

# CSS Flexbox — Beginner to Pro (Video-Style Script, Mobile■First)

- > **Audience**: New students who have never used Flexbox
- > **Goal**: Build intuition + muscle memory to layout real UI (navbar, forms, cards)
- > **Format**: Narration script + on-screen steps + exercises + checkpoints
- > **Assets**: Use the included `flexbox-masterclass.html`, `flexbox.css`, `flexbox.js`

## 0) Hook (00:00–00:45)

**Narration:**

“Ever spent hours nudging boxes with margins? Flexbox lets you snap UI pieces into place—perfect for mobile-first designs, navbars, forms, and card grids. In the next 30 minutes, you’ll go from zero to deploying a responsive layout with confidence.”

**On■screen:**

- Show before/after: stacked messy layout → clean responsive layout
- Display axes diagram (main vs cross)

## 1) Setup (00:45–02:00)

**Narration:**

“Create a project folder with three files. Link CSS/JS from your HTML.”

**Steps:**

1. Make a folder `flexbox-course/`
2. Add files: `index.html`, `style.css`, `app.js`
3. In `index.html` add the boilerplate and link CSS/JS.

**Code (index.html):**

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Flexbox Course</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <main class="container">
    <section id="s1" class="card">
      <h1>Flexbox Playground</h1>
      <div class="play">
        <div class="box">A</div>
        <div class="box">B</div>
```

```

        <div class="box">C</div>
    </div>
</section>
</main>
<script src="app.js"></script>
</body>
</html>

```

**\*\*Code (style.css):\*\***

```

*{box-sizing:border-box} body{margin:0;font:16px/1.5 system-ui;background:#0f172a;color:#e5e7eb}
.container{width:min(1100px,92%);margin:auto}
.card{background:#111827;border:1px solid #1f2937;border-radius:12px;padding:16px;margin:16px 0}
.play{border:1px dashed #334155;border-radius:12px;padding:12px;min-height:140px}
.box{background:#1e293b;border:1px solid #334155;border-radius:8px;padding:14px 18px;font-weight:

```

## 2) The Switch (display:flex) (02:00–04:00)

**\*\*Narration:\*\***

“The magic begins when the container becomes a flex container.”

**\*\*Code:\*\***

```

.play{
  display:flex;           /* enables flex behavior */
  flex-direction:column; /* mobile-first: stack items vertically */
  gap:12px;               /* clean spacing */
  align-items:center;     /* center items on cross-axis */
  justify-content:center; /* center along main-axis */
}

```

**\*\*Checkpoint:\*\***

- Resize the browser: items stack, spacing via `gap`, centered both ways.

**\*\*Pitfall Alert:\*\***

- `gap` is on the container (not margins on items). Easier and safer.

## 3) Axes, Justify, Align (04:00–07:00)

**\*\*Narration:\*\***

“Main axis depends on direction. With `row`, main axis is horizontal; with `column`, vertical. `justify-content` aligns along main; `align-items` aligns cross.”

**\*\*Mini demo:\*\***

```

/* Try switching at runtime */
@media (min-width:768px){
  .play{ flex-direction:row; } /* tablet+ → row layout */
}

```

#### **\*\*Exercise A:\*\***

- Toggle `flex-direction` between `column` and `row`.
- Try `justify-content: space-between | space-around | space-evenly`.
- Try `align-items: stretch | center | flex-start | flex-end`.

## **4) Wrapping + Align Content (07:00–10:00)**

#### **\*\*Narration:\*\***

“For multi-row responsive grids, turn wrapping on.”

#### **\*\*Code:\*\***

```
.play{
  display:flex;
  flex-direction:row;
  flex-wrap:wrap;      /* allow items to wrap to next line */
  gap:12px;
  align-content:start; /* multi-row spacing on cross-axis when wrap happens */
}
.box{ flex: 1 1 220px; min-width:0; }
```

#### **\*\*Why `min-width:0`?\***

- It allows text/images to shrink inside the item; prevents overflow.

#### **\*\*Exercise B:\*\***

- Add 6 boxes (A–F). Check that the layout auto-wraps into new rows.

## **5) The `flex` Shorthand (grow shrink basis) (10:00–14:00)**

#### **\*\*Narration:\*\***

“`flex: 1 1 220px` means: grow if space left, shrink if tight, start at ~220px.”

#### **\*\*Code:\*\***

```
.box{ flex: 1 1 220px; } /* fluid columns */
```

#### **\*\*Try:\*\***

- `flex: 0 1 300px` (don't grow, can shrink, base 300px)
- `flex: 1 0 200px` (grow allowed, no shrink) → watch overflow on narrow screens

#### **\*\*Pitfall Alert:\*\***

- Prefer `flex-basis` (the third value) over `width` inside flex rows.

## **6) Ordering & Per-Item Alignment (14:00–16:00)**

**\*\*Narration:\*\***

“You can visually reorder items without changing HTML, and override alignment per item.”

**\*\*Code:\*\***

```
.box:nth-child(1){ order:2 } /* A becomes last among A/B/C */
.box:nth-child(2){ order:1 } /* B first */
.box:nth-child(3){ order:3 } /* C middle, etc. */

.box.special{ align-self:flex-end } /* override container's align-items */
```

**\*\*Exercise C:\*\***

- Reorder boxes to `B, C, A`.
- Make box C align to bottom using `align-self:flex-end`.

## 7) Pattern 1 — Navbar (16:00–19:00)

**\*\*Narration:\*\***

“Classic spacer trick with `margin-left:auto`.”

**\*\*HTML:\*\***

```
<div class="navline">
  <div class="left"><strong>Brand</strong></div>
  <div class="spacer"></div>
  <nav class="right">
    <a href="#">Docs</a>
    <a href="#">Blog</a>
    <button class="btn">Sign up</button>
  </nav>
</div>
```

**\*\*CSS:\*\***

```
.navline{display:flex;align-items:center;gap:12px;padding:8px;border:1px solid #334155;border-ra
.spacer{margin-left:auto}
.right{display:flex;align-items:center;gap:8px}
```

**\*\*Exercise D:\*\***

- Turn this into a mobile-first navbar: stack on mobile, row on  $\geq 768\text{px}$ .

## 8) Pattern 2 — Mobile First Form Rows (19:00–23:00)

**\*\*Narration:\*\***

“Stack labels and inputs on mobile → align in rows on larger screens.”

**\*\*HTML:\*\***

```
<form class="form">
  <div class="row">
    <label>Name</label>
```

```

        <input placeholder="Jane Doe" required />
    </div>
    <div class="row">
        <label>Email</label>
        <input type="email" placeholder="you@example.com" required />
    </div>
</form>

```

**\*\*CSS:\*\***

```

.form .row{display:flex;flex-direction:column;gap:8px}
@media (min-width:768px){
    .form .row{flex-direction:row;align-items:center}
    .form .row>label{flex:0 0 140px}
    .form .row>:is(input,select,textarea){flex:1 1 auto;min-width:0}
}

```

**\*\*Checkpoint:\*\***

- Inputs stretch nicely and never overflow thanks to `min-width:0`.

## 9) Pattern 3 — Card Deck (Equal Heights) (23:00–26:00)

**\*\*HTML:\*\***

```

<div class="deck">
    <article class="card item"><h4>Card A</h4><p>...</p><button>View</button></article>
    <article class="card item"><h4>Card B</h4><p>...</p><button>View</button></article>
    <article class="card item"><h4>Card C</h4><p>...</p><button>View</button></article>
</div>

```

**\*\*CSS:\*\***

```

.deck{display:flex;flex-wrap:wrap;gap:12px;align-items:stretch}
.deck>.item{flex:1 1 240px;display:flex;flex-direction:column;gap:8px;min-width:0}

```

**\*\*Checkpoint:\*\***

- All cards same height due to `align-items:stretch` and internal flex.

## 10) Pattern 4 — Media Object (26:00–27:30)

**\*\*HTML:\*\***

```

<div class="media">
    
    <div class="content">
        <h4>Title</h4>
        <p>Fixed image + flexible content text that can wrap.</p>
    </div>
</div>

```

**\*\*CSS:\*\***

```

.media{display:flex;gap:12px;align-items:center}

```

```
.media img{width:96px;height:96px;flex:0 0 auto;object-fit:cover;border-radius:10px}
.media .content{min-width:0}
```

## 11) Debugging & Pitfalls (27:30–29:00)

- **Item won't shrink** → Add `min-width:0` to the flex item (very common)
- **Expect multiple rows** → Use `flex-wrap:wrap`
- **Misusing `align-content`** → Only works when wrapping creates multiple rows
- **Spacing** → Prefer container `gap` over margins

## 12) Final Challenge (29:00–30:00)

**Build a landing header** with: brand (left), nav links (right), a CTA button; below it a 3■up card deck; then a mobile■first form.

Use only Flexbox + the techniques above.

Try to finish in 20 minutes.

## Appendix: Minimal Snippets

**Spacer navbar**

```
.nav{display:flex;align-items:center}
.nav .spacer{margin-left:auto}
```

**Fluid columns**

```
.row{display:flex;flex-wrap:wrap;gap:12px}
.col{flex:1 1 220px;min-width:0}
```

**Form row**

```
.row{display:flex;flex-direction:column}
@media(min-width:768px){.row{flex-direction:row;align-items:center}}
```

Happy building! ■