



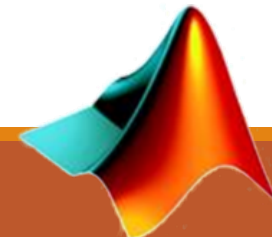
UNIVERSITÀ DEGLI STUDI DI SALERNO

Fondamenti di Informatica

Introduzione a MATLAB

Prof. Arcangelo Castiglione

A.A. 2016/17



MATLAB®

Introduzione a MATLAB: OUTLINE

- Caratteristiche
- Come si presenta
- Inserimento dei Comandi
- Variabili
- Operatori e Precedenza degli Operatori
- Workspace e Sessione di Lavoro
- Funzionalità di Help



MATLAB: Di che si tratta?

- MATLAB è una piattaforma per
 - **Calcolo scientifico**
 - **Elaborazione** dei dati
 - **Visualizzazione** dei dati
- MATLAB è sviluppato dall'azienda privata MathWorks
 - www.mathworks.it
- **MATLAB** → **MAT**rix **LAB**oratory



MATLAB: Caratteristiche

- Linguaggio **basato su espressioni** che rende semplice la programmazione
- **La visualizzazione dei dati processati è immediata e può avvenire anche mediante grafici**
 - 2-D e 3-D
 - Di tantissime tipologie
 - **Stampabili o memorizzabili**
 - Ad es. per essere inclusi in documenti Word, LaTeX, etc

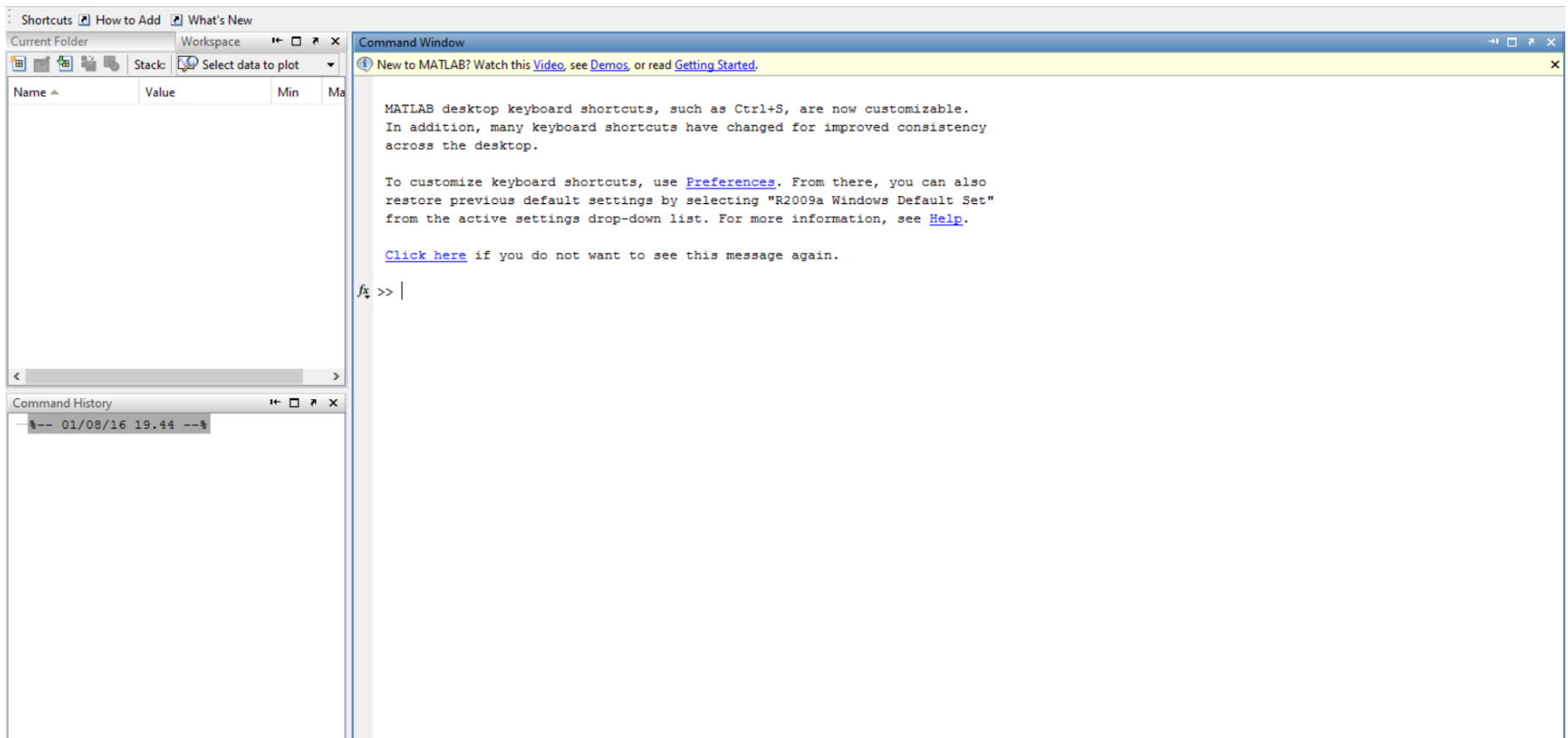


MATLAB: Ambiti di Utilizzo

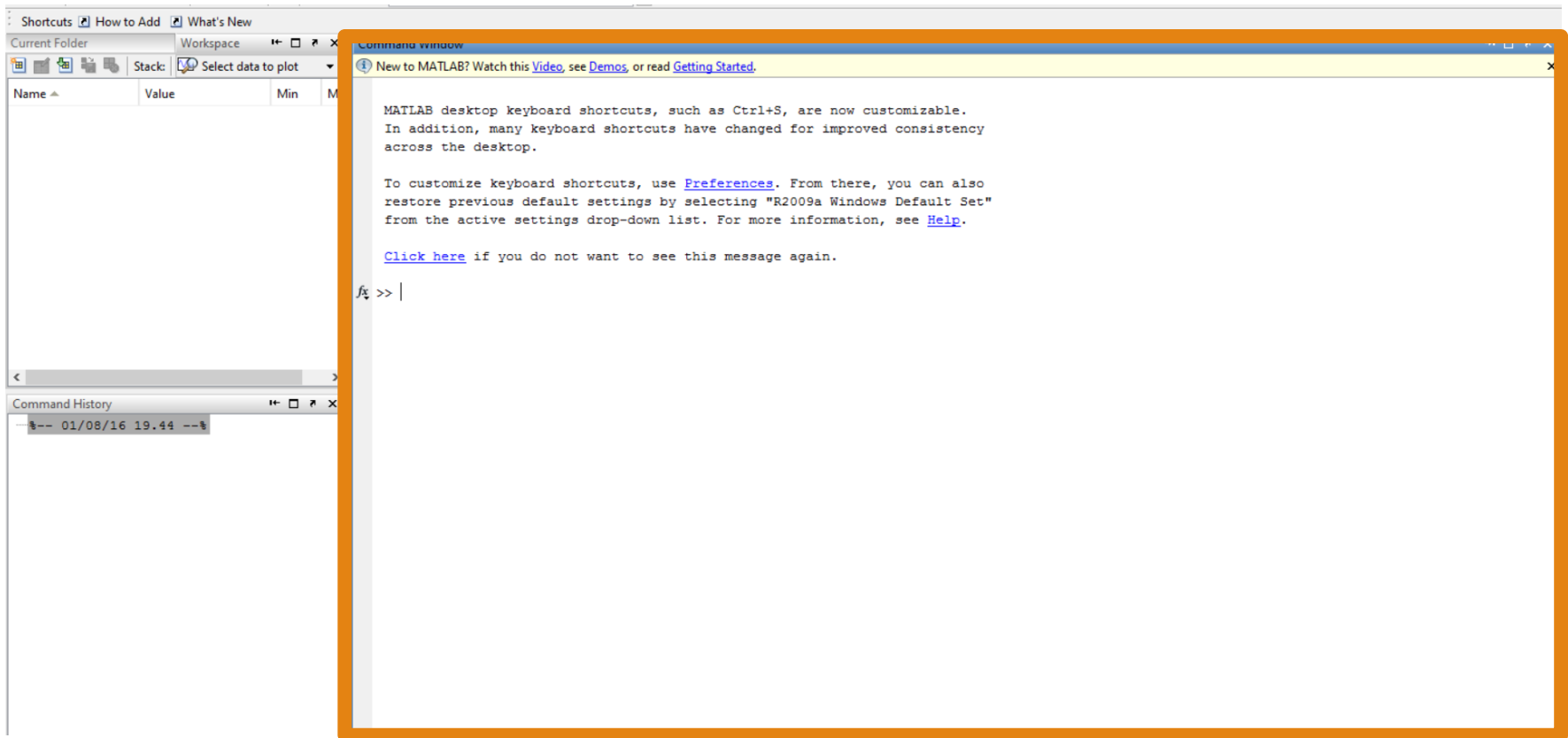
- MATLAB è
 - Usato nella **ricerca scientifica** e nella risoluzione di **problemi di ingegneria**
 - Estremamente duttile, oltre che facile da apprendere ed usare
 - Disponibile per tutti i sistemi operativi
 - Windows, OS X, Linux, etc
- MATLAB
 - Fornisce funzioni e strumenti specifici, ben consolidati, per affrontare problemi di tipo scientifico, ingegneristico, economico, etc
 - Permette di svolgere calcoli, anche estremamente complessi, in maniera corretta ed efficiente
 - A patto che i calcoli vengano impostati in maniera opportuna



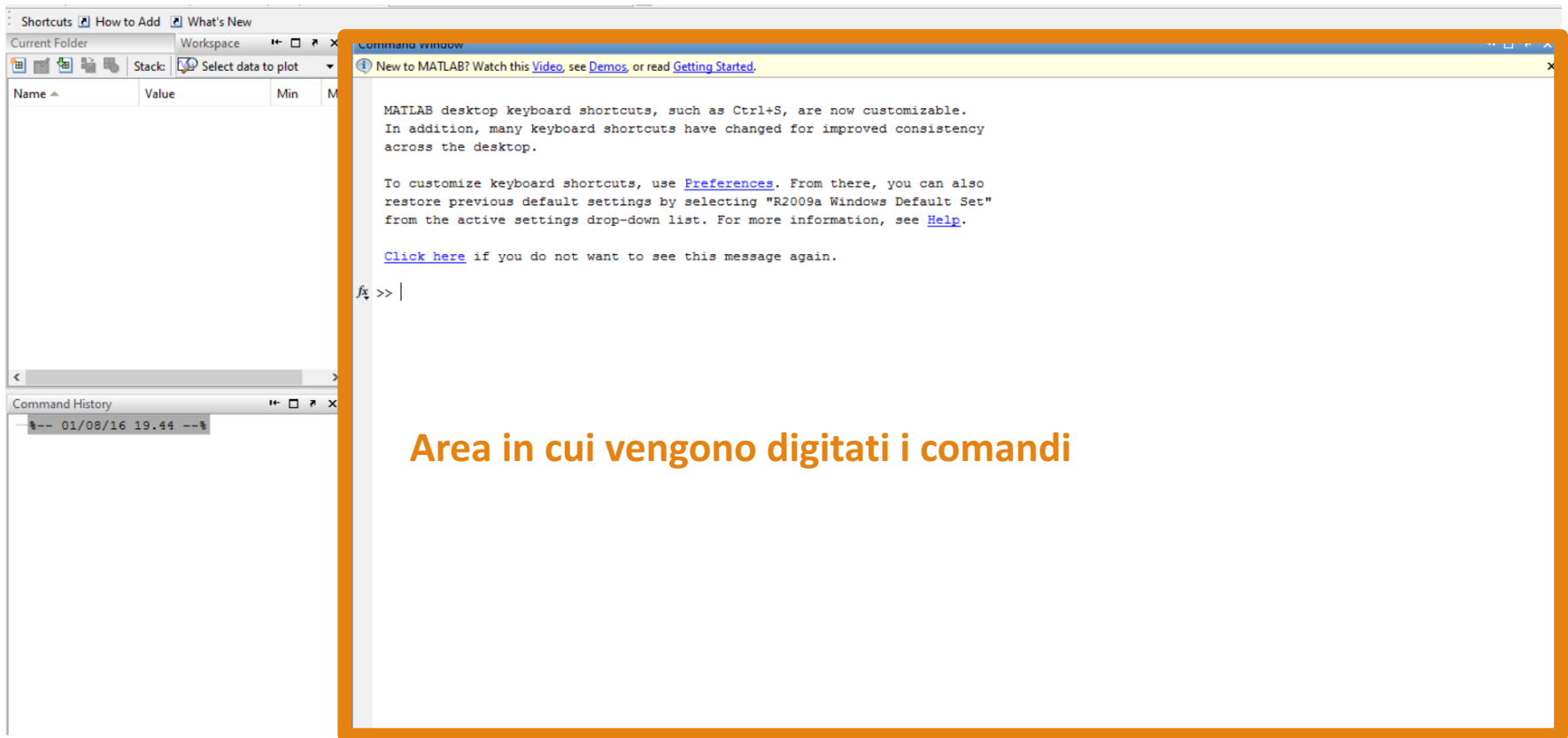
MATLAB: Come si Presenta



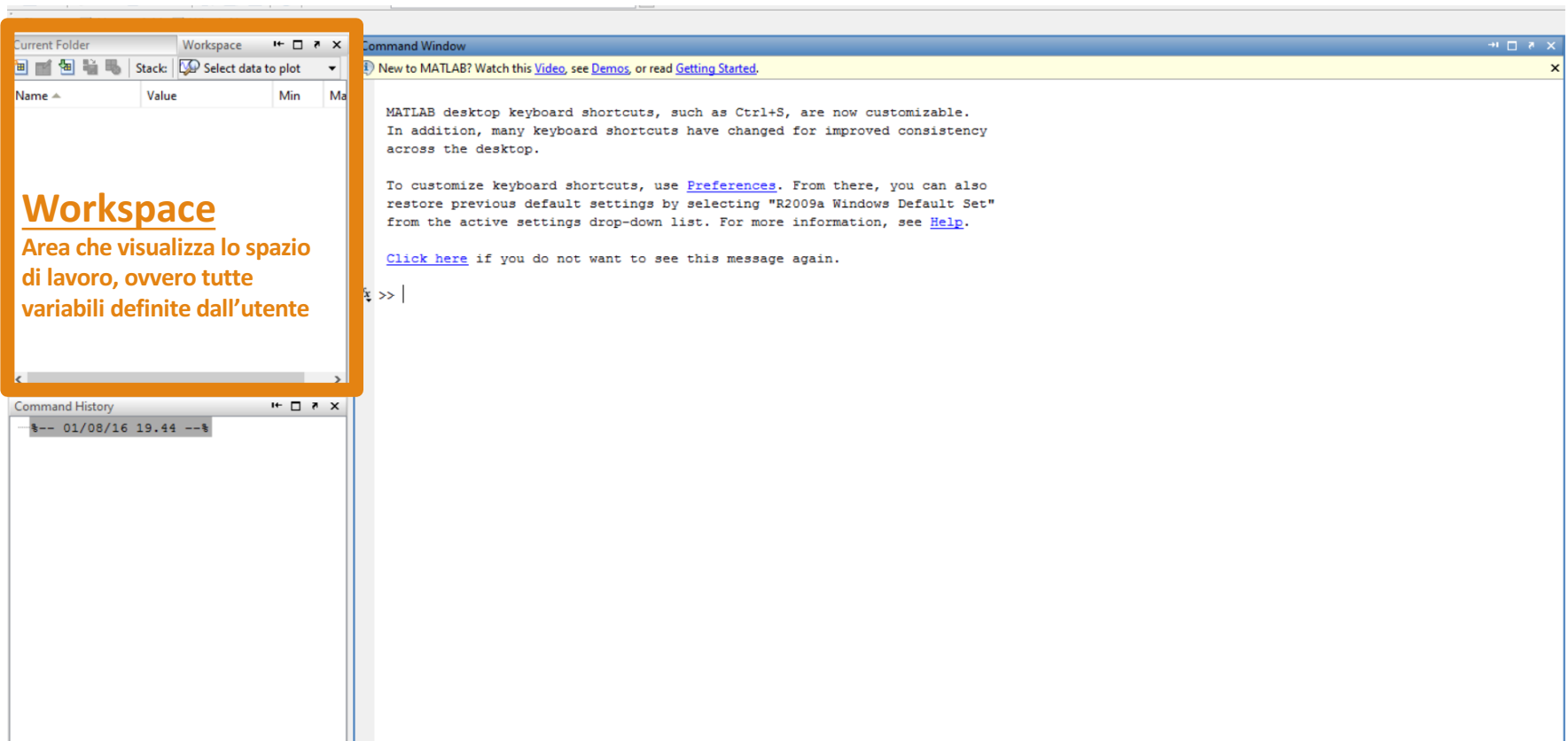
MATLAB: Command Window



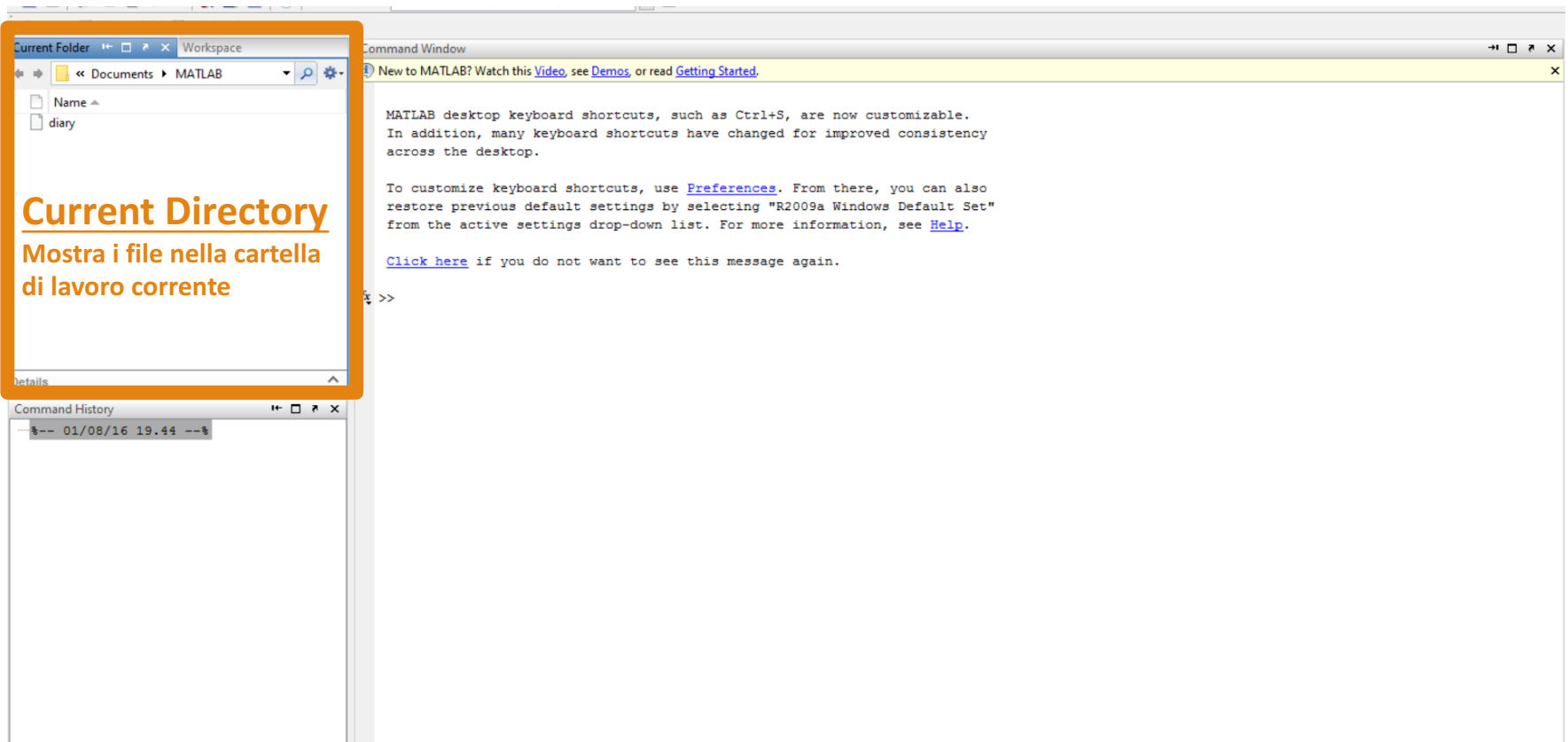
MATLAB: Command Window



MATLAB: Workspace & Current Directory



MATLAB: Workspace & Current Directory



MATLAB: Command History (Cronologia Comandi)



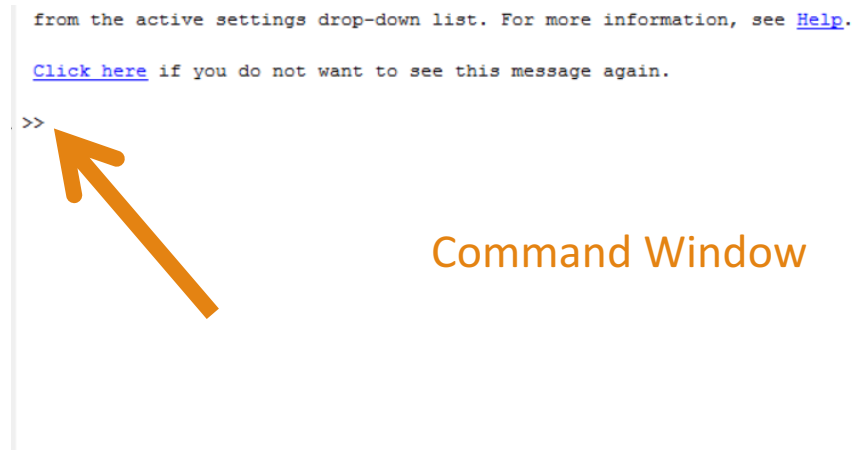
The screenshot displays the MATLAB desktop environment. On the left, a file browser shows the 'MATLAB' folder under 'Documents'. The main area is the 'Command Window', which contains a message about customizable keyboard shortcuts and a prompt 'f_x >>'. A 'Command History' window is open in the foreground, showing a list of commands with timestamps, such as '01/08/16 19.44'. The Command History window is highlighted with an orange border.

Command History
Mostra l'elenco dei comandi,
in ordine cronologico,
che sono stati digitati nella
Command Window

MATLAB: Inserimento dei Comandi – 1/3

- La prima modalità per l'inserimento dei comandi in MATLAB è quella mediante **Command Window**
 - I comandi vengono inseriti dopo il *prompt*, contrassegnato con «>>»

```
from the active settings drop-down list. For more information, see Help.  
Click here if you do not want to see this message again.  
>>
```



Command Window

MATLAB: Inserimento dei Comandi – Esempio

```
>>  
>> version  
  
ans =  
9.0.0.341360 (R2016a)  
  
>>
```

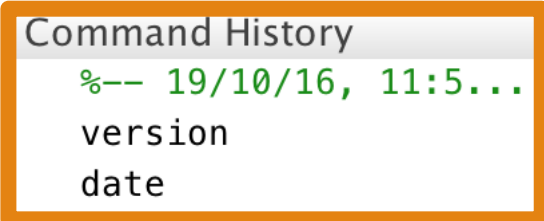
Mediante il comando **version** è possibile visualizzare la versione di MATLAB correntemente in uso

```
>>  
>> date  
  
ans =  
19-Oct-2016  
  
>>
```

Mediante il comando **date** è possibile visualizzare la data corrente

MATLAB: Inserimento dei Comandi – 2/3

- Utilizzo dei ***tasti freccia*** per scorrere i comandi precedentemente digitati nella **Command Window**
- Utilizzo della **Command History** per **ripetere i comandi già digitati**
- NOTA: Con il comando **clc** è possibile cancellare la **Command Window**



```
Command History
%-- 19/10/16, 11:5...
version
date
```



Command History contenente i comandi digitati in precedenza

MATLAB: Inserimento dei Comandi – 3/3

- Dal prompt è anche possibile inserire **espressioni aritmetiche**
- **Esempio 1** (Nota: in **grassetto** l'input dell'utente)

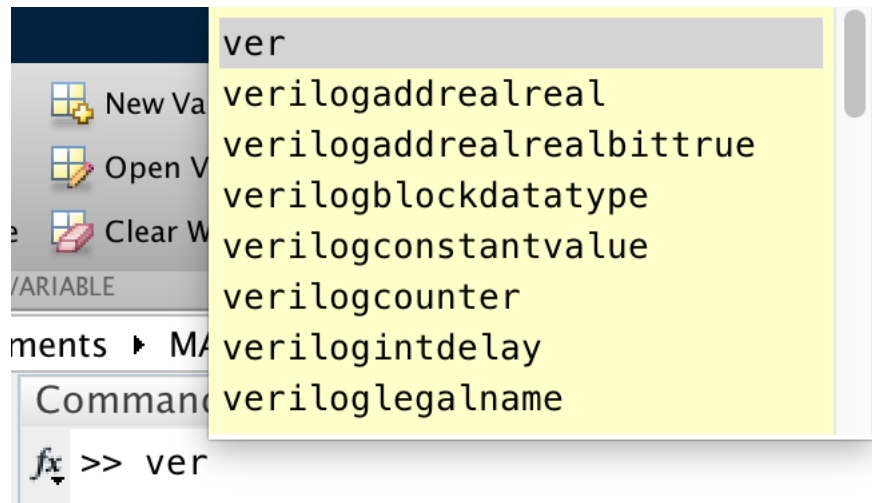
```
>> 4 + 2  
ans =  
    6
```

- **Esempio 2** (Nota: in **grassetto** l'input dell'utente)

```
>> log(4)  
ans =  
    1.3863
```

MATLAB: Inserimento dei Comandi – Tasto tab

- Utilizzando il *tasto tab* è possibile visualizzare tutti i comandi il cui nome comincia per la stringa testuale digitata fino a quel momento
- **Esempio**
 - Supponendo di aver scritto la stringa **ver** nel prompt, premendo il *tasto tab* otterremo



Le Variabili – 1/2

- Il risultato di ogni espressione viene **assegnato** ad **una variabile**
- **Variabile**: nome mnemonico utilizzato per memorizzare un dato

Le Variabili – 1/2

- Il risultato di ogni espressione viene **assegnato** ad **una variabile**
- **Variabile**: nome mnemonico utilizzato per memorizzare un dato

```
>> 4 + 2  
ans =  
    6
```

Le Variabili – 1/2

- Il risultato di ogni espressione viene **assegnato** ad **una variabile**
- **Variabile**: nome mnemonico utilizzato per memorizzare un dato

```
>> 4 + 2
ans =
 6
```



ans è una variabile ed in questo esempio contiene il valore 6

Le Variabili – 1/2

- Il risultato di ogni espressione viene **assegnato** ad **una variabile**
- **Variabile**: nome mnemonico utilizzato per memorizzare un dato

```
>> 4 + 2
```

```
ans =
```

NOTA: MATLAB assegna automaticamente il risultato di un'espressione ad una variabile speciale, denominata `ans`

`ans` è una variabile che contiene il valore 6
nell'esempio

Le Variabili – 2/2

- Le variabili possono essere utilizzate
 - Per memorizzare un dato
 - Ma anche nei calcoli successivi
- ***Esempio***

```
>> 4 + 2
ans =
    6
>> ans * 4
ans =
   24
```

Definizione di Variabili

- È possibile **definire** le variabili all'interno **dell'ambiente di lavoro (Workspace)** di MATLAB
- Le variabili **rimangono *attive* (in memoria)** fino a quando **non termina la sessione** di MATLAB
 - Quando il programma viene chiuso
- ***Esempio:*** definizione della variabile `p`

```
>> p = 3  
p =  
    3
```

Definizione di Variabili

- È possibile **definire** le variabili all'interno **dell'ambiente di lavoro (Workspace)** di MATLAB
- Le variabili **rimangono *attive* (in memoria)** fino a quando **non termina la sessione** di MATLAB
 - Quando il programma viene chiuso
- ***Esempio:*** definizione della variabile `p`

```
>> p = 3
```

```
p =  
3
```

Il simbolo `=` è detto operatore di assegnazione o sostituzione

Definizione di Variabili

- È possibile **definire** le variabili all'interno **dell'ambiente di lavoro (Workspace)** di MATLAB
- Le variabili **rimangono *attive* (in memoria)** fino a quando **non termina la sessione** di MATLAB
 - Quando il programma viene chiuso
- **Esempio:** definizione della variabile `p`


```
>> p = 3  
p =  
    3
```

N.B. l'operatore di assegnazione non deve essere confuso con l'operatore di uguaglianza, che è formato da due segni uguale (`==`)

L'Operatore di Assegnazione

- Quando definiamo una variabile, gli assegniamo un valore tramite il simbolo =

```
>> p = 3  
p =  
    3
```



Se digitiamo **p = 3**, stiamo indicando a MATLAB di assegnare il valore **3** alla variabile **p**

L'Operatore di Assegnazione

- Quando definiamo una variabile, gli assegniamo un valore tramite il simbolo =

```
>> p = 3  
p =  
    3
```

Il valore di `p` rimane tale, fino a quando non gli viene assegnato un nuovo valore

L'Operatore di Assegnazione

- Quando definiamo una variabile, gli assegniamo un valore tramite il simbolo **=**

```
>> p = 3
p =
    3
>> p * 10
ans =
    30
>> p
p =
    3
>> p = 7
p =
    7
```

Il valore di `p` rimane tale, fino a quando non gli viene assegnato un nuovo valore

L'Operatore di Assegnazione: Esercizio

```
>> z = 12  
z =  
    12  
>> m = z * 2  
m =  
    24  
>> z = z + 3
```

Quale sarà il valore di z ?
E di m ?

L'Operatore di Assegnazione: Esercizio (Soluzione)

```
>> z = 12  
z =  
    12  
>> m = z * 2  
m =  
    24  
>> z = z + 3
```

Quale sarà il valore di z ? 15
E di m ? 24

Assegnazione

`variabile = espressione`

- La **variabile** a sinistra del simbolo **=** è sostituita con il risultato dell'**espressione**
- *Esempio*

```
>> a = 1 + 3 - 2  
a =  
    2
```

Assegnazioni

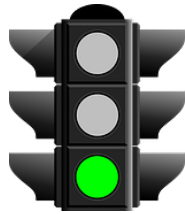
Corrette

$$x = 3$$

$$x = x + 3$$

$$y = x + 2$$

$$z = y * 4$$



Errate

$$3 = x$$

$$x + 2 = 20$$

$$y * 4 = z$$

$$z = 2 * x$$

Errata se e solo se x non è stata precedentemente definita




Che valore può assumere una variabile?

- Singolo numero (anche complesso)
- Stringa
- Vettore
- Matrice
- ...

Operatori per Operazioni Aritmetiche Scalari

Simbolo	Operazione	Sintassi in MATLAB
\wedge	Elevazione a Potenza a^b	a^b
$*$	Moltiplicazione $a b$	$a*b$
$/$	Divisione a destra (o <i>diretta</i>) $a / b = a : b$	a/b
\backslash	Divisione a sinistra (o <i>inversa</i>) $a \backslash b = b : a$	$a \backslash b$
$+$	Addizione $a + b$	$a+b$
$-$	Sottrazione $a - b$	$a-b$

Precedenza Operatori



Livello di Precedenza	Operazione
Primo	Parantesi valutate a partire dalla coppia più interna
Secondo	Elevazione a Potenza (^) valutata da sinistra a destra
Terzo	Moltiplicazione (*) Divisione valutate da sinistra a destra
Quarto	Addizione (+) Sottrazione (-) valutate da sinistra a destra

Precedenza Operatori: Esempi

Equivalenti {

- $a = 8 + 3 * 5$ [Il risultato è 23]
- $a = 8 + (3 * 5)$ [Il risultato è 23]
- $a = (8 + 3) * 5$ [Il risultato è 55]

- $a = 4^2 - 12 - 8 / 4 * 2$ [Il risultato è 0]
- $a = 4^2 - 12 - 8 / (4 * 2)$ [Il risultato è 3]

- $a = 3 * 4^2 + 5$ [Il risultato è 53]
- $a = (3 * 4)^2 + 5$ [Il risultato è 149]

Equivalenti {

- $a = 27^{(1/3)} + 9^{(0.5)}$ [Il risultato è 6]
- $a = 27^{(1/3)} + 9^{0.5}$ [Il risultato è 6]
- $a = 27^{1/3} + 9^{0.5}$ [Il risultato è 12]

Il Workspace – 1/2

- Il termine **workspace** in MATLAB indica i **nomi** ed i **valori** di tutte le variabili utilizzate nella sessione di lavoro corrente
- I nomi delle variabili
 - Devono iniziare con una lettera
 - Non possono essere più lunghi di 32 caratteri
 - **NOTA IMPORTANTE:** MATLAB fa differenza tra lettere maiuscole e minuscole (*case sensitive*)
 - La variabile **A** è **diversa** dalla variabile **a**

Il Workspace – 1/2

Consistent Naming Scheme

- I nomi di variabili devono avere delimitatori di parola
 - **NOTA IMPORTANTE:** I nomi delle variabili non possono contenere spazi
- Sono due le opzioni più comuni
 - **camelCase**
 - Prima lettera di ogni parola è in maiuscolo (di solito tranne la prima parola)
 - Esempio
 - `sommaPositivi`
 - **Underscore**
 - Underscore (trattino basso) tra le parole (`_`)
 - Esempio
 - `somma_positivi`

Il Workspace – 1/2

Consistent Naming Scheme

- I **nomi di variabili** devono avere delimitatori di parola
- **NOTA IMPORTANTE:** I nomi delle variabili non possono contenere spazi

- Sono
- cam
- Pri
- par
- E
- Und
- Un
- E

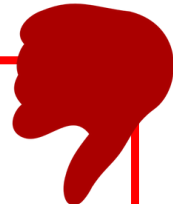
Usare nomi significativi e coerenti per le variabili

- I nomi delle variabili devono essere descrittivi

somma = somma + valore
...
media = somma / num_valori



A = A + B
...
C = A / T



- Usare nomi coerenti per le variabili che hanno lo stesso tipo di ruolo
 - *Esempio:* rappresentiamo due *somme* (stesso tipo di ruolo)
 - somma_positivi
 - somma_negativi

Il Workspace – 2/2

- Salvataggio delle variabili in un file
 - `save filename var`
 - Crea un file chiamato **filename.mat**, in cui verrà memorizzata la variabile **var** e i dati relativi ad essa

- *Esempio:* `p=5`

```
>> p=5  
  
p =  
  
    5  
  
>> save primoSave p
```

Il Workspace – 2/2

- Salvataggio delle variabili in un file
 - `save('filename', 'var1', 'var2', ...)`
 - Crea un file chiamato **filename.mat**, in cui verranno memorizzate le variabili **var1**, **var2**, ... e i dati relativi ad essa
 - **Esempio:** `p=5`, `q=7`

```
>> p=5

p =

     5

>> q=7

q =

     7

>> save ('secondoSave', 'p', 'q')
```


Il Workspace – 2/2

- Caricamento delle variabili da un file
 - `load 'filename.mat'`
 - Carica nel workspace tutti i dati e le variabili contenute in **filename.mat**
 - ***Esempio:*** secondoSave.mat

The image shows two screenshots of the MATLAB interface. The top screenshot shows an empty Command Window and an empty Workspace. The bottom screenshot shows the Command Window with the command `load 'secondoSave'` entered, where the file name is circled in blue. A blue arrow points from the text "Il .mat può anche essere omissso" to the circled file name. The Workspace on the right now contains two variables: 'p' with a value of 5 and 'q' with a value of 7. A large orange arrow points from the top screenshot to the bottom one.

Il .mat può anche essere omissso

```
Command Window
```

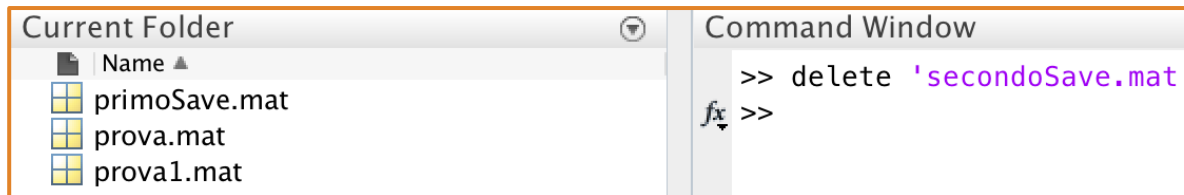
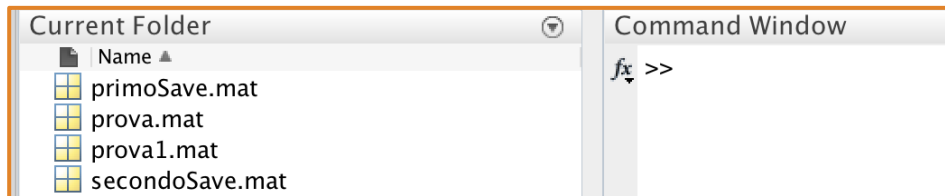
```
>> load 'secondoSave'
```

```
Workspace
```

Name	Value
p	5
q	7

Il Workspace – 2/2

- Cancellazione delle variabili contenute in un file
 - `delete 'filename.mat'`
 - Cancella dalla directory corrente (*Current Directory*) il file chiamato **filename.mat**
 - **Attenzione:** funziona anche con qualsiasi altro tipo di file
 - Non solo per i file .mat



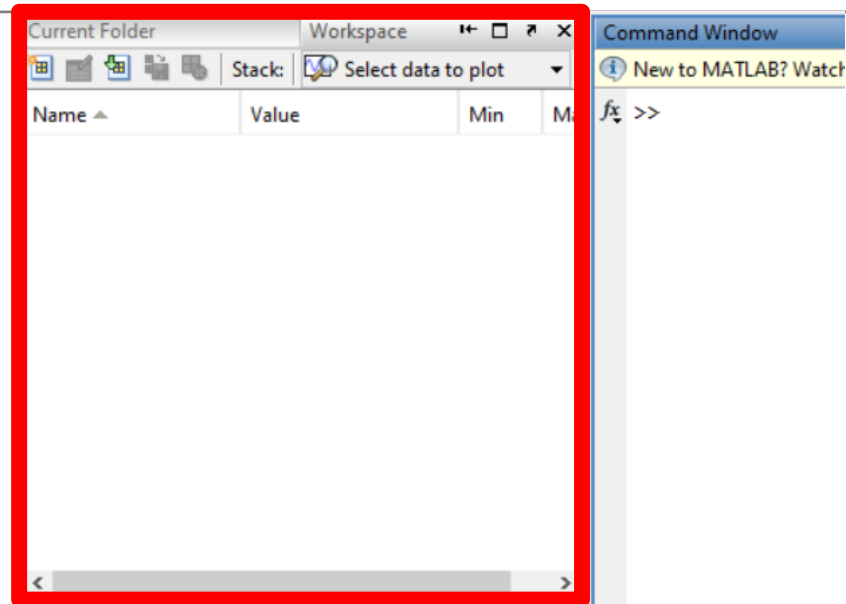
Sessione di Lavoro – 1/4

- **clear**
 - Elimina tutte le variabili dalla memoria

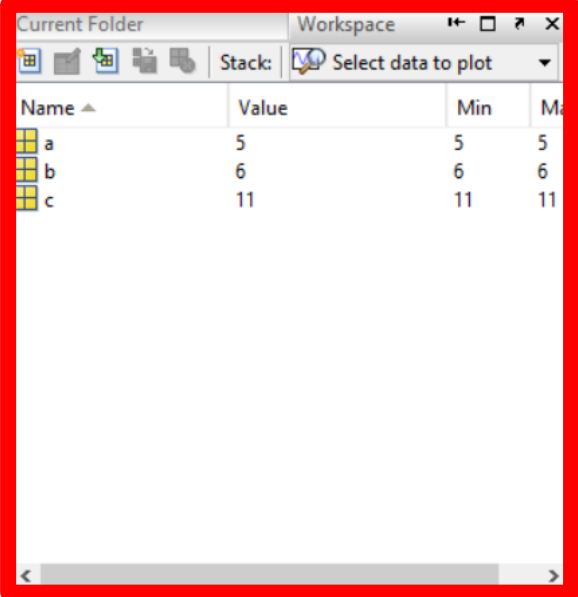
- **clear var1, var2**
 - Elimina le variabili `var1` e `var2` dalla memoria

Sessione di Lavoro – 1/4

Finestra Workspace (1)



Sessione di Lavoro – 1/4 Finestra Workspace (2)



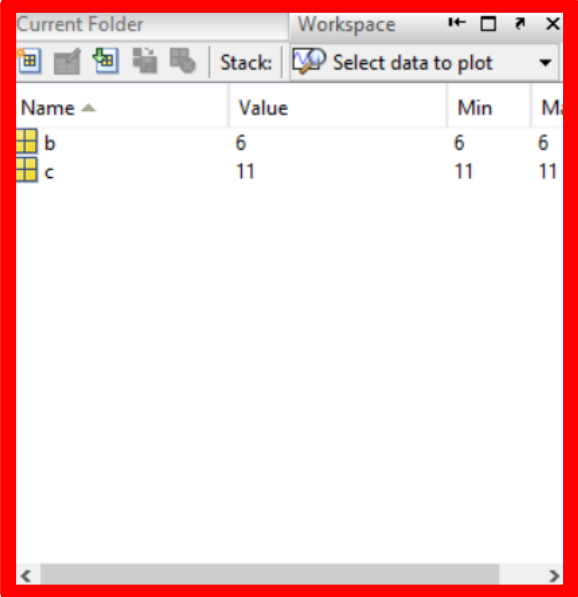
The screenshot shows the MATLAB Workspace window. The 'Current Folder' tab is active, and the 'Stack' is set to 'Select data to plot'. The workspace contains three variables: 'a' with a value of 5, 'b' with a value of 6, and 'c' with a value of 11. The 'Command Window' is also visible, showing the execution of the commands: `>> a = 5`, `>> b = 6`, and `>> c = a + b`. The output shows the values of 'a' (5), 'b' (6), and 'c' (11).

Name	Value	Min	Max
a	5	5	5
b	6	6	6
c	11	11	11

```
Command Window
New to MATLAB? Watch

>>
>> a = 5
a =
    5
>> b = 6
b =
    6
>> c = a + b
c =
   11
fx >>
```

Sessione di Lavoro – 1/4 Finestra Workspace (3)



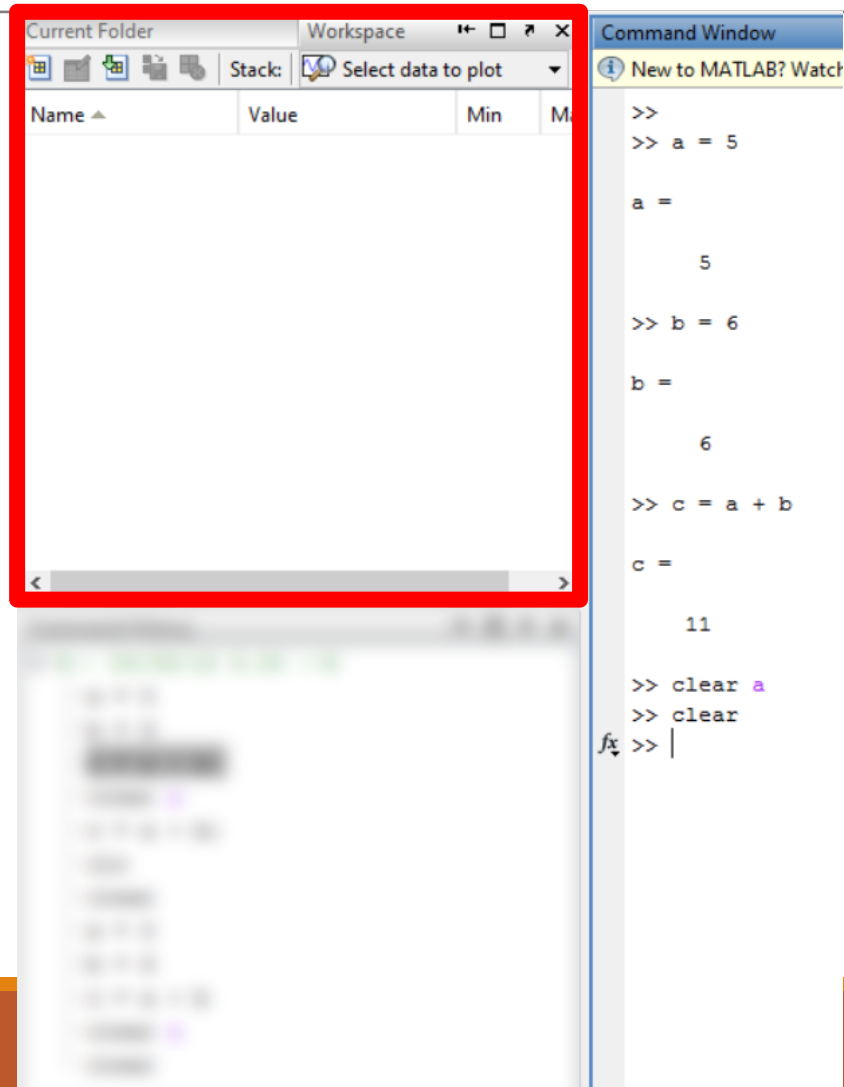
The screenshot shows the MATLAB Workspace window. The 'Current Folder' tab is active, and the 'Stack' is set to 'Select data to plot'. The workspace contains two variables: 'b' with a value of 6 and 'c' with a value of 11. The 'Command Window' is also visible, showing the execution of MATLAB commands: `>> a = 5`, `>> b = 6`, `>> c = a + b`, and `>> clear a`.

Name	Value	Min	Max
b	6	6	6
c	11	11	11

```
Command Window
New to MATLAB? Watch

>>
>> a = 5
a =
    5
>> b = 6
b =
    6
>> c = a + b
c =
    11
>> clear a
fx >> |
```

Sessione di Lavoro – 1/4 Finestra Workspace (4)



The image shows a screenshot of the MATLAB interface. The 'Workspace' window is highlighted with a red border and is currently empty. The 'Command Window' is visible on the right, showing the following commands and outputs:

```
>>  
>> a = 5  
  
a =  
  
    5  
  
>> b = 6  
  
b =  
  
    6  
  
>> c = a + b  
  
c =  
  
    11  
  
>> clear a  
>> clear  
fx >> |
```

Sessione di Lavoro – 2/4

- **who**

- Elenca le variabili che si trovano nel Workspace

- **whos**

- Elenca le variabili che si trovano nel Workspace e lo spazio che esse occupano in memoria
- È quello che viene mostrato graficamente nella finestra Workspace

Sessione di Lavoro – 2/4: Esempio

```
>> a = 3
a =
    3
>> b = 5
b =
    5
>> who
Your variables are:
a  b
>> whos
      Name      Size      Bytes  Class  Attributes
      a         1x1         8  double
      b         1x1         8  double
```

Sessione di Lavoro – 3/4

- **exist('nome')**
 - Determina se esiste una variabile (o un file) con uno specifico **nome**
 - Restituisce 1 se la variabile (o il file) esiste, 0 altrimenti

- **quit**
 - Chiude MATLAB

Sessione di Lavoro – 3/4: Esempio

```
>> clear a
>> whos
  Name      Size      Bytes  Class  Attributes
  b         1x1         8    double
>> exist('b')
ans =
     1
>> exist('a')
ans =
     0
>>
```

Sessione di Lavoro – 4/4

- ,
 - Separa le istruzioni

- ;
 - Esclude la visualizzazione del risultato di un'istruzione

- ...
 - Continua l'istruzione alla riga successiva

Sessione di Lavoro – 4/4: Esempio

```
>> x = 7, y = 6
x =
    7
y =
    6
>> r = 8;
>> h = 3;
>> v = 3.14*r^2*h
v =
  602.8800
>> u = 3 + 7 ...
- 4
u =
    6
```

Costanti e Variabili Predefinite – 1/2

- **ans**
 - Variabile temporanea che mantiene il risultato più recente
- **Inf**
 - Infinito
- **NaN**
 - Not a Number
- **pi**
 - Il numero π

Costanti e Variabili Predefinite – 2/2

- Nell'esempio precedente potremmo usare `pi` al posto di `3.14`...

```
>> r = 8
r =
    8
>> h = 15
h =
   15
>> V = r^2 * 3.14 * h
V =
 3014.4
```



```
>> r = 8
r =
    8
>> h = 15
h =
   15
>> V = r^2 * pi * h
V =
 3015.9
```

Istruzioni, Comandi e Funzioni

- **Istruzioni**

- Non possono avere argomenti
 - *Esempi: quit, clc*

- **Comandi**

- Non richiedono obbligatoriamente argomenti, ma possono averne
- Se hanno argomenti, questi ultimi NON sono racchiusi tra parentesi
 - *Esempio: clear x*

- **Funzioni**

- Richiedono argomenti che devono essere racchiusi tra parentesi
 - *Esempio: sqrt(x)*

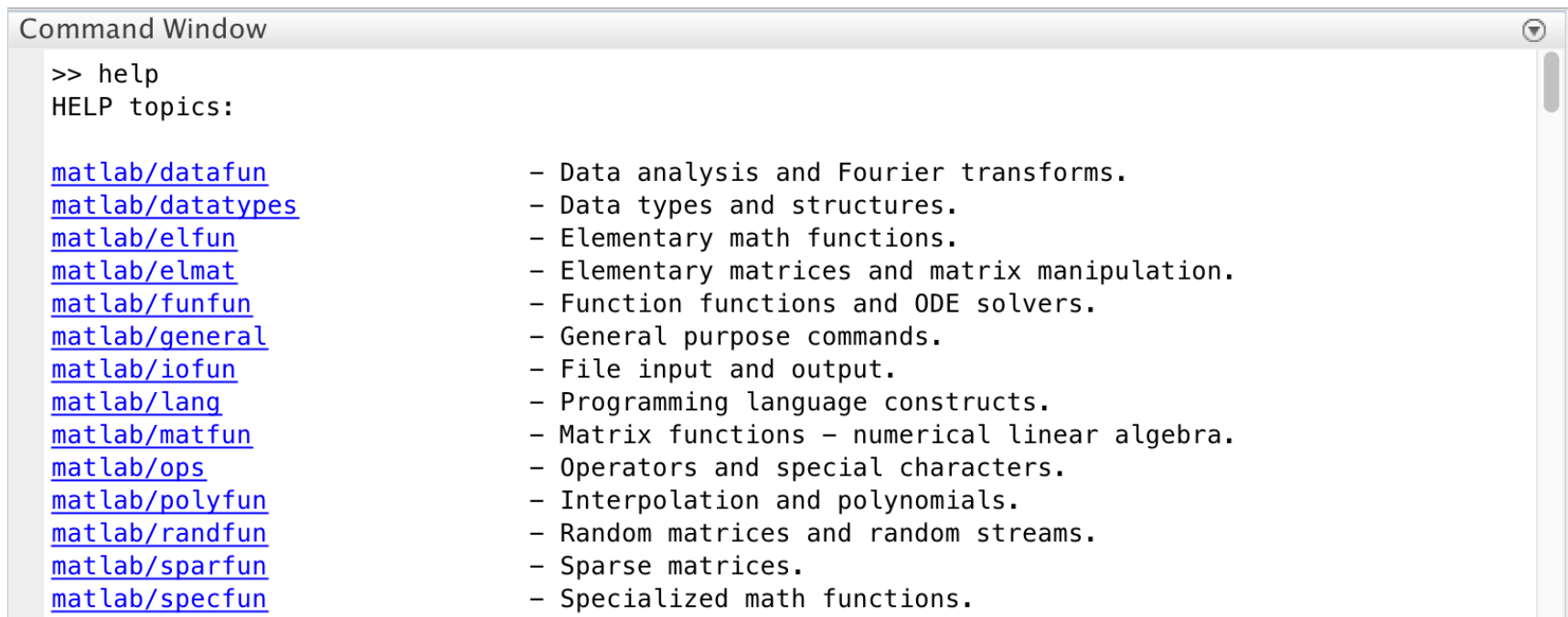
Funzionalità di Help

- È essenziale poter accedere alle informazioni sulla sintassi e le funzionalità di un linguaggio di programmazione, oltre ad eventuali esempi applicativi
 - Help
- Esistono due modalità attraverso cui l'utente può accedere a tali informazioni in MATLAB
 - Interfaccia testuale
 - Interfaccia grafica



Funzionalità di Help: Interfaccia Testuale

- L'aiuto basato su interfaccia testuale può essere ottenuto digitando **help** nel prompt dei comandi



```
Command Window
>> help
HELP topics:

matlab/datafun           - Data analysis and Fourier transforms.
matlab/datatypes        - Data types and structures.
matlab/elfun            - Elementary math functions.
matlab/elmat           - Elementary matrices and matrix manipulation.
matlab/funfun          - Function functions and ODE solvers.
matlab/general         - General purpose commands.
matlab/iofun           - File input and output.
matlab/lang            - Programming language constructs.
matlab/matfun          - Matrix functions - numerical linear algebra.
matlab/ops             - Operators and special characters.
matlab/polyfun         - Interpolation and polynomials.
matlab/randfun         - Random matrices and random streams.
matlab/sparfun        - Sparse matrices.
matlab/specfun        - Specialized math functions.
```

Funzionalità di Help: Interfaccia Testuale

- Per visualizzare i sotto-argomenti di uno degli argomenti (topics) mostrati in output dal comando **help**, bisogna digitare nel prompt dei comandi (>>) il nome dell'argomento di interesse

- **Esempio**

```
>> help matlab/general
```

```
General purpose commands.
```

```
MATLAB Version 9.0 (R2016a) 10-Feb-2016
```

```
General information.
```

```
syntax           - Help on MATLAB command syntax.
```

```
demo            - Run demonstrations.
```

```
ver             - MATLAB, Simulink and toolbox version information.
```

```
version        - MATLAB version information.
```

```
verLessThan    - Compare version of toolbox to specified version string.
```

```
logo           - Plot the L-shaped membrane logo with MATLAB lighting.
```

```
membrane      - Generates the MATLAB logo.
```

```
bench         - MATLAB Benchmark.
```



```
Managing the workspace.
```

Funzionalità di Help: Interfaccia Testuale

- Il comando **help** può anche essere usato per trovare informazioni su una determinata funzione
- *Esempio*

```
Command Window
>> help sin
sin      Sine of argument in radians.
sin(X) is the sine of the elements of X.

See also asin, sind.

Reference page for sin
Other functions named sin

fx >>
```

Funzionalità di Help: Interfaccia Testuale

- Nel caso in cui la funzione sia sconosciuta, o l'utente sia interessato ad una specifica parola chiave, può essere utilizzato il comando **lookfor**

- **Esempio**

```
Command Window
>> lookfor infinity
ceil
floor
inf
actdemo
dhint
dhinfopt
dintdemo
h2des1
hinfddata
hinfdes1
hinfkqjd
hinflim
hinfopt
hmatdemo
htdes1
josedemo
linf
linfdemo
normhinf
- Round towards plus infinity.
- Round towards minus infinity.
- Infinity.
- Demo of digital H-infinity hydraulic actuator design.
- Discrete H-Infinity control synthesis (bilinear transform version)
- Discrete H-Infinity control synthesis via Gamma iteration.
- Demo of H-Infinity design of double integrator plant.
- A function for use with the h-infinity Control Toolbox demonstrat.
- ABOUT HINFDM --- H_infinity DEMO
- A function for use with the h-infinity Control Toolbox demonstrat.
- Continuous H-Infinity control synthesis (Glover/Doyle formulae).
- Continuous H-Infinity control synthesis (Limebeer/Kasensally formu
- H-Infinity control synthesis via Gamma iteration.
- H-Infinity Fighter Design Demonstration (HIMAT).
- A function for use with the h-infinity Control Toolbox demonstrat.
- Demo of H-Infinity design for Large space structure.
- Continuous L-Infinity control synthesis.
- Demo of L-Infinity control design for large space structure.
- Continuous H-Infinity norm.
```

Funzionalità di Help: Interfaccia Testuale

- Il comando **lookfor** cercherà tutte le possibili corrispondenze con quanto cercato dall'utente
- Per scoprire i dettagli relativi ad uno dei risultati restituiti dal comando **lookfor**, può essere usato il comando **help**, così come visto in precedenza
- *Esempio*

```
>> help ceil
ceil    Round towards plus infinity.
         ceil(X) rounds the elements of X to the nearest integers
         towards infinity.

         See also floor, round, fix.

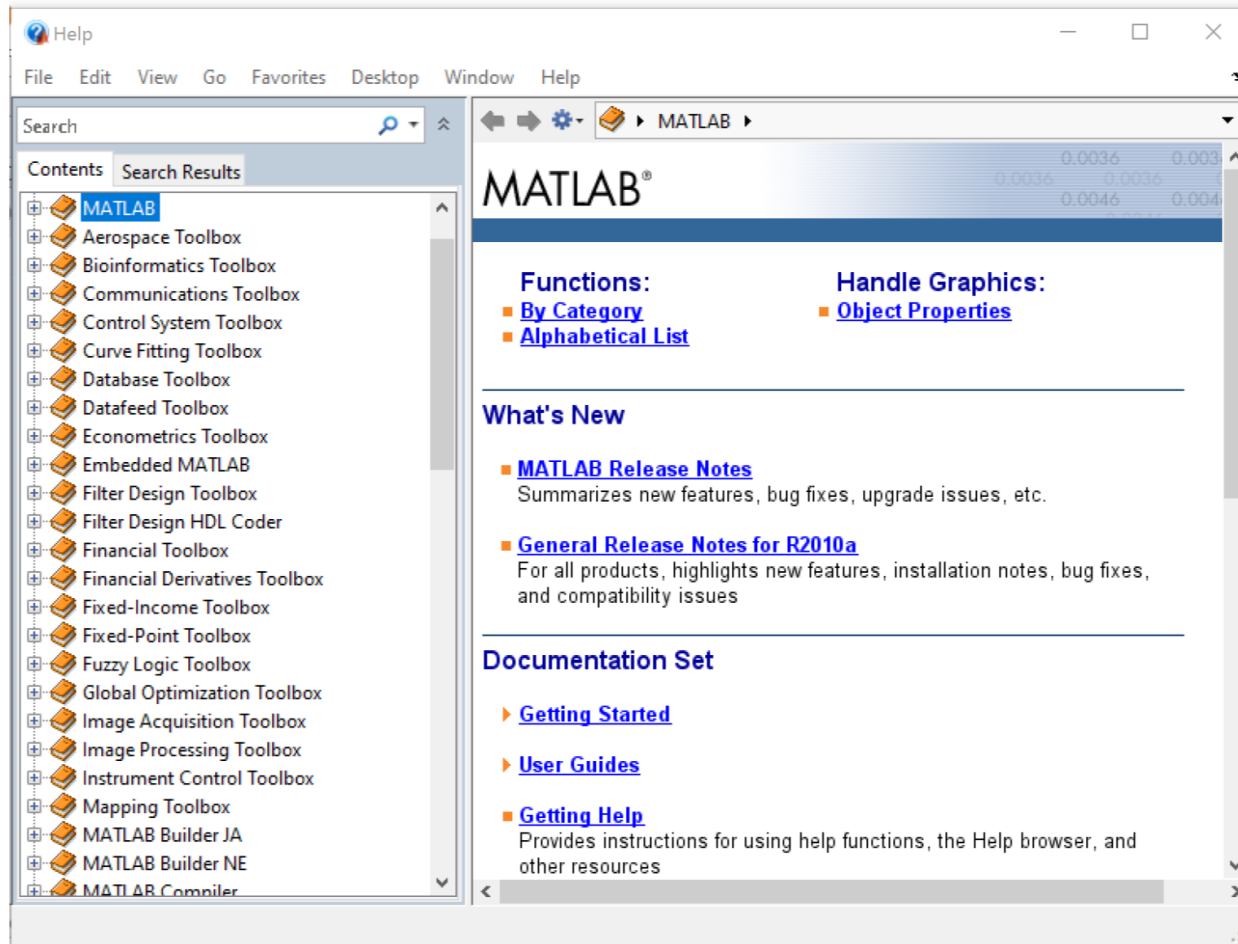
         Reference page for ceil
         Other functions named ceil

>>
```

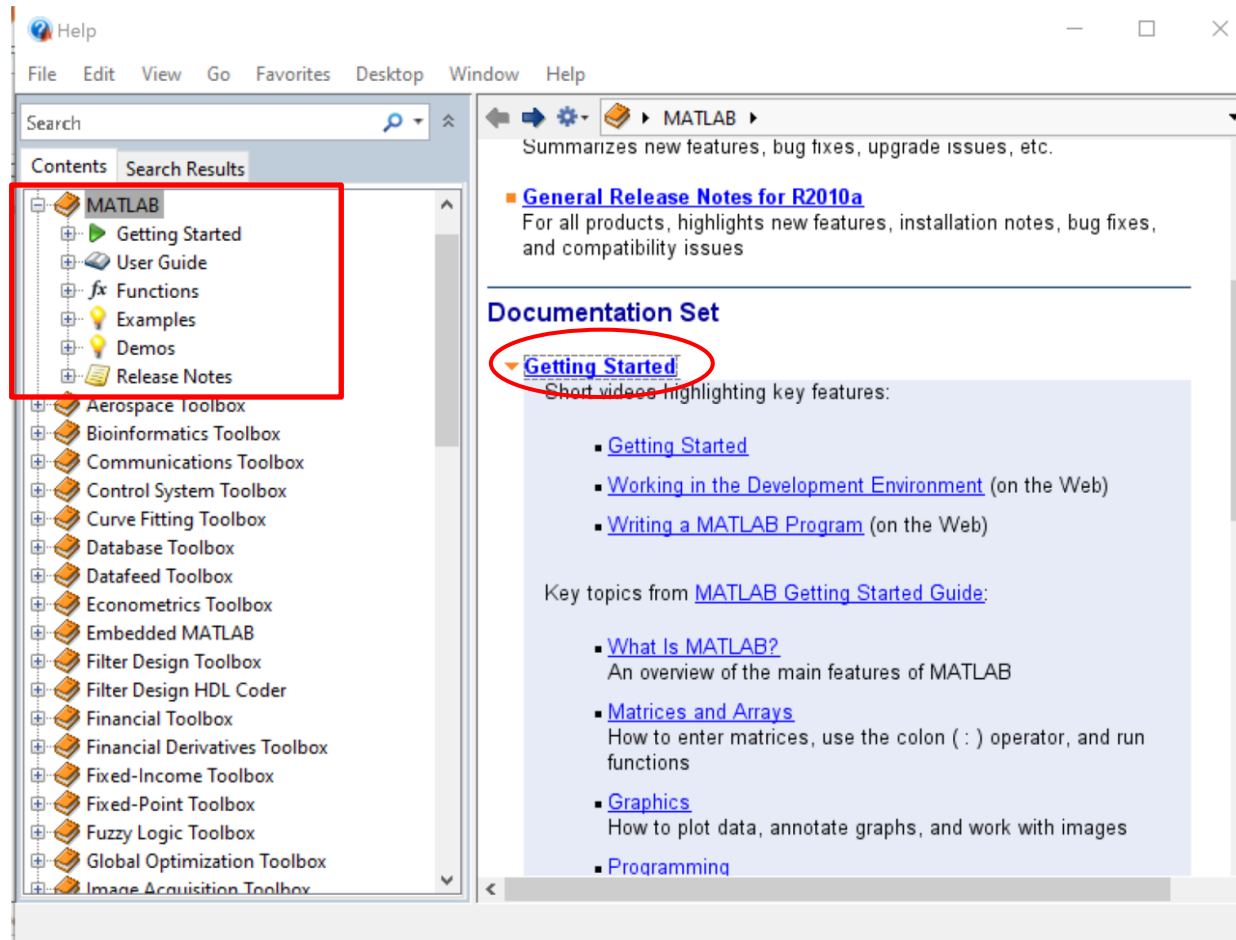
Funzionalità di Help: Interfaccia Grafica

- Una modalità più “user friendly” per accedere alle funzionalità di help è quella basata su interfaccia grafica
- Tale modalità può essere avviata digitando nel prompt uno dei seguenti due comandi
 - **helpbrowser**
 - **doc**

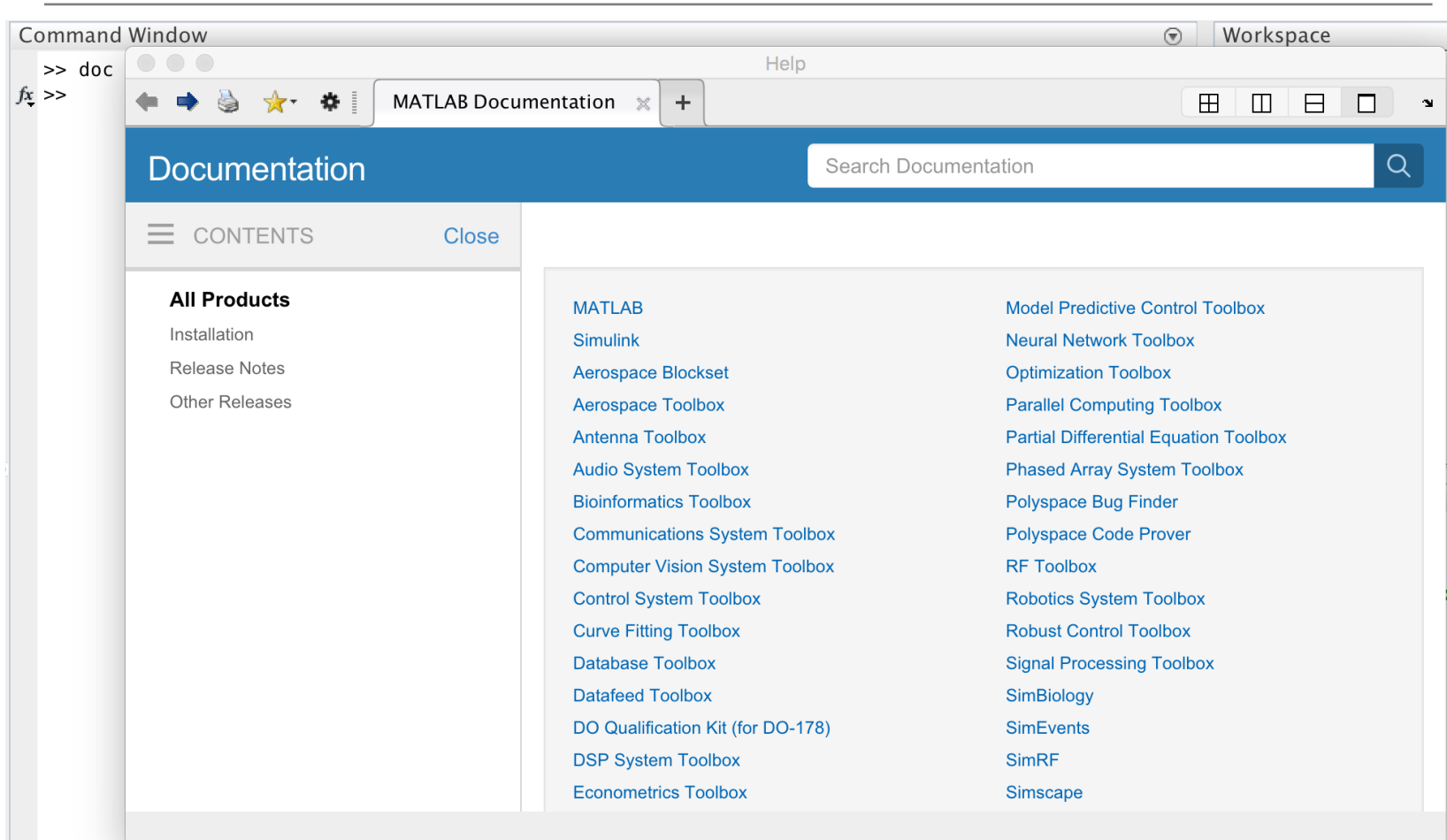
Funzionalità di Help: Interfaccia Grafica MATLAB 7



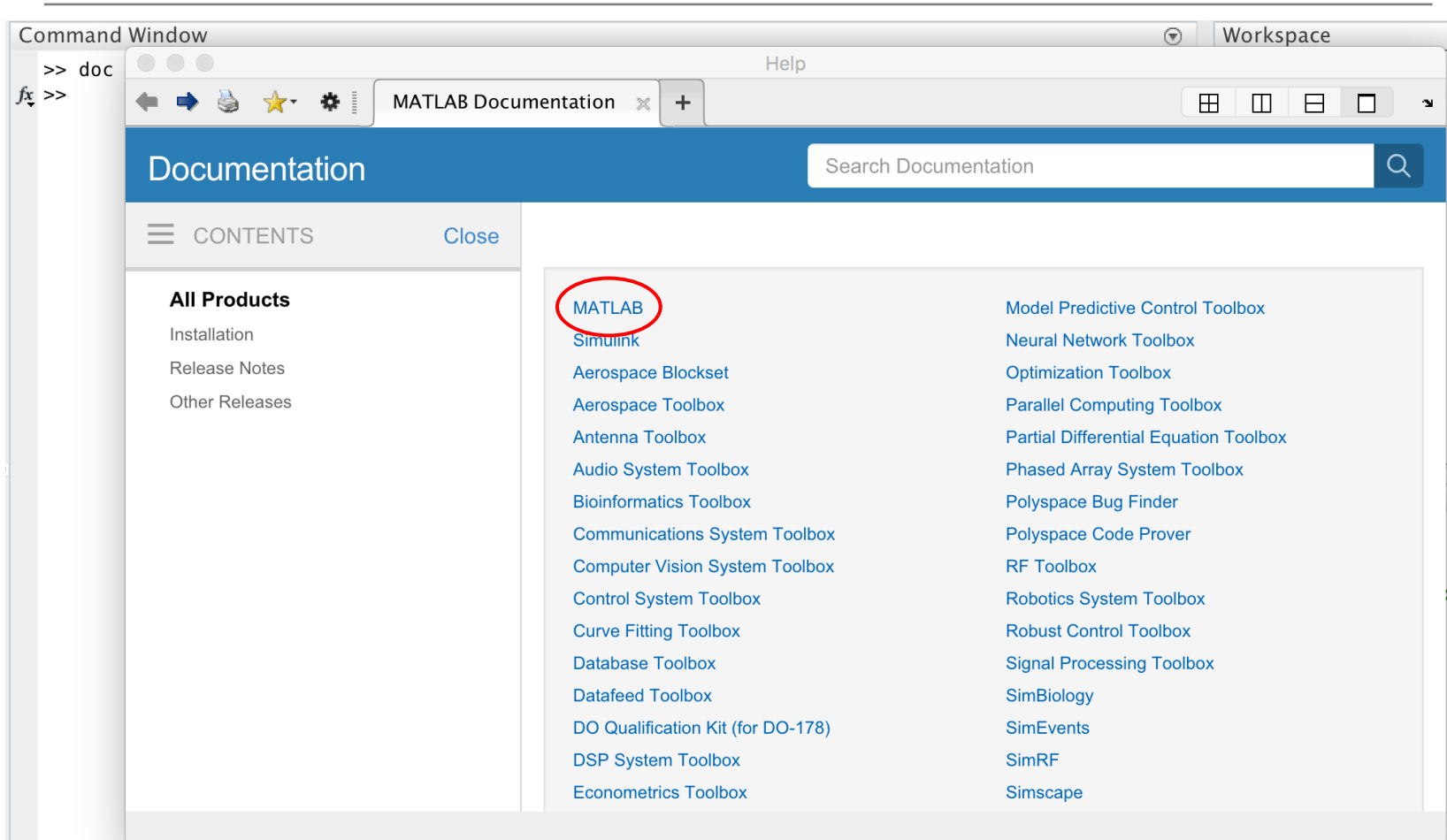
Funzionalità di Help: Interfaccia Grafica MATLAB 7



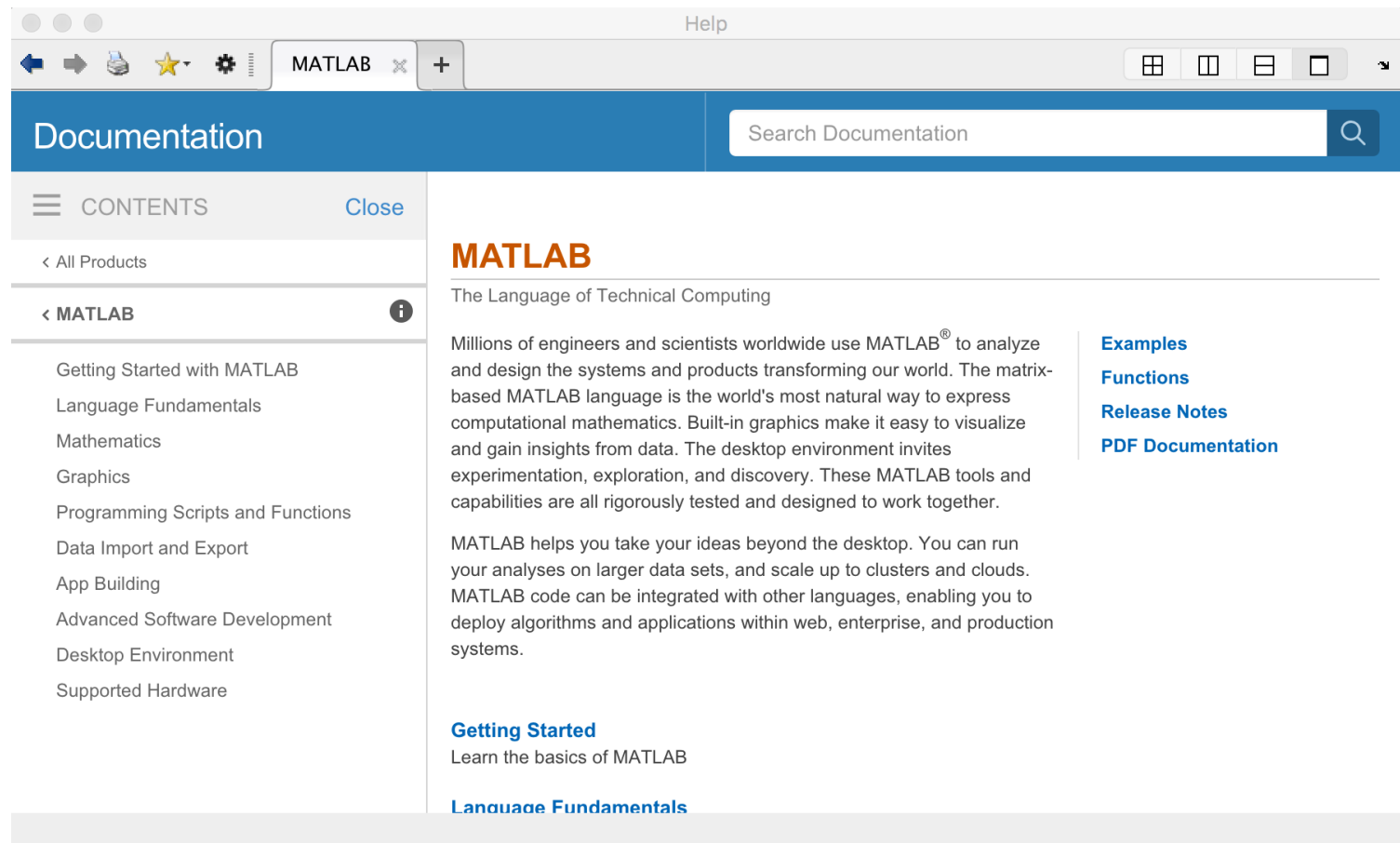
Funzionalità di Help: Interfaccia Grafica MATLAB 9



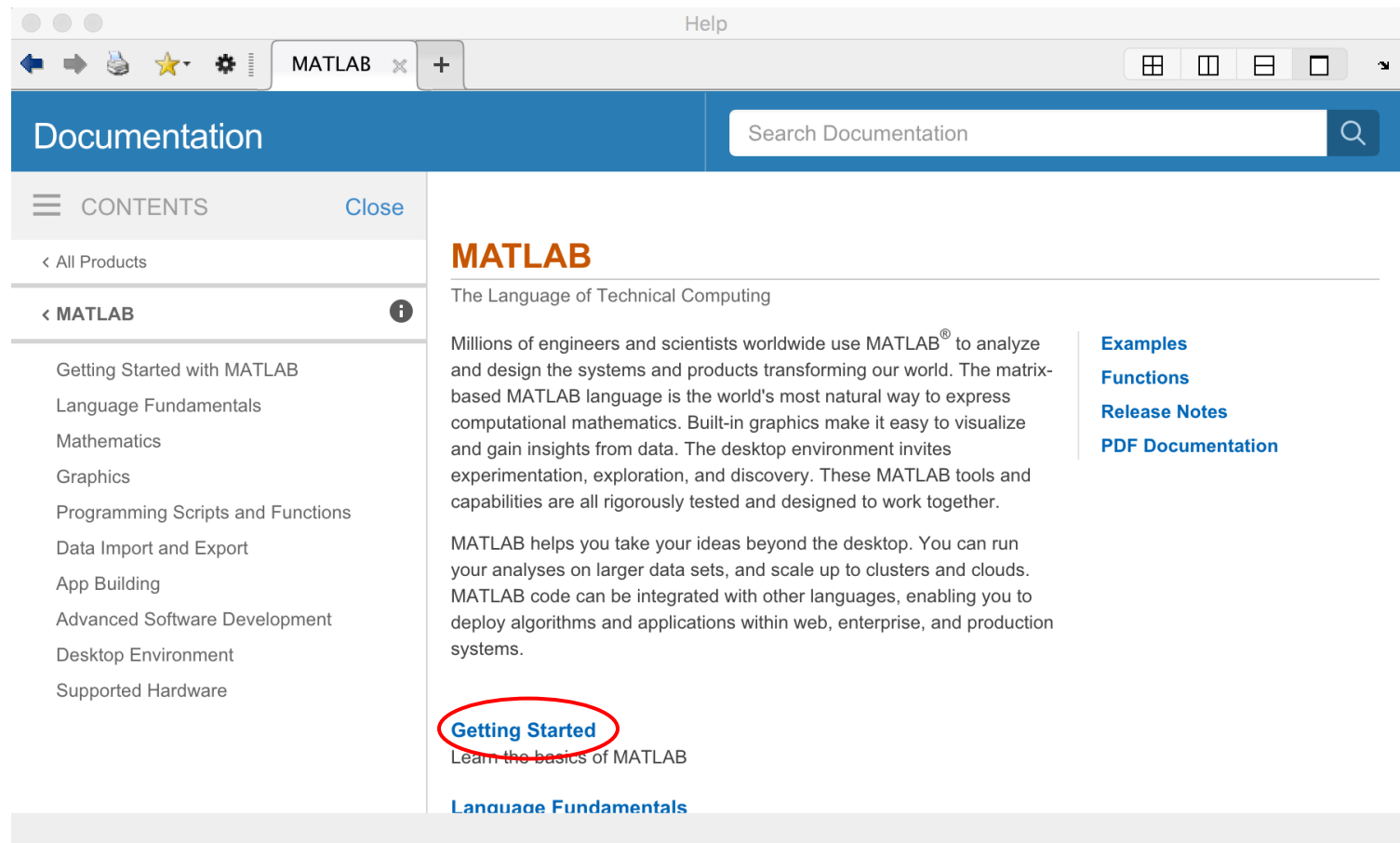
Funzionalità di Help: Interfaccia Grafica MATLAB 9



Funzionalità di Help: Interfaccia Grafica MATLAB 9



Funzionalità di Help: Interfaccia Grafica MATLAB 9



Funzionalità di Help: Interfaccia Grafica MATLAB 9

The screenshot shows the MATLAB 9 Help documentation interface. At the top, there is a browser-style address bar with the text "Getting Started with MATLAB" and a search bar labeled "Search Documentation". Below the address bar is a blue header with the word "Documentation" on the left and the search bar on the right. A left sidebar contains a "CONTENTS" menu with a "Close" button. The sidebar lists "All Products" and "MATLAB" (with an information icon). Under "MATLAB", the "Getting Started with MATLAB" section is expanded, showing a list of topics: Language Fundamentals, Mathematics, Graphics, Programming Scripts and Functions, Data Import and Export, App Building, Advanced Software Development, Desktop Environment, and Supported Hardware. The main content area displays the "Getting Started with MATLAB" page, which includes sections for "MATLAB Product Description" (with links to "MATLAB Product Description" and "System Requirements"), "Tutorials" (with links to "Desktop Basics", "Matrices and Arrays", "Array Indexing", "Workspace Variables", "Character Strings", "Calling Functions", "2-D and 3-D Plots", "Programming and Scripts", and "Help and Documentation"), and "Interactive Learning".

Funzionalità di Help: Interfaccia Grafica MATLAB 9

The screenshot shows the MATLAB 9 Help documentation interface. The window title is "Getting Started with MATLAB". The left sidebar contains a "CONTENTS" menu with "Getting Started with MATLAB" selected. The main content area displays the "Getting Started with MATLAB" page, with "Desktop Basics" circled in red under the "Tutorials" section.

Documentation Search Documentation

CONTENTS Close

- < All Products
- < **MATLAB** ⓘ

Getting Started with MATLAB

- Language Fundamentals
- Mathematics
- Graphics
- Programming Scripts and Functions
- Data Import and Export
- App Building
- Advanced Software Development
- Desktop Environment
- Supported Hardware

Getting Started with MATLAB

- MATLAB Product Description
- System Requirements

Tutorials

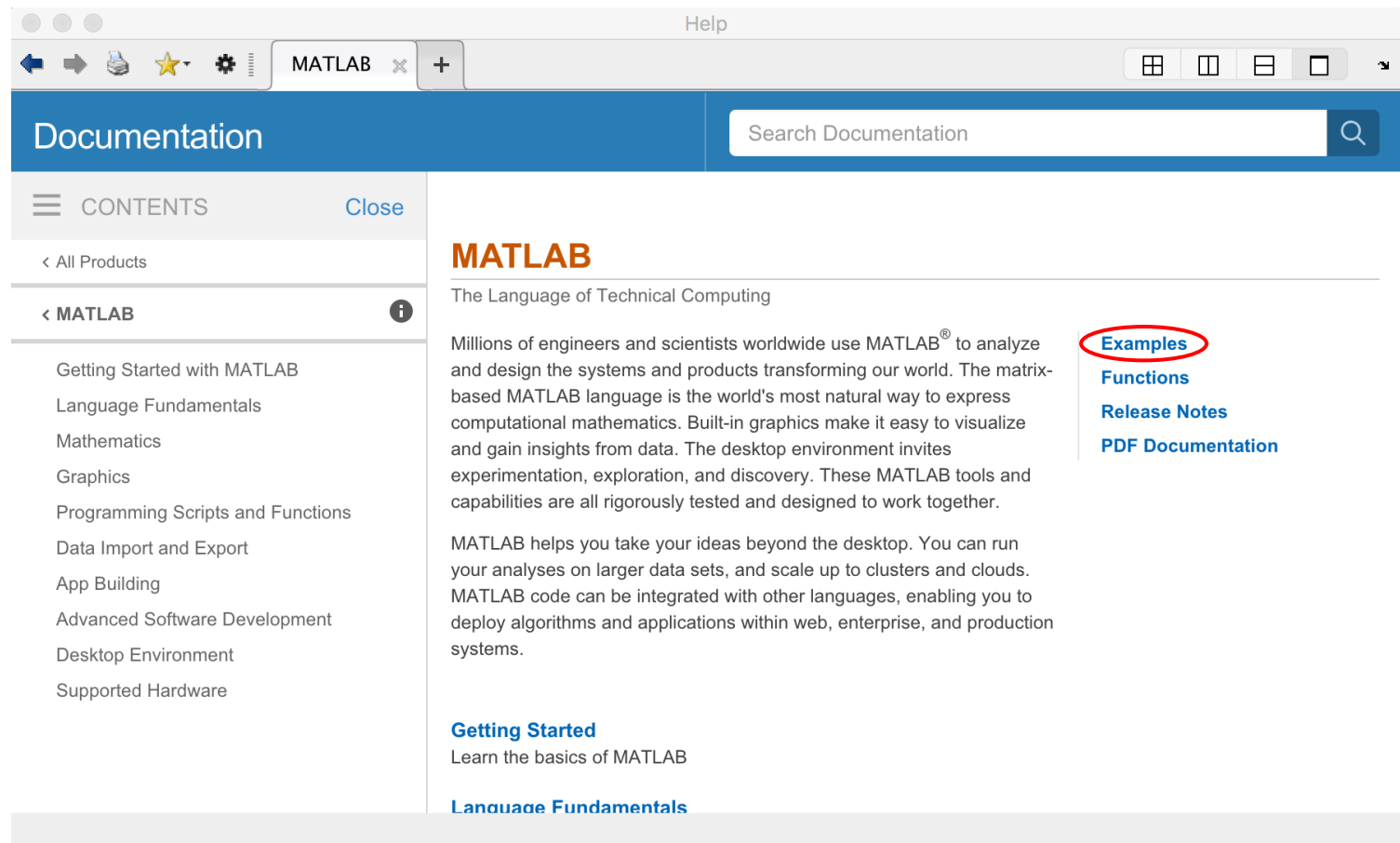
- Desktop Basics**
- Matrices and Arrays
- Array Indexing
- Workspace Variables
- Character Strings
- Calling Functions
- 2-D and 3-D Plots
- Programming and Scripts
- Help and Documentation

Interactive Learning

Funzionalità di Help: Interfaccia Grafica MATLAB 9

The screenshot displays the MATLAB 9 Help interface. At the top, there is a navigation bar with a search box labeled "Search Documentation". Below this, a blue header reads "Documentation". A left sidebar contains a "CONTENTS" menu with a "Close" button, showing a tree structure: "All Products", "MATLAB", and "Getting Started with MATLAB". The "Getting Started with MATLAB" section is expanded, showing "Desktop Basics" as the current page. Below the sidebar, the text "ON THIS PAGE" and "Next in Getting Started with MATLAB" are visible. The main content area features the title "Desktop Basics" and a paragraph: "When you start MATLAB®, the desktop appears in its default layout." Below the text is an embedded image of the MATLAB R2015a desktop environment. The desktop shows the MATLAB interface with a ribbon menu (HOME, PLOTS, APPS), a Command Window with the text "New to MATLAB? See resources for Getting Started." and a cursor, and a Workspace window.

Funzionalità di Help: Interfaccia Grafica MATLAB 9



The screenshot displays the MATLAB 9 Help interface. At the top, there is a search bar labeled "Search Documentation" and a "Documentation" header. Below the header, a "CONTENTS" sidebar is visible, listing various topics under the "MATLAB" category. The main content area features the "MATLAB" title, a subtitle "The Language of Technical Computing", and a paragraph describing the language's use by engineers and scientists. A sidebar on the right contains links to "Examples", "Functions", "Release Notes", and "PDF Documentation". The "Examples" link is circled in red. At the bottom of the main content area, there are links for "Getting Started" and "Language Fundamentals".

Documentation

Search Documentation

CONTENTS Close

< All Products

< MATLAB ⓘ

- Getting Started with MATLAB
- Language Fundamentals
- Mathematics
- Graphics
- Programming Scripts and Functions
- Data Import and Export
- App Building
- Advanced Software Development
- Desktop Environment
- Supported Hardware

MATLAB

The Language of Technical Computing

Millions of engineers and scientists worldwide use MATLAB® to analyze and design the systems and products transforming our world. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets, and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems.

[Examples](#)

[Functions](#)

[Release Notes](#)

[PDF Documentation](#)

[Getting Started](#)
Learn the basics of MATLAB

[Language Fundamentals](#)

Funzionalità di Help: Interfaccia Grafica MATLAB 9

The screenshot displays the MATLAB 9 Help documentation interface. At the top, a browser window shows the 'MATLAB Examples' tab. The main header is 'Documentation' with a search bar. A left sidebar contains a 'CONTENTS' menu with 'MATLAB Examples' selected. The main content area is titled 'MATLAB Examples' and features a 'Getting Started' section with four items:

- Getting Started with MATLAB (7 min)** - Video
- Working in The Development Environment (5 min, 21 sec)** - Video
- Using the Live Editor (5 min)** - Video
- Introduction to the Live Editor** - Live Script

Riferimenti

- Libro: MATLAB un'introduzione per gli ingegneri, McGraw-Hill
 - Capitolo 1
 - Paragrafi 1.1 e 1.2 (TUTTO)