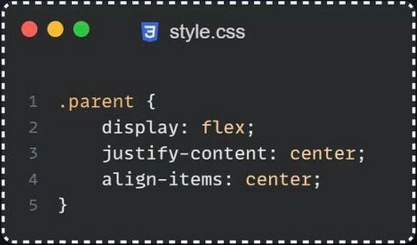
1. Using Flexbox

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure.

Apply the following properties to .parent will center .child horizontally and vertically.

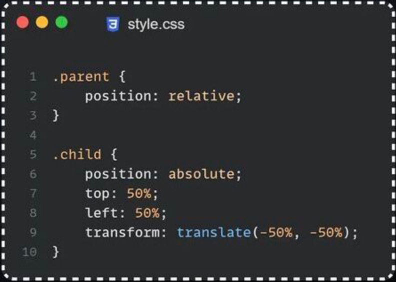


```css
.parent {
    display: flex;
    justify-content: center;
    align-items: center;
}
```

2. Using Position

The position property specifies the type of positioning method used for an element (static, relative, fixed, absolute or sticky). We only need relative and absolute.

Apply following properties to .parent and .child will center .child horizontally and vertically.



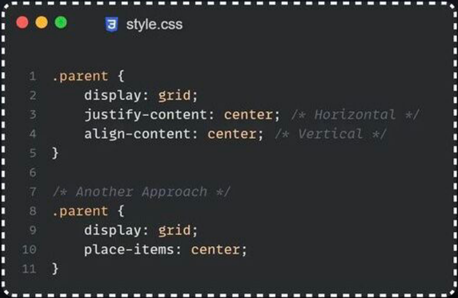
```css
.parent {
    position: relative;
}

.child {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
```

3. Using CSS grid

The CSS Grid Layout Module offers a grid-based layout system, with rows and columns. we can center the child element with this as well.

Apply following properties to .parent will center .child horizontally and vertically.



style.css

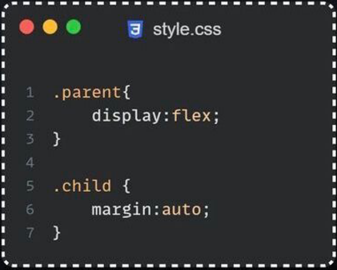
```css
.parent {
    display: grid;
    justify-content: center; /* Horizontal */
    align-content: center; /* Vertical */
}

/* Another Approach */
.parent {
    display: grid;
    place-items: center;
}
```

4. Using margin: auto on a flex item

Flexbox introduced a pretty awesome behavior for auto margins. Now, it not only horizontally centers an element as it did in block layouts, but it also centers it in the vertical axis.

Apply following properties to .parent will center .child horizontally and vertically.



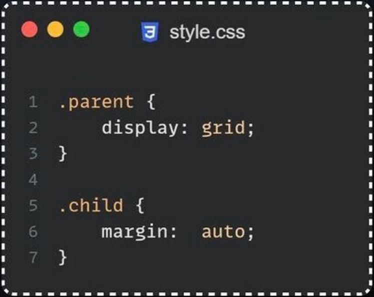
```css
.parent{
    display:flex;
}

.child {
    margin:auto;
}
```

5. Using margin: auto on a grid item

Similarly to Flexbox, applying margin: auto on a grid item centers it on both axes.

Apply following properties to .parent will center .child horizontally and vertically.



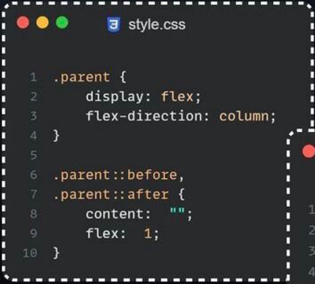
```css
.parent {
    display: grid;
}

.child {
    margin:  auto;
}
```

6. Pseudo-elements on a flex container

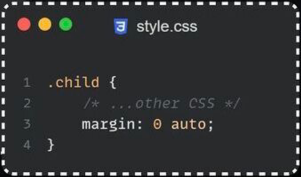
Not the most practical approach in the world, but we can also use flexible, empty pseudo-elements to push an element to the center.

Apply following properties to .parent will center .child horizontally and vertically.



```css
.parent {
    display: flex;
    flex-direction: column;
}

.parent::before,
.parent::after {
    content: "";
    flex: 1;
}
```



```css
.child {
    /* ...other CSS */
    margin: 0 auto;
}
```