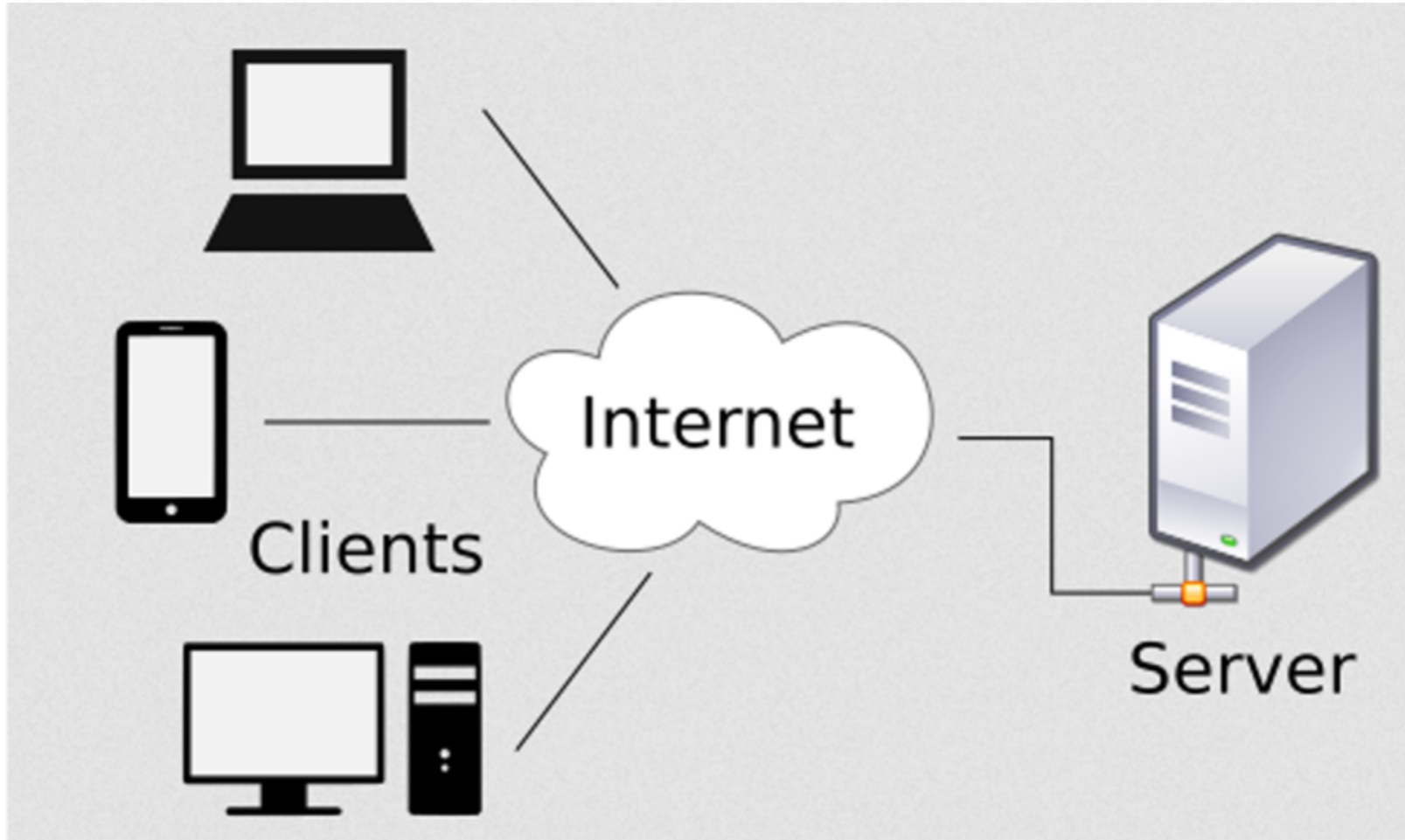


Grundlagen

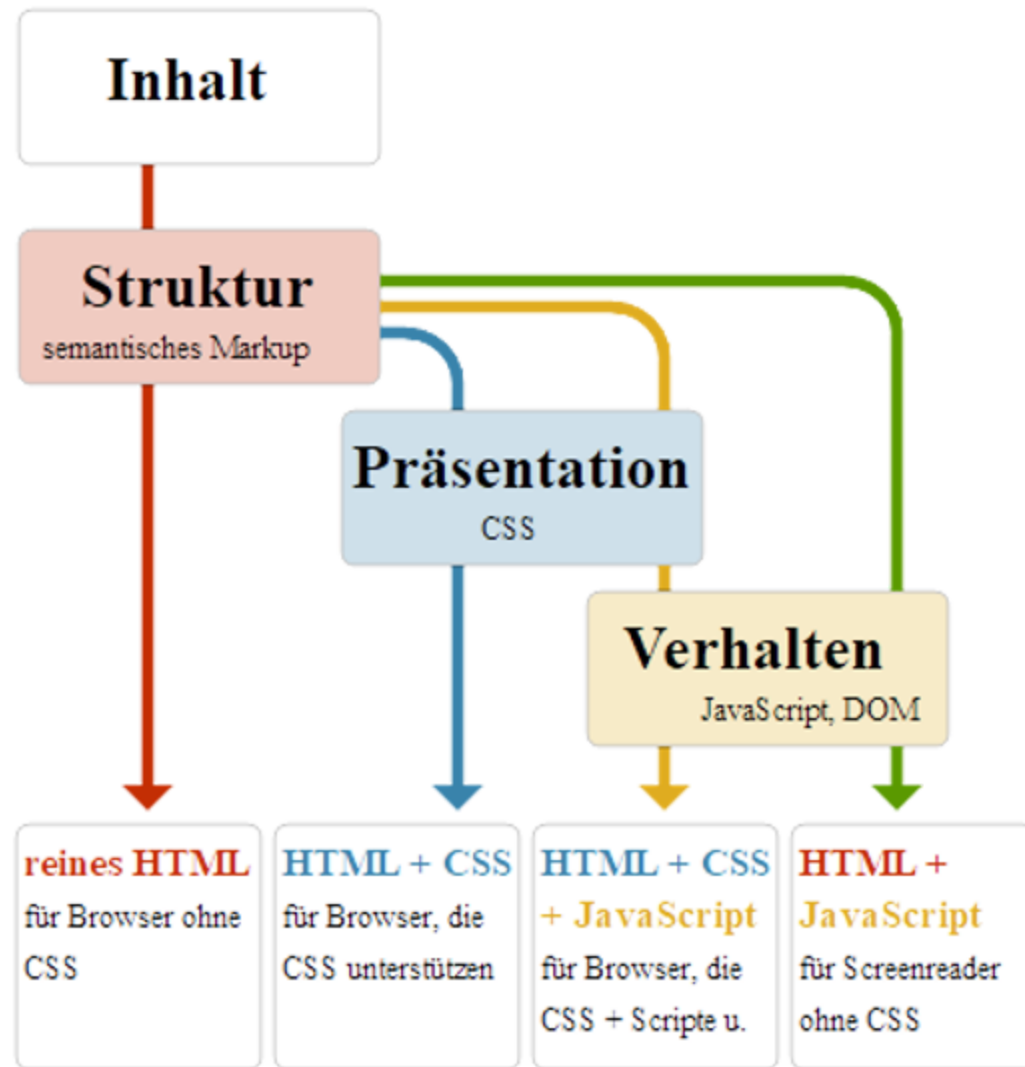
Client - Server



URL

<https://www.philipackermann.de:80/books/web.html?language=de#chapter7>

Aufbau von Webapplikationen





Linux



Apache



MySQL



PHP

MERN



Werkzeuge

Inspektor

Konsole

Debugger

Netzwerkanalyse

Stilbearbeitung

Laufzeitanalyse

Speicher

>>

...

X

HTML durchsuch

+

✎

Stile filtern

:hov .cls

+

📄

Layout

Berechnet

Änderungen

Schriftarten

▼

<!DOCTYPE html>

<html lang="de">

>

</html>

<body>

>

</body>

</html>

Element {

body, h1, dark.css:7

h2, h3, h4,

a, p, blockquote cite {

color: #ddd;

}

body { dark.css:2

background-color: #333;

}

body { style.css:8

font-family: sans-serif;

}

body { normalize.css:23

margin: 0;

}

Geerbt von html

html { normalize.css:11

line-height: 1.15;

webkit-text-size-adjust: 100%;

}

▼ Flexbox

Flex-Behälter oder -Element auswählen, um fortzufahren.

▼ Raster

Raster hervorheben

☐ ul.wp-block-gallery.columns-2.is-cropped

☐ ul.wp-block-gallery.columns-3.is-cropped

☐ ul.wp-block-gallery.columns-3.is-cropped

☐ ul.wp-block-gallery.columns-3.is-cropped

Anzeigeeinstellungen für Raster

☒ Zeilennummern anzeigen

☐ Bereichsnamen anzeigen

☐ Unendliche Linien

▼ Box-Modell

margin

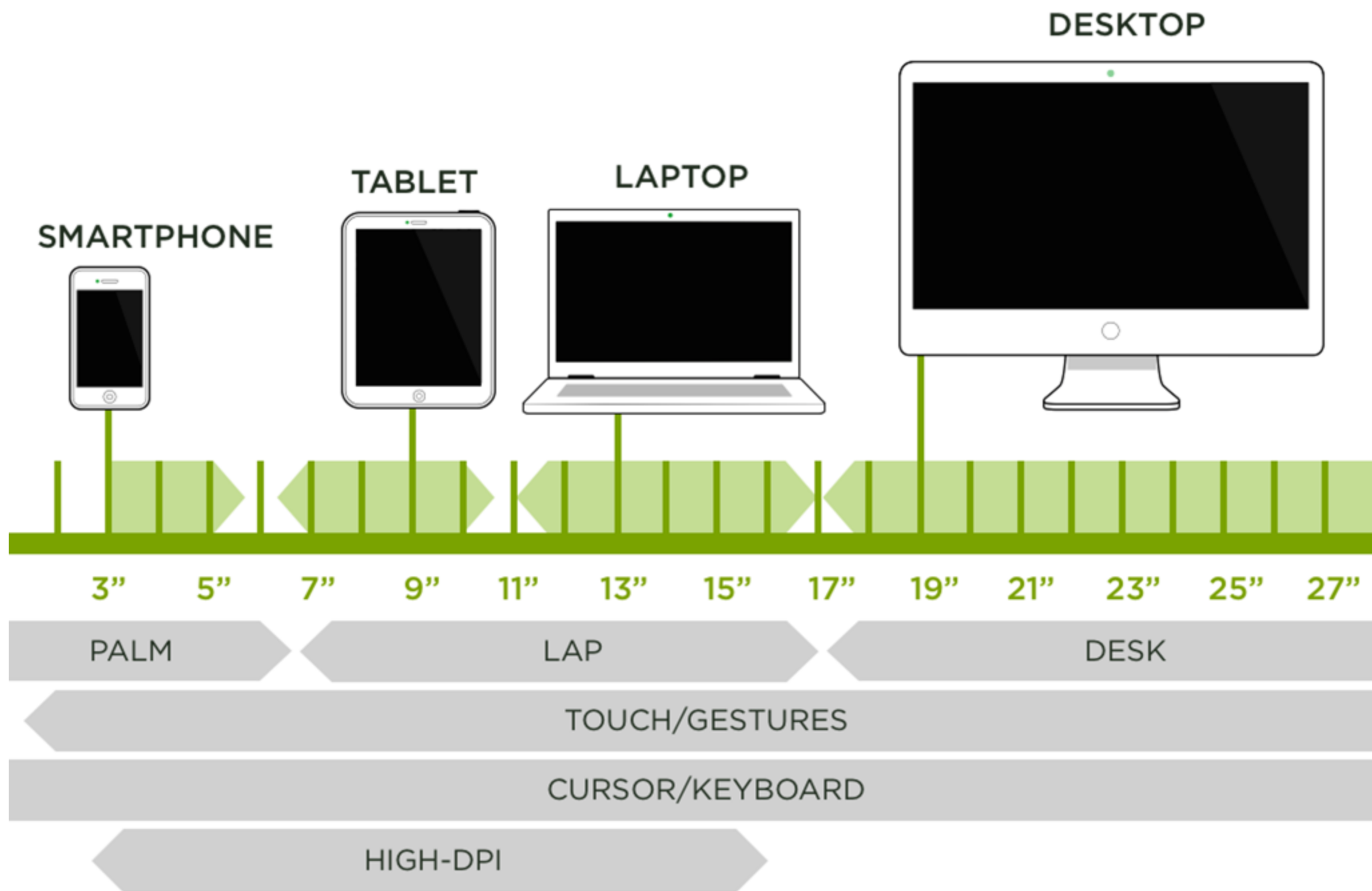
border

padding

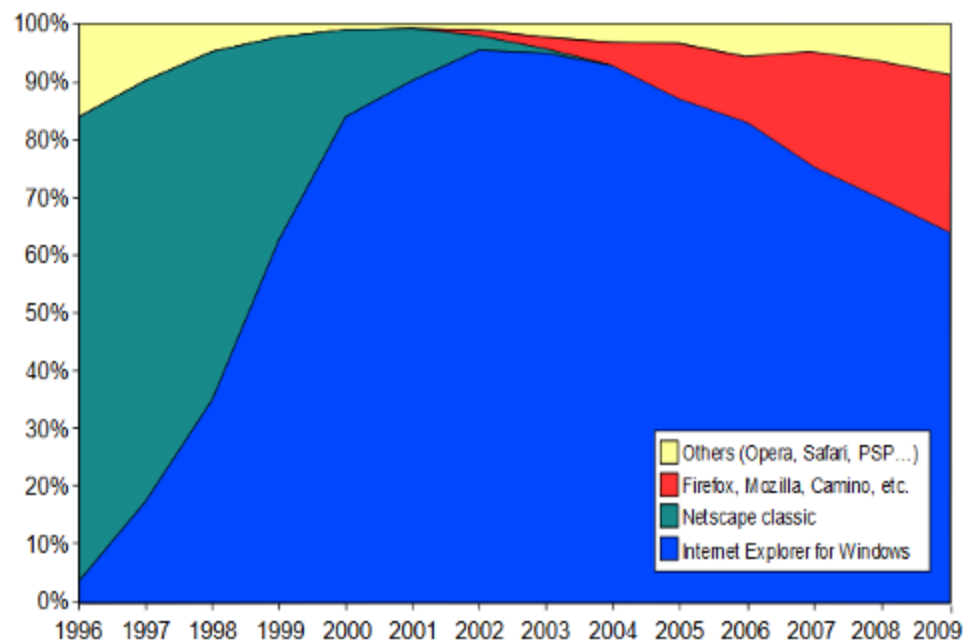
823 × 2397.2

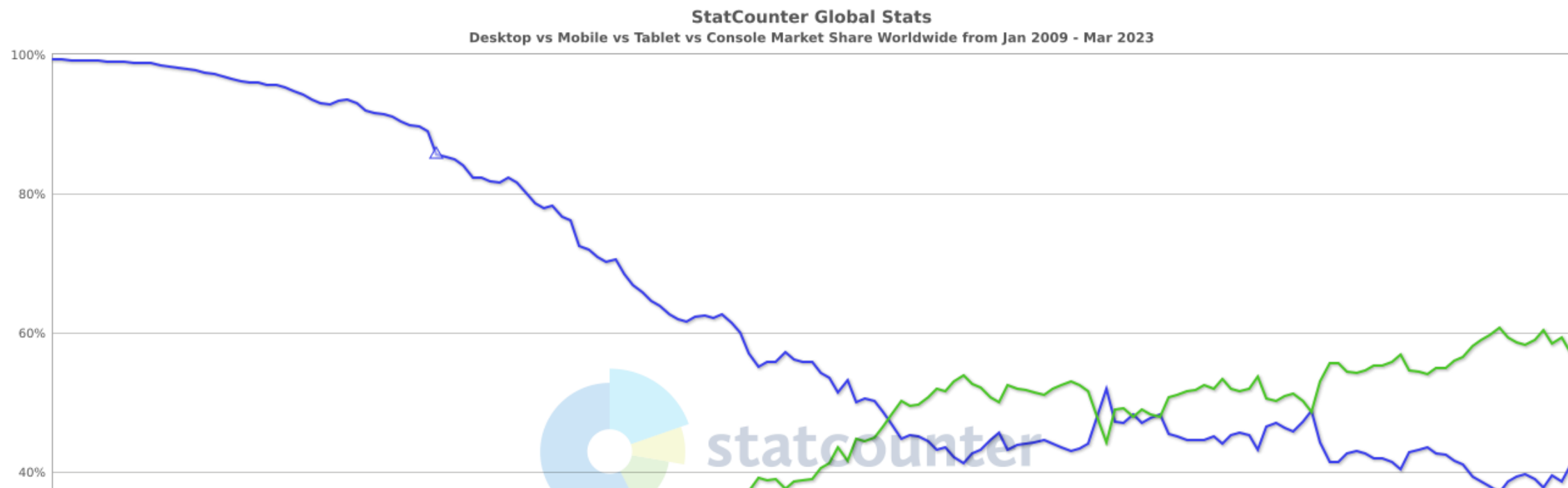
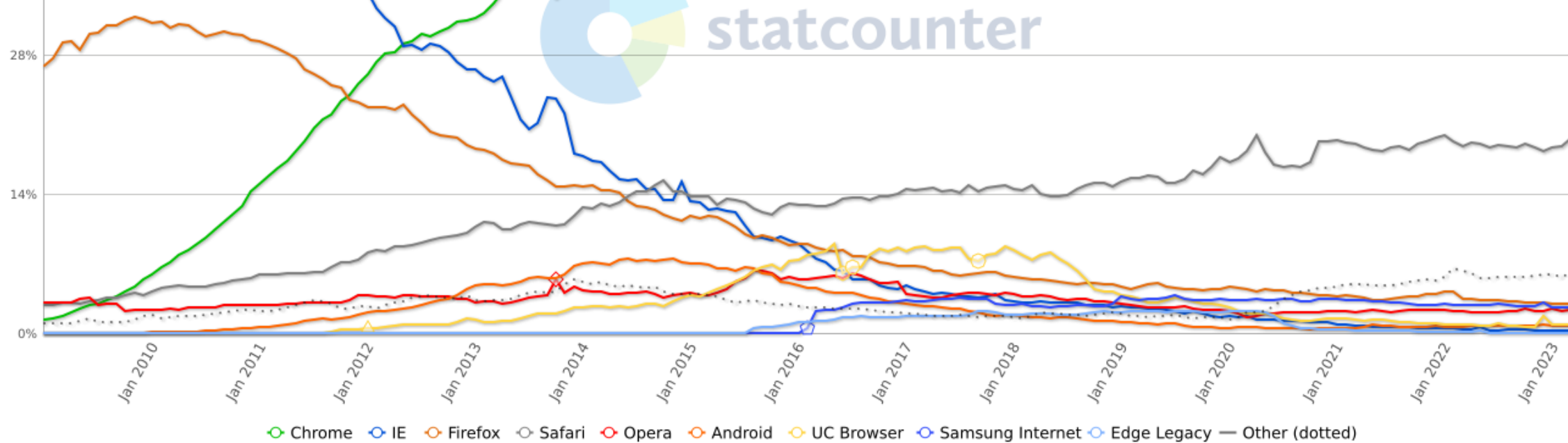
html > body

Webseiten strukturieren mit HTML



Browser Wars





Layoutkonzepte

- <http://info.cern.ch/hypertext/WWW/TheProject.html>
- Framesets
- Tabellen
- Cascading Style Sheets (CSS)
- Fixed vs. Liquid Layout
- Responsive Webdesign
- Device Agnostic
- Mobile First

Grundstruktur

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Titel</title>
  </head>
  <body>

  </body>
</html>
```

Webseiten gestalten mit CSS

Einbindung

Selektoren

Box Model

Einheiten

Farben

Flexbox

Grid

Media Queries

Webanwendungen deployen und hosten

Webanwendungen organisieren und verwalten

Webseiten interaktiv machen mit JavaScript

Web-APIs verwenden

Webanwendungen testen

Webprotokolle verwenden

HTTP

Websockets

Webformate verwenden

Rastergrafiken

Vektorgrafiken

Single-Page-Applikationen implementieren

JavaScript auf der Serverseite verwenden

Webservices implementieren

REST

Daten in Datenbanken speichern

Webanwendungen absichern

Die Performance von Webanwendungen optimieren