Kolleg/Aufbaulehrgang
für Informatik

Schriftliche Leistungsüberprüfung
WMC (Web- und mobile Computing), 3BKIF
18. Dezember 2025

Name: _____ Punkte: _____

Hinweis: Es können 0 bis 4 Antworten pro Frage richtig sein. Kreuzen Sie alle zutreffenden Aussagen an.

HTML

1. HTML-Attribute (Boolean, data-*, id)
Welche Aussagen sind technisch korrekt?

- ☐ Ein Boolean-Attribut (z. B. `disabled`, `checked`) gilt als *wahr*, wenn es im Tag vorhanden ist (unabhängig vom Wert).
- ☐ `disabled="false"` macht ein Formularfeld wieder aktiv, weil der Wert `false` ausgewertet wird.
- ☐ `data-foo="bar"` kann in JavaScript typischerweise über `element.dataset.foo` ausgelesen werden.
- ☐ Eine `id` darf beliebig oft im selben HTML-Dokument vorkommen, weil CSS/JS ohnehin "alle" findet.

2. Formulare und Validierung
Welche Aussagen sind korrekt?

- ☐ Das Attribut `for` eines `<label>` muss auf die `id` des zugehörigen Form-Controls zeigen.
- ☐ Ein `placeholder` ersetzt ein `<label>` vollständig und ist aus Accessibility-Sicht gleichwertig.
- ☐ `required` aktiviert Browser-Validierung, die durch `novalidate` am `<form>` unterdrückt werden kann.
- ☐ `autocomplete="off"` garantiert in allen Browsern, dass keine Auto-Vervollständigung stattfindet.

3. Parsing, DOM und Fehlerkorrektur
Welche Aussagen treffen zu?

- ☐ Ohne `<!doctype html>` wird immer automatisch der Standards-Modus verwendet.
- ☐ Whitespace zwischen Inline-Elementen erzeugt im DOM niemals Textknoten.
- ☐ Das DOM ist immer identisch mit dem ursprünglichen HTML-Quelltext und kann sich zur Laufzeit nicht ändern.
- ☐ Der HTML-Parser kann bei ungültiger Verschachtelung den DOM-Baum korrigieren (z. B. Tags implizit schließen).

CSS

4. Cascade, Spezifität und Sonderfälle
Welche Aussagen sind korrekt?

- ☐ Eine Declaration mit `!important` schlägt eine ansonsten gleichartige Declaration ohne `!important` (unabhängig von der Spezifität).

- ☐ Selektoren in `:where(...)` tragen *keine* Spezifität bei.
- ☐ Inline-Styles werden immer von späteren Stylesheet-Regeln überschrieben, auch ohne `!important`.
- ☐ Eine später definierte Regel gewinnt immer, selbst wenn sie weniger spezifisch ist.

5. Box Model und `box-sizing`
Welche Aussagen sind korrekt?

- ☐ Standardmäßig ist `box-sizing: content-box` aktiv.
- ☐ Bei `box-sizing: border-box` umfasst `width` auch `padding` und `border`.
- ☐ `margin` gehört zur "Box" und wird von `background-color` mit eingefärbt.
- ☐ `padding` darf negativ sein, um Elemente enger zu machen.

6. Selektoren: fortgeschritten
Welche Aussagen sind korrekt?

- ☐ `a[href^="https://"]` selektiert Links, deren `href` mit `https://` beginnt.
- ☐ `.card:not(.active)` selektiert Elemente mit Klasse `card`, die *nicht* die Klasse `active` besitzen.
- ☐ `ul > li:first-child` selektiert ein `li`, das erstes Kind *und* direktes Kind eines `ul` ist.
- ☐ `input[type="checkbox"]:checked` selektiert aktivierte Checkboxen.

7. Flexbox: typische Denkfehler
Welche Aussagen sind korrekt?

- ☐ Bei `flex-direction: row` ist die Hauptachse vertikal.
- ☐ `justify-content` richtet Elemente entlang der Querachse aus.
- ☐ `flex-wrap` ist standardmäßig `wrap`.
- ☐ `order` ändert automatisch die Tab-Reihenfolge und Screenreader-Reihenfolge identisch zur visuellen Reihenfolge.

8. CSS Grid: `repeat`, `minmax`, implizite Tracks
Gegeben ist folgender CSS-Code:

```css
.container {
  display: grid;
  grid-template-columns: repeat(3, minmax(120px, 1fr));
  grid-auto-rows: 80px;
  gap: 8px;
}
```

Welche Aussagen sind korrekt?

- ☐ Es entstehen drei Spalten, jede mindestens 120px breit; restlicher Platz wird mit `fr` verteilt.
- ☐ `grid-auto-rows` steuert die Höhe von implizit entstehenden Zeilen.
- ☐ `gap` setzt sowohl Zeilen- als auch Spaltenabstand (hier: 8px).
- ☐ Grid-Items werden automatisch zu `display: block`, egal welchen `display`-Wert sie zuvor hatten.

9. Webfonts und Fallbacks
Welche Aussagen sind korrekt?

- ☐ In `font-family: "Inter", Arial, sans-serif;` wird die erste verfügbare Schrift aus der Liste verwendet.
- ☐ `font-display: swap` kann "unsichtbaren Text" vermeiden, indem zuerst ein Fallback gerendert wird.
- ☐ In `@font-face` darf `src` nur genau ein einziges Format enthalten.
- ☐ Webfonts funktionieren nur bei Same-Origin; CORS blockiert externe Fonts immer.

Beurteilungsschlüssel für 36 Punkte: 0-18 / 19-22 / 23-27 / 28-31 / 32-36

Viel Erfolg!