

# Regole sintattiche

## Simboli

• Ogni sequenza di caratteri alfanumerici che inizi con un carattere alfabetico e non contenga spazi o caratteri speciali è interpretata come un unico simbolo. • Le lettere maiuscole sono considerate distinte dalle lettere minuscole. • Un simbolo può essere utilizzato in quanto tale per il calcolo simbolico o come nome per rappresentare un oggetto (numero, espressione, funzione). • Tutti i simboli che rappresentano funzioni di sistema iniziano con la lettera maiuscola.

?? nome per ottenere informazioni su un simbolo

## Numeri

• I numeri sono considerati come entità atomiche al pari dei simboli. • Vi sono quattro tipi di numeri: interi, razionali, “reali” e complessi. • La grandezza dei numeri interi è limitata solo dalle dimensioni della memoria. • I numeri razionali sono rappresentati nella forma  $n/m$ . • I numeri “reali” (floating point) sono rappresentati con il punto decimale o con la notazione scientifica. • I numeri complessi sono rappresentati nella forma  $x + y i$ . • I numeri sono internamente identificati con le intestazioni `Integer`, `Rational`, `Real` o `Complex` (non visibili, ma utilizzabili nei modelli).

## Operatori

• I caratteri speciali `+ - * / _ , ; : . ^ ' " ' % ! ? # & @ < > ~ \` sono utilizzati per indicare operatori di vario genere e con diverse sintassi. • L'operatore di moltiplicazione è `*`, esso può essere omesso lasciando uno spazio tra i fattori. • I due punti non indicano la divisione. • Alcuni operatori sono rappresentati da una sequenza due o più caratteri speciali contigui (senza spazi).

## Parentesi

( ) per indicare la precedenza nelle espressioni  
[ ] per racchiudere gli argomenti di una funzione/elementi di un'espressione  
{ } per racchiudere gli elementi di una lista (vettore/matrice)  
[[ ]] per gli indici che individuano un elemento in una lista (vettore/matrice)  
(\* \*) per racchiudere eventuali commenti da non considerare nel calcolo

## Punteggiatura

, virgola per separare gli elementi di una lista (tra par. graffe), gli argomenti di una funzione/elementi di un'espressione (tra par. quadre), gli indici (tra doppie par. quadre)  
; punto e virgola per separare e concatenare diverse espressioni

## Spaziatura

• Spazi consecutivi hanno lo stesso valore di un solo spazio. • Gli spazi prima e dopo gli operatori, le parentesi e la punteggiatura sono opzionali. • Un'espressione può anche essere scritta su più righe. • Ogni ritorno a capo viene automaticamente interpretato come uno spazio se la parte di espressione che lo precede risulta sintatticamente incompleta. Altrimenti si può ottenere lo stesso risultato terminando la riga con `\` (che indica la concatenazione con la riga successiva).

## Espressioni

• Un'espressione è una qualunque scrittura che rispetti le regole sintattiche descritte. • La valutazione di un'espressione e la visualizzazione del risultato si ottiene digitando `shift-return`. • Se si immettono più espressioni contemporaneamente (disponendole su diverse righe), la valutazione e la visualizzazione dei rispettivi risultati avviene in sequenza. • Se le espressioni sono concatenate con il punto e virgola, esse vengono valutate tutte in sequenza, ma si ottiene come risultato solo quello relativo all'ultima. • Il risultato di un calcolo può essere utilizzato per ulteriori calcoli:

% ultima espressione ottenuta come output  
%% penultima espressione ottenuta come output  
%n espressione ottenuta come output numero  $n$

# Funzioni matematiche

## Operazioni aritmetiche

<code>a + b + ...</code>	$a + b + \dots$	addizione
<code>a b ...</code>	$a b \dots$	moltiplicazione
<code>a - b</code>	$a - b$	sottrazione
<code>a / b</code>	$a/b$	divisione
<code>a ^ b</code>	$a^b$	potenza
<code>Sqrt[a]</code>	$\sqrt{a}$	radice quadrata
<code>Max[a,b,...]</code>	$\max\{a,b,\dots\}$	massimo
<code>Min[a,b,...]</code>	$\min\{a,b,\dots\}$	minimo
<code>Sum[ai, {i,1,n}]</code>	$\sum_{i=1}^n a_i$	sommatoria
<code>Product[ai, {i,1,n}]</code>	$\prod_{i=1}^n a_i$	produttoria

## Espressioni algebriche

<code>Expand[esp]</code>	espansione dei prodotti al livello più esterno
<code>ExpandAll[esp]</code>	espansione di tutti i prodotti
<code>PowerExpand[esp]</code>	espansione di potenze e logaritmi
<code>Numerator[esp]</code>	numeratore di un'espressione razionale
<code>Denominator[esp]</code>	denominatore di un'espressione razionale
<code>Together[esp]</code>	somma di espressioni espressioni razionali
<code>Simplify[esp]</code>	semplificazione

## Polinomi

<code>PolynomialQ[p, {x,y,...}]</code>	$p$ è un polinomio nelle variabili $x, y, \dots$ ?
<code>Exponent[p,x]</code>	grado del polinomio $p$ rispetto ad $x$
<code>Coefficient[p,t]</code>	coefficiente di $t$ (ad esempio $x^n$ ) in $p$
<code>Collect[p,x]</code>	raccoglimento secondo le potenze di $x$
<code>Factor[p]</code>	fattorizzazione a coefficienti interi
<code>InterpolatingPolynomial[{{x1,y1},...},x]</code>	polinomio $p$ nella variabile $x$ t.c. $p(x_1) = y_1, \dots$

## Proposizioni

<code>True</code>	V	costante booleana Vero
<code>False</code>	F	costante booleana Falso
<code>p &amp;&amp; q &amp;&amp; ...</code>	$p \wedge q \wedge \dots$	coniunzione (And)
<code>p    q    ...</code>	$p \vee q \vee \dots$	disgiunzione inclusiva (Or)
<code>Not[p]</code>	$\sim p$	negazione logica
<code>a === b</code>	$a = b$	identità tra espressioni
<code>a !== b</code>	$a \neq b$	non identità tra espressioni

## Relazioni aritmetiche

<code>a == b</code>	$a = b$	uguaglianza
<code>a != b</code>	$a \neq b$	disuguaglianza
<code>a &gt;= b</code>	$a \geq b$	maggiore o uguale
<code>a &lt;= b</code>	$a \leq b$	minore o uguale
<code>a &gt; b</code>	$a > b$	maggiore
<code>a &lt; b</code>	$a < b$	minore

## Equazioni

Reduce[{eq,...},{x,...}]

semplificazione di equazioni

Solve[{eq,...},{x,...}]

soluzione di equazioni

NSolve[{eq,...},{x,...},...}]

soluzione numerica di equazioni

FindRoot[{eq,...},{x,a},...}]

ricerca numerica di una soluzione

## Sostituzioni

esp /. reg

sostituzioni mediante una regola o lista di regole

esp //. reg

sostituzioni ripetute finché possibile

a -> b

regola di sostituzione con valutazione immediata

a :> b

regola di sostituzione con valutazione differita

## Numeri interi

IntegerQ[n]

$n$  è un intero?

PrimeQ[n]

$n$  è un numero primo?

Prime[n]

$n$ -esimo numero primo

FactorInteger[n]

scomposizione in fattori primi

Quotient[n,m]

quoziente intero

Mod[n,m]

resto del quoziente intero

GCD[n,m,...]

$(n, m, \dots)$

massimo comune divisore

LCM[n,m,...]

$[n, m, \dots]$

minimo comune multiplo

## Numeri "reali"

NumberQ[x]

$x$  è un numero?

N[x]

valore numerico di  $x$  con 6 cifre significative

N[x,n]

valore numerico di  $x$  con  $n$  cifre significative

Sign[x]

segno di  $x$  ( $\pm 1$  se  $x$  è pos/neg, 0 se  $x$  è nullo)

Abs[x]

$|x|$

valore assoluto

Floor[x]

$\lfloor x \rfloor$

massimo intero minore o uguale ad  $x$

Round[x]

intero più vicino ad  $x$

Rationalize[x]

conversione di  $x$  in  $n/m$  se abbastanza vicino

Random[tipo,{a,b}]

numero pseudocasuale intero o reale tra  $a$  e  $b$

## Costanti numeriche

E

$e$

numero di Nepero

I

$i$

unità immaginaria

Pi

$\pi$

pi greco

Degree

$180/\pi$

grado espresso in radianti

Infinity

$\infty$

infinito positivo

## Funzioni elementari

Exp[x]

$e^x$

esponenziale

Log[x]

$\log x$

logaritmo naturale

Log[a,x]

$\log_a x$

logaritmo in base  $a$

Cos[x]

$\cos x$

coseno

Sin[x]

$\sin x$

seno

Tan[x]

$\operatorname{tg} x$

tangente

ArcCos[x]

$\arccos x$

arcocoseno

ArcSin[x]

$\arcsin x$

arcoseno

ArcTan[x]

$\operatorname{arctg} x$

arcotangente

## Calcolo differenziale

Limit[s, x -> a]	$\lim_{x \rightarrow a} s$	limite
f'[x]	$f'(x)$	funzione derivata
D[s, x]	$\partial s / \partial x$	derivazione parziale
Dt[s, x]	$ds/dx$	derivazione totale
Series[f, {x, a, n}]		espansione in serie di potenze
Normal[s]		eliminazione degli infinitesimi
Integrate[s, x]	$\int s dx$	integrazione indefinita
Integrate[s, {x, a, b}]	$\int_a^b s dx$	integrazione definita
NIntegrate[s, {x, a, b}]	$\int_a^b s dx$	integrazione numerica
DSolve[{eq, ...}, y, {x, ...}]		soluzione di equazioni differenziali
NDSolve[{eq, ...}, y, {{x, a, b}, ...}]		soluzione numerica di equazioni differenziali
Interpolation[{{x1, y1}, {x2, y2}, ...}]		funzione interpolante $f$ (polinomiale a tratti) tale che $f(x_1) = y_1, f(x_2) = y_2, \dots$
Fit[{{x1, y1}, {x2, y2}, ...}, {s, ...}, x]		combinazione lineare delle espressioni $s, \dots$ nella variabile $x$ che meglio approssima i punti $(x_1, y_1), (x_2, y_2), \dots$ nel senso dei minimi quadrati

## Vettori e matrici

VectorQ[v]		$v$ è un vettore?
MatrixQ[A]		$A$ è una matrice?
Dimensions[A]		dimensioni di una matrice
v[[i]]	$v_i$	$i$ -esima componente del vettore $v$
A[[i, j]]	$a_{i,j}$	elemento di posto $i, j$ nella matrice $A$
v . w	$\langle v, w \rangle$	prodotto scalare se $v$ e $w$ sono vettori
A . B	$AB$	prodotto matriciale se $A$ e $B$ sono matrici
IdentityMatrix[n]	$I_n$	matrice identità di ordine $n$
Inverse[A]	$A^{-1}$	matrice inversa
Transpose[A]	${}^t A$	matrice trasposta

## Forme speciali

HoldForm[s]	visualizzazione di un'espressione non valutata
FullForm[s]	visualizzazione della rappresentazione interna
MatrixForm[m]	visualizzazione di una matrice in righe e colonne

## Grafica

Plot[y, {x, a, b}]	grafico di funzioni un una variabile
ListPlot[{{x, y}, ...}]	grafico di una funzione per punti
ParametricPlot[{x, y}, {t, a, b}]	grafico di una curva parametrizzata
Plot3D[z, {x, a, b}, {y, c, d}]	grafico di funzioni in due variabili
ParametricPlot3D[{x, y, z}, {t, a, b}]	grafico di una curva parametrizzata
ParametricPlot3D[{x, y, z}, {t, a, b}, {s, c, d}]	grafico di una superficie parametrizzata
Show[{g, ...}, opzioni]	visualizzazione di grafici
Graphics[{primitive}]	grafico bidimensionale
Graphics3D[{primitive}]	grafico tridimensionale
Point[{x, y}]	punto $(x, y)$
Line[{{x1, y1}, {x2, y2}, ...}]	spezzata di vertici $(x_1, y_1), (x_2, y_2), \dots$
Polygon[{{x1, y1}, {x2, y2}, {x3, y3}, ...}]	poligono di vertici $(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots$

# Funzioni di programmazione

## Assegnazioni

<code>Clear[x]</code>	eliminazione di tutte le assegnazioni relative ad $x$
<code>x = esp</code>	assegnazione con valutazione immediata
<code>x := esp</code>	assegnazione con valutazione differita
<code>f[x,...] = esp</code>	assegnazione funzionale con valutazione immediata
<code>f[x,...] := esp</code>	assegnazione funzionale con valutazione differita

## Modelli

<code>x_</code>	qualsunque espressione
<code>x_tipo</code>	qualsunque espressione del tipo indicato
<code>x__</code>	qualsunque sequenza di una o più espressioni
<code>x__tipo</code>	qualsunque sequenza di una o più espressioni del tipo indicato
<code>x___</code>	qualsunque sequenza di zero, una o più espressioni
<code>x___tipo</code>	qualsunque sequenza di zero, una o più espressioni del tipo indicato
<code>mod /; cond</code>	modello con condizione
<code>mod ? test</code>	modello con test funzionale

## Espressioni

<code>Head[esp]</code>	intestazione / tipo di un'espressione
<code>First[esp]</code>	primo elemento di un'espressione
<code>Rest[esp]</code>	lista degli elementi successivi al primo
<code>Last[esp]</code>	ultimo elemento di un'espressione
<code>Part[esp, i]</code>	$i$ -esimo elemento di un'espressione
<code>Length[esp]</code>	lunghezza (numero di elementi) di un'espressione
<code>Depth[esp]</code>	profondità (numero di livelli) di un'espressione
<code>LeafCount[esp]</code>	numero degli atomi presenti in una espressione
<code>AtomQ[esp]</code>	l'espressione è un atomo (simbolo o numero)?
<code>FreeQ[esp, x]</code>	l'espressione non contiene $x$ come sottoespressione?
<code>MemberQ[esp, x]</code>	l'espressione contiene $x$ come elemento?
<code>MatchQ[esp, mod]</code>	l'espressione corrisponde al modello?

## Liste

<code>ListQ[l]</code>	$l$ è una lista?
<code>List[x, y, ...]</code>	lista costituita dagli elementi $x, y, \dots$
<code>Table[s, {i, 1, n}]</code>	lista dei valori di $s$ al variare di $i$ da 1 ad $n$
<code>Join[l1, l2, ...]</code>	concatenazione di liste
<code>Flatten[l]</code>	appiattimento di una lista di liste
<code>Sort[l]</code>	ordinamento degli elementi di una lista
<code>Position[l, x]</code>	lista delle posizioni in cui $x$ occorre nella lista $l$
<code>Count[l, x]</code>	numero delle occorrenze di $x$ nella lista $l$
<code>Prepend[l, x]</code>	lista $l$ con l'aggiunta di $x$ come primo elemento
<code>Append[l, x]</code>	lista $l$ con l'aggiunta di $x$ come ultimo elemento
<code>Insert[l, x, i]</code>	lista $l$ con l'inserimento di $x$ come $i$ -esimo elemento
<code>Select[l, test]</code>	sottolista degli elementi che soddisfano test funzionale
<code>Cases[l, mod]</code>	sottolista degli elementi che corrispondono ad un modello
<code>Complement[l1, l2]</code>	lista differenza (a meno dell'ordine e senza ripetizioni)
<code>Intersection[l1, l2, ...]</code>	intersezione di liste (a meno dell'ordine e senza ripetizioni)
<code>Union[l1, l2, ...]</code>	unione di liste (a meno dell'ordine e senza ripetizioni)

## Funzioni

<code>Identity</code>	funzione identità
<code>Function[{x,...},esp]</code>	funzione definita da un'espressione con variabili $x, \dots$
<code>Compile[{x,...},s]</code>	funzione compilata definita da un'espressione con variabili $x, \dots$
<code>Composition[f,g,...]</code>	composizione di funzioni
<code>Nest[f,x,n]</code>	valore ottenuto dopo l' $n$ -esima iterazione di $f$ su $x$
<code>NestList[f,x,n]</code>	lista dei valori ottenuti dalle prime $n$ iterazioni di $f$ su $x$
<code>Apply[f,l]</code>	applicazione di $f$ alla sequenza degli elementi della lista $l$
<code>Map[f,l]</code>	applicazione di $f$ a ciascun elemento della lista $l$

## Strutture

<code>esp1;esp2;...</code>	concatenazione di espressioni
<code>Block[{x=x0,...},esp]</code>	valutazione di un'espressione con valori locali delle variabili $x, \dots$
<code>Do[esp,{i,1,n}]</code>	valutazione di un'espressione ripetuta al variare di $i$ da 1 ad $n$
<code>While[cond,esp]</code>	ciclo <i>While</i> (valutazione ripetuta finché la condizione è soddisfatta)
<code>If[cond,esp1,esp2]</code>	struttura <i>If ... Then ... else ...</i> (se la condizione è soddisfatta viene valutata la prima espressione, altrimenti la seconda)

## Valutazione

<code>Evaluate[esp]</code>	valutazione forzata di un'espressione
<code>Hold[esp]</code>	<code>Hold</code> impedisce la valutazione di un'espressione
<code>ReleaseHold[esp]</code>	eliminazione di <code>Hold</code> e <code>HoldForm</code> (e conseguente valutazione)