Engineer

Conception web adaptative

Mise en page qui s’adapte à l’écran de lecture

Ce que c’est

La conception web adaptative consiste à construire une page dont la mise en page s’adapte à l’écran sur lequel elle est lue, du téléphone étroit tenu verticalement au grand moniteur, plutôt qu’un site distinct pour chaque type d’appareil. En anglais, la forme longue responsive web design est le titre de l’article qui l’a nommée : celui d’Ethan Marcotte dans A List Apart, publié le 25 mai 2010. Il s’est inspiré de ce qu’il décrivait comme une discipline émergente, l’architecture adaptative, où murs et installations bougent et changent de forme à l’approche des personnes.

Marcotte appelait les grilles fluides, les images flexibles et les requêtes média « les trois ingrédients techniques de la conception web adaptative ». Une grille fluide dimensionne ses colonnes en proportion de l’espace disponible plutôt qu’en pixels fixes : la mise en page s’étire et se rétrécit avec la fenêtre. Une image flexible ne peut jamais devenir plus large que la colonne qui la contient. Une requête média permet à une feuille de style de changer de décision quand les seules proportions ne suffisent plus : lorsque trois colonnes deviennent trop étroites à lire, par exemple, la page passe à une seule.

La spécification Media Queries Level 4, projet de recommandation candidate du W3C daté du 19 février 2026, définit une requête média comme une expression logique prenant deux valeurs possibles, vrai ou faux : un test d’un aspect du navigateur ou de l’appareil qui affiche le document. Le test utilisé par les mises en page adaptatives est `width`, qui correspond sur un écran à la largeur de la zone d’affichage, barre de défilement comprise. Les longueurs d’une requête sont les mêmes longueurs CSS qu’ailleurs : un pixel y est donc un pixel CSS et non physique, et un `em` est mesuré depuis la taille de police initiale du navigateur, modifiable par le lecteur, jamais depuis une taille définie par la page.

Rien de cela ne fonctionne sur un téléphone sans demande de la page. Sans élément meta de zone d'affichage, les navigateurs mobiles disposent la page comme dans une fenêtre d'ordinateur, généralement large de 980 à 1 024 pixels CSS, puis réduisent le résultat. Chaque requête média voit donc un ordinateur. Le projet CSS Viewport présente ce comportement comme une mesure de compatibilité pour les pages conçues uniquement pour de grands écrans, inutile pour celles qui fonctionnent bien sur de petits écrans. La ligne `width=device-width, initial-scale=1` de l'exemple ci-dessous est la manière dont une page indique qu'elle en fait partie.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce que la conception web adaptative ? » pour éclairer une décision réelle. Nous pouvons vous aider à relier ce concept au système que vous construisez ou achetez. Des interfaces rapides, utilisables sur téléphone et cohérentes dans tous les navigateurs de vos clients.

Découvrir nos prestations Contactez-nous →

Vérifiez les hypothèses derrière le terme

Appuyez-vous sur l'explication et les réalisations liées pour formuler les questions de votre projet. Nous pouvons examiner les hypothèses avec vous avant le choix d'un outil. Développement cartographique GIS full-stack — cartographie full-stack, de la base au navigateur

Développement cartographique GIS full-stack Contactez-nous →

Pourquoi c'est important

L'argument de Marcotte portait autant sur la maintenance que sur l'apparence. Plutôt que d'adapter des conceptions isolées à un nombre toujours croissant d'appareils, il proposait de les traiter comme les facettes d'une même expérience. Une page, une adresse et une feuille de style permettent à une correction d'atteindre tous les lecteurs à la fois. Une nouvelle taille d'écran est servie par le même site, puisque la mise en page n'a jamais été écrite pour une taille particulière.

Le zoom est la raison sous-estimée par de nombreuses équipes. Augmenter le zoom grossit le pixel CSS et rétrécit donc la zone d'affichage comptée en pixels CSS. Les Web Content Accessibility Guidelines 2.2, recommandation du W3C datée du 12 décembre 2024, exigent au niveau AA, dans le critère de succès 1.4.10 Redistribution (Reflow), que le contenu soit utilisable sans défilement dans deux dimensions lorsque la largeur disponible est de 320 pixels CSS. Elles précisent que c'est ce que devient une fenêtre large de 1 280 pixels CSS à 400 % de zoom. À cette largeur de départ, si un lecteur règle le zoom du navigateur de son ordinateur portable à 400 %, chaque requête média de la page reçoit une largeur de zone d'affichage de 320 pixels. L'explication du critère par le W3C

note que les sites utilisent couramment la conception adaptative pour les petites zones d'affichage et que les contenus qui s'y adaptent aident aussi les personnes zoomant sur de grands appareils, pas seulement celles qui lisent sur téléphone.

Les octets sont la troisième raison. Le standard HTML donne trois motifs distincts pour proposer plusieurs fichiers d'une même image : le choix selon la zone d'affichage, la direction artistique et le choix du format d'image. Une photo occupant la largeur d'un téléphone nécessite beaucoup moins de pixels que sur un grand moniteur. Avec `srcset` et `sizes`, le navigateur peut récupérer le fichier plus petit sans script.

La largeur n'est pas la seule caractéristique à laquelle une page puisse répondre. Media Queries Level 4 décrit aussi le dispositif de pointage principal : `(pointer: coarse)` correspond à un dispositif de précision limitée, comme un écran tactile, et `(hover: none)` à un dispositif principal sans survol, ou à l'absence de dispositif de pointage. L'article de Marcotte proposait déjà d'utiliser les requêtes média pour donner aux liens de plus grandes cibles sur les petits écrans.

À quoi cela ressemble en pratique

Une liste de cartes affichant une colonne sur téléphone, deux sur une tablette tenue horizontalement et trois sur un grand écran commence par la ligne de zone d'affichage et une image que le navigateur peut choisir lui-même :

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

<img
  src="pool-400.jpg"
  srcset="pool-400.jpg 400w, pool-800.jpg 800w, pool-1600.jpg 1600w"
  sizes="(max-width: 40em) 100vw, (max-width: 64em) 50vw, 33vw"
  width="1600"
  height="900"
  alt="A tide pool at low water"
/>
```

La feuille de style contient les règles de mise en page :

```
.cards {
  display: grid;
  grid-template-columns: 1fr;
```

```
  gap: 1rem;
}

@media (width >= 40em) {
  .cards {
    grid-template-columns: repeat(2, 1fr);
  }
}

@media (width >= 64em) {
  .cards {
    grid-template-columns: repeat(3, 1fr);
  }
}

.cards img {
  max-width: 100%;
  height: auto;
}
```

La grille est la partie fluide : les colonnes `1fr` se partagent toute la largeur disponible, sans taille fixe. Les deux requêtes média déclenchent les changements de disposition et, étant écrites en `em`, suivent la taille de texte par défaut du lecteur. Une personne ayant choisi une police par défaut plus grande garde une seule colonne à des largeurs de fenêtre où les autres en voient déjà deux : c'est précisément ce que nécessitent ces lignes plus longues. La dernière règle garde chaque image dans sa colonne.

L'attribut `sizes` indique au navigateur, avant tout calcul de mise en page, la largeur d'affichage de l'image : la première condition vraie l'emporte et la dernière valeur s'applique si aucune ne l'est. Avec une zone d'affichage large de 39em, la première condition est vraie : l'image compte donc pour 39em de large. À 41em, c'est la deuxième, et l'image compte pour la moitié de la largeur, soit 20,5em. La fenêtre légèrement plus large nécessite le fichier plus petit, car la mise en page a changé. Le standard HTML détaille le même effet dans son exemple. À partir de ces nombres et des descripteurs `w` de `srcset`, le navigateur calcule la densité de chaque fichier sur cet écran et peut choisir n'importe lequel, en tenant compte de la densité de pixels, du zoom et éventuellement du réseau. Les attributs `width` et `height` lui permettent de réserver l'espace avant le téléchargement du fichier.

Une étude sur l'accélération des contrôles navigateur du site montre la même idée transformée en test. Dans ce travail réalisé en août 2026, les contrôles pilotant un navigateur sans interface ont reçu un tableau commun de vingt-deux

formats de fenêtre, dérivés de toutes les requêtes média effectivement présentes dans la feuille de style. Le contrôle de mise en page, qui examine la page dans chaque format de fenêtre prévu par la conception, est ainsi passé de 95 à 405 secondes. Ces formats viennent de la feuille de style, pas d'une liste d'appareils : les pages sont donc contrôlées aux véritables points de rupture de leur conception.

Où les problèmes surviennent

Des points de rupture qui interrogent l'appareil. Les exemples de Marcotte en 2010 interrogeaient `max-device-width`, la largeur de l'écran entier. Media Queries Level 4 déconseille désormais `device-width`, `device-height` et `device-aspect-ratio`, interdit aux auteurs de les utiliser et observe qu'ils servent souvent à détecter les appareils mobiles alors qu'ils ne disent rien de l'espace réellement disponible pour la page. Interrogez plutôt `width` et placez chaque point de rupture là où le contenu cesse de fonctionner, pas là où s'arrête un téléphone populaire.

Un intervalle entre deux points de rupture. Une feuille de style utilisant `(max-width: 320px)` pour une mise en page et `(min-width: 321px)` pour la suivante semble complète. Pourtant, la spécification souligne qu'une zone d'affichage d'une largeur strictement comprise entre 320 et 321 pixels ne correspond à aucune. Les largeurs fractionnaires existent : elles viennent de densités de pixels non entières et du zoom. À ces largeurs, aucun bloc ne s'applique. Utilisez la syntaxe d'intervalles, `(width <= 320px)` et `(width > 320px)`, pour que chaque largeur appartienne exactement à l'un des deux.

Le zoom est désactivé. Une ligne de zone d'affichage contenant `user-scalable=no` ou un `maximum-scale` inférieur à 2 échoue à la règle de test du W3C pour le critère de succès 1.4.4 Redimensionnement du texte (Resize Text), qui demande de pouvoir agrandir le texte à 200 pour cent. La règle précise que les navigateurs d'ordinateur ignorent cet élément et que, sur mobile, la plupart des navigateurs actuels l'ignorent par défaut ou proposent un réglage d'accessibilité permettant le zoom malgré tout. Les personnes encore touchées utilisent donc d'anciens navigateurs mobiles. Omettez ces deux propriétés et assurez-vous plutôt que la mise en page reste utilisable avec le zoom. Une police réduite dans un point de rupture étroit demande le même soin : l'explication de la redistribution publiée par le W3C l'autorise, mais seulement si le lecteur peut encore obtenir d'une manière ou d'une autre un texte deux fois plus grand que sa taille normale.

Les images ignorent la mise en page. Avec des descripteurs `w` dans `srcset` sans `sizes`, le standard HTML fait supposer au navigateur que l'image occupe toute la zone d'affichage. Pour une image affichée sur un tiers de l'écran, le navigateur peut donc télécharger un fichier environ trois fois plus large que nécessaire. Les pourcentages sont interdits dans `sizes` : utilisez `vw`. Lorsque `srcset` emploie des descripteurs `w`, les navigateurs qui comprennent `srcset` ignorent `src` : ne changer que `src` ne modifie donc rien pour la plupart des lecteurs. Écrivez `sizes` pour correspondre au CSS et modifiez les deux ensemble quand la mise en page change. Choisir les images par script n'est pas une correction : le standard avertit que cela peut télécharger l'image deux fois, ou pas du tout lorsque les scripts sont désactivés.

Un élément large pousse toute la page de côté. Un tableau de données, une longue ligne de code ou un contenu intégré de largeur fixe peut dépasser 320 pixels CSS et faire défiler toute la page horizontalement. Chaque paragraphe se lit alors par des allers-retours. L'explication de la redistribution publiée par le W3C exempte les contenus nécessitant deux dimensions, comme les tableaux, cartes et vidéos, mais pas les paragraphes voisins. Placez le tableau dans son propre conteneur défilant avec `overflow-x: auto` et donnez aux images et vidéos `max-width: 100%`, afin que seule la partie qui en a besoin défile.

Un écran étroit est pris pour un écran tactile. Un ordinateur portable zoomé à 400 % est étroit et possède une souris ; une grande tablette est large et n'a aucun survol. La mise en page relève de `width`, et les tailles de cibles et effets de survol de `pointer` et `hover`. Media Queries Level 4 conseille aussi de considérer `any-pointer` et `any-hover`, car un appareil peut offrir plusieurs moyens de pointer.

Pour aller plus loin

- Responsive Web Design — l'article d'Ethan Marcotte du 25 mai 2010 qui a nommé l'approche et ses trois ingrédients.
- Media Queries Level 4 — caractéristique `width`, syntaxe d'intervalles, unités relatives, pointage et survol, et raison de la dépréciation de `device-width`.
- Standard HTML : images — `srcset`, `sizes`, trois motifs pour proposer plusieurs fichiers et exemple détaillé à 29em et 32em.
- Comprendre le critère de succès 1.4.10 : redistribution — pourquoi 320 pixels CSS, lien avec le zoom et contenus exemptés.

- La zone d’affichage meta permet le zoom — règle de test du W3C pour `user-scalable` et `maximum-scale`, avec sa note sur la prise en charge par les navigateurs.
- CSS Viewport Module Level 1 — élément meta de zone d’affichage, propriétés et raison pour laquelle les navigateurs mobiles supposent une largeur d’ordinateur.

Glossaire

Choisissez selon vos contraintes réelles

Présentez le processus de travail, les interfaces existantes et les utilisateurs qui ont besoin du changement. Nous pouvons définir une livraison testable et une transmission permettant à votre équipe d'assurer la maintenance des éléments livrés.

Consulter les tarifs Contactez-nous →

Rendez la terminologie concrète

Demandez comment « Qu’est-ce que la conception web adaptative ? » s'applique à votre organisation. Un exemple concret tiré de votre système actuel suffit pour engager la discussion.

Découvrir nos prestations

welcome@engineer.company +45 50 10 61 84 Copenhague, Danemark

Emportez cette page avec vous

Fiche de contact

Créé avec soin et amour par D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Droit d'auteur © 2025 – aujourd'hui · Engineer ApS · Tous droits réservés.

Imprimé depuis https://engineer.company/fr/glossary/responsive-design/