

013-Kennzahlen-Kacheln-Tooltips

Frontend-Aufgabe
Anforderungsniveau: Einfach

1 Ziel der Aufgabe

Sie bauen eine Kachelreihe, die Betriebskennzahlen eines Dienstleisters anzeigt. Ein Teil der Werte wird gekürzt oder abgekürzt dargestellt. Geprüft wird zweierlei: ob Sie Zahlen mit `Intl.NumberFormat` sauber formatieren, und ob Sie Tooltips genau dort setzen, wo durch die Darstellung Information verloren geht – und nur dort. Alles andere an dieser Aufgabe ist Beiwerk.

Veranschlagte Bearbeitungszeit: 2–3 Stunden.

Verwendete Techniken:

- React mit Next.js (App Router) oder Vite, TypeScript
- Formatierung von Zahlen, Währung und Vorzeichen mit `Intl.NumberFormat`
- Zugängliche Tooltips (Maus, Tastatur, Screenreader)

2 Aufgabenbeschreibung

1. Projekt aufsetzen

Legen Sie ein Projekt mit React und TypeScript an, entweder mit Next.js (App Router) oder mit Vite. Für die Gestaltung ist Ihnen die Wahl freigestellt.

```
npm create vite@latest kennzahlen -- --template react-ts
cd kennzahlen && npm install
npm install lucide-react # ODER: npm install @phosphor-icons/react
```

Für Icons gilt nur die knappe Regel der übrigen Aufgabenserie: Sie stammen aus *einem* Set (Lucide oder Phosphor, nicht gemischt, keine Emojis als UI-Icon). Mehr ist dazu nicht verlangt.

2. Datenquelle anlegen

Legen Sie die Kennzahlen als lokale Datei `src/data/kennzahlen.json` ab und importieren Sie sie direkt; ein Ladezustand ist nicht gefordert. Verwenden Sie die folgenden fünf Einträge unverändert – sie sind so gewählt, dass jede Kachel einen anderen Formatierungsfall abdeckt:

```
[
  { "id": "mttr", "kuerzel": "MTTR", "bezeichnung": "Mean Time To Repair",
    "wert": 4.2, "einheit": "h", "trend": -0.6 },
  { "id": "tickets", "kuerzel": "Offene Tickets",
    "bezeichnung": "Offene Tickets (alle Mandanten)",
    "wert": 1284, "einheit": "", "trend": 112 },
  { "id": "sla", "kuerzel": "SLA",
    "bezeichnung": "Service Level Agreement - Erfüllungsquote",
    "wert": 98.7, "einheit": "%", "trend": 0.4 },
  { "id": "umsatz", "kuerzel": "Umsatz MTD",
    "bezeichnung": "Umsatz seit Monatsbeginn",
```

```

    "wert": 1482905.4, "einheit": "EUR", "trend": -23410.15 },
  { "id": "reaktion", "kuerzel": "Ers. Reaktion",
    "bezeichnung": "Durchschnittliche Erstreaktionszeit",
    "wert": 12, "einheit": "min", "trend": 0 }
]

```

3. Kachel und Zahlenformatierung

Eine Kachel `KennzahlKachel` zeigt oben das Kürzel, mittig den formatierten Wert mit Einheit, darunter den Trend gegenüber dem Vormonat und unten rechts einen Icon-Button ohne sichtbare Beschriftung (*Details anzeigen*).

Für die Formatierung gilt durchgehend das deutsche Gebietsschema und ausschließlich `Intl.NumberFormat`; eigene Ersetzungen mit `replace` oder `toFixed` sind nicht zulässig.

- Hauptwerte ab 10 000 werden kompakt dargestellt (`notation: compact`), darunter genau.
- Der Trend wird *nie* gekürzt und trägt immer ein Vorzeichen – auch das Plus.
- Der Umsatz wird als Währung ausgegeben (`style: currency, currency: EUR`), nicht als Zahl mit angehängtem Kürzel.
- Nachkommastellen werden bewusst gesetzt statt zufällig übernommen: 4,2 h behält eine Stelle, der genaue Umsatz zwei.

```

const kompakt = new Intl.NumberFormat("de-DE", {
  notation: "compact",
  maximumFractionDigits: 1,
});
const genau = new Intl.NumberFormat("de-DE", {
  minimumFractionDigits: 2,
  maximumFractionDigits: 2,
});
const trend = new Intl.NumberFormat("de-DE", {
  signDisplay: "exceptZero",
  maximumFractionDigits: 2,
});

```

Die genaue Zeichenfolge liefert die Laufzeitumgebung – `kompakt` gibt für den Umsatz etwa „1,5 Mio.“ aus, `genau` dagegen 1.482.905,40. Verlassen Sie sich auf `Intl` und schreiben Sie kein Ergebnis fest.

4. Tooltips setzen – der Kern der Aufgabe

Setzen Sie Tooltips ausschließlich dort, wo dem Betrachter Information fehlt:

- **Abkürzungen und Fachbegriffe:** MTTR, SLA, MTD, Ers. – und jede weitere Abkürzung, die in den Daten auftaucht. Der Tooltip nennt die ausgeschriebene Bezeichnung.
- **Gekürzte Zahlen:** Wo „1,5 Mio.“ steht, nennt der Tooltip den genauen Wert mit Einheit.
- **Icon-Button:** Der Button ohne sichtbare Beschriftung bekommt einen Tooltip mit seiner Bedeutung.

Kein Tooltip gehört an Werte, die vollständig sichtbar sind (4,2 h, 12 min, 98,7 %), keiner wiederholt bloß den danebenstehenden Text, und keiner sitzt auf einer ganzen Kachel statt auf der einen Stelle, um die es geht. Nutzen Sie eine Tooltip-Bibliothek Ihrer Wahl, etwa `@radix-ui/react-tooltip` oder `@floating-ui/react`.

5. Raster und Trend

Die Kacheln stehen in einem Raster, das auf schmalen Bildschirmen einspaltig umbricht. Positive und negative Trends unterscheiden sich durch Icon *und* Farbe. Fokus-Zustände der Buttons müssen sichtbar sein.

3 Anforderungen

- Alle Zahlen laufen über `Intl.NumberFormat`. Kompakte und genaue Darstellung sind getrennte Formatierer, der Trend trägt ein Vorzeichen, der Umsatz ist eine Währungsausgabe.
- Tooltips erscheinen bei Hover *und* bei Tastaturfokus und sind über `aria-describedby` mit ihrem Auslöser verbunden. Ein Tooltip enthält niemals die einzige Beschriftung eines Bedienelements.
- Icons sind dekorativ und damit `aria-hidden`; die Bedeutung des Icon-Buttons steht in `aria-label`. Ein Icon-Set, keine Mischung.
- TypeScript ist streng konfiguriert (`strict: true`), der Build läuft ohne Fehler durch.
- Keine UI-Bibliothek, die fertige Kennzahl-Kacheln mitbringt. Tooltip- und Icon-Bibliothek sind erlaubt, das Layout schreiben Sie selbst.

4 Abgabe

- Ein Git-Repository mit nachvollziehbaren Commits. Keine `node_modules` im Repository.
- Eine `README.md` mit der Startanleitung (Installation, Entwicklungsserver, Build) und einer kurzen Tabelle: welche Stelle hat einen Tooltip bekommen, welche bewusst nicht – jeweils mit einem Satz Begründung.

```
npm install
npm run dev
npm run build
```

5 Bewertungskriterien

- **Tooltip-Urteil (stärkste Gewichtung)**: Sitzen die Tooltips genau an den Stellen mit Informationsverlust? Fehlt einer, wo gekürzt wird? Steht einer dort, wo ohnehin alles sichtbar ist? Hängt der Tooltip am betroffenen Element oder pauschal an der Kachel?
- **Zahlenformatierung**: korrektes deutsches Format, saubere Trennung von kompakter und genauer Darstellung, Vorzeichen beim Trend, Währung als Währung, keine Handarbeit an den Zeichenketten.
- **Zugänglichkeit**: Tooltip per Tastatur erreichbar, Icon-Button hat einen Namen, Fokus ist sichtbar.
- **Codequalität**: Typisierung, wiederverwendbare Kachelkomponente, Formatierer einmal angelegt statt bei jedem Rendern neu, verständliche Namen.
- **README**: Ist die Begründungstabelle nachvollziehbar? Sie zeigt, ob die Entscheidungen bewusst getroffen wurden.