1. **Differences between flex and grid.**

| **Feature** | **Flexbox** | **CSS Grid** |
| --- | --- | --- |
| Layout Type | One-dimensional (row or column) | Two-dimensional (rows and columns) |
| Direction of Control | Single axis (horizontal or vertical) | Both axes (rows and columns) |
| Alignment | Alignment within a single line | Alignment in both rows and columns |
| Common Use Cases | Navigation menus, toolbars, centering | Complex page layouts, grid-based designs |
| Nested Layouts | Suited for nested layouts within a Flex container | Powerful for creating nested grids |
| Browser Support | Good support, more universal | Good support, slightly more modern browsers |
| Compatibility | Suitable for one-dimensional layouts | Ideal for two-dimensional layouts |

1. **Properties for parent and child.**

In the context of the Flexbox layout model, "parent properties" typically refer to the properties that are applied to the parent container (the element with **display: flex;** or **display: inline-flex;**) to control the layout and alignment of its child elements. These properties help define how the children are distributed within the flex container.

Some of the key parent properties in Flexbox include:

1. **display**: This is the initial property that activates Flexbox on a container. It can have the values **flex** or **inline-flex**, depending on whether you want to create a block-level or inline-level flex container.
2. **flex-direction**: Determines the direction in which the flex items are placed within the flex container. It can be set to values like **row**, **row-reverse**, **column**, or **column-reverse**.
3. **flex-wrap**: Controls whether the flex items should wrap onto a new line if they overflow the container. Values can be **nowrap**, **wrap**, or **wrap-reverse**.
4. **justify-content**: This property defines how the flex items are aligned along the main axis of the flex container. Common values are **flex-start**, **flex-end**, **center**, **space-between**, and **space-around**.
5. **align-items**: Specifies how the flex items are aligned along the cross axis of the container. Options include **flex-start**, **flex-end**, **center**, **baseline**, and **stretch**.
6. **align-content**: Used to align the flex lines (lines of flex items) along the cross axis when there is extra space in the container. This property is particularly relevant when you have multiple rows or columns of flex items. Values include **flex-start**, **flex-end**, **center**, **space-between**, and **space-around**.
7. **flex-flow**: A shorthand property that combines **flex-direction** and **flex-wrap** into a single declaration. For example, **flex-flow: row wrap;** would set the direction to "row" and enable wrapping.
8. **gap** (or **row-gap** and **column-gap**): Specifies the space between flex items within the container, both horizontally and vertically.

These parent properties allow you to control the overall layout and alignment of the flex container and its child elements, providing flexibility and control over the design of your web page or application.