **Stanford University:**

Introduction to Computer Science and Programming (MIT)
Computer Science 101 (Stanford)



Lezioni di informatica al MIT

PRO

- E' presente un **mole immensa** di contenuti, **anche in italiano**
- Completamente **gratuito**

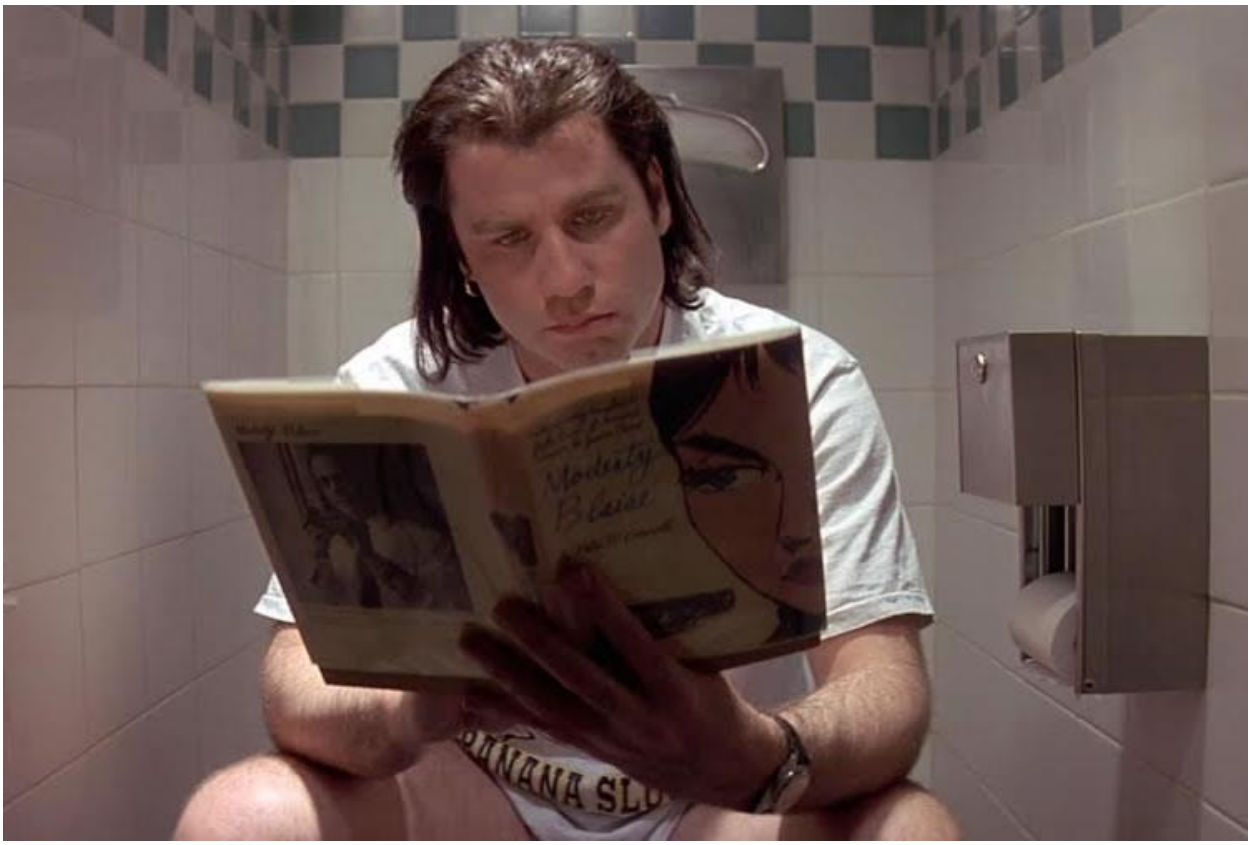
CONTRO

- Può risultare **difficile scoprire le perle** in un oceano di contenuti mediocri
- Anche se previsto dalla piattaforma, non è garantito che gli istruttori forniscano assistenza ed aiuto agli studenti

In sintesi: *A patto di armarsi di pazienza e non scoraggiarsi durante la ricerca, YouTube può riservare delle grosse sorprese: alcune videolezioni sono di qualità indiscutibile, e possono contribuire molto ad aiutare chi muove i primi passi nel mondo della programmazione. E' comunque da considerarsi una risorsa complementare, da affiancare magari ad una piattaforma di live coding come Coding Ground per mettere in pratica i concetti appresi.*

Ma un buon libro?

ovvero, 'La carta conserva sempre il suo fascino'



Siamo più o meno tutti d'accordo: videocorsi, corsi interattivi e tutorial online sono strumenti meravigliosi, accelerano l'apprendimento e rendono la programmazione più accessibile e divertente.

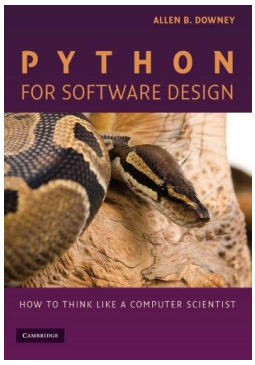
Questo non significa però che stumenti più *old-style* siano da deprecare: un libro in molte circostanze e per molte persone è una **soluzione ugualmente valida**.

Sebbene esistano degli schemi generali, ogni persona impara in modo diverso dall'altra: mentre alcuni risultano molto stimolati dall'aspetto interattivo di un corso online altri, magari di indole più riflessiva o più inclini a metabolizzare le informazioni prima di metterle in pratica, potrebbero beneficiarne meno. Inoltre quando gli argomenti si complicano e diventano impegnativi, studiare su un browser (con distrazioni 'social' ad un solo click di distanza...) può diventare faticoso.

Chiaramente essendo la programmazione un **abilità pratica**, il libro non può sostituire completamente l'esperienza che si può ottenere scrivendo del codice su un computer: è buona norma quindi (come quasi tutti gli autori di testi informatici consigliano) alternare **sessioni di lettura e studio** a momenti di programmazione vera e propria, dove **fare esercizio** risolvendo piccoli problemi legati agli argomenti studiati più di recente. Detto questo, vorrei precisare che i tre testi consigliati sono

stati scelti in quanto focalizzati sull'**imparare a programmare** piuttosto che sull'imparare un nuovo linguaggio di programmazione.

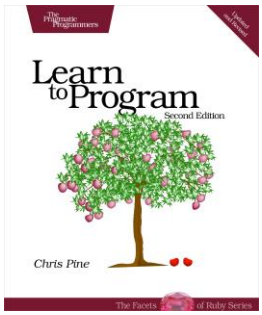
How to Think Like a Computer Scientist, *di Allen B. Downey*



Un testo pensato appositamente per chi comincia, molto ben scritto e comprensibile. Sfrutta il linguaggio Python per presentare tutti i concetti chiave della programmazione.

Link Amazon
Link alla versione web gratuita
Link alla traduzione in italiano

Learn to Program, *di Chris Pine*



Anche questo è un testo pensato per un pubblico completamente a digiuno di concetti di programmazione. Usa Ruby come linguaggio per presentare in maniera chiara e semplice tutti gli argomenti di base.

Link Amazon
Link alla versione web gratuita
Link alla traduzione in italiano

Learn to Program with Scratch, *di Majed Marji*

Utilizzando Scratch, un simpatico linguaggio di programmazione che utilizza blocchi colorati invece di istruzioni testuali, l'autore introduce in maniera graduale il lettore nel mondo della programmazione. Adatto a tutte le età.

Link Amazon
Ebook gratuito

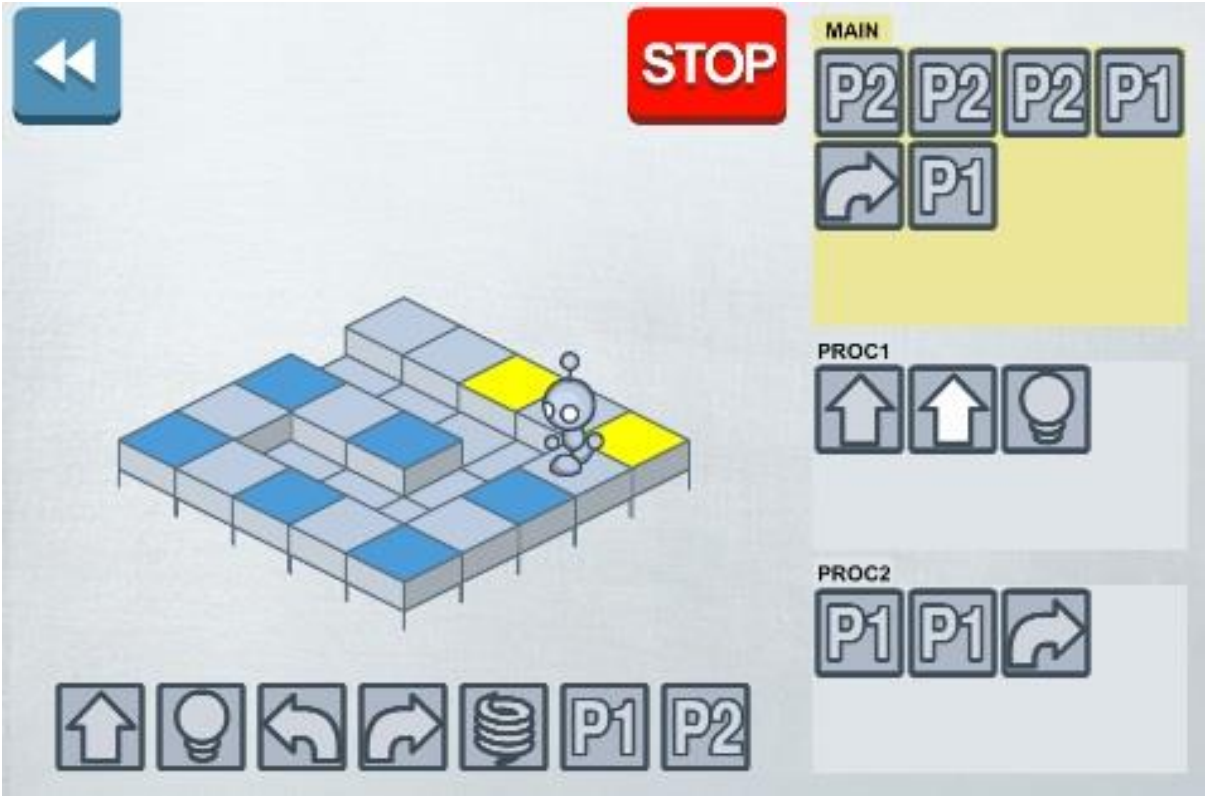
La via del gioco

Giocando si impara meglio

In questa sezione vorrei presentarti alcuni ‘simpatici giochini’ ideati e sviluppati con il duplice obiettivo di **divertire** ed **insegnare** i concetti alla base del **pensiero computazionale**. Integrare queste attività ludiche in un percorso più articolato può sicuramente portare vantaggi migliorando ed accelerando l’apprendimento.

http://www.lightbot.com

Lightbot è un bellissimo Puzzle Game dove dovremo aiutare un simpatico robottino a superare una serie di livelli di difficolta sempre crescente. Invece di pilotare direttamente il personaggio come avviene solitamente in questo genere di gioco, il giocatore dovrà utilizzare delle azioni elementari (*vai avanti*, *gira a destra*, *salta*, *accendi la luce,* eccetera) per **costruire dei programmi** che se eseguiti da **Lightbot** lo aiutino a raggiungere ed **illuminare** alcune **speciali mattonelle**.



Eseguendo il programma, Lightbot cercherà di illuminare tutte le mattonelle blu

Lightbot è disponibile in varie versioni: web, desktop (Windows e Mac) e per dispositivi mobile (sia iOS che Android). Esiste anche una versione a pagamento (circa 2 €) con moltissimi livelli e funzionalità aggiuntive.

http://www.codecombat.com

Code Combat, ambientato in uno scenario Fantasy, ci sfida a scrivere del vero codice per aiutare degli eroi a svolgere varie missioni, come fuggire da un sotterraneo, uccidere dei mostri o raccogliere tesori. Tra i molti linguaggi supportati troviamo oltre ai notissimi *Javascript* e *Python* anche i meno noti (ma pur interessanti) *LUA* e *Clojure*.

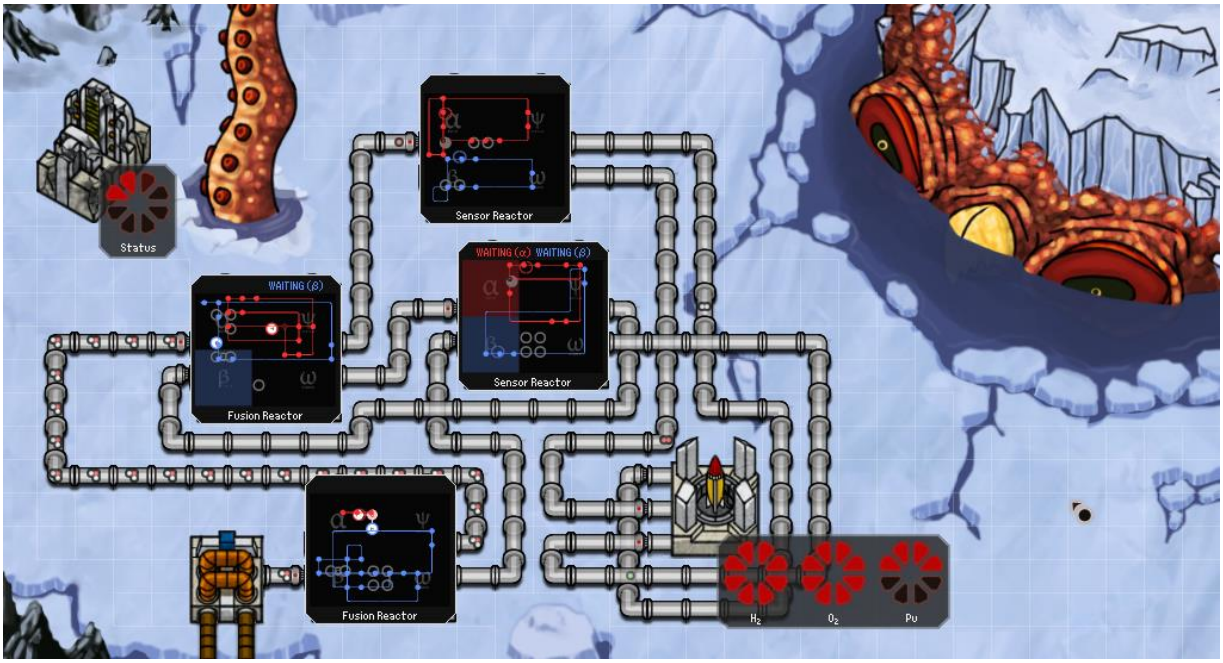
In questo livello il nostro eroe dovrà sopravvivere e recuperare tutte le gemme. Saprai aiutarlo?

Man mano che progrediamo nel gioco, i nostri eroi acquistano **nuove abilità**, equipaggiamenti e potenziamenti che si traducono in **nuove funzioni** e costrutti per la scrittura del **codice di controllo**. Anche se disegnato appositamente per stimolare un pubblico giovane (dai 10 anni in su), il gioco si rivela adatto e divertente per ogni fascia di età.

Code Combat è attualmente disponibile come Browser game ed anche in versione iPad. Non è chiaro se gli sviluppatori stiano o meno lavorando ad una versione Android.

http://www.zachtronics.com/spacechem/

Spacechem, tra tutti i giochi presentati è sicuramente il più complesso e sfidante. Disponibile per PC Windows, Mac e Linux, Spacechem mette il giocatore al comando di una futuristica **raffineria spaziale** che deve produrre composti chimici e sostanze necessarie a fronteggiare i pericoli della **colonizzazione spaziale**.



Questa piovra gigante minaccia la nostra raffineria spaziale!

Per creare questi materiali a partire da **elementi chimici di base**, il giocatore deve costruire dei 'circuiti di trasformazione' trascinando elementi/comandi sulla griglia di gioco. Anche se non insegna propriamente alcun linguaggio di programmazione reale, progredendo attraverso i vari livelli di Spacechem il giocatore **impara moltissimi concetti di base del pensiero computazione**, come la sequenza di istruzioni, la ripetizione la selezione. Nei livelli più avanzati inoltre, dovendo gestire più linee di trasformazione contemporanee, il gioco ci presenta anche concetti avanzati come il multithreading e la sincronizzazione di processi. Anche se nel lungo periodo riesce a regalare al giocatore una buona predisposizione al problem-solving ed al pensiero computazionale, Spacechem è da considerarsi un **passatempo molto impegnativo,** sicuramente non adatto ai più piccoli e che potrebbe risultare scoraggiante per molti.

http://thefoos.com/

The Foos è un simpaticissimo gioco di piattaforma dove grazie ad istruzioni elementari (tipo salta, avanza, ruota o attacca) dobbiamo aiutare dei **buffi personaggi** a completare varie missioni, che vanno dal recupero di gustose ciambelle alla fuga da perfidi mostriciattoli.

Caccia alla ciambella... appena due passi a destra ed è fatta!

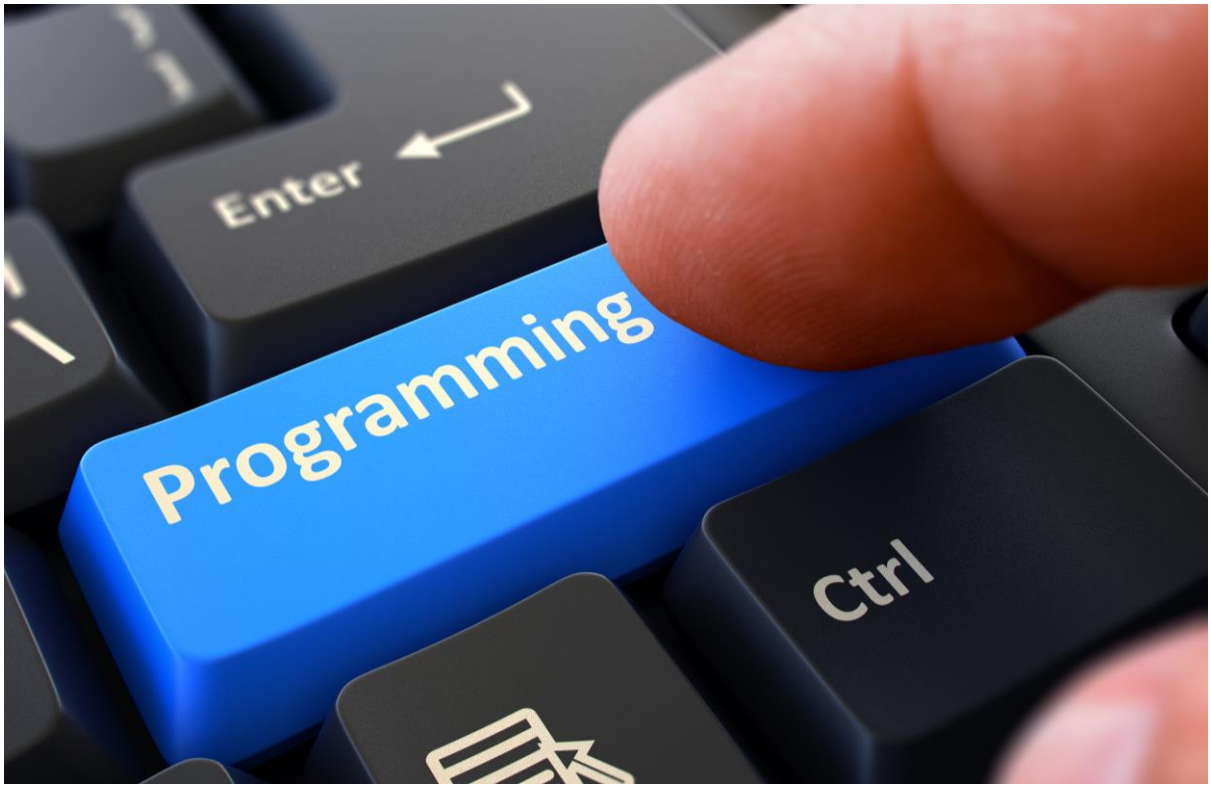
Oltre alle istruzioni di base, man mano che si progredisce nel gioco vengono sbloccate tutte le **strutture di controllo** classiche della programmazione (cicli e selezioni) in modo da poter risolvere i livelli/puzzle con **veri e propri programmi visuali**.

Ripetiamo tre volte: piazza la dinamite ed aspetta!

Nonostante The Foos sia pensato principalmente per un **pubblico di giovanissimi**, può risultare molto divertente ed istruttivo per **aspiranti programmatori di tutte le età**. Disponibile in molte versioni, sia mobile (iOS, Android e Kindle) che desktop (solo per Mac OS X), è presente anche una versione Web giocabile attraverso Browser.

Agisci adesso!

Piani d'azione in comode confezioni



Nonostante la ricchezza di spunti proposti, qualcuno potrebbe essere ancora confuso: a volte avere troppe opzioni tra le quali scegliere può avere l'effetto collaterale di **rallentare** se non **bloccare le decisioni**.

Per questo vorrei suggerirti alcuni **piani d'azione** da seguire alla lettera e con cieca fiducia, in modo da 'rompere il ghiaccio' e cominciare ad **agire subito!**

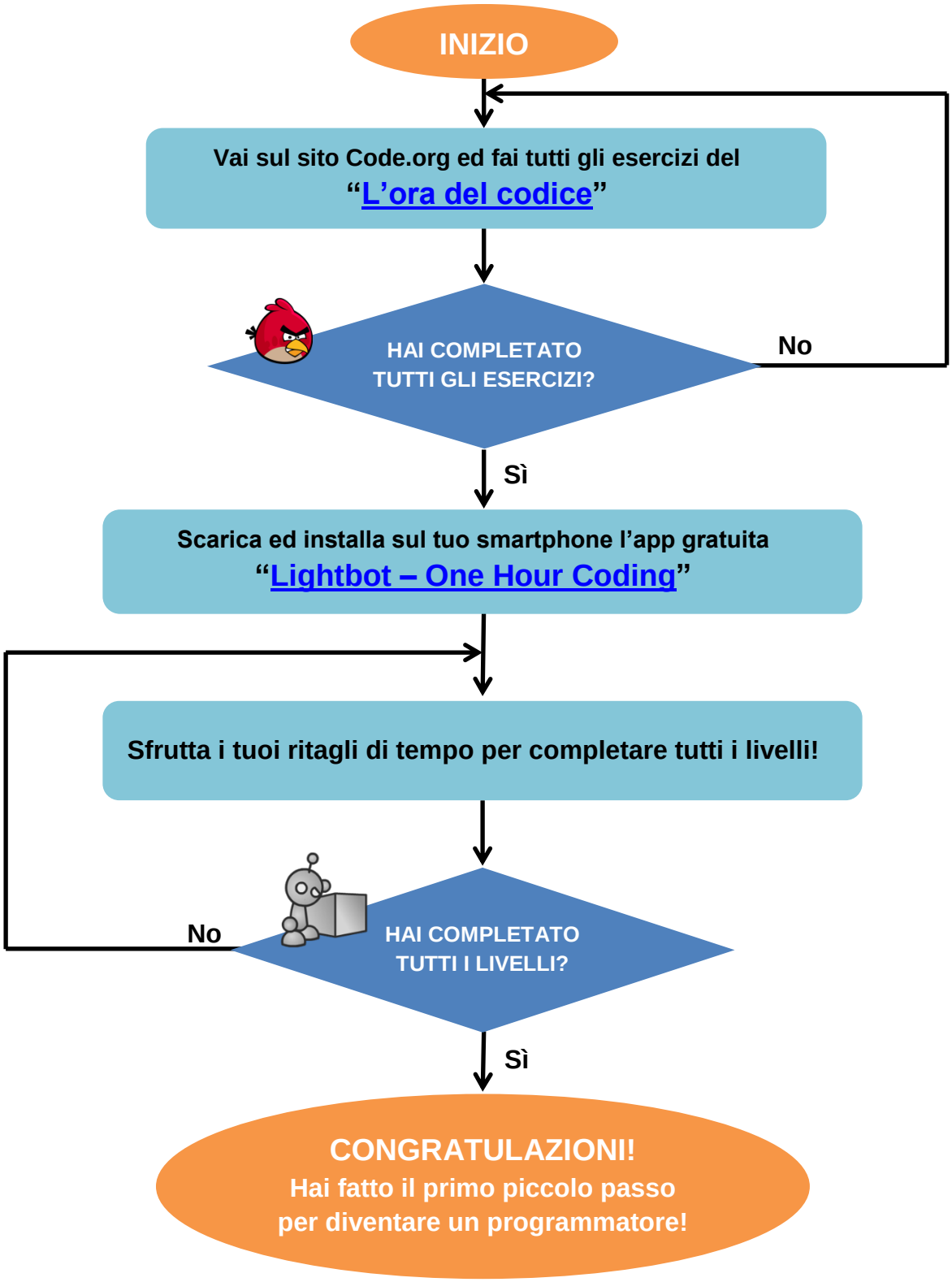
Per indicarti i passi da seguire, userò uno tra i più antichi strumenti adoperati dagli informatici: i diagrammi di flusso. I diagrammi di flusso sono dei **semplici schemi** per rappresentare una sequenza di istruzioni da seguire per portare a termine un compito.

Blocchi colorati, frecce, titoli ed etichette ti forniranno una sorta di **mappa**, per cominciare a muoverti verso la direzione giusta e l'obiettivo di imparare, finalmente, a programmare!

Pronti, partenza... VIA!

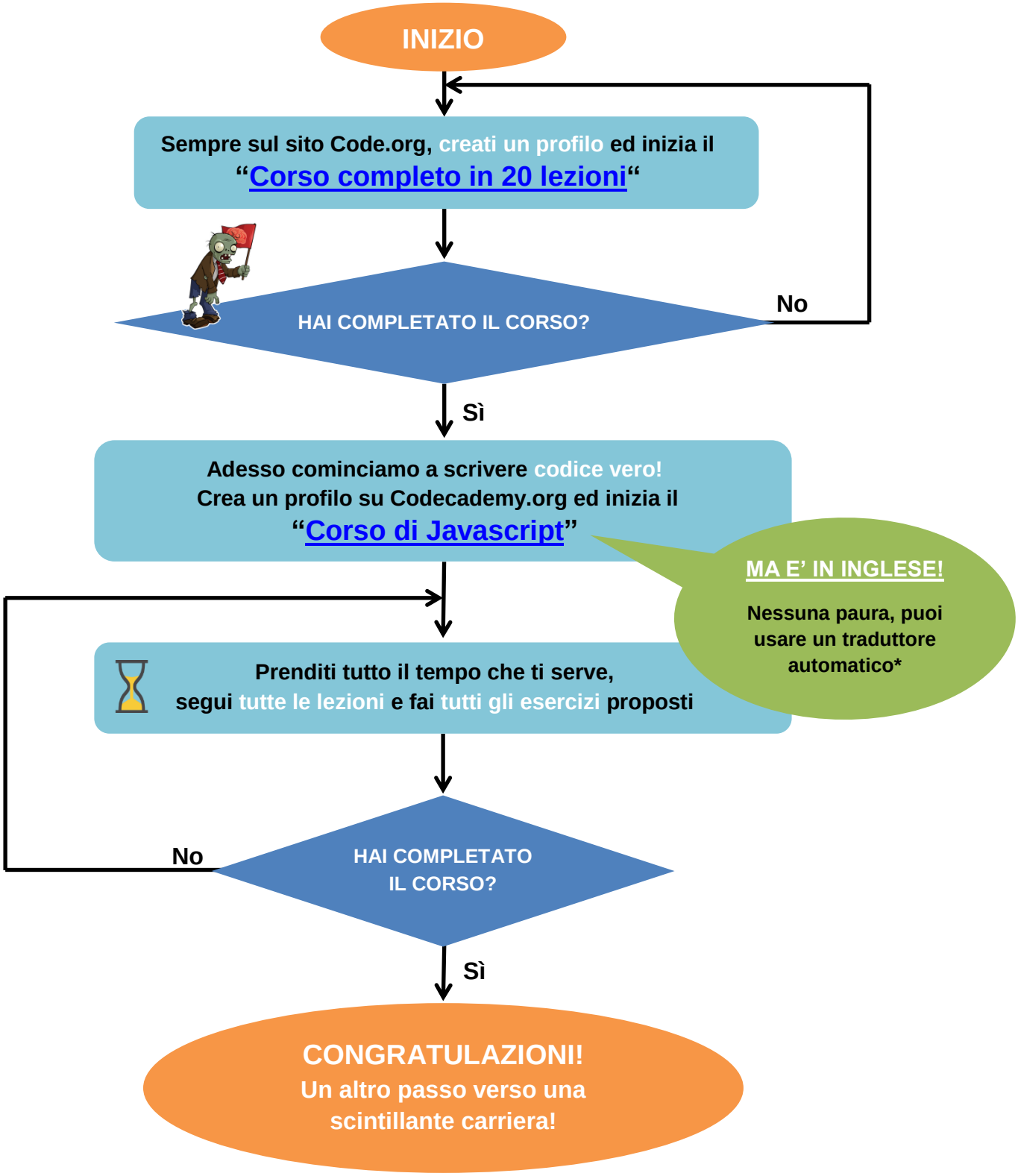
Piano d'azione N.1

Per rompere il ghiaccio



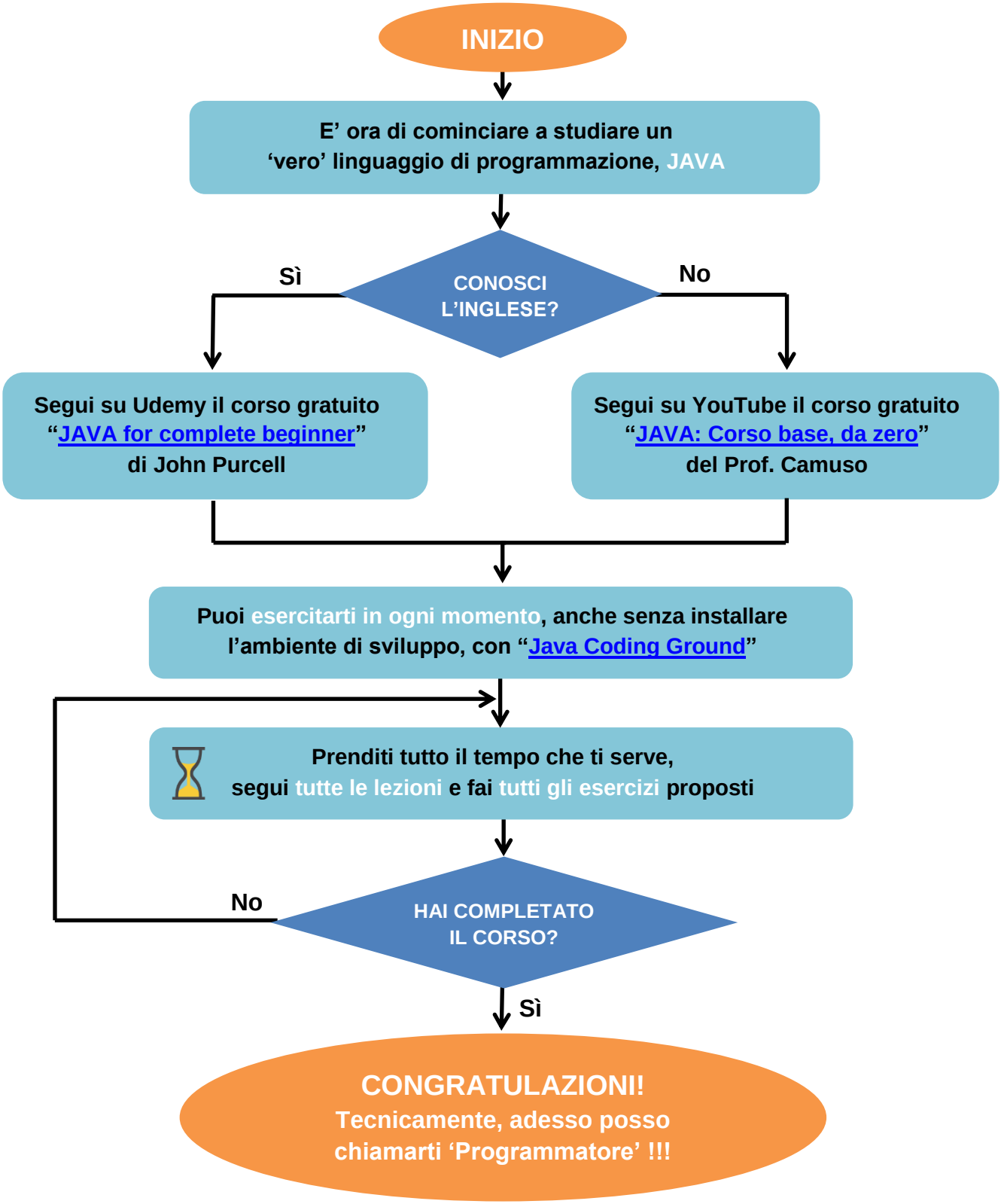
Piano d'azione N.2

Le cose si fanno serie adesso... ma non troppo!



Piano d'azione N.3

Quando il gioco si fa duro...



* Do you speak english?

Ovvero, l'italiano è la lingua più bella del mondo!

Non sarò il primo (e neanche l'ultimo) a ribadire come **l'inglese sia fondamentale** per chi ha intenzione di fare seriamente il **mestiere dello sviluppatore** software. E non solo perchè la maggior parte dei corsi per imparare a programmare sia in questa lingua. L'importanza è dettata principalmente dal semplice ed inconfutabile fatto che **ogni novità**, ogni aggiornamento, ogni espansione di questo tumultuoso mondo viene resa disponibile prima (ed a volte solamente) **in lingua inglese**.

Detto questo, è ugualmente vero che per **apprendere nuove abilità** disporre di materiale didattico nella propria **lingua madre** rappresenta un **enorme vantaggio** ed accelera notevolemente l'apprendimento.

Uno strumento che potrai utilizzare per sopperire in parte alla mancanza di materiale in italiano, è un **traduttore automatico**. Inizialmente abbastanza approssimativi, al giorno d'oggi questi oggetti hanno raggiunto una **qualità della traduzione più che discreta** (soprattutto da e verso l'inglese).

Il traduttore automatico di pagine web, se non è integrato direttamente nel browser come per Chrome ed Internet Explorer, può sempre essere installato come plugin aggiuntivo (Firefox, Google Translator Plugin; Safari: Transalte Plugin).

Una raccomandazione: questo genere di strumento non riesce a fare distinzione tra il testo inglese di **una spiegazione** ed il testo inglese di **un programma**. Dovete quindi essere sempre consapevoli che **il codice sorgente dei programmi non va mai tradotto** in italiano.

Se l'istruzione per stampare un messaggio sullo schermo è **PRINT**, non è corretto utilizzare la versione italiana **STAMPARE**. Sarebbe bello avere un linguaggio di programmazione che sfrutti per le parole chiave l'Italiano, ma purtroppo al momento non esiste.

Nel seguito cercherò di fornirti una piccola guida **all'uso consapevole e proficuo del traduttore** automatico di Chrome, in modo che tu possa beneficiare il più possibile delle risorse gratuite che ti ho consigliato.

Guida all'uso consapevole del traduttore automatico

Se lo conosci, non ti uccide!

Supponiamo di voler cominciare a studiare il linguaggio Python su una delle fantastiche guide di Tutorialspoint, e supponiamo di saperne davvero poco di inglese.

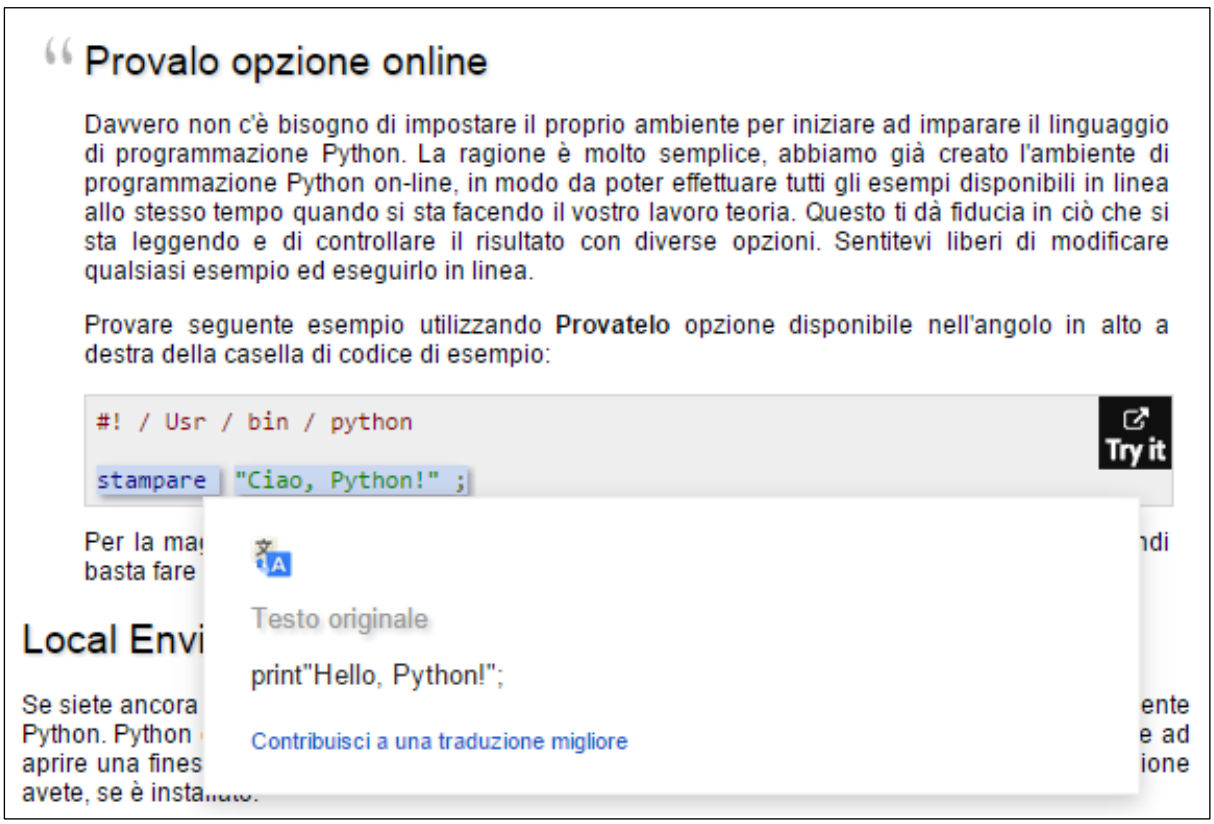
Il **primo passo** da fare sarà quello di premere il **tasto destro del mouse** sulla pagina da tradurre e selezionare la voce "**Traduci in italiano**":

Proprio un bel tutorial, ma in inglese! Non per molto ancora...

Come per magia, tutto il testo verrà tradotto in un battito di ciglia nella nostra amata lingua madre! Su altri browser il procedimento è analogo (ad esempio per Internet Explorer la voce del menù corrispondente è "Traduci con Bing" e così via).

Che bello l'italiano!

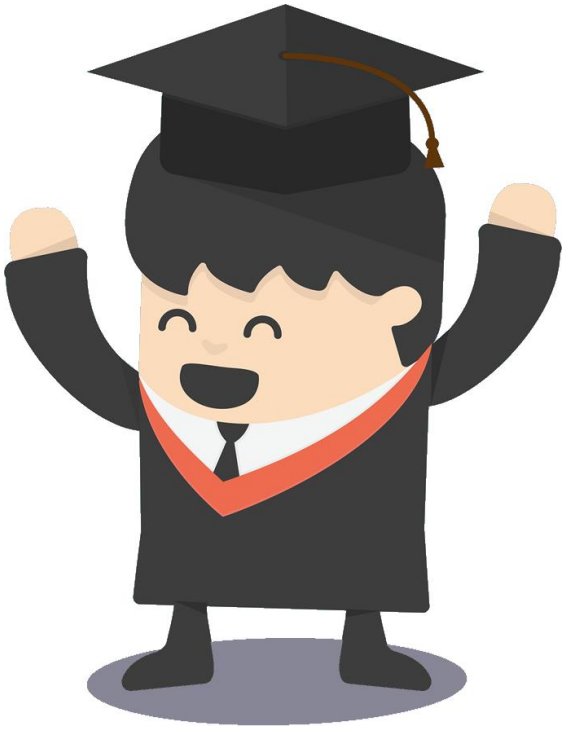
Se si hanno dei **perplessità sulla traduzione** (ad esempio perché è stato tradotto il codice sorgente di un programma) posizionando il **mouse sopra una frase** sarà possibile osservare in una finestrina **la versione originale**:



Questo è un esempio di codice sorgente, ed il traduttore ci mostra l'originale

Congratulazioni!

La fine del principio



Se sei giunto alla fine di questo modesto '*Manuale di sopravvivenza dell'aspirante programmatore*' significa che finalmente hai vinto l'inerzia ed **hai fatto la prima mossa**, la più difficile.

Certo il percorso è ancora lungo, e sicuramente non in discesa. Fortunatamente, una volta costruite delle **buone e solide basi** non ci sarà obiettivo che tu non possa raggiungere!

I suggerimenti e gli strumenti didattici che ti ho presentato, se opportunamente sfruttati, possono aiutarti a sviluppare quel '**pensiero computazionale**' che rappresenta **le fondamenta** di ogni buon programmatore.

Il mio ultimo consiglio è dunque quello di **non correre** ed indugiare il più possibile sullo **sperimentare**, **comprendere** ed **assimilare** tutto ciò che le cosiddette 'Risorse per i principianti' hanno da offrirti.

Posso assicurarti che non te ne pentirai, e l'investimento darà presto i suoi frutti.

In bocca al lupo!

Rimaniamo in contatto?

Se hai gradito questo piccolo manuale, desideri restare in contatto con me ed essere costantemente aggiornato su notizie e risorse per imparare a programmare, ti invito a seguirmi sul mio blog:

www.programmadazero.com

O se preferisci, attraverso i più noti social network:

https://www.facebook.com/programmadazero/

https://twitter.com/programmadazero

https://plus.google.com/+Programmadazero

https://www.youtube.com/+Programmadazero

Infine, per chiarimenti, critiche o richieste particolari puoi contattarmi direttamente all'indirizzo di posta elettronica:

luca@programmadazero.com