Complete CSS and HTML for Float Property

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Float Property Examples</title>
  <style>
    /* CSS Float Property */

    /*
      1. float: left;
      - The element is floated to the left of its container.
      - Text and inline elements will wrap around it on the right side.
    */
    .float-left {
      float: left;
      width: 200px;
      background-color: lightblue;
      margin-right: 10px; /* Adds space between the floated element and
surrounding content */
    }

    /*
      2. float: right;
      - The element is floated to the right of its container.
      - Text and inline elements will wrap around it on the left side.
    */
    .float-right {
      float: right;
      width: 200px;
      background-color: lightcoral;
      margin-left: 10px; /* Adds space between the floated element and
surrounding content */
    }

    /*
      3. float: none;
      - The element is not floated (default value).
      - It remains in the normal document flow.
    */
    .no-float {
      float: none;
      background-color: lightgreen;
    }
    /*
      4. float: inherit;
      - The element inherits the float value from its parent.
    */
    .inherit-float {
      float: inherit;
    }

    /*
      Clearing Floats:
      - When elements are floated, they are removed from the normal document flow, which can cause
        layout issues (e.g., parent container collapsing). To fix this, use the `clear` property.
      - Values for `clear`:
        - none (default): No clearing.
        - left: Clears floats on the left side.
        - right: Clears floats on the right side.
        - both: Clears floats on both sides.
    */
    .clearfix::after {
      content: "";
      display: table;
      clear: both; /* Ensures the container wraps around floated elements */
    }

    /* Example of Clearing Floats */
    .container {
      border: 2px solid black;
      padding: 10px;
    }

    .container::after {
      content: "";
      display: table;
      clear: both; /* Clears floats inside the container */
    }

    /*
      Float and Inline Elements:
      - Floated elements behave like block-level elements, even if they are inline by default.
    */
    span {
      float: left; /* Inline element behaves like a block-level element */
      background-color: lightyellow;
      padding: 10px;
    }
```

```css
    /*
      Float and Width:
      - If no width is specified, a floated element will shrink to fit its content.
      - It's a good practice to set a width for floated elements to avoid layout issues.
    */
    .float-width {
      float: left;
      width: 50%; /* Takes up 50% of the container's width */
      background-color: lightpink;
    }

    /*
      Float and Margins:
      - Margins on floated elements can be used to create space between the floated element
        and surrounding content.
    */
    .float-margin {
      float: left;
      width: 200px;
      margin: 10px; /* Adds space around the floated element */
      background-color: lightseagreen;
    }

    /*
      Float and Overflow:
      - Using `overflow: auto` or `overflow: hidden` on a parent container can also clear floats.
    */
    .overflow-clear {
      overflow: auto; /* Clears floats inside the container */
      border: 2px solid black;
      padding: 10px;
    }

    /*
      Additional Notes:
      - Floats are not ideal for modern layout designs. Use Flexbox or Grid for better control.
      - Floats can cause issues with parent container height (collapsing). Use clearfix or overflow to fix.
      - Avoid using floats for complex layouts, as they can lead to maintenance challenges.
    */

    /* Example of Float in a Layout */
    .header {
      background-color: lightgray;
      padding: 10px;
    }

    .sidebar {
      float: left;
      width: 25%;
      background-color: lightblue;
      padding: 10px;
    }

    .main-content {
      float: right;
      width: 70%;
      background-color: lightcoral;
      padding: 10px;
    }

    .footer {
      clear: both; /* Clears floats to ensure the footer is below the
floated elements */
      background-color: lightgreen;
      padding: 10px;
    }
  </style>
</head>
<body>

  <!-- Example 1: Float Left -->
  <h2>1. Float Left</h2>
  <div class="float-left">Floated Left</div>
  <p>This is some text that wraps around the floated element. The floated
element is positioned to the left, and the text flows around it on the right
side.</p>

  <hr>

  <!-- Example 2: Float Right -->
  <h2>2. Float Right</h2>
  <div class="float-right">Floated Right</div>
  <p>This is some text that wraps around the floated element. The floated
element is positioned to the right, and the text flows around it on the left
side.</p>

  <hr>

  <!-- Example 3: Float None -->
  <h2>3. Float None</h2>
  <div class="no-float">Not Floated</div>
  <p>This element is not floated and remains in the normal document flow.
</p>
```

```html
  <hr>

  <!-- Example 4: Float Inherit -->
  <h2>4. Float Inherit</h2>
  <div style="float: left;">
    Parent Floated Left
    <div class="inherit-float">Child Inherits Float</div>
  </div>
  <p>The child element inherits the float value from its parent.</p>

  <hr>

  <!-- Example 5: Clearing Floats -->
  <h2>5. Clearing Floats</h2>
  <div class="container">
    <div class="float-left">Floated Left</div>
    <div class="float-right">Floated Right</div>
  </div>
  <p>This container uses a clearfix to ensure it wraps around the floated elements.</p>

  <hr>

  <!-- Example 6: Float and Inline Elements -->
  <h2>6. Float and Inline Elements</h2>
  <span>Inline Element Floated Left</span>
  <p>This is an inline element (span) that is floated to the left. It behaves like a block-level element when floated.</p>

  <hr>

  <!-- Example 7: Float and Width -->
  <h2>7. Float and Width</h2>
  <div class="float-width">Floated with 50% Width</div>
  <p>This floated element has a width of 50%, so it takes up half of the container's width.</p>

  <hr>

  <!-- Example 8: Float and Margins -->
  <h2>8. Float and Margins</h2>
  <div class="float-margin">Floated with Margins</div>
  <p>This floated element has margins to create space around it.</p>

  <hr>

  <!-- Example 9: Float and Overflow -->
  <h2>9. Float and Overflow</h2>
  <div class="overflow-clear">
```

```
    <div class="float-left">Floated Left</div>
    <div class="float-right">Floated Right</div>
  </div>
  <p>This container uses `overflow: auto` to clear floats.</p>

  <hr>

  <!-- Example 10: Float in a Layout -->
  <h2>10. Float in a Layout</h2>
  <div class="header">Header</div>
  <div class="sidebar">Sidebar (Floated Left)</div>
  <div class="main-content">Main Content (Floated Right)</div>
  <div class="footer">Footer (Cleared Floats)</div>

</body>
</html>
```