

# Klimafreundliche Websites: Die Checklist für Developer:innen

Von °Cleaner Web

Die folgende Liste könnt ihr direkt in eurem PDF-Reader abhaken, sobald ihr die Änderungen und Prüfungen auf eurer Website erledigt habt. Es gibt keinen Grund, sie auszudrucken – lieber klimaschonender am Rechner bearbeiten und mit euren Kolleg:innen teilen.

Wer mehr Informationen zu dieser Liste erhalten möchte, kann auf unserer Website mehr über die einzelnen Punkte erfahren:

[cleaner-web.com/kriterien-klimabewusste-website](https://cleaner-web.com/kriterien-klimabewusste-website)

Auch freuen wir uns immer über Feedback und konstruktive Kritik!

Meldet euch unter:

[kontakt@cleaner-web.com](mailto:kontakt@cleaner-web.com)



## Bilder, Videos, Animationen

### Autoplay von Videos

	Autoplay-Videos entfernen oder ersetzen
	Videos im Hintergrund durch Grafiken oder einfach zu berechnende Animationen ersetzen

### Lazy Load für Bilder

	Lazy Load über sauberen Code oder zur Not via Plugin einsetzen
--	--

### Verwendung moderner Bildformate wie WebP oder AVIF

	Möglichkeiten von Server, Website oder Plugin ausnutzen, um WebP oder AVIF auszuliefern
--	---

### Responsive Bildgrößen

	Bilder responsive einbinden
--	-----------------------------

### Webgerechte Kodierung der verwendeten Bilder

	JPG-Bilder vor dem Hochladen optimieren mit Qualitätslevel 60-70
	Einsatz moderner Bildformate (siehe oben)
	Bei sehr umfangreichen Seiten oder Seiten mit Zumindest 60.000 Besucher:innen pro Monat: Einsatz eines CDN mit integrierter Bildoptimierung
	sinnvolle Begrenzung von Bildgrößen (Dimensionen) und der



	Anzahl von Bildern
--	--------------------

## Effiziente Animationen

	Aufwändige Animationen entfernen
	zur Not Video der Animation erstellen & einsetzen

## Skripte

### Gesamtgröße aller geladener Skripte

	Analyse: Welche Skripte sind notwendig und welche Funktionen können anders verwirklicht werden?
	Wenn zu viele Skripte notwendig sind für einfache Funktionen: System wechseln.
	Notwendige Skripte erst laden, wenn sie gebraucht werden.
	Notwendige Skripte erst laden, wenn sie vom User aufgerufen werden.
	Größe der Skripte beim Laden auf max. 300 KB reduzieren.

### Ungenutztes JavaScript

	Analyse: Welches JavaScript ist überflüssig?
	Ersetzen der Dateien

### Laufzeiten JavaScript



	Payloads reduzieren
	Code reduzieren
	Code Splitting
	Check: auf notwendige Teile reduziert?
	Caching einsetzen

## Effizienter Einsatz von Modulen in JavaScript Paketen

	Analyse, welche Module optimiert werden können
	Umsetzung

## Legacy JavaScript nur an veraltete Browser ausliefern

	JavaScript, das für veraltete Browser gedacht ist, nicht an moderne Browser ausliefern
--	--

## Minimierung der JavaScript Dateien

	Eigene Dateien minimieren (minify)
	Bei CMS zur Not Plugin dafür einsetzen.
	Drittanbieter-Dateien durch minimierte Version ersetzen.



## CSS

### Umfang der CSS Dateien

	Analyse: Welches CSS ist überflüssig?
	ggf. Plugins konfigurieren

### Minimierung der CSS Dateien

	Eigene Dateien minimieren (minify)
	Bei CMS zur Not Plugin dafür einsetzen
	Drittanbieter-Dateien durch minimierte Version ersetzen

## Server

### Energiemix der verwendeten Server

	Beim aktuellen Hoster nachfragen, mit welchem Energiemix dort die Server betrieben werden
	Falls graue Energiequellen genutzt werden: Website zu anderem Hoster umziehen

### Caching auf dem Server einsetzen

	Page-Cache einsetzen
	ggf. weitere Cache einsetzen



## Datenübertragung

### Text-Kompression am Server

	Einsatz von GZIP und idealerweise Brotli am Server konfigurieren
--	--

### Verwendung von HTTP/2

	Server auf HTTP/2 einstellen
	Falls bereits unterstützt, gerne auch schon HTTP/3 einsetzen

### Verwendung von HTTPS

	prüfen, ob Assets über HTTP geladen werden
	ansonsten diese entfernen oder ersetzen

### Anzahl der Netzwerkanfragen

	HTTP Caching einsetzen
	Dateien zusammenlegen
	Bilder reduzieren
	Schriftarten reduzieren
	Skripte reduzieren



## Weitere Punkte

### Browser-Cache

	Statische Dateien für mindestens ein Jahr in den Cache legen
--	--

### Gesamtgröße der Website

	Bilder reduzieren, die direkt beim Aufrufen der Seite geladen werden. Diese in modernen Formaten einbinden.
	Caching einsetzen
	Lazy Loading einsetzen
	Gesamtgröße der Website auf unter 1.500 KB reduzieren

### Komplexität und Umfang der Seite über die DOM-Größe

	Inhalte reduzieren, wo es geht und sinnvoll ist.
	Komplexität des Codes drosseln

### Umfang des Hauptablaufs des Rendering-Moduls

	Indirektes Optimieren durch Reduzieren der Seitengröße, umständlichen Codes, vieler Plugins, schlechter Multi-Purpose-Themes, zu umfangreicher CSS- und Javascript-Dateien, usw.
--	--

### Gesamtgröße aller geladenen Schriften

	Analyse, welche Schriften geladen werden und welche davon benötigt werden
	Unnötige Schriften entfernen



	Gesamtgröße aller geladenen Schriften auf unter 200 KB reduzieren
--	---

## Redirects

	die aktuelle Zieladresse eines Links verwenden
	Wenn diese sich im Lauf der Zeit ändert, den eigenen Link korrigieren, anstatt sich auf die Weiterleitung zu verlassen.

## Ladezeitpunkt von Drittanbieter-Code

	Drittanbieter-Code auf eine Weise einbetten, die Datenmenge reduziert
--	---

