



Le liste sono oggetti matematici composte da elementi, come gli insiemi. Ad esempio, la lista che ha come elementi $2, a, b, 7$ si scrive $\{2, a, b, 7\}$

Una lista è dotata di una struttura d'ordine determinata dalla **posizione** degli elementi

Range[a,b] dà la sequenza degli interi compresi tra a e b

Dati una lista A , un'espressione e , una variabile x , la lista **Table[e,{x,A}]** elenca i valori che assume e mentre x percorre A

Table[e, {x,Range[a,b]}] si può abbreviare con **Table[e, {x,a,b}]**

Usare un simbolo di costante è un modo per abbreviare espressioni.

Un simbolo A viene definito da un'espressione e inviando in input la regola $A:=e$

non sarà prodotto un output in risposta a questo input.

Divisors[n] dà la lista dei divisori naturali di un numero intero n

Sum[x,{x,A}] dà la somma degli elementi di una lista A

Le liste

Esempio 1. L'espressione $\{a\}$ è una lista che ha a come unico elemento; $\{\{a\}\}$ è una lista, che ha come unico elemento la lista $\{a\}$.
In una lista ogni elemento occupa una *posizione*. La lista $\{7, \{x, 3\}\}$ ha due elementi: 7 (nella posizione 1) e l'elemento $\{x, 3\}$ (nella posizione 2).

Definizione. Una lista è un'espressione della forma $\{a_1, a_2, \dots, a_n\}$ in cui a_1, a_2, \dots, a_n sono espressioni e n è un numero naturale.

Il numero n nella definizione 1 è la *lunghezza* della lista e conta il numero delle posizioni della lista. La lista è detta *vuota* quando $n=0$ e si scrive $\{\}$.

Una differenza fondamentale tra liste e insiemi è l'*ordine degli elementi*:

$\{1, 3, 5, 7\}, \{1, 5, 3, 7\}, \{7, 5, 3, 1\}, \{3, 5, 1, 7\}, \{1, 7, 3, 5\}$

sono liste distinte tra loro, pur determinando lo stesso insieme.

Un'altra differenza tra liste e insiemi è la *ripetizione degli elementi*:

$\{1, 1, 5\}, \{1, 5, 5, 5\}, \{1, 1, 5\}, \{1, 5\}, \{1, 1, 1, 5\}$

sono liste distinte tra loro, pur determinando lo stesso insieme.

Esempio 2. Scrivere la lista dei quadrati dei numeri interi compresi tra 15 e 27 .
Costruiamo la lista dei numeri interi compresi tra 15 e 27 :

In `Range[15,27]`

Out `{15,16,17,18,19,20,21,22,23,24,25,26,27}`

e concludiamo con la lista di x^2 con x che percorre `Range[15,27]`

In `Table[x^2, {x, Range[15,27]}]`

Out `{225,256,289,324,361,400,441,484,529,576,625,676,729}`

Esercizio 1. Costruire la lista dei cubi dei numeri interi compresi tra 10 e 100 .

Esercizio 2. Costruire la lista di coppie di interi $\{x, 2x+1\}$, x da 11 a 17 :

`{{11,23},{12,25},{13,27},{14,29},{15,31},{16,32},{17,35}}`

Definizione di costanti

Esempio 3. Rivisitiamo l'esempio 2 scrivendo una definizione per la lista dei numeri interi compresi tra 15 e 27 :

In `A := Range[15,27]`

calcoliamo la lista dei quadrati degli elementi di A :

In `Table[x^2, {x,A}]`

Out `{225,256,289,324,361,400,441,484,529,576,625,676,729}`

Esercizio 3. Calcolare la lista dei numeri naturali che sono potenze quarte dei divisori di 960 .

Somma degli elementi di una lista

Esercizio 4. Calcolare la somma dei divisori di 216



L' *unione* di due liste **A**, **B** si scrive **A∪B** è la lista degli elementi non ripetuti che occorrono in almeno una delle liste.

L'*intersezione* di due liste **A**, **B** si scrive **A∩B** è la lista degli elementi non ripetuti che compaiono in entrambe le liste.

Date due liste **A**, **B**, la lista *complementare* di **B** in **A** si scrive **Complement[A,B]** ed è la lista degli elementi non ripetuti che stanno in **A** ma non in **B**.

Max[A] dà il valore massimo degli elementi di una lista **A**.

Un *predicato* è una funzione che ha due soli risultati, detti *valori di verità*, **True** e **False**

Il simbolo di *uguaglianza* è **==**

Length[L] dà il numero degli elementi di una lista **L**

OddQ[n] significa "n è dispari"

Dati una lista **L** e un predicato **p**, **Select[L, p]** è la lista degli elementi di **L** che soddisfano **p**

A∧B è la congiunzione logica di due formule **A**, **B**, con il significato di "A e B"

A∨B è la disgiunzione logica di due formule **A**, **B**, con il significato di "A o B"

Unione e intersezione di liste. Liste complementari.

Esempio 4. La lista dei numeri interi compresi tra 1 e 100 che sono multipli di 2 oppure di 5 è data da

In `Table[2x, {x, 1, 50}] ∪ Table[5x, {x, 1, 20}]`

Esercizio 5. Costruire la lista dei numeri interi compresi tra 1 e 10000 che sono multipli di 20 oppure quadrati perfetti.

Esercizio 6. Costruire la lista dei numeri interi che sono divisori di 10^{10} e multipli di 2^9 .

Esercizio 7. Quanti sono i divisori di 10^9 che non sono divisori di 10^8 ?

Definizione di simboli di funzione

Esempio 5. L' espressione `Sum[x, {x, Divisors[n]}]` calcola la somma dei divisori naturali di un numero intero **n**. Abbreviamo questa espressione con

In `σ[n_] := Sum[x, {x, Divisors[n]}]`

σ diviene il simbolo di una funzione che può essere applicata a numeri interi:

In `σ[6]`

Out 12

Esercizio 8. Qual è il valore massimo della somma dei divisori di un numero intero **n** compreso tra 1 e 100?

Esempio 6. L' espressione `PrimeQ[n]` significa "n è un numero primo". Il simbolo `PrimeQ` è un predicato di *Mathematica*. Definiamo un predicato **perfetto** tale che `perfetto[n]` significhi "la somma dei divisori di n è il doppio di n"

In `perfetto[n_] := Sum[n, {n, Divisors[n]}] == 2n`

Verifichiamo che 6 è un numero perfetto, mentre 36 non lo è

In `{perfetto[6], perfetto[36]}`

Out {True, False}

Esercizio 9. Definire un predicato **p** tale che `p[n]` significhi "il numero dei divisori di n è un terzo di n". Verificare che ci sono tre numeri interi compresi tra 1 e 25 che soddisfano (cioè, rendono vero) il predicato **p**.

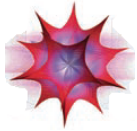
Esercizio 10. Definire un predicato **p** tale che `p[n]` significhi "il numero dei divisori di n è dispari". Determinare la lista dei numeri interi compresi tra 1 e 100 che hanno un numero dispari di divisori.

Esercizio 11. Definire un predicato **q** tale che `q[n]` significhi "n non è divisore di 4800".

Definire un predicato **r** che dica che un numero è dispari e divisore di 4800. Determinare la lista dei numeri interi compresi tra 1 e 100 che sono dispari e divisori di 4800.

Esercizio 12. Definire un predicato **t** tale che `t[n]` significhi "il numero dei divisori di n non è divisore di 4800".

Determinare la lista dei numeri interi compresi tra 1 e 120 che soddisfano il predicato **t**.



Esperimenti aleatori: spazio dei campioni e frequenza di un evento.

Esempio 8. Definiamo lo spazio dei campioni del *lancio di un dado*:

```
In Ω := { [1], [2], [3], [4], [5], [6] }
```

E' possibile simulare il lancio di un dado:

```
In RandomChoice[Ω]
```

```
Out [1]
```

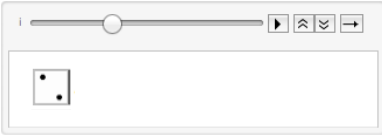
Simuliamo una sequenza di 13 prove dell' *lancio di un dado*:

```
In sequenza = Table[RandomChoice[Ω], {x, 1, 13}]
```

```
Out { [1], [5], [2], [5], [3], [1], [1], [5], [6], [6], [1], [2], [5], [5] }
```

```
In ListAnimate[sequenza]
```

```
Out
```



Calcoliamo la frequenza assoluta dell' *esito* [1] nella sequenza di prove:

```
In Count[sequenza, [1]]
```

```
Out 3
```

Calcoliamo le frequenze assolute di tutti i risultati possibili:

```
In Table[{x, Count[sequenza, x]}, {x, Ω}]
```

```
Out { { [1], 2 }, { [2], 3 }, { [3], 2 }, { [4], 1 }, { [5], 4 }, { [6], 1 } }
```

Esercizio 13. Definire lo spazio dei campioni del *lancio di una moneta*. Simulare una sequenza di 300 *lanci di una moneta* e calcolare le frequenze assolute dei risultati ottenuti.

Esempio 9. Ridefiniamo lo spazio dei campioni del *lancio di un dado*:

```
In Ω := Range[1, 6]
```

Un *evento* è un sottoinsieme di Ω e spesso viene determinato da una funzione definita su Ω , a valori **True**, **False**. L' *evento esce un numero maggiore di 3* è determinato dal predicato x definito da

```
In X[ω_] := ω > 3
```

In una sequenza di 13 prove del lancio di un dado:

```
In esiti = Table[RandomChoice[Ω], {x, 1, 13}]
```

definiamo la lista degli esiti che hanno realizzato (favorevoli a) *esce un numero maggiore di 3*:

```
In fp = Select[esiti, p]
```

calcoliamo la frequenza assoluta dell' *evento* nell' *esperimento*:

```
In Length[fp]
```

Esercizio 14. Definire lo spazio dei campioni dell' *estrazione di una pallina da un' urna contenente 90 palline numerate da 1 a 90*. Simulare una sequenza di 200 estrazioni e osservare la frequenza dell' *evento* *è un numero pari*.

RandomChoice[A]
estrae un elemento a caso da una lista **A**

La definizione **A=e** associa ad un simbolo **A** il valore che l' *espressione e* ha nel momento in cui la definizione viene inviata in input.

L' *output* di **ListAnimate[L]** è un' *animazione* degli elementi della lista **L**

Count[L,x] dà la *frequenza assoluta* con cui **x** compare come elemento di una lista **L**

Con la regola che la definisce, l' *espressione* **X[ω]** sarà tradotta in $\omega > 3$ e poi valutata in **True** oppure **False**

Dati una lista **L** e un predicato **p**, la lista degli elementi di **L** che soddisfano **p** è data da **Select[L, p]**

Il predicato **EvenQ** è definito sui numeri interi: **EvenQ[n]** significa "n è pari"



`Length[Ω]` è il numero di elementi dello spazio Ω

Probabilità su spazi finiti.

Esempio 11. Calcolare la probabilità che escano 6 oppure 7 teste lanciando 10 monete. Definiamo lo spazio dei campioni dell' esperimento:

```
In U := {t,c}; Ω := Tuples[U,10]; k := Length[Ω]
```

Definiamo la probabilità uniforme sugli eventi elementari:

```
In P[ω_] := 1/Length[Ω] /; MemberQ[Ω,ω];
```

Definiamo la probabilità su un generico evento:

```
In P[A_] := Sum[P[ω],{ω,A}];
```

Definiamo la variabile aleatoria x che conta il numero di teste in un esito ω

```
In x[ω_] := Count[ω,t]
```

Definiamo il predicato caratteristico dell' evento "escono 6 oppure 7 teste":

```
In h[ω_] := 6 ≤ x[ω] ≤ 7
```

Valutiamo la probabilità dell' evento determinato da h :

```
In P[Select[Ω,h]]
```

Esercizio 15. Un'urna contiene 2 palline bianche e 3 rosse. Qual è la probabilità che escano 6 palline bianche in 10 estrazioni con restituzione?

Prodotto cartesiano di liste ed esperimenti aleatori.

Esempio 12. Definire lo spazio dei campioni dell' estrazione con restituzione di due carte da un mazzo di 40. Definire l' evento: la prima carta è un asso e la seconda non è di cuori.

Rappresentiamo una carta come una coppia della forma {seme, valore}

```
In semi := {cuori, fiori, picche, quadri}
```

```
In valori := {A,2,3,4,5,6,7,J,Q,K}
```

```
In U = Tuples[{semi,valori}]
```

Definiamo le funzioni **seme** e **valore**, sull'insieme U delle carte:

```
In seme[{s_,v_}] := s; valore[{s_,v_}] := v;
```

Definiamo lo spazio dei campioni dell' estrazione di due carte con restituzione:

```
In Ω := Tuples[U,2]
```

Definiamo il predicato "la 1° carta estratta è un asso" e "la 2° non è di cuori":

```
In p[{x_,y_}] := (valore[x]==A) ∧ (valore[y]≠cuori)
```

Otteniamo l' evento caratteristico del predicato p :

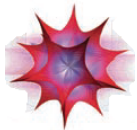
```
In Select[Ω,p]
```

La coppia {fiori, A} rappresenta l'asso di fiori;
la coppia {picche, J} sarà il fante di picche

`Tuples[{X, Y}]` fornisce la lista *prodotto cartesiano* di due liste X e Y

`Tuples [U, n]` è la lista *potenza cartesiana* $U \times U \times \dots \times U$ delle n -sequenze di elementi di una lista U

Il simbolo \wedge è il connettivo logico **And**, di *congiunzione*



Espressioni ed elementi grafici, primitive grafiche.

Ogni figura che si può disegnare è descrivibile con un' *espressione grafica*.

Per descrivere gli elementi grafici di una figura si usa un sistema di coordinate

Un punto è rappresentato da una coppia di numeri, le sue coordinate in un riferimento cartesiano.

Le *primitive grafiche* sono gli *elementi grafici* di base per comporre le *espressioni grafiche*.

Line[s] è la primitiva grafica che rappresenta la linea poligonale passante per i punti di una lista **s**

Circle[a, r] è la primitiva grafica che rappresenta il circolo di centro **a** e raggio **r**

Gli elementi grafici sono di due tipi: *primitive grafiche* come linee, poligoni, cerchi e *direttive grafiche* come colore, spessore delle linee, grandezza dei punti, trasparenza.

Red, Blue, Green, Yellow sono *direttive grafiche* che rappresentano colori.

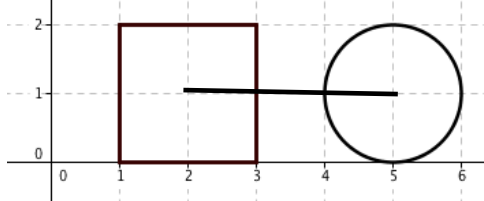
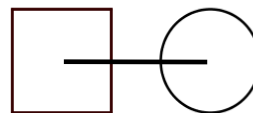
Disk[a, r] è la primitiva grafica che rappresenta il disco di centro **a** e raggio **r**

Thickness[0.01] è la direttiva grafica usata in figura (2) per avere maggior spessore nel quadrato **b**

Le *espressioni grafiche* hanno in generale la forma:

Graphics [lista di elementi grafici]

Esempio 13. Disegnare la figura composta da un quadrato, un cerchio e il segmento con estremi i loro centri.



Il cerchio di centro **{5, 1}** e raggio **1**, è dato dall' espressione

Circle[**{5, 1}**, **1**]

Il segmento con estremi i punti **{2, 1}** e **{5, 1}** è rappresentato da

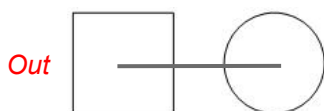
Line[**{{2, 1}, {5, 1}}**]

Il quadrato è rappresentato da

Line[**{{1, 0}, {3, 0}, {3, 2}, {1, 2}, {1, 0}}**]

La figura si ottiene applicando **Graphics** alla lista dei tre elementi grafici:

In **Graphics** [**{Circle**[**{5, 1}**, **1**], **Line**[**{{2, 1}, {5, 1}}**], **Line**[**{{1, 0}, {3, 0}, {3, 2}, {1, 2}, {1, 0}}**]



Esempio 14. Ridefiniamo le tre primitive grafiche dell' esempio 11 con i colori:

In **q** := **{Green, Line**[**{{1, 0}, {3, 0}, {3, 2}, {1, 2}, {1, 0}}**]

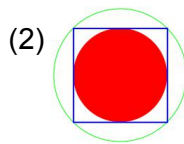
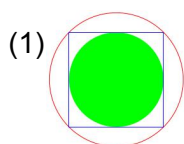
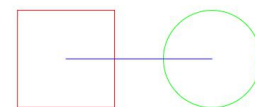
c := **{Red, Circle**[**{5, 1}**, **1**]

d := **{Red, Line**[**{{2, 1}, {5, 1}}**]

In **Graphics** [**{Green, q, Red, c, d, Blue}**]

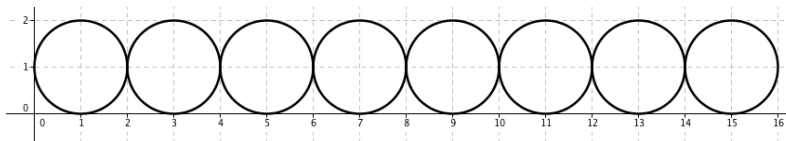


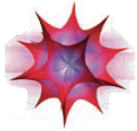
Esercizio 16. Ridisegnare la figura dell' esempio 12 modificando i colori.



Esercizio 17. Disegnare un disco **a** inscritto nel quadrato **b** ed un cerchio **c** circoscritto al quadrato **b**, con due diverse colorazioni (1) e (2) come in figura.

Esercizio 18. Disegnare 8 cerchi di raggio **1** e centri **{1,1}, {3,1}, ..., {15,1}**





Statistica descrittiva: caratteri, frequenze e classi

Esempio 15. Importiamo le quotazioni del titolo Apple NASDAQ dal 1 gennaio 2013 alla data attuale, registrandole su una lista U :

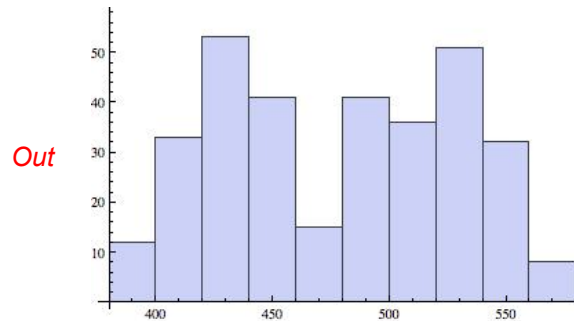
```
In U = FinancialData["NASDAQ:AAPL", "2013"]
```

Definiamo il carattere **prezzo** su U :

```
In prezzo[{data_, x_}] := x
```

Vediamo un istogramma del carattere **prezzo** su U :

```
In Histogram[Table[prezzo[u], {u, U}]]
```



Il primo dato della lista è $\{2013, 1, 2\}, 532.39\}$

Ogni dato è della forma $\{data, prezzo\}$.

Le *modalità* del carattere **prezzo** sono i valori **prezzo[u]** al variare di $u \in U$

La suddivisione delle modalità viene decisa dall'istogramma in modo automatico: in questo esempio è una partizione di intervalli di ampiezza 10.

Floor $[x, d]$ è il più grande multiplo di d inferiore o uguale a x

$c[u] = q$ significa che **prezzo[u]** è compreso nell'intervallo da q a $q+10$

Due dati u, v di U tali che $c[u] = c[v]$ sono nella stessa classe della partizione definita da c su U

Histogram $[A, \{d\}]$ è un istogramma della lista di dati A , con ampiezza delle classi d

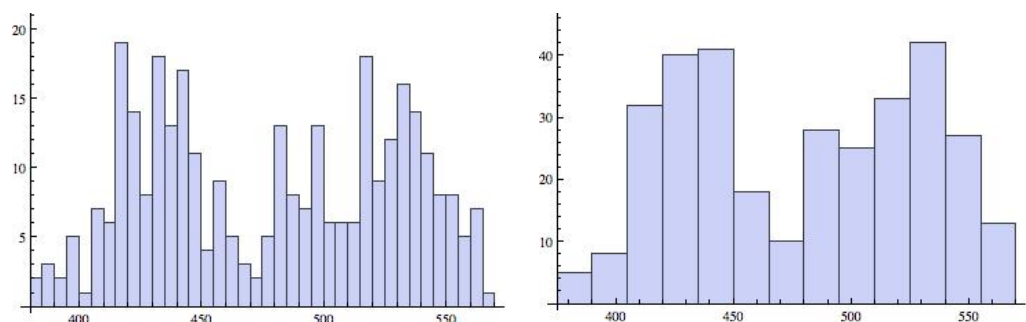
Esercizio 19. Si consideri un nuovo carattere c sull'universo U dell'esempio 15 definito da $c[u] := \text{Floor}[\text{prezzo}[u], 10]$

Verificare che le modalità di c formano una progressione aritmetica di ragione 10 e che l'istogramma di c è lo stesso del carattere **prezzo**.

La *frequenza assoluta (relativa)* di una modalità m del carattere c è data dalla frequenza assoluta (relativa) di m nella lista $\{c[u]: u \in U\}$ delle modalità di c .

Esercizio 20. Calcolare le frequenze assolute e le frequenze relative del carattere c sull'universo U dell'esercizio 27.

Esercizio 21. Si consideri la lista dei dati: $A = \text{Table}[\text{prezzo}[u], \{u, U\}]$, dell'esercizio 15. Fare due nuovi istogrammi della lista A in cui l'ampiezza delle classi sia rispettivamente 5 e 15.



Esercizio 22. Leggiamo le temperature medie giornaliere osservate nella città di Venezia dal 1 gennaio 2012 al 31 dicembre 2013:

```
In U = WeatherData["Venezia", "Temperature",
  {{2012, 1, 1}, {2013, 12, 31}, "Day"}]
```

Definire il carattere **temperatura** su U , costruire la lista delle temperature medie giornaliere osservate e farne un istogramma. Fare quindi un istogramma in cui l'ampiezza delle classi sia 0.5 gradi.