

Die neuen CSS-Queries @layer, @scope und @container

vienn
eva
.com



Contao²⁰²⁴
KONFERENZ

ÜBER MICH

- Biologiestudium, Erlangen
- vienneva.com, Wien, seit 2002
- Contao seit 2.9.4
- hands-on, learning by doing
- HTML, CSS, (JS, PHP)
- Web Accessibility
- ein erwachsener Sohn
- Jazz Sängerin

Dr. Eva Broermann



@layer, @scope und @container

01. Worum geht's?

02. @layer

03. @container

04. @scope

05. Fazit

- Was tut diese Query?
- Wofür setze ich das ein?
- Can I (already) use (it)?
- Vorteile? Nachteile?

01.

Worum geht's?

@layer	@container	@scope
Schichtung von Stylesheets	Styles in Abhängigkeit von Containergröße	Styles in definierten Bereichen (Nähe)
W3C-Candidate Recommendation	W3C- Working Draft	W3C- Working Draft
93.59% Abdeckung, seit 2022 in den großen Browsern	90.97% Abdeckung, seit 2023 in den großen Browsern	82.53% Abdeckung, 2024 eingeschränkt in großen Browsern
	va. inline size	noch nicht in FF

01. Worum geht's? - Die CSS Kaskade

Reihenfolge der Stylesheets

Browser (User agent)

Benutzer, lokal im Browser (User)

Webdesigner:in (Author)

Reihenfolge der Style-Regeln

```
p { color:red; }
```

```
p { color:green; }
```

Spezifität des Selektors

#id	.klasse	selektor
x100	x10	x1

```
p.feuer { color:red; }
```

```
p { color:green; }
```

02. @layer

- Schichtung von Styles
- Erste Nennung der Layer bestimmt Reihenfolge
- Der spätere Layer gewinnt
- **Unabhängig von der Spezifität!**
- **Unlayered** hat immer **Vorrang**

<https://codepen.io/Eva-Broermann/pen/QWYeVeM>

```
@layer reset, basic, mystyle;

@import url(https://vienna.com/files/layout/css/style.css)
layer(mystyle);

@layer reset {
  h1, h2, h3, h4, p, ul, input{margin:0;}

  @layer colors {
    .green {color:green;}
  }
}

@layer basic{
  h1 { border:0 0 0 2px solid black;}
}

body{
  background-color:#fefefe;
}


@layer mystyle.colors {
}
```


02. @layer - wofür ist es gut?


“@layer ist ein CSS-light”

- > Arbeit mit Komponenten, Frameworks, Einbindung von Dritt-CSS
- > leichtes Ändern der Reihenfolge
- > leichtes Überschreiben (hebelt Spezifität aus)
- > kürzere Selektoren, weniger !important;

02. @layer - can I use?

CSS at-rule: @layer  Usage % of all users Global 92.46%

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 * *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
4-98	12-98	3.1-15.3	2-96	10-84			3.2-15.3	4-17.0								
99-128	99-128	15.4-17.6	97-129	85-113	6-10		15.4-17.6	18.0-24		12-12.1		2.1-4.4.4				2.5
129	129	18.0	130	114	11	129	18.0	25	all	80	15.5	129	130	14.9	13.52	3.1
130-132		18.1-TP	131-133				18.1									

Unterstützt der Browser @layer nicht, werden gelayerte Styles ignoriert und nur ungelayerete Styles angewendet.

02. @layer - Links

css-tricks.com/css-cascade-layers/

blog.openreplay.com/css-cascade-layers/

kulturbanause.de/blog/css-cascade-layers-kaskadenebenen-statt-spezifitaet/

www.matuzo.at/blog/2022/100daysof-day68/

03. @container

- Größenabfrage des Containers
- ähnlich @media
- vorrangig Breite
- Styles nur für Elemente im Container
- width innerhalb Query = content box!
- Container units cqw, cqh, cqi, cqb

<https://codepen.io/Eva-Broermann/pen/mdNVaVj>

```
.parent {
  container-type: inline-size;
  container-name: parent;
}
.parent {
  container: parent / inline-size;
}
.child {
  border: 8px solid black;
}

@container parent (max-width: 480px) {
  .child {
    border-width: 4px;
  }
}
```

03. @container - wofür ist es gut?

@container stylt nach Breite des Containers

- Styling unabhängig vom Viewport
- Skalierung über Containergröße
- Einheitliches Aussehen
- abhängig vom vorhandenen Platz
 - z.B. wenn linke oder rechte Spalte vorhanden

03. @container - can I use?



03. @container - can I use?

CSS at-rule: @container: Style queries for custom properties

Usage Global % of all tracked 73.15%

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
4-110	12-110			10-96				4-21								
111-128	111-128	3.1-17.6	2-129	97-113	6-10		3.2-17.6	22-24		12-12.1		2.1-4.4.4				2.5
129	129	18.0	130	114	11	129	18.0	25	all	80	15.5	129	130	14.9	13.52	3.1
130-132		18.1-TP	131-133				18.1									

03. @container - Links

www.youtube.com/watch?v=resGqSXCxxA

Sven Wolfermann, Contao Konferenz 2023

developer.mozilla.org/en-US/blog/getting-started-with-css-container-queries

kulturbanause.de/blog/css-container-queries-mit-container/

css-tricks.com/css-container-queries/

04. @scope

<https://codepen.io/Eva-Broermann/pen/GRVZXLe>

- Definiert einen Geltungsbereich
- Gewichtet die Nähe zum Scope-Selektor (Zugehörigkeit)
- Kann auf sich selbst Bezug nehmen
- Begrenzung der Tiefe möglich

```
@scope (#main) {  
  a {  
    text-decoration: underline;  
  }  
  
  & {  
    background: white;  
  }  
  
  :scope {  
    background: magenta;  
  }  
}  
  
@scope (.mod_article) to (.news_full) {  
}
```

04. @scope - wofür ist es gut?

- Styles auf einen bestimmten Unterbereich einschränken
 - = Styles an einen Bereich binden
- gewichtet räumliche Nähe
- leichter Varianten bilden
- funktioniert auch inline (bezieht Scope aus Elternelement)
- funktioniert von – bis
 - z.B Stylen eines Hüllelements aber nicht des Contents

04. @scope - can I use?

Scoped Styles: the @scope rule - WD

Usage % of all users
Global 82.53%

Limited availability across major browsers

Allows CSS rules to be scoped to part of the document, with upper and lower limits described by selectors.

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
4-103	12-103			10-89												
104-117	104-117	3.1-17.3		90-105			3.2-17.3									
118-128	118-128	17.4-17.6	2-129	106-113	6-10		17.4-17.6	4-24		12-12.1		2.1-4.4.4				2.5
129	129	18.0	130	114	11	129	18.0	25	all	80	15.5	129	130	14.9	13.52	3.1
130-132		18.1-TP	131-133				18.1									

04. @scope - Links

12daysofweb.dev/2023/css-scope/

www.youtube.com/watch?v=YPmUlwVh5vk

developer.chrome.com/docs/css-ui/at-scope?hl=de

05.

Fazit

@layer	@container	@scope
Schichtet Styleregeln und Stylesheets	Styling abhängig von Containergröße	Begrenzt den Gültigkeitsbereich von Styleregeln
<ul style="list-style-type: none">• zentrale Deklaration• einfaches Umschichten• Leicht-Gewichtung von Styleregeln• hebt Selektor-Spezifität aus	<ul style="list-style-type: none">• spart @media-Queries• bezogen auf Größe des Containers• einheitliche Darstellung abhängig vom vorhandenen Platz• Container selbst nicht ansprechbar	<ul style="list-style-type: none">• gewichtet Nähe• gut für Varianten• spart Selektorketten• Bereich: innerhalb, oder von-bis• scope selbst kann angesprochen werden
<ul style="list-style-type: none">• layers überschreiben sich• gewöhnungbedürftig	<ul style="list-style-type: none">• derzeit nur size query• style queries noch experimentell	<ul style="list-style-type: none">• noch nicht in FF

05.

	Vorteile	Nachteile
@layer	Hilfreich beim Einbinden von Frameworks, Komponenten. Erspart lange Selektorketten und !important.	Gewöhnungsbedürftig, da unabhängig von Spezifität
@container	Spart @media Queries. Berücksichtigt vorhandenen Platz. Direktere Definition. Einheitlichkeit	
@scope	Definiert den Geltungsbereich. Berücksichtigt räumliche Nähe. Erspart lange Selektorketten. Gut für Varianten.	Noch nicht in Firefox

Danke für's Zuhören!

vienn
eva
.com



Contao²⁰²⁴
KONFERENZ