📘 CSS শিখুন পানির মতো সহজ করে By জিল্লুর রহমান

সূচিপত্র:

📖 অধ্যায় ১: CSS পরিচিতি (Introduction to CSS) - 👉

📖 অধ্যায় ২: কালার এবং টাইপোগ্রাফি (Colors and Typography) - 👉

📖 অধ্যায় ৩: CSS সিলেক্টর এবং স্পেসিফিসিটি (CSS Selectors and Specificity) - 👉

📖 অধ্যায় ৪: CSS বক্স মডেল (The CSS Box Model) - 👉

📖 অধ্যায় ৫: লেআউট ফান্ডামেন্টালস: ডিসপ্লে এবং ফ্লোট (Layout Fundamentals) - 👉

📖 অধ্যায় ৬: CSS পজিশনিং এবং Z-Index (CSS Positioning and Z-Index) - 👉

📖 অধ্যায় ৭: লিস্ট এবং ন্যাভিগেশন মেনু স্টাইলিং (Styling Lists and Menus) - 👉

📖 অধ্যায় ৮: ফ্লেক্সিবল বক্স লেআউট (Flexbox) - 👉

📖 অধ্যায় ৯: CSS গ্রিড লেআউট (CSS Grid Layout) - 👉

📖 অধ্যায় ১০: রেসপন্সিভ এবং মিডিয়া কোয়েরি (Responsive and Media Queries) - 👉

📖 অধ্যায় ১১: CSS ট্রানজিশন এবং অ্যানিমেশন (CSS Transitions & Animations) - 👉

📖 অধ্যায় ১২: অ্যাডভান্সড CSS: ভ্যারিয়েবল, ফাংশন, এবং মডার্ন ফিচার - 👉

📖 অধ্যায় ১৩: Extra CSS প্রপার্টি এবং ফিচার - 👉

> আমি আমার সর্বোচ্চ দিয়ে CSS-কে সুন্দর, সহজ এবং বাস্তবিক ভাবে দেখানোর চেষ্টা করেছি, যা বিগিনার থেকে অ্যাডভান্সড পর্যন্ত সবার জন্য উপযুক্ত। বইটিতে CSS-এর সমস্ত প্রপার্টি, ইউনিট, প্যাটার্ন ও ফিচার নিয়ে ১৩টি ক্যাটাগরিতে ভাগ করে সাজানো হয়েছে, যেন আমরা CSS সকল প্রপার্টি এবং কাজের সম্পর্কে ক্লিয়ার ধারণা পাবো এবং মনে রাখতে খুব সহজ হয়।
>
> প্রতিটি অধ্যায়ে আছে স্পষ্ট উদাহরণ, হাতে-কলমে প্র্যাকটিস টাস্ক, ১০টি করে MCQ কুইজ, এবং আপনার CSS স্কিল চেক করার জন্য হোমওয়ার্ক, যাতে শেষ পর্যন্ত বলতে পারেন হ্যাঁ, আলহামদুলিল্লাহ! এখন আমি CSS-এ পারি এবং পোর্টফোলিও তে স্কিল হিসাবে অ্যাড করতে পারি।

📖 অধ্যায়-১: CSS পরিচিতি (Introduction to CSS)

🔷 CSS কি?

CSS (Cascading Style Sheets) হলো ওয়েব পেজের লুক এবং ফিল নিয়ন্ত্রণ করার ভাষা। HTML দিয়ে আমরা কনটেন্টের স্ট্রাকচার তৈরি করি, আর CSS দিয়ে সেই কনটেন্টের রঙ, ফন্ট, মার্জিন, লে-আউট ইত্যাদি স্টাইল করি।এক কথায় HTML ডকুমেন্টে কে CSS ব্যবহার করে স্টাইলিং এর কাজ করা হয়।

🔷 কেন CSS শিখবেন?

- HTML কোড কে বা layout কে মনের মতো ডেডাইজ বা স্টাইল করতে
- ওয়েবসাইট সুন্দর ও ব্যবহারবান্ধব করতে
- এক কোড বিভিন্ন ডিভাইসে স্টাইলিং করতে
- দ্রুত, সহজ ও পরিষ্কার কোড লিখতে
- ইন্টারএকটিভ ও অ্যানিমেটেড ওয়েবসাইট তৈরি করতে

📝 CSS লেখার তিনটি পদ্ধতি

1. **Inline CSS**: HTML এলিমেন্টের style অ্যাট্রিবিউটে লেখা

```
<h1 style="color: blue;">এটি একটি প্যারাগ্রাফ</p>
```

2. **Internal CSS**: HTML ফাইলের `<head>` সেকশনে `<style>` ট্যাগের মধ্যে লেখা

```
<head>
  <style>
    h1 { color: blue; }
  </style>
</head>
```

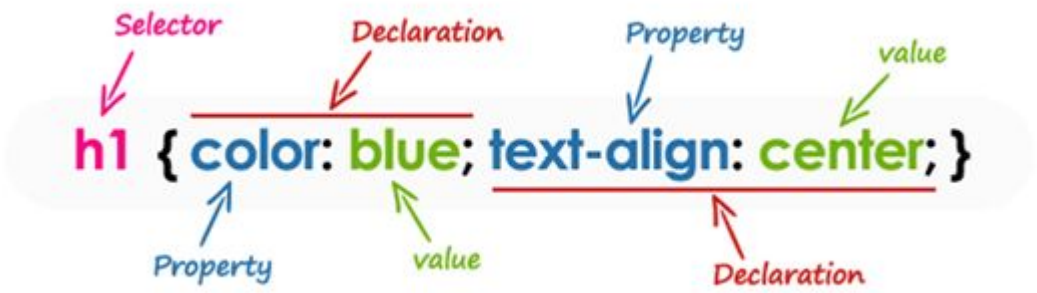
3. **External CSS**: আলাদা .css ফাইলে লেখা এবং HTML ফাইলে লিংক করা

```
<link rel="stylesheet" href="styles.css">
```

🎯 CSS সিনট্যাক্স

```
selector {
    property: value;
```

```
}
```



🔷 CSS এর তিনটি অংশ:

- সিলেক্টর: HTML এলিমেন্ট সিলেক্ট করে।
- প্রপার্টি: স্টাইলের ধরন নির্ধারণ করে।
- ভ্যালু: স্টাইলের ভ্যালু নির্ধারণ করে।

🎯 উদাহরণ

```css
h1 {
  color: blue;
  font-size: 120px;
  text-align: center;
}
```

📊 ইউনিট ও পরিমাপ বার বার লাগবে এগুলো সম্পর্কে জেনে নেই:

Unit	অর্থ	ব্যবহার ক্ষেত্র
`px`	Pixel (স্থির)	ফিক্সড ভ্যালু, এক পিক্সেল একটা ডট এর সমান
`em`	Parent font-size	স্কেল-ফ্রেন্ডলি প্যাডিং ও মার্জিনে।
`rem`	Root font-size	সারাবিশ্বে একরকম স্কেল ধরে রাখতে।
`%`	Parent-এর শতকরা হার	ফ্লুইড প্রস্থ বা উচ্চতার জন্য, 100% মনে সম্পূর্ণ জুড়ে
`vh` / `vw`	Viewport height/width	ফুল-স্ক্রিন সেকশন বা হিরো ব্যানারে।

📊 বেসিক CSS প্রপার্টি সমুহ:

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
color	color-name, hex, rgb(), rgba()	টেক্সটের রঙ সেট করে
font-size	px, em, rem, %	টেক্সটের সাইজ নির্ধারণ করে

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
text-align	left, right, center, justify	টেক্সটের অ্যালাইনমেন্ট সেট করে

💡 CSS এর কিছু গুরুত্বপূর্ণ পয়েন্ট

1. CSS কেস সেনসিটিভ, এর মানে Uppercase এবং Lowercase এর দিকে খেয়াল রাখতে হবে।
2. প্রতিটি ডিক্লেয়ারেশন বা CSS লাইন সেমিকোলন দিয়ে শেষ করতে হয়, না হলে Error আসবে।
3. একাধিক ভ্যালু স্পেস দিয়ে আলাদা করতে হয়।
4. কমেন্ট লেখার জন্য /* */ ব্যবহার করা হয়।
5. External CSS সবচেয়ে বেশি ব্যবহৃত হয়।
6. সহজে CSS সিনট্যাক্স মনে রাখতে, বারবার অনুশীলন করুন।

📝 প্র্যাকটিস টাস্ক

1. একটি HTML ফাইল তৈরি করুন এবং তিনটি পদ্ধতিতে CSS অ্যাপ্লাই করুন
2. একটি প্যারাগ্রাফের টেক্সট কালার লাল করুন এবং ব্যাকগ্রাউন্ড হলুদ করুন
3. একটি হেডিং কে সেন্টার অ্যালাইন করুন এবং ফন্ট সাইজ 124px করুন

🤔 MCQ কুইজ

1. CSS এর পূর্ণরূপ কি?
    - a) Cascading Style Sheets
    - b) Creative Style Sheets
    - c) Computer Style Sheets
    - d) Colorful Style Sheets
2. নিচের কোনটি ইনলাইন CSS এর উদাহরণ?
    - a) `<style>p { color: red; }</style>`
    - b) `<p style="color: red;">`
    - c) `p { color: red; }`
    - d) `@style(color: red)`
3. CSS এ কমেন্ট কিভাবে লিখা হয়?
    - a) // কমেন্ট
    - b) < !-- কমেন্ট -->
    - c) /* কমেন্ট */
    - d) # কমেন্ট
4. text-align প্রপার্টির সঠিক ভ্যালু কোনটি?

- a) top
- b) middle
- c) center
- d) up

5. কোন CSS পদ্ধতি সবচেয়ে বেশি প্রচলিত?

   - a) Inline CSS
   - b) Internal CSS
   - c) External CSS
   - d) Mixed CSS

6. CSS এ একটি ক্লাস সিলেক্টর কিভাবে লেখা হয়?

   - a) #className
   - b) .className
   - c) @className
   - d) *className

7. CSS এ background-color এর ভ্যালু হিসেবে কোনটি ব্যবহার করা যায় না?

   - a) red
   - b) #FF0000
   - c) rgb(255,0,0)
   - d) color-red

8. font-size এর ইউনিট হিসেবে কোনটি ব্যবহার করা যায়?

   - a) kg
   - b) px
   - c) meter
   - d) celsius

9. CSS এ সেমিকোলন (;) কোথায় ব্যবহার করা হয়?

   - a) প্রতিটি সিলেক্টরের শেষে
   - b) প্রতিটি প্রপার্টির শেষে
   - c) প্রতিটি ডেক্লারেশনের শেষে
   - d) প্রতিটি ভ্যালুর শেষে

10. CSS এ কোন ব্র্যাকেট ব্যবহার করা হয়?

    - a) ()
    - b) []

- c) {}
- d) <>

📚 হোমওয়ার্ক

1. একটি সম্পূর্ণ HTML পেজ তৈরি করুন যেখানে:
   - একটি হেডিং থাকবে
   - তিনটি প্যারাগ্রাফ থাকবে
   - একটি লিস্ট থাকবে
2. উপরের HTML পেজে তিনটি পদ্ধতিতে CSS অ্যাপ্লাই করুন:
   - Inline CSS দিয়ে হেডিং এর কালার পরিবর্তন করুন
   - Internal CSS দিয়ে প্যারাগ্রাফ গুলোর ফন্ট সাইজ পরিবর্তন করুন
   - External CSS দিয়ে লিস্টের স্টাইল পরিবর্তন করুন
3. আপনার কোড এবং আউটপুট এর স্ক্রিনশর্ট নিয়ে গ্রুপে পোস্ট করুন

📖 অধ্যায় ২: কালার এবং টাইপোগ্রাফি (Colors and Typography)

ওয়েবসাইট ডিজাইনে রং ও টেক্সট স্টাইলিং খুবই গুরুত্বপূর্ণ ভূমিকা পালন করে। এই অধ্যায়ে আমরা শিখবো কীভাবে CSS দিয়ে লেখা/টেক্সটের রঙ, ফন্ট, আকার, ও অন্যান্য টেক্সট-সংশ্লিষ্ট স্টাইল নিয়ন্ত্রণ করা যায়। পাশাপাশি ব্যাকগ্রাউন্ড রঙ সেট করা ও টেক্সটের চারিত্রিক বৈশিষ্ট্য (বোল্ড, ইটালিক ইত্যাদি) নির্ধারণ করাও শিখবো।

🎨 কালার প্রপার্টি সমূহ

📊 কালার সংক্রান্ত প্রপার্টি

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
color	color-name, hex, rgb(), rgba(), hsl(), hsla()	টেক্সটের রঙ সেট করে
background-color	color-name, hex, rgb(), rgba(), transparent	এলিমেন্টের ব্যাকগ্রাউন্ড রঙ সেট করে
border-color	color-name, hex, rgb(), rgba()	বর্ডারের রঙ সেট করে
outline-color	color-name, hex, rgb(), rgba(), invert	আউটলাইনের রঙ সেট করে
opacity	(0.0 - 1.0) 0 থেকে 1 পর্যন্ত	এলিমেন্টের স্বচ্ছতা/ঝাপসা-গার এর পরিমাণ নির্ধারণ করে

🎯 উদাহরণ

```
h1 {
  color: #FF5733;
  background-color: rgba(255, 255, 255, 0.5); /* 50% স্বচ্ছ লাল */
  opacity: 0.8;
}
```

🎨 কালার ভ্যালু টাইপ

1. কালার নেম

```
color: red;
color: blue;
color: green;
....
```

2. হেক্স কালার

```
color: #FF0000; /* লাল */
color: #00FF00; /* সবুজ */
color: #0000FF; /* নীল */
....
```

3. RGB/RGBA

```
color: rgb(255, 0, 0);     /* লাল */
color: rgba(255, 0, 0, 0.5); /* 50% স্বচ্ছ লাল */
....
```

4. HSL/HSLA

```
color: hsl(0, 100%, 50%);     /* লাল */
color: hsla(0, 100%, 50%, 0.5); /* 50% স্বচ্ছ লাল */
....
```

📝 টাইপোগ্রাফি প্রপার্টি সমূহ

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
font-family	Arial, sans-serif, etc.	টেক্সটের ফন্ট টাইপ/পরিবার সেট করে
font-size	px, em, rem, %, keywords	টেক্সটের সাইজ সেট করে

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
font-weight	normal, bold, 100-900	টেক্সটের বোল্ডনেস সেট করে
font-style	normal, italic, oblique	টেক্সটের স্টাইল সেট করে
text-align	left, right, center, justify	টেক্সট অ্যালাইনমেন্ট সেট করে
text-decoration	none, underline, overline, line-through	টেক্সট ডেকোরেশন সেট করে
line-height	number, length, %	লাইনের উচ্চতা সেট করে
letter-spacing	normal, length	অক্ষরের মধ্যে দূরত্ব সেট করে
word-spacing	normal, length	শব্দের মধ্যে দূরত্ব সেট করে
text-transform	none, capitalize, uppercase, lowercase	টেক্সট ট্রান্সফরমেশন সেট করে
text-shadow	h-shadow v-shadow color (1, 2, red)	টেক্সটে ছায়া যোগ করে

📝 আরো এক্সট্রা কিছু টাইপোগ্রাফি প্রপার্টি

প্রপার্টি	সম্ভবতঃ ভ্যালু	বর্ণনা
`text-overflow`	`clip`, `ellipsis`, `"…"`	অতিরিক্ত টেক্সট কে কেটে দেখানোর জন্য; `max-width` + `white-space: nowrap` কম্বো।
`white-space`	`normal`, `nowrap`, `pre`, `pre-wrap`, `pre-line`, `break-spaces`	স্পেস ও লাইনের ফ্লো কন্ট্রোল; কোড ব্লকে `pre` মোড মাস্ট।
`writing-mode`	`horizontal-tb`, `vertical-rl`, `vertical-lr`	লেখা উল্টো-খাড়া রোটেশন; মাঙ্গা-স্টাইল টাইটেল বানাতে।
`caret-color`	`<color>`	ইনপুট কার্সারের রঙ; ব্র্যান্ড থিম ইনপুটে ম্যাচ করাও।
`accent-color`	`<color>`	চেকবক্স/রেডিও বাটনের থিম রঙ; ব্র্যান্ডেড ফর্মে এক লাইনে।
`color-scheme`	`normal`, `light`, `dark`, `only light`, `only dark`	ব্রাউজারকে থিম সিগন্যাল; OS ডার্ক-মোড কন্ট্রোল।
`text-indent`	`<length>`, `<percentage>`, `hanging`	প্রথম লাইনের ইনডেন্ট; ব্লক-কোড স্টাইলে হ্যাংগিং।
`text-align-last`	`auto`, `left`, `right`, `center`, `justify`, `start`, `end`	প্যারাগ্রাফের শেষ লাইনের অ্যালাইনমেন্ট; ম্যাগাজিন-স্টাইল কলামে কাজে লাগে।

প্রপার্টি	সম্ভবতঃ ভ্যালু	বর্ণনা
`font-stretch`	`ultra-condensed` → `ultra-expanded`, `50%–200%`	ফন্টের প্রস্থ টানাটানি; কেবল ভ্যারিয়েবল-সাপোর্টেড ফন্টেই কাজ করে।
`font-variant`	`normal`, `small-caps`, `all-small-caps`, `titling-caps`	স্মল-ক্যাপস ইত্যাদি; শিরোনামে ক্লসিক লুক আনার জন্য।
`font-variant-ligatures`	`none`, `common-ligatures`, `discretionary-ligatures`, `historic-ligatures`, …	টাইপোগ্রাফি লিগেচার টগল; লোগোটাইপে কাস্টম ইফেক্টের জন্য।
`font-size-adjust`	`none`, `<number>`	x-height ভিত্তিক আকার সামঞ্জস্য; সিস্টেম ফন্ট ব্যাকআপ।
`font-optical-sizing`	`auto`, `none`	ভ্যারিয়েবল ফন্টের অপটিক্যাল সাইজিং চালু/বন্ধ; পাঠযোগ্যতা বাড়াতে।
`font-variation-settings`	`'<axis>' <value>` (e.g. `'wght' 100`)	ফাইন-টিউন ভ্যারিয়েবল ফন্টের অ্যাক্সিস; UI-রেঞ্জ স্লাইডার দিয়ে লাইভ টুইক।

🎯 উদাহরণ

```
h1 {
  font-family: "Times New Roman", serif;
  font-size: 120px;
  font-weight: bold;
  font-style: italic;
  line-height: 1.5;
  text-align: center;
  text-decoration: underline;
  text-transform: capitalize;
  letter-spacing: 1px;
  word-spacing: 3px;
  text-shadow: 2px 2px 5px blue
}
```

🌐 গুগল ফন্ট ব্যবহার

```
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
```

```
body {
    font-family: 'Roboto', sans-serif; /* গুগল ফন্ট থেকে ফন্টটি আসে সরাসরি ওয়েবসাইট লোড হবে */
```

```
}
```

🔷 ইম্পর্টেন্ট কিছু সাজেশন

- রঙ নির্বাচন করুন এমন ভাবে যাতে টেক্সট পড়তে সুবিধা হয়।
- ফন্ট নির্বাচন করুন এমন যা পড়তে সহজ ও সুন্দর হয়।

📝 প্র্যাকটিস টাস্ক

1. একটি হেডিং তৈরি করুন:
    - টেক্সট কালার #FF5733 করুন
    - ফন্ট সাইজ 32px করুন
    - ফন্ট ওয়েট bold করুন
    - টেক্সট সেন্টার অ্যালাইন করুন
2. একটি প্যারাগ্রাফ তৈরি করুন:
    - গুগল ফন্ট থেকে 'Poppins' ফন্ট ব্যবহার করুন
    - লাইন হাইট 1.6 করুন
    - লেটার স্পেসিং 1px করুন
    - টেক্সট justify অ্যালাইন করুন

🤔 MCQ কুইজ

1. rgba() এ শেষের ভ্যালু কিসের জন্য ব্যবহৃত হয়?
    - a) রঙের গাঢ়তা
    - b) স্বচ্ছতা (opacity)
    - c) চওড়া
    - d) উচ্চতা
2. font-weight এর সর্বনিম্ন ভ্যালু কত?
    - a) 0
    - b) 1
    - c) 100
    - d) 200
3. line-height এর ডিফল্ট ভ্যালু কত?
    - a) 1
    - b) 1.2

- c) 2
- d) 0

4. text-transform: capitalize; কি করে?

- a) সব অক্ষর বড় হাতের করে
- b) সব অক্ষর ছোট হাতের করে
- c) প্রতি শব্দের প্রথম অক্ষর বড় হাতের করে
- d) কোন পরিবর্তন করে না

5. কোন ইউনিট রিলেটিভ নয়?

- a) em
- b) rem
- c) px
- d) %

6. HSL এর 'S' এর পূর্ণরূপ কি?

- a) Style
- b) Saturation
- c) System
- d) Screen

7. font-style: oblique; এর ফলাফল কি?

- a) টেক্সট বোল্ড হয়
- b) টেক্সট ইটালিক হয়
- c) টেক্সট আন্ডারলাইন হয়
- d) টেক্সট ছোট হয়

8. text-decoration: line-through; এর ফলাফল কি?

- a) টেক্সটের উপর লাইন
- b) টেক্সটের নিচে লাইন
- c) টেক্সটের মাঝ দিয়ে লাইন
- d) টেক্সটের পাশে লাইন

9. letter-spacing এর নেগেটিভ ভ্যালু দিলে কি হয়?

- a) অক্ষরগুলো দূরে সরে যায়
- b) অক্ষরগুলো কাছে সরে আসে
- c) কিছুই হয় না
- d) এরর দেখায়

10. @font-face কি জন্য ব্যবহৃত হয়?
    - a) ফন্ট সাইজ পরিবর্তন
    - b) কাস্টম ফন্ট লোড
    - c) ফন্ট কালার পরিবর্তন
    - d) ফন্ট স্টাইল পরিবর্তন

📚 হোমওয়ার্ক

1. একটি ব্লগ পোস্ট টেমপ্লেট তৈরি করুন:
    - হেডিং, সাবহেডিং এবং প্যারাগ্রাফ থাকবে
    - প্রতিটি এলিমেন্টে আলাদা ফন্ট ফ্যামিলি ব্যবহার করুন
    - কালার স্কিম ঠিক করুন (হেডিং, টেক্সট, ব্যাকগ্রাউন্ড)
    - টেক্সট ফরম্যাটিং প্রপার্টি ব্যবহার করুন
2. একটি গ্রেডিয়েন্ট ব্যাকগ্রাউন্ড তৈরি করুন:
    - লিনিয়ার গ্রেডিয়েন্ট
    - রেডিয়াল গ্রেডিয়েন্ট
    - মাল্টিপল কালার স্টপ
3. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৩: CSS সিলেক্টর এবং স্পেসিফিসিটি (CSS Selectors and Specificity)

CSS এর Selectors নির্ধারণ করে কোন HTML এলিমেন্টগুলোর ওপর নির্দিষ্ট স্টাইল প্রপার্টি Applay কড়া হয়। এ অধ্যায়ে আমরা বিভিন্ন ধরনের সিলেক্টর সম্পর্কে জানবো - যেমন এলিমেন্ট সিলেক্টর, ক্লাস, আইডি, এট্রিবিউট, সন্তান সিলেক্টর, ছদ্ম-ক্লাস (pseudo-class) এবং ছদ্ম-এলিমেন্ট (pseudo-element)। পাশাপাশি বুঝবো CSS specificity বা অগ্রাধিকার কিভাবে কাজ করে, সব কিছুই দেখবো এক সাথে।

CSS সিলেক্টর হলো এমন প্যাটার্ন যা HTML এলিমেন্টকে টার্গেট করে স্টাইল প্রয়োগ করে।

📊 সিলেক্টর টাইপ

সিলেক্টর টাইপ	সিনট্যাক্স	উদাহরণ	বর্ণনা
ইউনিভার্সাল	`*`	`* { margin: 0; padding: 0; }`	সব এলিমেন্টকে টার্গেট করে।

সিলেক্টর টাইপ	সিনট্যাক্স	উদাহরণ	বর্ণনা
টাইপ/এলিমেন্ট	`element`	`p { color: blue; }`	নির্দিষ্ট HTML ট্যাগের সকল এলিমেন্টকে টার্গেট করে।
ক্লাস	`.className`	`.header { font-size: 24px; }`	যে কোনো এলিমেন্টে মানিয়ে নেওয়া CSS ক্লাস যুক্ত সব এলিমেন্টকে টার্গেট করে।
আইডি	`#idName`	`#logo { width: 100px; }`	অনন্য আইডি যুক্ত একক এলিমেন্টকে টার্গেট করে।
অ্যাট্রিবিউট	`[attr]`	`a[target] { background-color: red; }`	নির্দিষ্ট অ্যাট্রিবিউট যুক্ত সব এলিমেন্টকে টার্গেট করে।
অ্যাট্রিবিউট মান	`[attr="val"]`	`[type="email"] { width: 100%; }`	নির্দিষ্ট অ্যাট্রিবিউট মান থাকা এলিমেন্টকে টার্গেট করে।
মান দিয়ে শুরু	`[attr^="foo"]`	`[href^="https"] { color: green; }`	অ্যাট্রিবিউট মান `foo` দিয়ে শুরু হলে টার্গেট করে।
মান দিয়ে শেষ	`[attr$=".pdf"]`	`[href$=".pdf"]::after { content: "📄"; }`	অ্যাট্রিবিউট মান `.pdf` দিয়ে শেষ হলে টার্গেট করে।
মান অন্তর্ভুক্ত	`[attr*="bar"]`	`[data-info*="bar"] { font-weight: bold; }`	অ্যাট্রিবিউট মানে `bar` স্ট্রিং কোথাও থাকলে টার্গেট করে।
ডেসেন্ড্যান্ট সিলেক্টর	`A B`	`div p { font-size: 100px; }`	`div` এর ভিতরে যেকোনো স্তরে থাকা `p` ট্যাগকে টার্গেট করে।
চাইল্ড সিলেক্টর	`A > B`	`div > p { font-size: 100px; }`	`div` এর সরাসরি সন্তান `p` ট্যাগকে টার্গেট করে।
অ্যাজেসেন্ট সিবলিং	`A + B`	`div + p { font-size: 100px; }`	`div` এর পরেই অবস্থিত প্রথম `p` ট্যাগকে টার্গেট করে।
জেনারেল সিবলিং	`A ~ B`	`div ~ p { font-size: 100px; }`	`div` এর পরের একই লেভেলের সব `p` ট্যাগকে টার্গেট করে।
পসুডো-ক্লাস সিলেক্টর	`A:hover`	`p:hover { color: red; }`	নির্দিষ্ট ইন্টারঅ্যাকশন/স্টেট অনুযায়ী এলিমেন্টকে টার্গেট করে।
পসুডো-এলিমেন্ট সিলেক্টর	`A::first-line`	`p::first-line { font-size: 120px; }`	এলিমেন্টের কোনো নির্দিষ্ট অংশ (প্রথম লাইন/প্রথম অক্ষর) টার্গেট করে।

🔷 কমন সিলেক্টর

```css
/* Tag/Element selector */
p {
  font-size: 120px;
}

/* Class selector */
.intro {
  color: blue;
}

/* ID selector */
#header {
  font-size: 20px;
}

/* Pseudo-class selector */
a:hover {
  text-decoration: underline;
}
```

🔄 কম্বিনেটর সিলেক্টর

```css
/* ডিসেনডেন্ট সিলেক্টর (স্পেস) */
div p { color: blue; }

/* চাইল্ড সিলেক্টর (>) */
div > p { color: red; }

/* অ্যাডজাসেন্ট সিবলিং (+) */
h1 + p { margin-top: 20px; }

/* জেনারেল সিবলিং (~) */
h1 ~ p { color: gray; }
```

🎯 স্যুডো-ক্লাস সিলেক্টর

```css
/* স্টেট-বেসড স্যুডো-ক্লাস */
a:hover { color: red; }
input:focus { border-color: blue; }
button:active { background: #ddd; }

/* স্ট্রাকচারাল স্যুডো-ক্লাস */
li:first-child { font-weight: bold; }
li:last-child { margin-bottom: 0; }
li:nth-child(odd) { background: #f4f4f4; }
```

🎯 স্যুডো-এলিমেন্ট সিলেক্টর

```
/* প্রথম লাইন এবং প্রথম লেটার */
p::first-line { font-weight: bold; }
p::first-letter { font-size: 2em; }

/* বিফোর এবং আফটার */
.quote::before { content: ""; }
.quote::after { content: ""; }
```

⚖️ স্পেসিফিসিটি (Specificity)

স্পেসিফিসিটি হলো CSS এর প্রায়োরিটি সিস্টেম। এটি নির্ধারণ করে যে কোন টাইপের সিলেক্টর কেমন প্রায়োরিটি পাবে।

স্পেসিফিসিটি ক্যালকুলেশন:

1. ইনলাইন স্টাইল = 1000
2. আইডি = 100
3. ক্লাস, অ্যাট্রিবিউট, স্যুডো-ক্লাস = 10
4. এলিমেন্ট, স্যুডো-এলিমেন্ট = 1

উদাহরণ:

```
/* স্পেসিফিসিটি = 1 */
p { color: black; }

/* স্পেসিফিসিটি = 10 */
.text { color: blue; }

/* স্পেসিফিসিটি = 100 */
#header { color: red; }

/* স্পেসিফিসিটি = 11 (1 এলিমেন্ট + 1 ক্লাস) */
p.text { color: green; }
```

💡 গুরুত্বপূর্ণ !important

```
p { color: red !important; } /* সর্বোচ্চ প্রায়োরিটি */
```

⚠️ সতর্কতা: !important ব্যবহার এড়ানো উচিত, কারণ এটি CSS এর স্বাভাবিক কাজের ধারা নষ্ট করে।

🎯 Pseudo-class ও Pseudo-element সিলেক্টরের অ্যাডভান্সড ব্যবহার

সিলেক্টর	সিনট্যাক্স	উদাহরণ	বর্ণনা
`:active`	`a:active`	`<a href="#">Click</a>`	ক্লিক ধরে রাখলে প্রয়োগ হয়।
`:hover`	`button:hover`	`<button>Hover Me</button>`	মাউস পয়েন্টার এলিমেন্টের ওপর গেলে স্টাইল চালু হয়।
`:focus`	`input:focus`	`<input type="text">`	কীবোর্ড বা ট্যাবে ফোকাস আসলে স্টাইলিং হয়।
`:visited`	`a:visited`	`<a href="#">Link</a>`	ইউজার ভিজিট করা লিঙ্কগুলোতে আলাদা স্টাইল দেয়।
`:first-child`	`p:first-child`	`<p>আমি প্রথম</p>`	প্যারেন্টের প্রথম সন্তান এলিমেন্ট হলে টার্গেট হয়।
`:last-child`	`p:last-child`	`<p>আমি শেষ</p>`	প্যারেন্টের শেষ সন্তান এলিমেন্ট হলে স্টাইল প্রয়োগ হয়।
`:nth-child(n)`	`li:nth-child(2)`	`<li>দ্বিতীয়</li>`	নির্দিষ্ট অবস্থানের সন্তান এলিমেন্ট টার্গেট করে।
`:nth-last-child(n)`	`li:nth-last-child(1)`	`<li>শেষ চাইল্ড</li>`	শেষ থেকে n-তম সন্তান এলিমেন্ট টুঁকে ধরে।
`:first-of-type`	`p:first-of-type`	`<p>প্রথম প্যারাগ্রাফ</p>`	প্যারেন্টে প্রথম `p` ট্যাগ হলে টার্গেট করে।
`:last-of-type`	`p:last-of-type`	`<p>শেষ প্যারাগ্রাফ</p>`	প্যারেন্টে শেষ `p` ট্যাগ হলে স্টাইল যায়।
`:nth-of-type(n)`	`div:nth-of-type(2)`	`<div>দ্বিতীয় ডিভ</div>`	একই টাইপের n-তম এলিমেন্টে স্টাইল দেয়।
`:only-child`	`p:only-child`	`<p>একইমাত্র চাইল্ড</p>`	প্যারেন্টের একমাত্র সন্তান হলে প্রয়োগ

সিলেক্টর	সিনট্যাক্স	উদাহরণ	বর্ণনা
			হয়।
`:only-of-type`	`span:only-of-type`	`<span>একমাত্র স্প্যান</span>`	ওই টাইপের একমাত্র এলিমেন্ট হলে টার্গেট করে।
`:checked`	`input:checked`	`<input type="checkbox" checked>`	চেক করা ইনপুটের জন্য আলাদা স্টাইল।
`:disabled`	`input:disabled`	`<input type="text" disabled>`	নিষ্ক্রিয় ইনপুটের স্টাইল নির্ধারণ করে।
`:enabled`	`input:enabled`	`<input type="text">`	সক্রিয় ইনপুটে স্টাইল প্রয়োগ হয়।
`:required`	`input:required`	`<input required>`	রিকোয়ার্ড ইনপুট ফিল্ড টার্গেট করে।
`:optional`	`input:optional`	`<input>`	অপশনাল ইনপুট ফিল্ডে স্টাইলিং যায়।
`:read-only`	`input:read-only`	`<input readonly>`	রিড-অনলি ইনপুটে আলাদা স্টাইল।
`:read-write`	`textarea:read-write`	`<textarea></textarea>`	লেখার জন্য খোলা এলিমেন্টে প্রয়োগ হয়।
`:target`	`#section:target`	`<div id="section"></div>`	URL ফ্র্যাগমেন্ট মিলে থাকা এলিমেন্টে ফোকাস।
`:empty`	`div:empty`	`<div></div>`	কোনো সন্তান না থাকলে স্টাইলিং হয়।
`:root`	`:root`	`<html>`	ডকুমেন্টের মূল (root) এলিমেন্ট।
`:not(selector)`	`p:not(.active)`	`p:not(.active) { opacity: .5; }`	নির্দিষ্ট সিলেক্টর বাদে বাকি সব টার্গেট করে।
`:has(selector)`	`article:has(img)`	`article:has(img) { display: grid; }`	ভেতরে নির্দিষ্ট চাইল্ড থাকলে প্যারেন্টকে টার্গেট করে।

সিলেক্টর	সিনট্যাক্স	উদাহরণ	বর্ণনা
`:is(sel, sel)`	`:is(h1,h2,h3)`	`:is(h1,h2,h3){ font-weight:600; }`	গ্রুপিং করে স্পেস রিলিফ, এতে specificity বাড়ে না।
`:where(sel)`	`:where(section)`	`:where(section){ margin-block:2rem; }`	নানাভাবে গ্রুপিং, specificity ০।
`::before`	`a::before`	`a::before { content: "▶"; }`	এলিমেন্টের কন্টেন্টের আগে কাস্টম কন্টেন্ট যোগ করে।
`::after`	`a::after`	`a::after { content: "↗"; }`	এলিমেন্টের কন্টেন্টের পরে কাস্টম কন্টেন্ট যোগ করে।
`::first-line`	`p::first-line`	`p::first-line { text-transform:uppercase; }`	টেক্সটের প্রথম লাইনের জন্য স্পেশাল স্টাইল।
`::first-letter`	`p::first-letter`	`p::first-letter { font-size:2rem; }`	টেক্সটের প্রথম অক্ষর বড় অথবা আলাদা স্টাইল।

🔷 Pseudo-class উদাহরণ:

```
/* :active */
/* যখন লিঙ্কে ক্লিক করে ধরে রাখা হয়, তখন এই স্টাইল প্রয়োগ হয় */
a:active {
  background-color: yellow;
}

/* :hover */
/* যখন মাউস নিয়ে যায়, তখন ব্যাকগ্রাউন্ড পরিবর্তন হয় */
button:hover {
  background-color: green;
}

/* :focus */
/* ইনপুট ফিল্ডে ক্লিক করলে বা টাইপ করতে গেলে স্টাইল প্রয়োগ হয় */
input:focus {
  border-color: blue;
}

/* :visited */
/* ব্যবহারকারী আগে ভিজিট করা লিঙ্কের রঙ পরিবর্তন করা যায় */
```

```css
a:visited {
  color: purple;
}

/* :first-child */
/* একটি এলিমেন্টের প্রথম চাইল্ড হলে স্টাইল প্রয়োগ হয় */
p:first-child {
  font-weight: bold;
}

/* :last-child */
/* একটি এলিমেন্টের শেষ চাইল্ড হলে স্টাইল প্রয়োগ হয় */
p:last-child {
  font-style: italic;
}

/* :nth-child */
/* নির্দিষ্ট সংখ্যার অবস্থানে থাকা এলিমেন্টকে স্টাইল করা হয় */
li:nth-child(2) {
  color: red;
}

/* :nth-last-child */
/* শেষে দিক থেকে গণনা করে নির্দিষ্ট অবস্থানের চাইল্ড */
li:nth-last-child(1) {
  background-color: lightgray;
}

/* :first-of-type */
/* একই টাইপের প্রথম এলিমেন্টে প্রয়োগ হয় */
p:first-of-type {
  color: orange;
}

/* :last-of-type */
/* একই টাইপের শেষ এলিমেন্টে প্রয়োগ হয় */
p:last-of-type {
  color: brown;
}

/* :only-child */
/* যখন একটি এলিমেন্ট তার প্যারেন্টের একমাত্র চাইল্ড হয় */
div:only-child {
  padding: 10px;
}

/* :checked */
/* চেক করা বক্স বা রেডিও বাটনে প্রয়োগ হয় */
input:checked {
  outline: 2px solid red;
}

/* :disabled */
```

```css
/* নিষ্ক্রিয় ইনপুট ফিল্ডে প্রয়োগ হয় */
input:disabled {
  background-color: #eee;
}

/* :required */
/* যেসব ইনপুট ফিল্ড বাধ্যতামূলক */
input:required {
  border: 2px solid green;
}

/* :optional */
/* ঐচ্ছিক ইনপুট ফিল্ডে প্রয়োগ হয় */
input:optional {
  border: 1px dashed blue;
}

/* :read-only */
/* যেখানে ইউজার লিখতে পারে না এমন ইনপুট ফিল্ডে প্রয়োগ হয় */
input:read-only {
  background-color: #f5f5f5;
}

/* :target */
/* URL এ #section থাকলে সেই এলিমেন্টে স্টাইল প্রয়োগ হয় */
#section:target {
  border: 3px solid blue;
}

/* :empty */
/* যেসব এলিমেন্ট একদম ফাঁকা থাকে */
div:empty {
  display: none;
}

/* :not */
/* যেসব এলিমেন্ট নির্দিষ্ট selector না */
p:not(.active) {
  opacity: 0.5;
}

/* :has */
/* যেসব এলিমেন্টের ভিতরে নির্দিষ্ট কিছু আছে */
div:has(img) {
  border: 2px dashed red;
}
```

ক্যালকুলেশন উদাহরণ

`.nav a:hover` → 10 (class) + 1 (element) + 10 (pseudo-class) = **21**
`#footer p::first-line` → 100 (ID) + 1 (element) + 1 (pseudo-element) = **102**

টাই-ব্রেকার

- যদি একই স্পেসিফিসিটি স্কোর হয়, **পরে** লোড হওয়া স্টাইল জিতবে।

⚔️ **Specificity War Tip**
`:where()` এর 0-ওজন গ্রুপ সিলেক্টর ব্যবহার করে, কাস্টম প্রপার্টির মাধ্যমে থিম সুইচ করতে পারো – স্পেসিফিসিটি ওয়্যার এড়াতে!

📌 ইম্পর্টেন্ট পয়েন্ট

- সর্বদা ID এর বদলে **ক্লাস** ব্যবহার করার চেষ্টা করুন।
- স্পেসিফিসিটি বুঝতে সমস্যা হলে ওয়েব টুল ব্যবহার করুন (Specificity Calculator)।

📝 প্র্যাকটিস টাস্ক

1. নিচের HTML স্ট্রাকচার ব্যবহার করে সিলেক্টর প্র্যাকটিস করুন:

```
<div class="container">
    <h1 id="title">মূল শিরোনাম</h1>
    <p class="text">প্রথম প্যারাগ্রাফ</p>
    <p class="text highlight">দ্বিতীয় প্যারাগ্রাফ</p>
    <ul>
        <li>প্রথম আইটেম</li>
        <li>দ্বিতীয় আইটেম</li>
        <li>তৃতীয় আইটেম</li>
    </ul>
</div>
```

2. উপরের HTML এ নিম্নলিখিত স্টাইল প্রয়োগ করুন:
    - সব টেক্সট এলিমেন্টের কালার সেট করুন
    - হাইলাইট ক্লাসের ব্যাকগ্রাউন্ড কালার সেট করুন
    - লিস্টের বিজোড় আইটেমগুলোর স্টাইল করুন
    - হভার ইফেক্ট যোগ করুন

🤔 MCQ কুইজ

1. স্পেসিফিসিটিতে আইডি সিলেক্টরের মান কত?

    - a) 1

- b) 10
- c) 100
- d) 1000

2. কোন সিলেক্টর সব এলিমেন্টকে সিলেক্ট করে?

- a) #
- b) .
- c) *
- d) >

3. p > span এর অর্থ কি?

- a) p এর সব span
- b) p এর ডাইরেক্ট চাইল্ড span
- c) p এর পরের span
- d) p এর আগের span

4. :nth-child(even) কি সিলেক্ট করে?

- a) প্রথম এলিমেন্ট
- b) শেষ এলিমেন্ট
- c) জোড় নাম্বার এলিমেন্ট
- d) বিজোড় নাম্বার এলিমেন্ট

5. ::before এবং ::after এ content প্রপার্টি:

- a) অপশনাল
- b) রিকোয়ার্ড
- c) ডিপ্রিকেটেড
- d) ইগনোর করা হয়

6. div + p কি সিলেক্ট করে?

- a) div এর সব p
- b) div এর পরের p
- c) div এর আগের p
- d) div এর ভিতরের p

7. [type="text"] কি ধরনের সিলেক্টর?

- a) ক্লাস সিলেক্টর
- b) আইডি সিলেক্টর
- c) অ্যাট্রিবিউট সিলেক্টর

- d) এলিমেন্ট সিলেক্টর

8. :hover স্যুডো-ক্লাস কখন কাজ করে?
   - a) এলিমেন্ট সিলেক্ট করলে
   - b) এলিমেন্ট ক্লিক করলে
   - c) মাউস এলিমেন্টের উপর নিলে
   - d) এলিমেন্ট ফোকাস করলে
9. স্পেসিফিসিটিতে ক্লাস সিলেক্টরের মান কত?
   - a) 1
   - b) 10
   - c) 100
   - d) 1000
10. !important এর ব্যবহার:
    - a) সর্বদা করা উচিত
    - b) কখনোই করা উচিত নয়
    - c) শুধু প্রয়োজনে করা উচিত
    - d) শুধু ডেভেলপমেন্টে করা উচিত

📚 হোমওয়ার্ক

1. একটি ন্যাভিগেশন মেনু তৈরি করুন:
   - মেনু আইটেম হোভার ইফেক্ট
   - অ্যাক্টিভ স্টেট স্টাইলিং
   - সাব-মেনু আইটেম
   - মোবাইল রেসপনসিভ
2. একটি ফর্ম তৈরি করুন যেখানে বিভিন্ন স্টেট স্টাইলিং থাকবে:
   - ইনপুট ফোকাস স্টেট
   - ভ্যালিড/ইনভ্যালিড স্টেট
   - চেকবক্স/রেডিও বাটন কাস্টম স্টাইল
   - প্লেসহোল্ডার স্টাইলিং
3. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৪: CSS বক্স মডেল (The CSS Box Model)

ওয়েব পেজের প্রতিটি এলিমেন্টকে একটি আয়তাকার বাক্স হিসেবে ধরা হয়, যার চারপাশে বিভিন্ন ধরণের স্পেস ও সীমা থাকতে পারে। CSS এর অত্যন্ত গুরুত্বপূর্ণ ধারণা হল Box Model. এটি বোঝায় প্রতিটি HTML উপাদান (যেমন একটি < div >, < p >, ইত্যাদি)আসলে একটি বাক্স যা চারটি স্তর নিয়ে গঠিত:

- Content (বিষয়বস্তু) – টেক্সট বা ছবি বা যা আছে এলিমেন্টের ভিতরে।
- Padding (অভ্যন্তরীণ মার্জিন) – বিষয়বস্তুর চারপাশের ফাঁকা জায়গা, যা ব্যাকগ্রাউন্ডের রঙ ধারণ করে।
- Border (সীমারেখা) – প্যাডিং-এর বাইরে ঘিরে থাকা সীমানা, যা আউটলাইনের মতো চারপাশে থাকে।
- Margin (বহিঃস্থ মার্জিন) – বর্ডারের বাইরের ফাঁকা স্থান, যা পরবর্তী অন্যান্য এলিমেন্ট থেকে দূরত্ব বজায় রাখতে সহায়ক।

এ অধ্যায়ে আমরা শিখবো কিভাবে margin, padding, border প্রপার্টি ব্যবহার করে element এর চারপাশের spacing ও বর্ডার নিয়ন্ত্রণ করতে হয়। এছাড়াও width, height, এবং CSS box-sizing প্রপার্টি সম্পর্কেও আলোচনা করবো।

📊 বক্স মডেল প্রপার্টি

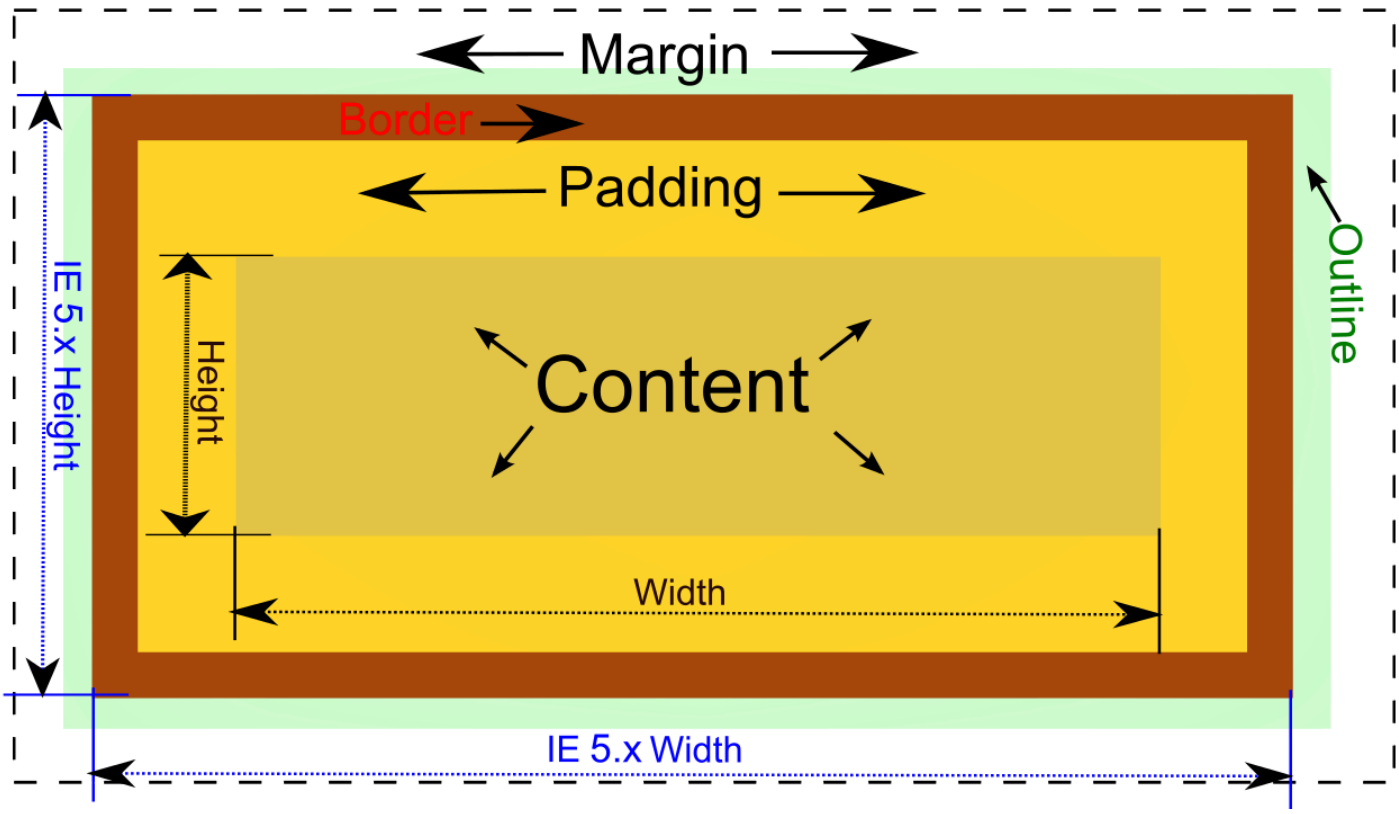
📦 সম্পূর্ণ বক্স-মডেল প্রপার্টি টেবিল

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
`margin`	`<length>`, `<percentage>`, `auto`	এলিমেন্টের বাইরের ফাঁকা জায়গা; `auto` দিলে ব্লক অনুভূমিক সেন্টারে আসে।
`margin-top`	`<length>`, `<percentage>`, `auto`	উপরের মার্জিন সেট করে।
`margin-right`	`<length>`, `<percentage>`, `auto`	ডান দিকে স্পেস কন্ট্রোল করে।
`margin-bottom`	`<length>`, `<percentage>`, `auto`	নীচের মার্জিন সেট করে।
`margin-left`	`<length>`, `<percentage>`, `auto`	বাম দিকে ফাঁক নির্ধারণ করে।
`padding`	`<length>`, `<percentage>`	কনটেন্ট ও বর্ডারের ভেতরের ফাঁকা জায়গা; `%` হলে প্যারেন্টের প্রস্থের সাথে স্কেল করে।
`padding-top`	`<length>`, `<percentage>`	উপরের প্যাডিং কণ্ঠে ফাঁক তৈরি করে।
`padding-right`	`<length>`, `<percentage>`	ডানপাশের প্যাডিং নির্ধারণ করে।
`padding-bottom`	`<length>`, `<percentage>`	নিচের প্যাডিং কনটেন্ট থেকে দূরত্ব বাড়ায়।
`padding-left`	`<length>`, `<percentage>`	বামপাশে কনটেন্টের পাশে ফাঁক তৈরি করে।

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
`border-width`	`<length>`, `thin`, `medium`, `thick`	বর্ডারের পুরুত্ব সেট করে; `0` দিলে বর্ডার অদৃশ্য হয়।
`border-style`	`none`, `solid`, `dashed`, `dotted`, `double`, `groove`, `ridge`, `inset`, `outset`	বর্ডারের ধরন নির্ধারণ করে।
`border-color`	`<color>`	বর্ডারের রঙ নির্ধারণ করে।
`border`	`width style color`	এক লাইনে বর্ডার-পুরুত্ব, স্টাইল ও রঙ সব সেট করে।
`border-radius`	`<length>`, `<percentage>`	এলিমেন্টের কোনগুলোকে ডিম-rounded করে; `50%` গেলে বৃত্ত হয় (square এ)।
`outline`	`width style color`	ফ্লো-নিরপেক্ষ আউটলাইন যোগ করে; ফোকাস ইউআইতে কাজে লাগে।
`outline-offset`	`<length>`	আউটলাইন বর্ডারের বাইরে কত দূরে থাকবে তা নির্ধারণ করে।
`box-sizing`	`content-box`, `border-box`	width/height গণনায় padding ও border কবে যোগ হবে তা ঠিক করে; `border-box` এ সব মিলে গণনা হয়।
`width`	`<length>`, `<percentage>`, `min-content`, `max-content`, `fit-content`, `auto`	এলিমেন্টের প্রস্থ সেট করে; flex এ `flex:0 0 X` এর মতো ভ্যালুও চলে।
`height`	`<length>`, `<percentage>`, `auto`	এলিমেন্টের উচ্চতা নির্ধারণ করে; `auto` হলে কনটেন্ট অনুযায়ী বাড়ে।
`min-width`	`<length>`, `<percentage>`	সর্বনিম্ন প্রস্থ সীমা নির্ধারণ করে।
`min-height`	`<length>`, `<percentage>`	সর্বনিম্ন উচ্চতা সীমা নির্ধারণ করে।
`max-width`	`<length>`, `<percentage>`	সর্বোচ্চ প্রস্থ সীমা নির্ধারণ করে।
`max-height`	`<length>`, `<percentage>`	সর্বোচ্চ উচ্চতা সীমা নির্ধারণ করে।
`overflow`	`visible`, `hidden`, `scroll`, `auto`, `clip`	অতিরিক্ত কন্টেন্ট দেখানোর নিয়ন্ত্রণ; `hidden/clip` কেটে ফেলবে, `scroll` স্ক্রলবার দেখাবে, `auto` প্রয়োজনমতো।
`overflow-x` / `overflow-y`	একই `overflow` ভ্যালু	নির্দিষ্ট এক্স (horizontal) বা ওয়াই (vertical) দিকের অতিরিক্ত কন্টেন্টের আচরণ কন্ট্রোল করে।

প্রপার্টি	সম্ভাব্য ভ্যালু	বর্ণনা
`box-shadow`	`none` , `<offset-x> <offset-y> <blur> <spread> <color>` , `inset`	এলিমেন্টের চারিপাশে ছায়া ফেলে; কার্ড-ডেপথ বা UI হাইলাইটে কাজে লাগে।
`box-decoration-break`	`slice` , `clone`	মাল্টি-লাইন ইনলাইন-বক্সে বর্ডার ও রেডিয়াস কিভাবে ব্রেক হবে তা নির্ধারণ করে।
`aspect-ratio`	`<ratio>` , `auto`	width/height অনুপাত লক করে; ইমেজ বা ভিডিও প্লেসহোল্ডারে কাজে লাগে।

🩺 বক্স-মডেল অ্যানাটমি



- **Content**: টেক্সট/ইমেজ—তোমার রিয়েল ডাটা
- **Padding**: কনটেন্ট আর বর্ডারের মাঝে একটু “ব্রিদিং রুম”
- **Border**: দৃশ্যমান “প্রতিরক্ষা-দেয়াল”
- **Margin**: প্রতিবেশীর সাথে সোশ্যাল ডিস্টেন্স

💡 **Debug Hack:**

```
* { outline: 1px solid red; }
```

calc() ফাংশন সহ height, width এবং padding এর উদাহরণ:

```
.box {
  width: calc(100% - 40px); /* প্যারেন্ট থেকে 40px বিয়োগ করে প্রস্থ নির্ধারণ */
  height: calc(50vh - 100px); /* ভিউপোর্ট হাইট থেকে নির্দিষ্ট মান বিয়োগ করে উচ্চতা নির্ধারণ */
  padding: calc(10px + 1em); /* ফিক্সড এবং রেসপনসিভ ইউনিট একসাথে মিশিয়ে প্যাডিং */
  border: 2px solid black;
  margin: 20px;
  box-sizing: border-box;
}
```

🔎 ব্যাখ্যা:

- calc() ফাংশনের মাধ্যমে আমরা dynamic মান নির্ধারণ করতে পারি।
- এটি বিভিন্ন unit (px, %, em, vh) একসাথে হিসাব করতে পারে।
- এই ফাংশন রেসপনসিভ ডিজাইনে অনেক কার্যকরী।

```
.box {
  width: 300px;
  height: 150px;
  padding: 20px;
  border: 2px solid black;
  margin: 15px;
  box-sizing: border-box;
}
```

🎯 বক্স সাইজিং

```
/* ডিফল্ট বক্স সাইজিং */
.content-box {
    box-sizing: content-box;
    width: 200px;
    padding: 20px;
    border: 5px solid #000;
    /* মোট width = 200px + (20px × 2) + (5px × 2) = 250px */
}

/* বর্ডার বক্স */
.border-box {
    box-sizing: border-box;
    width: 200px;
    padding: 20px;
    border: 5px solid #000;
    /* মোট width = 200px (padding এবং border সহ) */
}
```

📏 মার্জিন এবং প্যাডিং শর্টহ্যান্ড

```css
/* চার দিকে সমান */
margin: 10px;
padding: 10px;

/* উপর-নিচ | ডান-বাম */
margin: 10px 20px;
padding: 10px 20px;

/* উপর | ডান-বাম | নিচে */
margin: 10px 20px 15px;
padding: 10px 20px 15px;

/* উপর | ডান | নিচে | বাম */
margin: 10px 20px 15px 25px;
padding: 10px 20px 15px 25px;
```

🎨 বর্ডার স্টাইল

```css
/* বর্ডার শর্টহ্যান্ড */
border: 1px solid #000;

/* আলাদা আলাদা প্রপার্টি */
border-width: 1px;
border-style: solid;
border-color: #000;

/* স্পেসিফিক সাইড */
border-top: 1px solid #000;
border-right: 2px dashed #333;
border-bottom: 3px dotted #666;
border-left: 4px double #999;
```

🌟 বর্ডার স্টাইল টাইপ

স্টাইল	উদাহরণ
solid	———
dashed	----
dotted	....
double	═══
groove	3D গ্রুভ
ridge	3D রিজ

স্টাইল	উদাহরণ
inset	3D ইনসেট
outset	3D আউটসেট

🎨 বর্ডার রেডিয়াস

```
/* সব কর্নারে সমান */
border-radius: 10px;

/* আলাদা আলাদা কর্নার */
border-radius: 10px 20px 30px 40px;

/* বৃত্তাকার */
border-radius: 50%;
```

💡 নেগেটিভ মার্জিন

```
/* এলিমেন্টকে উপরে তুলে */
.element {
    margin-top: -20px;
}

/* এলিমেন্টকে বামে সরিয়ে */
.element {
    margin-left: -20px;
}
```

📌 ইম্পর্টেন্ট পয়েন্ট

- স্পেসিং সঠিকভাবে নির্ধারণ করতে প্যাডিং ও মার্জিন বুঝতে হবে।
- border-box ব্যবহার করলে সহজে বক্স সাইজ নিয়ন্ত্রণ করা যায়।

📝 প্র্যাকটিস টাস্ক

1. একটি কার্ড কম্পোনেন্ট তৈরি করুন:

```
<div class="card">
    <img src="image.jpg" alt="Card Image">
    <h3>কার্ড টাইটেল</h3>
    <p>কার্ড ডেসক্রিপশন টেক্সট</p>
    <button>আরও দেখুন</button>
</div>
```

2. কার্ডে নিম্নলিখিত স্টাইল প্রয়োগ করুন:
    - ফিক্সড width এবং height
    - প্যাডিং এবং মার্জিন
    - বর্ডার এবং বর্ডার রেডিয়াস
    - শ্যাডোইফেক্ট
    - হোভার ইফেক্ট

🤔 MCQ কুইজ

1. box-sizing: border-box; এর কাজ কি?

    - a) শুধু কনটেন্ট সাইজ গণনা করে
    - b) padding এবং border সহ width গণনা করে
    - c) শুধু margin গণনা করে
    - d) কোনটিই না

2. margin: 10px 20px এর অর্থ কি?

    - a) সব দিকে 10px
    - b) উপরে-নিচে 10px, ডানে-বামে 20px
    - c) উপরে-নিচে 20px, ডানে-বামে 10px
    - d) শুধু উপরে 10px, নিচে 20px

3. border-radius: 50%; দিলে কি হয়?

    - a) বর্গাকার হয়
    - b) বৃত্তাকার হয়
    - c) ত্রিভুজাকার হয়
    - d) আয়তাকার হয়

4. কোন প্রপার্টি নেগেটিভ ভ্যালু নিতে পারে?

    - a) padding
    - b) margin
    - c) border
    - d) width

5. margin: auto; এর ব্যবহার কি?

    - a) এলিমেন্টকে ডানে নেয়
    - b) এলিমেন্টকে বামে নেয়
    - c) এলিমেন্টকে হরাইজন্টালি সেন্টার করে
    - d) এলিমেন্টকে ভার্টিকালি সেন্টার করে

6. border শর্টহ্যান্ডে কয়টি ভ্যালু থাকে?
    - a) 2
    - b) 3
    - c) 4
    - d) 5
7. margin collapse কি?
    - a) মার্জিন যোগ হওয়া
    - b) বড় মার্জিন জিতে যাওয়া
    - c) মার্জিন বিয়োগ হওয়া
    - d) মার্জিন গুণ হওয়া
8. width: 100%; এর অর্থ কি?
    - a) 100 পিক্সেল
    - b) প্যারেন্ট এর সম্পূর্ণ প্রস্থ
    - c) ভিউপোর্ট এর সম্পূর্ণ প্রস্থ
    - d) 100 ইউনিট
9. inline এলিমেন্টে কোনটি কাজ করে না?
    - a) padding
    - b) margin-left
    - c) margin-top
    - d) border
10. box-shadow এর কতটি ভ্যালু থাকতে পারে?
    - a) 2
    - b) 3
    - c) 4
    - d) 2 থেকে 6

📚 হোমওয়ার্ক

1. একটি প্রোডাক্ট গ্রিড লেআউট তৈরি করুন:
    - প্রতিটি প্রোডাক্ট কার্ডে ইমেজ থাকবে
    - প্রোডাক্ট টাইটেল, প্রাইস, এবং বাটন থাকবে
    - কার্ডে শ্যাডো এবং হোভার ইফেক্ট থাকবে
    - রেসপন্সিভ গ্রিড লেআউট

2. একটি কনটেন্ট সেকশন তৈরি করুন:
    - ফিক্সড width কনটেইনার
    - প্যাডিং এবং মার্জিন ব্যবহার
    - বর্ডার এবং বর্ডার রেডিয়াস
    - নেস্টেড এলিমেন্টের স্পেসিং
3. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৫: লেআউট ফান্ডামেন্টালস: ডিসপ্লে এবং ফ্লোট (Layout Fundamentals: Display and Floats)

CSS-এ লেআউট কন্ট্রোলের জন্য কিছু গুরুত্বপূর্ণ প্রপার্টি হলো display, float এবং position (পরবর্তী অধ্যায়ে আলোচনা করা হবে)। display দিয়ে ঠিক করা হয় কোন উপাদান ব্লক স্তরের হবে না inline, float দিয়ে উপাদানকে সাধারণ প্রবাহ থেকে আলাদা করে বাঁ বা ডানদিকে অবস্থান করানো হয়, এবং clear দিয়ে float ক্লিয়ার করা হয়। এছাড়া visibility দিয়ে উপাদান লুকানো যায় (জায়গা রেখে) এবং vertical-align ইনলাইন উপাদানের উল্লম্ব সারিবদ্ধতা নিয়ন্ত্রণ করে। নিচে উদাহরণসহ এদের ব্যবহার দেখা যাক।

📊 লে-আউট ফান্ডামেন্টালস: Display & Floats

প্রপার্টি	Core Values / Units	বর্ণনা	Quick Tips
`display`	`none`, `block`, `inline`, `inline-block`, `flex`, `inline-flex`, `grid`, `inline-grid`, `table`, `table-row`, `table-cell`, `list-item`, `flow-root`, `contents`, `subgrid`, `ruby`, `inline ruby`	একটি এলিমেন্ট কীভাবে লেআউট ফ্লোতে হাজির হবে তা নির্ধারণ করে	`flow-root` দিয়ে সহজে BFC তৈরি করে, clearfix-হ্যাক ছাড়াই কাজ করে
`float`	`left`, `right`, `inline-start`, `inline-end`, `none`	এলিমেন্টকে ফ্লো থেকে তুলে ভাসমান অবস্থায় রাখে, চারপাশে টেক্সট ঘিরে যায়	ছবির পাশে টেক্সট-র‍্যাপ বা পুরনো 2-কলাম লেআউটে কাজে লাগবে
`clear`	`none`, `left`, `right`, `both`, `inline-start`, `inline-end`	পরবর্তী এলিমেন্টকে ফ্লোটেড উপাদানের নিচে নামিয়ে সাফ ফ্লো বজায় রাখে	ফুটারে `clear: both;` ব্যবহার করে ক্লাসিক clearfix প্রয়োগ করুন
`overflow`	`visible`, `hidden`, `auto`, `scroll`, `clip`	বক্স সীমা ছাড়িয়ে কন্টেন্ট কীভাবে দেখানো হবে তা নিয়ন্ত্রণ করে; BFC তৈরি করতে পারে	`overflow: auto` বা `hidden` = দ্রুত BFC = clearfix-এর সহজ বিকল্প

প্রপার্টি	Core Values / Units	বর্ণনা	Quick Tips
`vertical-align`	`baseline`, `top`, `middle`, `bottom`, `<length>`	ইনলাইন কিংবা টেবিল সেলে উপাদানগুলোকে উল্লম্বভাবে ঠিক অবস্থানে আনে	`inline-block` এ স্থায়ী লাইন-উলম্বতা পেতে প্রয়োগ করুন

> 💡 Block Formatting Context (BFC) Cheat-Sheet: `overflow:hidden|auto`, `float`, `display:inline-block` / `flow-root`, `position: absolute`, `contain: layout` —সবাই নতুন আলাদা ফ্লো গড়ে, মার্জিন-কলাপ্স আটকায়, ফ্লোট ক্লিয়ার করে।

🔍 Display এর মূল মন্ত্র

- **Block-level**: নতুন লাইনে শুরু করে, পুরো প্রস্থ জুড়ে বিস্তৃত হয়; শেষে স্বয়ংক্রিয় লাইন-ব্রেক।
- **Inline**: টেক্সটের সঙ্গে উড়তে থাকে—প্রস্থ ও উচ্চতা না দেখে, শুধু কন্টেন্টের জায়গাটা ঘিরে নেয়।
- **Inline-block**: ইনলাইনের মতো ফ্লো মেনে বসে, কিন্তু নিজস্ব প্রস্থ-উচ্চতা থাকতে পারে।
- **Flex & Grid**: আধুনিক এক-মাত্রিক (Flex) ও দুই-মাত্রিক (Grid) লেআউট ইঞ্জিন; ডিটেইলে পরে ডাইভ করব।
- **Flow-root**: নিজে একটা Block Formatting Context তৈরি করে, ভাসমান সন্তানদের উচ্চতা ধরার ট্রেন্ডি সমাধান।
- **Display Switch**: – মিডিয়া-কোয়েরিতে `display: block ↔ none` ধরে ন্যাভ টগ্‌ল হ্যাক।

📦 ব্লক লেভেল এলিমেন্ট

```css
div {
    display: block;
    width: 100%;
    height: 100px;
    margin: 10px 0;
}
```

📝 ইনলাইন এলিমেন্ট

```css
span {
    display: inline;
    /* width এবং height কাজ করবে না */
    padding: 0 10px;
}
```

🔄 ইনলাইন-ব্লক

```css
.inline-block {
    display: inline-block;
    width: 150px;
```

```
    height: 150px;
    margin: 10px;
}
```

🌊 ফ্লোট প্রপার্টি

ফ্লোট বেসিক

```
.float-left {
    float: left;
    width: 200px;
    margin-right: 15px;
}

.float-right {
    float: right;
    width: 200px;
    margin-left: 15px;
}
```

ক্লিয়ার ফিক্স

```
/* ক্লিয়ার ফিক্স */
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
```

📌 `display: block` এর ধর্ম

✅ ধর্ম (Behavior):

- একটি নতুন লাইনে শুরু হয়।
- পুরো প্রস্থ (parent container যতটুকু চওড়া) দখল করে।
- `height`, `width`, `margin`, `padding` সবকিছু সহজে কাজ করে।

✅ Padding সম্পর্ক:

- যেকোনো দিকে `padding` দিলে, সেই দিকের স্পেস তৈরি হয়।
- যেমনঃ উপর, নিচ, ডান, বাম — সবদিকের `padding` কাজ করে পুরোপুরি।

✅ উদাহরণ:

```
<div style="display: block; padding: 20px; background: lightblue;">
  আমি block এলিমেন্ট
</div>
```

📌 `display: inline` এর ধর্ম

✅ ধর্ম (Behavior)

- লাইনের **মধ্যেই** বসে (line-এর অংশ)
- শুধুমাত্র **যতটুকু কনটেন্ট আছে**, ততটুকু জায়গা নেয়
- `height` ও `width` মানে পরিবর্তন আনা যায় না

✅ Padding সম্পর্ক

- উপর-নিচে `padding` ভিজুয়ালি দেখা গেলেও আসলে কাজ করে না
- ডান-বাম `padding` ঠিকভাবে কাজ করে

✅ উদাহরণ

```
<span style="display: inline; padding: 10px 20px; background: pink;">
  আমি inline এলিমেন্ট
</span>
```

📌 `display: inline-block` এর ধর্ম

✅ ধর্ম (Behavior)

- `inline`-এর মতো **লাইনের মধ্যে** থাকে
- কিন্তু `block`-এর মতো `height`, `width`, `padding`, `margin`—সবই গ্রহণ করে

✅ Padding সম্পর্ক

- **চার দিকের** `padding`-ই সম্পূর্ণভাবে কাজ করে
- ইচ্ছেমতো `height` এবং `width` নির্ধারণ করা যায়
- Perfect balance between **inline** and **block** 🚀

✅ উদাহরণ

```
<span style="display: inline-block; padding: 15px; background: yellow;">
  আমি inline-block এলিমেন্ট
```

```
</span>
```

🔍 তুলনামূলক Chart:

টাইপ	নতুন লাইন	Width/Height কাজ করে	Padding (Top/Bottom)	Padding (Left/Right)	মন্তব্য
`block`	✅	✅	✅	✅	ফুল কাস্টমাইজড এলিমেন্ট
`inline`	❌	❌	❌ (ভিজুয়াল মাত্র)	✅	ছোট অংশ, স্টাইলিং সীমিত
`inline-block`	❌	✅	✅	✅	প্রফেশনাল ইউআই তৈরিতে ব্যবহৃত

🧠 মনে রাখার মন্ত্র:

- **block — রাজা** 👑 : নিজের লাইনে বসে, নিজ লাইনে সব জায়গা নিয়ে নেয়।
- **inline — ভদ্রলোক** 🤵 : লাইনের ভেতর শান্তভাবে থাকে, বেশি চাই না, যতটুকু এলিমেন্ট তাই জায়গা নেই
- **inline-block — স্মার্ট গ্যাং** 😎 : লাইনের মধ্যে থেকেও নিজের স্টাইল রাখতে পারে

📋 কমন কিছু HTML Element Display টাইপ এর লিস্ট

এলিমেন্ট	টাইপ	ব্যাখ্যা
`<div>`	Block	পূর্ণ প্রস্থে থাকে, নতুন লাইনে শুরু হয়। সাধারণত লেআউট নির্মাণে ব্যবহৃত হয়।
`<p>`	Block	একটি প্যারাগ্রাফ উপস্থাপন করে। নতুন লাইনে শুরু হয়।
`<h1> – <h6>`	Block	হেডিং উপাদান; প্রতিটি নিজস্ব ব্লক তৈরি করে।
`<section>`	Block	একটি স্বতন্ত্র সেকশন উপস্থাপন করে। নতুন লাইনে শুরু হয়।
`<article>`	Block	একটি স্বয়ংসম্পূর্ণ কনটেন্ট ব্লক।
`<nav>`	Block	নেভিগেশন লিংক এরিয়া বোঝাতে ব্যবহৃত হয়।
`<ul>`, `<ol>`, `<li>`	Block	লিস্ট তৈরি করে এবং প্রতিটি আইটেম নিজস্ব লাইনে থাকে।
`<form>`	Block	ইনপুট ফিল্ড নিয়ে একটি ফর্ম তৈরি করে।

এলিমেন্ট	টাইপ	ব্যাখ্যা
`<span>`	Inline	কনটেন্ট লাইনের মধ্যে থাকে। সাধারণত ছোট অংশ স্টাইল করতে ব্যবহৃত হয়।
`<a>`	Inline	লিংক তৈরি করে এবং লাইনের মধ্যে থাকে।
`<strong>`, `<em>`	Inline	টেক্সট হাইলাইট বা এমফাসিস করতে ব্যবহৃত হয়।
`<img>`	Inline	ইমেজ উপস্থাপন করে, টেক্সট লাইনের মধ্যে থাকে।
`<input>`	Inline	ইউজার ইনপুটের জন্য ব্যবহৃত হয়, সাধারণত লাইনের মধ্যেই থাকে।
`<label>`	Inline	ইনপুটের জন্য লেবেল উপস্থাপন করে।
`<abbr>`	Inline	সংক্ষিপ্ত রূপ বোঝাতে ব্যবহৃত হয়।

এলিমেন্ট	টাইপ	ব্যাখ্যা
`<button>`	Inline-block	একটি বাটন উপস্থাপন করে, যা ইনলাইন মতো বসে কিন্তু ব্লক এলিমেন্টের মতো dimension সেট করা যায়।
`<input type="submit">`	Inline-block	ইনলাইন অবস্থানে থাকে কিন্তু `width`, `height` দেয়া যায়।
`<img>`	Inline-block	ইনলাইন হিসেবে কাজ করে কিন্তু dimension পরিবর্তন করা যায়।
`<select>`	Inline-block	ড্রপডাউন তৈরি করে, ব্লকের মতো সাইজ দেয়া যায়।

📌 ইম্পর্টেন্ট পয়েন্ট

- Display প্রপার্টি দিয়ে এলিমেন্টের বিন্যাস খুব সহজে পরিবর্তন করা যায়।
- Floats ব্যবহারের সময় clearfix ক্লাস ব্যবহার করতে ভুলবেন না।

📝 প্র্যাকটিস টাস্ক

1. একটি দুই-কলাম লেআউট তৈরি করুন:

```
<div class="container clearfix">
    <div class="sidebar">সাইডবার কনটেন্ট</div>
    <div class="main-content">মূল কনটেন্ট</div>
</div>
```

2. একটি হেডার লেআউট তৈরি করুন:

```
<header class="sticky-header">
    <nav>
        <ul>
            <li><a href="#">হোম</a></li>
            <li><a href="#">সার্ভিস</a></li>
            <li><a href="#">যোগাযোগ</a></li>
        </ul>
    </nav>
</header>
```

🤔 MCQ কুইজ

1. display: none; এবং visibility: hidden; এর পার্থক্য কি?
    - a) কোন পার্থক্য নেই
    - b) none স্পেস নেয় না, hidden স্পেস নেয়
    - c) none অদৃশ্য করে, hidden মুছে ফেলে
    - d) উভয়ই স্পেস নেয়

2. position: absolute; এলিমেন্টের রেফারেন্স কি?
    - a) ব্রাউজার উইন্ডো
    - b) নিকটতম positioned প্যারেন্ট
    - c) সবসময় body
    - d) নিজের পজিশন

3. z-index কাজ করে না কেন?
    - a) position: static; থাকলে
    - b) float ব্যবহার করলে
    - c) display: block; থাকলে
    - d) margin দিলে

4. sticky position কখন কাজ করে?
    - a) স্ক্রল করার আগে
    - b) স্ক্রল করার পরে
    - c) স্ক্রল threshold পৌঁছালে
    - d) কখনোই না

5. float: left; দেওয়ার পর এলিমেন্ট:
    - a) ব্লক হয়ে যায়
    - b) ইনলাইন হয়ে যায়

- c) ইনলাইন-ব্লক হয়ে যায়
- d) ব্লক-লাইক আচরণ করে

6. inline এলিমেন্টে কোনটি কাজ করে না?
   - a) padding-left
   - b) margin-top
   - c) border
   - d) padding-right
7. clearfix কেন ব্যবহার করা হয়?
   - a) ফ্লোট এলিমেন্ট মুছে ফেলতে
   - b) ফ্লোট এলিমেন্ট কনটেইন করতে
   - c) ফ্লোট এলিমেন্ট সেন্টার করতে
   - d) ফ্লোট এলিমেন্ট হাইড করতে
8. position: fixed; এলিমেন্ট কার সাপেক্ষে পজিশন নেয়?
   - a) প্যারেন্ট এলিমেন্ট
   - b) ভিউপোর্ট
   - c) body
   - d) html
9. display: inline-block; এর বৈশিষ্ট্য:
   - a) width/height নেয় না
   - b) নতুন লাইন নেয়
   - c) width/height নেয় + লাইনের মধ্যে থাকে
   - d) শুধু width নেয়
10. ফ্লোট এলিমেন্টের বৈশিষ্ট্য:
    - a) normal flow এ থাকে
    - b) normal flow থেকে বের হয়ে যায়
    - c) position fixed হয়ে যায়
    - d) display none হয়ে যায়

📚 হোমওয়ার্ক

1. একটি ব্লগ লেআউট তৈরি করুন:
   - হেডার (sticky)
   - মেইন কনটেন্ট (2/3 width)

- সাইডবার (1/3 width)
- ফুটার

2. একটি কার্ড গ্রিড তৈরি করুন:
   - ফ্লোট ব্যবহার করে
   - প্রতি সারিতে 3টি কার্ড
   - রেসপন্সিভ ব্যবহার
3. একটি ফিক্সড সোশ্যাল মিডিয়া আইকন বার:
   - স্ক্রিনের ডান পাশে
   - ভার্টিকাল লিস্ট
   - হোভার ইফেক্ট
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৬: CSS পজিশনিং এবং Z-Index (CSS Positioning and Z-Index)

CSS তে সঠিক পজিশনিং জানা খুব গুরুত্বপূর্ণ। ধরি, আপনি একটি ছবিকে লেখার ওপর বসাতে চান, বা স্ক্রল করলেও একটি মেনু উপরে আটকে রাখতে চান। এই ক্ষেত্রে position প্রপার্টি এবং তার বিভিন্ন মান বুঝতে হবে।

🗂️ সম্পূর্ণ প্রপার্টি

Property	Core Values / Units	ব্যাখ্যা	Quick Tips
`position`	`static`, `relative`, `absolute`, `fixed`, `sticky`	এলিমেন্টের নরমাল ফ্লো এবং রেফারেন্স ফ্রেম ঠিক করে। `sticky` হলে থ্রেশহোল্ড পার করে `fixed` হয়।	শুধুমাত্র পজিশন্ড, `flex` অথবা `grid` কন্টেইনারে কাজ করে।
`top/right/bottom/left`	`<length>`, `%`, `auto`	পজিশনড প্যারেন্টের রেফারেন্স ফ্রেমের উপর ভিত্তি করে অফসেট দেয়।	ব্যবহার করুন `inset` শর্টহ্যান্ড; logical ভার্সন আছে— `inset-block` / `inset-inline`।
`inset`	`<length>`, `%`, `auto`	চার দিকের logical/off-axis শর্টহ্যান্ড অফসেট।	`inset: 0` দিলে পুরো এলিমেন্ট স্ট্রেচ এবং কভার করে।

Property	Core Values / Units	ব্যাখ্যা	Quick Tips
`z-index`	`<integer>`, `auto`	স্ট্যাকিং অর্ডার নির্ধারণ করে; বড় মান উপরের লেয়ারে চলে যায়।	কাজ করে পজিশন্ড, `flex` বা `grid` কন্টেইনারের মধ্যে।
`contain`	`layout`, `paint`, `size`, `strict`, `content`	নতুন স্ট্যাকিং কন্টেক্সট তৈরি করে এবং পারফরম্যান্স বুস্ট করে।	বড় UI মডিউলে boundary আলাদা রাখতে ব্যবহার করুন।
`isolation`	`auto`, `isolate`	এলিমেন্টের জন্য ফোর্সড স্ট্যাকিং কন্টেক্সট তৈরি করে।	গ্রাফিক্স ব্লেন্ড মোড শিল্ড হিসেবে কাজে লাগে।

📍 Positioning Proparty

CSS পজিশনিং এর সাহায্যে এলিমেন্টের অবস্থান নির্ধারণ করা যায়।

Property Name	Property Value	বর্ণনা / ব্যাখ্যা
`position`	`static`, `relative`, `absolute`, `fixed`, `sticky`	এলিমেন্টের পজিশন নির্ধারণ
`top`	`px`, `%`, `em`	এলিমেন্টের উপর থেকে দূরত্ব
`right`	`px`, `%`, `em`	এলিমেন্টের ডান থেকে দূরত্ব
`bottom`	`px`, `%`, `em`	এলিমেন্টের নিচ থেকে দূরত্ব
`left`	`px`, `%`, `em`	এলিমেন্টের বাম থেকে দূরত্ব

🌌 Position নিয়ে আরও কিছু

- **static (ডিফল্ট)** ডকুমেন্ট ফ্লো পাল্টায় না; `top/right/bottom/left` কাজ করে না।
- **relative** নিজস্থানে থেকে অফসেট দেয়; `left/right/top/bottom` দিয়ে সরে যায়।
- **absolute** নিকটতম non-`static` প্যারেন্টের উপর পজিশনড হয়; ফ্লো থেকে বাদ পড়ে।
- **fixed** ভিউপোর্টে পিন হয়ে যায়; স্ক্রল এ অবস্থান অপরিবর্তিত থাকে।
- **sticky** প্রথমে `relative`; স্ক্রল থ্রেশহোল্ড পার হলে `fixed`; প্যারেন্টের স্ক্রোল বাউন্ডিতে আটকে থাকে।

🎨 পজিশন অফসেট প্রপার্টি

```
.element {
    position: relative;
    top: 10px;      /* উপর থেকে দূরত্ব */
    right: 20px;    /* ডান থেকে দূরত্ব */
    bottom: 30px;   /* নিচ থেকে দূরত্ব */
    left: 40px;     /* বাম থেকে দূরত্ব */
}
```

🔄 রিলেটিভ পজিশনিং

```
.parent {
    position: relative;
    height: 200px;
}

.child {
    position: relative;
    top: 20px;
    left: 20px;
    /* মূল পজিশন থেকে 20px উপরে এবং বামে সরবে */
}
```

📍 অ্যাবসলিউট পজিশনিং

```
.parent {
    position: relative;
    /* অ্যাবসলিউট চাইল্ডের রেফারেন্স পয়েন্ট */
}

.child {
    position: absolute;
    top: 0;
    right: 0;
    /* প্যারেন্টের উপর-ডান কোণায় বসবে */
}
```

📌 ফিক্সড পজিশনিং

```
.header {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    /* স্ক্রল করলেও এখানেই থাকবে */
}
```

```
.back-to-top {
    position: fixed;
    bottom: 20px;
    right: 20px;
    /* স্ক্রলে স্থির থাকবে */
}
```

🎯 স্টিকি পজিশনিং

```
.section-header {
    position: sticky;
    top: 0;
    background: white;
    /* স্ক্রল করলে top: 0 পজিশনে আটকে যাবে */
}
```

🎚️ Z-Index (Layering)

Z-Index এর সাহায্যে এলিমেন্টের স্ট্যাকিং অর্ডার বা লেয়ার নির্ধারণ করা যায়।

Property Name	Property Value	বর্ণনা / ব্যাখ্যা
`z-index`	number (integer)	লেয়ার বা স্ট্যাকিং অর্ডার নির্ধারণ

📊 Z-Index এবং স্ট্যাকিং কনটেক্সট

```
/* স্ট্যাকিং কনটেক্সট */
.parent {
    position: relative;
    z-index: 1;
}

/* চাইল্ড এলিমেন্ট */
.child-1 {
    z-index: 2;
    /* প্যারেন্টের স্ট্যাকিং কনটেক্সটের মধ্যে */
}

.child-2 {
    z-index: 3;
    /* child-1 এর উপরে থাকবে */
}
```

🎨 স্ট্যাকিং কনটেক্সট হায়ারার্কি

1. ফর্মাল স্ট্যাকিং কনটেক্সট:

- root এলিমেন্ট (HTML)
- position: relative/absolute + z-index
- position: fixed/sticky
- opacity < 1
- transform
- filter
- isolation: isolate

📝 প্র্যাকটিস টাস্ক

1. একটি মোডাল উইন্ডো তৈরি করুন:

```
<div class="modal-overlay">
    <div class="modal">
        <h2>মোডাল টাইটেল</h2>
        <p>মোডাল কনটেন্ট...</p>
        <button class="close">বন্ধ করুন</button>
    </div>
</div>
```

```
.modal-overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0, 0, 0, 0.5);
    z-index: 1000;
}

.modal {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background: white;
    padding: 20px;
    z-index: 1001;
}
```

2. একটি ড্রপডাউন মেনু তৈরি করুন:

```
.dropdown {
    position: relative;
}
```

```css
.dropdown-content {
    position: absolute;
    top: 100%;
    left: 0;
    display: none;
}

.dropdown:hover .dropdown-content {
    display: block;
}
```

📌 ইম্পর্টেন্ট পয়েন্ট

- relative এবং absolute পজিশনিংয়ের পার্থক্য খুব ভাল করে বুঝতে হবে।
- Z-Index শুধুমাত্র position নির্ধারিত এলিমেন্টে কাজ করে।

🤔 MCQ কুইজ

1. position: sticky; কাজ করার জন্য কি প্রয়োজন?
    - a) z-index
    - b) fixed parent
    - c) top/bottom/left/right value
    - d) transform
2. অ্যাবসলিউট পজিশনড এলিমেন্ট যদি কোন positioned প্যারেন্ট না পায়:
    - a) স্ট্যাটিক হয়ে যায়
    - b) রিলেটিভ হয়ে যায়
    - c) initial position এ থাকে
    - d) viewport এর সাপেক্ষে পজিশন নেয়
3. z-index নেগেটিভ ভ্যালু দিলে:
    - a) এরর দেখায়
    - b) কাজ করে না
    - c) স্ট্যাকে নিচে চলে যায়
    - d) স্ট্যাকে উপরে ওঠে
4. কোন পজিশন ভ্যালুতে z-index কাজ করে না?
    - a) relative
    - b) absolute
    - c) static
    - d) fixed

5. transform প্রপার্টি ব্যবহার করলে:
    - a) z-index কাজ করে না
    - b) নতুন স্ট্যাকিং কনটেক্সট তৈরি হয়
    - c) পজিশন fixed হয়ে যায়
    - d) পজিশন static হয়ে যায়
6. position: fixed; এর ক্ষেত্রে transform ব্যবহার করলে:
    - a) fixed behavior থাকে না
    - b) z-index কাজ করে না
    - c) স্ক্রলে মুভ করে
    - d) কোন প্রভাব পড়ে না
7. স্টিকি পজিশনে এলিমেন্ট:
    - a) সবসময় fixed থাকে
    - b) কখনোই স্ক্রল করে না
    - c) threshold পর্যন্ত স্ক্রল করে
    - d) শুধু উপরে যায়
8. position: relative; দিলে এলিমেন্ট:
    - a) normal flow থেকে বের হয়ে যায়
    - b) normal flow তে থাকে
    - c) fixed হয়ে যায়
    - d) absolute হয়ে যায়
9. পজিশন offset (top, right, bottom, left) কাজ করে:
    - a) শুধু px ইউনিটে
    - b) শুধু % ইউনিটে
    - c) যেকোন ভ্যালিড length ইউনিটে
    - d) শুধু em/rem ইউনিটে
10. স্ট্যাকিং কনটেক্সট সম্পর্কে কোনটি সত্য?
    - a) শুধু z-index দিলেই তৈরি হয়
    - b) position: relative দিলেই তৈরি হয়
    - c) নির্দিষ্ট প্রপার্টি কম্বিনেশনে তৈরি হয়
    - d) সব এলিমেন্টেই থাকে

📚 হোমওয়ার্ক

1. একটি কমপ্লেক্স ন্যাভিগেশন মেনু তৈরি করুন:
    - মেইন মেনু আইটেম
    - ড্রপডাউন সাবমেনু
    - মেগা মেনু
    - মোবাইল রেসপন্সিভ
2. একটি ইমেজ গ্যালারি তৈরি করুন:
    - গ্রিড লেআউট
    - ইমেজ ক্লিক করলে লাইটবক্স
    - লাইটবক্সে নেভিগেশন বাটন
    - ক্লোজ বাটন
3. একটি ফিক্সড ফুটার তৈরি করুন:
    - কনটেন্ট এরিয়া ছোট হলেও bottom এ থাকবে
    - স্ক্রল করলে position থাকবে
    - রেসপন্সিভ ডিজাইন
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৭: লিস্ট এবং ন্যাভিগেশন মেনু স্টাইলিং (Styling Lists and Navigation Menus)

HTML এ তালিকা (< ul > এবং < ol >) অনেক সময় নেভিগেশন মেনু, সাইডবার, ফুটারের লিঙ্ক ইত্যাদি তৈরি করতে ব্যবহৃত হয়। CSS দিয়ে এই লিস্টের ডিফল্ট bullet বা নাম্বার, ইনডেন্টেশন প্রভৃতি পরিবর্তন করে কাস্টম ডিজাইন করা যায়। নিচের টেবিলে লিস্ট স্টাইলের জন্য কিছু গুরুত্বপূর্ণ CSS প্রপার্টি দেওয়া হলো:

📝 লিস্ট প্রপার্টি

📜 সম্পূর্ণ লিস্ট প্রপার্টি এক সাথে

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`list-style-type`	`disc` , `circle` , `square` , `decimal` , `lower-roman` , `none` , `"📌"`	বুলেট বা নম্বর টাইপ নির্ধারণ করে; কাস্টম স্ট্রিং (ইমোজি) সাপোর্ট।	ইমোজি বুলেট হলে ভিউপোর্ট-স্কেল মাথায় রাখুন।
`list-style-position`	`inside` , `outside`	বুলেট টেক্সটের আগে ভেতরে (`inside`) না বাইরের মার্জিনে (`outside`) দেখাবে।	`inside` দিলে মাল্টিলাইন প্যারাগ্রাফ ডেন্টেড হয়।

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`list-style-image`	`url(<img>)`, `none`	ইমেজকে বুলেট হিসেবে সেট করে; সঠিক স্কেল না হলে ব্লার হতে পারে।	উচ্চ-রেজ্যুলেশন ইমেজ ব্যবহার করুন।
`list-style`	shorthand: `type position image`	`list-style-type`, `list-style-position`, `list-style-image` একসঙ্গে।	রিসেট করতে: `list-style: none;`
`::marker`	Pseudo-element: properties: `content`, `color`, `font-size`	`<li>::marker` দিয়ে মার্কার কাস্টমাইজ করে—কন্টেন্ট, রঙ, সাইজ সব কন্ট্রোল।	ব্রাউজারে মার্কার ফন্ট সাইজ ইনহেরিট করে।
`marker`	CSS shorthand pseudo-element	`::marker` -এর বিকল্প সিনট্যাক্স; একই কাজ করে।	গ্লোবাল ফন্ট সাইজ ইনহেরিট।
`counter-reset`	`<name> <int>`	কাউন্টার ভেরিয়েবল রিসেট করে; মাল্টি-লেভেল নম্বরিং শুরু করতে।	Breadcrumbs বা অধ্যায় নম্বরিং এ কাজে লাগে।
`counter-increment`	`<name> <int>`	প্রতিটি আইটেমে কাউন্টার মান বাড়ায়; স্বয়ংক্রিয় লিস্ট নম্বরিং।	নেস্টেড লিস্টে নম্বর স্বয়ংক্রিয় রাখতে।
`content`	`"<string>"`, `counter(<name>)`	পসুডো-এলিমেন্টে স্ট্রিং বা কাউন্টার মান ইনজেক্ট করে।	কাস্টম ব্রেডক্রাম্ব সেপারেটর তৈরি করতে।
`text-decoration`	`underline`, `overline`, `line-through`, `none`	টেক্সট আন্ডারলাইন, ওভারলাইন, স্ট্রাইকথ্রু ইত্যাদি কন্ট্রোল করে।	`text-decoration-skip-ink: auto;` দিয়ে স্মুথ লুক।

📊 লিস্ট স্টাইলিং প্রপার্টি

প্রপার্টি	ভ্যালু	বর্ণনা
list-style-type	none, disc, circle, square, decimal, etc.	বুলেট বা নাম্বারিং টাইপ
list-style-image	url('image.png')	কাস্টম বুলেট ইমেজ
list-style-position	inside, outside	বুলেট পজিশন
list-style	shorthand property	টাইপ, ইমেজ, পজিশন কম্বাইন্ড

🎯 বেসিক লিস্ট স্টাইলিং

```css
/* আনঅর্ডারড লিস্ট */
ul {
    list-style-type: disc;      /* ডিফল্ট বুলেট */
    list-style-type: circle;    /* ফাঁকা সার্কেল */
    list-style-type: square;    /* বর্গাকার বুলেট */
    list-style-type: none;      /* বুলেট নেই */
}

/* অর্ডারড লিস্ট */
ol {
    list-style-type: decimal;      /* 1, 2, 3 */
    list-style-type: lower-alpha;  /* a, b, c */
    list-style-type: upper-alpha;  /* A, B, C */
    list-style-type: lower-roman;  /* i, ii, iii */
    list-style-type: upper-roman;  /* I, II, III */
}
```

🎨 কাস্টম লিস্ট স্টাইলিং

```css
/* কাস্টম ইমেজ বুলেট */
ul {
    list-style-image: url('bullet.png');
}

/* CSS দিয়ে কাস্টম বুলেট */
ul {
    list-style: none;
    padding-left: 0;
}

ul li {
    position: relative;
    padding-left: 20px;
}

ul li::before {
    content: "→";
    position: absolute;
    left: 0;
    color: #007bff;
}
```

🎯 ন্যাভিগেশন মেনু স্টাইলিং

নেভিগেশন মেনু স্টাইল করতে ব্যবহৃত প্রপার্টি:

Property Name	Property Value	বর্ণনা / ব্যাখ্যা
`text-decoration`	`none`, `underline`	লিঙ্কের স্টাইল নির্ধারণ
`display`	`inline`, `block`, `inline-block`	লিঙ্কের ডিসপ্লে টাইপ নির্ধারণ
`padding`	`px`, `em`, `%`	লিঙ্কের প্যাডিং নির্ধারণ
`margin`	`px`, `em`, `%`	লিঙ্কের মার্জিন নির্ধারণ
`background-color`	`<color>`	ব্যাকগ্রাউন্ডের রঙ নির্ধারণ
`color`	`<color>`	টেক্সটের রঙ নির্ধারণ

হরাইজন্টাল ন্যাভিগেশন মেনু

```
<nav class="main-nav">
    <ul>
        <li><a href="#">হোম</a></li>
        <li><a href="#">সার্ভিস</a></li>
        <li><a href="#">পণ্য</a></li>
        <li><a href="#">যোগাযোগ</a></li>
    </ul>
</nav>
```

```
.main-nav ul {
    list-style: none;
    padding: 0;
    margin: 0;
    display: flex;
    background: #333;
}

.main-nav li {
    margin: 0;
    padding: 0;
}

.main-nav a {
    display: block;
    padding: 15px 20px;
    color: white;
    text-decoration: none;
    transition: background 0.3s;
}

.main-nav a:hover {
    background: #555;
}
```

ভার্টিকাল ন্যাভিগেশন মেনু

```
.vertical-nav ul {
    list-style: none;
    padding: 0;
    margin: 0;
    width: 200px;
}

.vertical-nav li {
    border-bottom: 1px solid #ddd;
}

.vertical-nav a {
    display: block;
    padding: 10px 15px;
    color: #333;
    text-decoration: none;
    transition: background 0.3s;
}

.vertical-nav a:hover {
    background: #f4f4f4;
}
```

🔄 ড্রপডাউন মেনু

```
<nav class="dropdown-nav">
    <ul>
        <li>
            <a href="#">প্রোডাক্ট</a>
            <ul class="submenu">
                <li><a href="#">নতুন আইটেম</a></li>
                <li><a href="#">বেস্ট সেলার</a></li>
                <li><a href="#">ডিসকাউন্ট</a></li>
            </ul>
        </li>
    </ul>
</nav>
```

```
.dropdown-nav ul {
    list-style: none;
    padding: 0;
    margin: 0;
}

.dropdown-nav li {
    position: relative;
}
```

```css
.submenu {
    position: absolute;
    top: 100%;
    left: 0;
    display: none;
    background: white;
    box-shadow: 0 2px 5px rgba(0,0,0,0.2);
    min-width: 200px;
}

.dropdown-nav li:hover > .submenu {
    display: block;
}

.submenu a {
    padding: 10px 15px;
    display: block;
    color: #333;
    text-decoration: none;
}

.submenu a:hover {
    background: #f4f4f4;
}
```

📝 প্র্যাকটিস টাস্ক

1. একটি মেগা মেনু তৈরি করুন:

```html
<nav class="mega-menu">
    <ul>
        <li>
            <a href="#">ক্যাটাগরি</a>
            <div class="mega-content">
                <!-- মেগা মেনু কনটেন্ট -->
            </div>
        </li>
    </ul>
</nav>
```

2. একটি মাল্টি-লেভেল ড্রপডাউন মেনু তৈরি করুন:

```html
<nav class="multi-level">
    <ul>
        <li>
            <a href="#">লেভেল ১</a>
            <ul>
                <li>
```

```
                    <a href="#">লেভেল ২</a>
                    <ul>
                        <li><a href="#">লেভেল ৩</a></li>
                    </ul>
                </li>
            </ul>
        </li>
    </ul>
</nav>
```

📌 ইম্পর্টেন্ট পয়েন্ট

- লিস্ট ও নেভিগেশন মেনু স্টাইল করতে **মার্জিন** ও **প্যাডিং** ভালোভাবে বুঝতে হবে
- **ডিসপ্লে** প্রপার্টি গুরুত্বপূর্ণ ভূমিকা পালন করে নেভিগেশন ডিজাইনে

🤔 MCQ কুইজ

1. list-style-type: none; এর কাজ কি?
    - a) বুলেট হাইড করে
    - b) লিস্ট আইটেম হাইড করে
    - c) প্যাডিং রিমুভ করে
    - d) মার্জিন রিমুভ করে
2. কোন প্রপার্টি লিস্টের ইনডেন্টেশন কন্ট্রোল করে?
    - a) text-indent
    - b) margin-left
    - c) padding-left
    - d) list-indent
3. list-style-position: inside; এর প্রভাব কি?
    - a) বুলেট টেক্সট এর বাইরে থাকে
    - b) বুলেট টেক্সট এর ভিতরে থাকে
    - c) বুলেট হাইড হয়
    - d) বুলেট সেন্টার হয়
4. display: flex; লিস্টে ব্যবহার করলে:
    - a) বুলেট হাইড হয়
    - b) আইটেমগুলো হরাইজন্টাল হয়
    - c) আইটেমগুলো ভার্টিকাল হয়
    - d) কিছুই হয় না

5. ড্রপডাউন মেনুতে position: absolute; কেন ব্যবহার করা হয়?
    - a) স্পেস সেভ করতে
    - b) নরমাল ফ্লো থেকে বের করতে
    - c) অন্য এলিমেন্ট এর উপরে আনতে
    - d) স্ক্রল করাতে
6. hover স্টেটে ড্রপডাউন দেখাতে কোনটি ব্যবহার করা হয়?
    - a) visibility: visible
    - b) opacity: 1
    - c) display: block
    - d) উপরের সবগুলো
7. মোবাইল মেনুতে hamburger আইকন কিভাবে অ্যাড করা হয়?
    - a) img ট্যাগ দিয়ে
    - b) background-image দিয়ে
    - c) content প্রপার্টি দিয়ে
    - d) font-awesome আইকন দিয়ে
8. nested লিস্টে margin/padding দেয়ার সময় কোনটি মনে রাখা দরকার?
    - a) প্যারেন্ট এর margin
    - b) চাইল্ড এর padding
    - c) বুলেটের পজিশন
    - d) উপরের সবগুলো
9. মেগা মেনুতে width: 100%; দিলে:
    - a) প্যারেন্ট এর width নেয়
    - b) ভিউপোর্ট এর width নেয়
    - c) ফিক্সড width নেয়
    - d) কনটেন্ট অনুযায়ী width নেয়
10. রেসপন্সিভ মেনুতে flex-direction: column; এর প্রভাব:
    - a) আইটেম হরাইজন্টাল হয়
    - b) আইটেম ভার্টিকাল হয়
    - c) আইটেম হাইড হয়
    - d) কোন প্রভাব নেই

📚 হোমওয়ার্ক

1. একটি ফুল-ফিচার্ড ন্যাভিগেশন সিস্টেম তৈরি করুন:
    - মেইন ন্যাভিগেশন
    - মেগা মেনু
    - ড্রপডাউন সাবমেনু
    - মোবাইল রেসপন্সিভ
    - অ্যানিমেশন ইফেক্ট
2. একটি সাইডবার ন্যাভিগেশন তৈরি করুন:
    - অ্যাকর্ডিয়ন স্টাইল মেনু
    - মাল্টি-লেভেল সাবমেনু
    - অ্যাক্টিভ স্টেট
    - হোভার ইফেক্ট
3. একটি ফুটার ন্যাভিগেশন তৈরি করুন:
    - মাল্টিপল কলাম
    - সোশ্যাল আইকন
    - কপিরাইট টেক্সট
    - রেসপন্সিভ লেআউট
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৮: ফ্লেক্সিবল বক্স লেআউট (Flexbox)

CSS Flexbox হল একপ্রকার layout মডেল যা দিয়ে পেজের উপাদানগুলোকে খুব সহজে সারিবদ্ধ ও বন্টন করা যায়, বিশেষ করে যখন তাদের container এর ডাইমেনশনে বিভিন্ন পরিবর্তন আসে (যেমন responsive design)। Flexbox এক-মাত্রিক লেআউট (একদিকের: row বা column) ব্যবস্থাপনার জন্য আদর্শ।

🎯 ফ্লেক্সবক্স পরিচিতি

ফ্লেক্সবক্স হলো একটি মডার্ন লেআউট সিস্টেম যা একটি ডাইমেনশনে (row বা column) আইটেম অ্যারেঞ্জ করতে ব্যবহৃত হয়।

📜 ফ্লেক্স কন্টেইনার প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`display`	`flex`, `inline-flex`	এলিমেন্টকে ফ্লেক্স কন্টেইনার বানায়	ইনলাইন কন্টেন্টে হলে `inline-flex` ইউজ করুন

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`flex-direction`	`row`, `row-reverse`, `column`, `column-reverse`	মূল অক্ষের (main axis) দিক নির্ধারণ করে	মোবাইলে ভার্টিকাল লেআউটে `column` ভাল কাজ
`flex-wrap`	`nowrap`, `wrap`, `wrap-reverse`	ফ্লেক্স আইটেম ভাঙবে কি না সেট করে	কার্ড-গ্রিডে `wrap` দিলে র‍্যাপ হবে
`flex-flow`	shorthand: `direction wrap`	`flex-direction` + `flex-wrap` একসঙ্গে সেট করে	ডিফল্ট নয়; প্রায়ই `row wrap` ব্যবহার হয়
`justify-content`	`flex-start`, `center`, `flex-end`, `space-between`, `space-around`, `space-evenly`, `stretch`	মূল অক্ষ বরাবর আইটেম ছড়িয়ে দেয়	শেষ আইটেমে গ্যাপ চাইলে `margin-inline-start: auto;`
`align-items`	`stretch`, `flex-start`, `center`, `flex-end`, `baseline`	একলাইন ফ্লেক্সে ক্রস অক্ষে (cross axis) অ্যালাইনমেন্ট	ডিফল্ট: `stretch`
`align-content`	same values as `align-items`	মাল্টি-লাইন ফ্লেক্সে ক্রস অক্ষে স্পেস বিতরণ	এক লাইনে কাজ করে না
`gap`	`<length>`	আইটেমগুলির মধ্যে সমান ফাঁক তৈরি করে	রেসপন্সিভে `clamp()` দিয়ে ফ্লুইড গ্যাপ
`row-gap`	`<length>`	প্রতিটি ফ্লেক্স র‍্যাপ লাইন (row) এর মধ্যে ফাঁক	`gap` এর বিকল্প
`column-gap`	`<length>`	প্রতিটি কলামের মধ্যে ফাঁক	`gap` এর বিকল্প
`place-content`	shorthand: `align-content` + `justify-content`	দুই অক্ষে একসঙ্গে স্পেস বিতরণ	`place-content: center;`
`place-items`	shorthand: `align-items` + `justify-items` (grid only)	ইনসেল (cell) পজিশন ফাঁক সেট করে	ফ্লেক্সে শুধুমাত্র `align-items` কাজ করে

🧩 ফ্লেক্স আইটেম/সেলফ প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`order`	`<integer>`	আইটেমের প্রদর্শন ক্রম নির্ধারণ করে	নেতিবাচক মান আগে যায়
`flex-grow`	`<number>`	অতিরিক্ত ফাঁকা জায়গায় অংশগ্রহণের অনুপাত নির্ধারণ	গ্রো মানের মোট যোগফল স্বয়ংক্রিয় সামঞ্জস্য করে

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
		করে	
`flex-shrink`	`<number>`	সংকোচনের অনুপাত নির্ধারণ করে	`0` দিলে সংকোচন বন্ধ থাকে
`flex-basis`	`<length>`, `auto`, `content`	আইটেমের প্রাথমিক বেসলাইন সাইজ নির্ধারণ করে	`auto` রাখলে কনটেন্ট বা width অনুযায়ী সাইজ হয়
`flex`	shorthand: `grow shrink basis`	`flex-grow`, `flex-shrink`, `flex-basis` একসাথে সেট করে	`flex: 1` = `flex: 1 1 0%`
`align-self`	`stretch`, `flex-start`, `center`, `flex-end`, `baseline`	একক আইটেমের ক্রস-অক্ষে অ্যালাইনমেন্ট ওভাররাইড করে	প্যারেন্টের `align-items` কে ওভাররাইড করে

🎯 বেসিক ফ্লেক্সবক্স লেআউট

```
<div class="container">
    <div class="item">আইটেম ১</div>
    <div class="item">আইটেম ২</div>
    <div class="item">আইটেম ৩</div>
</div>
```

```
.container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    gap: 20px;
}

.item {
    flex: 1; /* সমান width */
    padding: 20px;
    background: #f4f4f4;
}
```

🔄 ফ্লেক্স ডাইরেকশন

```
/* হরাইজন্টাল (ডিফল্ট) */
.row {
    display: flex;
    flex-direction: row;
}
```

```css
/* ভার্টিকাল */
.column {
    display: flex;
    flex-direction: column;
}

/* রিভার্স অর্ডার */
.reverse {
    flex-direction: row-reverse;
    /* বা */
    flex-direction: column-reverse;
}
```

🧭 মূল-অক্ষ vs ক্রস-অক্ষে

- **Main Axis** = `flex-direction` অনুযায়ী ফ্লো (ডিফল্ট `row` → horizontal)।
- **Cross Axis** = perpendicular (ডিফল্ট `row` → vertical)।
- Wrapping ON হলে প্রত্যেক wrap-line-এও এই নিয়ম প্রযোজ্য থাকে।

Flex Shorthand Nuggets

```css
.card { flex: 1 0 250px; }
/* grow 1, shrink 0, basis 250px → responsive masonry */
```

🎯 জাস্টিফাই কনটেন্ট এবং অ্যালাইন আইটেমস

```css
/* সেন্টার অ্যালাইনমেন্ট */
.center {
    display: flex;
    justify-content: center;
    align-items: center;
}

/* স্পেস বিটুইন */
.space-between {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

/* স্পেস অ্যারাউন্ড */
.space-around {
    display: flex;
    justify-content: space-around;
    align-items: center;
}
```

🔄 ফ্লেক্স র‍্যাপিং

```
/* নো র‍্যাপ (ডিফল্ট) */
.nowrap {
    display: flex;
    flex-wrap: nowrap;
}

/* র‍্যাপ */
.wrap {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
}
```

📏 ফ্লেক্স গ্রো এবং শ্রিংক

```
/* সমান width */
.equal {
    flex: 1;
}

/* নির্দিষ্ট অনুপাতে */
.double {
    flex: 2;
}

/* নো গ্রো */
.fixed {
    flex: 0 0 200px;
}
```

🎯 অর্ডার

```
.first {
    order: -1; /* প্রথমে আসবে */
}

.last {
    order: 1; /* শেষে যাবে */
}
```

🪄 Common Patterns & Tricks

Pattern	CSS Snippet	কী কাজ করে
Centered Box	`display: flex; align-items: center; justify-content: center;`	হরাইজন্টাল ও ভার্টিক্যাল সেন্টারিং
Auto Margin Spacer	`.nav li:last-child { margin-left: auto; }`	শেষ আইটেমকে ডানে ঠেলে
Equal Height Cards	`container { display: flex; align-items: stretch; }` `.card { flex: 1; }`	সমান উচ্চতার কার্ড তৈরি
Holy Albatross	`flex: 1 1 clamp(200px, 33%, 400px);`	ফ্লুইড-রেসপনসিভ কার্ড গ্রিড
Flex-Gap Fallback	`display: inline-flex; margin-right: 1rem;`	সম্পূর্ণ ব্রাউজার ব্যাকআপ

📌 ইম্পর্টেন্ট পয়েন্ট

- Flexbox দিয়ে সহজেই রেসপন্সিভ লেআউট তৈরি করা যায়।
- justify-content এবং align-items প্রপার্টিগুলো ভালোভাবে বুঝতে হবে।

📝 প্র্যাকটিস টাস্ক

1. একটি হিরো সেকশন তৈরি করুন:

```
<section class="hero">
    <div class="hero-content">
        <h1>মূল শিরোনাম</h1>
        <p>বর্ণনামূলক টেক্সট...</p>
        <button>আরও জানুন</button>
    </div>
    <div class="hero-image">
        <img src="hero.jpg" alt="Hero Image">
    </div>
</section>
```

```
.hero {
    display: flex;
    align-items: center;
    gap: 40px;
    padding: 40px;
}

.hero-content {
    flex: 1;
}
```

```
.hero-image {
    flex: 1;
}
```

2. একটি কার্ড গ্রিড তৈরি করুন:

```
.card-grid {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
}

.card {
    flex: 0 0 calc(33.333% - 20px);
    /* 3 কলাম গ্রিড */
}
```

🤔 MCQ কুইজ

1. display: flex; কোথায় দিতে হয়?
    - a) চাইল্ড এলিমেন্টে
    - b) প্যারেন্ট এলিমেন্টে
    - c) উভয়েতে
    - d) যেকোন একটিতে
2. flex-direction এর ডিফল্ট ভ্যালু কি?
    - a) column
    - b) row
    - c) row-reverse
    - d) column-reverse
3. align-items কোন এক্সিসে কাজ করে?
    - a) মেইন এক্সিস
    - b) ক্রস এক্সিস
    - c) উভয় এক্সিস
    - d) কোনটিই না
4. flex: 1; এর পূর্ণ রূপ কি?
    - a) flex: 1 1 0
    - b) flex: 1 0 0

- c) flex: 1 1 auto
- d) flex: 1 0 auto

5. gap প্রপার্টি কি করে?

   - a) বর্ডার যোগ করে
   - b) মার্জিন যোগ করে
   - c) প্যাডিং যোগ করে
   - d) আইটেমের মধ্যে স্পেস যোগ করে

6. justify-content: space-between; এর ফলাফল:

   - a) সব আইটেম সেন্টারে
   - b) প্রথম ও শেষ আইটেম এজে, মাঝের গুলো সমান স্পেসে
   - c) সব আইটেমের মধ্যে সমান স্পেস
   - d) আইটেমগুলো ওভারল্যাপ করে

7. flex-wrap: wrap; কখন কাজ করে?

   - a) সব সময়
   - b) কন্টেইনার ছোট হলে
   - c) কনটেন্ট বেশি হলে
   - d) কখনোই না

8. align-self কি করে?

   - a) সব আইটেম অ্যালাইন করে
   - b) একটি আইটেম অ্যালাইন করে
   - c) কন্টেইনার অ্যালাইন করে
   - d) কিছুই করে না

9. flex-basis এর ডিফল্ট ভ্যালু কি?

   - a) 0
   - b) auto
   - c) 100%
   - d) inherit

10. কোন প্রপার্টি ফ্লেক্স আইটেমের অর্ডার পরিবর্তন করে?

    - a) flex-order
    - b) order
    - c) z-index
    - d) position

📚 হোমওয়ার্ক

1. একটি ফুল-পেজ লেআউট তৈরি করুন:
    - হেডার (sticky)
    - সাইডবার (ফিক্সড width)
    - মেইন কনটেন্ট (ফ্লেক্সিবল)
    - ফুটার (bottom এ ফিক্সড)
2. একটি প্রাইসিং টেবিল লেআউট তৈরি করুন:
    - তিনটি প্রাইস কার্ড
    - রেসপন্সিভ ডিজাইন
    - পপুলার প্ল্যান হাইলাইট
    - ফিচার লিস্ট
3. একটি গ্যালারি গ্রিড তৈরি করুন:
    - মাসোনরি স্টাইল লেআউট
    - ইমেজ রেশিও মেইনটেইন
    - রেসপন্সিভ কলাম
    - হোভার ইফেক্ট
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ৯: CSS গ্রিড লেআউট (CSS Grid Layout)

CSS Grid Layout দু-মাত্রিক লেআউট ব্যবস্থা প্রদান করে, যেখানে আপনি একই সাথে কলাম ও সারি উভয় দিক বিবেচনা করে পেজের ডিজাইন গঠন করতে পারেন। এটি ওয়েব ডিজাইনে অত্যন্ত শক্তিশালী এবং জটিল লেআউট সহজে করতে সাহায্য করে।

🎛️ গ্রিড লেআউট পরিচিতি

CSS Grid হলো দুই-মাত্রিক (two-dimensional) লেআউট সিস্টেম যা রো এবং কলাম উভয় দিকেই কনট্রোল করতে পারে।

গ্রিড কন্টেইনার প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`display`	`grid` , `inline-grid` , `subgrid`	এলিমেন্টকে গ্রিড কন্টেইনারে রুপান্তর করে	`subgrid` এখন Firefox-এ স্টোবেল

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`grid-template-columns` / `grid-template-rows`	`<track-size>`, `repeat()`, `minmax()`, `auto-fit`, `auto-fill`	কলাম বা সারি ট্র্যাক (প্রসারিত অঞ্চল) সংজ্ঞায়িত করে	`repeat(auto-fit, minmax(200px, 1fr))` স্মার্ট প্যাটার্ন
`grid-template-areas`	string (ASCII-style map)	নাম দিয়ে লেআউট ব্লক ম্যাপ তৈরি করে	ASCII-আর্টের মতো ভিজ্যুয়াল ম্যাপ
`grid-template`	shorthand: `rows / columns / areas`	সারি, কলাম ও এরিয়া একসঙ্গে ডিক্লেয়ার করে	`grid-template: "h h" 70px "s m" 1fr / 200px 1fr;`
`grid-auto-columns` / `grid-auto-rows`	`<length>`, `auto`, `minmax()`	অটো-জেনারেটেড ট্র্যাক পার্টস সেট করে	ম্যাসনরি-স্টাইল হ্যাক
`grid-auto-flow`	`row`, `column`, `dense`, `row dense`	নতুন আইটেম কোথায় পূরণ হবে তা নির্দেশ করে	`dense` ফাঁকা সেলগুলোতে ঢুকে লে-আউট ফিল-আপ করে
`gap` / `row-gap` / `column-gap`	`<length>`	গ্রিড ট্র্যাকগুলোর মাঝে স্পেস তৈরি করে	`%`-ব্যালু ও সাপোর্টেড
`justify-content` / `align-content`	`start`, `center`, `end`, `space-between`, `space-around`, `stretch`	পুরো গ্রিডে স্পেস ডিস্ট্রিবিউশন নিয়ন্ত্রণ করে	`space-between` অথবা `center` বেশ কাজে লাগে
`place-content`	shorthand: `align-content` + `justify-content`	একসাথে দুই অক্ষে স্পেস নিয়ন্ত্রণ	`place-content: center;` দ্রুত সেন্টার করার ট্রিক
`justify-items`	`start`, `center`, `end`, `stretch`	প্রতিটি গ্রিড সেলে অনুভূমিক অ্যালাইন নিয়ন্ত্রণ	কন্টেইনার-লেভেলে সেট করুন
`align-items`	`start`, `center`, `end`, `stretch`	প্রতিটি গ্রিড সেলে উল্লম্ব অ্যালাইন নিয়ন্ত্রণ	আইটেম ভেতরে `place-self` দিয়ে ওভাররাইড করুন
`place-items`	shorthand: `align-items` + `justify-items`	একসাথে দুই অক্ষে সেল-লেভেল অ্যালাইন নিয়ন্ত্রণ	`place-items: center;` দিয়ে ইনসেল সেন্টার

গ্রিড আইটেম প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`grid-column-start` / `grid-column-end`	`<line>`, `span <n>`	আইটেম কোন কলাম লাইনে শুরু ও শেষ হবে	লাইনের নাম ব্যবহার করুন, যেমন `header-start`
`grid-row-start` / `grid-row-end`	`<line>`, `span <n>`	আইটেম কোন সারি লাইনে শুরু ও শেষ হবে	লাইনের নাম ব্যবহার করুন, যেমন `main-end`
`grid-column` / `grid-row`	shorthand: `start / end`	কলাম বা সারির শুরু ও শেষ লাইন একসাথে সেট করে	`grid-column: 2 / span 3;`
`grid-area`	`<row-start> / <col-start> / <row-end> / <col-end>` বা named	নাম দিয়ে আইটেমের এরিয়া ভিত্তিক প্লেসমেন্ট নির্ধারণ করে	`grid-area: header;`
`place-self`	`start`, `center`, `end`, `stretch`	একক আইটেমের অনুভূমিক ও উল্লম্ব অ্যালাইনমেন্ট ওভাররাইড করে	`place-self: center;`

🎯 বেসিক গ্রিড লেআউট

```
<div class="grid-container">
    <div class="item">১</div>
    <div class="item">২</div>
    <div class="item">৩</div>
    <div class="item">৪</div>
    <div class="item">৫</div>
    <div class="item">৬</div>
</div>
```

```
.grid-container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    gap: 20px;
    padding: 20px;
}

.item {
    background: #f4f4f4;
    padding: 20px;
    text-align: center;
}
```

🔄 গ্রিড টেমপ্লেট এরিয়াস

```css
.layout {
    display: grid;
    grid-template-areas:
        "header header header"
        "sidebar main main"
        "footer footer footer";
    grid-template-columns: 200px 1fr 1fr;
    gap: 20px;
}

.header { grid-area: header; }
.sidebar { grid-area: sidebar; }
.main { grid-area: main; }
.footer { grid-area: footer; }
```

📏 ফ্র্যাকশনাল ইউনিট (fr)

```css
.grid {
    display: grid;
    /* সমান 3 কলাম */
    grid-template-columns: 1fr 1fr 1fr;

    /* অসম কলাম */
    grid-template-columns: 2fr 1fr 1fr;

    /* মিক্সড ইউনিট */
    grid-template-columns: 200px 1fr 2fr;
}
```

🎯 গ্রিড ফাংশন

```css
/* repeat() ফাংশন */
.grid {
    grid-template-columns: repeat(3, 1fr);
    /* 1fr 1fr 1fr এর সমান */
}

/* minmax() ফাংশন */
.responsive-grid {
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    /* রেসপন্সিভ কলাম */
}

/* auto-fit vs auto-fill */
.auto-fit {
    grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));
    /* স্পেস ভরাট করে */
}
```

```
.auto-fill {
    grid-template-columns: repeat(auto-fill, minmax(100px, 1fr));
    /* এম্পটি কলাম রাখে */
}
```

🎨 গ্রিড অ্যালাইনমেন্ট

```
.grid-container {
    /* আইটেম অ্যালাইনমেন্ট */
    justify-items: center;    /* হরাইজন্টাল */
    align-items: center;      /* ভার্টিকাল */

    /* কনটেন্ট অ্যালাইনমেন্ট */
    justify-content: center;  /* হরাইজন্টাল */
    align-content: center;    /* ভার্টিকাল */
}

.grid-item {
    /* ইনডিভিজুয়াল আইটেম */
    justify-self: center;
    align-self: center;
}
```

🌠 রেস্পনসিভ কার্ড-গ্যালারি - Auto-Fit প্যাটার্ন

```
.gallery {
  display:grid;
  grid-template-columns: repeat(auto-fit, minmax(220px, 1fr));
  gap: 2rem;
}
```

- `auto-fit` খোলা কলাম গুটিয়ে কার্ড-লজিক স্মার্ট রাখে।

🧱 মেসনরি-স্টাইল গ্রিড (CSS Hack)

```
.masonry {
  column-width: 300px; /* fallback */
  columns: 3;
}
.card { break-inside: avoid; margin-bottom:1.5rem; }
/* Upcoming spec → 'grid-template-rows: masonry;' */
```

📌 ইম্পর্টেন্ট পয়েন্ট

- Grid Layout একাধিক কলাম এবং সারিতে এলিমেন্ট সাজাতে অত্যন্ত উপযোগী।

- Flexbox এক লাইনে আইটেম সাজানোর জন্য ভালো, Grid মাল্টি-ডাইমেনশনাল লেআউটের জন্য।

📝 প্র্যাকটিস টাস্ক

1. একটি ফুল পেজ লেআউট তৈরি করুন:

```
<div class="page-layout">
    <header>হেডার</header>
    <nav>ন্যাভিগেশন</nav>
    <main>মেইন কনটেন্ট</main>
    <aside>সাইডবার</aside>
    <footer>ফুটার</footer>
</div>
```

```
.page-layout {
    display: grid;
    grid-template-areas:
        "header header"
        "nav main"
        "nav sidebar"
        "footer footer";
    grid-template-columns: 200px 1fr;
    grid-template-rows: auto auto 1fr auto;
    min-height: 100vh;
}
```

2. একটি রেসপন্সিভ গ্যালারি তৈরি করুন:

```
.gallery {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 20px;
    padding: 20px;
}

.gallery-item img {
    width: 100%;
    height: 100%;
    object-fit: cover;
}
```

🤔 MCQ কুইজ

1. display: grid; কোথায় দিতে হয়?
    - a) চাইল্ড এলিমেন্টে

- b) প্যারেন্ট এলিমেন্টে
- c) উভয়েতে
- d) যেকোন একটিতে

2. 1fr এর অর্থ কি?

- a) 1 পিক্সেল
- b) অবশিষ্ট স্পেসের 1 ভাগ
- c) 1 পার্সেন্ট
- d) 1 rem

3. grid-template-areas এ একই নাম একাধিকবার দিলে:

- a) এরর দেখায়
- b) একটি এরিয়া বড় হয়
- c) লাস্ট একটি কাজ করে
- d) কোনটাই কাজ করে না

4. minmax() ফাংশনের কাজ কি?

- a) শুধু মিনিমাম সাইজ সেট করে
- b) শুধু ম্যাক্সিমাম সাইজ সেট করে
- c) মিনিমাম-ম্যাক্সিমাম রেঞ্জ সেট করে
- d) কোনটাই না

5. auto-fit এবং auto-fill এর পার্থক্য:

- a) কোন পার্থক্য নেই
- b) auto-fit খালি স্পেস ভরে, auto-fill করে না
- c) auto-fill খালি স্পেস ভরে, auto-fit করে না
- d) উভয়ই খালি স্পেস ভরে

6. grid-gap এর শর্টহ্যান্ড:

- a) gap
- b) spacing
- c) grid-spacing
- d) space

7. implicit grid কি?

- a) manually তৈরি গ্রিড
- b) auto-generated গ্রিড
- c) hidden গ্রিড

- d) visible গ্রিড

8. grid-auto-flow এর ডিফল্ট ভ্যালু:
   - a) row
   - b) column
   - c) dense
   - d) sparse
9. grid-template-columns: repeat(3, 1fr); এর ফলাফল:
   - a) 1 কলাম
   - b) 2 কলাম
   - c) 3 কলাম
   - d) 4 কলাম
10. justify-items এবং justify-content এর পার্থক্য:
    - a) কোন পার্থক্য নেই
    - b) justify-items গ্রিড আইটেম, justify-content পুরো গ্রিড
    - c) justify-content গ্রিড আইটেম, justify-items পুরো গ্রিড
    - d) উভয়ই একই কাজ করে

📚 হোমওয়ার্ক

1. একটি নিউজ ওয়েবসাইট লেআউট তৈরি করুন:
   - হেডলাইন সেকশন (বড় থাম্বনেইল)
   - নিউজ গ্রিড (ছোট থাম্বনেইল)
   - সাইডবার (পপুলার পোস্ট)
   - রেসপন্সিভ ডিজাইন
2. একটি ড্যাশবোর্ড লেআউট তৈরি করুন:
   - স্ট্যাটিসটিক্স কার্ড
   - চার্ট/গ্রাফ সেকশন
   - টেবিল/লিস্ট ভিউ
   - উইজেট গ্রিড
3. একটি পোর্টফোলিও গ্রিড তৈরি করুন:
   - ভিন্ন ভিন্ন সাইজের আইটেম
   - মাসোনরি লেআউট
   - ফিল্টার সিস্টেম

- হোভার ইফেক্ট

4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ১০: রেসপন্সিভ ডিজাইন এবং মিডিয়া কোয়েরি (Responsive Design and Media Queries)

Responsive ডিজাইন বলতে এমন ওয়েব পেজকে বোঝায় যা বিভিন্ন স্ক্রিনের আকার ও ডিভাইসে মানানসইভাবে নিজেকে পরিবর্তন করে। বড় ডেস্কটপ মনিটর থেকে ছোট মোবাইল ফোন - সবখানেই ওয়েবসাইটটি ব্যবহারযোগ্য ও সুন্দর থাকা চাই। CSS-এর বিভিন্ন বৈশিষ্ট্য ও প্রযুক্তি ব্যবহার করে responsive ডিজাইন করা যায়, যেমন আপেক্ষিক ইউনিট (%, em, rem, vw/vh), এবং মিডিয়া কোয়েরি ব্যবহার করে ভিন্ন স্ক্রিন সাইজে আলাদা CSS প্রয়োগ। এই অধ্যায় আমরা এর সব কিছু দেখবো

📱 Responsive Design কি?

রেসপন্সিভ ডিজাইন হলো এমন একটি পদ্ধতি যা ওয়েবসাইটকে সব ডিভাইসে (মোবাইল, ট্যাবলেট, ডেস্কটপ) সুন্দরভাবে দেখায়।

🎯 Media Queries

Media Queries CSS এর একটি ফিচার যা স্ক্রিনের প্রস্থ, উচ্চতা, রেজোলিউশন, ডিভাইস টাইপ অনুযায়ী স্টাইল চেঞ্জ করতে দেয়।

সম্পূর্ণ প্রপার্টি / অ্যাট্রিবিউট টেবিল এক সাথে রেসপনসিভ এর জন্য 🔪

Directive / Unit	Core Values / Syntax	বাংলা ব্যাখ্যা	Quick Tips
`@media`	`@media (feature) and/or (feature) { … }`	মিডিয়া-কোয়েরি ব্লক শুরু	`@media screen and (max-width: 600px)`
Media Types	`all`, `screen`, `print`, `speech`	কোন আউটপুট ডিভাইস	টাইপ বাদ দিলে `all` ধরা হয়
Width / Height	`(min-width: 768px)`, `(max-height: 800px)`	ভিউপোর্ট মাত্রা	শুধু ইন্টিজার + ইউনিট
Orientation	`(orientation: portrait)`, `(orientation: landscape)`	ডিভাইস ঘোরানোর ধরন	মোবাইল-টার্ন ডিটেক্ট
Aspect-ratio	`(min-aspect-ratio: 3/2)`	প্রস্থ:উচ্চতা অনুপাত	`/` সিম্বল দিয়ে

Directive / Unit	Core Values / Syntax	বাংলা ব্যাখ্যা	Quick Tips
Resolution	`(min-resolution: 300dpi)`	পিপিআই বা ডিপিআই	`dppx`, `dpcm` বিকল্প
Color Scheme	`(prefers-color-scheme: dark)`	ইউজার ডিভাইস থিম পছন্দ	সঙ্গে `color-scheme` হেডার
Reduced Motion	`(prefers-reduced-motion: reduce)`	অ্যানিমেশন হ্রাস অনুরোধ	`@media (prefers-reduced-motion) { animation: none; }`
Hover / Pointer	`(hover: hover)`, `(pointer: coarse)`	ইনপুট-ডিভাইস ক্ষমতা	মোবাইল vs মাউস লজিক
Container Query	`@container (min-width: 40ch) { … }`	পারেন্ট কন্টেইনার ব্রেক-পয়েন্ট	Edge & Chrome-এ স্টেবল
Viewport Units	`vw`, `vh`, `vmin`, `vmax`, `svh`, `lvh`, `dvh`	স্ক্রিন ভাগ ইউনিট	মোবাইল URL বার ঝিলিচ ফিক্স করতে `svh` ব্যবহার করুন
Fluid Function	`clamp(min, ideal, max)`	মিন → আইডিয়াল → ম্যাক্স স্কেল	`font-size: clamp(1rem, 2vw + 1rem, 2rem);`
Meta Viewport	`<meta name="viewport" content="width=device-width,initial-scale=1">`	মোবাইল রেস্পনসিভ চালু	প্রথম লাইনে `<head>`-এ যোগ করুন
Responsive Img	`srcset`, `sizes`	ডিভাইস অনুযায়ী ইমেজ লোড	`<img src="small.jpg" srcset="large.jpg 2x">`

📊 ভিউপোর্ট মেটা ট্যাগ

```
<head>

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

</head>
```

📱 মিডিয়া কোয়েরি সিনট্যাক্স

```css
@media screen and (max-width: 768px) {
    /* মোবাইল স্টাইল */
}

@media screen and (min-width: 768px) and (max-width: 1024px) {
    /* ট্যাবলেট স্টাইল */
}

@media screen and (min-width: 1024px) {
    /* ডেস্কটপ স্টাইল */
}
```

📊 ব্রেকপয়েন্ট

ডিভাইস	ব্রেকপয়েন্ট	ব্যবহার
মোবাইল	< 576px	ছোট স্ক্রিন
ল্যান্ডস্কেপ মোবাইল	>= 576px	মোবাইল হরাইজন্টাল
ট্যাবলেট	>= 768px	ট্যাবলেট/আইপ্যাড
ডেস্কটপ	>= 992px	ল্যাপটপ
লার্জ ডেস্কটপ	>= 1200px	বড় মনিটর
এক্সট্রা লার্জ	>= 1400px	অতি বড় স্ক্রিন

🎯 রেসপন্সিভ ইউনিট

```css
/* রিলেটিভ ইউনিট */
.container {
    width: 90%;           /* প্যারেন্ট এর % */
    font-size: 1.5rem;    /* রুট font-size এর গুণিতক */
    padding: 2em;         /* নিজের font-size এর গুণিতক */
    margin: 2vw;          /* ভিউপোর্ট width এর % */
    height: 50vh;         /* ভিউপোর্ট height এর % */
}

/* ক্যালকুলেশন */
.sidebar {
    width: calc(100% - 200px);
}
```

📱 মোবাইল-ফার্স্ট ডিজাইন

```css
/* বেস স্টাইল (মোবাইল) */
.container {
```

```
    width: 100%;
    padding: 10px;
}

/* ট্যাবলেট */
@media (min-width: 768px) {
    .container {
        width: 90%;
        padding: 20px;
    }
}

/* ডেস্কটপ */
@media (min-width: 1024px) {
    .container {
        width: 80%;
        max-width: 1200px;
        padding: 30px;
    }
}
```

🔄 ফ্লেক্সিবল ইমেজ

```
/* রেসপন্সিভ ইমেজ */
img {
    max-width: 100%;
    height: auto;
}

/* ব্যাকগ্রাউন্ড ইমেজ */
.hero {
    background-image: url('large.jpg');
    background-size: cover;
    background-position: center;
}

@media (max-width: 768px) {
    .hero {
        background-image: url('small.jpg');
    }
}
```

📱 রেসপন্সিভ ন্যাভিগেশন

```
<nav class="nav">
    <button class="nav-toggle">মেনু</button>
    <ul class="nav-menu">
        <li><a href="#">হোম</a></li>
        <li><a href="#">সার্ভিস</a></li>
```

```
        <li><a href="#">পণ্য</a></li>
        <li><a href="#">যোগাযোগ</a></li>
    </ul>
</nav>
```

```
/* ডেস্কটপ মেনু */
.nav-menu {
    display: flex;
    gap: 20px;
}

/* মোবাইল মেনু */
@media (max-width: 768px) {
    .nav-toggle {
        display: block;
    }

    .nav-menu {
        display: none;
        flex-direction: column;
    }

    .nav-menu.active {
        display: flex;
    }
}
```

🎯 রেসপন্সিভ গ্রিড সিস্টেম

```
.grid {
    display: grid;
    gap: 20px;
    padding: 20px;
}

/* মোবাইল: 1 কলাম */
@media (min-width: 576px) {
    .grid {
        grid-template-columns: repeat(2, 1fr);
    }
}

/* ট্যাবলেট: 2 কলাম */
@media (min-width: 768px) {
    .grid {
        grid-template-columns: repeat(3, 1fr);
    }
}
```

```
/* ডেস্কটপ: 4 কলাম */
@media (min-width: 992px) {
    .grid {
        grid-template-columns: repeat(4, 1fr);
    }
}
```

✨ Fluid-Type & Space clamp() ভোজবাজি

```
:root {
  --step-0: clamp(1rem, 0.5vw + 0.9rem, 1.2rem);
  --step-1: clamp(1.25rem, 0.8vw + 1rem, 1.6rem);
}
h1 { font-size: var(--step-1); }
```

📌 ইম্পর্টেন্ট পয়েন্ট

- সর্বদা **মোবাইল-ফার্স্ট অ্যাপ্রোচ** ব্যবহার করুন
- **breakpoints** ভালোভাবে পরিকল্পনা করে নিন: যেমন `480px`, `768px`, `1024px`, `1280px`

📝 প্র্যাকটিস টাস্ক

1. রেসপন্সিভ হিরো সেকশন তৈরি করুন:

```
<section class="hero">
    <div class="hero-content">
        <h1>স্বাগতম</h1>
        <p>আমাদের ওয়েবসাইটে আপনাকে স্বাগতম</p>
        <button>আরও জানুন</button>
    </div>
</section>
```

```
.hero {
    min-height: 80vh;
    padding: 40px;
    text-align: center;
}

@media (min-width: 768px) {
    .hero {
        text-align: left;
        padding: 60px;
    }
}
```

2. রেসপন্সিভ কার্ড গ্রিড:

```
.card-grid {
    display: grid;
    gap: 20px;
    padding: 20px;
}

.card {
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 2px 4px rgba(0,0,0,0.1);
}

/* রেসপন্সিভ গ্রিড */
@media (min-width: 576px) {
    .card-grid {
        grid-template-columns: repeat(2, 1fr);
    }
}

@media (min-width: 992px) {
    .card-grid {
        grid-template-columns: repeat(3, 1fr);
    }
}
```

🤔 MCQ কুইজ

1. ভিউপোর্ট মেটা ট্যাগ কেন ব্যবহার করা হয়?
    - a) ওয়েবসাইট স্পিড বাড়াতে
    - b) মোবাইলে স্কেলিং কন্ট্রোল করতে
    - c) SEO ইমপ্রুভ করতে
    - d) ফন্ট সাইজ ফিক্স করতে
2. min-width মিডিয়া কোয়েরি কখন কাজ করে?
    - a) স্ক্রিন সাইজ কম হলে
    - b) স্ক্রিন সাইজ বেশি হলে
    - c) স্ক্রিন সাইজ সমান হলে
    - d) স্ক্রিন অরিয়েন্টেশন চেঞ্জ হলে
3. মোবাইল-ফার্স্ট এপ্রোচে কোন মিডিয়া কোয়েরি ব্যবহার করা হয়?
    - a) max-width
    - b) min-width
    - c) max-height

- d) min-height

4. rem ইউনিট কিসের উপর ভিত্তি করে কাজ করে?
   - a) প্যারেন্ট এলিমেন্ট
   - b) রুট এলিমেন্ট
   - c) ভিউপোর্ট
   - d) কারেন্ট এলিমেন্ট
5. ভিউপোর্ট ইউনিট vw এর অর্থ কি?
   - a) ভিউপোর্ট হাইট
   - b) ভিউপোর্ট ওয়িডথ
   - c) ভার্টিকাল ওয়িডথ
   - d) ভিজিবল ওয়িডথ
6. রেসপন্সিভ ইমেজের জন্য নিচের কোনটি সঠিক?
   - a) width: 100%
   - b) max-width: 100%
   - c) min-width: 100%
   - d) fixed-width: 100%
7. calc() ফাংশন কি করে?
   - a) CSS এ ম্যাথ করে
   - b) ইমেজ রিসাইজ করে
   - c) ফন্ট সাইজ ক্যালকুলেট করে
   - d) কালার ক্যালকুলেট করে
8. কোন ব্রেকপয়েন্ট ট্যাবলেটের জন্য ব্যবহৃত হয়?
   - a) 576px
   - b) 768px
   - c) 992px
   - d) 1200px
9. orientation মিডিয়া কোয়েরি কি চেক করে?
   - a) স্ক্রিন সাইজ
   - b) ডিভাইস টাইপ
   - c) পোর্ট্রেট/ল্যান্ডস্কেপ
   - d) রেজোলিউশন
10. flexbox এ gap প্রপার্টি সাপোর্ট না করলে কি ব্যবহার করা যায়?

- a) padding
- b) margin
- c) spacing
- d) distance

📚 হোমওয়ার্ক

1. একটি রেসপন্সিভ লেআউট তৈরি করুন:
   - মোবাইল মেনু (হ্যামবার্গার)
   - হিরো সেকশন
   - ফিচার গ্রিড
   - টেস্টিমোনিয়াল স্লাইডার
   - কন্টাক্ট ফর্ম
2. একটি ব্লগ লেআউট তৈরি করুন:
   - রেসপন্সিভ হেডার
   - ব্লগ গ্রিড
   - সাইডবার (মোবাইলে নিচে)
   - পেজিনেশন
3. একটি ই-কমার্স প্রোডাক্ট পেজ:
   - প্রোডাক্ট গ্রিড
   - ফিল্টার সাইডবার
   - শপিং কার্ট
   - চেকআউট প্রসেস
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ১১: CSS ট্রানজিশন এবং অ্যানিমেশন (CSS Transitions and Animations)

CSS এর অন্যতম আকর্ষণীয় ক্ষমতা হল অ্যানিমেশন – যা জাভাস্ক্রিপ্ট ছাড়াই অনেক ইন্টার‍্যাকটিভ effect দিতে পারে। এর দুটি দিক: Transitions (ধাপে ধাপে property মানের পরিবর্তন) এবং Animations (keyframes সংজ্ঞা করে জটিল ধারাবাহিক animation)। এই অধ্যায়ে আমরা দুটো জিনিস ই দেখবো

🎯 ট্রানজিশন পরিচিতি

CSS ট্রানজিশন দিয়ে এলিমেন্টের প্রপার্টি ভ্যালু পরিবর্তনকে স্মুথ করা হয়।

📊 ট্রানজিশন প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`transition-property`	`<property>` , `all` , `none`	কোন CSS প্রপার্টির পরিবর্তনে ট্রানজিশন চালু হবে	GPU-হ্যাক: শুধুমাত্র `transform` বা `opacity` ট্রানজিশন করুন
`transition-duration`	`<time>`	ট্রানজিশনের মোট সময়কাল	`0.2s` – `0.4s` হল ইউজেবল সুইট-স্পট
`transition-timing-function`	`ease` , `linear` , `ease-in` , `ease-out` , `cubic-bezier()` , `steps()`	গতির বক্ররেখা (ইজিং) নিয়ন্ত্রণ করে	Material প্যাটার্ন: `cubic-bezier(.4,0,.2,1)`
`transition-delay`	`<time>`	ট্রানজিশন শুরুতে বিলম্ব	স্ট্যাগার এফেক্টে কাজে লাগে
`transition`	shorthand: `property duration timing-function delay`	সব ট্রানজিশন প্রপার্টি এক লাইনে সংক্ষিপ্ত করে	উদাহরণ: `transition: transform .3s ease;`

🎯 বেসিক ট্রানজিশন

```
/* সিঙ্গেল প্রপার্টি */
.button {
    background: blue;
    transition: background 0.3s ease;
}

.button:hover {
    background: darkblue;
}

/* মাল্টিপল প্রপার্টি */
.card {
    transform: scale(1);
    opacity: 1;
    transition: transform 0.3s ease, opacity 0.3s ease;
}

.card:hover {
    transform: scale(1.1);
    opacity: 0.8;
}
```

🎨 অ্যানিমেশন পরিচিতি

অ্যানিমেশন দিয়ে একটি এলিমেন্টের স্টাইল একাধিক স্টেপে পরিবর্তন করা যায়।

🎬 অ্যানিমেশন প্রপার্টি

🎬 CSS Animations প্রপার্টি

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`@keyframes`	`%`, `from`, `to` { … }	ফ্রেম-বেসড স্টাইল ডিফাইন করে	০% ও ১০০% একই করলে পরোক্ষভাবে লুপ তৈরি হয়
`animation-name`	`<keyframes-name>`	কোন `@keyframes` চালু হবে তা নির্ধারণ করে	কমা-সেপারেটেড একাধিক নামও সাপোর্ট করে
`animation-duration`	`<time>`	পুরো অ্যানিমেশনের দৈর্ঘ্য	সেকেন্ড (`s`) বা মিলিসেকেন্ড (`ms`) উভয়ই চলে
`animation-timing-function`	`ease`, `linear`, `ease-in`, `ease-out`, `cubic-bezier()`, `steps()`	প্রতি সাইকেলে ইজিং বক্ররেখা নিয়ন্ত্রণ করে	ইনলাইন বা `@keyframes` ওভাররাইড উভয়েই কাজ করে
`animation-iteration-count`	`<number>`, `infinite`	কয়বার অ্যানিমেশন চলবে	hover–loop এ সাবধান; অতিরিক্ত পুনরায় চক্র ভারী হতে পারে
`animation-direction`	`normal`, `reverse`, `alternate`, `alternate-reverse`	অ্যানিমেশনের চালানোর দিক	`alternate` দিলে ফরওয়ার্ড ও রিভার্স পাল্টাপাল্টি চলবে
`animation-delay`	`<time>`	شروع সময় বিলম্ব	নেগেটিভ হলে মাঝপথ থেকে শুরু হয়
`animation-fill-mode`	`none`, `forwards`, `backwards`, `both`	অ্যানিমেশনের আগে/পরে স্টেট ধরে রাখে	শেষের স্টেট রাখতে `forwards` অথবা `both` ব্যবহার করুন
`animation-play-state`	`running`, `paused`	ঘড়ি চালু/থামিয়ে অ্যানিমেশন নিয়ন্ত্রণ করে	JS দিয়ে `.paused` টগ্‌ল করতে পারে
`animation`	shorthand: `name duration timing-function delay iteration-count direction fill-mode play-state`	সব অ্যানিমেশন প্রপার্টি এক লাইনে সেট করে	উদাহরণ: `animation: bounce 1s ease-out 0.2s infinite both paused;`

Property	Core Values / Units	বাংলা ব্যাখ্যা	Quick Tips
`offset-path` / `offset-distance` / `offset-rotate`	`<path>`, `<length>`, `auto`	SVG বা CSS Path ধরে মুভমেন্ট এনেবল করে	Chrome-এ স্থিতিশীল; GPU-অপ্টিমাইজড
`@scroll-timeline` / `animation-timeline`	`scroll-timeline-name`, `animation-timeline`	স্ক্রল পজিশন ভিত্তিতে অ্যানিমেশনের টাইমলাইন তৈরি করে	Experimental; ব্রাউজার ফ্ল্যাগ দিয়ে পরীক্ষা করুন

🎯 কীফ্রেম অ্যানিমেশন

```
/* বেসিক অ্যানিমেশন */
@keyframes fadeIn {
    from {
        opacity: 0;
    }
    to {
        opacity: 1;
    }
}

/* মাল্টিপল স্টেপ */
@keyframes rainbow {
    0% { background: red; }
    25% { background: yellow; }
    50% { background: blue; }
    75% { background: green; }
    100% { background: red; }
}

/* অ্যানিমেশন অ্যাপ্লাই */
.element {
    animation: fadeIn 1s ease-in;
}

.colorful {
    animation: rainbow 5s infinite;
}
```

🎨 ট্রান্সফর্ম অ্যানিমেশন

```
/* 2D ট্রান্সফর্ম */
.element {
    transition: transform 0.3s ease;
}
```

```css
.element:hover {
    transform: scale(1.2) rotate(45deg);
}

/* 3D ট্রান্সফর্ম */
.card {
    transition: transform 0.5s ease;
    transform-style: preserve-3d;
}

.card:hover {
    transform: rotateY(180deg);
}
```

🎯 অ্যানিমেশন টাইমিং ফাংশন

```css
/* বিভিন্ন টাইমিং */
.ease { transition-timing-function: ease; }
.linear { transition-timing-function: linear; }
.ease-in { transition-timing-function: ease-in; }
.ease-out { transition-timing-function: ease-out; }
.ease-in-out { transition-timing-function: ease-in-out; }

/* কাস্টম টাইমিং */
.custom {
    transition-timing-function: cubic-bezier(0.68, -0.55, 0.265, 1.55);
}
```

🔄 লোডিং অ্যানিমেশন

```css
/* স্পিনার */
@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

.spinner {
    width: 40px;
    height: 40px;
    border: 4px solid #f3f3f3;
    border-top: 4px solid #3498db;
    border-radius: 50%;
    animation: spin 1s linear infinite;
}

/* পালস */
@keyframes pulse {
    0% { transform: scale(1); }
```

```
    50% { transform: scale(1.1); }
    100% { transform: scale(1); }
}

.pulse {
    animation: pulse 2s ease infinite;
}
```

📌 ইম্পর্টেন্ট পয়েন্ট

- Hover effect তৈরি করতে **transition** বেশ সহজ ও কার্যকর
- **Animation** দিয়ে ব্যানার, বাটন, লোডার তৈরি করা যায়

📝 প্র্যাকটিস টাস্ক

1. একটি হোভার এফেক্ট কার্ড তৈরি করুন:

```
<div class="card">
    <img src="image.jpg" alt="Card Image">
    <div class="card-content">
        <h3>কার্ড টাইটেল</h3>
        <p>কার্ড ডেসক্রিপশন</p>
    </div>
</div>
```

```
.card {
    position: relative;
    overflow: hidden;
    transition: transform 0.3s ease;
}

.card-content {
    position: absolute;
    bottom: -100%;
    width: 100%;
    background: rgba(0,0,0,0.8);
    color: white;
    padding: 20px;
    transition: bottom 0.3s ease;
}

.card:hover {
    transform: scale(1.05);
}

.card:hover .card-content {
    bottom: 0;
}
```

2. একটি অ্যানিমেটেড মেনু আইকন:

```
.menu-icon {
    width: 30px;
    height: 3px;
    background: black;
    position: relative;
    transition: all 0.3s ease;
}

.menu-icon::before,
.menu-icon::after {
    content: '';
    position: absolute;
    width: 100%;
    height: 100%;
    background: black;
    transition: all 0.3s ease;
}

.menu-icon::before { top: -8px; }
.menu-icon::after { top: 8px; }

.menu-icon.active {
    background: transparent;
}

.menu-icon.active::before {
    transform: rotate(45deg) translate(5px, 5px);
}

.menu-icon.active::after {
    transform: rotate(-45deg) translate(7px, -8px);
}
```

🤔 MCQ কুইজ

1. transition-duration এ কোন ইউনিট ব্যবহার করা যায়?
    - a) px
    - b) em
    - c) seconds
    - d) %
2. @keyframes এ from-to এর বিকল্প কি?
    - a) start-end
    - b) 0%-100%

- c) begin-finish
- d) first-last

3. animation-iteration-count: infinite; এর কাজ কি?

- a) একবার চলে
- b) দুইবার চলে
- c) অসীম বার চলে
- d) চলে না

4. transform-origin এর ডিফল্ট ভ্যালু কি?

- a) top left
- b) center center
- c) bottom right
- d) top center

5. কোন প্রপার্টি অ্যানিমেট করা যায় না?

- a) width
- b) color
- c) display
- d) opacity

6. transition-timing-function: ease-in এর অর্থ:

- a) ধীরে শুরু, দ্রুত শেষ
- b) দ্রুত শুরু, ধীরে শেষ
- c) সমান গতি
- d) ধীরে শুরু ও শেষ

7. transform: scale(2) rotate(45deg) এর ফলাফল:

- a) প্রথমে scale, পরে rotate
- b) প্রথমে rotate, পরে scale
- c) একসাথে দুটোই
- d) কোনটাই না

8. animation-fill-mode: forwards এর কাজ:

- a) শুরুর স্টেটে থাকে
- b) শেষের স্টেটে থাকে
- c) মাঝের স্টেটে থাকে
- d) নরমাল স্টেটে থাকে

9. কোনটি পারফরম্যান্সের জন্য ভালো?
    - a) margin অ্যানিমেশন
    - b) transform অ্যানিমেশন
    - c) width অ্যানিমেশন
    - d) height অ্যানিমেশন
10. transition-delay এর ডিফল্ট ভ্যালু কত?
    - a) 1s
    - b) 0.5s
    - c) 0s
    - d) 2s

📚 হোমওয়ার্ক

1. একটি অ্যানিমেটেড ল্যান্ডিং পেজ তৈরি করুন:
    - হেডিং টেক্সট অ্যানিমেশন
    - ইমেজ ফেড-ইন
    - বাটন হোভার ইফেক্ট
    - স্ক্রল অ্যানিমেশন
2. একটি অ্যানিমেটেড প্রাইসিং টেবিল:
    - হোভার ইফেক্ট
    - পপুলার প্ল্যান হাইলাইট
    - ফিচার লিস্ট অ্যানিমেশন
    - কল-টু-অ্যাকশন বাটন
3. একটি অ্যানিমেটেড পোর্টফোলিও:
    - প্রজেক্ট কার্ড হোভার
    - স্কিল বার অ্যানিমেশন
    - ইমেজ গ্যালারি ট্রানজিশন
    - কন্টাক্ট ফর্ম ইফেক্ট
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ১২: অ্যাডভান্সড CSS: ভ্যারিয়েবল, ফাংশন, এবং মডার্ন ফিচার

CSS ক্রমাগত উন্নত হচ্ছে, নতুন ফিচার যোগ হচ্ছে যা ডেভেলপারদের জীবন সহজ করছে। এই অধ্যায়ে আমরা কিছু modern CSS বৈশিষ্ট্য নিয়ে আলোচনা করব, যেমন CSS Variables (কাস্টম properties), কিছু দরকারী CSS ফাংশন

(যেমন calc()), এবং অন্যান্য আধুনিক কৌশল যেমন feature queries, filters ইত্যাদি।

🧰 প্রপার্টি ও ফাংশন

Feature	Syntax / Values	বাংলা বর্ণনা	Quick Tip
Custom Property	`--name: value;` → `var(--name, fallback)`	থিম-ভ্যালু সংরক্ষণ করে	টেমিং ও ডার্ক-মোডে কাজে লাগে
`calc()`	`calc(expr)`	রান-টাইম গণিত	ইউনিট-মিক্স সহজ করে
`clamp()`	`clamp(min, ideal, max)`	ফ্লুইড রেঞ্জ	ফ্লুইড টাইপ-সাইজের জন্য
`min()` / `max()`	`min(val1, val2)` / `max(val1, val2)`	ক্ষুদ্র/বৃহৎ ভ্যালু নির্বাচন	গ্রিড ফলব্যাকে কাজে লাগে
`color-mix()`	`color-mix(in lch, red 30%, blue)`	দুই রঙ মিশায়	Design-system শেড তৈরি
`color-contrast()`	`color-contrast(pink vs white, black)`	APCA-সহ কন্ট্রাস্ট চেক করে	অটোমেটিক এক্সেসিবিলিটি
Relative Color	`oklch(from var(--brand) l calc(l * 1.2))`	মূল রঙ থেকে নতুন শেড তৈরি করে	L, C, H টুইক সহজ
`@layer`	`@layer base { … }`	কাস্টম ক্যাসকেড লেয়ার তৈরি করে	Utility → Component → Override
CSS Nesting	`selector { & span { … } }`	নেস্টেড সিনট্যাক্স	Chrome/WebKit-এ স্টেবল
`:has()` Parent	`section:has(img:hover)`	সম্পর্কভিত্তিক সিলেক্টর	ইন্টার‍্যাকশন ব্রিজ
Anchor Position	`position-anchor: --pop; top: anchor(top);`	রেফারেন্স মার্কার	Tooltip অটোমেশন
Accent Color	`accent-color: var(--accent);`	ফর্ম উপাদান থিমিং	চেকবক্স/রেডিও দ্রুত থিম করা যায়
Logical Props	`margin-block: 1rem;`	RTL/LTR–এ ইন্টেনশনাল স্পেসিং	লজিক্যাল মার্জিন/প্যাডিং ব্যবহার করুন
`@scope`	`@scope (.card) { … }`	লোকাল ক্যাসকেড সীমাবদ্ধ করে	Stencil-স্টাইল আইসোলেশন

🎯 CSS কাস্টম প্রপার্টি (ভ্যারিয়েবল)

CSS ভ্যারিয়েবল আপনাকে রিইউজেবল ভ্যালু ডিফাইন করতে দেয়।

```
:root {
    --primary-color: #3498db;
    --secondary-color: #2ecc71;
    --text-color: #333;
    --spacing-unit: 1rem;
}

.box {
  background-color: var(--primary-color);
  padding: var(--spacing-unit);
}
.button {
    background-color: var(--primary-color);
    color: white;
    padding: var(--spacing-unit);
}
```

📊 CSS ফাংশন

Function	উদাহরণ	ব্যাখ্যা
`calc()`	`width: calc(100% - 50px);`	ক্যালকুলেশন ভিত্তিক মান নির্ধারণ
`clamp()`	`font-size: clamp(14px, 2vw, 18px);`	সর্বনিম্ন, আদর্শ এবং সর্বোচ্চ মানের সীমা নির্ধারণ
`min()`	`width: min(500px, 100%);`	সর্বনিম্ন মান ব্যবহার
`max()`	`width: max(300px, 50%);`	সর্বোচ্চ মান ব্যবহার
`var()`	`color: var(--text-color);`	কাস্টম ভ্যারিয়েবল ব্যবহার

🎨 CSS গ্রেডিয়েন্ট

```
.gradient-bg {
    /* লিনিয়ার গ্রেডিয়েন্ট */
    background: linear-gradient(45deg, #3498db, #2ecc71);

    /* রেডিয়াল গ্রেডিয়েন্ট */
    background: radial-gradient(circle, #3498db, #2ecc71);

    /* কোনিক গ্রেডিয়েন্ট */
    background: conic-gradient(from 45deg, #3498db, #2ecc71);
}
```

🔄 CSS ট্রান্সফর্ম ফাংশন

```css
.transform-example {
    /* 2D ট্রান্সফর্ম */
    transform: translate(100px, 50px) rotate(45deg) scale(1.5);

    /* 3D ট্রান্সফর্ম */
    transform: perspective(1000px) rotateY(45deg) translateZ(100px);
}
```

🎯 CSS শেপ ফাংশন

```css
.shape {
    /* ক্লিপ-পাথ */
    clip-path: polygon(50% 0%, 100% 50%, 50% 100%, 0% 50%);

    /* বর্ডার-রেডিয়াস */
    border-radius: 30% 70% 70% 30% / 30% 30% 70% 70%;
}
```

📱 CSS নিউ মিডিয়া কুয়েরি ফিচার

```css
/* প্রিফার্ড কালার স্কিম */
@media (prefers-color-scheme: dark) {
    :root {
        --bg-color: #333;
        --text-color: #fff;
    }
}

/* রিডুসড মোশন */
@media (prefers-reduced-motion: reduce) {
    * {
        animation: none !important;
        transition: none !important;
    }
}
```

🎨 CSS মাস্ক এবং ফিল্টার

```css
.effect {
    /* ইমেজ মাস্ক */
    mask-image: linear-gradient(to right, transparent, black);

    /* ফিল্টার */
    filter: blur(5px) brightness(1.2) contrast(1.1);
```

```
    backdrop-filter: blur(10px);
}
```

🚀 Modern CSS Features

Feature	Description
`@supports`	কোনো ব্রাউজার কোনো ফিচার সাপোর্ট করে কি না তা যাচাই
`scroll-behavior`	`smooth` ব্যবহার করে স্ক্রলিং স্মুথভাবে করা
`aspect-ratio`	এলিমেন্টের প্রস্থ ও উচ্চতার অনুপাত নির্ধারণ
`:is()`	একাধিক সিলেক্টরকে গ্রুপ করে
`:where()`	zero-specificity সিলেক্টর গ্রুপ
`container queries`	প্যারেন্ট এলিমেন্টের উপর ভিত্তি করে স্টাইল করা (CSS Container Queries)

🔷 উদাহরণ:

```
@supports (display: grid) {
  .container {
    display: grid;
  }
}

.card:where(.featured, .highlight) {
  border: 2px solid gold;
}
```

🧠 Responsive Typography with `clamp()` (CSS Typography-র ভবিষ্যৎ!)

🔍 `clamp()` কী?

`clamp()` হল একটি CSS ফাংশন যা **responsive value** তৈরি করতে ব্যবহৃত হয়। এটা তিনটি মান নেয়:

```
clamp(MIN, PREFERRED, MAX)
```

- MIN: সবচেয়ে ছোট মান, যেটার নিচে যাবে না।
- PREFERRED: ভিউপোর্ট বা স্ক্রিন সাইজ অনুযায়ী পরিবর্তনশীল মান।
- MAX: সর্বোচ্চ মান, যেটার উপরে যাবে না।

📐 Responsive Typography with `clamp()`

ক্যাম্প করা ফন্ট?
ভাই, কেন একটা মিডিয়া কিউরি নিয়ে মাস্টারি করতে যাচ্ছ, যখন এক লাইনে স্মার্টলি স্কেলিং পেতে পারে?
`clamp()` হল তোমার টেক্সটের হিরো মোমেন্ট—মোবাইল থেকে 4K স্ক্রিন সব জায়গায় স্বয়ংক্রিয় ভাবে ঠিক ঠিক!

⚠️ পুরনো পন্থা vs নতুন ট্রিক

আগে আমরা করতাম:

```
body {
  font-size: 16px;
}

@media (min-width: 768px) {
  body { font-size: 20px; }
}
```

🧪 উদাহরণ: Typography with clamp()

```
h1 {
  font-size: clamp(1.5rem, 2.5vw, 3rem);
}
```

📌 ব্যাখ্যা

- **1.5rem:** ছোট স্ক্রিনে হেডিং সাইজ হবে কমপক্ষে `1.5rem`
- **2.5vw:** ভিউপোর্টের প্রস্থের উপর ভিত্তি করে responsive পরিবর্তন
- **3rem:** বড় স্ক্রিনে হেডিং সাইজ `3rem` –এর বেশি হবে না

🎯 কিভাবে স্মার্টলি ভ্যালু ঠিক করব?

- **MIN:** মোবাইল-ফ্রেন্ডলি রেম/পিক্সেল মান (যেমন: `1rem`)
- **PREFERRED:** viewport–এর width অনুযায়ী `%vw` (যেমন: `2.5vw`)
- **MAX:** ডেস্কটপের জন্য যুক্তিসঙ্গত বড় মান (যেমন: `2.5rem`)

✅ বাস্তব উদাহরণ (Full Typography System):

```
:root {
  --font-base: clamp(1rem, 2vw, 1.5rem);
  --font-lg: clamp(1.25rem, 3vw, 2rem);
```

```
  --font-xl: clamp(1.5rem, 4vw, 2.5rem);
  --font-xxl: clamp(2rem, 6vw, 3.5rem);
}

body {
  font-size: var(--font-base);
}

h2 {
  font-size: var(--font-lg);
}

h1 {
  font-size: var(--font-xl);
}
```

📌 Clamp-এর সুবিধা

- ✅ এক লাইনে **responsive** মান সেট করা যায়
- ✅ **Media Query** লাগেনা
- ✅ মোবাইল থেকে **4K স্ক্রিন**—সব জায়গায় ইউনিফর্ম স্কেলিং
- ✅ **Typography**, **Padding**, **Margin**—যেকোনো কিছুতেই ব্যবহার করা যাবে

🧠 মনে রাখার ট্রিক

> "Clamp দিয়ে টেক্সট স্কেল করে সুন্দরভাবে,
> Media Query ফালায় দাও একদম সাহসে!"

📌 ইম্পর্টেন্ট পয়েন্ট

- Variables দিয়ে **large-scale** প্রজেক্টে কোড **রিইউজ্যাবল** করা যায়
- Functions `calc()` ও `clamp()` **রেসপন্সিভ ডিজাইনের** জন্য অপরিহার্য

📝 প্র্যাকটিস টাস্ক

1. ডার্ক/লাইট থিম সুইচিং:

```
:root {
    --bg: white;
    --text: black;
}

[data-theme="dark"] {
    --bg: #333;
    --text: white;
```

```
}

body {
    background-color: var(--bg);
    color: var(--text);
}
```

2. রেসপন্সিভ টাইপোগ্রাফি:

```
:root {
    --fluid-type: clamp(1rem, 3vw + 0.5rem, 2rem);
}

h1 {
    font-size: var(--fluid-type);
}
```

🤔 MCQ কুইজ

1. CSS ভ্যারিয়েবল কিভাবে ডিফাইন করা হয়?
    - a) $variable
    - b) --variable
    - c) @variable
    - d) _variable
2. var() ফাংশনে দ্বিতীয় প্যারামিটার কি?
    - a) ডিফল্ট ভ্যালু
    - b) অপশনাল ভ্যালু
    - c) ফলব্যাক ভ্যালু
    - d) অল্টারনেটিভ ভ্যালু
3. clamp() ফাংশনে কয়টি প্যারামিটার থাকে?
    - a) 2
    - b) 3
    - c) 4
    - d) 1
4. কোনটি CSS কাস্টম প্রপার্টির স্কোপ?
    - a) গ্লোবাল
    - b) লোকাল

- c) দুটোই হতে পারে
- d) কোনটিই না

5. backdrop-filter কি করে?

- a) এলিমেন্টের পিছনে ফিল্টার
- b) এলিমেন্টের সামনে ফিল্টার
- c) শুধু টেক্সটে ফিল্টার
- d) শুধু ইমেজে ফিল্টার

6. CSS মাস্ক-ইমেজ কি?

- a) ইমেজ হাইড করে
- b) ইমেজ ক্রপ করে
- c) ইমেজের ভিজিবিলিটি কন্ট্রোল করে
- d) ইমেজ রোটেট করে

7. prefers-color-scheme মিডিয়া কুয়েরি কি চেক করে?

- a) ব্রাউজার থিম
- b) ডিভাইস থিম
- c) ইউজার থিম প্রিফারেন্স
- d) ওয়েবসাইট থিম

8. linear-gradient() এ কয়টি কালার ইউজ করা যায়?

- a) শুধু 2টি
- b) শুধু 3টি
- c) যত ইচ্ছা
- d) সর্বোচ্চ 5টি

9. mask-image এর ডিফল্ট ভ্যালু কি?

- a) none
- b) transparent
- c) black
- d) white

10. CSS ভ্যারিয়েবল আপডেট করলে:

- a) পেজ রিলোড লাগে
- b) অটো আপডেট হয়
- c) জাভাস্ক্রিপ্ট লাগে
- d) CSS রিলোড লাগে

📚 হোমওয়ার্ক

1. একটি ডার্ক/লাইট থিম সিস্টেম তৈরি করুন:
    - CSS ভ্যারিয়েবল ব্যবহার
    - থিম সুইচিং
    - স্মুথ ট্রানজিশন
    - লোকাল স্টোরেজে থিম সেভ
2. একটি মডার্ন কার্ড কম্পোনেন্ট:
    - গ্রেডিয়েন্ট ব্যাকগ্রাউন্ড
    - ব্লার এফেক্ট
    - ক্লিপ-পাথ শেপ
    - হোভার এফেক্ট
3. রেসপন্সিভ হিরো সেকশন:
    - ফ্লুইড টাইপোগ্রাফি
    - ডায়নামিক স্পেসিং
    - ইমেজ মাস্কিং
    - পারফরমেন্স অপটিমাইজেশন
4. আপনার কোড গিটহাবে আপলোড করুন

📖 অধ্যায় ১৩: Extra CSS প্রপার্টি এবং ফিচার

এই অধ্যায়ে CSS-এর বাকি সকল অতিরিক্ত মডিউল, প্রপার্টি, ফিচার ও এ্যাট-রুলের মিশ্রণ নিয়ে আলোচনা করব, যা ১২টি মূল অধ্যায়ে কাভার হয়নি: শেপ, স্ক্রল স্ন্যাপ, মাল্টি-কলাম, ফিল্টার, প্রিন্ট, ইউআই টুইকস, এবং আরও অনেক কিছু।

✨ এডিশনাল সকল CSS প্রপার্টি & ফিচার:

Feature / Property	Syntax / Values	বাংলা বর্ণনা	Quick Tip		
Feature Query	`@supports (display: grid) { … }`	CSS ফিচার সাপোর্ট চেক করে	থার্ড-পার্টি উইজেট ইত্যাদি ওভাররাইডে কাজে লাগে		
Clip Path	`clip-path: inset() \	circle() \	polygon()`	এলিমেন্টের ভিজিবল	GPU-অপ্টিমাই

Feature / Property	Syntax / Values	বাংলা বর্ণনা	Quick Tip		
				এরিয়া কেটে ফেলে	ইফেক্টে ইউ করুন
Shape Outside	`shape-outside: circle(50%); shape-margin: 1rem;`	ভাসমান কন্টেন্ট চারপাশে টেক্সট ফ্লো কন্ট্রোল করে	মাল্টি-কলাম নিউজপেপার স্টাইলের জন্য পারফেক্ট		
Scroll Snap	`scroll-snap-type: x mandatory; scroll-snap-align: center;`	স্ক্রল পজিশনে স্বয়ংক্রিয় স্টপ	কার্ড গ্যালারিতে স্মুথ স্ন্যাপিং		
Smooth Scroll	`scroll-behavior: smooth;`	মসৃণ স্ক্রল এনিমেশন	URI হ্যাশ লিঙ্কে এফেক্টিভ		
Multi-column	`column-count: 3; column-gap: 2rem; column-rule: 1px solid #ccc;`	টেক্সট মাল্টি-কলামে ভাগ করে দেখায়	আর্টিকেলে পেজ-ব্রেক ফুটার কমাতে		
Hyphens	`hyphens: auto;`	শব্দ বিভাজন চালু করে	locale সাপোর্ট নিশ্চিত করুন		
Object-fit / position	`object-fit: cover; object-position: top right;`	ইমেজ বা ভিডিওর ফিটিং ও পজিশনিং	রেস্পনসিভ মিডিয়া হেডারে ইউজ করুন		
Filter Effects	`filter: blur(4px) brightness(.8);`	CSS-ভিত্তিক ছবি/ব্রাউজার এফেক্ট	হোভার-এ ফোকাস ডিপথ তৈরি করে		
Backdrop-filter	`backdrop-filter: blur(10px);`	ব্যাকগ্রাউন্ড ব্লার বা ব্রাইটনেস	গ্লাসমরফিজম ইফেক্টের মাস্ট টুল		
Masking	`mask-image: url(mask.svg); mask-mode: alpha;`	এলিমেন্টে মাস্ক ইমেজ অ্যাপ্লাই করে	স্প্রাইট চার্টে নির্দিষ্ট অংশ দেখাতে		

Feature / Property	Syntax / Values	বাংলা বর্ণনা	Quick Tip		
`@page`	`@page { margin:1in; size:A4 landscape; }`	প্রিন্ট পেজের মার্জিন ও সাইজ সেট করে	হেডার/ফুটার ও পেজ নম্বরিং এ কাজে লাগে		
cursor	`cursor: pointer;` \	`url(cursor.svg), auto;`	মাউস কার্সরের স্টাইল নির্ধারণ করে	UX হিন্ট হিসেবে কাস্টম কার্সর ব্যবহার করুন	
user-select	`user-select: none \	text \	all;`	টেক্সট সিলেকশন কন্ট্রোল করে	ড্র্যাগ টুল বা কাস্টম UI-তে দরকারি
pointer-events	`pointer-events: none \	auto;`	মাউস/টাচ ইন্টারঅ্যাকশন ইনহিবিট বা এ্যালাউ করে	ওভারলে বা ড্র্যাগ-ড্রপ UI-তে কাজে লাগে	
scrollbar-color	`scrollbar-color: #888 #ccc; scrollbar-width: thin;`	Firefox-এ স্ক্রলবার থিম ও থিকনেস কন্ট্রোল করে	Chrome-এ `::-webkit-scrollbar` ব্যবহার করুন		
resize	`resize: none \	both \	horizontal \	vertical;`	এলিমেন্ট রিসাইজ করার অনুমতি নিষেধ
`@import` / `@namespace`	`@import url(a.css); @namespace svg url(...);`	স্টাইলশিট বা XML নেমস্পেস লোড করে	প্রয়োজনমতো কম `@import` ইউজ করুন		
`@media print`	`@media print { … }`	প্রিন্ট-স্পেসিফিক CSS ব্লক চালু করে	নন-প্রিন্ট ফিচার হাইড করতে ব্যবহার করুন		

```css
.custom-element {
    cursor: pointer;
    user-select: none;
    resize: both;
    pointer-events: none;
}
```

📊 স্ক্রল বিহেভিয়ার প্রপার্টি

```css
.scroll-container {
    /* স্ক্রল বিহেভিয়ার */
    scroll-behavior: smooth;
    scroll-snap-type: y mandatory;
    overscroll-behavior: contain;

    /* স্ক্রলবার কাস্টমাইজেশন */
    scrollbar-width: thin;
    scrollbar-color: #888 #f1f1f1;
}

.scroll-item {
    scroll-snap-align: start;
    scroll-margin-top: 1rem;
}
```

🎨 মিক্সিং বিহেভিয়ার প্রপার্টি

```css
.mixed-blend {
    /* এলিমেন্টের মধ্যে মিক্সিং */
    mix-blend-mode: multiply;

    /* ব্যাকগ্রাউন্ডের সাথে মিক্সিং */
    background-blend-mode: overlay;

    /* আইসোলেশন */
    isolation: isolate;
}
```

🔤 রাইটিং মোড প্রপার্টি

```css
.writing-modes {
    /* রাইটিং মোড */
    writing-mode: vertical-rl;
    text-orientation: upright;
    direction: rtl;
```

```
    unicode-bidi: bidi-override;
}
```

🎯 কানটেন্ট জেনারেশন প্রপার্টি

```
.generated-content {
    /* কানটেন্ট জেনারেশন */
    content: attr(data-tooltip);
    quotes: "«" "»";
    counter-reset: section;
    counter-increment: section;

    /* কানটেন্ট প্লেসমেন্ট */
    content-visibility: auto;
    contain: content;
}
```

🖼️ ইমেজ রেন্ডারিং প্রপার্টি

```
.image-rendering {
    /* ইমেজ রেন্ডারিং */
    image-rendering: pixelated;
    object-fit: cover;
    object-position: center;

    /* ইমেজ সেট */
    image-set(
        "img.png" 1x,
        "img@2x.png" 2x
    );
}
```

🎨 কালার অ্যাডজাস্টমেন্ট প্রপার্টি

```
.color-adjustments {
    /* কালার অ্যাডজাস্টমেন্ট */
    color-adjust: economy;
    forced-color-adjust: none;
    print-color-adjust: exact;
}
```

📱 টাচ অ্যাকশন প্রপার্টি

```
.touch-action {
    /* টাচ বিহেভিয়ার */
```

```
    touch-action: pan-x pinch-zoom;
    -webkit-tap-highlight-color: transparent;
    -webkit-touch-callout: none;
}
```

🎯 কালাম প্রপার্টি

```
.columns {
    /* মাল্টি-কালাম লেআউট */
    columns: 3 200px;
    column-gap: 2rem;
    column-rule: 1px solid #ccc;
    column-span: all;
    break-inside: avoid;
}
```

📝 প্র্যাকটিস টাস্ক

1. কাস্টম স্ক্রলবার:

```
.custom-scrollbar {
    scrollbar-width: thin;
    scrollbar-color: #6366f1 #fff;
}

.custom-scrollbar::-webkit-scrollbar {
    width: 8px;
}

.custom-scrollbar::-webkit-scrollbar-track {
    background: #fff;
}

.custom-scrollbar::-webkit-scrollbar-thumb {
    background: #6366f1;
    border-radius: 4px;
}
```

2. মাল্টি-কালাম আর্টিকেল:

```
.article {
    columns: 2;
    column-gap: 2rem;
    column-rule: 1px solid #ccc;
}

.article h2 {
```

```
    column-span: all;
    margin-bottom: 1rem;
}
```

🤔 MCQ কুইজ

1. scroll-behavior: smooth; এর কাজ কি?
    - a) স্ক্রল বন্ধ করা
    - b) স্মুথ স্ক্রল অ্যানিমেশন
    - c) স্ক্রল স্পিড বাড়ানো
    - d) স্ক্রল লক করা
2. object-fit প্রপার্টি কোথায় ব্যবহৃত হয়?
    - a) সব এলিমেন্টে
    - b) শুধু ইমেজে
    - c) শুধু ভিডিওতে
    - d) ইমেজ এবং ভিডিওতে
3. pointer-events: none; এর ফলাফল:
    - a) মাউস ইভেন্ট ডিসেবল
    - b) কার্সর হাইড
    - c) ক্লিক ইভেন্ট শো
    - d) হোভার ইফেক্ট শো
4. columns প্রপার্টির ডিফল্ট ভ্যালু:
    - a) 1
    - b) auto
    - c) none
    - d) 0
5. mix-blend-mode কি করে?
    - a) কালার মিক্স
    - b) এলিমেন্ট মিক্স
    - c) টেক্সট মিক্স
    - d) ব্যাকগ্রাউন্ড মিক্স
6. writing-mode প্রপার্টি কি পরিবর্তন করে?
    - a) ফন্ট স্টাইল

- b) টেক্সট ডাইরেকশন
- c) টেক্সট অ্যালাইনমেন্ট
- d) টেক্সট কালার

7. content-visibility: auto; এর উপকারিতা:
   - a) পারফরম্যান্স বাড়ায়
   - b) কনটেন্ট হাইড করে
   - c) কনটেন্ট শো করে
   - d) কনটেন্ট ডিলিট করে
8. touch-action প্রপার্টি কোথায় ব্যবহৃত হয়?
   - a) ডেস্কটপে
   - b) মোবাইলে
   - c) টাচস্ক্রীনে
   - d) সব ডিভাইসে
9. forced-color-adjust এর ব্যবহার:
   - a) হাই কনট্রাস্ট মোডে
   - b) ডার্ক মোডে
   - c) লাইট মোডে
   - d) প্রিন্ট মোডে
10. break-inside: avoid; এর কাজ:
    - a) পেজ ব্রেক রোধ
    - b) লাইন ব্রেক রোধ
    - c) কালাম ব্রেক রোধ
    - d) কনটেন্ট ব্রেক রোধ

📚 হোমওয়ার্ক

1. একটি মাল্টি-কালাম ব্লগ লেআউট তৈরি করুন:
   - রেসপন্সিভ কালাম
   - কাস্টম স্ক্রলবার
   - অপটিমাইজড ইমেজ
   - প্রিন্ট স্টাইলিং
2. ইন্টারঅ্যাক্টিভ টাচ ইন্টারফেস:
   - টাচ জেসচার

- স্ক্রল স্ন্যাপিং
- স্মুথ অ্যানিমেশন
- হাই কনট্রাস্ট সাপোর্ট

3. কাস্টম কনটেন্ট টুলটিপ:

- অ্যাট্রিবিউট বেইজড কনটেন্ট
- পজিশনিং
- অ্যানিমেশন
- অ্যাক্সেসিবিলিটি

4. আপনার কোড গিটহাবে আপলোড করুন...

ধন্যবাদ আপনাকে!

সম্পূর্ণ ফ্রিতে বইটার আপডেট ও লেটেস্ট ভার্সন পেতে ভিজিট করুন 👇👇👇

📩 Download: “CSS শিখুন পানির মতো সহজ করে” - https://sorobindu.com/get-css-pdf-book/