CSS `display` Property – Senior Developer Reference

This document is a **complete, practical, and interview-ready guide** to the CSS `display` property. It is written from a **senior CSS developer perspective**, focusing on **real behavior, common mistakes, and real-world usage**.

---

1. What is the `display` Property?

The `display` property controls **how an element participates in the layout**. It defines whether an element starts on a new line, stays inline, wraps, or is not rendered.

---

2. Default Display Behavior

HTML Element	Default Display
`div`, `p`, `section`, `h1`	`block`
`span`, `a`, `strong`	`inline`
`img`, `button`	`inline-block`

---

3. `display: block`

• Starts on a new line • Takes full available width by default • Respects width, height, margin, and padding

Example

```
<div class="block-box">Block Element</div>
```

```
.block-box {
  display: block;
  width: 200px;
  height: 100px;
  background: lightblue;
}
```

Real usage: page sections, headers, cards, layout containers

---

4. `display: inline`

• Appears in the same line like text • Width and height are ignored • Vertical margin has no effect

Example

```
<span class="inline-text">Inline Element</span>
```

```
.inline-text {
  display: inline;
  width: 200px; /* ignored */
}
```

Real usage: text formatting, links inside content

---

5. `display: inline-block`

• Appears in the same line • Respects width and height • Wraps to next line when space is not available

Example (2×2 Layout)

```
<div class="wrapper">
  <div class="box">Box 1</div>
  <div class="box">Box 2</div>
  <div class="box">Box 3</div>
  <div class="box">Box 4</div>
</div>
```

```
.wrapper {
  width: 360px;
  margin: auto;
}

.box {
  display: inline-block;
  width: 150px;
  height: 100px;
  margin: 10px;
  background: lightcoral;
  text-align: center;
  line-height: 100px;
}
```

Real usage: cards, image galleries, button groups

---

6. `display: none`

• Element is not rendered • No space is occupied • Element still exists in the DOM

Example

```
.hidden {
  display: none;
}
```

Real usage: modals, dropdowns, conditional UI

---

7. `visibility: hidden`

• Element is invisible • Space is still reserved • Element remains in layout flow

Example

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

Real usage: temporarily hiding UI without layout shift

---

8. Navigation Menu Using `display`

HTML

```
<nav class="menu">
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Services</a>
  <a href="#">Contact</a>
</nav>
```

CSS

```
.menu {
  background: #222;
  text-align: center;
}

.menu a {
  display: inline-block;
  padding: 12px 20px;
  color: white;
  text-decoration: none;
}

.menu a:hover {
  background: #444;
}
```

Senior tip: Apply layout styles to the parent, not individual links

---

9. Tooltip Using `display`

HTML

```
<div class="tooltip">Hover me
  <span class="tip">Tooltip text</span>
</div>
```

CSS

```
.tooltip {
  display: inline-block;
  position: relative;
}

.tip {
  display: none;
  position: absolute;
  background: black;
  color: white;
  padding: 6px;
  top: 120%;
}

.tooltip:hover .tip {
  display: block;
}
```

Real usage: help icons, form hints

---

10. Modal / Popup Pattern

```
.modal {
  display: none;
}

.modal.active {
  display: block;
}
```

Senior tip: CSS controls visibility; JavaScript controls state

---

11. Comparison Table (Interview Favorite)

Property	New Line	Width/Height	Space Occupied	In DOM
block	Yes	Yes	Yes	Yes
inline	No	No	Yes	Yes
inline-block	No	Yes	Yes	Yes
display: none	No	No	No	Yes
visibility: hidden	No	Yes	Yes	Yes

---

12. Common Mistakes (Senior Notes)

• Assuming class names control layout • Using `display: inline` with width/height • Forgetting container width with inline-block • Confusing DOM removal with visibility

---

13. Interview Questions

1. What is the CSS display property?
2. Difference between inline and inline-block?
3. display: none vs visibility: hidden?
4. Why does inline ignore width and height?
5. How does inline-block wrapping work?
6. Can CSS remove an element from the DOM?
7. How do you create a navbar without Flexbox?
8. When would you use visibility hidden over display none?

---

14. Golden Rules (Remember These)

• Layout is controlled by `display`, not class names • CSS hides, JavaScript removes from DOM • Inline-block wraps based on available width • Parent controls alignment

---

End of Document