BILDER IN HTML5

MMWP2024 - LV07

INHALTSVERZEICHNIS

INHALTSSCHWERPUNKTE

- Erklärung der Bedeutung von Bildern für Webseiten
- Vorstellung von modernen Bildformaten
- Asynchrones Laden von Inhalten
- Responsive Bilder

VORAUSSETZUNG

Der Ausgangspunkt dieser Vorlesungsreihe ist das Wissen über funktionsweise von BrowserAPIs

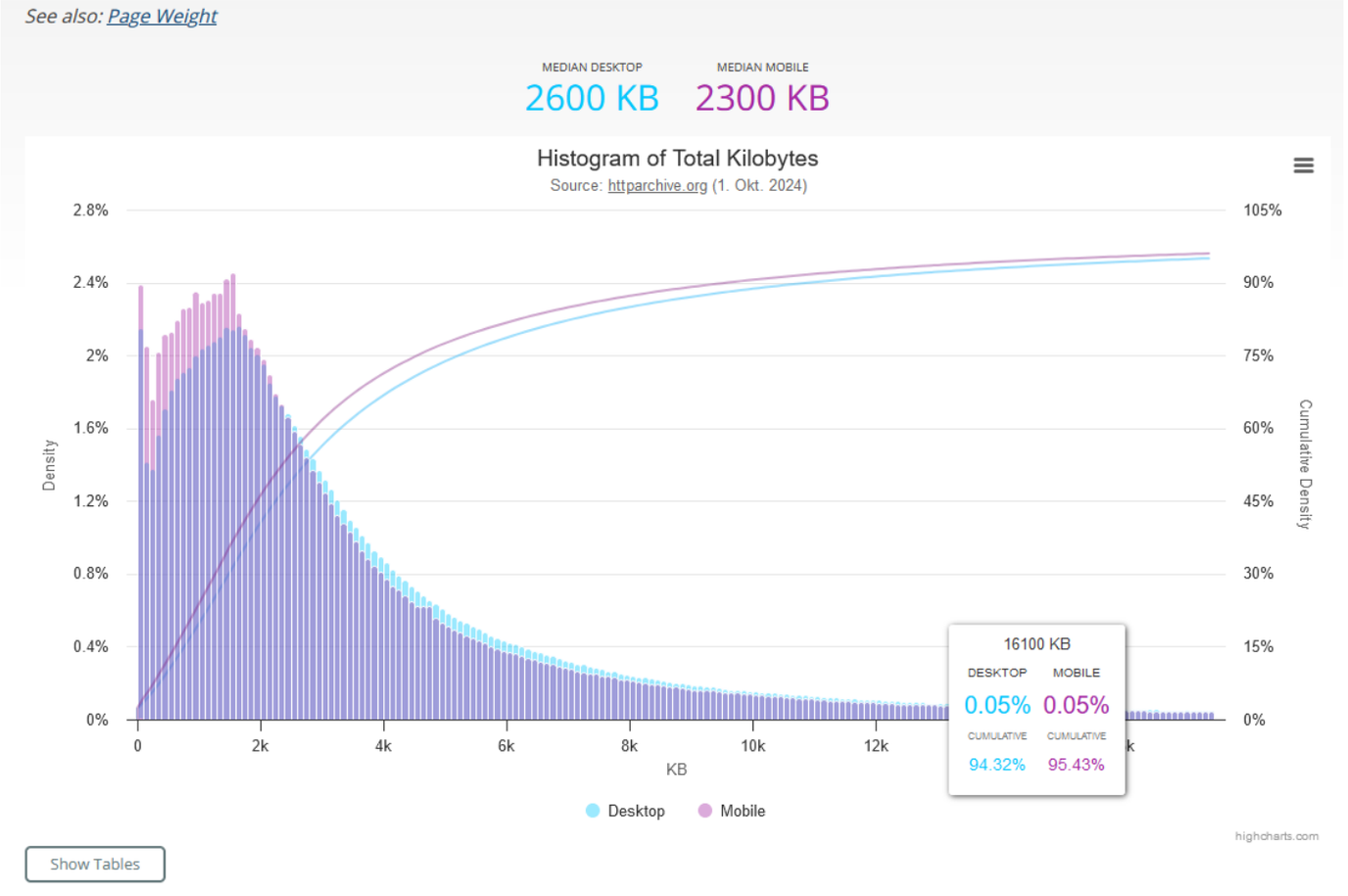
- Verständnis von Inhaltsplatzierung in HTML
- Responsive Webdesign
- Asynchrones Javascript zum Laden von Inhalten

ZIELE

Vorstellung von:

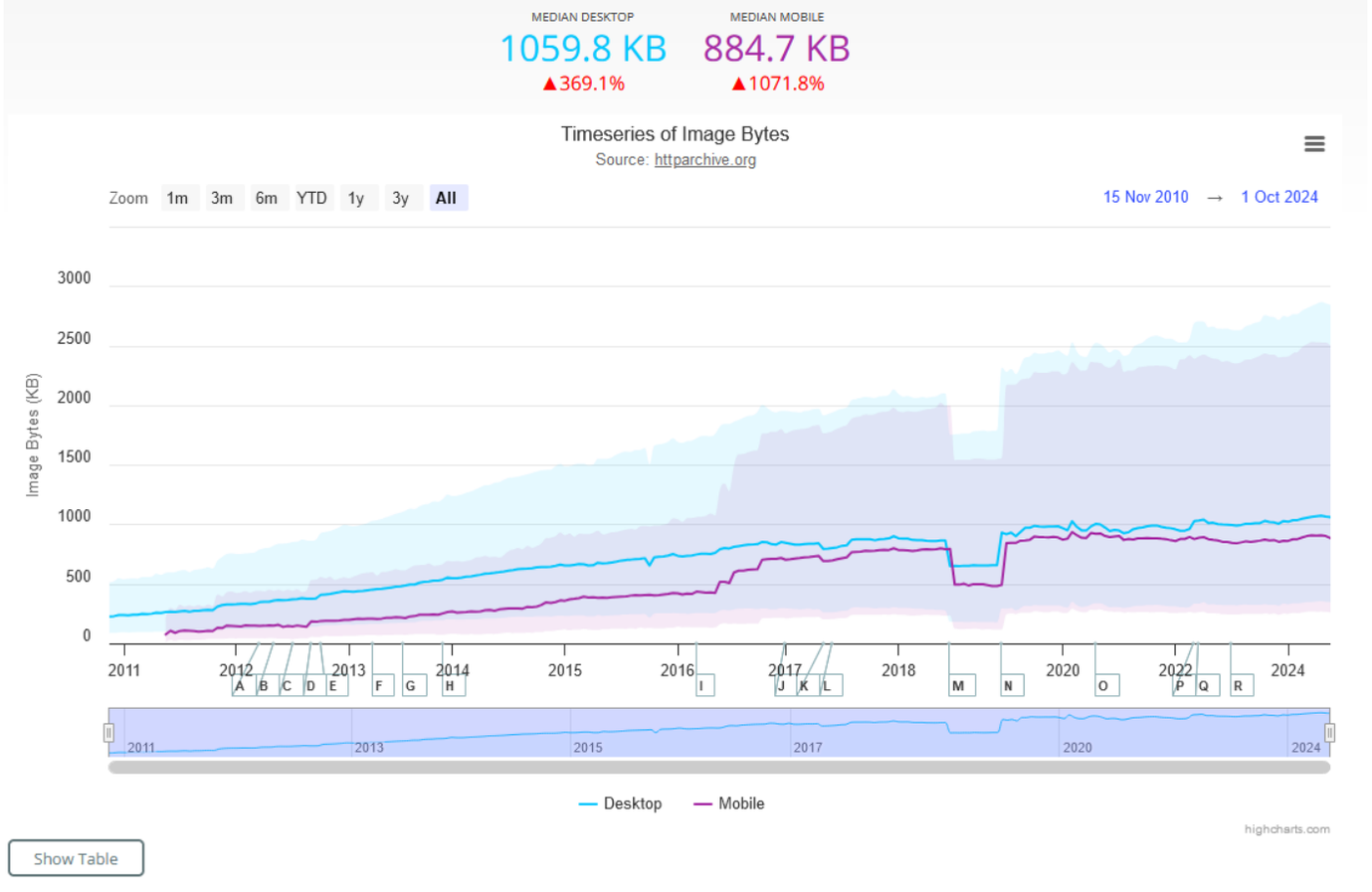
- Modernen Bilderformaten (APNG, AVIF, WebP)
- Beispiele von Inhaltsdarstellung
- Asynchrones Verhalten von Inhalten
- High Density Displays
- Bilder und Media Queries

SPEICHERGRÖSSEN VON WEBSEITEN



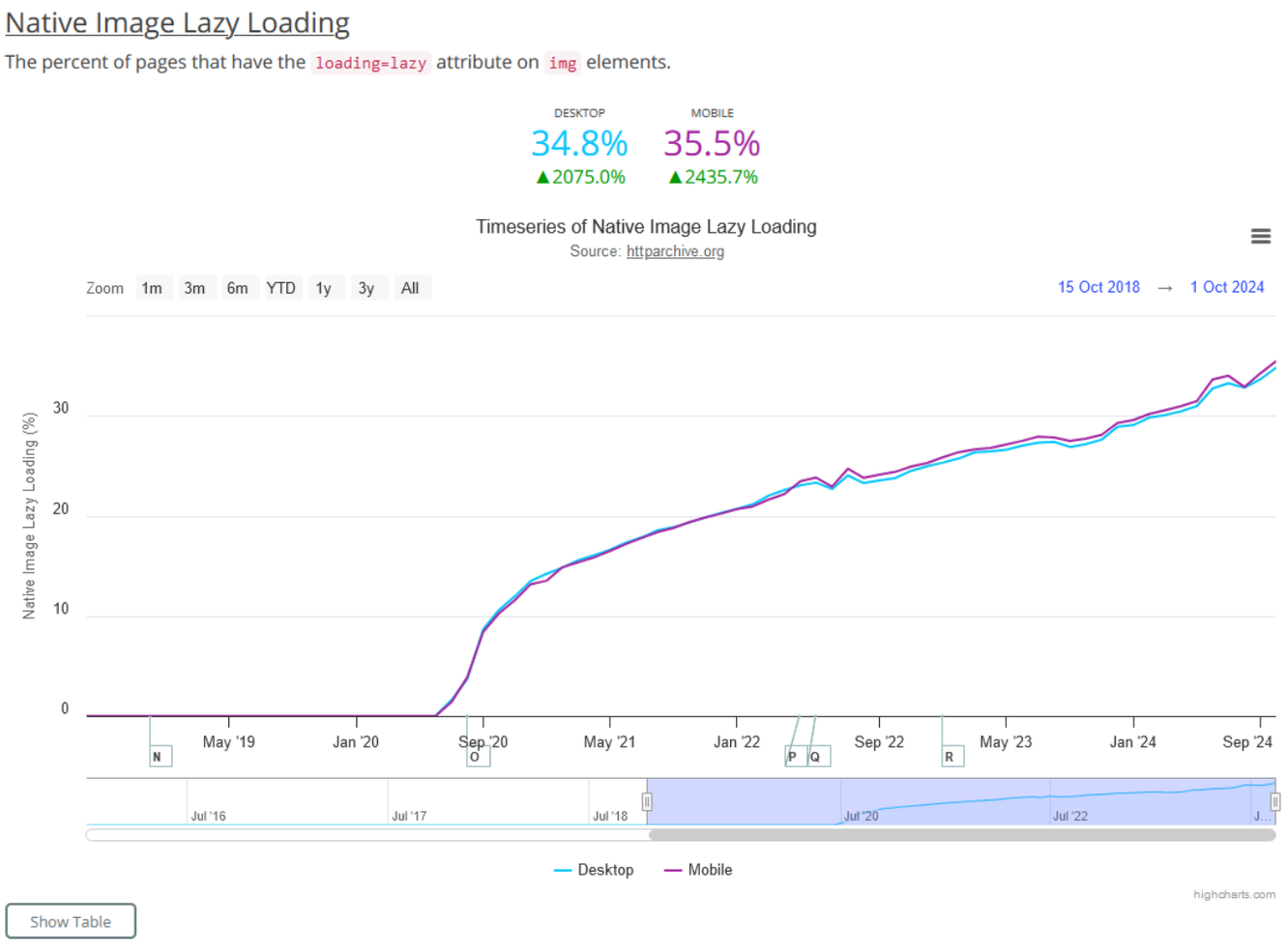
Quelle

SPEICHERGRÖSSEN VON BILDERN AUF WEBSEITEN



Quelle

ANTEIL VON ASYNCHRON GELADENEN BILDERN



Quelle

BILDER IN WEBSEITEN - 1

- Bilder im Kontext von Webseiten wurden bisher so betrachtet, dass die Pixelanzahl proportional zu den Abmaßen des Bildes bei der Darstellung auf einem gegebenen Bildschirm im Browserfenster ist
- Das Verhältnis zwischen Pixelanzahl und Abmaßen von Bildschirmen stimmt aber nicht mehr
- Es gibt zunehmend kleine Bildschirme hoher Auflösung (Siehe vorherige Vorlesungen)

BILDER IN WEBSEITEN - 2

- Standpunkt der Pixeldichte / Auflösung her die richtigen Bildversionen gleichen Datentyps den richtigen Bildschirmen zuzuordnen?
- Die Abmaße eines Bildes in Pixeln entspricht genau seinen Abmaßen in der Verarbeitung in CSS-Pixeln (dank viewport einstellung)
- Auch der Ansatz One-Size-Fits-All ist nicht korrekt (50MB 4K Bilder auf kleinen mobil Geräten bei 3G)

"HISTORISCHE" BILDFORMATE - 1

- JPEG
  - Familie von Standards verlustbehafteter Bildkompressionsalgorithmen
  - Umfangreiche Metadaten in EXIF-Header möglich
  - Unterstützt keine Transparenzen oder Animationen
- GIF
  - Verlustfreie Kompression
  - geringe Farbtiefe (8/16/32 Bit), Speicherung von Farbpaletten möglich (Kompression)
  - Transparenzen und Animationen möglich
  - Speicherung interlaced möglich

"HISTORISCHE" BILDFORMATE - 2

- PNG
  - Verlustfreie Kompression
  - blockbasierte Kodierung (gut zum asynchronen Laden)
  - Farbpaletten möglich
  - RGB-Kanäle und zusätzlicher Alpha-Kanal für Transparenzen
  - Animationen nicht möglich

WEITERE BILDFORMATE

- SVG
  - Vektorgrafikformat
  - Breite Browserunterstützung
  - Komprimiert oder im Quelltext (lesbares Format)
  - Einschluss von Bitmap-Grafiken z.B. JPEG, GIF, PNG möglich
  - Animationen nicht möglich
- WEBP
  - Breite Browserunterstützung erst seit kurzem
  - Verlustbehaftete und verlustfreie Kompression
  - Blockbasierte Kodierung
  - Entropiekodierung

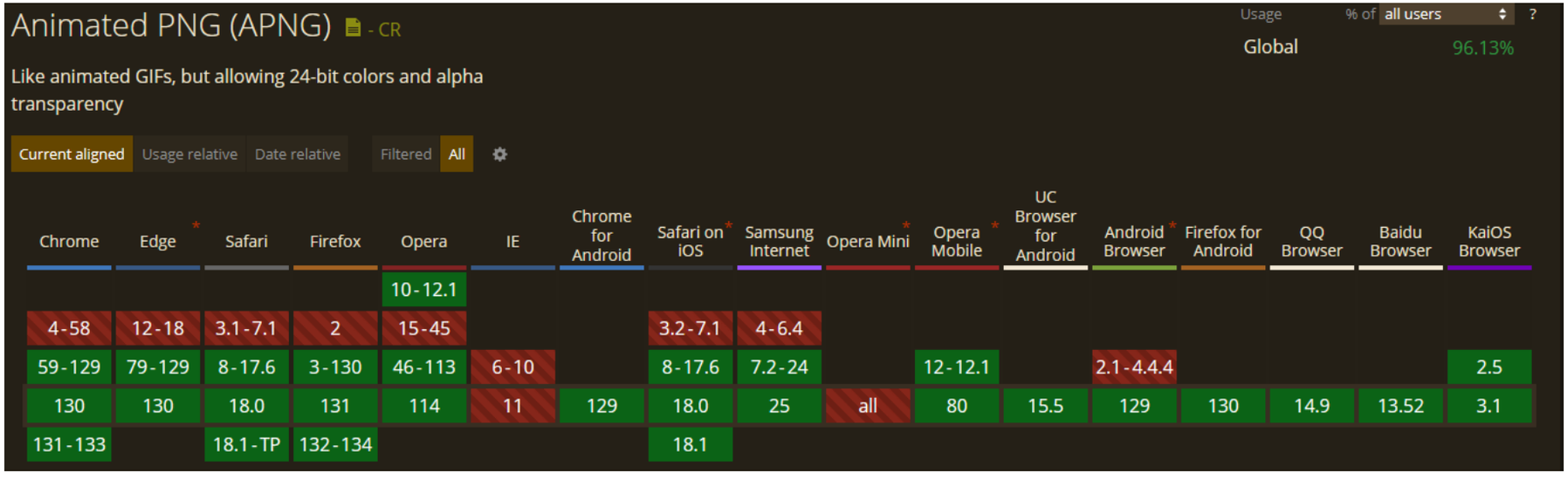
WEBP - ENTROPIECODIERUNG

- David A. Huffman hat 1952 einen Algorithmus zur Konstruktion von optimalen präfixfreien Codes angegeben
- Die häufigsten Elemente werden möglichst kurz codiert
- WebP setzt auf diese Technik, um kleinere Dateigrößen zu ermöglichen
- Statisch und nicht mehr einfach änderbar
- Mehr zu Entropiecodierung

"NEUE" BILDFORMATE

- APNG
  - APNG ist eine Erweiterung des PNG-Formats mit Unterstützung für animierte Bilder
  - Es ist als Ersatz für einfache animierte Bilder gedacht (Gif)
  - Unterstützt gleichzeitig 24-Bit-Bilder und 8-Bit-Transparenz
  - APNG ist rückwärtskompatibel mit PNG
  - Jeder PNG-Decoder kann die APNG-spezifischen Chunks ignorieren und ein einzelnes Bild anzeigen
  - Siehe Spezifikation
- AVIF (AV1 Image File Format)
  - Das AV1 Image File Format (AVIF) ist eine Codierung, die auf dem Open-Source-AV1-Videocodec basiert
  - Seit 2020 in Chromium unterstützt aber seit 2022 von Safari (Webkit) nicht mehr unterstützt
  - firefoxbasierte Browser haben im moment nur Teilsupport bis keinen (letzte Änderungen 2023)
  - Zeigt im Vergleich zu JPEG oder WebP deutliche Reduzierung der Dateigrößen (ohne Qualitätverluste oder Kompressionen)
  - Siehe AVIF BLog

APNG SUPPORT



Animated PNG (APNG) - CR

Like animated GIFs, but allowing 24-bit colors and alpha transparency

Usage % of all users ?

Global 96.13%

Current aligned | Usage relative | Date relative | Filtered | All

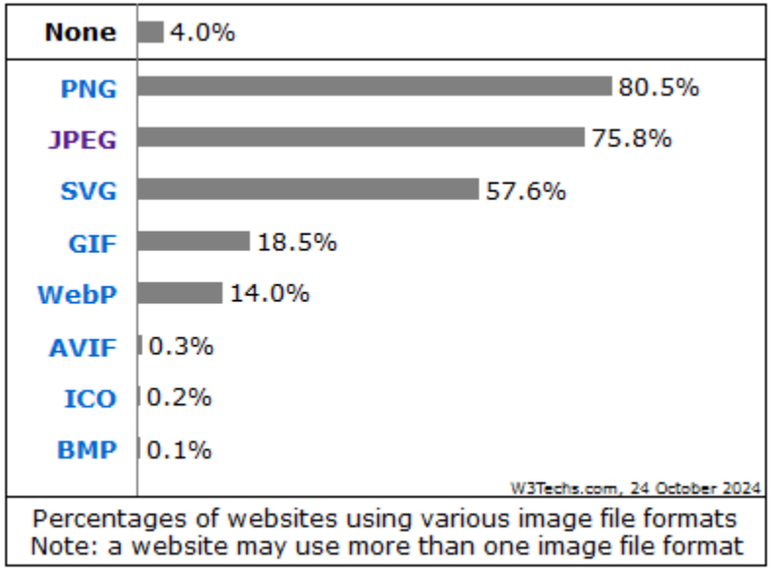
Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
				10-12.1												
4-58	12-18	3.1-7.1	2	15-45			3.2-7.1	4-6.4								
59-129	79-129	8-17.6	3-130	46-113	6-10		8-17.6	7.2-24		12-12.1		2.1-4.4.4				2.5
130	130	18.0	131	114	11	129	18.0	25	all	80	15.5	129	130	14.9	13.52	3.1
131-133		18.1-TP	132-134				18.1									

Quelle

AVIF SUPPORT

Quelle

ANTEILE DER BILDFORMATE AUF WEBSEITEN (2024)



Quelle

VERGLEICH DER ÄLTEREN BILDFORMATE

Eigenschaft	JPEG	PNG	GIF	WebP	SVG
fotorealistisch	+	+		+	
Computergrafik		+	+	+	+
Verlustbehaftet	+			+	
Verlustlos		+	+	+	+
Transparenz		+	+	+	+
Animation			+	+	+
Verlustfrei skalierbar					+
Browserunterstützung	+	+	+	+	+

Quelle

DAS BILDELEMENT <IMG> - 1

Attribut:	Funktion:	Verwendung:
src	Bindet ein Bild von einer Datei ein	**Pflichtattribut**
alt	Eine Bildbeschreibung als Text	**Pflichtattribut** (zur Barrierefreiheit)
crossorigin (*anonymous / use-credentials*)	Anonyme Nutzung oder mit Erlaubnis	Optional, bei Zugriff auf fremde Ressourcen
height	Höhe des Bildes	Optional, bei Wunschanpassung
sizes	Liste alternativ wählbarer Größen	Eine Bildgröße wird abhängig vom Bildschirm eingestellt
srcset	Liste alternativ wählbarer Bilddateien als Ressource	Eine Bilddatei wird abhängig vom Bildschirm geladen

Quelle

DAS BILDELEMENT <IMG> - 2

Attribut:	Funktion:	Verwendung:
title	Tooltip (angezeigt bei Mouse-Over)	Optional (dient zur Barrierefreiheit)
width	Legt die Breite des Bildes fest	Optional bei Wunschanpassung der Größe des Bildes
align	Ausrichtung	in CSS (links, rechts, mittig)
border	Rahmen	Nur in CSS verwenden
hspace	Abstand horizontal	Nur in CSS verwenden
vspace	Abstand vertikal	Nur in CSS verwenden

Quelle

BILDER IN HTML UND CSS

```
1 <img scr=''cat.jpg'' alt=''Katze''> <!-- als Bildelement im HTML-->
2 <body background=„hintergrund.jpg“> <!-- als Hintergrund im HTML -->
3 .main {background-image: url(''pattern.jpg'');} // als Hintergrund im CSS
4 .main:after {content: url(''pattern.jpg'');} // als Inhalt im CSS
5 .main:before {content: url(''pattern.jpg'');} // als Inhalt im CSS
```

BEFORE AND AFTER IM CSS

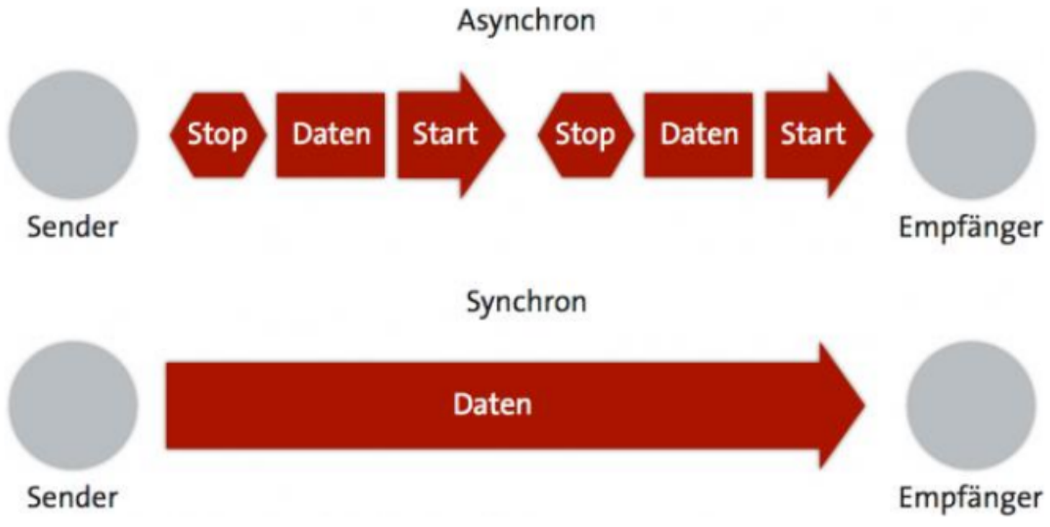
```
/* CSS2 Syntax */ element:after { Stileigenschaften }
/* CSS3 Syntax */ element::after { Stileigenschaften }

Die ::after Notation (mit zwei Doppelpunkten) wurde in CSS 3
eingeführt, um eine Unterscheidung zwischen Pseudoklassen
und Pseudoelementen einzuführen. Browser unterstützen auch
die :after Notation, wie sie in CSS 2 eingeführt wurde

selector : pseudo-class { property: value; }
selector :: pseudo-element { property: value; }
```

BILDER MIT JAVASCRIPT LADEN

```
function lazyload () {
  if(lazyloadThrottleTimeout) {
    clearTimeout(lazyloadThrottleTimeout);
  }

  lazyloadThrottleTimeout = setTimeout(function() {
    var scrollTop = window.pageYOffset;
    lazyloadImages.forEach(function(img) {
      if(img.offsetTop < (window.innerHeight + scrollTop)) {
      img.src = img.dataset.src;
      img.classList.remove('lazy');
    }
  });

  if(lazyloadImages.length == 0) {
```

Siehe CSS-Tricks

BILDER ASYNCHRON LADEN - IDEE



Quelle

AJAX (ASYNCHRONOUS JAVASCRIPT AND XML)

```
1  var xmlHttp = false; //globale Instanz von XMLHttpRequest
2  //XMLHttpRequest-Instanz erstellen
3  try {
4    xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
5  } catch(e) {
6    try {
7      //... für Internet Explorer
8      xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
9    } catch(e) {
10     xmlHttp = false;
11   }
12 }
13
14 //... für Mozilla, Opera, Safari usw.
15 if (!xmlHttp && typeof XMLHttpRequest != 'undefined') {
```

Siehe CSS-Tricks

JQUERY ASYNCHRONOUS IMAGE LOADER (JAIL)

```
<img class="lazy" src="/img/blank.gif" data-src="/img/image1.jpg"/>
<noscript>
<img src="/img/image1.jpg"/>
<noscript>
<script src="/lib/jquery.js"></script>
<script src="/src/jail.js"></script>
<script>
$(function(){
  $('img.lazy').jail();
});
</script>
```

Siehe Jail Beispiel

VORTEILE VON ASYNCHRONEN LADEN

- Entkoppelung von Darstellung relevanter Inhalte und der vollständigen Anzeige und Nutzbarkeit der Website
- Der Nutzer erhält schneller ein Ergebnis
- Die Website reagiert agiler, weil früher mehr zu sehen ist und genutzt werden kann
- Außerdem lässt sich steuern, wann der Inhalt anzeigt wird (z.B. über javascript)
- Dieser lässt sich optisch ansprechend darstellen, was den Joy-of-Use erhöhen kann

NACHTEILE VON ASYNCHRONEN LADEN

- Die Voraussetzung, dass JavaScript zur Verfügung stehen muss (Noscript)
- In der Wirkung kann die Seite allgemein als „unruhig“ empfunden werden (Bewegungen)
- Durch zu viele AJAX-Requests kann Ladezeit auch negativ beeinflusst werden
- Caching-management, damit Inhalte nicht mehrfach geladen werden (z. B. nach einem Seitenwechsel)
- Siehe Blogbeitrag zur Ladezeitenoptimierung

MASSNAHMEN ZUR OPTIMIERUNG

Nach Testergebnissen von E. Schlag sind folgende Optimierungen effektiv:

Maßnahme	Ladezeitreduktion [%]	Datenreduktion [%]
Caching (client- und serverseitig)	42-82	99-100
Infinite Scroll	40-62	63-80
Lazy Loading	36-58	
W-Descriptor	32	23-53
WebP über picture	19-26	
Inline-SVG		9

User-Agent-Sniffing/srcset allein: ohne Wirkung

LIVE-BEISPIEL

- Unoptimiert
- Asynchron
- Mehrfachsupport
- Zugeschnittene Bilder

VORTEILE VON NEUEN DATENTYPEN

Beispiel und technische Erklärungen

HIGH DENSITY PIXEL

Die Pixeldichte von Bildschirmen

$$Pixeldichte = \frac{\sqrt{(Breite\ in\ Pixel)^2 + (Höhe\ in\ Pixel)^2}}{Bildschirmdiagonale\ in\ Zoll}$$

- Pixel per Inch (PPI) ist das Analogon zur Größe DPI in der Druckereiindustrie
- ab 240 PPI spricht man von High Density
- ab 287 PPI soll ein durchschnittlicher Mensch die Einzelpixel im Bild nicht mehr sehen können (modulo Sehschärfe und Geräteabstand)

HIGH DENSITY DISPLAYS

JPEG (unkomprimiert) 71,0 KB JPEG (50% komprimiert) 11,9 KB Ausschnitt-Vergrößerung

Quelle

HIGH DENSITY DISPLAYS - MASSEINHEITEN

- Um Bilder nicht sinnlos klein dargestellt zu bekommen, wurden verschiedene PixelMaße eingeführt
- Physische Pixel: Tatsächliche Ausmaße eines Objektes auf einem Display in Pixeln (Breite mal Höhe)
- Logische Pixel: Konstanter, für die interne Kalkulation verwendeter Wert, auch dips (device-independent pixels) oder CSS-Pixel genannt
- Bei unterschiedlichen Zoom-Leveln einer Webseite werden immer die CSS-Pixel-Werte angegeben

VIEWPORT

- Für Desktops ist bzw. war der sichtbare Bereich im Browserfenster gleich dem Bereich für Layout-Berechnungen
- Bei Smartphones, Tablets etc. wurde dies direkt eingeführt
- Visual Viewport: sichtbarer Bereich
- Layout Viewport: Bereich für Layout-Berechnungen (auch teilweise unsichtbar)
- Standard ist somit, den Layout Viewport zu rendern und das Ergebnis in den Visual Viewport hineinzuzoomen (meist zu verkleinern)

HIGH DENSITY DISPLAYS - SMARTHPONES

- Im Ergebnis sieht man die gesamte Webseite auf dem Display des Smartphones, Tablets etc.: Gerätedisplaybreite ist gleich Webseitenbreite.
- Um vor dem Rendern den Standard Layout Viewport zu überschreiben und den Visual Viewport mit dem Layout Viewport gleichzusetzen, gibt es zwei alternative Anweisungen:

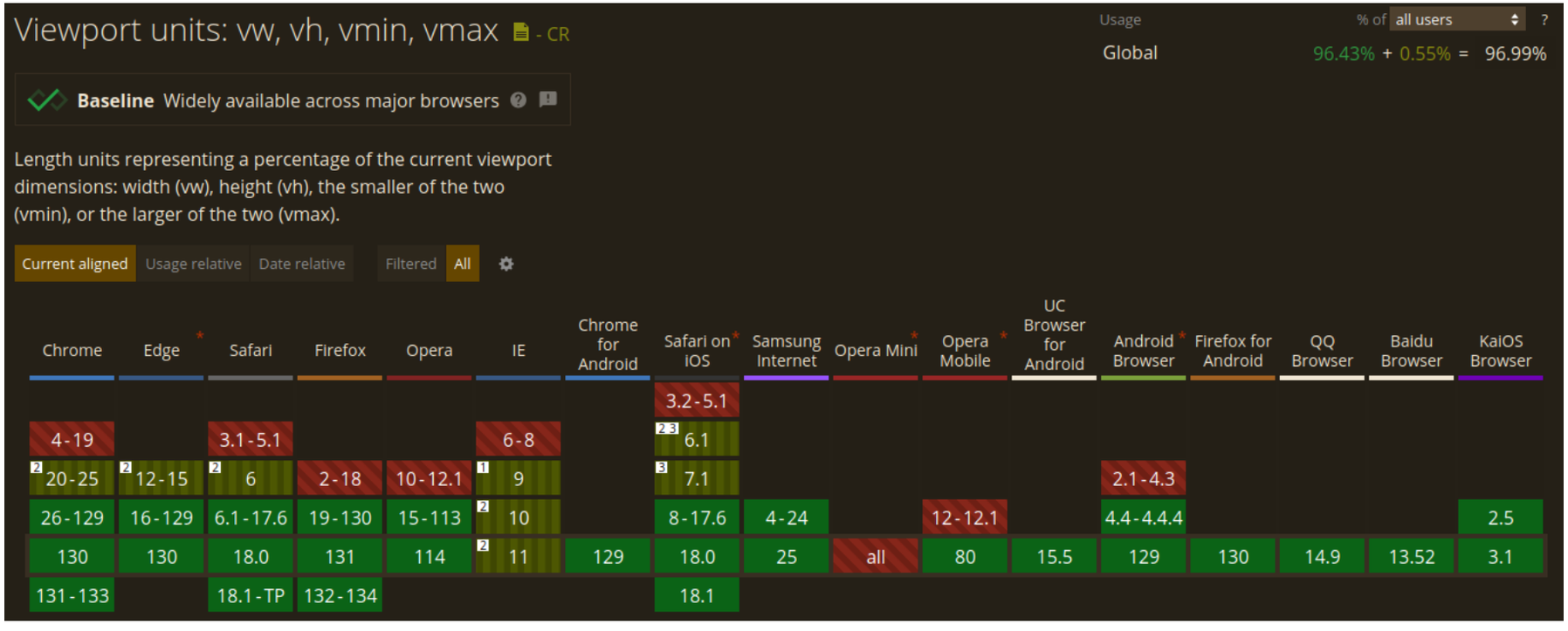
```
HTML
<meta name="viewport" content="width=device-width, initial-scale=1">

CSS
@viewport {width : device-width; zoom : 100%;}
```

VIEWPORT METHODEN UND PARAMETER

Eigenschaft	CSS-Deklaration		meta-Tag	
	Schlüssel	Werte	Schlüssel	Werte
Breite	width	auto, Länge, Prozentsatz	width	device-width, Zahl
Start-Zoom	zoom	auto, Zahl, Prozentsatz	initial-scale	Zahl
Zoom-Veränderung	user-zoom	fixed, zoom	user-scale	yes, no
Ausrichtung	orientation	auto, portrait, landscape	-	-

Quelle

VIEWPORT SUPPORT



Quelle

MIT VIEWPORT TAG UND OHNE VOREINSTELLUNG

Beispiel

TESTSEITE VON PETER-PAUL KOCH

- Eine sehr gute Testseite für Browser gibt es unter
- Grid und responsive Testseite
- Dort kann man über zwei Links die Weite des Viewports setzen bzw. im Standardmodus die natürliche Weite (DPR) des Viewports abfragen

HISTORISCHE DATEN ÜBER VIEWPORT

- Das meta-Element viewport wurde von Apples iPhone-Safari-Browser etwa 2008 eingeführt
- „CSS Device Adaption Module Level 1“ – Editor‘ Draft, 05. 07. 2018 (siehe Draft Approach)
- Vernünftiges Aussehen erzielt man mit responsivem Design und seinen Methoden

DEVICE PIXEL RATIO (DPR)

- Displays unter 200 DPI haben DPR=1
- Displays zwischen 200 und 300 DPI haben DPR=2
- Displays mit mehr als 300 PPI: DPI = (PPI /150)abgerundet
- Neuere Algorithmen zur Bestimmung der Darstellungsgröße von Bildern bauen auf der Größe DPR auf

DEVICE PIXEL RATIO (DPR) - BEISPIELE - 1

- Beispiel zur Nutzung eines bestimmten Bildes bei mindestens DPR=2 bzw. mindestens DPI=192:
- So kann vor dem Hochladen eines Bildes entschieden werden, welches Bild es sein soll (96dpi = 1dppx, Fallback)
- Resolution Dokumentation

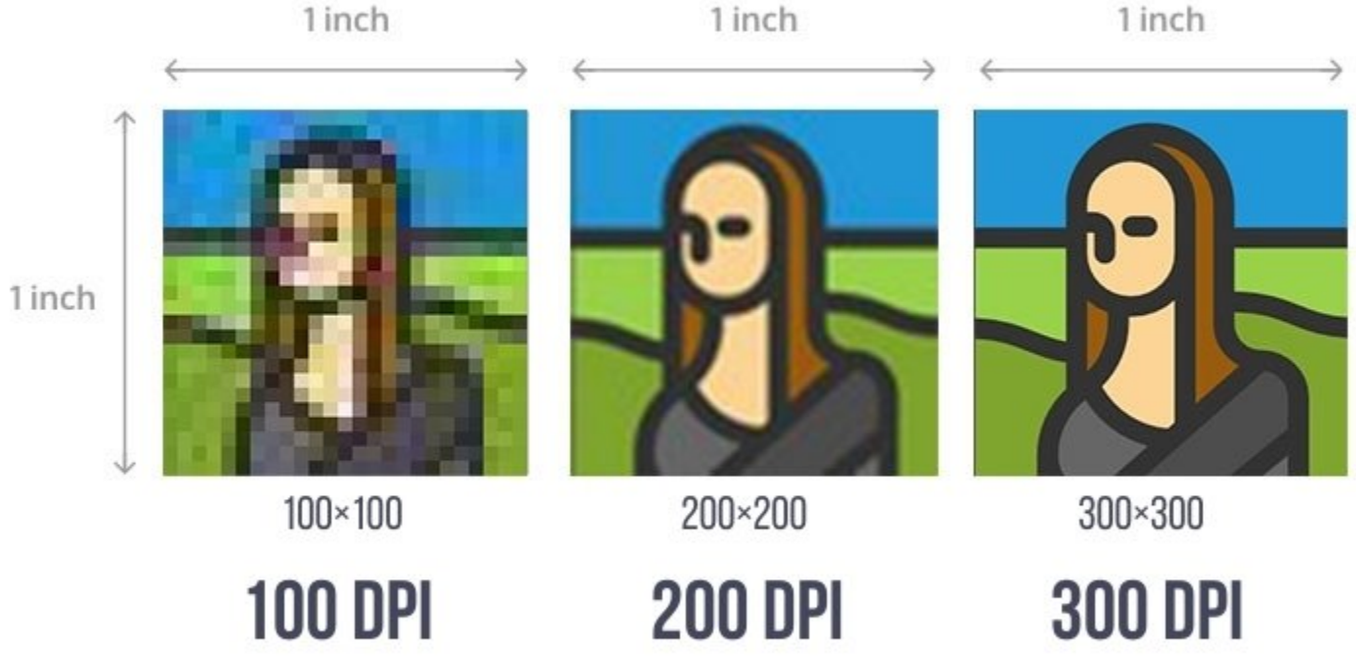
```
@media only screen and (-webkit-min-device-pixel-ratio: 2),
  only screen and (min-resolution: 2dppx),
  only screen and (min-resolution: 192dpi),
  .btn {background-image: url(image@2x.jpg);
}
```

DEVICE PIXEL RATIO (DPR) - BEISPIEL - 2

- Beispiel zur Nutzung eines bestimmten Bildes bei mindestens DPR=2 bzw. mindestens DPI=192:
- So kann vor dem Hochladen eines Bildes entschieden werden, welches Bild es sein soll (96dpi = 1dppx, Fallback)

```
<figure>
  <picture>
    <source media="(min-resolution: 4dppx)" srcset="image@4x.jpg">
    <source media="(min-resolution: 2dppx)" srcset="image@2x.jpg">
    <img src="small.jpg" alt="The president giving an award.">
  </picture>
  <figcaption>Beispiel</figcaption>
</figure>
```

BEISPIEL



Quelle

DEVICE PIXEL RATIO (DPR) - BEISPIEL - 2

- HTML Living Standard HTML-Picture standard(Updated 23 October 2024)
- zur Unterstützung älterer Browser gibt es z.B. Picturefill (Responsive Image Polyfill) (siehe Picturefill)
- Polyfill bzw. Emulation von <picture>-Element (Es stellt Bilder je nach Bildschirmgröße, Viewport-Größe und Auflösung angemessen dar)

BILDER ZUR WAHL JE NACH DPR - 1

Das WHATWG hat folgende Syntax vorgeschlagen, um minimalistisch eine Auswahllösung zu schaffen:

```
<img src="/uploads/100-marie-lloyd.jpg"
   srcset="/uploads/150-marie-lloyd.jpg 1.5x,/uploads/200-marie-lloyd.jpg 2x
   alt="" width="100" height="150">
```

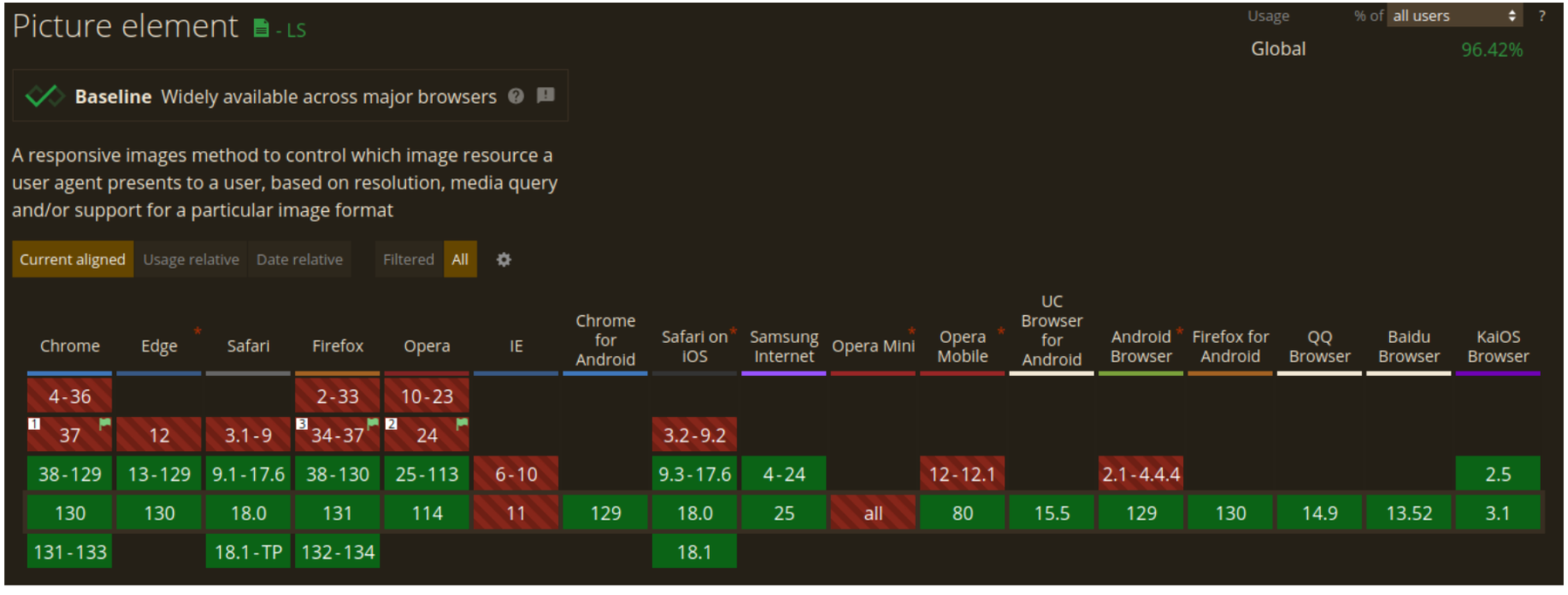
BILDER ZUR WAHL JE NACH DPR - 2

```
<div class="container">
  <img src="https://www.mediaevent.de/xhtml/img/needles-1440.jpg"
     srcset="https://www.mediaevent.de/xhtml/img/needles-780.jpg 780w,
  https://www.mediaevent.de/xhtml/img/needles-1024.jpg 1024w,
  https://www.mediaevent.de/xhtml/img/needles-1440.jpg 1440w,
  https://www.mediaevent.de/xhtml/img/needles-2400.jpg 2400w"
     sizes = "(min-width: 1600px) 1200px, // ViewPort mindestens 1600 px
  (min-width: 1400px) 1100px, // ViewPort mindestens 1400 px, nimm Bild
  (min-width:1000px) 900px, // ViewPort mindestens 1000 px, nimm Bild mi
  100vw"
  // kleinerer ViewPort: volle Breite des ViewPorts (100vw)
  width="780" height="521"
  alt="HTML img srcset – Bilder je nach Breite des Viewports laden">
</div>
```

BILDER ZUR WAHL JE NACH DPR - 3

```
<img src="one.png"
    srcset="two.png 100w, three.png 500w, four.png 1000w"
    sizes="(min-width: 900px) 1000px,
(max-width: 900px) and (min-width: 400px) 50em,
( not (orientation: portrait) ) 300px,
( (orientation: landscape) or (min-width: 1000px) ) 50vw,
100vw">
```

PICTURE - SUPPORT



Quelle

RESPONSIVE IMAGES

- Es wurde eine Lösung gesucht, die allein für ein einziges img-Element schon funktioniert und wesentlich flexibler reagiert
- HTML Living Standard: Embedded content standard(Updated 23 October 2024)
- Picture-Element als Container mit:
  - source-Element zum Anbieten verschiedener Bilder
  - img-Element als Fallback zur Darstellung eines Default-Bildes

VORSTELLUNGEN

Variable	Dem Entwickler beim schreiben des Codes bekannt?	Dem Browser während des ladens der Seite bekannt?
Viewport-Dimensionen	Nein	Ja
Bildgröße im Verhältnis zum Viewport	Ja	Nein (Abhilfe: `sizes`)
Bildschirmauflösung	Nein	Ja
Dimensionen der Bilddateien	Ja	Nein (Abhilfe: `srcset`)

Quelle

RESPONSIVE BILDER BEISPIELE - 1

```
<img sizes="100vw"
  srcset="wolf-400.jpg 400w, wolf-800.jpg 800w, wolf-1600.jpg
  1600w"
  src="wolf-400.jpg"
  alt="The red wolf">
```

Einheit	Beschreibung
vw: Viewport-Breite	prozentuale Breite des Anzeigebereichs (Viewport): 100vw = Breite des Viewports
vh: Viewport-Höhe	prozentuale Höhe des Anzeigebereiches: 100vh = Höhe des Viewports
w	maximale Breite in CSS-Pixeln, bis zu der das entsprechende Bild zu laden ist

RESPONSIVE BILDER BEISPIELE - 1

- Wenn der Bildschirm des Nutzers 320 CSS-Pixel groß ist, dann müsste die Spezifikation lauten: wolf-400.jpg 1.25x, wolf-800.jpg 2.5x, wolf-1600.jpg 5x
- Wenn der Bildschirm des Nutzers 1200 CSS-Pixel groß ist, dann müsste die Spezifikation lauten: wolf-400.jpg 0.33x, wolf-800.jpg 0.67x, wolf-1600.jpg 1.33x
- Der Browser des Nutzers kann die korrekte Wahl des zu ladenden Bildes einleiten

RESPONSIVE BILDER BEISPIELE - 2

- Mit Responsive Design werden drei Layouts definiert:
- Schmaler Viewport: eine Spalte Bilder
- Mittlerer Viewport: zwei Spalten Bilder
- Breiter Viewport: drei Spalten Bilder
- Beispielsweise Breakpoints: 30em und 50em

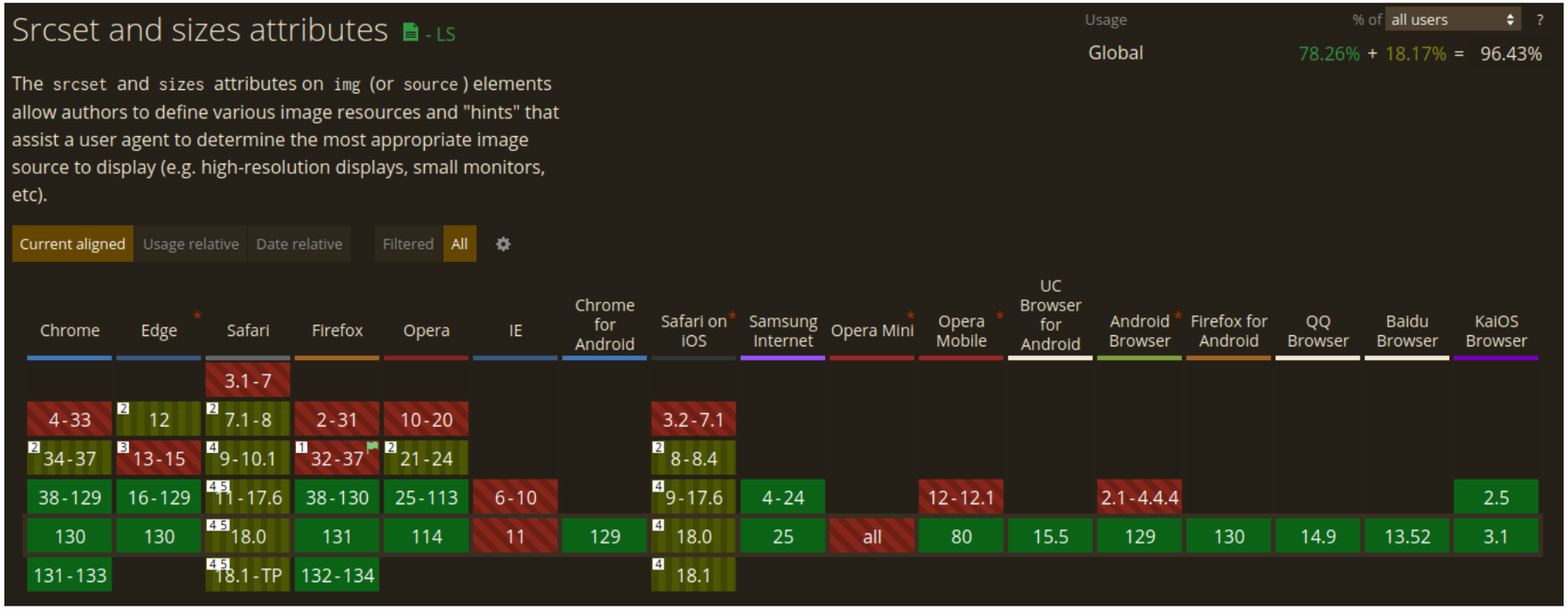
RESPONSIVE BILDER BEISPIELE - 2

```
  <img sizes="(max-width: 30em) 100vw, (max-width: 50em) 50vw, calc(33vw – 1
     srcset="swing-200.jpg 200w, swing-400.jpg 400w, swing-800.jpg 800w,
swing-1600.jpg 1600w"
     src="swing-400.jpg"
     alt="Kettlebell Swing">
```

RESPONSIVE BILDER BEISPIELE - 2

- Ist die Viewport-Breite 29em, dann wird die Bedingung (max-width: 30em) erfüllt und 100vw wird benutzt
- Das Bildmaß für die Wahl der Bilddatei ist dann 29em
- Wenn die Viewport-Breite 32em ist, dann wird die Bedingung (max-width: 30em) nicht erfüllt, jedoch die Bedingung (max-width: 50em)
- Also: der Wert 50vw wird benutzt und das Ergebnis ist: die Bilder werden schmaler

SRCSET - SUPPORT



Quelle

- Größtes Webbild im Internet
- Ladeverhalten von Bildformaten

WEITERFÜHRENDE INFORMATIONEN

- HTML Responsive Images Guide (12.11.2024)
- Mozilla - Responsive images (12.11.2024)
- Tutorial auf Kulturbanause

QUELLEN

- Framework „Adaptive Images“ (minimalistisch, effektiv): Quelle
- weitere Tipps und Links zu Bibliotheken bzw. Frameworks: Quelle

ABSPANN

Siebtes Level geschafft weitere Folgen!

Fragen und Feedback?