
Overflow

It seems like you're asking about handling overflow in CSS. The overflow property in CSS is used to specify how the content should behave when it exceeds the size of its container. Here's a brief overview:

1. visible (default): Content is not clipped, and it may overflow the container.

```
.example {  
  overflow: visible;  
}
```

2. hidden: Content that overflows the container is simply hidden.

```
.example {  
  overflow: hidden;  
}
```

3. scroll: Adds scrollbars to the container, allowing users to scroll and see the hidden content.

```
.example {  
  overflow: scroll;  
}
```

4. auto: Similar to scroll, but the browser will add scrollbars only if necessary.

```
.example {  
  overflow: auto;  
}
```

5. inherit: Inherits the overflow property from the parent element.

```
.example {  
  overflow: inherit;  
}
```

6. initial: Sets the overflow property to its default value.

```
.example {  
  overflow: initial;  
}
```

You can apply these properties to various HTML elements like divs, paragraphs, or any other container elements. For example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Overflow Example</title>
  <style>
    .container {
      width: 200px;
      height: 200px;
      border: 1px solid #000;
      overflow: scroll; / or any other value /
    }
  </style>
</head>
<body>
  <div class="container">
    <!-- Your content goes here -->
  </div>
</body>
</html>
```

text overflow

The text-overflow property in CSS is used to specify how overflowed content that is not displayed due to a fixed width and height can be indicated to the user. This property is typically used in conjunction with the white-space and overflow properties.

Here are the possible values for text-overflow:

1. clip: The default value. It clips the text at the box's content edge and does not display an ellipsis or any indicator.

```
p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: clip;  
}
```

2. ellipsis: Displays an ellipsis (...) to represent clipped text.

```
p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

3. string: Displays a custom string (other than an ellipsis) to represent clipped text. The string should be placed within quotation marks.

```
p {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: "Read more";  
}
```

Here's an example demonstrating the use of text-overflow with ellipsis:

```
css  
/ Apply text-overflow: ellipsis to a paragraph /  
.overflow-example {  
  white-space: nowrap;  
  overflow: hidden;  
  text-overflow: ellipsis;  
  width: 200px; / Set a fixed width /  
}
```

html

```
<!-- Apply the style to a paragraph with long content -->
```

```
<p class="overflow-example">
```

This is a very long piece of text that will be truncated with an ellipsis if it exceeds the specified width.

```
</p>
```

In this example, the paragraph has a fixed width, and if the content is too long to fit within that width, it will be truncated with an ellipsis at the end. Adjust the width property to control the maximum width before the text is truncated.

visibility

In CSS, the visibility property is used to control the visibility of an element. It can take two main values: visible (default) and hidden.

Here's a brief overview:

1. visible (default): The element is visible.

```
.example {  
  visibility: visible;  
}
```

2. hidden: The element is hidden, but it still takes up space in the layout.

```
.example {  
  visibility: hidden;  
}
```

When an element's visibility is set to hidden, it becomes invisible, but it still occupies space on the page. It's different from display: none;, which would make the element not only invisible but also remove it from the document flow.

Here's an example of how you might use it in HTML and CSS:

html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Visibility Example</title>
```

```
  <style>
```

```
    .hidden-element {
```

```
      visibility: hidden;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p>This is a visible paragraph.</p>
```

```
  <p class="hidden-element">This is a hidden paragraph.</p>
```

```
</body>
```

```
</html>
```

In this example, the second paragraph with the class `hidden-element` is set to have hidden visibility. Even though it is not visible, it still affects the layout of the page. If you want to completely remove an element from the layout, you might use `display: none;`.

cursor property

The CSS `cursor` property is used to define the type of cursor that should be displayed when the mouse pointer is over an element. It allows you to customize the appearance of the cursor to indicate different interactions with the user. Here are some common values for the `cursor` property:

1. `auto`: The browser determines the cursor to display.

CSS

```
cursor: auto;
```

2. `pointer`: The cursor is a pointing hand, usually indicating a clickable link.

CSS

```
cursor: pointer;
```

3. `default`: The default cursor, typically an arrow.

CSS

cursor: default;

4. move: Indicates that the dragged item can be moved.

CSS

cursor: move;

5. text: Indicates text, usually used when hovering over text content.

CSS

cursor: text;

6. crosshair: Displays a crosshair, often used for selecting a region.

CSS

cursor: crosshair;

7. not-allowed: Indicates that the specified action is not allowed or supported.

CSS

cursor: not-allowed;

8. wait: Indicates that the program is busy and the user should wait.

CSS

cursor: wait;

9. help: Indicates that help is available.

CSS

cursor: help;

10. e-resize, n-resize, ne-resize, etc.: Specifies a resizable edge in the corresponding direction.

CSS

cursor: e-resize; / East (right) resize cursor /

cursor: n-resize; / North (up) resize cursor /

cursor: ne-resize; / North-east (up-right) resize cursor /

These are just a few examples, and there are several other values you can use for the cursor property. The appropriate value depends on the desired user experience and the context in which the cursor is being used.