

Mein optimiertes Dokument

6. August 2023

1 Website-Info

ChatGPT <https://chat.openai.com/>

1.1 Erstelle und Strukturiere mir Webseiteninhalte

Aufgabenstellung 1: Erstelle und Strukturiere mir Webseiteninhalte

Ziel dieser Aufgabe ist die Erstellung und Strukturierung von Dateien für eine Website gemäss den angegebenen Spezifikationen.

Anforderungen:

1. Root-Ordner (Hauptverzeichnis):

- Erstellen Sie eine `index.html` im Root-Ordner der Webseite.

2. CSS-Ordner:

- Erstellen Sie eine CSS-Datei im Ordner `css`. Die genaue Bezeichnung der Datei kann frei gewählt werden, sollte jedoch beschreibend sein, z.B. `main.css`.

3. Bilder-Ordner:

- Platzieren Sie einen Bildplatzhalter im Ordner `img`. Dies kann ein Standardbild sein, das später durch ein tatsächliches Bild ersetzt wird.

4. PHP-Ordner:

- Erstellen Sie eine PHP-Datei im Ordner `php`. Der Name der Datei sollte den Inhalt oder den Zweck der Datei widerspiegeln.

5. HTML-Ordner:

- Erstellen Sie eine zusätzliche HTML-Datei im Ordner `html`. Die Bezeichnung dieser Datei kann frei gewählt werden, sollte jedoch den Inhalt oder Zweck der Datei deutlich machen.

Hinweise: - Achten Sie auf eine klare und saubere Strukturierung der Dateien und Ordner.
- Beachten Sie die Best Practices bei der Benennung von Dateien und Ordnern, um Verwirrung oder Konflikte zu vermeiden.

Abgabe: - Ein Zip-Archiv oder einen Link zu einem Repository, das alle erstellten Dateien und Ordner enthält. - Ein kurzer Bericht (optional), in dem die Struktur und der Zweck jeder Datei beschrieben wird.

Deadline: - Abgabedatum: 1.9.2023

Bewertungskriterien: - die Qualität des Codes, die Einhaltung der Best Practices und die Ästhetik des Designs.

1.1.1 Überprüfe und Optimierte die Webseiteninhalte

Aufgabenstellung 2: Überprüfe und Optimierte die Webseiteninhalte

Ziel dieser Aufgabe ist die Analyse und Optimierung von Webinhalten, um eine bestmögliche Benutzererfahrung über verschiedene Geräte hinweg sicherzustellen.

Anforderungen:

1. Best Practices für html, CSS und php:

- Untersuchen Sie das aktuelle CSS und vergewissern Sie sich, dass es den gängigen Best Practices entspricht.

2. Barrierefreiheit:

- Stellen Sie sicher, dass die Webseite für alle Benutzer zugänglich ist, einschliesslich Menschen mit Behinderungen.

3. Responsive Design:

- Die Webseite sollte auf unterschiedlichen Geräten - von Mobiltelefonen über Tablets bis hin zu Desktop-Computern - gut lesbar und nutzbar sein.

4. Einheitlichkeit der MaSseinheiten:

- Verwenden Sie, wo immer möglich, die MaSseinheit "em" in Ihrem CSS, um Skalierbarkeit und Flexibilität über verschiedene Geräte und Bildschirmgrößen hinweg sicherzustellen.

5. Ziel-Dateien:

- Die Überprüfung und Anpassungen sollten sich auf "Index.html" sowie die zugehörigen PHP-Seiten erstrecken.

Abgabe: - Überarbeitetes CSS-Dokument - Überarbeitete index.html und PHP-Dateien, falls notwendig - Ein kurzer Bericht über die vorgenommenen Änderungen und deren Auswirkungen auf die Benutzererfahrung.

1.1.2 Kontrolliere nochmals

ChatGPT

Nehmen Sie die Rolle eines erfahrenen Webprogrammierer und Webdesigner ein

↪ .

Aufgabe: Kontrolliere nochmals unter Berücksichtigung von

- Best Practices
- Barrierefreiheit
- Responsive Design: gute Lesbarkeit auf Handy, Tablett und große
↪ Bildschirme
- Text soll vertikal zentriert und im Blickfeld des Betrachters sein
- Maßeinheiten in em
- Native Lazy Loading (für moderne Browser)
- Performance

Datei = " "

für index.html, html und php, css

1.1.3 CSS optimieren

```
# ChatGPT
optimiere meine CSS und Vermeide teure CSS-Selektoren

- Zusammenfassung von Wiederholungen
- Vermeidung von spezifischen Kombinationen:
- Optimierung der Media Queries:

datei = " "
```

1.2 Schreibstile

```
# ChatGPT
# Favorit: Expositorisch
# Schreibstile: Beschreibend, Expositorisch, Reflektierend, Akademisch,
    ↪ Kritisch
Schreibstil: Expositorisch
Erstellen Sie je nach Schreibstil eine kurze (ca. 35 Wörter) und
    ↪ ansprechende Zusammenfassung des folgenden Artikels. Die
    ↪ Zusammenfassung sollte für jemanden ohne wissenschaftlichen
    ↪ Hintergrund verständlich sein und gleichzeitig die wichtigsten
    ↪ Informationen genau wiedergeben.
Artikel: " "
```

1.3 HTML-Grundgerüsts

1. Best Practices:

- Das DOCTYPE-Element ist korrekt am Anfang des Dokuments angegeben.
- Das lang-Attribut im <html>-Tag ist gesetzt und zeigt an, dass der Inhalt auf Deutsch ist.
- Es gibt eine klare Trennung von Struktur (HTML) und Stil (CSS).
- Es wird ein verständlicher und barrierefreier Alternativtext für Bilder verwendet.

2. Barrierefreiheit:

- Semantische HTML-Tags wie <header>, <nav>, <main> und <footer> werden verwendet, um den Inhalt zu strukturieren.
- Es werden Rollen wie role="banner", role="navigation", role="main", und role="contentinfo" ↪ " verwendet, um die Semantik weiter zu erhöhen.
- Es gibt einen Alternativtext für das Bild, der seine Bedeutung oder Funktion beschreibt.

3. Responsive Design:

- Der Meta-Viewport-Tag <meta name="viewport" content="width=device-width, initial ↪ -scale=1.0"> ist vorhanden und sorgt dafür, dass die Seite korrekt auf verschiedenen Bildschirmgrößen skaliert.
- Für eine vollständige Überprüfung des responsiven Designs müssten wir den CSS-Code betrachten und die Webseite in verschiedenen Bildschirmgrößen testen. Es ist wichtig, dass das Layout der Webseite sich anpasst und der Inhalt auf verschiedenen Geräten gut lesbar bleibt.

4. MaSSeinheiten in `em`:

- Der bereitgestellte HTML-Code verwendet keine direkten Stilangaben, daher müssen alle GröSSen- und Abstandsangaben im CSS in `em`-Einheiten gemacht werden.

Was noch zu tun ist:

- Testen Sie die Webseite mit einem Screenreader oder Barrierefreiheits-Tool, um sicherzustellen, dass sie für alle Benutzer zugänglich ist.
- Überprüfen Sie alle anderen HTML- und PHP-Seiten mit den gleichen Kriterien.
- Stellen Sie sicher, dass alle externen Ressourcen (Bilder, Skripte, Stylesheets) korrekt geladen werden und keine gebrochenen Links vorhanden sind.
- Das CSS sollte regelmäSSig überprüft werden, um sicherzustellen, dass es den Best Practices entspricht und effizient ist.

Es wäre auch hilfreich, regelmäSSige Tests mit echten Benutzern durchzuführen, um Feedback zu Design, Usability und Barrierefreiheit zu erhalten. Dies wird Ihnen helfen, Bereiche zu identifizieren, die weiter optimiert werden müssen.

1.4 Performance einer Website zu verbessern

Die Performance einer Website zu verbessern, kann das Benutzererlebnis erheblich steigern und sich positiv auf die SEO auswirken.

1. Bilder optimieren:

- Verwende Formate wie WebP, das oft kleinere DateigröSSen als PNG oder JPEG bietet.
- Komprimiere Bilder, ohne die Qualität wesentlich zu beeinträchtigen.
- Nutze Responsive Images mit dem `srcset` Attribut, um verschiedene BildgröSSen für verschiedene Bildschirmauflösungen bereitzustellen.

2. Minimieren und kombinieren von Dateien:

- Minimiere CSS, JavaScript und HTML.
- Kombiniere mehrere CSS- oder JS-Dateien in eine einzelne Datei, um HTTP-Anfragen zu reduzieren.

3. Verwende Lazy Loading:

- Lade Bilder, Videos und Iframes erst, wenn sie im Viewport des Benutzers erscheinen.

4. Browser-Caching nutzen:

- Durch das Festlegen von Ablaufzeiten für Ressourcen können wiederkehrende Besucher Inhalte aus ihrem Browser-Cache laden.

5. Verwende Content Delivery Networks (CDNs):

- Ein CDN kann Inhalte schneller an Benutzer weltweit liefern, indem es Kopien deiner Ressourcen auf mehreren Servern verteilt.

6. Optimierte die Reihenfolge der Ressourcen:

- Lade kritischen CSS und JS zuerst und verschiebe nicht-kritische JS-Ressourcen mit dem `defer` Attribut.

7. Verringere Serverantwortzeit:

- Betrachte Hosting-Optionen und optimiere Backend-Code. Ein effizientes Webhosting und ein optimierter Server können die Antwortzeit verbessern.

8. Verwende HTTP/3 oder HTTP/2:

- Diese neueren Protokolle bieten Verbesserungen gegenüber HTTP/1.1, wie z.B. das Multiplexing von Anfragen.

9. Reduziere externe Ressourcen und Plugins:

- Jede externe Ressource (z.B. Schriftarten, Widgets, Plugins) kann die Ladezeit erhöhen.

10. Optimierte CSS und JavaScript:

- Vermeide teure CSS-Selektoren und reduziere die Tiefe von JavaScript-Aufrufen.

11. Nutze serverseitige Kompression:

- Verwende Gzip oder Brotli, um Dateigrößen vor dem Senden an den Client zu verringern.

12. Verbessere die Datenbankleistung:

- Optimierte Abfragen, setze Indizes ein und überlege, einen Cache wie Redis oder Memcached zu verwenden.

13. Mobile Optimierung:

- Stelle sicher, dass deine Website für mobile Geräte optimiert ist, da immer mehr Nutzer über Mobilgeräte zugreifen.

14. Überwache und messe regelmäSSig:

- Verwende Tools wie Google PageSpeed Insights, GTmetrix oder WebPageTest, um die Performance deiner Website regelmäSSig zu überprüfen.

1.5 Links

- SVG <https://svgooptimizer.com/de/>
- MOV to MP4/WEBM Converter <https://cloudconvert.com/mov-to-webm>
- XAMPP Apache + MariaDB + PHP <https://www.apachefriends.org/de/index.html>

1.6 Webdesign

- Download-Link für eine PDF-Datei
- Footer und die Bilder mittig positionieren und den Bildnamen anzuzeigen
- Darstellung der HTML `<pre><code>` Umgebung anpassen, Quellcode linksbündig darstellen und breite Texte mit Zeilenumbrüchen und Einrückungen versehen
- Hover- und Fokus-Stil für Links und Bilder, Bilder nebeneinander anordnen, Bildgröße
- CSS-Anweisungen optimieren und CSS-Variablen verwenden, um den Code sauberer und wartbarer zu machen

- Roboto-Font aus Google Fonts
- Inhalt sowohl horizontal als auch vertikal zu zentrieren und sich dabei am oberen Rand des Ansichtsfensters zu befinden

Browser-Caching ist ein effektives Mittel, um die Ladegeschwindigkeit einer Website zu verbessern.

```
# .htaccess-Datei für Apache-Server:
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType text/html "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType text/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1 month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 1 month"
</IfModule>
```

E-Tag (Entity Tag) ist ein Teil des HTTP-Protokolls, das von Webservern und Browsern verwendet wird, um die Gültigkeit einer bestimmten Version einer Ressource zu überprüfen.

```
# .htaccess-Datei für Apache-Server:
FileETag MTime Size
```

1.7 Webdesign-Merkmale von Spiegel

Die Website von “spiegel.de” ist die Online-Ausgabe von Deutschlands bekanntem Nachrichtenmagazin, dem SPIEGEL. Wie bei vielen Nachrichtenportalen, weist sie bestimmte Design- und Funktionsmerkmale auf, um den Inhalt klar und ansprechend zu präsentieren. Hier sind einige charakteristische Webdesign-Merkmale, die man auf solchen groSSen Nachrichtenportalen häufig findet:

1. **Responsives Design:** Die Website ist so gestaltet, dass sie auf verschiedenen Geräten – von Mobiltelefonen über Tablets bis hin zu Desktops – gut aussieht und funktioniert.
2. **Header mit Branding:** Oft gibt es ein prominent platziertes Logo und Navigationsmenü am oberen Rand der Seite.
3. **Mehrspaltiges Layout:** Auf der Startseite und in den Kategorien Seiten finden sich oft mehrere Spalten mit verschiedenen Nachrichtenkategorien oder Schlagzeilen.
4. **Bilder und Multimedia:** GroSSe, auffällige Bilder und eingebettete Videos sind häufig zu finden, um Geschichten zu ergänzen und die Aufmerksamkeit der Leser zu fesseln.
5. **Inhaltskarten:** Artikel können in “Karten” dargestellt werden, die einen Titel, ein Bild und vielleicht eine kurze Beschreibung enthalten.
6. **Scrollbare Inhalte:** Besonders auf der Startseite gibt es oft unendliches Scrollen oder “Lazy Loading”, bei dem neue Inhalte geladen werden, wenn der Benutzer zum Ende der Seite scrollt.

7. **Footer-Bereich:** Hier finden sich oft Links zu Impressum, Datenschutzerklärung, Kontaktinformationen und anderen wichtigen Seiten.
8. **Interaktive Elemente:** Dazu gehören Dropdown-Menüs, Suchfelder, Buttons für soziale Medien und mehr.
9. **Werbung:** GroSSe Nachrichtenseiten haben oft Display-Werbung in verschiedenen Formaten und GröSSen.
10. **Dunkler Modus:** Einige moderne Websites, einschlieSSlich Nachrichtenportalen, bieten einen “dunklen Modus” an, der besonders in Umgebungen mit wenig Licht angenehm für die Augen ist.
11. **Eingebettete Widgets:** Dazu können Live-Ticker, Umfragen oder interaktive Grafiken gehören.

Es ist wichtig zu beachten, dass die spezifischen Designentscheidungen, die für “spiegel.de” getroffen wurden, auf einer Vielzahl von Faktoren basieren, einschlieSSlich Markenidentität, Benutzererfahrung und Geschäftszielen. Wenn Sie das Design einer bestimmten Website nachahmen oder sich davon inspirieren lassen möchten, ist es ratsam, diese Faktoren zu berücksichtigen und sicherzustellen, dass Ihr Design den Bedürfnissen und Erwartungen Ihrer Zielgruppe entspricht.

1.8 Schreibstile

Favorit: Expositorisch

*# Schreibstile: Beschreibend, Expositorisch, Reflektierend, Akademisch,
↪ Kritisch*

Schreibstil: Expositorisch

Erstellen Sie je nach Schreibstil eine kurze (ca. 35 Wörter) und

- ↪ ansprechende Zusammenfassung des folgenden Artikels. Die
- ↪ Zusammenfassung sollte für jemanden ohne wissenschaftlichen
- ↪ Hintergrund verständlich sein und gleichzeitig die wichtigsten
- ↪ Informationen genau wiedergeben.

Artikel: " "

Es gibt zahlreiche Schreibstile, die je nach Kontext, Zweck und Publikum variieren können. Hier sind einige der gängigsten Schreibstile:

1. **Erzählerisch (narrativ):** Dieser Stil erzählt eine Geschichte, sei es Fiktion oder Non-Fiktion.
2. **Beschreibend:** Hierbei wird der Fokus auf detaillierte Beschreibungen gelegt, um ein Bild oder ein Gefühl für den Leser zu erzeugen.
3. **Überzeugend (argumentativ):** Der Zweck ist es, den Leser von einer bestimmten Meinung, Idee oder Perspektive zu überzeugen.
4. **Expositorisch:** Dieser Stil informiert oder erläutert ein Thema. Er wird oft in akademischen und journalistischen Kontexten verwendet.
5. **Reflektierend:** Hier reflektiert der Autor über persönliche Erfahrungen und Erlebnisse, oft in einem Tagebuch oder in persönlichen Essays.

6. **Direktiv (imperativ):** Dieser Stil gibt Anweisungen oder Anordnungen, wie in Kochrezepten oder Bedienungsanleitungen.
7. **Lyrisch:** Hierbei handelt es sich um einen poetischen Schreibstil, der oft in Gedichten, Liedtexten und manchmal in Prosa vorkommt.
8. **Kritisch:** Analyse und Bewertung eines Themas oder Werkes, oft in Buchkritiken oder Filmrezensionen.
9. **Journalistisch:** Hierbei geht es um das objektive Berichten von Nachrichten und aktuellen Ereignissen.
10. **Akademisch:** Ein formeller Stil, der in wissenschaftlichen Arbeiten, Aufsätzen und Forschungsartikeln verwendet wird.
11. **Dialogisch:** Dieser Stil betont den Dialog zwischen Charakteren oder zwischen Autor und Leser.
12. **Satirisch:** Ein humorvoller oder ironischer Stil, der oft dazu verwendet wird, Kritik zu üben oder auf Probleme aufmerksam zu machen.
13. **Epistolisch:** Ein Stil, der in Briefen oder Korrespondenzen verwendet wird.

Es ist wichtig zu beachten, dass viele Texte oft eine Kombination aus verschiedenen Stilen enthalten können, je nach Bedarf und Kontext des Schreibens.

2 Bewertung von HTML-Datei und CSS-Code

HTML-Datei

Unter Berücksichtigung Ihrer Anforderungen und basierend auf der bereitgestellten HTML-Datei sind hier einige Anmerkungen und Verbesserungen, um eine effektive, barrierefreie und responsive Webseite zu erstellen:

1. Best Practices:

- Sie haben bereits das HTML5 DOCTYPE, die charset Deklaration, und Meta-Viewport-Tags, das ist großartig.

2. Barrierefreiheit:

- Die Verwendung von `role` Attributen ist gut, diese können in einigen Fällen jedoch redundant sein, da HTML5-Elemente wie `<header>`, `<nav>`, `<main>` und `<footer>` bereits diese semantischen Rollen implizieren.
- Stellen Sie sicher, dass der Kontrast zwischen Text- und Hintergrundfarben ausreichend ist.
- Verwenden Sie ARIA-Attribute sparsam und nur dann, wenn sie notwendig sind.
- Es wäre auch gut, wenn Sie andere semantische HTML-Elemente wie `<article>`, `<section>` oder `<aside>` verwenden würden, wo es sinnvoll ist.

3. Responsive Design:

- Sie haben den Meta-Viewport-Tag verwendet, das ist sehr wichtig für ein Responsive Design. Der CSS-Code, den Sie zuvor bereitgestellt haben, enthält bereits Media-Queries für verschiedene Bildschirmgrößen.

4. Text vertikal zentrieren:

- Sie können Flexbox oder Grid verwenden, um den Inhalt innerhalb von `.container` vertikal zu zentrieren. Hier ist eine Möglichkeit mit Flexbox:

```
• .container {  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    height: 100vh; /* Mindestens die Höhe des Viewports */  
}
```

5. MaSSeinheiten in em:

- Die Verwendung von `em` für Ihre MaSSeinheiten ist bereits in Ihrem CSS umgesetzt.

6. Native Lazy Loading:

- Wenn Sie Bilder hinzufügen, können Sie das `loading="lazy"` Attribut verwenden, um native Lazy Loading zu aktivieren. Zum Beispiel:
- ``

7. Performance:

- Verkleinern und komprimieren Sie Ihre CSS- und JS-Dateien.
- Überlegen Sie, ob Sie ein Tool wie [Lighthouse](#) verwenden möchten, um Ihre Seite auf Performance-Probleme zu überprüfen.
- Optimieren Sie Bilder für das Web.
- Sie könnten auch in Erwägung ziehen, eine Content Delivery Network (CDN) zu verwenden, um Ihre Webseite schneller zu laden.

Abschließend sieht Ihr HTML-Grundgerüst schon recht solide aus. Sie sollten jedoch weiterhin Tools wie den [WAVE Accessibility Tool](#) oder [Lighthouse](#) verwenden, um sicherzustellen, dass Ihre Webseite sowohl barrierefrei als auch leistungsstark ist.

CSS-Code

Ihr CSS-Code scheint gut strukturiert und durchdacht zu sein. Es sind viele Best Practices implementiert. Hier sind jedoch einige Anmerkungen und Optimierungen, die Sie berücksichtigen könnten:

1. Best Practices:

- Sie verwenden CSS-Variablen, was ein ausgezeichneter Ansatz ist, um den Code wartbarer zu machen.
- Sie haben die Schatten, Farben und Schriftarten zentralisiert, was eine konsistente Gestaltung erleichtert.

2. Barrierefreiheit:

- Sie haben einen Fokus-Stil für Links hinzugefügt, was gut für die Barrierefreiheit ist. Es könnte hilfreich sein, auch Fokus-Stile für andere interaktive Elemente hinzuzufügen.

3. Responsive Design:

- Ihre Medienabfragen sehen solide aus, und Sie haben verschiedene Stile für verschiedene Bildschirmgrößen definiert.

4. Vertikale Zentrierung:

- Sie haben das `display: flex` verwendet, aber Sie haben nicht wirklich das vertikale Zentrieren von Text in Containern implementiert. Sie können dies durch die Verwendung von `align-items: center` in flex-Containern erreichen.

5. MaSSeinheiten in em:

- Sie haben `em`-Einheiten konsequent verwendet, was gut ist. Beachten Sie jedoch, dass die Basis-SchriftgröSSe (`--base-font-size`) in `px` definiert ist. Wenn Sie die volle Flexibilität von `em` nutzen möchten, könnten Sie die Basis-SchriftgröSSe in `rem` oder `%` definieren.

6. Native Lazy Loading:

- Sie haben keine direkte Implementierung von Lazy Loading für Bilder. Um native Lazy Loading zu implementieren, fügen Sie `loading="lazy"` zum `img`-Tag hinzu:
- ``

7. Performance:

- Sie haben Schatten und Übergänge verwendet, die die Performance beeinflussen könnten, besonders auf mobilen Geräten. Es ist jedoch schwer zu sagen, ohne die gesamte Seite in Aktion zu sehen. Tools wie [Google PageSpeed Insights](#) können hilfreich sein, um dies zu überprüfen.
- Achten Sie darauf, externe Schriftarten (wie Roboto) effizient zu laden, um die Webseiten-Performance nicht zu beeinträchtigen.

8. Sonstige Anmerkungen:

- Die `.responsive-video` Klasse ist definiert, aber es fehlen Stile, die sie wirklich responsiv machen. Sie könnten einen Wrapper mit einem festen Verhältnis verwenden und das Video darin absolut positionieren, um es responsiv zu machen.
- Die `--responsive-img-max-width` Variable wird im CSS nicht verwendet. Wenn Sie diese Variable für die maximale Breite von Bildern verwenden möchten, stellen Sie sicher, dass sie in Ihrem Code implementiert ist.
- Achten Sie darauf, dass Sie Alt-Texte für alle Bilder verwenden, um die Barrierefreiheit und SEO zu verbessern.

Insgesamt haben Sie bereits einen guten Ansatz gewählt, aber wie bei allem im Webdesign gibt es immer Raum für Verbesserungen und Anpassungen, abhängig von den spezifischen Anforderungen und Zielen Ihrer Webseite.