

# Complete CSS and HTML for Display Property

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

    /*
      1. display: block;
      - The element generates a block-level box.
      - It starts on a new line and takes up the full width available.
    */
    .block-element {
      display: block;
      background-color: lightblue;
      padding: 10px;
      margin: 10px 0;
    }

    /*
      2. display: inline;
      - The element generates an inline-level box.
      - It does not start on a new line and only takes up as much width as
its content.
    */
    .inline-element {
      display: inline;
      background-color: lightcoral;
      padding: 5px;
      margin: 0 5px;
    }

    /*
      3. display: inline-block;
      - The element generates an inline-level box but behaves like a block-
level box.
      - It does not start on a new line, but you can set width, height,
padding, and margins.
    */
    .inline-block-element {
      display: inline-block;
      width: 150px;
```

```

    height: 50px;
    background-color: lightgreen;
    padding: 10px;
    margin: 10px;
}

/*
  4. display: none;
  - The element is completely removed from the document flow.
  - It is not rendered and does not take up any space.
*/
.hidden-element {
    display: none;
}

/*
  5. display: flex;
  - The element becomes a flex container, and its children become flex
items.
*/
.flex-container {
    display: flex;
    justify-content: space-between;
    background-color: lightyellow;
    padding: 10px;
    margin: 10px 0;
}

.flex-item {
    background-color: lightseagreen;
    padding: 10px;
    margin: 5px;
}

/*
  6. display: grid;
  - The element becomes a grid container, and its children become grid
items.
*/
.grid-container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 10px;
    background-color: lightpink;
    padding: 10px;
    margin: 10px 0;
}

.grid-item {
    background-color: lightskyblue;

```

```

padding: 10px;
text-align: center;
}

/*
  7. display: inline-flex;
  - The element behaves like an inline-level box but acts as a flex
container.
*/
.inline-flex-container {
display: inline-flex;
background-color: lightseagreen;
padding: 10px;
margin: 10px 0;
}

/*
  8. display: inline-grid;
  - The element behaves like an inline-level box but acts as a grid
container.
*/
.inline-grid-container {
display: inline-grid;
grid-template-columns: repeat(2, 1fr);
gap: 10px;
background-color: lightskyblue;
padding: 10px;
margin: 10px 0;
}

/*
  9. display: table;
  - The element behaves like a <table> element.
*/
.table-container {
display: table;
width: 100%;
background-color: lightgray;
margin: 10px 0;
}

.table-row {
display: table-row;
}

.table-cell {
display: table-cell;
padding: 10px;
border: 1px solid black;
}

```

```
/*
  10. display: list-item;
  - The element behaves like a <li> element.
*/
.list-item {
  display: list-item;
  margin-left: 20px;
}

/*
  11. display: inherit;
  - The element inherits the display value from its parent.
*/
.inherit-display {
  display: inherit;
}

/*
  12. display: initial;
  - Resets the display property to its default value.
*/
.initial-display {
  display: initial;
}

/*
  13. display: revert;
  - Resets the display property to the value defined by the browser's
  stylesheet.
*/
.revert-display {
  display: revert;
}

/*
  14. display: unset;
  - Acts as 'inherit' if the element inherits a value from its parent,
  otherwise acts as 'initial'.
*/
.unset-display {
  display: unset;
}

/* Additional Styling for Examples */
.example {
  border: 2px solid black;
  padding: 10px;
  margin: 10px 0;
}
```

```
</style>
</head>
<body>

  <!-- Example 1: Block -->
  <h2>1. Display: Block</h2>
  <div class="block-element">Block Element</div>
  <div class="block-element">Another Block Element</div>

  <hr>

  <!-- Example 2: Inline -->
  <h2>2. Display: Inline</h2>
  <span class="inline-element">Inline Element 1</span>
  <span class="inline-element">Inline Element 2</span>
  <span class="inline-element">Inline Element 3</span>

  <hr>

  <!-- Example 3: Inline-Block -->
  <h2>3. Display: Inline-Block</h2>
  <div class="inline-block-element">Inline-Block Element</div>
  <div class="inline-block-element">Another Inline-Block Element</div>

  <hr>

  <!-- Example 4: None -->
  <h2>4. Display: None</h2>
  <div class="hidden-element">This element is hidden (display: none).</div>
  <div>This element is visible.</div>

  <hr>

  <!-- Example 5: Flex -->
  <h2>5. Display: Flex</h2>
  <div class="flex-container">
    <div class="flex-item">Flex Item 1</div>
    <div class="flex-item">Flex Item 2</div>
    <div class="flex-item">Flex Item 3</div>
  </div>

  <hr>

  <!-- Example 6: Grid -->
  <h2>6. Display: Grid</h2>
  <div class="grid-container">
    <div class="grid-item">Grid Item 1</div>
    <div class="grid-item">Grid Item 2</div>
    <div class="grid-item">Grid Item 3</div>
  </div>
```

<hr>

<!-- Example 7: Inline-Flex -->

<h2>7. Display: Inline-Flex</h2>

<div class="inline-flex-container">

    <div class="flex-item">Inline-Flex Item 1</div>

    <div class="flex-item">Inline-Flex Item 2</div>

</div>

<hr>

<!-- Example 8: Inline-Grid -->

<h2>8. Display: Inline-Grid</h2>

<div class="inline-grid-container">

    <div class="grid-item">Inline-Grid Item 1</div>

    <div class="grid-item">Inline-Grid Item 2</div>

</div>

<hr>

<!-- Example 9: Table -->

<h2>9. Display: Table</h2>

<div class="table-container">

    <div class="table-row">

        <div class="table-cell">Table Cell 1</div>

        <div class="table-cell">Table Cell 2</div>

    </div>

    <div class="table-row">

        <div class="table-cell">Table Cell 3</div>

        <div class="table-cell">Table Cell 4</div>

    </div>

</div>

<hr>

<!-- Example 10: List-Item -->

<h2>10. Display: List-Item</h2>

<div class="list-item">List Item 1</div>

<div class="list-item">List Item 2</div>

<hr>

<!-- Example 11: Inherit -->

<h2>11. Display: Inherit</h2>

<div style="display: block;">

    Parent with Display: Block

    <div class="inherit-display">Child with Display: Inherit</div>

</div>

```
<hr>

<!-- Example 12: Initial -->
<h2>12. Display: Initial</h2>
<div class="initial-display">Display: Initial (Resets to default)</div>

<hr>

<!-- Example 13: Revert -->
<h2>13. Display: Revert</h2>
<div class="revert-display">Display: Revert (Resets to browser default)
</div>

<hr>

<!-- Example 14: Unset -->
<h2>14. Display: Unset</h2>
<div class="unset-display">Display: Unset (Acts as inherit or initial)
</div>

</body>
</html>
```