W3C CSS GRID LAYOUT (Ejemplos de uso)

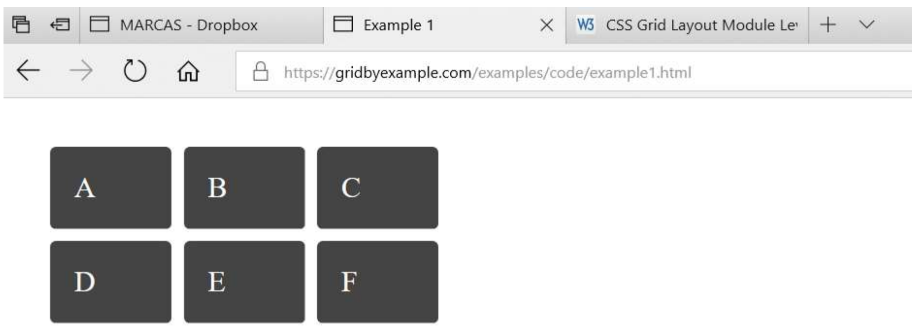
http://www.w3.org/TR/css-grid-1/

1/ Defining a Grid

To define a Grid use display:grid or display:inline-grid on the parent element. You can then create a grid using the grid-template-columns and grid-template-rows properties.

I am using the grid-gap property to create a gap between my columns and rows of 10px. This property is a shorthand for grid-column-gap and grid-row-gap so you can set these values individually.

All direct children of the parent now become grid items and the auto-placement algorithm lays them out, one in each grid cell. Creating extra rows as needed.


MARCAS - Dropbox
Example 1
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example1.html
A
B
C
D
E
F

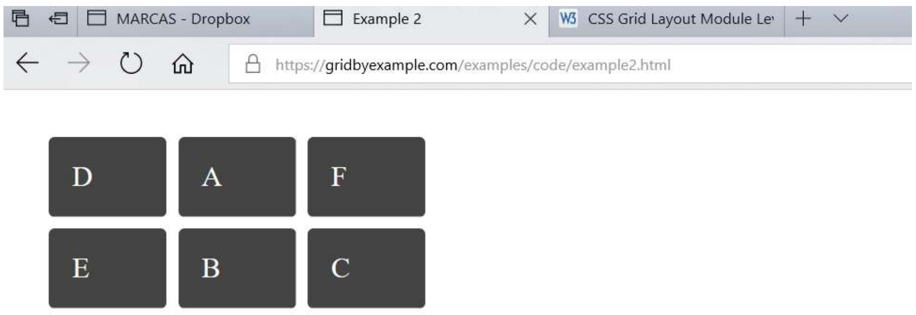
Código:

```
<div class="wrapper">
  <div class="box a">A</div>
  <div class="box b">B</div>
  <div class="box c">C</div>
  <div class="box d">D</div>
  <div class="box e">E</div>
  <div class="box f">F</div>
</div>
body {
  margin: 40px;
}
```

```
.wrapper {
  display: grid;
  grid-template-columns: 100px 100px 100px;
  grid-gap: 10px;
  background-color: #fff;
  color: #444;
}
.box {
  background-color: #444;
  color: #fff;
  border-radius: 5px;
  padding: 20px;
  font-size: 150%;
}
```

2/ Line-based placement

Using the Grid I defined in example 1, I am positioning the elements in my markup (six divs with a class of box and classes from a to f) using line-based placement properties. This example is more verbose than it needs to be as a demonstration of the properties. In reality if an item will only span one grid track you may omit the -end value.


MARCAS - Dropbox
Example 2
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example2.html
D
A
F
E
B
C

Código

HTML: el mismo del ejemplo anterior

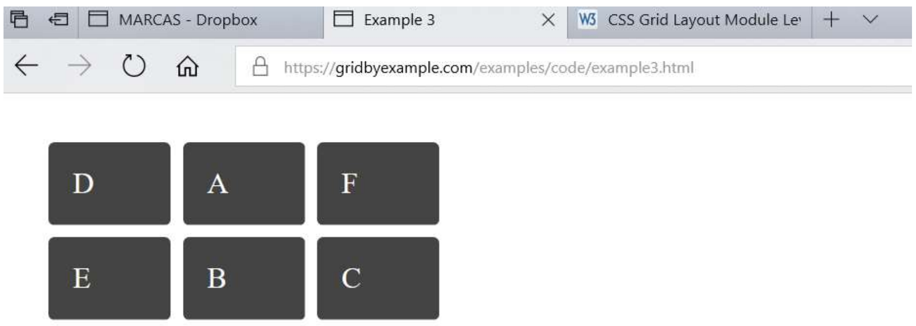
```
body {
  margin: 40px;
}
```

```
.wrapper {
 display: grid;
 grid-template-columns: 100px 100px 100px;
 grid-gap: 10px;
 background-color: #fff;
 color: #444;
}

.box {
 background-color: #444;
 color: #fff;
 border-radius: 5px;
 padding: 20px;
 font-size: 150%;
}

.a {
      grid-column-start: 2;
      grid-column-end: 3;
      grid-row-start: 1;
      grid-row-end: 2;
  }
  .b {
      grid-column-start: 2;
      grid-column-end: 3;
      grid-row-start: 2;
      grid-row-end: 3;
  }
  .c {
      grid-column-start: 3;
      grid-column-end: 4;
      grid-row-start: 2;
      grid-row-end: 3;
  }
  .d {
      grid-column-start: 1;
      grid-column-end: 2;
```

```
    grid-row-start: 1;

    grid-row-end: 2;

  }

  .e {

    grid-column-start: 1;

    grid-column-end: 2;

    grid-row-start: 2;

    grid-row-end: 3;

  }

  .f {

    grid-column-start: 3;

    grid-column-end: 4;

    grid-row-start: 1;

    grid-row-end: 2;

  }
```

3/ Line-based placement shorthand - grid-row and grid-column

We can achieve the same result as in example 2 using a shorthand syntax declaring the start and end values at once. Values are separated with a / and again it would be valid to omit the / and the end value as we are spanning only one track.


MARCAS - Dropbox
Example 3
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example3.html
D
A
F
E
B
C

Código

HTML no cambia

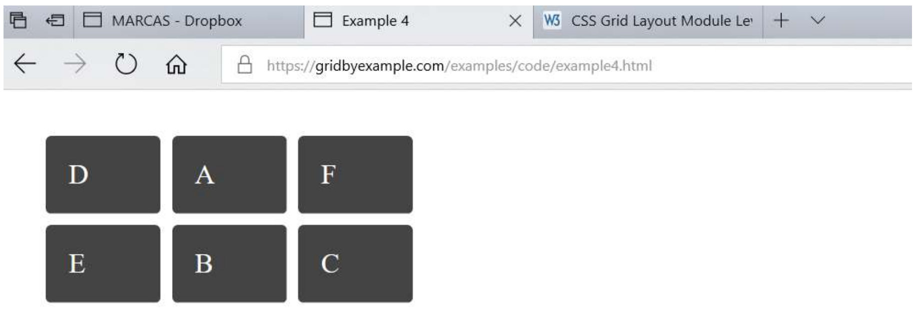
```
body {

  margin: 40px;

}
```

```
.wrapper {
    display: grid;
  grid-gap: 10px;
  grid-template-columns: 100px 100px 100px;
    background-color: #fff;
    color: #444;
  }

  .box {
    background-color: #444;
    color: #fff;
    border-radius: 5px;
    padding: 20px;
    font-size: 150%;

  }

 .a {
  grid-column: 2 / 3;
  grid-row: 1 / 2;
 }
 .b {
  grid-column: 2 / 3;
  grid-row: 2 / 3;
 }
 .c {
  grid-column: 3 / 4;
  grid-row: 2 / 3;
 }
 .d {
  grid-column: 1 / 2;
  grid-row: 1 / 2;
 }
 .e {
  grid-column: 1 / 2;
  grid-row: 2 / 3;
 }
```

```
.f {
  grid-column: 3 / 4;
  grid-row: 1 / 2;
}
```

4/ Line-based placement shorthand - grid-area

We can achieve the same result as in example 2 and 3 declaring all four values with the grid-area property. Again values are separated with a /. The order of the values is row-start/column-start/row-end/column-end.


MARCAS - Dropbox
Example 4
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example4.html
D
A
F
E
B
C

Código

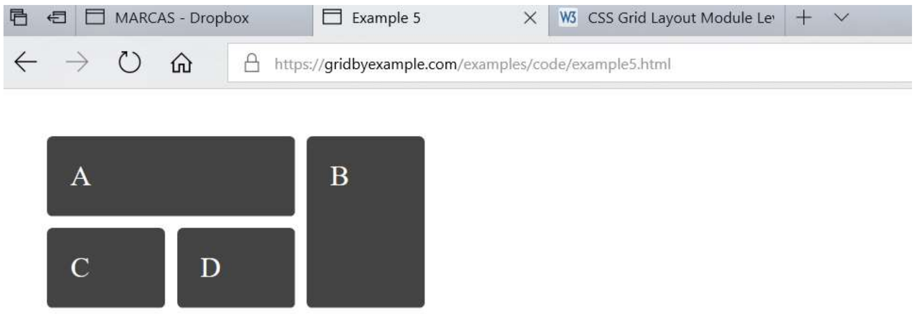
HTML no cambia

```
body {
   margin: 40px;
 }

 .wrapper {
 display: grid;
 grid-gap: 10px;
    grid-template-columns: 100px 100px 100px;
    background-color: #fff;
    color: #444;
 }

 .box {
    background-color: #444;
    color: #fff;
    border-radius: 5px;
    padding: 20px;
    font-size: 150%;

}

.a {
    grid-area: 1 / 2 / 2 / 3;
}
.b {
    grid-area: 2 / 2 / 3 / 3;
}
.c {
    grid-area: 2 / 3 / 3 / 4;
}
.d {
    grid-area: 1 / 1 / 2 / 2;
}
.e {
    grid-area: 2 / 1 / 3 / 2;
}
.f {
    grid-area: 1 / 3 / 2 / 4;
}
```

5/ Line-based placement spanning tracks

To create Grid Areas that are larger than a single grid track we specify an end line that is more than one track away.

Here I am using the grid-column and grid-row shorthand and have omitted the end value for any items that span one row or column track.


MARCAS - Dropbox
Example 5
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example5.html
A
B
C
D

Código

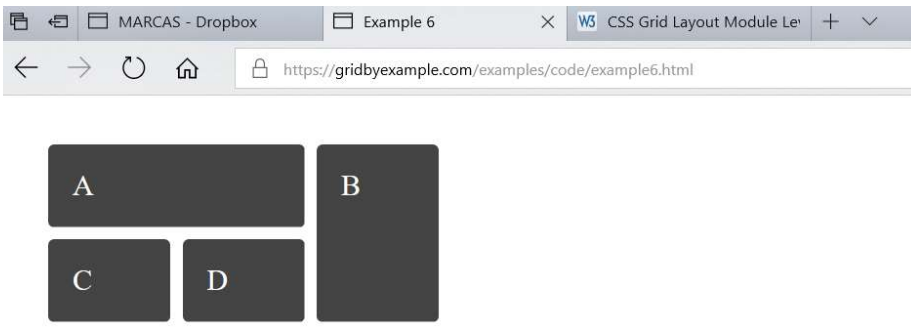
HTML no cambia

```
body {
  margin: 40px;
}

.wrapper {
    display: grid;
    grid-gap: 10px;
        grid-template-columns: 100px 100px 100px;
        background-color: #fff;
        color: #444;
    }

    .box {
        background-color: #444;
        color: #fff;
        border-radius: 5px;
        padding: 20px;
        font-size: 150%;


    }

    .a {
        grid-column: 1 / 3;
        grid-row: 1;
    }
    .b {
```

```
    grid-column: 3 ;

    grid-row: 1 / 3;

  }

  .c {

    grid-column: 1 ;

    grid-row: 2 ;

  }

  .d {

    grid-column: 2;

    grid-row: 2;

  }
```

6/ Line-based placement spanning tracks with the span keyword

We can also span using the span keyword. This example creates the same layout as the one in example 5. I am using the span keyword rather than targeting the Grid Line by number. I am also using the defaults for row and column end, which is to span 1.


MARCAS - Dropbox
Example 6
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example6.html
A
B
C
D

Código

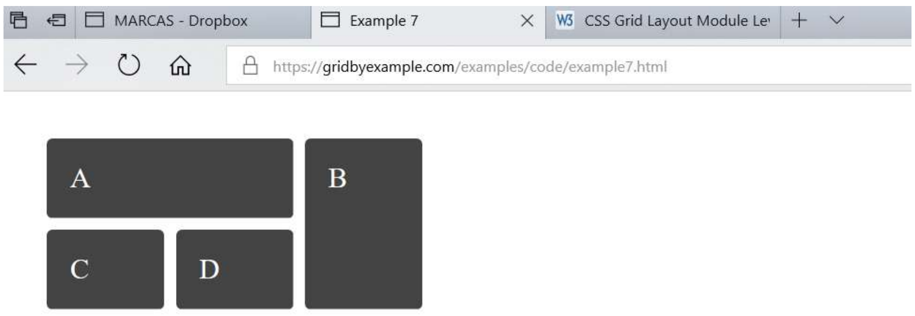
HTML no cambia

```
body {

 margin: 40px;

}


.wrapper {

    display: grid;

  grid-gap: 10px;

    grid-template-columns: 100px 100px 100px;
    background-color: #fff;
    color: #444;
}

.box {
    background-color: #444;
    color: #fff;
    border-radius: 5px;
    padding: 20px;
    font-size: 150%;

}

.a {
    grid-column: 1 / span 2;
}
.b {
    grid-column: 3 ;
    grid-row: 1 / span 2;
}
.c {
    grid-column: 1 ;
    grid-row: 2 ;
}
.d {
    grid-column: 2 ;
    grid-row: 2 ;
}
```

7/ Line-based placement named lines

We can name lines rather than targeting them by number. Name the line inside brackets. In the code below you can see that I name the very first column line col1-start then comes the 100 pixel first column track. Having named the lines you can use the names, rather than numbers.

You always have the line numbers to use - even if you name some or all of your lines.


MARCAS - Dropbox
Example 7
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example7.html
A
B
C
D

Código

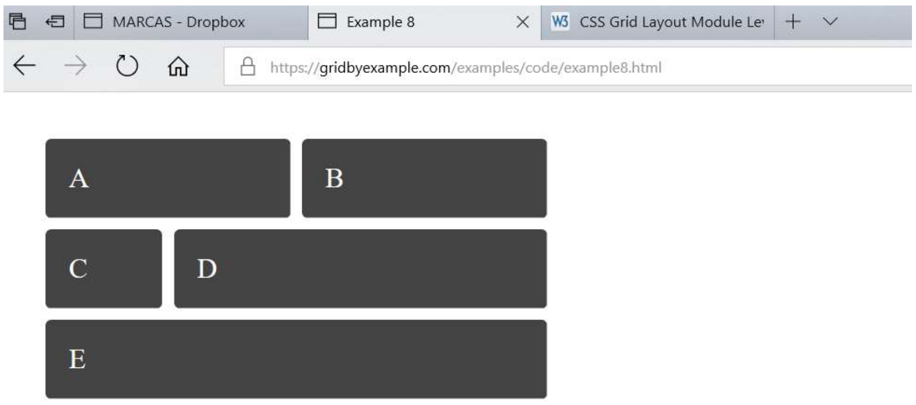
HTML no cambia

```
body {
  margin: 40px;
}

.wrapper {
    display: grid;
  grid-gap: 10px;
    grid-template-columns: [col1-start] 100px  [col2-start] 100px  [col3-start] 100px [col3-end];
    grid-template-rows: [row1-start] auto [row2-start] auto [row2-end];
    background-color: #fff;
    color: #444;
  }

  .box {
    background-color: #444;
    color: #fff;
    border-radius: 5px;
    padding: 20px;
    font-size: 150%;

  }

  .a {
    grid-column: col1-start / col3-start;
    grid-row: row1-start ;
}
.b {
   grid-column: col3-start ;
   grid-row: row1-start / row2-end;
}
.c {
   grid-column: col1-start;
   grid-row: row2-start ;
}
.d {
   grid-column: col2-start ;
   grid-row: row2-start ;
}
```

8/ Line-based placement named lines with spans

You can give lines the same name and then use the span keyword to target lines of a certain name. This is really useful if you want to create a complex grid with multiple content tracks and gutters.

I have made a slightly larger grid here and have named all of the Grid Lines before the content tracks with col and all of the lines before the row tracks with row. I can then start at a certain column line by using col <line number> and span by saying span <number of lines>.


MARCAS - Dropbox
Example 8
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example8.html
A
B
C
D
E

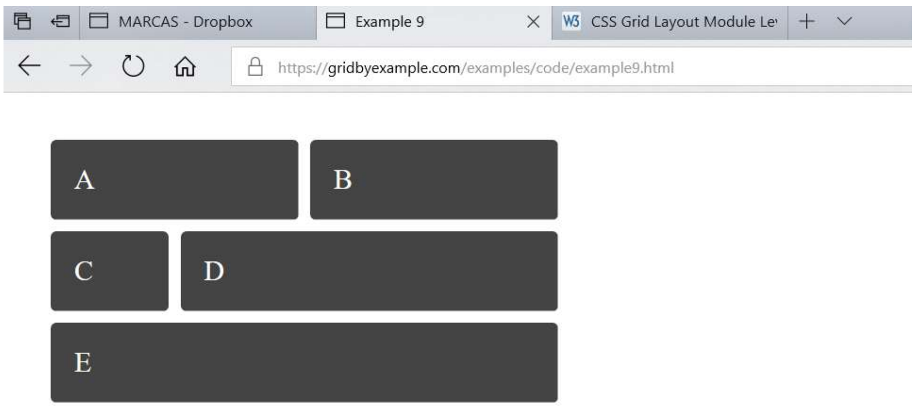
Código

HTML se mantiene

```
body {
  margin: 40px;
}

.wrapper {
      display: grid;
    grid-gap: 10px;
      grid-template-columns: [col] 100px [col] 100px [col] 100px [col] 100px  ;
      grid-template-rows: [row] auto [row] auto [row] ;
      background-color: #fff;
      color: #444;
    }

    .box {
      background-color: #444;
      color: #fff;
      border-radius: 5px;
      padding: 20px;
      font-size: 150%;

    }

    .a {
      grid-column: col / span 2;
      grid-row: row ;
    }
    .b {
      grid-column: col 3 / span 2 ;
      grid-row: row ;
    }
    .c {
      grid-column: col ;
      grid-row: row 2 ;
    }
    .d {
      grid-column: col 2 / span 3 ;
      grid-row: row 2 ;
    }
  .e {
     grid-column: col / span 4;
     grid-row: row 3;
  }
```

9/ Using repeat notation

In example 8 we repeated the same definitions to create our grid with named lines. We could save some typing by using repeat notation. The values for repeat are the number of times you want the expression to repeat and then the expression.


MARCAS - Dropbox
Example 9
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example9.html
A
B
C
D
E

Código

HTML no cambia

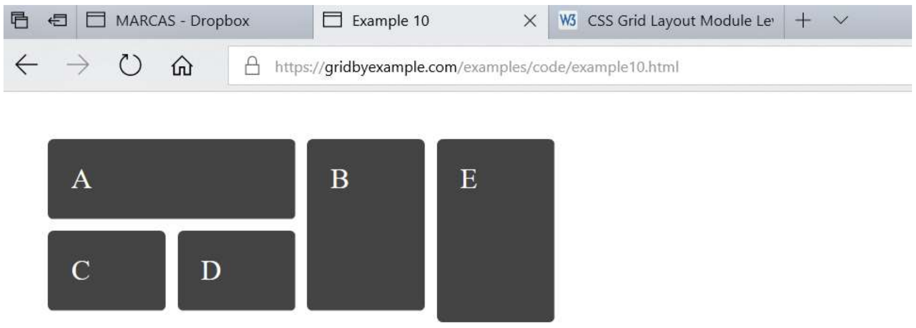
```
body {
 margin: 40px;
}

.wrapper {
     display: grid;
  grid-gap: 10px;
     grid-template-columns: repeat(4, [col] 100px ) ;
     grid-template-rows: repeat(3, [row] auto  );
     background-color: #fff;
     color: #444;
  }
  .box {
     background-color: #444;
     color: #fff;
     border-radius: 5px;
     padding: 20px;
     font-size: 150%;

  }

 .a {
     grid-column: col / span 2;
     grid-row: row ;
  }
  .b {
     grid-column: col 3 / span 2 ;
     grid-row: row ;
  }
  .c {
     grid-column: col ;
     grid-row: row 2 ;
  }
  .d {
     grid-column: col 2 / span 3 ;
     grid-row: row 2 ;
  }

  .e {
     grid-column: col / span 4;
     grid-row: row 3;
  }
```

10/ Explicit and Implicit Grid

When we use grid-template-columns and grid-template-rows we create an Explicit Grid. However if we try and place an item outside of that grid the browser will create an Implicit Grid line or lines in order to hold that item.

In the code below I have put e between grid column lines 4 and 5, these are not described with grid-template-rows, so an implicit grid line 5 is created.

By default the implicit grid tracks created by the implicit lines will be auto sized. However, you can size the tracks with the grid-auto-columns and grid-auto-rows properties. I have sized my auto tracks as 100px to match the rest of the column tracks in my grid.


MARCAS - Dropbox
Example 10
CSS Grid Layout Module Le
https://gridbyexample.com/examples/code/example10.html
A
B
E
C
D

Código

HTML se mantiene

```
body {
 margin: 40px;
}
.wrapper {
    display: grid;
  grid-gap: 10px;
    grid-template-columns: 100px 100px 100px;
  grid-auto-columns: 100px;
    background-color: #fff;
    color: #444;
  }

  .box {
    background-color: #444;
    color: #fff;
    border-radius: 5px;
    padding: 20px;
    font-size: 150%;
 }

.a {
    grid-column: 1 / 3;
    grid-row: 1;
 }
 .b {
    grid-column: 3 ;
    grid-row: 1 / 3;
 }
 .c {
    grid-column: 1 ;
    grid-row: 2 ;
 }
 .d {
    grid-column: 2;
    grid-row: 2;
 }

 .e {
    grid-column: 4 / 5;
    grid-row: 1 / 4;
 }
```