

Lists and List Manipulation in Mathematica

Lists are the main data collection element in Mathematica. They can be used as simple collections of data, as tuples or sets.

In Mathematica, list elements are unordered and can be of any types of data. These data can be numeric and string, image, function, plot or any kind of expression.

In some lists, list elements are not related to each other and they are simple collections or sets if they do not include any same element only once.

Lists when used as sets, may be generated after some set builder algorithm. In this case the elements of the list are related to each other since they have build according to the same set builder algorithm.

One dimensional lists are called "vectors".

Vectors can be made, by simply hand addition. Examples :

```
In[226]:= data1 = {"Main05 16 25", 23, 4, 36, 5, 4, 7, 2}
```

```
Out[226]= {Main05 16 25, 23, 4, 36, 5, 4, 7, 2}
```

The variable data1 is a Mathematica list, but it may also be considered as set, since it does not include any same element more then once.

In Mathematica, there is also different prebuilt functions that can be used for generating sets (which are considered as lists by Mathematica).

One of the very simple set builders is is a function finding the divisors of any integer :

```
In[227]:= Divisors [12]
```

```
Out[227]= {1, 2, 3, 4, 6, 12}
```

```
In[228]:= data2 = {1, 2, 3, 4, 6, 12}
```

```
Out[228]= {1, 2, 3, 4, 6, 12}
```

An ordered set is build and stored as variable named ``data2``.

Range [] function construct a list according,

`Range [imax]` generates the list {1, 2, ..., i_{max}}.

`Range[imin, imax]` generates the list {i_{min}, ..., i_{max}}.

`Range[imin, imax, di]` generates the list {i_{min}, ..., i_{max}} with steps di.

```
In[229]:= data3 = Range [3, 20, 2]
```

```
Out[229]= {3, 5, 7, 9, 11, 13, 15, 17, 19}
```

RandomInteger[] and RandomReal [] can also construct sets (lists).

Uasge of these prebuild functions for making a set with pseudorandom elements :

`RandomInteger [{imin, imax}, n]` gives a list of
n pseudorandom integers in the range {i_{min}, i_{max}}.

```
In[230]:= data4 = RandomInteger [ {5, 10}, 10]
```

```
Out[230]= {10, 7, 10, 10, 10, 5, 5, 10, 6, 8}
```

Table [] predefined function is an effective means to devise a set :

`Table[expression, n]` Produce a set including n copies of the given expression.

```
In[231]:= Table [  $\frac{2y + 5x}{5y - 3x}$ , 5]
```

```
Out[231]= {  $\frac{60 + 5x}{150 - 3x}$ ,  $\frac{60 + 5x}{150 - 3x}$ ,  $\frac{60 + 5x}{150 - 3x}$ ,  $\frac{60 + 5x}{150 - 3x}$ ,  $\frac{60 + 5x}{150 - 3x}$  }
```

`Table[expression with an iterator (let us call i), {i, imax}` generates a list of the values of *expr* when *i* runs from 1 to i_{max}.

```
In[232]:= Table[y = 3 x, {x, 10}]
```

```
Out[232]:= {3, 6, 9, 12, 15, 18, 21, 24, 27, 30}
```

`Table` [expression with an iterator (let us call i), $\{i_{\min}, i_{\max}\}$] generates a list of the values of $expr$ when i runs from i_{\min} to i_{\max} .

```
In[233]:= Table[e^x, {x, 3, 7}]
```

```
Out[233]:= {e^3, e^4, e^5, e^6, e^7}
```

```
In[234]:= Table[e^x, {x, 3, 7}] // N
```

```
Out[234]:= {20.0855, 54.5982, 148.413, 403.429, 1096.63}
```

or,

```
In[235]:= N[Table[e^x, {x, 3, 7}]]
```

```
Out[235]:= {20.0855, 54.5982, 148.413, 403.429, 1096.63}
```

`Table` [expression with an iterator (let us call i), $\{i_{\min}, i_{\max}, di\}$] generates a list of the values of $expr$ when i runs from i_{\min} to i_{\max} , in steps of di .

```
In[236]:= N[Table[e^x, {x, 3, 7, 2}]]
```

```
Out[236]:= {20.0855, 148.413, 1096.63}
```

`Table`[$expr, \{i, \{i_1, i_2, \dots\}\}$] uses the successive values i_1, i_2, \dots .

`Table`[

```
In[237]:= Table[2 x, {x, {2, 3, 5, 6}}]
```

```
Out[237]:= {4, 6, 10, 12}
```

Prime numbers can be listed

`Table` [Primes [n], $\{n, i_{\min}, i_{\max}\}$] generates a list of the values of *Prime Numbers* when i runs from i_{\min} to i_{\max} .

```
In[238]:= Table[Prime[n], {n, 5, 25}]
```

```
Out[238]= {11, 13, 17, 19, 23, 29, 31, 37, 41, 43,
          47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97}
```

`Table[Primes[n], {n, i_{\min} , i_{\max} , di }]`
 generates a list of the values of *Prime Numbers*
 when i runs from i_{\min} to i_{\max} in steps of di .

```
In[239]:= Table[Prime[n], {n, 5, 25, 2}]
```

```
Out[239]= {11, 17, 23, 31, 41, 47, 59, 67, 73, 83, 97}
```

An advanced way is using user defined functions. One example is given in "Mathematica Navigator".

A positive integer is a perfect number if it is equal to the sum of its proper divisors. A proper divisor is a divisor that is not the number itself. One example of perfect numbers is number 6.

```
In[240]:= Divisors[6]
```

```
Out[240]= {1, 2, 3, 6}
```

Since the prebuilt `Divisors[n]` function includes the number equal to the number itself, if the sum of the divisors (not proper) is equal to $2n$.

Now, the program given is,

1 stage : testing whether the number is perfect :

```
In[241]:= perfectQ[n_] := Total[Divisors[n]] == 2 n
```

```
In[242]:= perfectQ[6]
```

```
Out[242]= True
```

```
In[243]:= findPerfect[n_] := Select[Range[n], perfectQ[#] &]
```

This means, screen the numbers from 1 to n , select the numbers which make the defined `perfectQ[n_]` function true. the symbol `#` is a placeholder signaled by `&`. Every time n takes replaces `#` and be evaluated. Let us find what are the perfect numbers existing in the range 1 to 100.

```
In[244]:= findPerfect [100]
```

```
Out[244]= {6, 28}
```

in the range [0,10000]

```
In[245]:= findPerfect [10000]
```

```
Out[245]= {6, 28, 496, 8128}
```

We could also use DivisorSigma [] function

```
In[246]:= Select [Range [10000], DivisorSigma [1, #] == 2 # &]
```

```
Out[246]= {6, 28, 496, 8128}
```

Nested lists

Nested lists (list inside a list) may be generated by simple hand input or by using the Table [] function. We will limit ourselves by constructing two elements lists within a list.

```
In[247]:= data5 = {{1, 2}, {3, 4}}
```

```
Out[247]= {{1, 2}, {3, 4}}
```

```
In[248]:= data6 = Table [i + j, {i, 1, 10}, {j, 1, 2}]
```

```
Out[248]= {{2, 3}, {3, 4}, {4, 5}, {5, 6}, {6, 7},
           {7, 8}, {8, 9}, {9, 10}, {10, 11}, {11, 12}}
```

Another Tact,

```
In[249]:= A = {1, 2, 3};
```

```
In[250]:= B = {3, 4, 5};
```

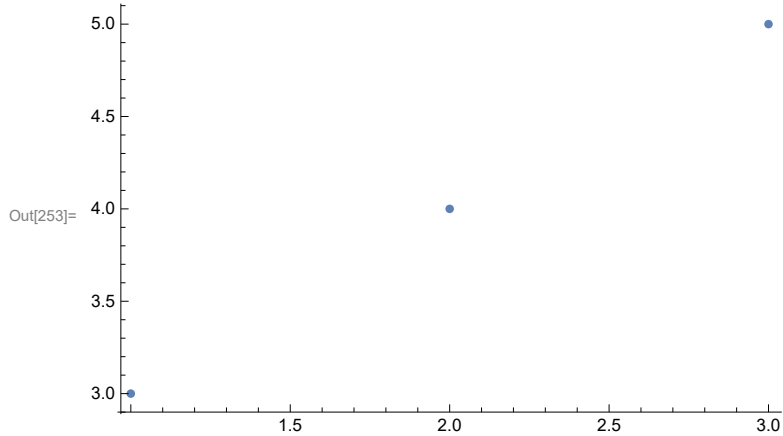
```
In[251]:= If [Length[A] == Length[B], LengthData = Length[B] ]
```

```
Out[251]= 3
```

```
In[252]:= data7 = Table [{A[[x]], B[[x]]}, {x, 1, LengthData}]
```

```
Out[252]= {{1, 3}, {2, 4}, {3, 5}}
```

```
In[253]:= ListPlot [ data7 ]
```



The last way is more flexible and allow to generate a set containing any element pairs at any length.

List Arithmetic

Lists can be added, subtracted, multiplied or divided.

```
In[254]:= data7 + 2
```

```
Out[254]= { {3, 5}, {4, 6}, {5, 7} }
```

```
In[255]:= {1, 2, 3} + {5, 6, 7}
```

```
Out[255]= {6, 8, 10}
```

```
In[256]:= {1, 2, 3} - {5, 6, 7}
```

```
Out[256]= {-4, -4, -4}
```

```
In[257]:= {1, 2, 3} {5, 6, 7}
```

```
Out[257]= {5, 12, 21}
```

```
In[258]:= {1, 2, 3} / {5, 6, 7}
```

```
Out[258]= { 1/5, 1/3, 3/7 }
```

```
In[259]:=  $\epsilon^{\text{data7}}$  // N
```

```
Out[259]= {{2.71828, 20.0855},
           {7.38906, 54.5982}, {20.0855, 148.413}}
```

Tabulating Lists

TableForm

`TableForm[list]` : Makes a table form a list.

```
In[260]:= TableForm[data7]
```

```
Out[260]/TableForm=
  1    3
  2    4
  3    5
```

Table Directions

`TableDirections` rule is an option for `TableForm` and `MatrixForm` which specifies whether successive dimensions should be arranged as rows or columns.

`TableDirections`→`Column` is the default choice and indicates that the first dimension will be the first element of the columns of the formed table.

We can notice from the new table formed from the list `data7` that the first column of the rows of the new formed table are the first elements of the elements of `data7`. We can change this by adding `TableDirections`→`Column` option to the `Row` and the first row of the new formed table will consist from the first elements of the elements of the `data7`.

```
In[261]:= TableForm[data7, TableDirections → Row]
```

```
Out[261]/TableForm=
  1 2 3
  3 4 5
```

This is the transpose of the `data7` table.

```
In[262]:= TableForm[Transpose[data7]]
Out[262]/TableForm=
  1    2    3
  3    4    5
```

Table Spacings

Sets the distance between table elements. Examples from Wolfram Docs
TableSpacing :

```
TableForm[Array[a_&, {3, 2}], TableSpacing -> {0, 0}]
```

Here, $a_{\text{+++}}$ is the place holder of table elements & indicates this and {3,2} specifies 3 rows, 2 columns. Let us see his action :

```
In[263]:= TableForm[Array[a_&, {3, 2}], TableSpacing -> {0, 0}]
Out[263]/TableForm=
  a1,1 a1,2
  a2,1 a2,2
  a3,1 a3,2
```

```
In[264]:= TableForm[Array[a_&, {3, 2}], TableSpacing -> {2, 4}]
Out[264]/TableForm=
  a1,1      a1,2
  a2,1      a2,2
  a3,1      a3,2
```

```
In[265]:= TableForm[data7, TableSpacing -> {2, 4}]
Out[265]/TableForm=
  1      3
  2      4
  3      5
```

TableAlignments

TableAlignments $\{a_1, a_2, \dots\}$ specifies alignments for successive dimensions.

For dimensions that are given as columns, possible alignments are Left,

Center and Right, For dimensions that are given as rows, possible alignments are Bottom, Center and Top.

The default is Automatic which is Bottom for rows and Left for columns.

```
In[266]:= data8 = {{Item, Supplier, Number}, {Pencil, China, 30},
  {Pen, US, 12}, {Eraser, Turkey, 35}}
```

```
Out[266]= {{Item, Supplier, Number}, {Pencil, China, 30},
  {Pen, US, 12}, {Eraser, Turkey, 35}}
```

```
In[267]:= TableForm[data8, TableSpacing -> {2, 2}]
```

Out[267]/TableForm=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[268]:= TableForm[data8, TableSpacing -> {2, 2},
  TableAlignments -> Center]
```

Out[268]/TableForm=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

{Vertical,Horizontal} alignments :

```
In[269]:= TableForm[data8, TableSpacing -> {2, 2},
  TableAlignments -> {Left, Top}]
```

Out[269]/TableForm=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Table Headings

Arrange labels to be printed for entries in each dimension of a table or matrix.

```
In[270]:= data9 = {{Pencil, China, 30},
                 {Pen, US, 12}, {Eraser, Turkey, 35}}
```

```
Out[270]= {{Pencil, China, 30}, {Pen, US, 12}, {Eraser, Turkey, 35}}
```

```
In[271]:= TableForm[data9, TableSpacing → {2, 2},
                    TableAlignments → {Left, Top}, TableHeadings → Automatic]
```

```
Out[271]/TableForm=
```

	1	2	3
1	Pencil	China	30
2	Pen	US	12
3	Eraser	Turkey	35

```
In[272]:= TableForm[data9, TableSpacing → {2, 2},
                    TableAlignments → {Left, Top}, TableHeadings →
                    {{1, 2, 3}, {Item, Country of Origin, Number}}]
```

```
Out[272]/TableForm=
```

	Item	Country of Origin	Number
1	Pencil	China	30
2	Pen	US	12
3	Eraser	Turkey	35

```
In[273]:= TableForm[data9, TableSpacing → {2, 2},
                    TableAlignments → {Left, Top},
                    TableHeadings → {None, {Item, Country of Origin, Number}}]
```

```
Out[273]/TableForm=
```

	Item	Country of Origin	Number
	Pencil	China	30
	Pen	US	12
	Eraser	Turkey	35

Column and Row

Column has the same options as Grid, with the following exceptions. For Grid, the default value of Alignment is {Center, Baseline}, but for Column the default value is {Left, Baseline}. Thus, the elements in the column are, by default, aligned left. In addition, Column also has the special option ColumnAlignments, with default value Left. However, usually we do not need either Alignment or ColumnAlignments because we can simply add to the command a second argument that indicates the alignment. The alignment can be Left, Center, Right, and "c" (where c is a character).

Column[given list] makes a column from a given list

```
In[274]:= Column[data8]
          {Item, Supplier, Number}
Out[274]= {Pencil, China, 30}
          {Pen, US, 12}
          {Eraser, Turkey, 35}
```

```
In[275]:= TableForm[Column[data8]]
Out[275]/TableForm=
          {Item, Supplier, Number}
          {Pencil, China, 30}
          {Pen, US, 12}
          {Eraser, Turkey, 35}
```

```
In[276]:=
```

Grid

Tabular representation of data has proved to be an efficient way to transmit information. Mathematica has good tools for preparing tables. TableForm is familiar from earlier versions of Mathematica, whereas Grid, Column, and Row are new in version 6. Grid is versatile and powerful for even complex tables, giving the possibility of detailed formatting.

Grid [] function, is another way to generate tables from a data list.

Usage :

`Grid [{ {expr11, expr12, ... } , {expr21, expr22, ... } , ... }]`

In[277]= `Grid [{ {1, 2, 3} , {11, 12, 13} }]`

Out[277]=

1	2	3
11	12	13

In[278]= `Grid [{ {1, 2, 3} , {11, 12, 13} }] // Text`

Out[278]=

1	2	3
11	12	13

In[279]= `Grid [data8] // Text`

Out[279]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Grid Options

General forms of declaring Grid options,

value this option will be applied to all columns.

valueAll

{valuecol} only for columns

{valuecol , valuerow} apply valuecol for column and valuerow for rows

Values valuecol and valuerow can be a single value or a list of values of the following general form :

{ *a*₁, ..., *a*_{*k*}, { *b*₁, ..., *b*₁ }, *c*₁, ..., *c*_{*m*} } Apply *a*₁, ..., *a*_{*k*} at the beginning, *c*₁, ..., *c*_{*m*} at the end, and cyclically the sequence *b*₁, ..., *b*₁ in between

- Some typical special cases are as follows :

$\{a_1, \dots, a_k, \{ \}, c_1, \dots, c_m\}$ Apply the default value in between
 $\{a_1, \dots, a_k, \{b_1, \dots, b_1\}\}$ Apply cyclically b_1, \dots, b_1 through the end
 $\{a_1, \dots, a_k\}$ Apply a_1, \dots, a_k at the beginning and then the default value
 $\{a_1\}$ Apply a_1 at the beginning and then the default value
 $\{\{b_1, \dots, b_1\}\}$ Apply cyclically b_1, \dots, b_1
 $\{\{b_1, b_2\}\}$ Apply cyclically b_1 and b_2 -that is, alternate between b_1 and b_2
 $\{\{b_1\}\}$ or b_1 Apply repeatedly b_1

```
In[280]:= w = { {Range[7]}, {Range[8, 14]} }
```

```
Out[280]:= { { {1, 2, 3, 4, 5, 6, 7} }, { {8, 9, 10, 11, 12, 13, 14} } }
```

Alignment

Horizontal and vertical alignment of items. Default for Rows {Baseline}, for columns {Centered}. Other options, {Left}, {Right},{.} (Columns are aligned at the decimal point). In horizontal (columnar) arrangement Left, Right, Center and "c" (where c is any character) , Vertical (Row wise) alignment Bottom, Center, Baseline and Top.

Alignment → {Left, Baseline} is a {valuecol , valuerow} type input. Column will be aligned to the left and rows will be aligned to the baseline.

```
In[281]:= Grid[ data8, Alignment → {Left, Baseline} ] // Text
```

```

Item    Supplier  Number
Pencil  China      30
Pen      US         12
Eraser  Turkey     35

```

```
Out[281]=
```

Dividers

Drawing lines between rows and columns. Default None.

Dividers→{None,All} is a {valuecol , valuerow} type input. All rows will have dividers columns will not. This will result as :

```
In[282]:= Grid [data8,
  Alignment → {Left, Baseline},
  Dividers → {None, All}
] // Text
```

```
Out[282]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Dividers→{All,None} is a {valuecol , valuerow} type input. All columns will have dividers rows not.

```
In[283]:= Grid [data8,
  Alignment → {Left, Baseline},
  Dividers → {All, None}
] // Text
```

```
Out[283]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Alignment→ {Left, Baseline} is a {valuecol , valuerow} type input. All columns and rows will have dividers.

```
In[284]:= Grid[data8,
  Alignment → {Left, Baseline},
  Dividers → {All, All}
] // Text
```

Out[284]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Dividers→All is a valueAll type of input. All columns and rows will have dividers.

```
In[285]:= Grid[data8,
  Alignment → {Left, Baseline},
  Dividers → All
] // Text
```

Out[285]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Dividers→{2→ True} is a {valuecol} type of input. That will only affect columns and trace a divider after 2nd. column :

```
In[286]:= Grid[data8,
  Alignment → {Left, Baseline},
  Dividers → {2 → True}
] // Text
```

```
Out[286]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Dividers→{None,2→ True} is a {valuecol, valuerow} type of input. Columns won't have dividers but one divider will be drawn after 2nd. row.

```
In[287]:= Grid[data8,
  Alignment → {Left, Baseline},
  Dividers → {None, 2 → True}
] // Text
```

```
Out[287]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Dividers→{2→ True , 2→ True} is a {valuecol, valuerow} type of input. One divider will be drawn after 2nd. column, the other at 2nd. row.


```
In[288]:= Grid[data8,
  Alignment → {Left, Baseline},
  Dividers → {2 → True, 2 → True}
] // Text
```

```
Out[288]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Spacings

Spacings is distance in units of current font size, between rows and columns. Automatic , usually means {0.7, 0.4}

```
In[289]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → Automatic
] // Text
```

```
Out[289]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[290]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1}
] // Text
```

```
Out[290]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

We can see the effect of spacing rule.

ItemStyle

One of the exciting options. ItemStyle arrange the embellishments of the row and columns. Examples illustrates the usage of this option.

```
In[291]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → Red
] // Text
```

Item	Supplier	Number
-------------	-----------------	---------------

Pencil	China	30
---------------	--------------	-----------

Out[291]=

Pen	US	12
------------	-----------	-----------

Eraser	Turkey	35
---------------	---------------	-----------

```
In[292]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → Red
] // Text
```

Item	Supplier	Number
-------------	-----------------	---------------

Pencil	China	30
---------------	--------------	-----------

Out[292]=

Pen	US	12
------------	-----------	-----------

Eraser	Turkey	35
---------------	---------------	-----------

```
In[293]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {{None}, {Directive[Blue]}}
] // Text
```

Item	Supplier	Number
------	----------	--------

Pencil	China	30
--------	-------	----

Out[293]=

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

```
In[294]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {None, 1 → Blue}
] // Text
```

Item	Supplier	Number
------	----------	--------

Pencil	China	30
--------	-------	----

Out[294]=

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

```
In[295]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → Directive[Blue, Bold, 12]
] // Text
```

Item	Supplier	Number
------	----------	--------

Pencil	China	30
--------	-------	----

Out[295]=

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

```
In[296]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle →
    {Automatic, {2 → {Directive[Red, Bold, 12, Italic]}}}
] //
Text
```

Item	Supplier	Number
------	----------	--------

<i>Pencil</i>	<i>China</i>	<i>30</i>
---------------	--------------	-----------

Out[296]=

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

```
In[297]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {Automatic, 1 → Bold}
] // Text
```

Item	Supplier	Number
-------------	-----------------	---------------

Pencil	China	30
--------	-------	----

```
Out[297]=
```

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

```
In[298]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {1 → Bold, 1 → Bold}
] // Text
```

Item	Supplier	Number
-------------	-----------------	---------------

Pencil	China	30
---------------	-------	----

```
Out[298]=
```

Pen	US	12
------------	----	----

Eraser	Turkey	35
---------------	--------	----

```
In[299]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {1 → Blue, 1 → Blue}
] // Text
```

Item	Supplier	Number
------	----------	--------

Pencil	China	30
--------	-------	----

Out[299]=

Pen	US	12
-----	----	----

Eraser	Turkey	35
--------	--------	----

Background

Colors of the background.

```
In[300]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  ItemStyle → {1 → Bold, Automatic},
  Background → GrayLevel[0.7, 0.3]
] // Text
```

Item	Supplier	Number
------	----------	--------

Pencil	China	30
---------------	-------	----

Out[300]=

Pen	US	12
------------	----	----

Eraser	Turkey	35
---------------	--------	----

GrayLevel[0.7, 0.3] means, gray at level 0.7 with opacity 0.3.

```
In[301]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, LightBlue}
] // Text
```

```
Out[301]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Rows with alternating background :

```
In[302]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background →
    {Automatic, {{White, Lighter[LightMagenta]}}}
] //
Text
```

```
Out[302]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[303]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic,
    {{Lighter[LightBlue], Lighter[LightMagenta]}}}
] //
Text
```

Out[303]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[304]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {Red}}
] // Text
```

Out[304]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35


```
In[305]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background →
    {Automatic, {1 → LightPink, 2 → LightBlue}}
] //
Text
```

```
Out[305]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

Frame

```
In[306]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → True
] // Text
```

```
Out[306]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[307]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → All
] // Text
```

Out[307]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[308]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {All, False}
] // Text
```

Out[308]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[309]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {False, All}
] // Text
```

Out[309]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

FrameStyle

```
In[310]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Dashed
] // Text
```

Out[310]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[311]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]]
] // Text
```

Out[311]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

ItemSize

Width and height of each item; examples of values: Automatic (separately size items to fit within the total formatting width; long items may take several lines), All (make all items the same width and height), Full (allow each item its full width and height; long items are not divided into several lines), w (give all items width w, measured in ems), {w, h} (give all items width w and height h, with h measured in line heights)

```
In[312]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → Automatic
] // Text
```

Out[312]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[313]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → All
] // Text
```

Out[313]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[314]:= Grid[data8,  
  Alignment → {Left, Baseline},  
  Spacings → {1.1, 1.1},  
  Background → {Automatic, {1 → LightPink}},  
  Frame → {None, All},  
  FrameStyle → Directive[Red, Dashed, Thickness[2]],  
  ItemSize → Full  
] // Text
```

Out[314]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[315]:= Grid[data8,  
  Alignment → {Left, Baseline},  
  Spacings → {1.1, 1.1},  
  Background → {Automatic, {1 → LightPink}},  
  Frame → {None, All},  
  FrameStyle → Directive[Red, Dashed, Thickness[2]],  
  ItemSize → {Scaled[0.3], Scaled[0.07]}  
] // Text
```

```
Out[315]=
```

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[316]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → {10, 2}
] // Text
```

Out[316]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

BaseLine

Base style specifications for the grid; examples of values : {}, Blue (all items are blue), Directive[Red, Bold, Italic] (all items are red, bold, and italic)


```
In[317]:= Grid[data8,  
  Alignment → {Left, Baseline},  
  Spacings → {1.1, 1.1},  
  Background → {Automatic, {1 → LightPink}},  
  Frame → {None, All},  
  FrameStyle → Directive[Red, Dashed, Thickness[2]],  
  ItemSize → {10, 2},  
  BaseStyle → {}  
] // Text
```

Out[317]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[318]:= Grid[data8,  
  Alignment → {Left, Baseline},  
  Spacings → {1.1, 1.1},  
  Background → {Automatic, {1 → LightPink}},  
  Frame → {None, All},  
  FrameStyle → Directive[Red, Dashed, Thickness[2]],  
  ItemSize → {10, 2},  
  BaseStyle → Blue  
] // Text
```

Out[318]=

Item	Supplier	Number
Pencil	China	30
Pen	US	12
Eraser	Turkey	35

```
In[319]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → {10, 2},
  BaseStyle → Directive[Red, Bold, Italic]
] // Text
```

Out[319]=

<i>Item</i>	<i>Supplier</i>	<i>Number</i>
<i>Pencil</i>	<i>China</i>	<i>30</i>
<i>Pen</i>	<i>US</i>	<i>12</i>
<i>Eraser</i>	<i>Turkey</i>	<i>35</i>

BaselinePosition

How the grid should be positioned inside text; examples of values: Automatic (means Axis),

Axis (axis of the middle row in the grid),

Baseline (baseline of the middle row in the grid),

Bottom (bottom of the whole grid),

Center (halfway from top to bottom),

Top (top of the whole grid).

```

In[320]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → {10, 2},
  BaseStyle → Directive[Red, Bold, Italic],
  BaselinePosition → Center
] // Text

```

Out[320]=

<i>Item</i>	<i>Supplier</i>	<i>Number</i>
<i>Pencil</i>	<i>China</i>	<i>30</i>
<i>Pen</i>	<i>US</i>	<i>12</i>
<i>Eraser</i>	<i>Turkey</i>	<i>35</i>

```
In[321]:= Grid[data8,  
  Alignment → {Left, Baseline},  
  Spacings → {1.1, 1.1},  
  Background → {Automatic, {1 → LightPink}},  
  Frame → {None, All},  
  FrameStyle → Directive[Red, Dashed, Thickness[2]],  
  ItemSize → {10, 2},  
  BaseStyle → Directive[Red, Bold, Italic],  
  BaselinePosition → Top  
] // Text
```

Out[321]=

<i>Item</i>	<i>Supplier</i>	<i>Number</i>
<i>Pencil</i>	<i>China</i>	<i>30</i>
<i>Pen</i>	<i>US</i>	<i>12</i>
<i>Eraser</i>	<i>Turkey</i>	<i>35</i>

```

In[322]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → {10, 2},
  BaseStyle → Directive[Red, Bold, Italic],
  BaselinePosition → Bottom
] // Text

```

<i>Item</i>	<i>Supplier</i>	<i>Number</i>
<i>Pencil</i>	<i>China</i>	<i>30</i>
<i>Pen</i>	<i>US</i>	<i>12</i>
<i>Eraser</i>	<i>Turkey</i>	<i>35</i>

Out[322]=

```
In[323]:= Grid[data8,
  Alignment → {Left, Baseline},
  Spacings → {1.1, 1.1},
  Background → {Automatic, {1 → LightPink}},
  Frame → {None, All},
  FrameStyle → Directive[Red, Dashed, Thickness[2]],
  ItemSize → {10, 2},
  BaseStyle → Directive[Red, Bold, Italic],
  BaselinePosition → Baseline
] // Text
```

```
Out[323]=
```

<i>Item</i>	<i>Supplier</i>	<i>Number</i>
<i>Pencil</i>	<i>China</i>	<i>30</i>
<i>Pen</i>	<i>US</i>	<i>12</i>
<i>Eraser</i>	<i>Turkey</i>	<i>35</i>

Examples

Example 1

```
In[324]:= v = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 1, 11, 12}};
```

```
In[325]:= Row[ {Grid[v, Frame → All],
  Grid[v, Frame → {All}],
  Grid[v, Frame → {{All}}],
  Grid[v, Frame → {{{All}}]}], ", "
```

```
Out[325]=
```

1	2	3	4
5	6	7	8
9	1	11	12

,

1	2	3	4
5	6	7	8
9	1	11	12

,

1	2	3	4
5	6	7	8
9	1	11	12

,

1	2	3	4
5	6	7	8
9	1	11	12

```
In[326]:= Grid[v, Frame → {All, None}]
```

```
Out[326]=
```

1	2	3	4
5	6	7	8
9	1	11	12

No need to complicate things :

```
In[327]:= Grid[v, Frame → {1 → True, None}]
```

```
Out[327]=
```

1	2	3	4
5	6	7	8
9	1	11	12

```
In[328]:= Grid[v, Frame → All]
```

```
Out[328]=
```

1	2	3	4
5	6	7	8
9	1	11	12

```
Out[ ]=
```

1	2	3	4
5	6	7	8
9	1	11	12

Now for rows :

```
In[329]:= Grid[v, Frame → {1 → True}]
```

```
Out[329]=
```

1	2	3	4
5	6	7	8
9	1	11	12


```
In[330]:= Grid[v, Frame → {None, 1 → True}]
```

```
Out[330]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

```
In[331]:= Grid[v, Frame → {None, All}]
```

```
Out[331]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Exceptional Values

If we have some exceptional formatting for some columns, rows, or items, we can add definitions for these cases as rules.

Exceptional values can be given as rules (a single rule or a list of rules) :

{rulesj} Apply rulesj for specific columns

{rulesj, rulesi} Apply rulesj for specific columns and rulesi for specific rows

{valc, rulesj} Apply valc for columns but rulesj to specific columns

{{valc, rulesj}, {valr, rulesi}} Apply valc for columns and valr for rows but rulesj to

specific columns and rulesi to specific rows

{valc, valr, rulesij} Apply valc and valr for columns and rows but rulesij for specific items

(not applicable for Dividers and Spacings)

Here, rulesj may be, for example, 4 → True, meaning that the value True is used for the fourth column.

Similarly, rulesi may be, for example, 3 → Green, meaning that the value Green is used for the third row.

Also, rulesij may be, for example, {2, 5} → Red, meaning that the value Red is used for the item at position (2, 5).

In addition, rulesij may define a range of items. An example is {{2, 4}, {1, 3}} → Red, meaning that the value Red is used for items with row index in {2, 4} and column index in {1, 3}.

More for frames,

Targeting only one framed column :

```
In[332]:= Grid[v, Frame → {{1 → True}, None}]
```

```
Out[332]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Targeting two framed columns : (Always be specific !)

```
In[333]:= Grid[v, Frame → {{1 → True, 3 → True}, None}]
```

```
Out[333]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Targeting a column and a row :

```
In[334]:= Grid[v, Frame → {{1 → True}, {3 → True}}]
```

```
Out[334]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Targeting two columns and one row :

```
In[335]:= Grid[v, Frame → {{1 → True, 3 → True}, {3 → True}}]
```

```
Out[335]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Targeting a single one cell :

```
In[336]:= Grid[v, Frame → {None, None, {3, 3} → True}]
```

```
Out[336]= 

|   |   |    |    |
|---|---|----|----|
| 1 | 2 | 3  | 4  |
| 5 | 6 | 7  | 8  |
| 9 | 1 | 11 | 12 |


```

Targeting two cells :

```
In[337]:= Grid[v,
  Frame → {None, None, {{2, 3} → True, {2, 4} → True}}]
1 2 3 4
Out[337]= 5 6 

|   |   |
|---|---|
| 7 | 8 |
|---|---|


9 1 11 12
```

Targeting a {{column cell},{row cell}} region : {valcol, valrow, rulesij}

```
In[338]:= Grid[v, Frame → {None, None, {{2, 3}, {2, 4}} → True}]
1 2 3 4
Out[338]= 5 

|   |   |    |    |
|---|---|----|----|
| 6 | 7 | 8  |    |
| 9 | 1 | 11 | 12 |


```

Row indices 2 to 3 , column indices 2 to 4.

The grid may also contain text and graphics :

```
In[339]:= t1 = "Here is the cumulative distribution
function (CDF) of the
standard normal distribution with mean
0 and standard deviation 1.\nThe CDF is  $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$ .";
```

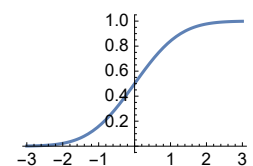
```
t2 =
```

```
"Here is the probability density function (PDF) of the
standard normal distribution with mean 0 and
standard deviation 1.\nThe PDF is  $\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ .";
```

```
g1 = Plot[CDF[NormalDistribution[0, 1], x],
{x, -3, 3}, ImageSize -> 120];
g2 = Plot[PDF[NormalDistribution[0, 1], x],
{x, -3, 3}, ImageSize -> 120];
Grid[{{t1, g1}, {t2, g2}}, Alignment -> {Left, Center},
ItemSize -> {{30, 30}}] // Text
```

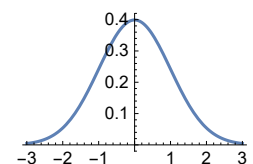
Here is the cumulative distribution
function (CDF) of the
standard normal distribution with mean
0 and standard deviation 1.

The CDF is $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$.



Here is the probability density function (PDF) of the
standard normal distribution with mean 0 and
standard deviation 1.

The PDF is $\frac{1}{\sqrt{2\pi}} e^{-x^2/2}$.



Options of the Grid

Row and Column Sums

A new matrix,

```
In[344]:= m = { {6, 21.2, 3.05, 64.2},
               {34, 9.582, 143.17, 8.702}, {985, 0.6914, 70.4, 126.6} };
```

```
In[345]:= colsums = Total[m]
```

```
Out[345]= {1025, 31.4734, 216.62, 199.502}
```

```
In[346]:= rowsums = Total /@ m
```

```
Out[346]= {94.45, 195.454, 1182.69}
```

```
In[347]:= totsum = Total[rowsums]
```

```
Out[347]= 1472.6
```

Then the table :

```
In[348]:= firstRow =
```

```
{ {"Rows", "Col 1", "Col 2", "Col 3", "Col 4", "Sums" } }
```

```
Out[348]= { {Rows, Col 1, Col 2, Col 3, Col 4, Sums} }
```

```
In[349]:= mainRows =
```

```
Join[ { {"Row 1", "Row 2", "Row 3" } }T, m, {rowsums}T, 2 ]
```

```
Out[349]= { {Row 1, 6, 21.2, 3.05, 64.2, 94.45},
            {Row 2, 34, 9.582, 143.17, 8.702, 195.454},
            {Row 3, 985, 0.6914, 70.4, 126.6, 1182.69} }
```

```
In[350]:= lastRow = {Join[ {"Sums" }, colsums, {totsum} ] }
```

```
Out[350]= { {Sums, 1025, 31.4734, 216.62, 199.502, 1472.6} }
```

```
In[351]:= t = Join[firstRow, mainRows, lastRow]
```

```
Out[351]= {{Rows, Col 1, Col 2, Col 3, Col 4, Sums},
           {Row 1, 6, 21.2, 3.05, 64.2, 94.45},
           {Row 2, 34, 9.582, 143.17, 8.702, 195.454},
           {Row 3, 985, 0.6914, 70.4, 126.6, 1182.69},
           {Sums, 1025, 31.4734, 216.62, 199.502, 1472.6}}
```

```
In[352]:= Grid[t]
```

```
Out[352]=
  Rows  Col 1  Col 2  Col 3  Col 4  Sums
  Row 1   6    21.2  3.05  64.2  94.45
  Row 2  34    9.582 143.17 8.702 195.454
  Row 3 985    0.6914 70.4  126.6 1182.69
  Sums 1025   31.4734 216.62 199.502 1472.6
```

Here the transpose may be written as Esc + tr + Esc

Enhancing the Grid

Alignment

** Alignment Horizontal and vertical alignment of items; examples of values: {Center, Baseline} (columns are centered, rows are at baseline), Left (columns are aligned left), Right (columns are aligned right), "." (columns are aligned at the decimal point). In horizontal (or column) alignment, we can use Left, Center, Right, and "c" (where c is a character). In vertical (or row) alignment, we can use Bottom, Center, Baseline, and Top.

As an example, we consider the table t we formed previously. First, we align the columns at the right:

```
In[353]:= Grid[t, Alignment → Right] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

```
Out[353]=
```

Right alignment is good for integers. Then we align the columns at the decimal point :

```
In[354]:= Grid[t, Alignment → "."] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

```
Out[354]=
```

Columns with integers are seemingly untouched, but column with a decimal point are aligned with the decimal point. In the reality integers are considered as having a decimal zero and aligned with it. Note that the column headers are aligned such that they end at the position of the decimal point, and this causes the headers to be too far to the left. For the row headers, the right alignment is not good.

Now we align the first column left, the second column right, and the rest of the columns at the decimal point:

```
In[355]:= Grid[t, Alignment → {{Left, Right, {"."}}}] // Text
      Rows  Col 1  Col 2      Col 3    Col 4    Sums
      Row 1    6    21.2      3.05    64.2    94.45
Out[355]= Row 2   34    9.582    143.17   8.702   195.454
      Row 3  985    0.6914   70.4    126.6   1182.69
      Sums 1025   31.4734  216.62  199.502 1472.6
```

Note that here the value of the option is of the form `{valcol}` mentioned previously in a box. This form of the option means that `valcol` is applied for columns. Here, `valcol` is `{Left, Right, {"."}}` and this is of the general form `{a1, a2, {b1}}` mentioned before. This form of the option means that `a1` is applied for the first column, `a2` for the second column, and `b1` for the rest of the columns.

The previous table is quite good. However, the column headers should be aligned better. We define these exceptional alignments as rules:

```
In[356]:= Grid[t, Alignment → {{Left, Right, {"."}},
      Baseline, {{1, 1}, {2, 6}} → Center}] // Text
      Rows  Col 1    Col 2    Col 3    Col 4    Sums
      Row 1    6    21.2    3.05    64.2    94.45
Out[356]= Row 2   34    9.582   143.17   8.702   195.454
      Row 3  985    0.6914   70.4    126.6   1182.69
      Sums 1025   31.4734  216.62  199.502 1472.6
```

Here, the value of the option is of the form `{valcol, valrow, rulesij}` mentioned previously in a box. This form of the option means that `valcol` is applied for columns, `valrow` for rows, and `rulesij` for special items. Actually, `valrow` or `Baseline` is the default for rows but we have to define something for the rows so that we can define rules for single items. The rule `{{1, 1}, {2, 6}} → Cen-`

ter defines that the column alignment should be Center for items with row indices in the range {1, 1} and column indices in the range {2, 6}.

Dividers

Dividers Where to draw lines; examples of values : None, All (all items become boxed), Center (all interior dividers), {None, All} (no column lines, all row lines, lines also before the first row and after the last row), {2 → True} (a line after the first column), {None, 2 → True} (a line below the first row), {2 → True, 2 → True} (a line after the first column and below the first row)

Note that if we give Dividers a list of values, the first value corresponds with the divider before the first column or above the first row. The last value that can be given concerns the divider after the last column or below the last row.

First, we draw all dividers:

```
In[357]:= Grid[t, Dividers → All, Alignment → {{Left, Right, {"."}},
             Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Then we draw all interior dividers:

```
In[358]:= Grid[t, Dividers → Center,
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we ask for no column dividers and all row dividers:

```
In[359]:= Grid[t, Dividers → {None, All},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we ask for no row dividers and all column dividers:

```
In[360]:= Grid[t, Dividers → {All, None},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we add, for both the columns and the rows, the second divider:

```
In[361]:= Grid[t, Dividers → {2 → True, 2 → True},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Now we add, for columns, the second divider and the next-to-last divider:

```
In[362]:= Grid[t, Dividers → {{2 → True, -2 → True}, None},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Spacings

Spacings Space between columns and rows (in units of the current font size); examples of values : Automatic (usually means {0.7, 0.4} : the space between columns is 0.7 and that between rows is 0.4), 1.2 (the space between columns is 1.2 and the space between rows is the default 0.4)

Note that if we give Spacings a list of values, the first value corresponds with the space before the first column or above the first row, and the last value that can be given concerns the space after the last column or below the last row. However, the first or last value does not have any effect unless there is a frame line or divider line before the first column or row or after the last column or row.

We make the space between columns 2 (the space between rows is the default 0.4):

```
In[363]:= Grid[t, Spacings → 2,
  Dividers → {None, None},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Now make the space between columns 2 and the space between rows 0.1:

```
In[364]:= Grid[t, Spacings → {2, 0.1},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we fine|tune the spacings:

```
In[365]:= Grid[t, Spacings → {1.5, {0, 1, {0.5}}, 1, 0},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Out[365]= Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Here, the space between columns is 1.5. The space below the first row and above the last row is 1; for the rows in between, we use the space of 0.5.

Note that Mathematica assumes that the spacings begin above the first row and extend below the last row. Thus, to get the space 1 after the first row and above the last row, we have to define a space also above the first row and below the last row. We used the value 0 for these spaces, but in this example these values do not have any effect.

Note that Mathematica assumes that the spacings begin above the first row and extend below the last row. Thus, to get the space 1 after the first row and above the last row, we have to define a space also above the first row and below the last row. We used the value 0 for these spaces, but in this example these values do not have any effect.

If we have a frame or dividers above the first row and below the last row, then the first and last values of Spacings do have an effect because these spacings define how much space should be around the first and last divider. Here, we use the value 1 for these spacings:

```
In[476]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Spacings → {1.5, {1, 1}, {0.5}, 1, 1},
  Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

ItemStyle

ItemStyle Styles of columns and rows; examples of values: None, Blue (all items are blue), {1 → Red} (the first column is red), {Automatic, 1 → Bold} (the first row is bold), {1 → Directive[Red, Bold, 14]} (the first column is red, bold, and size 14), {1 → Bold, 1 → Bold} (the first column and the first row are bold)

```
In[477]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  ItemStyle → {Automatic, 1 → Bold},
  Spacings → {1.5, {1, 1, {0.5}}, 1, 1}},
  Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Then we use the bold style for both the first column and the first row :

```
In[478]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  ItemStyle → {1 → Bold, 1 → Bold},
  Spacings → {1.5, {1, 1, {0.5}}, 1, 1}},
  Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Now the first and last column and row are bold :

```
In[479]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  ItemStyle →
    {{1 → Bold, -1 → Bold}, {1 → Bold, -1 → Bold}},
  Spacings → {1.5, {1, 1, {0.5}}, 1, 1},
  Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we use both the bold style and a color:

```
In[471]:= Grid[t, ItemStyle → {{1 → Directive[Bold, Blue]},
  {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}}, 1, 1}, Dividers →
    {None, {1 → True, 2 → True, -2 → True, -1 → True}},
  Alignment → {{Left, Right, {"."}}, Baseline,
    {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Background

Background Colors of the background; examples of values: None, GrayLevel[0.9] (all items are gray), {Automatic, {{ White, LightGray}}} (columns are default, rows alternate between white

Define a gray background for the table:

```
Grid[t, Background → LightGray,
  ItemStyle → {{1 → Directive[Bold, Blue]}},
  {1 → Directive[Bold, Red]}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers →
  {None, {1 → True, 2 → True, -2 → True, -1 → True}},
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Out[470]=

Next, we define a gray background for the first row (GrayLevel 0.9, Opacity 0.5 :

```
In[480]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic, 1 → GrayLevel[0.9, 0.5]},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Out[480]=

Now we define an alternating background for the rows :

```
In[468]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic,
    {{White, GrayLevel[0.8, 0.5]}}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Define then the first and last rows to be light red :

```

In[467]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic, {LightRed,
    {White, LightGray}}, LightRed}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text

```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we define the background for a range of items or for a single item :

```
In[466]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic, Automatic,
    {{2, 4}, {2, 5}} → LightGray},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

```
{ Automatic, Automatic, {{{5, 5}, {2, 5}} → Darker[Yellow, 0.1], {{2, 4}, {6, 6}} → Darker[Yellow, 0.1], {5, 6} → Red}}
```

Out[466]=

```

In[465]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic, Automatic,
    {{{5, 5}, {2, 5}} → Darker[Yellow, 0.1], {{2, 4},
      {6, 6}} → Darker[Yellow, 0.1], {5, 6} → Red}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text

```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Out[465]=

```
In[464]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Background → {Automatic, Automatic, {5, 6} → Pink},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}, Dividers → {None,
    {1 → True, 2 → True, -2 → True, -1 → True}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Frame

Where to draw frames; examples of values: None, True (frame around the whole grid), All (all items become boxed), {All} (frame around each column), {None, All} (frame around each row)

Add frame for all the table

```

In[463]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{1 → True, 6 → True, 7 → True},
    {1 → True, 2 → True, -2 → True, -1 → True}},
  Frame → True,
  Background → {Automatic, Automatic, {5, 6} → Pink},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}] // Text

```

Out[463]=

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Add a frame for all items (this can also be obtained with Dividers → All) :


```
In[462]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → All,
  Background → {Automatic, Automatic, {5, 6} → Pink},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

```
In[461]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Frame → {None, All},
  Background → {Automatic, Automatic, {5, 6} → Pink},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}] // Text
```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Next, we use both dividers and a frame:

```
In[460]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{2 → True, -2 → True},
    {2 → True, -2 → True}},
  Frame → True,
  Background → {Automatic, Automatic, {5, 6} → Pink},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}] // Text
```

Out[460]=

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Frame for a single Item :

```

In[459]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{1 → True, 2 → True,
    -2 → True, -1 → True},
    {1 → True, 2 → True, -2 → True, -1 → True}},
  Frame → {None, None, {3, 3} → True},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Background → {Automatic, Automatic, {5, 6} → Pink},
  Spacings → {1.5, {1, 1}, {0.5}, 1, 1}}
] // Text

```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

FrameStyle

Style of frames and dividers; examples of values: Automatic, Red, Directive[Gray,Thickness[2]]

We define some styles for frames :

```

In[458]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{1 → True, 2 → True,
    -2 → True, -1 → True},
    {1 → True, 2 → True, -2 → True, -1 → True}},
  Frame → {None, None, {3, 3} → True},
  FrameStyle → Directive[Red, Thickness[2], Dashing[3]],
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Background → {Automatic, Automatic, {5, 6} → Pink},
  Spacings → {1.5, {1, 1}, {0.5}, 1, 1}}
] // Text

```

Out[458]=

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Colored Dividers :

```

In[457]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{Directive[Blue, Thickness[2], Dashing[3]],
    {1 → True, 2 → True, -2 → True, -1 → True}},
    {Directive[Blue, Thickness[2], Dashing[3]],
    {1 → True, 2 → True, -2 → True, -1 → True}}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}
] // Text

```

Out[457]=

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

We can get rid of the dashing and get straight line dividers.

```

In[456]:= Grid[t,
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{Directive[Blue, Thickness[2]],
  {1 → True, 2 → True, -2 → True, -1 → True}},
  {Directive[Blue, Thickness[2]],
  {1 → True, 2 → True, -2 → True, -1 → True}}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
  {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}
] // Text

```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.454
Row 3	985	0.6914	70.4	126.6	1182.69
Sums	1025	31.4734	216.62	199.502	1472.6

Out[456]=

ItemSize

Width and height of each item; examples of values: Automatic (separately size items to fit within the total formatting width; long items may take several lines), All (make all items the same width and height), Full (allow each item its full width and height; long items are not divided into several lines), w (give all items width w, measured in ems), {w, h} (give all items width w and height h, with h measured in line heights; the default value of h is 1)

```

In[455]:= Grid[t,
  ItemSize → All,
  Alignment → {{Left, Right, {"."}},
    Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{Directive[Blue, Thickness[2]],
    {1 → True, 2 → True, -2 → True, -1 → True}},
    {Directive[Blue, Thickness[2]],
    {1 → True, 2 → True, -2 → True, -1 → True}}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}
] // Text

```

Rows	Col 1	Col 2	Col 3	Col 4	Sums
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.474
Row 3	985	0.6914	70.4	126.6	1182.6914
Sums	1025	31.4734	216.62	199.502	1472.6654

Lastly we can add a label,

```
In[493]:= Labeled[Grid[t,
  ItemSize → All,
  Alignment → {{Left, Right, {"."}},
  Baseline, {{1, 1}, {2, 6}} → Center},
  Dividers → {{Directive[Blue, Thickness[2]],
    {1 → True, 2 → True, -2 → True, -1 → True}},
    {Directive[Blue, Thickness[2]],
    {1 → True, 2 → True, -2 → True, -1 → True}}},
  ItemStyle → {{1 → Directive[Bold, Blue]},
    {1 → Directive[Bold, Red]}}},
  Spacings → {1.5, {1, 1, {0.5}, 1, 1}}
],
```

```
Style["\nGeneral Table\n", Red, Bold], Top] // Text
```

General Table

Rows	Col 1	Col 2	Col 3	Col 4	Sum
Row 1	6	21.2	3.05	64.2	94.45
Row 2	34	9.582	143.17	8.702	195.504
Row 3	985	0.6914	70.4	126.6	1182.6914
Sums	1025	31.4734	216.62	199.502	1472.2634

Out[493]=

Tables in Mathematica may be somehow difficult to master. The aim of these examples is to provide some working media which may suit to somebody's needs and use it without hassles of the Mathematica language.

These material is largely based on Mathematica Navigator version 3.1 targeting Mathematica Releases 7 and up. Most material is revised and

reshaped for to be more comprehensive. Mathematica Navigator is perhaps largest and well organised textbook with countless examples, very advisable for learning Mathematica.

I hope that this manuscript will be useful for whom searching a shortest ways to present Mathematica materials.