# Organizing and Accessing Data in MATLAB

This reference shows common use cases, so it is not an exhaustive list.
The >> icon provides links to relevant sections of the MATLAB documentation.

## Representing Data

### Homogenous

| Data type | | Purpose | Syntax |
|---|---|---|---|
| Double, single, (u)int8, (u)int16, (u)int32, (u) int64, complex | | Numeric arrays, matrix computations, math | `[1,2,3], [1;2;3], uint8(), int16(),…` |
| String | | Text arrays | `"hello world"` |
| Char | | Single characters, character arrays | `'hello'` |
| cellstr | | Cell arrays of characters | `{'hello','world'}` |
| Categorical | | Discrete, nonnumeric data | `categorical()` |
| Datetime | | Absolute dates and timestamps, including time zones | `datetime('July 12, 2001 08:15:01')` |
| Duration | | Elapsed times | `duration(h,m,s), hours(), minutes(),…` |
| Calendar of duration | | Relative time based on calendar | `caldays(), calweeks(),…` |
| Logical | | True/false, test state, identify data by condition | `logical(), ==, ~=, >, >=, <, <=, &, &&, \|, \|\|` |
| Other specialized types | | sparse, enum, custom, … | >> |

### Heterogeneous

| Data type | | Purpose | Syntax |
|---|---|---|---|
| Table | | Mixed-type, column-oriented data (spreadsheet-like). Store metadata. | `table(x,y,z), array2table` |
| Timetable | | Timestamped tabular data | `timetable(t,x,y)`<br>`table2timetable,array2timetable` |
| Structure | | Fields can contain data of any size and type. Ideal for nonrectangular data. | `struct()`<br>`s.Field = 42;` |
| Structure array | | Array of structures (described above) | `s = [s1,s2], s(2).Field = 42;` |
| Cell array | | Each cell in the array can contain any data type, any size | `cell(), {pi,ones(5), "hello"}` |
| Tall array | | MATLAB data types can be made "tall" when data does not fit in memory | `ds = datastore(), T = tall(ds)` |
| Dictionary | | Object that maps unique keys to values | `d= dictionary(keys,values)` |

MathWorks®
*Accelerating the pace of engineering and science*

## Data Selection

Use array indexing to select data.

Linear indexing for 1D arrays:

`X(1)` First element

`X(end)` Last element

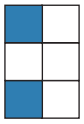
Row, column indexing for multidimensional arrays:

`A(1,2)`

`A(1,1,2)`

Select multiple with a vector:

`A([1,3],1)`



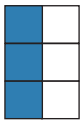
Use colon : to select a range:

`A(1:3,1)`

`A(:,1)` All rows, column 1

`A(1,:)` Row 1, all columns

`A(1:2:end,:)` Every other row



Remove data from array:

`A(1,:) = [];`

## Logical Indexing

Use logical expressions to select data.

Elements of X greater than 7:

`Y = X(X > 7)`

X =

| |
|---|
| 8 |
| 3 |
| 9 |

X =

| |
|---|
| 8 |
| 3 |
| 9 |

Y =

| |
|---|
| 8 |
| 9 |

Combine conditions:

`X(X > 3 && X <= 7)`

`X(X > 3 || X <= 7)`

Elements of S not equal to "hello world":

`S(S ~= "hello world")`

### Multidimensional arrays

Use condition to identify rows or columns:

`idx = X > 7;`

`A(idx, :)`

### Tables and timetables

`T(:,vartype('numeric'))`

`TT(timerange(t1,t2),:)`

## Container Indexing

Using parentheses () for indexing retains the initial data type. Access the underlying data with curly braces {}. Tables and structures also allow you to reference data by name.

**Examples >>**

| Type | Subset | Contents |
|---|---|---|
| Table | Returns a table:<br>`T(1,2)`<br>`T(:,'A')`<br>`T(:,{'A',…`<br>`'B'})` | Returns underlying data:<br>`T{1,2}`<br>`T{:,'A'}`<br>`T.A`<br>`T.Rows`<br>`T.Variables` |
| Timetable | Same as above:<br>`TT('Apr 1,… 2004',5)` | Same as above:<br>`TT.Time` |
| Cell array | Returns a cell:<br>`C(1,2)` | `C{1,2}`<br>`C{:}` -> comma separated list |
| Structure | Returns a struct:<br>`S(1,1)` | `S.Field` |

mathworks.com