CSS `overflow` property

The CSS `overflow` property controls how content is displayed when it exceeds the bounds of its container. It helps you manage what happens when an element's content is too large to fit inside its designated area, such as a div or a section.

Syntax

```
element {
  overflow: value;
}
```

Values of the `overflow` Property

1. **`visible`** (default):
    - Content overflows the container, but nothing is hidden. The overflowing content will be fully visible outside the container's bounds.
    - Example:

      ```
      div {
        overflow: visible;
      }
      ```

    - Use case: When you want to ensure that all content is shown, even if it spills out of its container.
2. **`hidden`** :
    - Content that overflows is clipped and hidden, meaning it won't be visible outside the container. No scrollbars are added.
    - Example:

      ```
      div {
        overflow: hidden;
      }
      ```

    - Use case: When you want to ensure that any overflow content is cut off, useful in cases like creating image galleries or controlling the layout strictly.
3. **`scroll`** :

- Adds scrollbars to the container (both vertical and horizontal) so the user can scroll to see the overflowing content, even if the content fits. Scrollbars will always be visible, regardless of whether the content overflows.
- Example:

```
div {
  overflow: scroll;
}
```

- Use case: When you want to give the user the ability to scroll through content, even if it doesn't overflow.

4. **`auto`**:
   - Scrollbars are added **only if** the content overflows. If the content fits within the container, no scrollbars are shown. This is often the most commonly used value for the `overflow` property.
   - Example:

```
div {
  overflow: auto;
}
```

   - Use case: When you want scrollbars to appear dynamically only when needed.

Other Related Properties

1. **`overflow-x` and `overflow-y`**:
   - These properties allow you to control the horizontal (`overflow-x`) and vertical (`overflow-y`) overflow independently.
   - Example:

```
div {
  overflow-x: hidden;  /* Hide horizontal overflow */
  overflow-y: scroll;  /* Allow vertical scrolling */
}
```

2. **`overflow: clip`** (CSS Overflow Level 4 Draft):
   - Clips the content at the container's edge, but does not add scrollbars. This is useful when you want a more lightweight version of `hidden` without additional scrolling behavior.

- Example:

```
div {
  overflow: clip;
}
```

Practical Examples

1. **Overflow Hidden (Image Gallery)**:

```
.gallery {
  width: 300px;
  height: 200px;
  overflow: hidden;
}
```

   This will crop any image that's larger than 300x200px, preventing it from overflowing the container.

2. **Overflow Scroll (Scrollable Div)**:

```
.scroll-box {
  width: 300px;
  height: 150px;
  overflow: auto;
}
```

   If the content inside the box exceeds 300x150px, scrollbars will appear, allowing the user to scroll through the content.

3. **Vertical Scroll Only**:

```
.scroll-box {
  height: 200px;
  overflow-y: scroll;
  overflow-x: hidden;
}
```

   This will only add a vertical scrollbar, and horizontal overflow will be clipped.

Summary

- The `overflow` property is useful for managing how content is handled when it exceeds its container's boundaries.
- Common values include `visible` (no clipping), `hidden` (content is cut off), `scroll` (always show scrollbars), and `auto` (show scrollbars when needed).
- You can also control overflow for each direction with `overflow-x` and `overflow-y`.