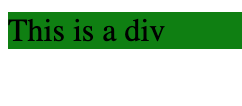
1. **Write three examples of specificity**
   1. **Inline style**<h2 style=”display:none”>Titulo que no se mostrará</h2>
   2. **div span { color: blue; }**
   3. **Id**

<div id="a">This is a div</div>

**<style>**

div#a {background-color: green;}

#a {background-color: yellow;}

div[id=a] {background-color: blue;}

**</style>**

1. **Complete the tutorials.**
   1. **Complete the 28 levels**
   2. **Write a comparative table (pros and cons) between flexbox and grid**

|  | **Flexbox** | **Grid** |
| --- | --- | --- |
| **Layout** | Is one-dimensional layout model | Two-dimensional model |
| **Control** | Organize things horizontally/vertically in one direction. | Two directions. |
| **Overlapping** | Cannot intentionally overlap elements. | Helps to create layouts with overlapping. |
| **Based** | Content | Container |
| **Recommended** | Display items evenly | Pre-establishment of cells to store items in int |
| **Used for** | Scaling, one-sided aligning and organized elements within a container | Define a large-scale layout with more complex and subtle designs. |

* 1. **Research about the best scenarios to use grid or flex**

| **GRID** | **FLEXBOX** |
| --- | --- |
| CSS Grids helps you create the outer layout of the webpage. You can build complex as well responsive design with this. This is why it is called ‘layout first’. | Flexbox mostly helps align content & move blocks. |
| CSS grids are for 2D layouts. It works with both rows and columns. | Flexbox works better in one dimension only (either rows OR columns). |
| For complex design | For small design to implement |
| For have a gap between block element | For align elements |
| For layout-first design | For content-first |

But, It will be more time saving and helpful if you use both at the same time.