Wolfram *Mathematica*
Il software di riferimento per la Didattica, la Ricerca e lo Sviluppo

ADALTA
Distributore ufficiale per l'Italia di Wolfram Research
www.adalta.it/wolfram

WebSeminar Mathematica

Lezione 1

I primi 60' con *Mathematica*

Crescenzio Gallo – Università di Foggia
crescenzio.gallo@unifg.it

Note:

- Il materiale visualizzato durante questo seminario è disponibile per il download all'indirizzo http://www.crescenziogallo.it/unifg/seminario-mathematica-2014/
- Il materiale utilizzato è tratto dai webinar pubblicati da Adalta e prodotti dal dott. Roberto Cavaliere (*Mathematica* Technic Sales Manager, *r.cavaliere@adalta.it*)

12 – 26 Giugno 2014

Agenda

L'ambiente *Mathematica*

- Front end e kernel
- Primi passi con il front end
- Help e Documentation Center
- Le Palette
- Un help speciale: WolframAlpha e il linguaggio naturale

Le basi del linguaggio Mathematica

- Definire variabili e funzioni
- Il concetto di opzione

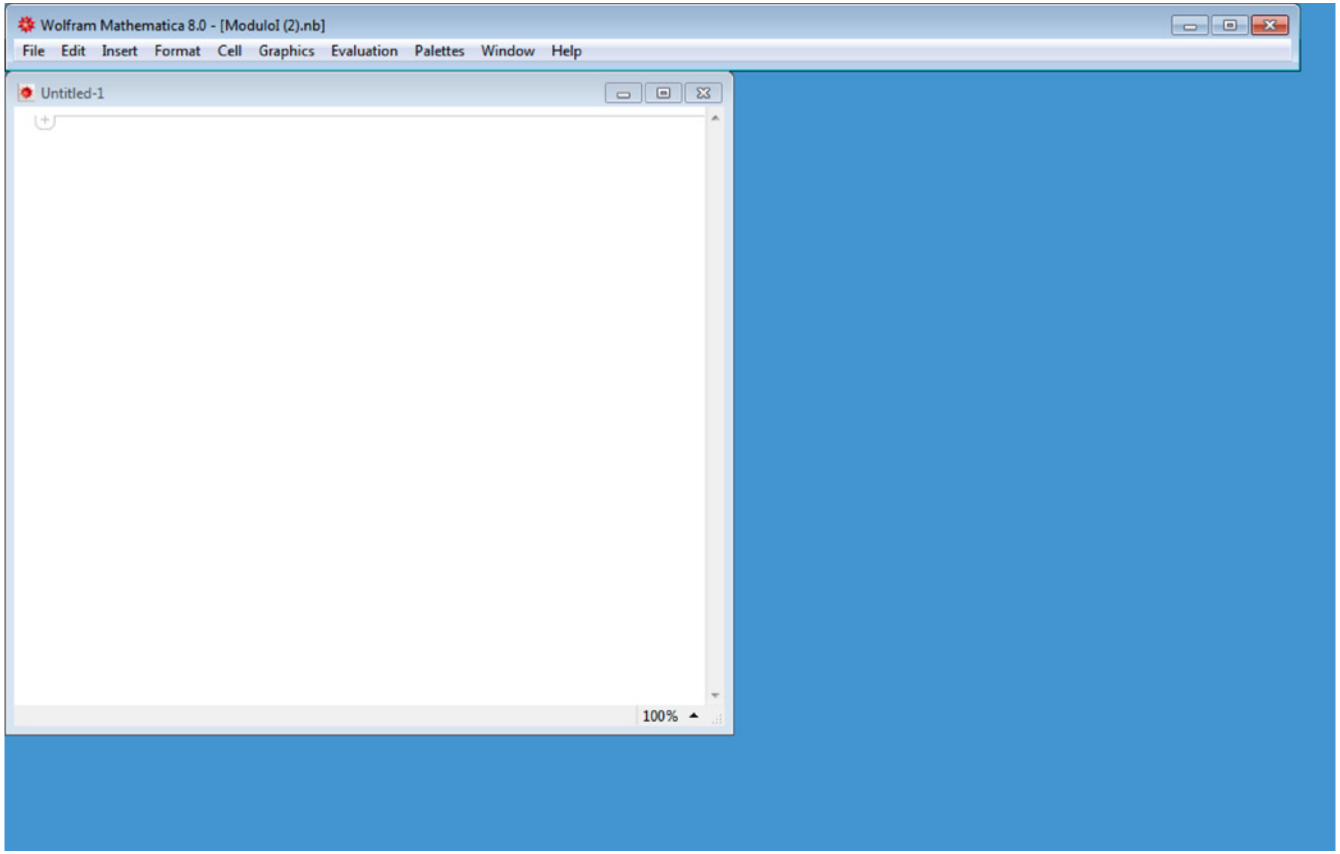
Esempi di funzionalità

- Calcolo numerico
- Calcolo simbolico
- Grafica
- Applicazioni dinamiche ed interattive
- Sorgenti dati integrate

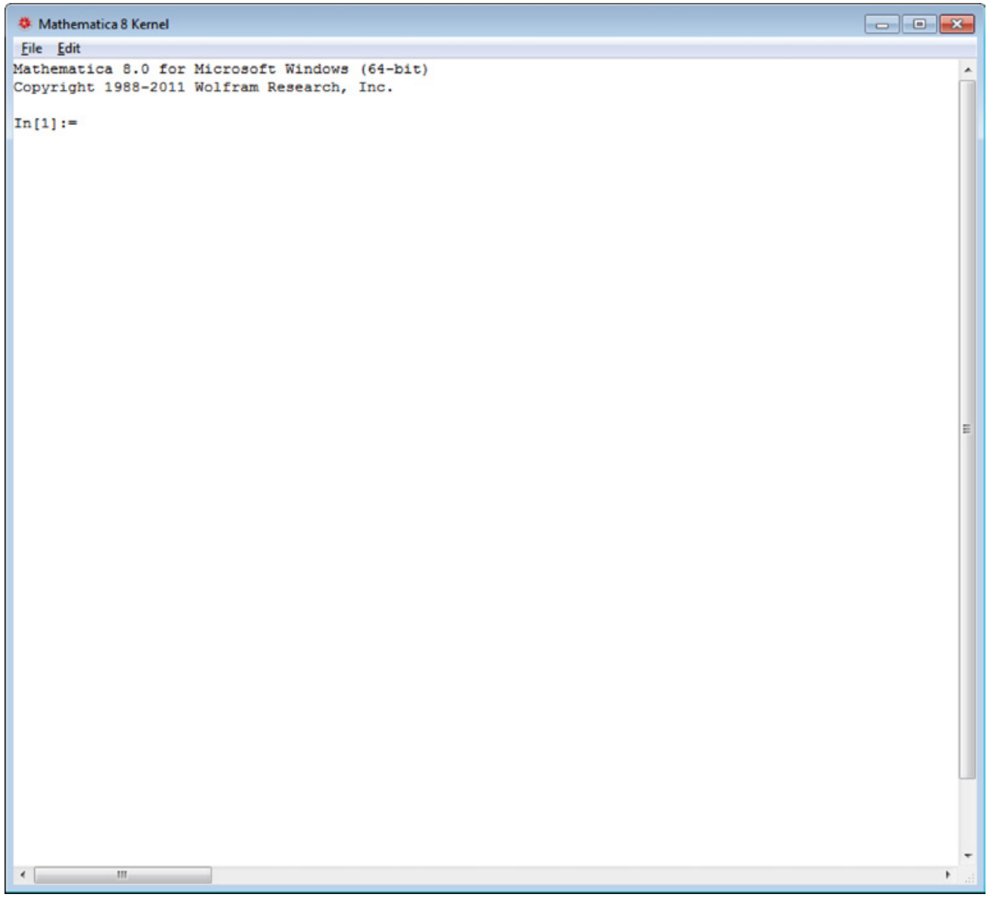
L'ambiente *Mathematica:* front end e kernel

L'architettura interna di *Mathematica* consiste di due moduli separati:

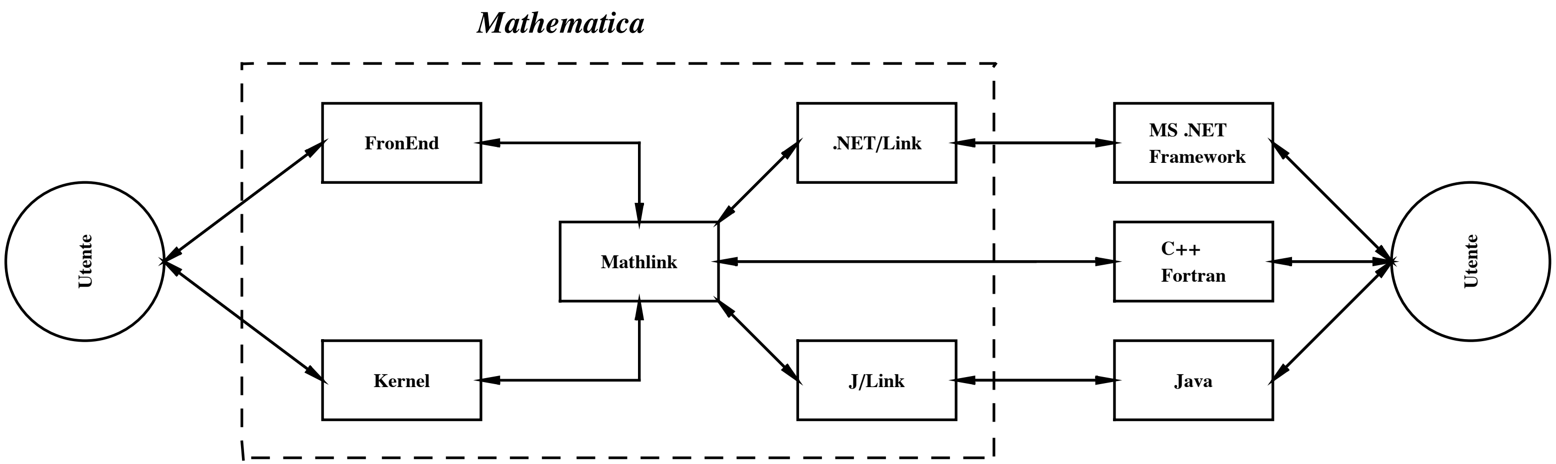
L'interfaccia utente o **front end**: il modulo dedicato al dialogo con l'utente (interpretazioni delle linee di input e rappresen tazione degli output).



Il motore di calcolo o **kernel**, dedicato alle computazioni vere e proprie



I due moduli sono in collegamento attraverso il *MathLink*, una libreria in grado di far dialogare il kernel con diversi altri sistemi.

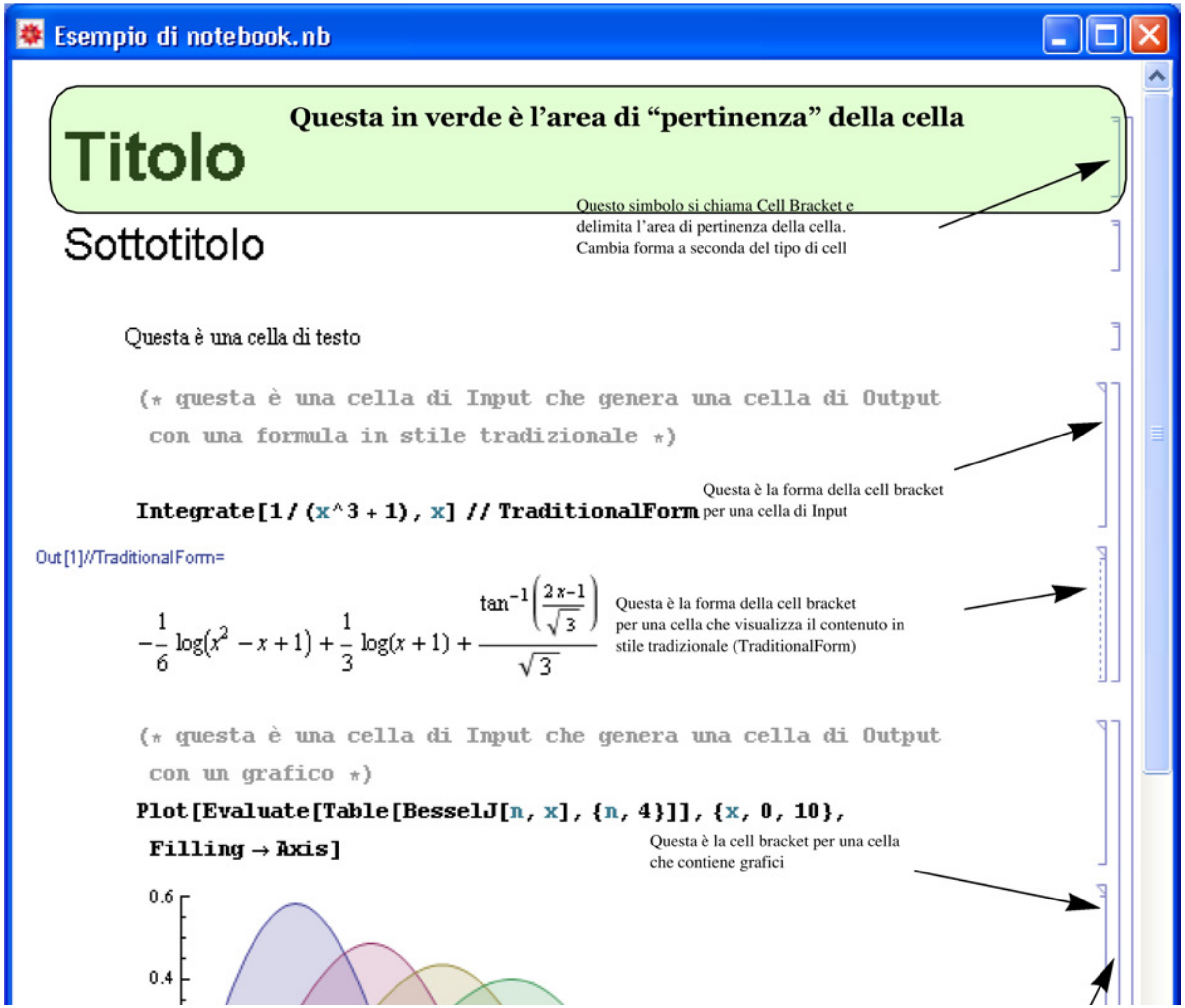

Mathematica
FronEnd
.NET/Link
MS .NET
Framework
Utente
Mathlink
C++
Fortran
Utente
Kernel
J/Link
Java

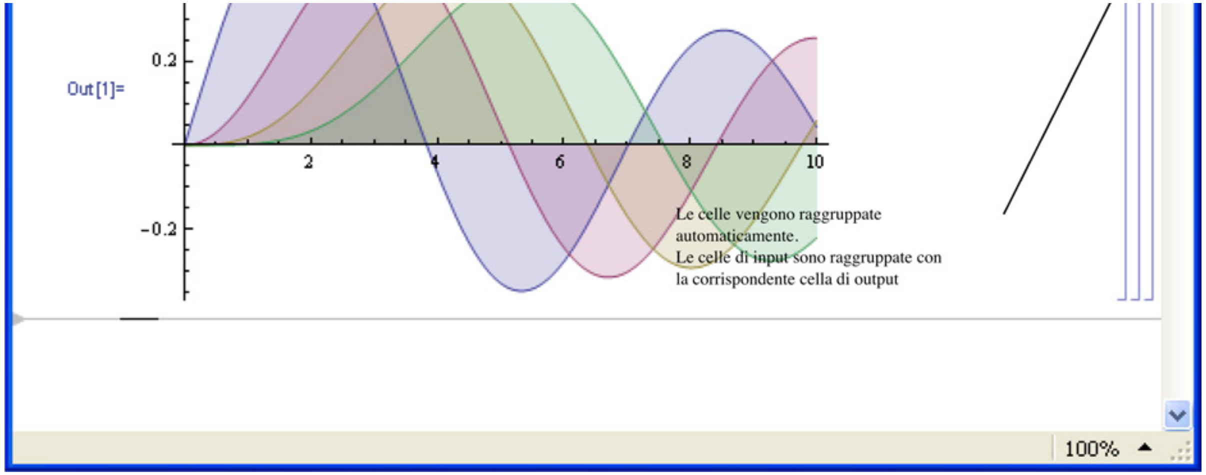
L'ambiente *Mathematica*: primi passi con il front end

» Notebook, Cell e CellBracket

Quando si utilizza il FE tutte le interazioni tra l'utente e *Mathematica* avvengono per il tramite di un Notebook

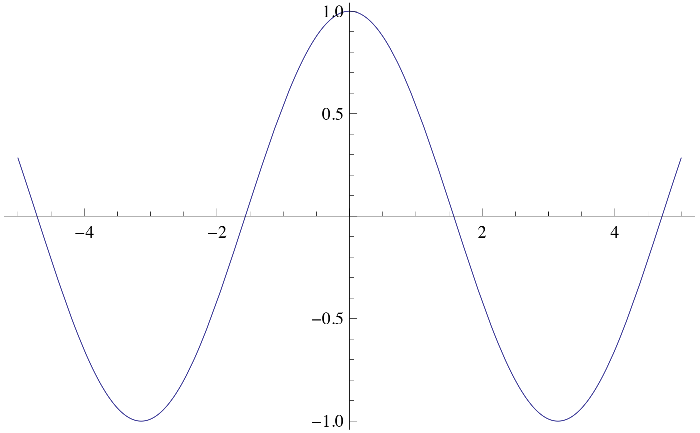
I notebook sono documenti attivi suddivisi in unità chiamate Cell (cella). Ogni cella ha una sua lista di proprietà che possono renderla differente dalle altre sia nell'aspetto che nel comportamento.





Un'altra tipologia di cell bracket che si incontra spesso è quella relativa ai gruppi di celle chiusi.
Quando si hanno più celle in un gruppo si possono nascondere alcune celle, ad esempio il codice per generare un grafico

```
Plot[Cos[x], {x, -5, 5}]
```



» La selezione

Per selezionare solo una parte del contenuto di una cella si può operare in due modi. Il primo è quello usuale di selezionare parte che interessa con il puntatore del mouse trascinandolo da un estremo all'altro della parte in questione, mantenendo pr muto il tasto sinistro del mouse. Il secondo modo è molto utile soprattutto per gestire le singole parti delle celle di input, dove ha spesso l'esigenza di evidenziare un'espressione o parte di essa. Tale modo di selezione delle parti di una cella si attiva semplic mente con dei click successivi sul tasto sinistro del mouse puntato sulla parte da selezionare.

```
Solve[{x^2 - 2 y + 1 == 0, x - 2 y + 1 == 0}]
```

$\left\{\left\{y \to \frac{1}{2}, x \to 0\right\}, \{y \to 1, x \to 1\}\right\}$

Se si vuole selezionare l'intera cella, ad esempio per modificarne una proprietà generale di cella, bisogna cliccare sulla cell bracket


```
Solve[{x^2 - 2 y + 1 == 0, x - 2 y + 1 == 0}]
```

$\left\{\left\{y \to \frac{1}{2}, x \to 0\right\}, \{y \to 1, x \to 1\}\right\}$

L'ambiente *Mathematica*: primi passi con il front end

Mathematica ha poche regole da rispettare in maniera rigorosa per quanto riguarda la sintassi

» Eseguire calcoli

Per chiedere l'esecuzione di un calcolo o di un qualsiasi comando, dopo aver scritto l'espressione in una nuova cella bisogna dig itare contemporaneamente i tasti SHIFT+RET. Questa combinazione di tasti è stata stabilita dalla Wolfram per lasciare inalternato significato standard del semplice tasto Invio (RET) che serve per mandare il testo a capo.

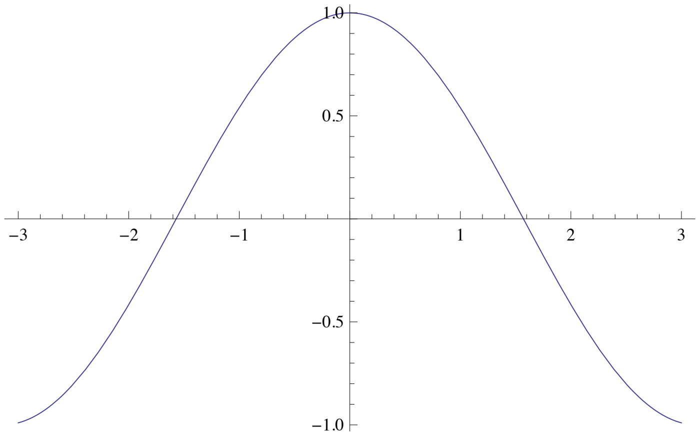
```
2 + 2
```

4

» Maiuscole/minuscole

Mathematica è "case sensitive" ossia fa distinzione tra lettere maiuscole e minuscole. Questo significa che la variabile **x** è divers dalla variabile **X** oppure che il comando **Plot** non funziona se si scrive **plot**.

```
Plot[Cos[x], {x, -3, 3}]
```



» Le funzioni native

Per convenzione tutte le parole codice di *Mathematica* iniziano con una lettera maiuscola. Se una parola codice è composta da du parole, entrambe hanno la prima lettera maiuscola (esempi).

» Il significato dei nomi di funzioni

Tutti i nomi di comandi hanno una etimologia che rispecchia il tipo di operazione o calcolo che eseguono (esempi). Se non ricorda bene il nome di una funzione e la si vuole cercare nella documentazione di *Mathematica* basta usare parole chiave relativ al tipo di operazione che si vuole fare e ricordarsi che *Mathematica* è scritto in Inglese (esempi).

» La moltiplicazione

Si può indicare in diversi modi, quello più diffuso è il carattere Blank ossia lo spazio. Altri simboli sono il classico * o il simbolo che si ottiene scrivendo \ [Times]

2 x

$2\,x$

2 * x

$2\,x$

2 × x

$2\,x$

» Uso delle parentesi

Tipo	Esempio	Descrizione
( )	x (x + 1)	le parentesi tonde vengono usate per il raggruppamento di espressioni e per impostare la propedeuticità delle operazioni
[ ]	Cos[x]	le parentesi quadre vengono usate dalle funzioni per il passaggio degli argomenti
{ }	{a, b, c}	le parentesi graffe vengono usare per rappresentare le liste
[[ ]]	{a, b, c}[[2]] ⟶ b	le doppie parentesi quadre vengono usate per indicizzare gli elementi di una espressione

Come riportato in tabella, gli argomenti delle funzioni o comandi avvengono tramite parentesi quadre [ ] e non tonde ( ) come i molti altri linguaggi. Se vi sono più argomenti essi vengono separati dalla virgola. La sintassi tipica di un comando *Mathematica* è

Nome[argomento1, argomento2, ..., opzione1, opzione2, ...]

dove si intende come **argomento** un'espressione fornita dall'utente come elemento su cui fare i calcoli o le operazioni ment **opzione** è una direttiva relativa al particolare comando usato e di cui l'utente ne chiede uno specifico valore (in seguito si ripren derà questo concetto molto importante, degli argomenti opzionali). Vediamo alcuni esempi.

Calcolo di un integrale, dove si specificano solo due argomenti: la funzione integranda e la variabile indipendente rispetto al quale eseguire l'integrazione:

```
r = Integrate[x Log[x], x]
```

$$\frac{1}{2} x^2 \log(x) - \frac{x^2}{4}$$

Grafico della funzione integrale, dove si specificano solo due argomenti: la funzione da disegnare e l'intervallo per la variabi indipendente (questo argomento è necessario):

```
Plot[r, {x, 0, 2}]
```

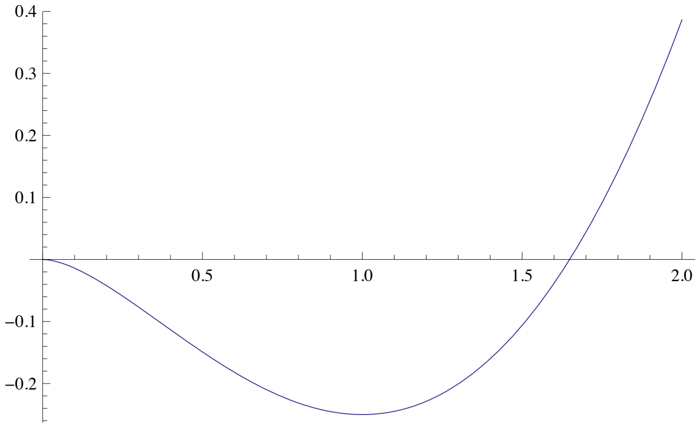
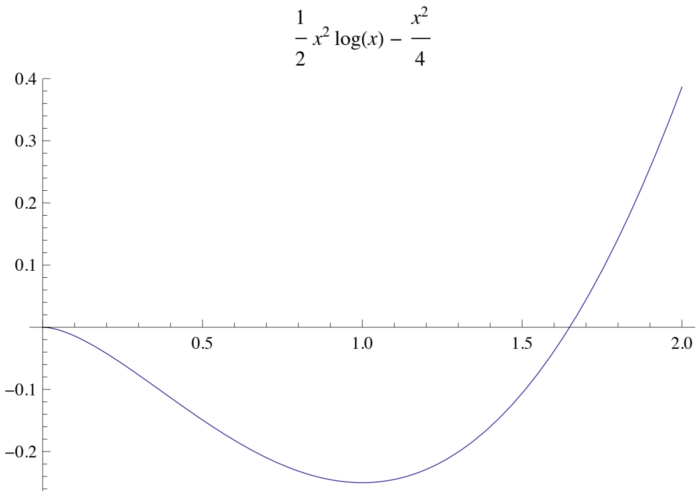


Grafico della funzione integrale con l'aggiunta di commento tramite l'argomento opzionale PlotLabel:

```
Plot[r, {x, 0, 2}, PlotLabel → r]
```



Ulteriori dettagli sulla sintassi, sulle nozioni di base e su come *Mathematica* opera si possono trovare nella sua documentazion

» L'ordine cronologico degli input/output

Quando si valuta una cella, il front end genera una label progressiva del numero di input seguito dal corrispondente output. I questo modo ad ogni passo possiamo sapere quale numero di input abbiamo valutato e, se abbiamo bisogno del corrisponden output, possia ripescarlo senza doverlo ricalcolare.

```
Quit[]
```

```
2 + 2
```

4

```
3 + 3
```

6

```
%1
```

4

Se inseriamo più di una linea sulla stessa cella di input verranno generati corrispondenti numeri di linea:

```
2 + 2;
3 + 3;
4 + 4;
5 + 5
```

10

10 è il risultato dell'ultima riga che ha il progressivo In[7]

I predicati

Tutti i comandi che terminano con una Q (che sta per Question) sono funzioni di tipo predicato: sostanzialmente non fanno altı che applicare un test, rispondendo con True o False a seconda che il test sia verificato o mer (`lista completa dei predicati`).

Le variabili di ambiente

Ci sono diverse variabili di sistema che forniscono informazioni sul sistema installato e su altri aspetti legati anche al piattaforma. Tali variabili iniziano sempre con il carattere $.

```
$PasswordFile
```

/Library/Mathematica/Licensing/mathpass

```
$LicenseID
```

L3518–3501

```
$MachineID
```

5118–99525–59903

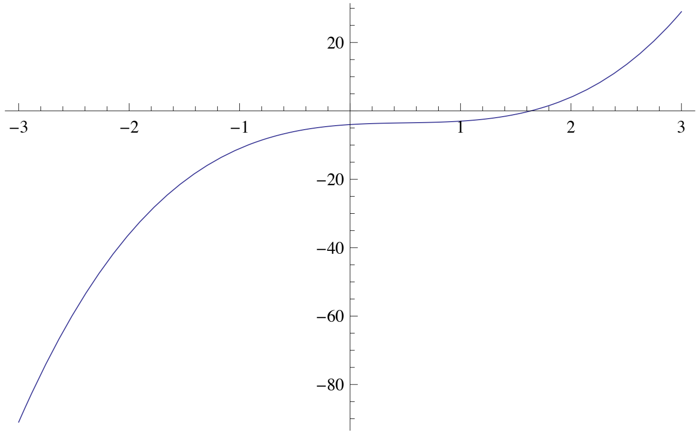
L'ambiente *Mathematica*: primi passi con il front end

Eseguire calcoli e computazioni: *Mathematica* come calcolatrice

In molte situazioni viene naturale usare *Mathematica* come semplice calcolatrice, ossia invocare una o più funzioni per esegui calcoli specifici in una determinata sequenza per risolvere un problema, semplice o complesso che sia. Ad esempio si vuole dise nare una certa funzione, calcolarne la derivata prima e poi la seconda e visualizzare le tre funzioni contemporaneamente, calcola l'integrale indefinito e poi quello definito. Questa sequenza di operazioni tipicamente si esegue con una serie di chiamate a fu zioni di base.

Disegnare la funzione:

```
Plot[2 x^3 - 3 x^2 + 2 x - 4, {x, -3, 3}]
```



Calcolare la derivata prima:

```
D[2 x^3 - 3 x^2 + 2 x - 4, x]
```

$6 x^2 - 6 x + 2$

e poi la seconda:

```
D[2 x^3 - 3 x^2 + 2 x - 4, {x, 2}]
```

$12 x - 6$

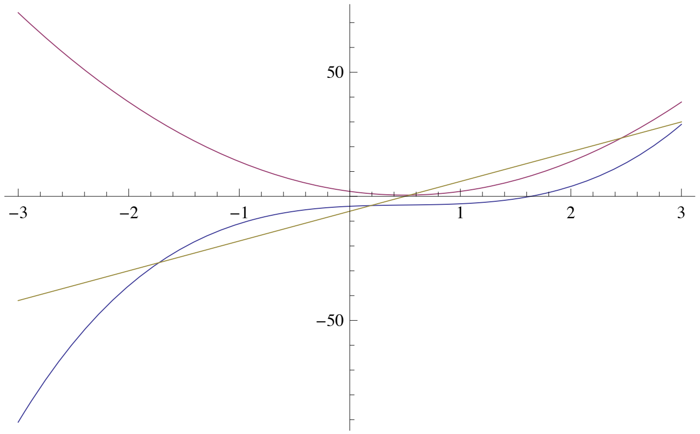
Verificare quando la derivata seconda si annulla:

```
Solve[% == 0, x]
```

$$\left\{\left\{x \to \frac{1}{2}\right\}\right\}$$

Visualizzare il grafico della funzione e delle sue due prime derivate:

```
Plot[{-4 + 2 x - 3 x^2 + 2 x^3, 2 - 6 x + 6 x^2, -6 + 12 x}, {x, -3, 3}]
```



Calcolare l'integrale indefinito:

```
Integrate[2 x^3 - 3 x^2 + 2 x - 4, x]
```

$$\frac{x^4}{2} - x^3 + x^2 - 4\,x$$

e poi quello definito:

```
Integrate[2 x^3 - 3 x^2 + 2 x - 4, {x, -3, 3}]
```

-78

Si vuole verificare la forma delle soluzioni generali di una equazione di secondo grado del tipo $a\,x^2 + b\,x + c == 0$

```
Solve[a x^2 + b x + c == 0, x]
```

$$\left\{\left\{x \to \frac{-\sqrt{b^2 - 4 a c} - b}{2 a}\right\}, \left\{x \to \frac{\sqrt{b^2 - 4 a c} - b}{2 a}\right\}\right\}$$

◀ | ▶

L'ambiente *Mathematica*: help e Documentation Center

Mathematica dispone di diversi livelli di help. Quello più semplice è da riga di comando e si ottiene anteponendo il simbolo ? nome di un comando.

```
? Integrate
```

Integrate[f, x] gives the indefinite integral $\int f\, dx$.

Integrate[f, {x, x_{min}, x_{max}}] gives the definite integral $\int_{x_{min}}^{x_{max}} f\, dx$.

Integrate[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}, …] gives the multiple integral $\int_{x_{min}}^{x_{max}} dx \int_{y_{min}}^{y_{max}} dy \ldots f$. ≫

Si può anche utilizzare il carattere jolly * se non si ricorda il nome completo. In tal caso verrà restituito un elenco di tutte le paro codice che contengono le lettere indicate.

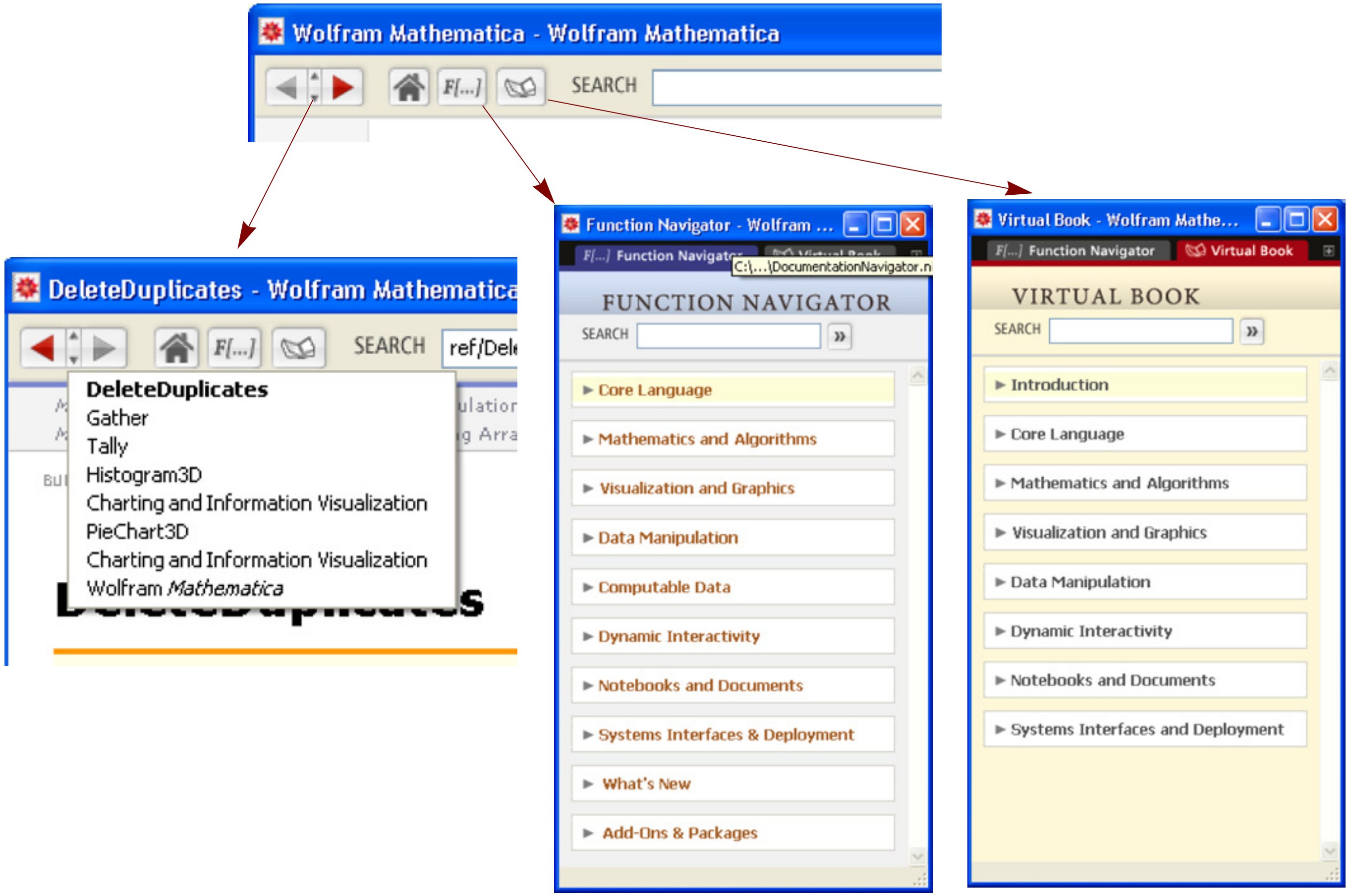
```
? Integ*
```

▼ System`

Integer	IntegerExpone-nt	IntegerPart	IntegerQ	IntegerString	Integrate
IntegerDigits	IntegerLength	IntegerPartition-s	Integers	Integral	

Questo tipo di help è molto sintentico, riepiloga il principale compito del comando e la sintassi di base per il caso più semplice. I genere si ricorre a tale help per una veloce consultazione della sintassi e dei casi di utilizzo, quando già si conosce la funzione o comando da utilizzare. In altri casi è opportuno fare riferimento al sistema completo di help chiamato Documentation Center.

Il campo etichettato SEARCH è quello dove si possono scrivere le parole chiave per la ricerca dei documenti. Se la parola che inserisce corrisponde esattamente (anche nelle maiuscole) ad una parola codice, viene aperta direttamente la pagina corrispo dente alla Reference Page (pagina di riferimento) del comando. In tutti gli altri casi, viene restituita una pagina con dei link al varie pagine di documentazione dove compare la parola (o le parole) inserita(e). Se non vengono trovati riferimenti viene rest tuita una pagina vuota. La barra dei comandi del visualizzatore della documentazione (anche riferito come help browser) vie descritta dalla seguente figura:



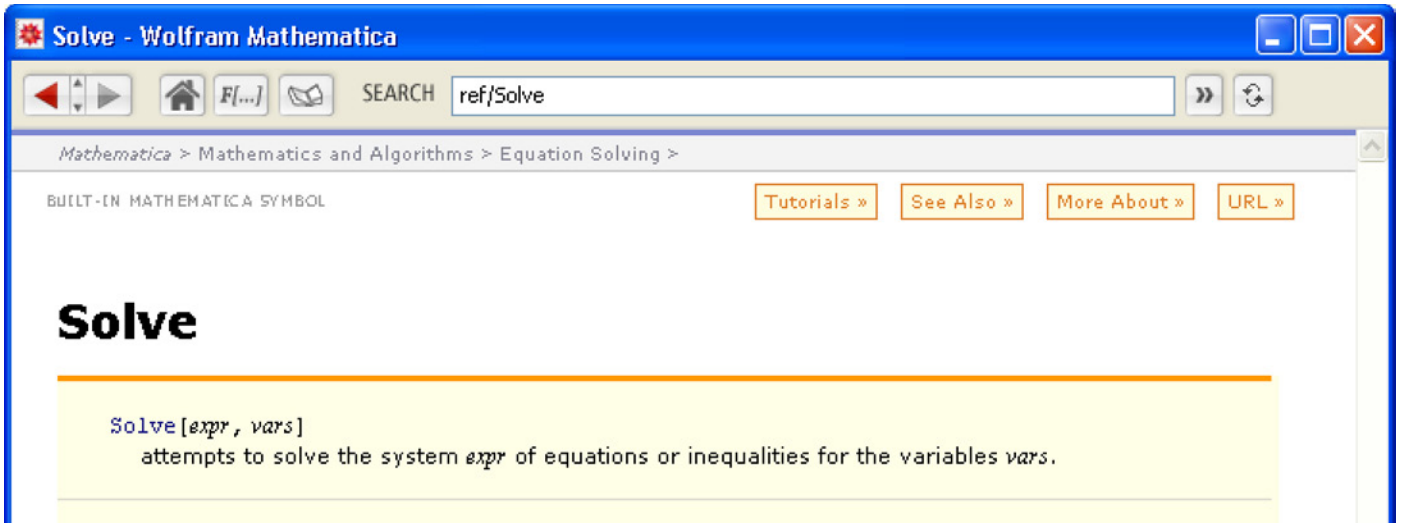
L'aspetto più significativo da conoscere in merito al centro di documentazione è la struttura organizzativa dei documenti, la loro classificazione ed il loro scopo.

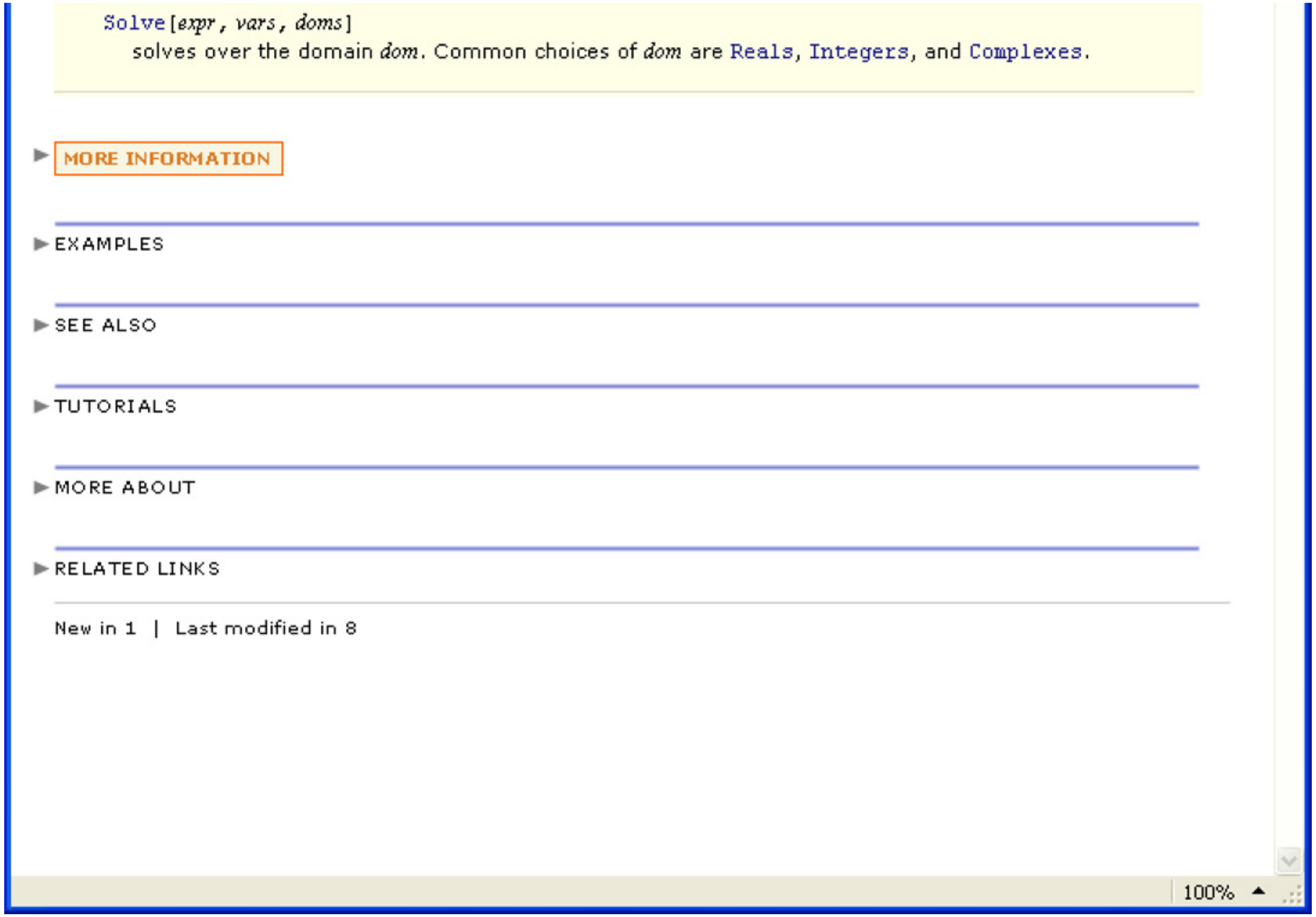
Tipo di documento	Contenuto	Esempio
Reference Page (pagina di riferimento)	Pagina principale relativa al comando	NDSolve
Guide (guida)	Elenco di link alle pagine dei principali comandi relativi ad una specifica area	Manipolazione delle formule
Tutorial (tutorial)	Tutorial relativo ad un certo argomento	Introduzione al comando Manipulate
How to (come fare a...)	Istruzioni passo per passo su come fare una determinata operazione	Analisi statistiche

L'intero sistema di documentazione è replicato fedelmente sul web all'indirizzo http://reference.wolfram.com/mathematica/guide/Mathematica.html.
Ciascun tipo di pagina ha una sua struttura ben definita e replicata in tutte le pagine dello stesso tipo. Ad esempio, le pagine di riferimento, che sono quelle più indicative e ricche di informazioni per ciascun comando, hanno la seguente struttura:

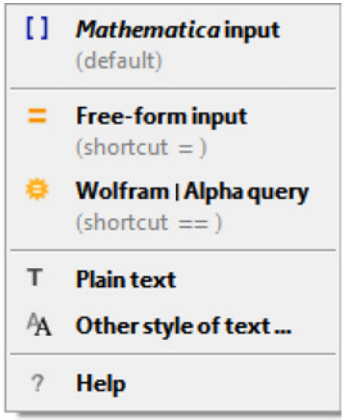
Elemento	Contenuto/Scopo
Titolo	Nome del comando cui la pagina fa riferimento
Sintassi di base	Principali modi di utilizzo della funzione con specifica degli argomenti
MORE INFORMATION	Informazioni sull'implementazione del comando, sulle opzioni, sulle tipologie di input ammessi, sui metodi risolutivi o sui casi particolari
EXAMPLES	Esempi di utilizzo
SEE ALSO	Elenco di comandi correlati
TUTORIALS	Elenco dei tutorial di interesse relativi al comando in questione (questa sezione potrebbe non esserci)
MORE ABOUT	Lista di guide dove sono elencati comandi correlati
RELATED LINKS	Collegamenti a demonstrations o siti Wolfram con ulteriori informazioni (ad esempio per la sorgenti dati integrate vi sono i link alla pagina dove sono citate le fonti esterne utilizzate)
Riga degli aggiornamenti	Indica in quale versione di Mathematica il comando è stato introdotto ed in quale è stato modificato l'ultima volta

La figura che segue mostra la pagina di riferimento per il comando Solve nella sua struttura completa (visibile quando tutti gruppi di celle sono chiusi) e con la sua configurazione di default ossia con la sezione EXAMPLES espansa:





Nella versione 8 è stato anche aggiunto un ulteriore aiuto "rapido" per l'inserimento celle, quello che compare quando si clicca sul bottone a forma di crocetta a sinistra della barra di inserimento. La figura seguente ne mostra il contenuto:


[ ] *Mathematica* input
(default)
Free-form input
(shortcut =)
Wolfram | Alpha query
(shortcut ==)
Plain text
Other style of text ...
Help

L'ambiente *Mathematica*: le Palette

Mathematica dispone anche del meccanismo della Palette, ossia delle pulsantiere che mettono a disposizione le principali funzioni (oltre quelle dei menu) in maniera semplice ed immediata, ossia tramite bottoni.

Esempio di "assistente" per la scrittura di testi: la palette Classroom Assistant (menu Palettes).

Esempio di personalizzazione dei grafici: la palette Drawing Tools (menu Graphics).

Cosa è *Mathematica*: un help speciale - WolframAlpha e il linguaggio naturale

Con la versione 8 *Mathematica* integra anche il motore computazionale WolframAlpha. Tra le tante cose che si possono chiedere WolframAlpha c’è anche il supporto per imparare ad usare *Mathematica* tramite il linguaggio naturale.

W|A include circa dieci trilioni di data sets sugli argomenti più svariati. Ci sono diversi modi per richiamare WolframApha dall’interno di *Mathematica*, sia da linea di codice che programmaticamente.

input di *Mathematica* in linguaggio naturale (Inglese)

interrogazione Wolfram|Alpha

codice inline espresso in linguaggio naturale ed incassato in script di codice *Mathematica*

Esempi di domande:

Dini surface
Fermat theorem
nutrition facts cheese
population history in Italy
how far is Milan from Rome
GDP history in France
boiling point of sulphur
earthquake in Italy 1980

Tramite la sequenza di tasti SHIFT + CTRL + = si può far comparire il simbolo che indica un riquadro dentro il quale possiamo scrivere un’espressione in linguaggio naturale e WolframAlpha tenterà di trasformarla in input di *Mathematica*. Tale riquadro può essere integrato dentro qualsiasi linea di input di *Mathematica*.

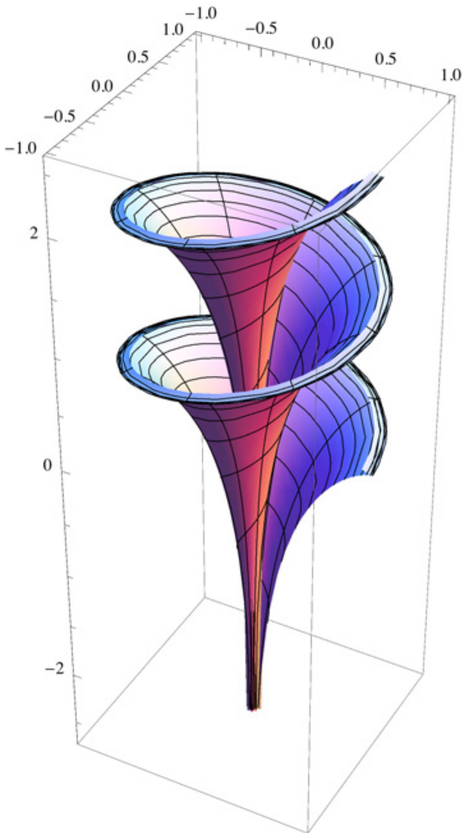
```
Manipulate[ derivative of f(x)
            D[f[x], x]          , {f, {Sin, Cos, Tan, Csc, Sec}}]
```

Allo stesso modo con la sequenza SHIFT + = si può far comparire il simbolo che indica un input che verrà elaborato da WolframAlpha e, se esiste una corrispondente espressione *Mathematica,* verrà fornita, altrimenti verrà dato un output in stile WolframAlpha.

dini surface

Example plot

```
ParametricPlot3D[{Cos[u] * Sin[v], Sin[u] * Sin[v], 0.2 * u + Cos[v] + Log[Tan[0.5 * v]]},
  {u, 0, 4 * Pi}, {v, 0.001, 2}]
```



integrate bessel j2

```
Integrate[BesselJ[2, x], x]
```

$$\frac{1}{24} x^3 \, {}_1F_2\left(\frac{3}{2}; \frac{5}{2}, 3; -\frac{x^2}{4}\right)$$

take derivative of %

```
D[%, x]
```

$$\frac{1}{8} x^2 \, {}_1F_2\left(\frac{3}{2}; \frac{5}{2}, 3; -\frac{x^2}{4}\right) + \frac{1}{8} x^2 \left(\frac{8 J_2(x)}{x^2} - {}_1F_2\left(\frac{3}{2}; \frac{5}{2}, 3; -\frac{x^2}{4}\right)\right)$$

simplify %

Result

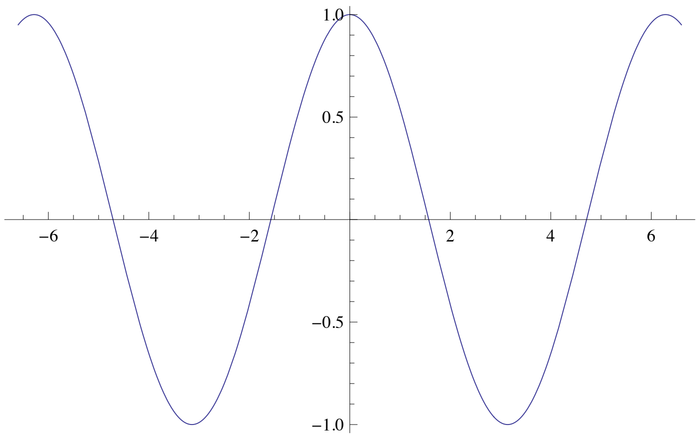
```
Simplify[%]
```

$J_2(x)$

plot cos(x)

Plots (1 of 2)

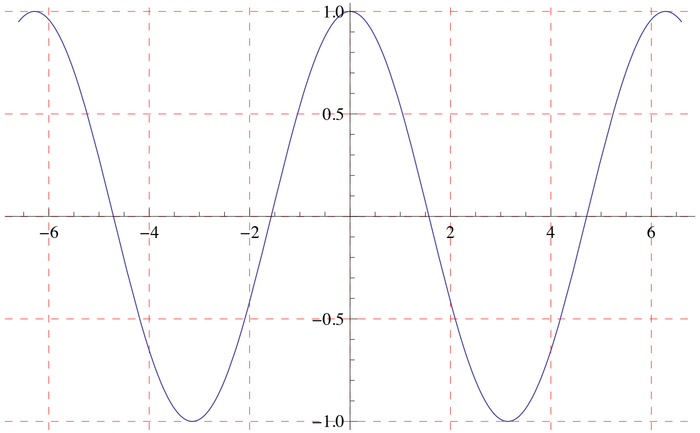
```
Plot[Cos[x], {x, -6.6, 6.6}]
```



add red dashed gridlines »

Input interpretation

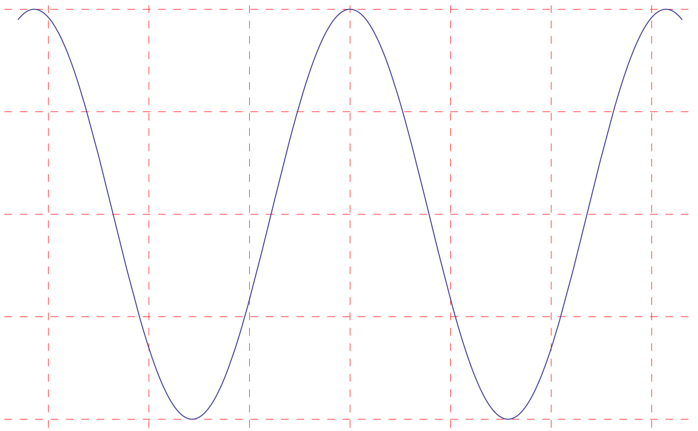
```
Show[%, GridLines -> Automatic, GridLinesStyle -> Directive[Red, Dashed]]
```



remove axes

Input interpretation

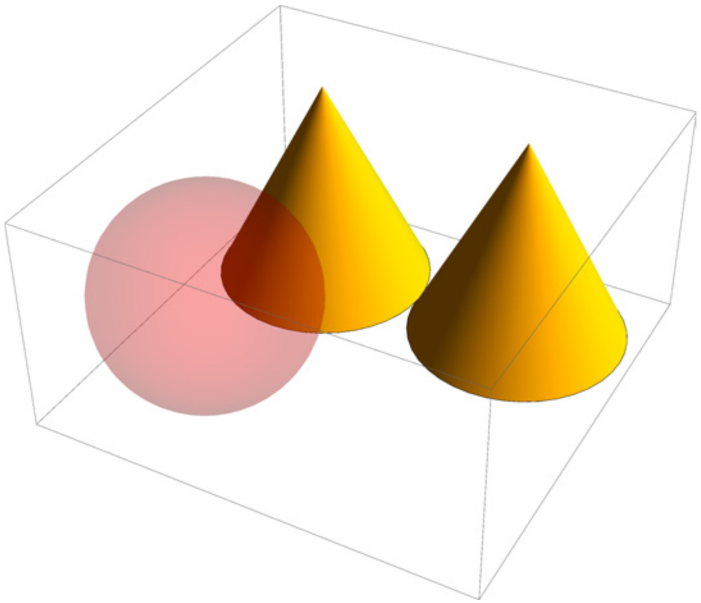
```
Show[%, Axes -> None]
```



show transparent red sphere and 2 yellow cones »

Result

```
Graphics3D[{{Opacity[0.2], Red, Sphere[]},
  {Yellow, Array[Translate[Cone[], {2*#1, 2, 0}] & , 2, 0]}}]
```



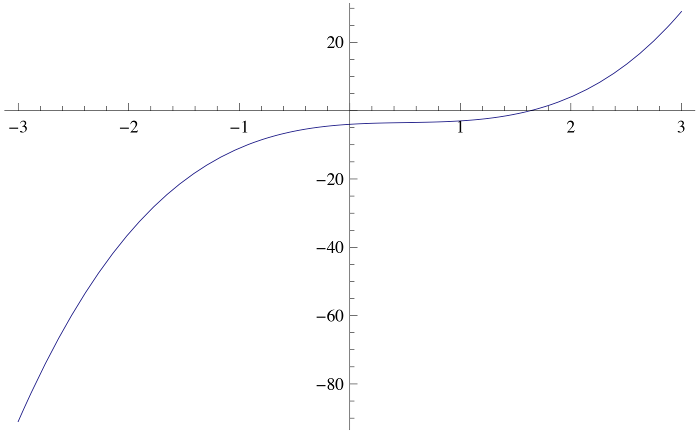
Le basi del linguaggio *Mathematica*: definire variabili e funzioni

In molti altri casi, dopo poche operazioni ci si rende conto che si comincia ad aver bisogno di sintetizzare il codice, evitare di ripetere alcuni argomenti e dunque si desiderano definire delle variabili. L'assegnazione di un valore ad una variabile può essere fatta principalmente con due comandi: **Set** (simbolo =) oppure **SetDelayed** (simbolo **:=**).

```
f = 2 x^3 - 3 x^2 + 2 x - 4
```

$2x^3 - 3x^2 + 2x - 4$

```
Plot[f, {x, -3, 3}]
```



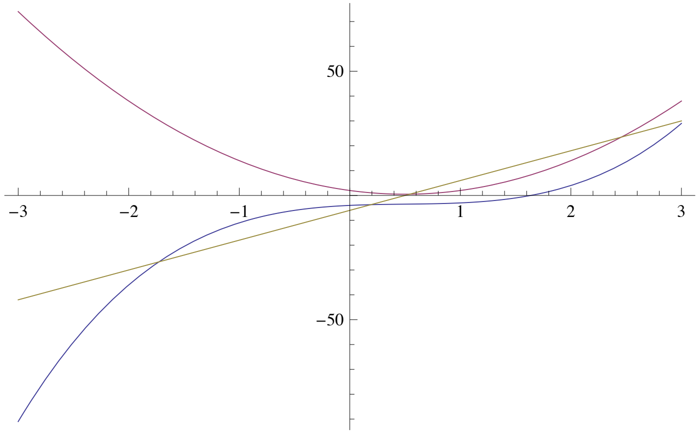

```
d1 = D[f, x]
```

$6x^2 - 6x + 2$

```
d2 = D[f, {x, 2}]
```

$12x - 6$

```
Plot[{f, d1, d2}, {x, -3, 3}]
```

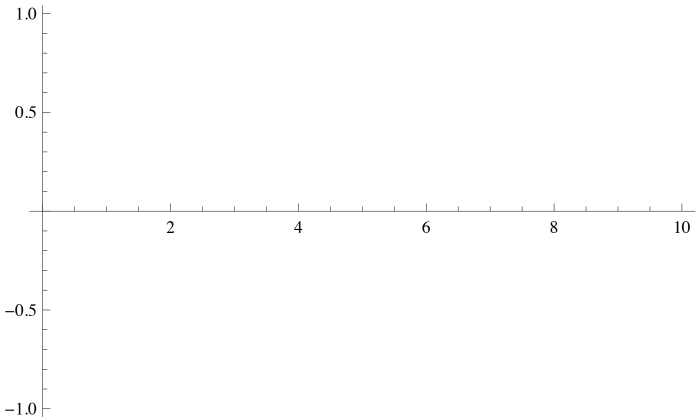


L'aver definito *f* come una variabile e non come una funzione, ossia non aver previsto alcun argomento da fornire per poter inv
care *f*, ha da una parte semplificato la scrittura della definizione, ma, dall'altra, l'ha resa più rigida. Infatti, se si desidera cambia
la variabile indipendente nell'istruzione **Plot** o **Integrate**, non si può.

```
Integrate[f, y]
```

$\left(2x^3 - 3x^2 + 2x - 4\right) y$

```
Plot[f, {t, 0, 10}]
```



Si provi a vedere la definizione associata ad *f*

```
? f
```

```
Global`f
```

$f = -4 + 2\,x - 3\,x^2 + 2\,x^3$

Si comprende che *f* dipende strettamente dalla variabile chiamata *x*. Inoltre, se alla *x* si assegna un valore, *f* diventerà un numero non più una funzione ed in alcuni casi questo diventa un problema per la corretta esecuzione delle operazioni. Si riportano alcu esempi.

```
x = 10
```

10

Ovviamente *f* mantiene sempre la sua definizione

```
? f
```

```
Global`f
```

$f = -4 + 2\,x - 3\,x^2 + 2\,x^3$

Però non appena si utilizza *f* in una espressione di input, essa viene sostituita con il suo valore e poi alla *x* viene a sua volta sostitu ito il suo valore 10.

```
D[f, x]
```

General::ivar : 10 is not a valid variable. ≫

$$\frac{\partial 1716}{\partial 10}$$

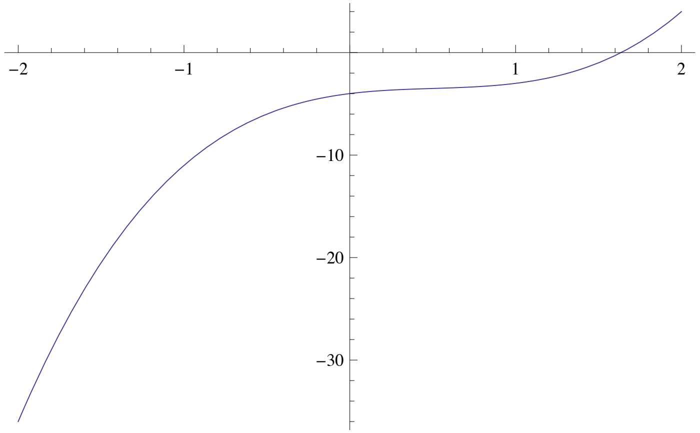
```
Integrate[f, x]
```

Integrate::ilim : Invalid integration variable or limit(s) in 10. ≫

$$\int 1716\, d10$$

La **Plot** funziona, perchè prima di valutare la *f* essa definisce i valori per la variabile *x* considerata “interna” alla **Plot** stessa e p la sostituisce nella espressione di *f*.

```
Plot[f, {x, -2, 2}]
```



Dunque, è sempre preferibile isolare la variabili di un'espressione da quelle "globali", usando una definizione che preveda anch
l'introduzione di argomenti: si passa dal definire una variabile *f* a definire una funzione *f*(*x*). Nel caso precedente:

```
Clear[f, x]
```

```
f[x_] = -4 + 2 x - 3 x^2 + 2 x^3
```

$2x^3 - 3x^2 + 2x - 4$

```
?? f
```

Global`f

```
f[x_] = -4 + 2 x - 3 x^2 + 2 x^3
```

Ora per poter usare *f* bisogna anche indicare un argomento, che può essere una qualsiasi espressione di *Mathematica*

```
f[r]
```

$2r^3 - 3r^2 + 2r - 4$

```
f[1]
```

-3

```
f[Cos[w]]
```

$2\cos^3(w) - 3\cos^2(w) + 2\cos(w) - 4$

```
f["stringa"]
```

$2\,\text{stringa}^3 - 3\,\text{stringa}^2 + 2\,\text{stringa} - 4$

Ora si provi ad osservare cosa succede nel seguente caso.

```
x = 10
```

10

```
g[x_] = -4 + 2 x - 3 x^2 + 2 x^3
```

1716

```
?? g
```

```
Global`g
```

```
g[x_] = 1716
```

Si noti che si è usato lo stesso codice per definire $f(x)$

```
?? f
```

```
Global`f
```

```
f[x_] = -4 + 2 x - 3 x^2 + 2 x^3
```

La differenza tra i due risultati è dovuta al comando **Set** (simbolo =). Infatti la corretta interpretazione di una riga del tipo

espressione_sinistra = *espressione_destra*

è la seguente:

"valuta *espressione_destra* ed il risultato ottenuto assegnalo all'espressione che risulta dalla valutazione d *espressione_sinistra*".

Entrambe le valutazioni (lato sinistro e destro) vengono effettuate nel momento della definizione. Questo significa che la situ azione di memoria corrente influenza la definizione. Da cui nell'esempio precedente la funzione $g(x)$ è diventata un numero pe chè al momento della sua definizione la variabile x aveva il valore 10. Dunque $g(x)$ ha preso il risultato della valutazione dell'espre sione $-4 + 2\,x - 3\,x^2 + 2\,x^3$

Si riporta un altro esempio, dove la valutazione immediata comporta addirittura un errore.

```
Clear[f, g, x]
```

```
f[x_] = x[[1]] + x[[2]]
```

Part::partd : Part specification $x[[1]]$ is longer than depth of object. ≫

Part::partd : Part specification $x[[2]]$ is longer than depth of object. ≫

Part::partd : Part specification $x[[1]]$ is longer than depth of object. ≫

General::stop : Further output of Part::partd will be suppressed during this calculation. ≫

$x[[1]] + x[[2]]$

Non è possibile definire una funzione che estrae una parte dell'argomento utilizzando il **Set** perchè il lato destro dell'assegnazion viene valutato immediatamente, ma al momento della sua esecuzione x ancora non esiste: si suppone esisterà quando f verrà inv cata.
Dunque, oltre a **Set** esiste il comando **SetDelayed** che consente di rimandare la valutazione del lato destro dell'assegnazione momento in cui la funzione viene invocata. Si ripropongono gli esempi di prima:

```
x = 10
```

10

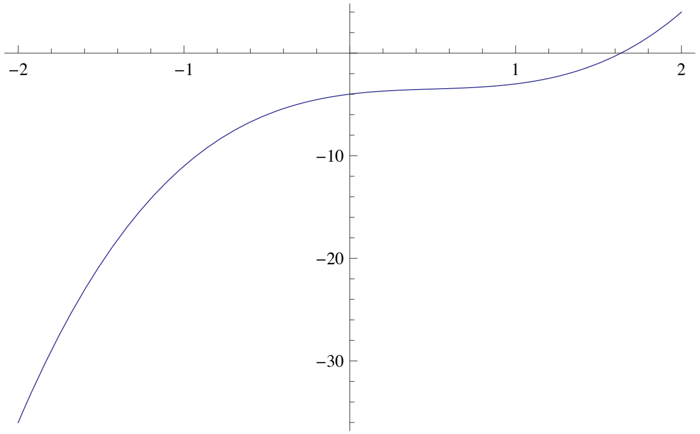
```
g[x_] := -4 + 2 x - 3 x^2 + 2 x^3
```

```
?? g
```

Global`g

```
g[x_] := -4 + 2 x - 3 x^2 + 2 x^3
```

```
Plot[g[x], {x, -2, 2}]
```



```
f[x_] := x⟦1⟧ + x⟦2⟧
f[{2, 2}]
```

4

Le basi del linguaggio *Mathematica*: Il concetto di opzione

Quasi tutte le funzioni native di *Mathematica* mettono a disposizione una o più opzioni. Il concetto è molto semplice: ci sono sem pre tanti modi diversi per eseguire una certa operazione, spesso con il raggiungimento del medesimo risultato ma con differen rappresentazioni. Ad esempio, il grafico di una funzione - pur essendo sempre lo stesso - può essere rappresentato in numero stili diversi. Anche nel caso degli algoritmi di calcolo, le opzioni possono fornire un comodo strumento per personalizzare l'es cuzione di una determinata operazione.
Vediamo ad esempio quante opzioni ha l'istruzione **Plot**.

```
Options[Plot]
```

{AlignmentPoint → Center, AspectRatio → $\frac{1}{\phi}$, Axes → True, AxesLabel → None, AxesOrigin → Automatic, AxesStyle → {}, Background → None, BaselinePosition → Automatic, BaseStyle → {}, ClippingStyle → None, ColorFunction → Automatic, ColorFunctionScaling → True, ColorOutput → Automatic, ContentSelectable → Automatic, CoordinatesToolOptions → Automatic, DisplayFunction :→ $DisplayFunction, Epilog → {}, Evaluated → Automatic, EvaluationMonitor → None, Exclusions → Automatic, ExclusionsStyle → None, Filling → None, FillingStyle → Automatic, FormatType :→ TraditionalForm, Frame → False, FrameLabel → None, FrameStyle → {}, FrameTicks → Automatic, FrameTicksStyle → {}, GridLines → None, GridLinesStyle → {}, ImageMargins → 0., ImagePadding → All, ImageSize → Automatic, ImageSizeRaw → Automatic, LabelStyle → {}, MaxRecursion → Automatic, Mesh → None, MeshFunctions → {#1 &}, MeshShading → None, MeshStyle → Automatic, Method → Automatic, PerformanceGoal :→ $PerformanceGoal, PlotLabel → None, PlotLegends → None, PlotPoints → Automatic, PlotRange → {Full, Automatic}, PlotRangeClipping → True, PlotRangePadding → Automatic, PlotRegion → Automatic, PlotStyle → Automatic, PreserveImageOptions → Automatic, Prolog → {}, RegionFunction → (True &), RotateLabel → True, TargetUnits → Automatic, Ticks → Automatic, TicksStyle → {}, WorkingPrecision → MachinePrecision}

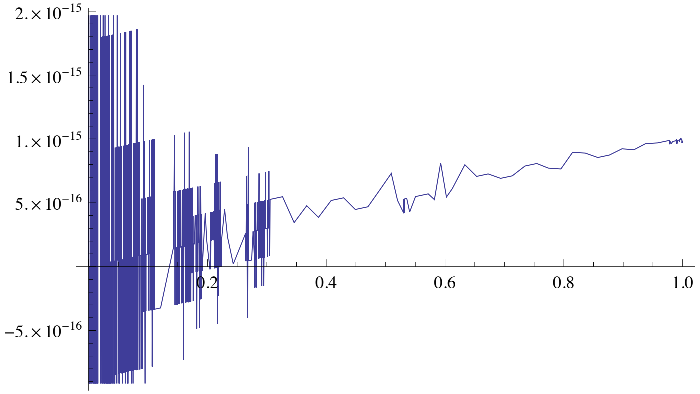
```
Length[%]
```

59

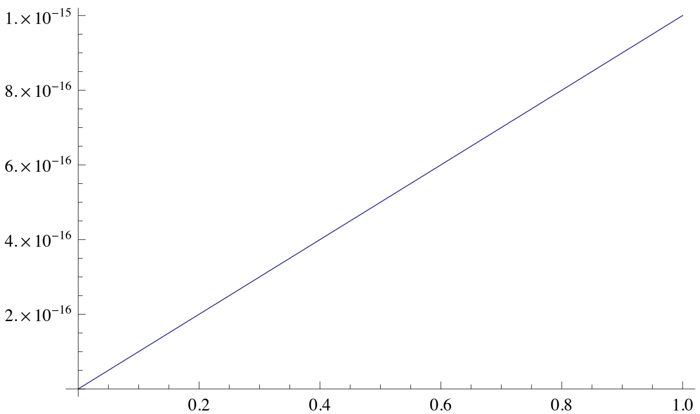
Vediamone qualcuna.

La precisione del calcolo (WorkingPrecision)

```
Plot[z/10^15 + Im[WeierstrassZeta[z, {3, 4}]], {z, 0, 1}]
```




```
Plot[z / 10^15 + Im[WeierstrassZeta[z, {3, 4}]], {z, 0, 1}, WorkingPrecision → 32]
```

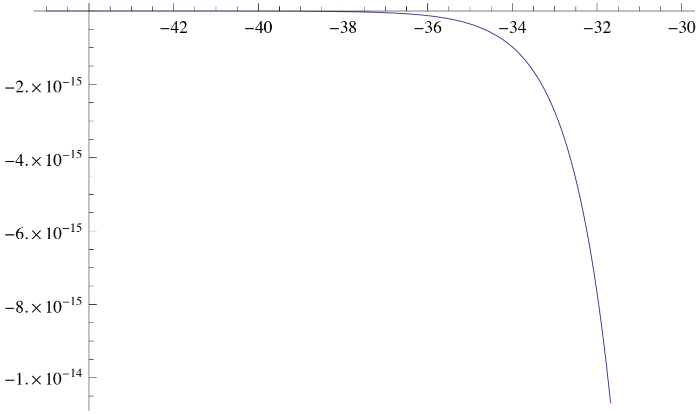



```
Clear[f, x]
```

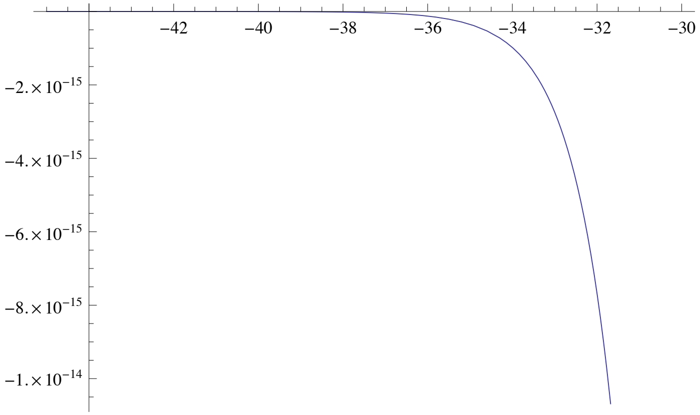
$$f = \int_{-\infty}^{t} \frac{20\,\mathrm{Exp}[x]}{x}\, dx$$

ConditionalExpression[20 Ei(t), Im(t) $\neq$ 0 $\lor$ Re(t) < 0]

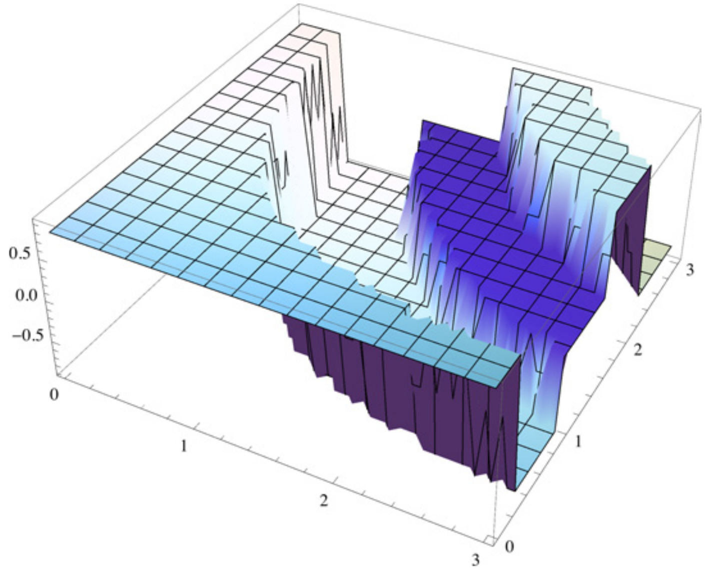
```
Plot[Sin[f], {t, -45, -30}]
```



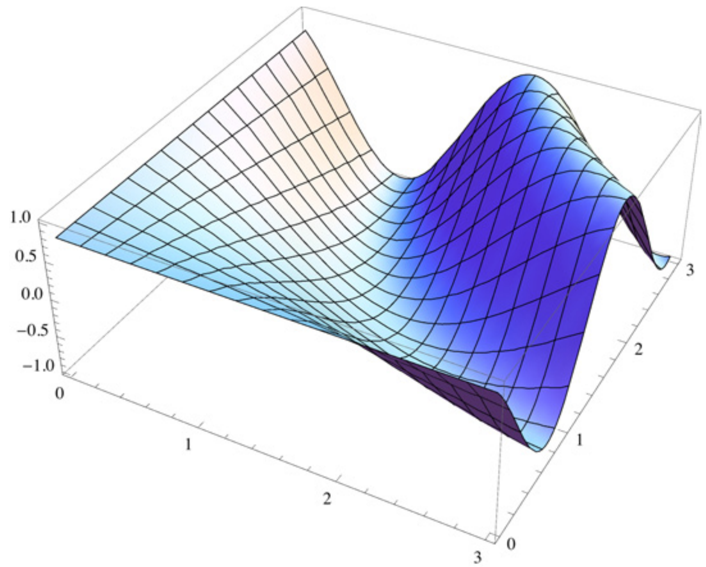

```
Plot[Sin[f], {t, -45, -30}, WorkingPrecision → 32]
```




```
Plot3D[Sin[x y + 10^16], {x, 0, 3}, {y, 0, 3}]
```

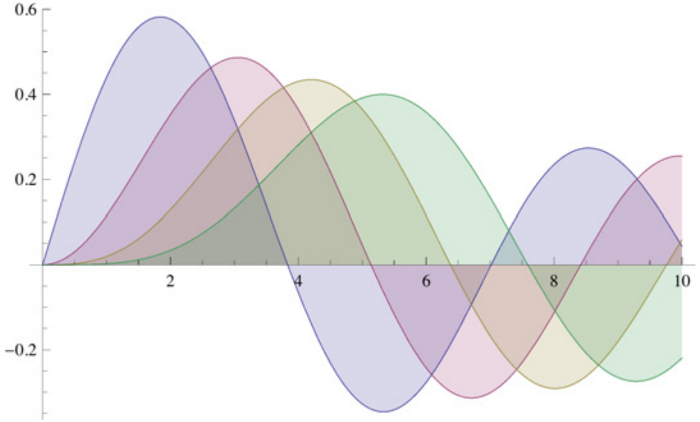


Plot3D[Sin[x y + 10 ^ 16], {x, 0, 3}, {y, 0, 3}, WorkingPrecision → 20]

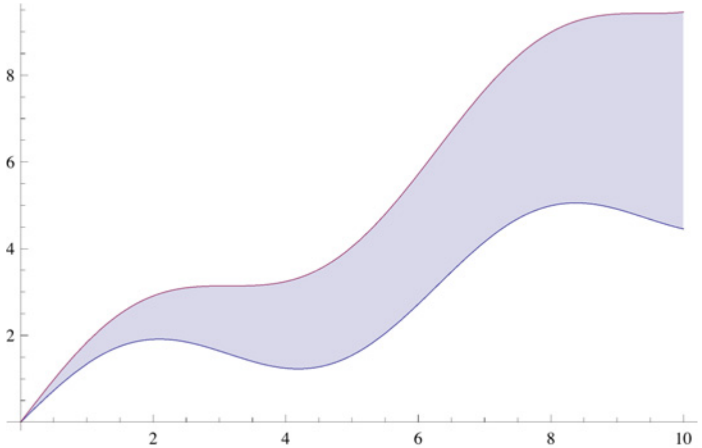


Il riempimento dell'area sottostante la curva o tra più curve (Filling):

Plot[Evaluate[Table[BesselJ[n, x], {n, 4}]], {x, 0, 10}, Filling → Axis]

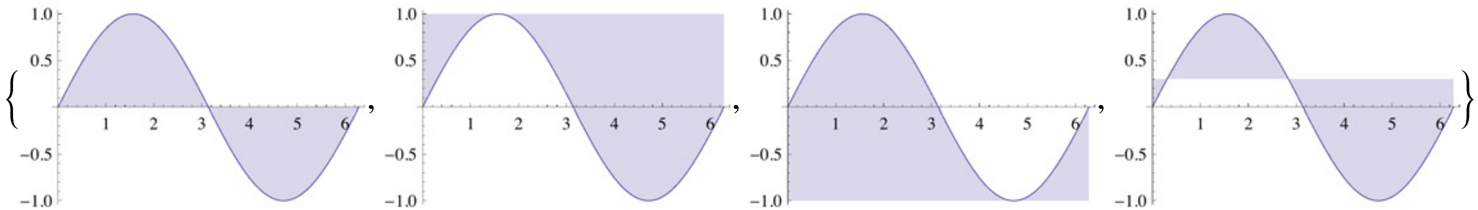



```
Plot[{Sin[x] + x / 2, Sin[x] + x}, {x, 0, 10}, Filling → {1 → {2}}]
```



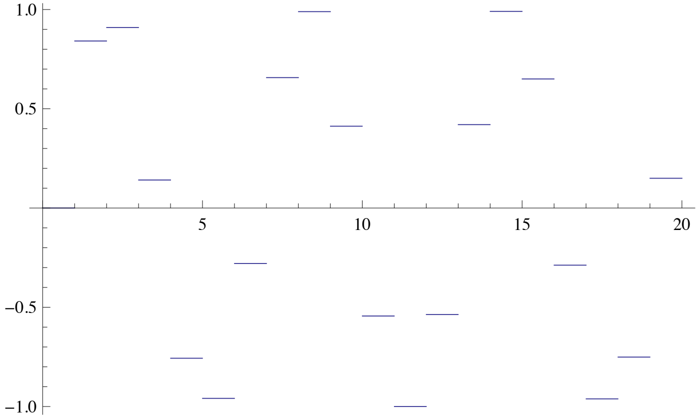
Varie possibilità di personalizzare il filling:

```
Table[Plot[Sin[x], {x, 0, 2 Pi}, Filling → f], {f, {Axis, Top, Bottom, 0.3}}]
```

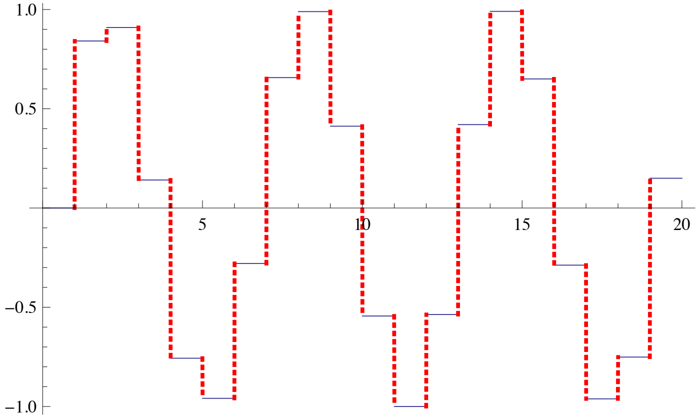


Evidenziare le singolarità (Exclusions / ExclusionsStyle):

```
Plot[Sin[Floor[x]], {x, 0, 20}]
```

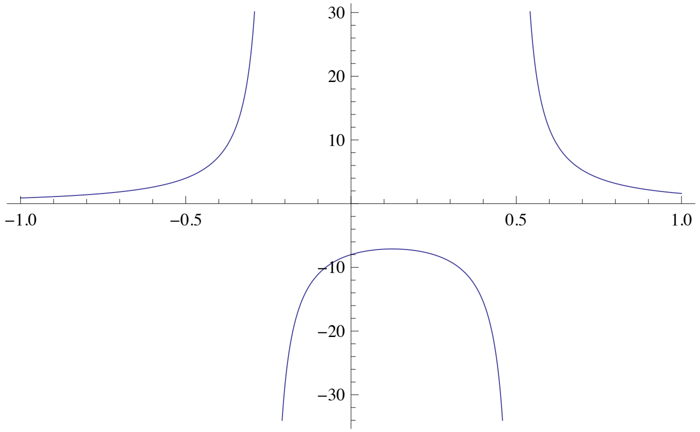



```
Plot[Sin[Floor[x]], {x, 0, 20}, ExclusionsStyle → Directive[{Dashing[{.0025, .01}], Red, Thick}]]
```



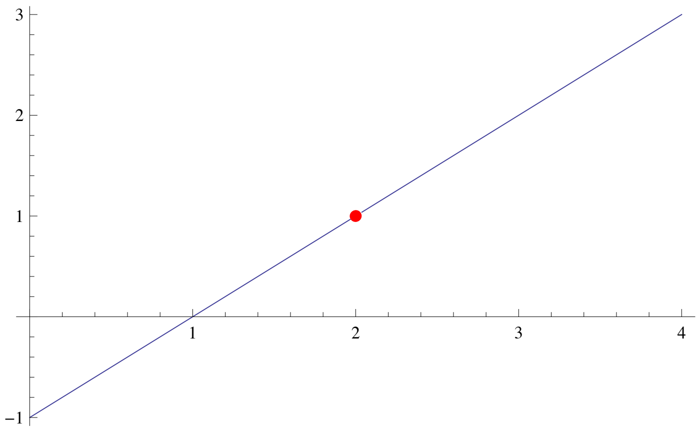
È possibile specificare i punti da escludere:

$$\text{Plot}\left[\frac{1}{\left(x+\frac{1}{4}\right)\left(x-\frac{1}{2}\right)}, \{x, -1, 1\}, \text{Exclusions} \to \left\{x == -\frac{1}{4}, x == \frac{1}{2}\right\}\right]$$

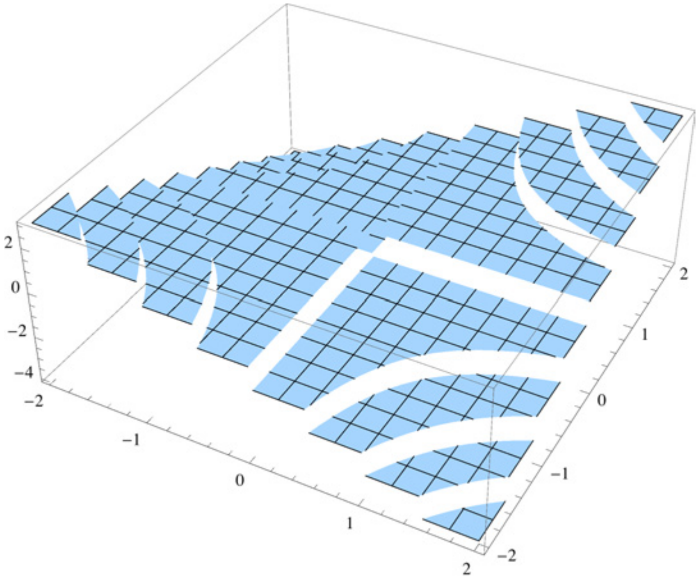


Anche le singolarità eliminabili possono essere indicate utilizzando la stessa tecnica:

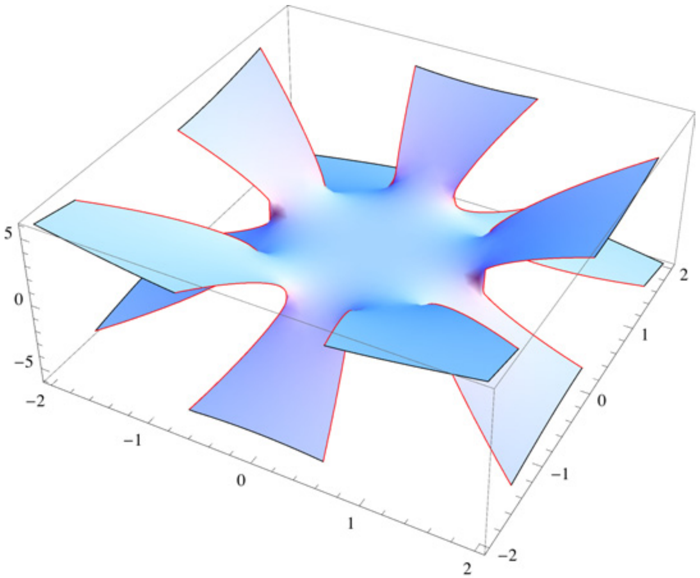
```
Plot[(x^2 - 3 x + 2)/(x - 2), {x, 0, 4}, Exclusions → {x == 2},
 ExclusionsStyle → {None, {AbsolutePointSize[6], Red}}]
```



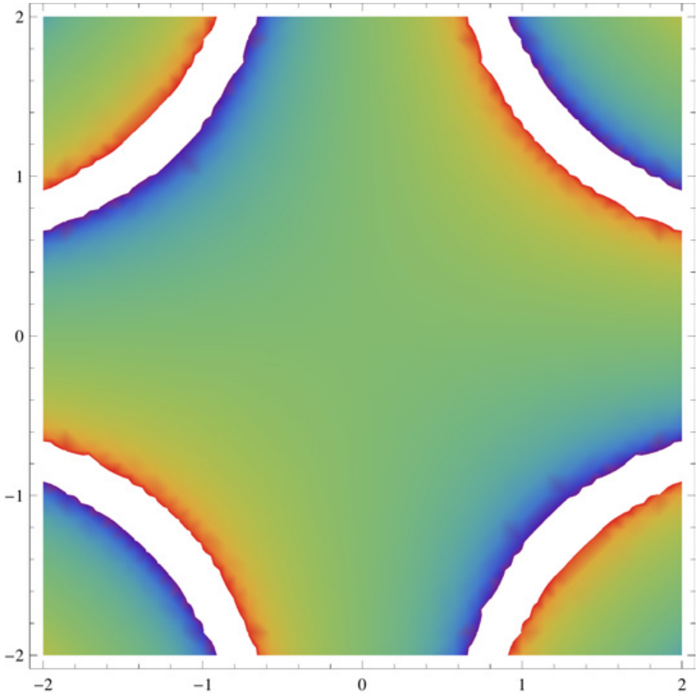

```
Plot3D[Floor[x y ], {x, -2, 2}, {y, -2, 2}]
```




```
Plot3D[Im[ArcSin[(x + I y) ^ 5]], {x, -2, 2}, {y, -2, 2}, ExclusionsStyle → {None, Red}, Mesh → None]
```



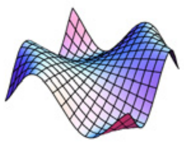

```
DensityPlot[Tan[x y], {x, -2, 2}, {y, -2, 2}, ColorFunction → "Rainbow", Exclusions → {Cos[x y] == 0}]
```



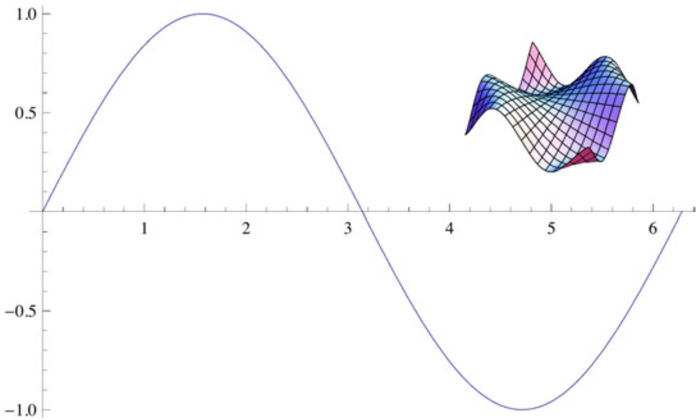
Aggiungere elementi non direttamente collegati al calcolo del grafico (Epilog / Inset).

Inset è una direttiva (non un'opzione) che permette di aggiungere un oggetto all'interno di un grafico, ad esempio trami l'opzione **Epilog**. **Inset** è molto più potente della direttiva **Text** che permette di inserire un testo. Infatti con **Inset** si possor inserire anche oggetti quali panel, slider, ecc.

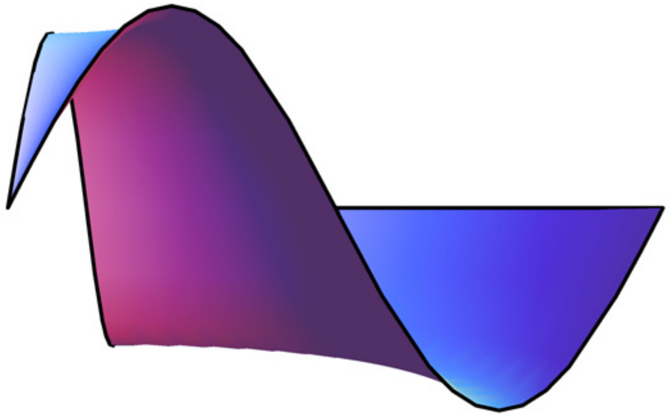
```
gr = Plot3D[Sin[x y], {x, -2, 2}, {y, -2, 2}, Axes → False, ImageSize → 100, Boxed → False]
```



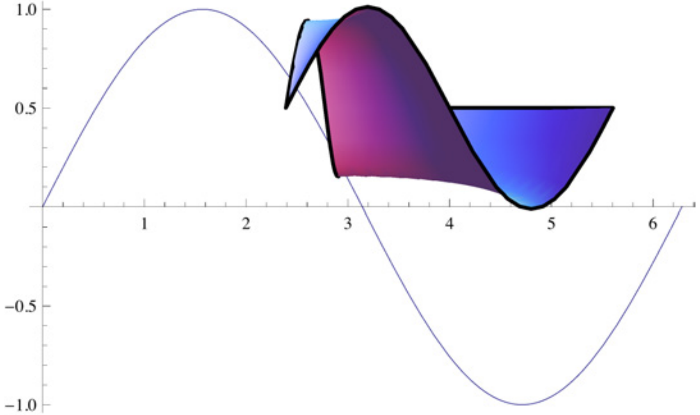
```
Plot[Sin[x], {x, 0, 2 π}, Epilog → Inset[gr, {5, .5}]]
```



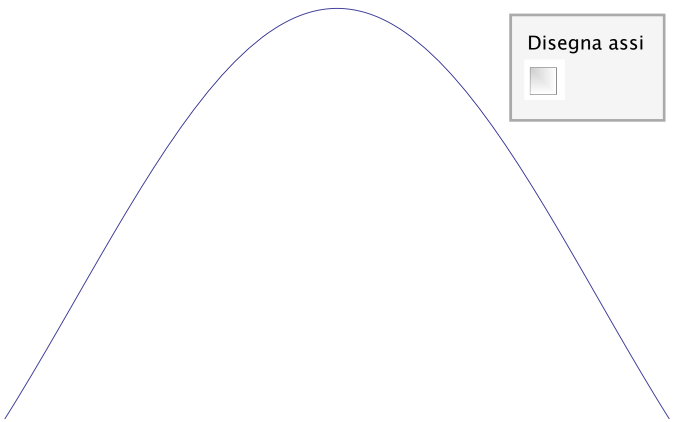

```
gr = RevolutionPlot3D[Sin[x], {x, 0, 2 Pi}, {θ, 0, Pi / 2}, Axes → False,
  Boxed → False, ViewPoint → Front, Mesh → None, BoundaryStyle → Thick]
```



```
Plot[Sin[x], {x, 0, 2 π}, Epilog → Inset[gr, {4, .5}]]
```




```
Dynamic[Plot[Cos[x], {x, -2, 2}, Axes → assi, ImagePadding → 0,
  Epilog → Inset[Panel[Column[{"Disegna assi", Checkbox[Dynamic[assi]]}]], {1.5, .8}]]]
```



Vediamo un esempio di opzione per la modifica dell'algoritmo da usare per un determinato calcolo.

```
NIntegrate[1 / Sqrt[x], {x, 0, 1}, MaxPoints -> 20]
```

NIntegrate::maxp : The integral failed to converge after 33 integrand
evaluations. NIntegrate obtained 1.95581 and 0.067813 for the integral and error estimates. ≫

1.95581

Modificando il numero massimo di punti:

```
NIntegrate[1 / Sqrt[x], {x, 0, 1}, MaxPoints -> 1000]
```

2.

```
NIntegrate[Exp[Cos[x]], {x, 0, 10}, Method → {"RiemannRule", "Type" → "Left"}, PrecisionGoal → 2]
```

12.1732

```
NIntegrate[Exp[Cos[x]], {x, 0, 10}, Method → {"RiemannRule", "Type" → "Right"}, PrecisionGoal → 2]
```

12.0847

Per default **Solve** utilizza le funzioni inverse per risolvere equazioni complesse non polinomiali.

```
Solve[Sin[x] == 1 / 2., x]
```

Solve::ifun :
Inverse functions are being used by Solve, so some solutions may not be found; use Reduce for complete solution information. ≫

$\{\{x \to 0.523599\}\}$

Si può chiedere a **Solve** di usare il metodo `Reduce` per trovare il set completo di soluzioni:

```
Solve[Sin[x] == 1 / 2., x, Method → Reduce]
```

Solve::ratnz : Solve was unable to solve the system with inexact coefficients.
The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

$\{\{x \to \text{ConditionalExpression}[6.28319\, c_1 + 0.523599,\ c_1 \in \mathbb{Z}]\},\ \{x \to \text{ConditionalExpression}[6.28319\, c_1 + 2.61799,\ c_1 \in \mathbb{Z}]\}\}$

Esempi di funzionalità: calcolo simbolico

Es. Equazioni.

```
Solve[a x^4 + b x + c == 0, x]
```

$$\left\{\left\{x \to \frac{1}{2} \sqrt{\frac{4 \sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9 a b^2 + \sqrt{3} \sqrt{27 a^2 b^4 - 256 a^3 c^3}}} + \frac{\sqrt[3]{9 a b^2 + \sqrt{3} \sqrt{27 a^2 b^4 - 256 a^3 c^3}}}{\sqrt[3]{2}\; 3^{2/3} a}} - \right.\right.$$

$$\frac{1}{2} \sqrt{\left(-\frac{2 b}{a \sqrt{\frac{4 \sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9 a b^2+\sqrt{3} \sqrt{27 a^2 b^4-256 a^3 c^3}}} + \frac{\sqrt[3]{9 a b^2+\sqrt{3} \sqrt{27 a^2 b^4-256 a^3 c^3}}}{\sqrt[3]{2}\; 3^{2/3} a}}} - \right.}$$

$$\left.\left. \frac{\sqrt[3]{9 a b^2 + \sqrt{3} \sqrt{27 a^2 b^4 - 256 a^3 c^3}}}{\sqrt[3]{2}\; 3^{2/3} a} - \frac{4 \sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9 a b^2 + \sqrt{3} \sqrt{27 a^2 b^4 - 256 a^3 c^3}}}\right)\right\},$$

$$\left\{x \to \frac{1}{2}\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\ c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}+\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\ 3^{2/3}\,a}}+\frac{1}{2}\sqrt{\left(-\frac{2\,b}{a\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\ c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}+\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\ 3^{2/3}\,a}}}-\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\ 3^{2/3}\,a}-\frac{4\sqrt[3]{\frac{2}{3}}\ c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\ \sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}\right)}\right\},$$

$$\left\{x \to -\frac{1}{2}\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}} + \frac{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}}{\sqrt[3]{2}\, 3^{2/3}\, a}} - \frac{1}{2}\sqrt{\left(\frac{2\, b}{a\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}} + \frac{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}}{\sqrt[3]{2}\, 3^{2/3}\, a}}} - \frac{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}}{\sqrt[3]{2}\, 3^{2/3}\, a} - \frac{4\sqrt[3]{\frac{2}{3}}\, c}{\sqrt[3]{9\, a\, b^2 + \sqrt{3}\, \sqrt{27\, a^2\, b^4 - 256\, a^3\, c^3}}}\right)}\right\},$$

$$\left\{x\to\frac{1}{2}\sqrt{\left(\frac{2\,b}{a\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\,c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}+\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\,3^{2/3}\,a}}}-\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\,3^{2/3}\,a}-\frac{4\sqrt[3]{\frac{2}{3}}\,c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}\right)}-\frac{1}{2}\sqrt{\frac{4\sqrt[3]{\frac{2}{3}}\,c}{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}+\frac{\sqrt[3]{9\,a\,b^2+\sqrt{3}\,\sqrt{27\,a^2\,b^4-256\,a^3\,c^3}}}{\sqrt[3]{2}\,3^{2/3}\,a}}\right\}\Bigg\}$$

Ora anche **Solve** ammette la specifica del dominio di appartenenza delle variabili.

```
Solve[a x^2 + b x + c == 0, x]
```

$$\left\{\left\{x\to\frac{-\sqrt{b^2-4\,a\,c}-b}{2\,a}\right\},\left\{x\to\frac{\sqrt{b^2-4\,a\,c}-b}{2\,a}\right\}\right\}$$

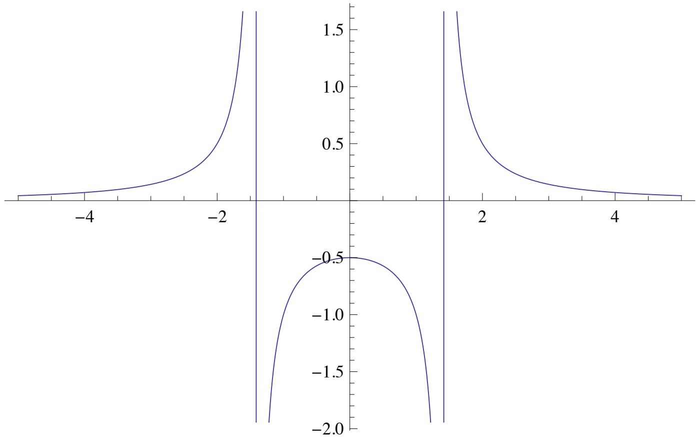
```
Solve[a x^2 + b x + c == 0, x, Reals]
```

$$\left\{\left\{x \to \text{ConditionalExpression}\left[-\frac{1}{2}\sqrt{\frac{b^2-4\,a\,c}{a^2}} - \frac{b}{2\,a}, \left(a < \frac{b^2}{4\,c} \bigwedge c > 0\right) \bigvee \left(c < 0 \bigwedge a > \frac{b^2}{4\,c}\right)\right]\right\},\right.$$

$$\left.\left\{x \to \text{ConditionalExpression}\left[\frac{1}{2}\sqrt{\frac{b^2-4\,a\,c}{a^2}} - \frac{b}{2\,a}, \left(a < \frac{b^2}{4\,c} \bigwedge c > 0\right) \bigvee \left(c < 0 \bigwedge a > \frac{b^2}{4\,c}\right)\right]\right\}\right\}$$

Utilizzo di **Reduce** per il calcolo degli asintoti della funzione $\frac{1}{y^2-2}$ sfruttando la definizione di limite

```
Plot[1/(y^2 - 2), {y, -5, 5}]
```




```
Reduce[ForAll[M, M > 0,
  Exists[δ, δ > 0, ForAll[y, Element[x | y, Reals] && 0 < Abs[y - x] < δ, Abs[1/(y^2 - 2)] > M]]], x]
```

$$x = -\sqrt{2} \bigvee x = \sqrt{2}$$

Es. Integrali.

$$\int \frac{1}{x^5 - 1} \, dx$$

$$\frac{1}{20}\left(\left(\sqrt{5} - 1\right) \log\left(x^2 - \frac{1}{2}\left(\sqrt{5} - 1\right) x + 1\right) - \left(1 + \sqrt{5}\right) \log\left(x^2 + \frac{1}{2}\left(1 + \sqrt{5}\right) x + 1\right) + 4 \log(1 - x) - 2 \sqrt{2\left(5 + \sqrt{5}\right)} \tan^{-1}\left(\frac{4x - \sqrt{5} + 1}{\sqrt{2\left(5 + \sqrt{5}\right)}}\right) - 2\sqrt{10 - 2\sqrt{5}} \tan^{-1}\left(\frac{4x + \sqrt{5} + 1}{\sqrt{10 - 2\sqrt{5}}}\right)\right)$$

$$\int_{-\infty}^{\infty} \frac{\mathrm{Boole}\left[\frac{1}{2} < \sin(x) < \frac{2}{3}\right]}{x^2} \, dx$$

$$\frac{1}{2} \mathrm{Tan}\left[\frac{1}{2} \mathrm{ArcTan}\left[\frac{2}{\sqrt{5}}\right]\right] \left(1 - \mathrm{Cot}\left[\frac{1}{2} \mathrm{ArcTan}\left[\frac{2}{\sqrt{5}}\right]\right]^2 + 2\sqrt{3}\, \mathrm{Cot}\left[\frac{1}{2} \mathrm{ArcTan}\left[\frac{2}{\sqrt{5}}\right]\right]\right)$$

Volendo si può chiedere il risultato in forma tradizionale:

```
TraditionalForm[%]
```

$$\frac{1}{2} \tan\left(\frac{1}{2} \tan^{-1}\left(\frac{2}{\sqrt{5}}\right)\right)\left(1 - \cot^2\left(\frac{1}{2} \tan^{-1}\left(\frac{2}{\sqrt{5}}\right)\right) + 2\sqrt{3} \cot\left(\frac{1}{2} \tan^{-1}\left(\frac{2}{\sqrt{5}}\right)\right)\right)$$

Aggiungere ipotesi

$$\mathbf{Assuming}\left[x \in \mathbb{R}, \int |2 - |x|| \, dx\right]$$

$$\begin{cases} -2\,x - \frac{x^2}{2} & x \le -2 \\ 4 + 2\,x + \frac{x^2}{2} & -2 < x \le 0 \\ 4 + 2\,x - \frac{x^2}{2} & 0 < x \le 2 \\ 8 - 2\,x + \frac{x^2}{2} & \text{True} \end{cases}$$

```
Series[Min[x^2, 1 + a] Cos[x] + Sin[x], {x, 0, 2}]
```

$$\begin{cases} (1 + a) + x + \left(-\frac{1}{2} - \frac{a}{2}\right) x^2 + O[x]^3 & x \in \text{Reals} \,\&\&\, a \le -1 \\ x + x^2 + O[x]^3 & \text{True} \end{cases}$$

```
Series[Min[x^2, 1 + a] Cos[x] + Sin[x], {x, 0, 2}, Assumptions → a > 0]
```

$x + x^2 + O[x]^3$

Esempi di funzionalità: calcolo numerico

Anche dal punto di vista numerico *Mathematica* ha raggiunto (ed in molti casi superato) i migliori software numerici.

Es. Integrali.

Integrazione numerica di una funzione discontinua:

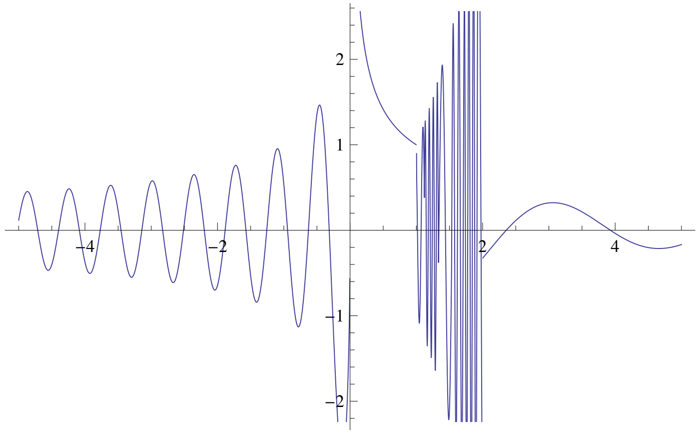
$$\texttt{fun} = \begin{cases} \frac{\texttt{Sin[10 x]}}{\sqrt{\texttt{-x}}} & -\infty < \texttt{x} < 0 \\ \frac{1}{\sqrt{\texttt{x}}} & 0 < \texttt{x} < 1 \\ \texttt{Sin[2000 x] x}^2 & 1 < \texttt{x} < 2 \\ \frac{\texttt{Cos[2 x]}}{\texttt{x}} & 2 < \texttt{x} < \infty \end{cases};$$

```
TraditionalForm[fun]
```

$$\begin{cases} \frac{\sin(10\,x)}{\sqrt{-x}} & -\infty < x < 0 \\ \frac{1}{\sqrt{x}} & 0 < x < 1 \\ x^2 \sin(2000\,x) & 1 < x < 2 \\ \frac{\cos(2\,x)}{x} & 2 < x < \infty \end{cases}$$

Costruiamone il grafico

```
Plot[fun, {x, -5, 5}]
```



Calcoliamone l'integrale da -∞ a +∞

```
NIntegrate[fun, {x, -Infinity, Infinity}]
1.74592
```

Una delle funzioni più frequentemente utilizzata in *Mathematica* è la **N** che serve per convertire una espressione simbolica in u valore numerico. Quando abbiamo terminato un calcolo simbolico e vogliamo vedere il valore numerico possiamo invocare la sul risultato.

```
Cos[4]
Cos[4]
```

```
N[Cos[4]]
-0.653644
```

Tramite **N** possiamo scegliere anche il livello di precisione

```
N[π]
3.14159
```

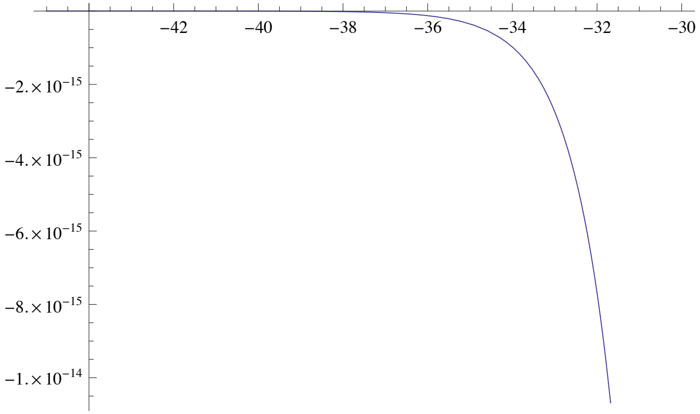
```
N[π, 32]
3.1415926535897932384626433832795
```

Gli algoritmi per la gestione della precisione arbitraria nel calcolo numerico sono utilizzati da tutte le funzioni numeriche, ancl quelle della grafica.

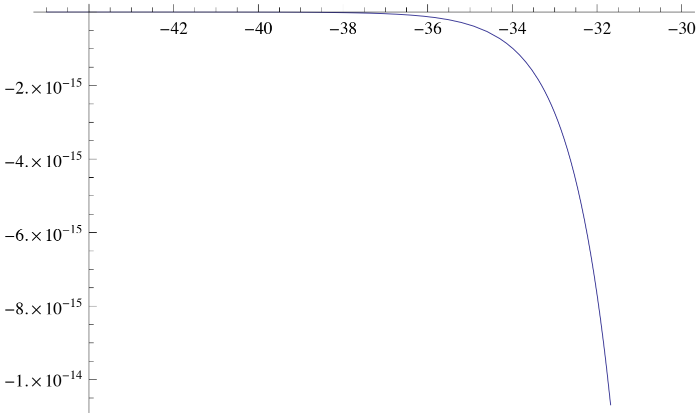
$$f = \int_{-\infty}^{t} \frac{20\,\mathrm{Exp}[x]}{x}\, dx$$

```
ConditionalExpression[20 ExpIntegralEi[t], Im[t] ≠ 0 || Re[t] < 0]
```

```
Plot[Sin[f], {t, -45, -30}]
```




```
Plot[Sin[f], {t, -45, -30}, WorkingPrecision → 32]
```



Anche i numeri random possono avere una precisione arbitraria.

```
data = RandomVariate[BinormalDistribution[1 / 2], 10 ^ 4, WorkingPrecision → 120]; // AbsoluteTiming
{0.876270, Null}
```

```
Length[data]
```

```
10 000
```

```
data // First
```

```
{0.00779005705637295735130163675582578612384410374049700451370897158209352364112897856639423586499379806891363710602373843215,
 -0.743967940296303273791397626269286349331971724669467794589040905350024431034419671718383681490123696195381804038420823345}
```

Se si vuole impostare la precisione uguale per tutti i valori reali nell'ambito di una sessione di lavoro, si può usare la variabi **$Pre** ed il comando **SetPrecision**

```
$Pre = Function[{input}, SetPrecision[input, 32]];
```

```
2 + 3
```

```
5.0000000000000000000000000000000
```

```
Cos[4]
```

```
-0.65364362086361191463916818309778
```

$$\frac{1}{7}$$

```
0.14285714285714285714285714285714
```

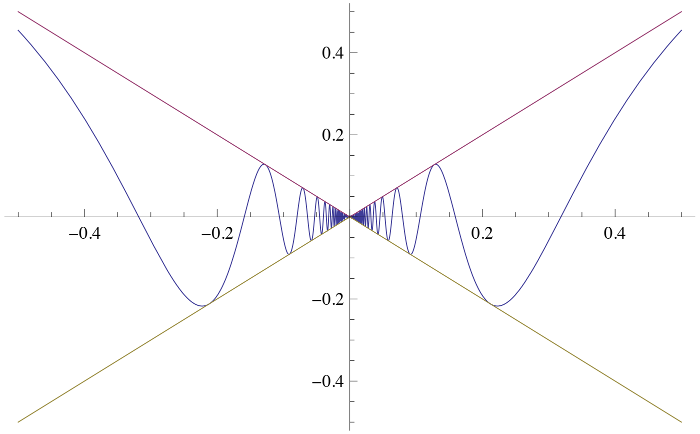
Esempi di funzionalità: la grafica

Mathematica dispone di centinaia di funzioni dedicate espressamente alla grafica in due e tre dimensioni, per funzioni note anali camente o in forma tabellare. Le principali funzioni sono

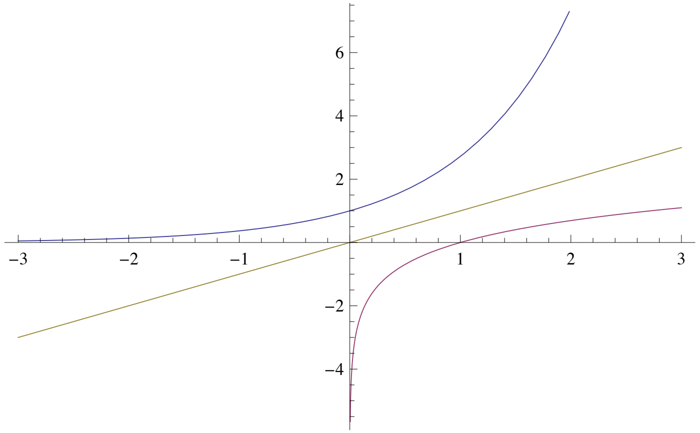
```
? Plot
```

Plot[f, {x, x_{min}, x_{max}}] generates a plot of f as a function of x from x_{min} to x_{max}.
Plot[{f_1, f_2, …}, {x, x_{min}, x_{max}}] plots several functions f_i. ≫

```
Plot[{x Sin[1 / x], Abs[x], -Abs[x]}, {x, -1 / 2, 1 / 2}]
```



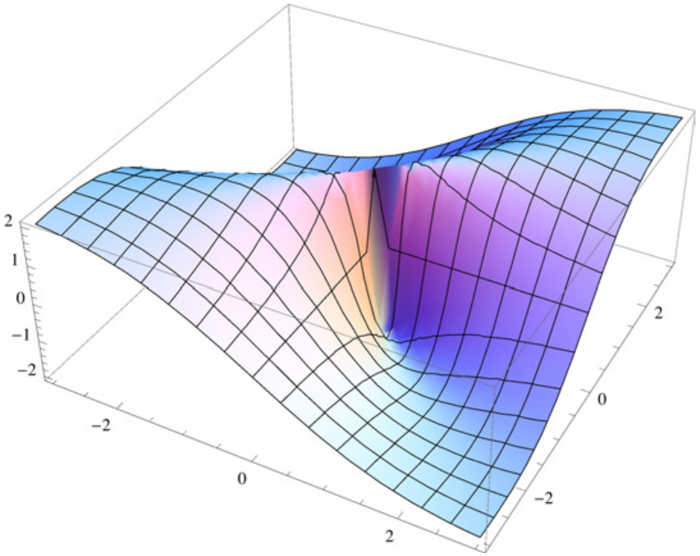

```
Plot[{Exp[x], Log[x], x}, {x, -3, 3}]
```



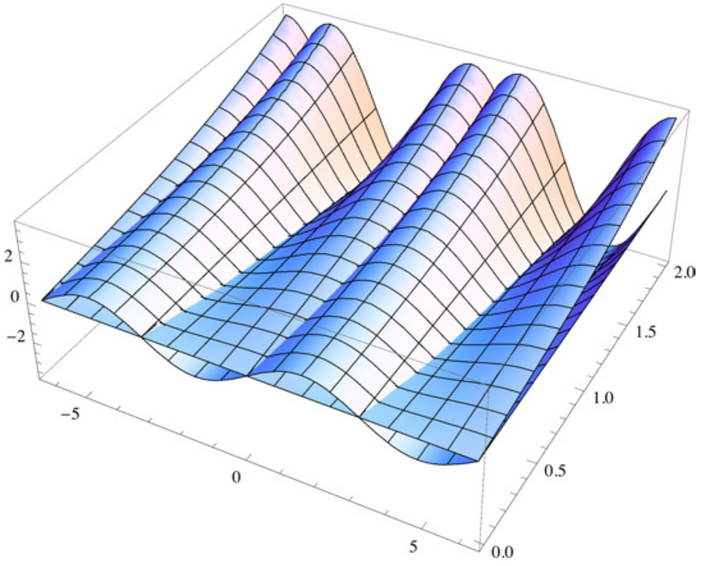

```
?Plot3D
```

Plot3D[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] generates a three-dimensional plot of f as a function of x and y.
Plot3D[{f_1, f_2, ...}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] plots several functions. ≫

Plot3D$\left[\dfrac{10\,x\,y}{2\,x^2 + 3\,y^2}, \{x, -3, 3\}, \{y, -3, 3\}\right]$



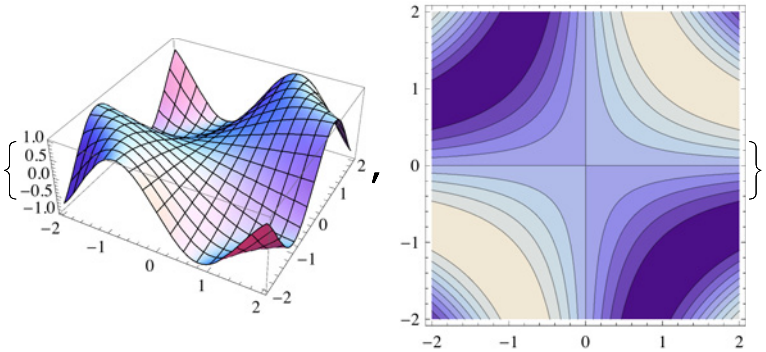

```
Plot3D[{Re[Sin[x + I y]], Im[Sin[x + I y]]}, {x, -2 Pi, 2 Pi}, {y, 0, 2}]
```



? ContourPlot

ContourPlot[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] generates a contour plot of f as a function of x and y.
ContourPlot[f == g, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] plots contour lines for which $f = g$.
ContourPlot[{f_1 == g_1, f_2 == g_2, …}, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] plots several contour lines. ≫

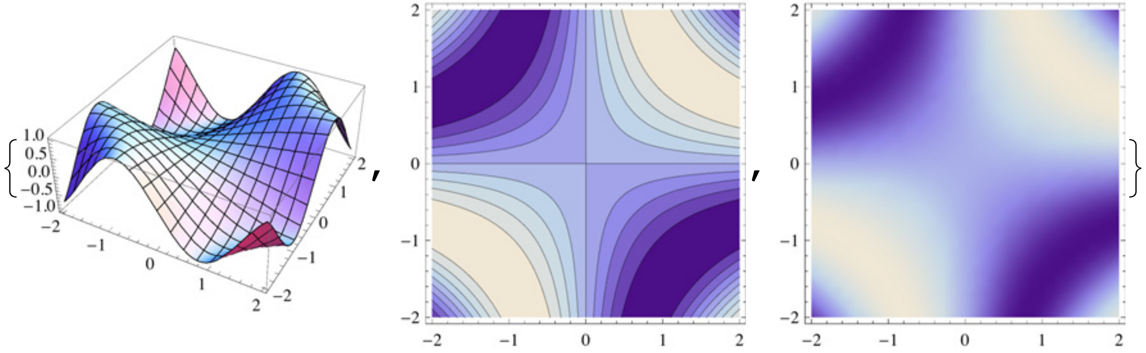
```
{Plot3D[Sin[x y], {x, -2, 2}, {y, -2, 2}], ContourPlot[Sin[x y], {x, -2, 2}, {y, -2, 2}]}
```



? DensityPlot

DensityPlot[f, {x, x_{min}, x_{max}}, {y, y_{min}, y_{max}}] makes a density plot of f as a function of x and y. ≫

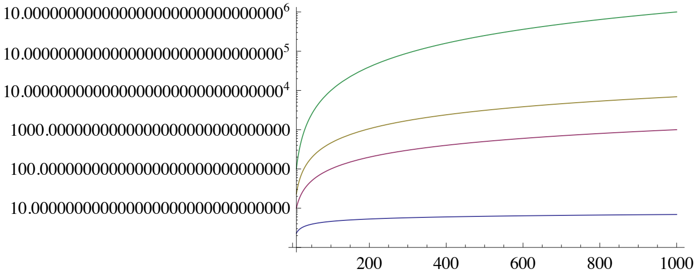
```
{Plot3D[Sin[x y], {x, -2, 2}, {y, -2, 2}],
 ContourPlot[Sin[x y], {x, -2, 2}, {y, -2, 2}], DensityPlot[Sin[x y], {x, -2, 2}, {y, -2, 2}]}
```




```
?LogPlot
```

LogPlot[f, {x, x_{min}, x_{max}}] generates a log plot of f as a function of x from x_{min} to x_{max}.
LogPlot[{f_1, f_2, …}, {x, x_{min}, x_{max}}] generates log plots of several functions f_i. ≫

```
LogPlot[Tooltip@{Log[n], n, n Log[n], n^2}, {n, 10, 10^3}]
```



Esempi di funzionalità: applicazioni dinamiche ed interattive

Mathematica ha rivoluzionato il concetto di computazione interattiva e dinamica, introducendo funzioni dinamiche che istantanemente creano interfacce intuitive e interattive. Le computazioni sottostanti vengono eseguite in run-time.

```
Integrate[1 / (x^3 + 1), x]
```

0.57735026918962576450914878050196 ArcTan[0.57735026918962576450914878050196 (−1.0000000000000000000000000000000 + 2.0000000000000000000000000000000 x)] + 0.33333333333333333333333333333333 Log[1.0000000000000000000000000000000 + x] − 0.16666666666666666666666666666667 Log[1.0000000000000000000000000000000 − 1.0000000000000000000000000000000 x + x^2]

Alla base di Manipulate c'è la funzione **Dynamic**

```
Clear[a, b, c]
```

```
Grid[{{"Statico", Solve[a x^2 + b x + c == 0, x]}, {"Dinamico", Dynamic[Solve[a x^2 + b x + c == 0, x]]}},
 Alignment -> Left, Dividers → All]
```

Statico	$\left\{\left\{x \to \frac{-\sqrt{b^2-4ac}-b}{2a}\right\}, \left\{x \to \frac{\sqrt{b^2-4ac}-b}{2a}\right\}\right\}$
Dinamico	$\left\{\left\{x \to \frac{1}{2}\left(-5-\sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(\sqrt{5}-5\right)\right\}\right\}$

Assegniamo ora dei valori ai parametri a, b, c

```
{a = 0, b = 1, c = -2}
```

{0, 1, −2}

Si può creare un oggetto di tipo SetterBar per modificare più semplicemente i valori di a, b e c

```
SetterBar[Dynamic[a], {-1, 0, 1}]
```

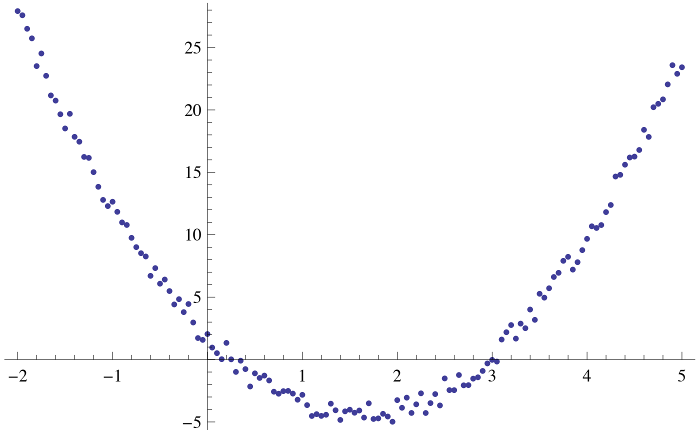
−1	0	1

```
Panel[
 Column[
  {Row[{"Imposta a ", SetterBar[Dynamic[a], Range[-5, 5]]}],
   Row[{"Imposta b ", SetterBar[Dynamic[b], Range[-5, 5]]}],
   Row[{"Imposta c ", SetterBar[Dynamic[c], Range[-5, 5]]}]}]]
```

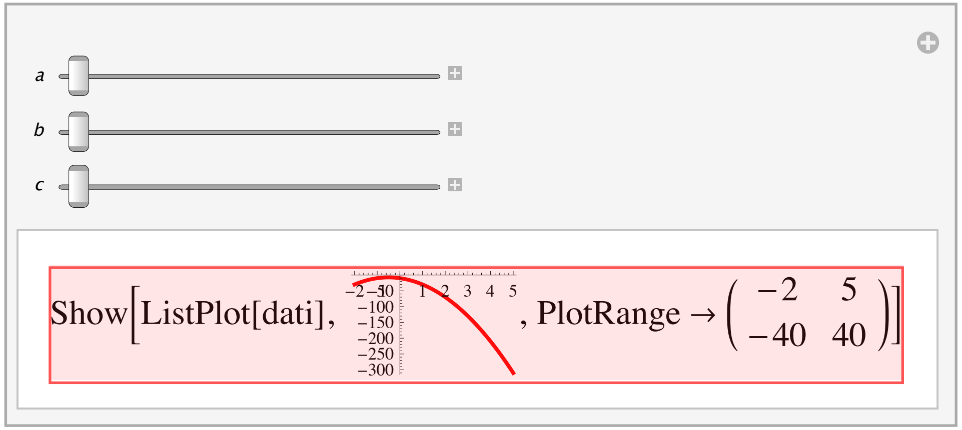
Esempio: un data fitting manuale.

```
dati = Table[{x, 2.5 x^2 - 8 x + 1.87 + RandomReal[{-1, 1}]}, {x, -2, 5, 0.05}];
```

```
ListPlot[dati]
```

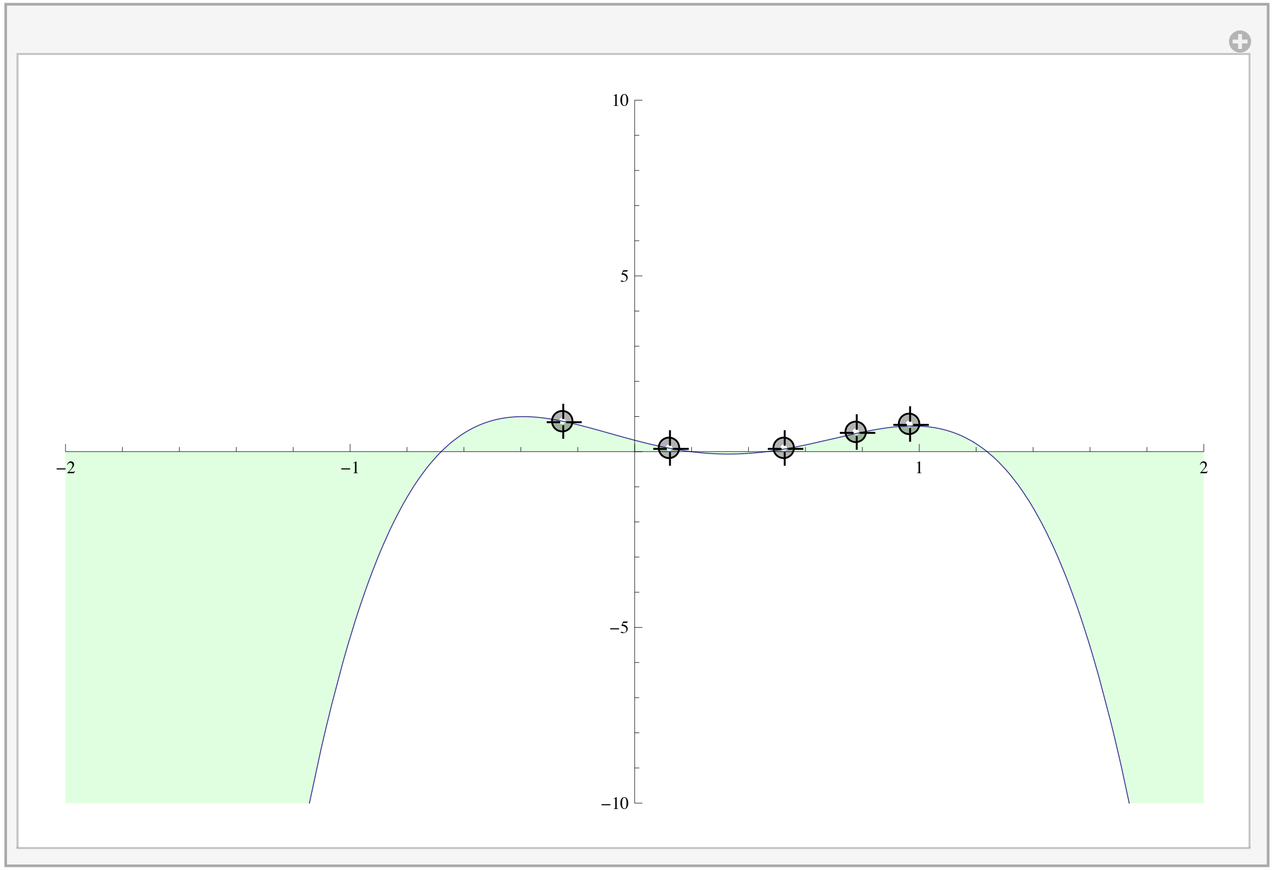



```
Manipulate[
 Show[
  ListPlot[dati],
  Plot[a x^2 + b x + c, {x, -2, 5}, PlotStyle → {Thick, Red}], PlotRange → {{-2, 5}, {-40, 40}}],
 {a, -10, 10},
 {b, -10, 10},
 {c, -10, 10}]
```



Esempio: una interpolazione manuale.

```
Manipulate[
 Plot[InterpolatingPolynomial[pts, x], {x, -2, 2}, PlotRange → {{-2, 2}, {-10, 10}},
  Filling → Axis, FillingStyle → LightGreen, ImageSize → 600],
 {{pts, {{-0.25, 0.86}, {0.53, 0.08}, {0.97, 0.73}, {0.12, 0.10}, {0.78, 0.52}}},
  {-2, -10}, {2, 10}, Locator, LocatorAutoCreate → True}]
```


10
5
−5
−10
−2
−1
1
2

Esempi di funzionalità: le sorgenti dati

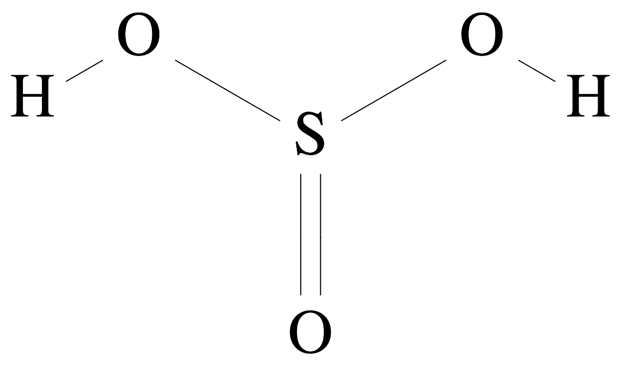
Sorgenti dati computabili

Mathematica mette a disposizione una serie di banche dati affidabili, robuste ed aggiornate costantemente. Questo è un elenc completo delle banche dati **Computable Data**

Vediamone alcuni esempi.

ChemicalData contiene le proprietà di oltre 44.000 compositi:

```
ChemicalData["SulfurousAcid"]
```



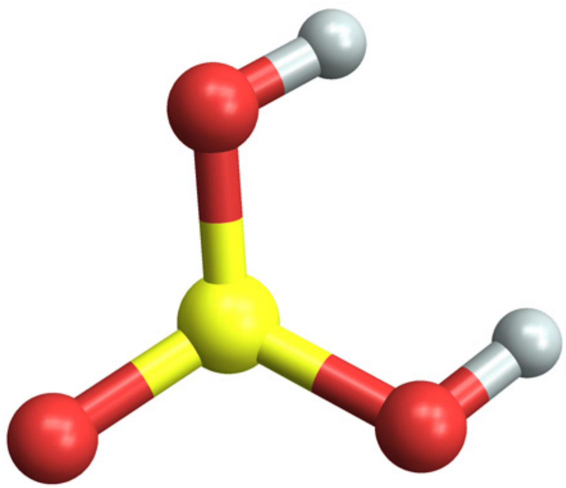
```
ChemicalData["SulfurousAcid", "Properties"]
```

{AcidityConstant, AcidityConstants, AdjacencyMatrix, AlternateNames, AtomPositions, AutoignitionPoint, BeilsteinNumber, BoilingPoint, BondTally, CASNumber, CHColorStructureDiagram, CHStructureDiagram, CIDNumber, Codons, ColorStructureDiagram, CombustionHeat, CompoundFormulaDisplay, CompoundFormulaString, CriticalPressure, CriticalTemperature, Density, DensityGramsPerCC, DielectricConstant, DOTHazardClass, DOTNumbers, EdgeRules, EdgeTypes, EGECNumber, ElementMassFraction, ElementTally, ElementTypes, EUNumber, FlashPoint, FlashPointFahrenheit, FormalCharges, FormattedName, GmelinNumber, HBondAcceptorCount, HBondDonorCount, HenryLawConstant, HildebrandSolubility, HildebrandSolubilitySI, InChI, IonEquivalents, Ions, IonTally, IsoelectricPoint, IsomericSMILES, IUPACName, LogAcidityConstant, LowerExplosiveLimit, MDLNumber, MeltingBehavior, MeltingPoint, Memberships, MolarVolume, MolecularFormulaDisplay, MolecularFormulaString, MolecularWeight, MoleculePlot, Name, NFPAFireRating, NFPAHazards, NFPAHealthRating, NFPALabel, NFPAReactivityRating, NonHydrogenCount, NonStandardIsotopeCount, NonStandardIsotopeNumbers, NonStandardIsotopeTally, NSCNumber, OdorThreshold, OdorType, PartitionCoefficient, pH, Phase, RefractiveIndex, Resistivity, RotatableBondCount, RTECSClasses, RTECSNumber, SideChainAcidityConstant, SMILES, Solubility, SolubilityType, SpaceFillingMoleculePlot, StandardName, StructureDiagram, SurfaceTension, TautomerCount, ThermalConductivity, TopologicalPolarSurfaceArea, UpperExplosiveLimit, VanDerWaalsConstants, VaporDensity, VaporizationHeat, VaporPressure, VaporPressureTorr, VertexCoordinates, VertexTypes, Viscosity}

```
ChemicalData["SulfurousAcid", "AcidityConstant"]
```

0.0169824

```
ChemicalData["SulfurousAcid", "MoleculePlot"]
```



Ovviamente i dati sono disponibili in formato e struttura tali da poter essere immediatamente disponibili in *Mathematica*. Pe tanto, si possono programmare anche complesse applicazioni che sfruttano tali dati e creano report, grafici, modelli, ecc. semplic mente richiedendo i dati ai server Wolfram. Ecco un esempio di come una chiamata a **ChemicalData** si innesta facilmente in ur porzione di codice *Mathematica*.

Una distribuzione dei pesi molecolari:

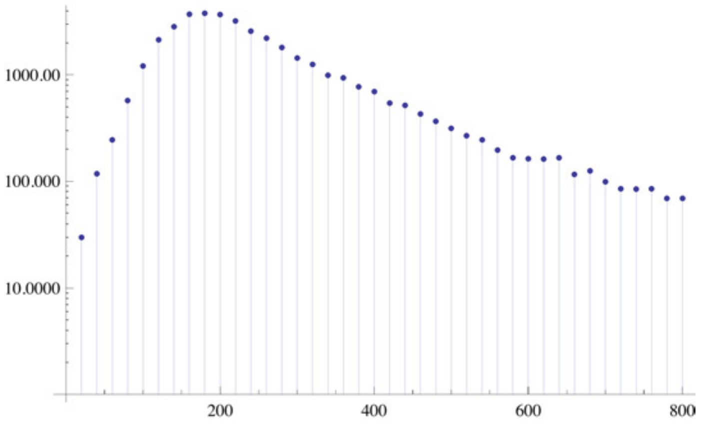
```
all = ChemicalData[];
all = Map[ChemicalData[#, "MolecularWeight"] &, all];
all = Cases[all, _Real];
dist = BinCounts[all, {0, 800, 20}];

$Pre = Function[{input}, SetPrecision[input, 6]];

dist = Transpose[{Table[i, {i, 20, 800, 20}], dist}]
```

$$\begin{pmatrix} 20.0000 & 30.0000 \\ 40.0000 & 117.000 \\ 60.0000 & 244.000 \\ 80.0000 & 574.000 \end{pmatrix}$$

100.000	1208.00
120.000	2132.00
140.000	2847.00
160.000	3721.00
180.000	3815.00
200.000	3691.00
220.000	3212.00
240.000	2580.00
260.000	2221.00
280.000	1806.00
300.000	1444.00
320.000	1255.00
340.000	994.000
360.000	937.000
380.000	772.000
400.000	694.000
420.000	541.000
440.000	516.000
460.000	429.000
480.000	365.000
500.000	313.000
520.000	267.000
540.000	245.000
560.000	196.000
580.000	166.000
600.000	163.000
620.000	161.000
640.000	166.000
660.000	116.000
680.000	125.000
700.000	99.0000

$$\begin{pmatrix} 720.000 & 85.0000 \\ 740.000 & 84.0000 \\ 760.000 & 85.0000 \\ 780.000 & 69.0000 \\ 800.000 & 69.0000 \end{pmatrix}$$

```
ListLogPlot[dist, Filling → Axis]
```



WeatherData fornisce dati meteo in tempo reale da tutte le stazioni del mondo:

```
WeatherData[{"Foggia", 5}]
```

{LIBF, LIBA, LIBE, LIRT, D3927}

```
stazione = "LIBF";
```

```
WeatherData[stazione, "Properties"]
```

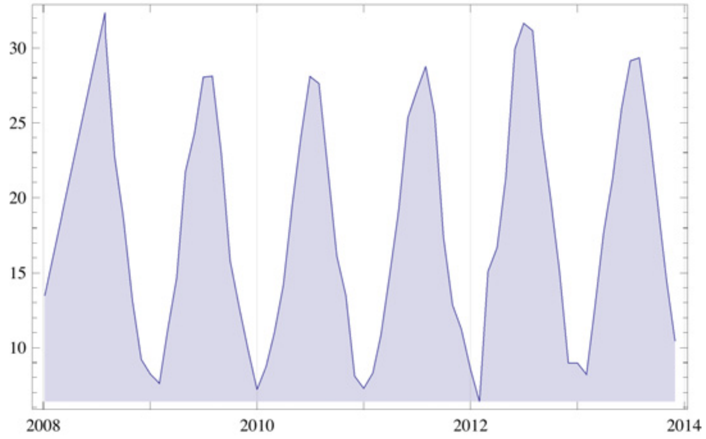
{AlternateStandardNames, CloudCoverFraction, CloudHeight, CloudTypes, Conditions, Coordinates, DewPoint, Elevation, Humidity, Latitude, Longitude, MaxTemperature, MaxWindSpeed, MeanDewPoint, MeanHumidity, MeanPressure, MeanStationPressure, MeanTemperature, MeanVisibility, MeanWindChill, MeanWindSpeed, Memberships, MinTemperature, NCDCID, PrecipitationAmount, PrecipitationRate, PrecipitationTypes, Pressure, PressureTendency, SnowAccumulation, SnowAccumulationRate, SnowDepth, StationName, StationPressure, Temperature, TotalPrecipitation, Visibility, WBANID, WindChill, WindDirection, WindGusts, WindSpeed, WMOID}

```
$Pre =.
```

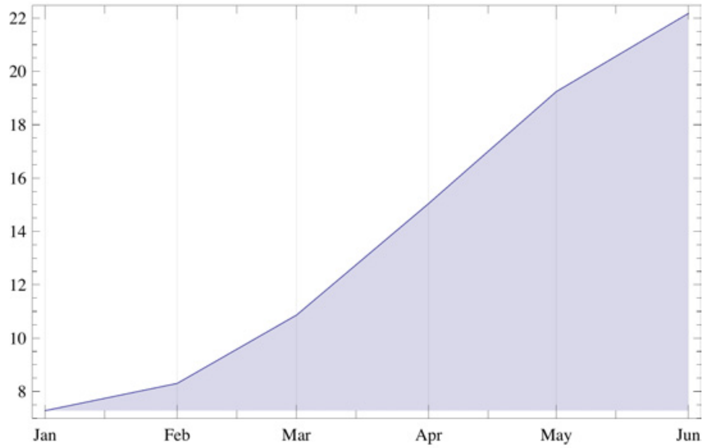
```
WeatherData[stazione, "Temperature", "DateValue"]
```

{{2014, 6, 10, 17, 50, 0}, 29.}

```
DateListPlot[WeatherData[stazione, "MeanTemperature",
  {{2000, 12, 31}, {2013, 12, 31}, "Month"}], Joined → True, Filling → Bottom]
```

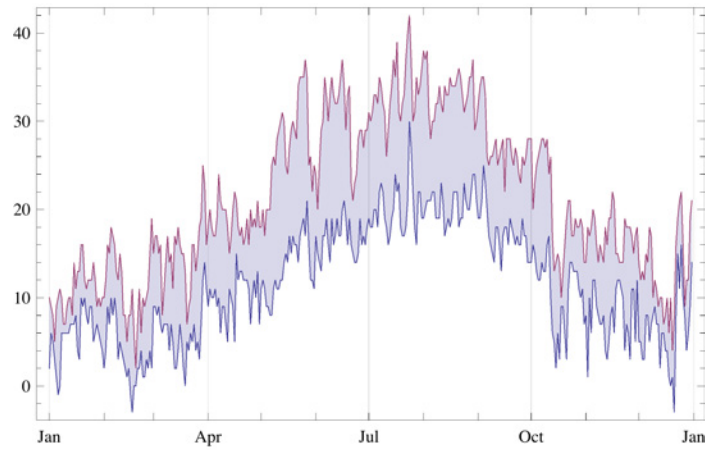



```
DateListPlot[WeatherData[stazione, "Temperature", {{2011, 1, 1}, {2011, 6, 1}, "Month"}],
 Joined → True, Filling → Bottom]
```




```
min = WeatherData[stazione, "MinTemperature", {{2009, 1, 1}, {2009, 12, 31}, "Day"}];
max = WeatherData[stazione, "MaxTemperature", {{2009, 1, 1}, {2009, 12, 31}, "Day"}];
```

```
DateListPlot[{min, max}, Joined → True, Filling → {1 → {2}}]
```


40
30
20
10
0
Jan
Apr
Jul
Oct
Jan

Conclusioni

Mathematica è un ambiente molto potente sia per il calcolo sia per lo sviluppo di applicazioni. La sua sintassi è molto semplice m sicuramente prendere dimestichezza con tutte le sue funzionalità richiede un periodo iniziale di training.

L'importante è comprendere non solo le regole di sintassi e di utilizzo delle varie funzioni ma anche il concetto di linguaggio sin bolico e stile funzionale di programmazione.