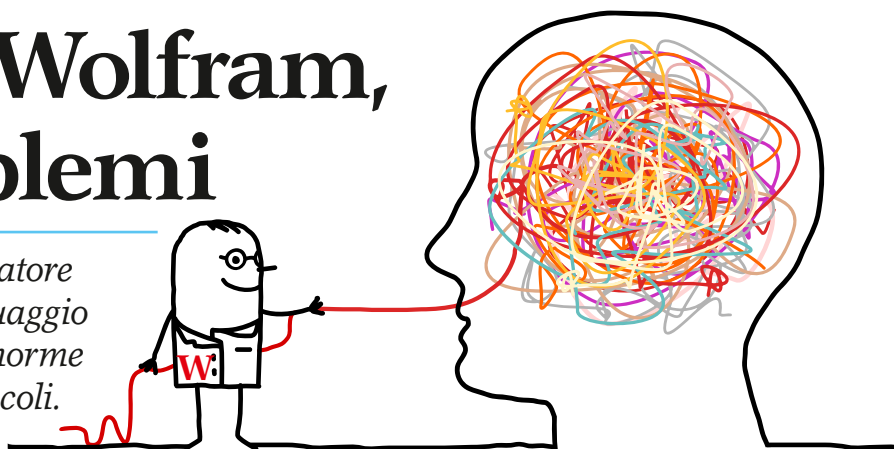


## Mi chiamo Wolfram, risolvo problemi

*Dall'autore di Mathematica e creatore di Wolfram Alpha un nuovo linguaggio di programmazione. Un potere enorme a disposizione di computer minuscoli.*



Stephen Wolfram è stato un bambino prodigio: nato nel 1959, a tredici anni stava già riordinando i suoi appunti sulle particelle elementari. Wolfram è stato anche un adulto prodigio, nel '79 era professore al Caltech e si è dedicato alla ricerca nel campo della fisica quantistica, è anche uno fra i pochissimi nerd con uno spiccato senso degli affari. Negli anni '80 crea Mathematica: un software per il calcolo simbolico. Una rivista ne approfitta per uno di quei titoli che vengono così bene in Inglese: "Physics whiz goes into the biz" (Mago della fisica si dà agli affari). Avanti veloce fino ai giorni nostri e ritroviamo su Business Insider un altro di questi titoli, una di quelle esagerazioni

che sembrano uscite da un quotidiano degli anni '50: "Computer Genius Builds Language That Lets Anyone Calculate Anything" (genio del computer crea un linguaggio che permette a chiunque di calcolare qualunque cosa). Un titolo fantastico, che fa ricordare bionde casalinghe ricciolute armate di Kitchen Aid, reggiseni a punta, tacchi a spillo e automobili con le pinne.

### Arriva Wolfram Language

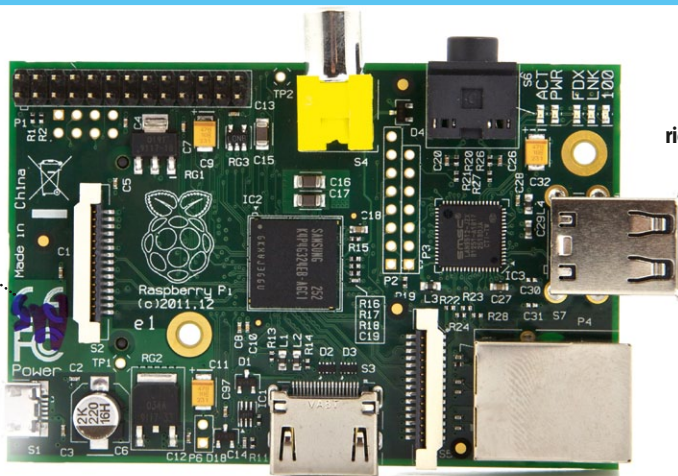
Un nuovo linguaggio è certamente qualcosa che fa incuriosire, ma quello che ci ha fatto definitivamente ruotare le antenne nella direzione di Wolfram Language sono stati due post

particolarmente stuzzicanti. Il primo, nel blog di Wolfram, comunicava che il nuovo linguaggio sarebbe stato gratuitamente a disposizione degli utenti del Raspberry Pi, che è un delizioso moduletto, delle dimensioni di una carta di credito, armato di un Arm, che arriva in una scatola che ricorda una confezione di caramelle alle vitamine.

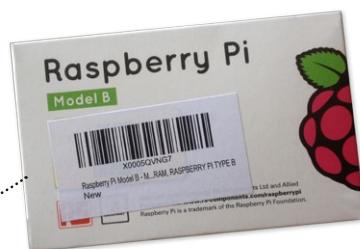
Il secondo, apparso su Gamasutra e su diversi altri siti dedicati ai giochi, annunciava l'integrazione del nuovo linguaggio con il framework per la creazione di giochi Unity.

Una frase in particolare ci ha fatto entrare in risonanza: "quando inizi a programmare, sai quello che hai intenzione di fare, ma nemmeno dopo mezz'ora

Il dettaglio della scheda, con i connettori per video, tastiera, mouse, rete e alimentazione



La scatola del Raspberry Pi ricorda una confezione di caramelle vitaminiche



All'interno si nasconde il piccolo computer.



stai cercando di risolvere un problema totalmente diverso, che è stato creato dal processo che hai iniziato quando hai cercato di fare qualcosa che avrebbe dovuto essere semplice fin dall'inizio". In altri termini, quante volte ci mettiamo alla tastiera, diciamo per mettere in ordine i conti di casa e finiamo per perderci in qualcosa di altro: che so, nel disegno di una struttura di database con cache dinamica in array di array associativi.

Il Wolfram Language promette di mettere a nostra disposizione uno stile di programmazione simbolico e funzionale che rende superflue le strutture di controllo e ci dà accesso a fatti e conoscenza espressi in modo simbolico.

## Unire i puntini

Per capire bene cosa sta arrivando, conviene girarsi indietro e unire i puntini, come diceva Steve Jobs nel celebre emozionante discorso alla cerimonia di Stanford nel 2005. In un articolo del suo blog, Wolfram ammette di non avere espressamente lavorato per arrivare a creare un ambiente di programmazione rivoluzionario, ma di essersi reso conto che i passaggi necessari per la nuova tecnologia sono stati il cammino degli ultimi 25 anni.

Il primo passo è Mathematica, un linguaggio potente per il calcolo simbolico, creato da Wolfram sul finire degli anni '80 per automatizzare il calcolo simbolico e rendere la vita più semplice a chi campa di equazioni.

Il secondo, è Wolfram Alpha, un motore di ricerca fattuale realizzato in Mathematica. Si tratta di un motore di ricerca molto particolare, che non indicizza termini senza sapere cosa significano, ma cerca di accumulare e catalogare conoscenza. A Wolfram Alpha possiamo fare domande come *"stopping power of lead against 5 MeV alpha particles"*, per conoscere il potere di arresto del piombo bombardato con particelle alfa intorno a 5 MeV di energia, *"highest mountain on the moon"* per scoprire quanto può essere alta una montagna in assenza di erosione atmosferica (tanto), oppure *"C# minor pentatonic"* per conoscere tutto sulla scala musicale che ha fatto le fortune di una generazione col rock e delle due precedenti col blues.

Se conoscete un appassionato di matematica, o fisica che non ha sentito parlare di Wolfram Alpha mostrateglielo appena possibile: un universo di

conoscenza, calcolo simbolico, dati e visualizzazioni accessibile senza sforzo.

Collegando i puntini, quindi, partiamo da un linguaggio funzionale e simbolico, continuiamo con un motore di inferenza e di accumulo di conoscenza, aggiungiamo un protocollo di interrogazione remota basato su web service e un cloud per rispondere alle richieste e mettere online le funzioni create col linguaggio. Quello che appare è un modello distribuito di calcolo, che può mettere conoscenza e elaborazione simbolica a disposizione di un computer che strizza una Debian in una schedina con un Arm a 700 MHz e 512 Mb di memoria.

Visto da qui, "permette a chiunque di calcolare qualunque cosa" è sintesi giornalistica più che grossolana esagerazione.

## Mettiamoci le mani sopra

Abbiamo cercato a lungo il download di una versione di valutazione, ma non ne abbiamo trovati. Per questo, abbiamo deciso di ordinare con consegna urgente un Raspberry Pi revisione B con 512 M di memoria, un investimento di 35 euro di cui non ci pentiamo.

Abbiamo preparato una scheda SD, formattandola con lo strumento della SD Card Association, secondo le semplici e chiare istruzioni che troviamo su [www.raspberrypi.org/downloads](http://www.raspberrypi.org/downloads).

Successivamente, abbiamo scaricato il software Noobs, che ha per sigla un gioco di parole sui "nuovi arrivati". Scartata la caramella, collegati un po' di fili volanti, rete, mouse, tastiera e Hdmi, e fornita l'alimentazione con il caricatore di un telefono, abbiamo iniziato la procedura grafica guidata per scaricare il software preferito. Abbiamo scelto la distribuzione Raspbian, una Debian



**Il boot del software noobs permette di scegliere la distribuzione da installare, possiamo selezionare fra Linux e ambienti più mirati, come una piattaforma media center.**

modificata. In meno di trenta minuti abbiamo visto la traccia del primo boot Debian correre veloce su uno schermo da 24 pollici, decisamente sproporzionato per questo scruciollo, più piccolo del mouse a cui è collegato.

Abbiamo scelto l'avvio con l'interfaccia a carattere, come da lunga tradizione. Meglio avere una shell utilizzabile prima di avviare l'ambiente X, non si sa mai. Per default, l'utente di riferimento è pi e la password raspberry. Dopo il login abbiamo avviato l'ambiente grafico con startx.

L'ambiente grafico della Raspbian è il leggerissimo Lxde, che ha un menu di lancio che incorpora una buona varietà di programmi e un'icona sulla barra di avvio per lanciare Midori (che vuol dire verde in giapponese), un browser basato su WebKit, come Safari o Chrome. Sul desktop troviamo le icone per mandare in esecuzione l'interprete Wolfram in una finestra di terminale, possiamo avviare il programma anche da una shell qualsiasi, volendo, senza nemmeno entrare in ambiente grafico, conservando un bel po' di risorse di memoria e di calcolo.

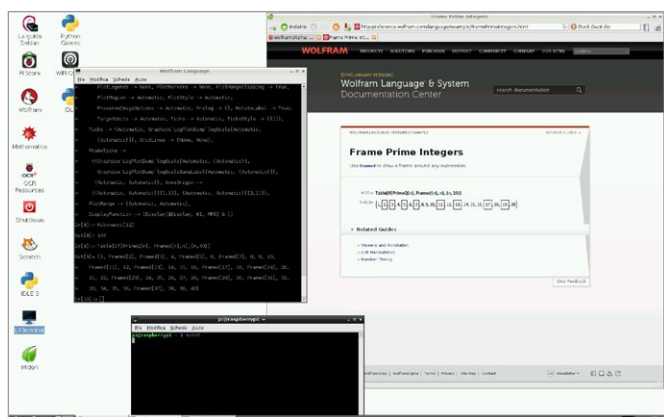
Mathematica ci permette di programmare in ambiente grafico e quindi di sfruttare le ottime capacità di visualizzazione del linguaggio.

Notiamo, di passaggio, che il Pi è il secondo computer che ha in dotazione una copia gratuita di Mathematica, il precedente era il NeXT. Di nuovo Steve Jobs.

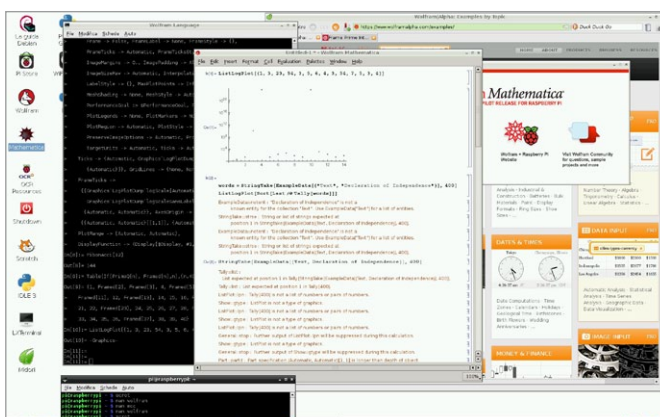
Di passaggio, notiamo che lavorare su un groviglio di fili attaccato a una scheda nuda buttata sul tavolo ci ha fatto sentire molto bene e ha rievocato tempi più ricchi di ottimismo.

Le prestazioni del computer sono più che buone, considerando che la Cpu non ha nemmeno un dissipatore. Certo,

**«Raspberry Pi è il secondo computer che ha in dotazione una copia gratuita di Mathematica, il primo era NeXT di Steve Jobs.»**



La Cpu non lesina potenza, una finestra sull'interprete e un browser per navigare nella documentazione. La riga di comando ci è servita per lo screen shot.



L'esempio che abbiamo eseguito come prova di sviluppo permette di stampare una tabella di cifre mettendo in evidenza i numeri primi.

siamo lontani dai nostri quad core di adesso, ma come esperienza non siamo lontani da quello che era un buon desktop Pentium negli anni '90. La navigazione col browser per andare a caccia di documentazione, non è affatto penosa e anche l'uso di siti piuttosto esigenti, come Flickr, che abbiamo usato per l'upload delle schermate, non ci ha messo alla prova quanto certi netbook che abbiamo usato pochi anni fa.

## Un linguaggio "Twitter"

Mettendo in secondo piano, per ora, le caratteristiche peculiari del linguaggio, la prima cosa che colpisce di Wolfram Language è il rapporto fra codice utile e codice di impalcatura, che può anche essere infinito, nel senso che il secondo è zero.

Qualunque frammento è eseguibile direttamente senza richiedere codice, inclusione di librerie o altre infrastrutture sintattiche.

Si possono concatenare gli operatori sui dati iniziali, si può usare uno stile di programmazione funzionale, passando i risultati di una funzione alla successiva e non c'è la distinzione stretta di altri linguaggi fra codice, dati e output dell'elaborazione.

In aggiunta a questo, il linguaggio abbonda di costrutti di alto livello, infine il motore online di Wolfram Alpha può offrire l'accesso a fonti dati che vanno dalle condizioni meteo, all'andamento di borsa.

Tutto questo collabora a ottenere in uno stile di programmazione sintetico al massimo, tanto che frammenti di programma di una o due righe, a misura di tweet, producono risultati

sorprendentemente complessi.

Facciamo qualche esempio. Ecco cosa serve per disegnare una cartina geografica, che contiene due città americane e ne mette in evidenza il contorno:

```
GeoListPlot[{
  Entity["City", {"NewYork",
    "NewYork", "UnitedStates"}],
  Entity["City", {"Boston",
    "Massachusetts", "UnitedStates"}]
}]
```

Questo esempio è ancora più interessante: l'output è una cartina mondiale in cui sono contornate e evidenziate in rosso le nazioni che hanno partecipato alla guerra di Corea:

```
GeoListPlot[DeleteCases[
  EntityValue[Entity
    [{"HistoricalEvent",
    "KoreanWarBegins"},
    "CountriesInvolved"],
  Entity["HistoricalCountry", _]],
  GeoBackground -> "ReliefMap"]
```

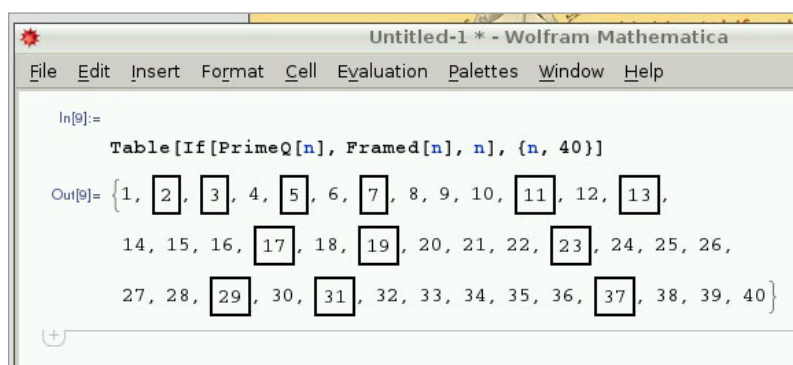
Possiamo usare dati geoeconomici, ad esempio, per realizzare un grafico del prodotto interno lordo rispetto alla popolazione. Nel codice richiediamo entrambe le scale logaritmiche e un tooltip con il nome della nazione in ogni punto.

```
ListLogLogPlot[
  Tooltip[CountryData[#,
    "Population"], CountryData[#,
    "GDP"]],
  CountryData[#, "Name"] &
  /@ CountryData["Countries"]]
```

Una tabella dei primi duecento numeri interi, in cui i numeri primi sono evidenziati con un rettangolo

```
Table[If[PrimeQ[n], Framed[n], n], {
  n, 20}]
```

La ricerca e l'evidenziazione con un cerchio rosso delle celle non sovrapposte ad altre in un'immagine microscopica rappresentata dalla variabile *i*



Possiamo eseguire Wolfram Language dentro la finestra di Mathematica, usando al meglio le possibilità di visualizzazione grafica.



```
cells = SelectComponents[▼
  DeleteBorderComponents ▼
  [Binarize[i, {0, .7}]], ▼
  {"Area", "Holes"}, ▼
  100 < #1 < 1000 && #2 > 0 &];
circles =
  ComponentMeasurements[
    ImageMultiply[i, ▼
      cells], {"Centroid", ▼
        "EquivalentDiskRadius"}][[
      All, 2]];
Show[i, Graphics[{Red, Thick, ▼
  Circle @@ # & /@ circles}]]
```

È chiaro che con esempi di questo

genere la discussione sulla sintassi, le strutture dati e le strutture di controllo diventa secondaria per dare la cifra del linguaggio. Notiamo però che tutti gli esempi che abbiamo mostrato implicano un trattamento automatico delle liste di dati, senza bisogno di esplicitare cicli.

Notiamo en passant che l'operatore `/@` indica di mappare un'operazione su un set di elementi.

In ogni caso, non manca il consueto corredo di `Do`, `For`, `While`, `If`, `Switch` e le solite cose.

Il linguaggio ha una sintassi ampia, anche perché non abbiamo trovato

traccia di strumenti per la segmentazione o la creazione di librerie, in compenso è facile creare funzioni e esportarle nel cloud.

Insomma, c'è parecchio da masticare, ma per fortuna la documentazione online è chiara e estensiva.

## Conclusioni

Non abbiamo finito di parlare di `maker` nel numero scorso, che ci siamo trovati immersi nello hacking artigianale, testimoni di un movimento molto interessante che ancora una volta ha per motore l'intelligenza del software e lo hardware a basso costo, come ai tempi dell'Apple II.

Chissà se è un segno dei tempi e se sta tornando l'epoca dell'ottimismo in cui si crea qualcosa di nuovo e gli hobbisti sono protagonisti. Questa volta, non attrezziamo camere oscure in camera da letto o saldiamo transistor con un occhio allo schema, ma abbiamo mattoni di livello molto più alto con cui giocare e possiamo pensare di costruirci, per ipotesi, un antifurto con riconoscimento facciale mettendo insieme componenti facilmente reperibili e di basso costo.

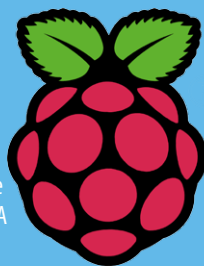
**Non sappiamo prevedere che effetto avrà** sulla diffusione del linguaggio la commistione con i giochi e i piccoli dev e, che è solo all'inizio nei piani del produttore. Il blog di Wolfram mostra un chiaro interesse per la Internet of things.

Di certo, vedremmo bene una copia del linguaggio in ogni dipartimento di matematica, fisica, chimica, statistica e sociologia e in molti altri ambiti scientifici o legati alla pubblicazione, dove l'accesso a una ricca libreria di dati e funzioni di visualizzazione permette di creare infografiche rapidamente. Qualunque sia il corso delle cose nel futuro, è innegabile che una copia di Mathematica e un computer extra con cui usarla valgono bene i meno di 40 euro di un Raspberry Pi.

Il potenziale didattico e creativo del linguaggio è enorme e le possibilità per l'educazione e la ricerca sono spettacolari, soprattutto considerando il basso costo dello hardware.

In effetti, quando i mattoni sono così economici, ci si può anche proporre, per esempio, di investigare come si comporta un cluster di cento computerini a macinare big data. •

## VITA DI RASPBERRY PI



Il Raspberry Pi è nato in laboratorio nel 2011 ed è stato messo in vendita la prima volta il 29 febbraio 2012. La versione attuale è la versione B, che in questo momento è venduta con 512 MByte di memoria a circa 35 euro. La versione B offre una connessione di rete Ethernet, due porte Usb e 512M di memoria. La versione A si trova ancora, a un prezzo inferiore di dieci dollari.

La board nasce intorno a un processore Broadcom BCM2835, un Soc (System on a Chip) che contiene la Cpu, la Gpu, un Dsp, la memoria principale e una porta Usb. Broadcom, il produttore del chip, lo classifica un processore applicativo dedicato per applicazioni multimedia, con video 1080p.

**Il processore ha eccellenti prestazioni** nella codifica e decodifica video e audio. Se le prestazioni della Cpu Arm a 700 MHz sono confrontabili con gli 0,3 GFlops di un Pentium 300 MHz anni '90, la Gpu offre 1,5 Gtexel al secondo, più o meno le prestazioni video di una Xbox del 2001. Tutto questo, senza richiedere uno scambiatore di calore o un ventilatore per il raffreddamento. Il consumo della scheda è di soli 3,5 W. Proprio per questa vocazione multimediale, le connessioni audio/video della schedina inglese sono una porta Hdmi, e un connettore video coassiale e un jack audio stereo come seconda possibilità. Non si parla di unità a disco: la memoria di massa è affidata a un connettore per una scheda SD, ma la distribuzione Debian sta comoda anche in 8G.

C'è anche un connettore Csi per il collegamento di una sorgente video, come la telecamera dedicata, una evoluzione del sottosistema video del Nokia N8. La distribuzione Raspbian ha diversi comandi per scattare foto o catturare video. I formati più popolari, come Jpeg e H264 sono supportati. Uno dei tutorial sul sito [raspberrypi.org](http://raspberrypi.org) mostra come collegare un sensore di movimento e realizzare un sistema per riprendere video quando qualcosa si muove in prossimità della telecamera con poche righe di Python. Python, per inciso, è incluso nella Raspbian sia in versione 2, sia in versione 3. La Cpu del Raspberry Pi viene da molto lontano. Si basa su quell'architettura Arm, nata come Acorn Risc Machine nei laboratori della Acorn, uno dei pionieri inglesi dell'informatica, che creò un computer per la BBC e una piccola workstation dell'epoca, l'Archimedes. Per un certo periodo la Acorn fu partecipata da Olivetti, che seguì la nascita del processore. Olivetti, però era interessata all'architettura Pc e non vedeva molto valore in un marchio inglese che produceva computer con un sistema operativo britannico. Stranamente, Olivetti non vedeva molte possibilità nemmeno nel processore che finì per essere il motore dell'Apple Newton e si liberò frettolosamente di questo asset. Chi volesse approfondire questa storia può trovare più dettagli nel resoconto di Fuggetta ([alfonsofuggetta.org/?p=10770](http://alfonsofuggetta.org/?p=10770)), o meglio ancora in quello in Inglese di Ellee Seymour, che si trova facilmente cercando il nome e la keyword Olivetti. L'architettura Arm ha avuto moltissima fortuna ed è apparsa nel Kindle 2, nel Nintendo 3DS e in tutti gli smartphone di Nokia, HTC, LG, Motorola e Samsung, oltre a buona parte dei tablet Android e Windows RT. I processori Apple all'interno degli iPhone e iPad, inoltre, si basano sull'architettura Arm. I sistemi operativi che girano su Arm includono iOS, Android, Windows Phone, Windows RT, Firefox OS e una varietà di sistemi unix. Sommando tutto, si tratta dell'architettura di processore più fortunata del mondo.