Engineer

Zone d'affichage

la partie d'une page réellement visible dans une fenêtre

Ce que c’est

La zone d’affichage, ou viewport, est le rectangle dans lequel un navigateur met un document en page et le dessine : sur un écran, l’espace à l’intérieur des barres d’outils de la fenêtre ; sur papier, la zone de page de chaque feuille. CSS la prend constamment pour référence. Le bloc conteneur initial, la boîte dans laquelle est mis en page l’élément racine, en reprend les dimensions ; une requête média sur `width` ou `height` porte sur celles-ci, et les unités `vw` et `vh` en sont des centièmes. Sur un téléphone, le terme viewport et son pluriel viewports ont besoin d’un qualificatif. Les spécifications en proposent plusieurs : de mise en page, visuelle, petite, grande et dynamique.

Sur un ordinateur, la zone d’affichage et la fenêtre se confondent presque. Sur un téléphone, elles se séparent. Comme tant de documents ont été conçus pour des fenêtres d’ordinateur, les navigateurs mobiles mettent une page sans instructions en page sur une largeur fixe imitant une fenêtre d’ordinateur, généralement de 980 à 1 024 pixels CSS selon la spécification CSS Viewport, puis réduisent le résultat pour le faire tenir sur l’écran. Une page échappe à ce comportement grâce à un seul élément dans l’en-tête HTML du document, qui accepte `width`, `height`, trois limites d’échelle, `user-scalable` et `interactive-widget` :

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

Le zoom sépare la zone d’affichage en deux. La zone d’affichage de mise en page est celle sur laquelle CSS s’appuie pour disposer la page. Le zoom par pincement n’y touche pas, si bien que les retours à la ligne restent en place. La zone d’affichage visuelle est la partie de la page présente sur l’écran à cet instant : elle rétrécit lorsque le lecteur zoome et se déplace lorsqu’il fait glisser la vue. Sans réglage particulier, un clavier à l’écran ne réduit que la zone d’affichage visuelle ;

`interactive-widget=resizes-content` demande au navigateur de redimensionner aussi la zone de mise en page, et `overlays-content` de ne redimensionner ni l'une ni l'autre.

Les téléphones ajoutent des éléments d'interface du navigateur qui s'escamotent. La barre d'adresse occupe de la place au chargement d'une page et se rétracte lorsque le lecteur fait défiler celle-ci : la hauteur visible change ainsi sous une page qui n'a pas bougé. CSS Values 4 nomme trois tailles pour cela : la grande zone d'affichage suppose toutes les barres rétractables masquées, la petite les suppose toutes visibles, et la dynamique suit leur état effectif. Les unités portent les préfixes `lvh`, `svh` et `dvh`, avec les équivalents en largeur ; `vh` et `vw` sans préfixe sont définis par rapport à la grande zone d'affichage. Une interface qui se superpose au contenu sans le déplacer, généralement le clavier à l'écran, ne modifie aucune de ces longueurs.

Utilisez le concept pour prendre une décision

Utilisez « Qu'est-ce qu'une zone d'affichage ? » 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

Le design adaptatif repose sur la zone d'affichage. Un point de rupture portant sur la largeur dans une feuille de style interroge celle de la zone d'affichage. Une page qui omet l'élément `meta` est donc mise en page à une largeur d'ordinateur même sur un téléphone, puis réduite jusqu'à rendre le texte illisible, aussi soigneusement que soient écrites ses requêtes média.

La hauteur est l'axe difficile. La largeur peut changer avec la rotation ou le redimensionnement ; la hauteur peut aussi changer lorsque la barre d'outils glisse, lorsque le clavier s'ouvre et lorsque le téléphone pivote. Tout élément dimensionné selon la hauteur de l'écran doit donc préciser de quelle hauteur il s'agit. La petite zone d'affichage permet au contenu de tenir lorsque les barres rétractables sont visibles et laisse une bande vide lorsqu'elles se rétractent. La grande remplit l'écran et place du contenu derrière les barres lorsqu'elles sont

affichées. La dynamique tient compte des changements d'état des barres, et la spécification avertit elle-même qu'elle peut redimensionner le contenu pendant le défilement, ce qui déplace tout ce qui se trouve en dessous.

Les écrans ont aussi des coins. Un téléphone aux coins arrondis ou doté d'une découpe pour la caméra n'est pas un rectangle. Avec le réglage prédéfini `auto`, toute la page reste visible. Ajouter `viewport-fit=cover` demande que la zone d'affichage soit le rectangle circonscrit à l'ensemble de l'écran physique et confie à la page la tâche de tenir le texte à l'écart de la découpe grâce aux variables `env(safe-area-inset-*)`. Ces retraits doivent être nuls sur un écran rectangulaire ; une seule règle de marge intérieure convient donc aux deux types.

Le positionnement fixe dépend lui aussi de la zone d'affichage. Une boîte fixe se place par rapport à la zone d'affichage de mise en page, à laquelle CSS Positioned Layout donne les dimensions de la zone dynamique. Les bandeaux de cookies, fonds de fenêtres modales et menus plein écran restent ainsi immobiles pendant le défilement. Chacun suppose qu'aucun ancêtre n'a repris ce rôle. Les propriétés `transform`, `will-change` et `contain` peuvent toutefois attribuer ce rôle à un ancêtre.

L'impression est la zone d'affichage qu'une feuille de style réservée à l'écran ne rencontre jamais. Sur papier, les unités de zone d'affichage mesurent la zone de page, une requête média sur `width` porte sur la boîte de page, et un élément fixe se répète sur chaque page.

À quoi cela ressemble en pratique

Les pages de ce site déclarent `width=device-width, initial-scale=1, viewport-fit=cover`. Elles sont donc mises en page à la largeur du téléphone et vont d'un bord à l'autre sous une encoche. La feuille de style assume la seconde partie : la marge intérieure latérale vaut `max(1rem, env(safe-area-inset-left))`, avec son équivalent à droite. Le texte reste ainsi à un rem du bord et ne rentre davantage que là où se trouve une découpe. Sur les écrans peu hauts en paysage, le minimum passe à `max(2rem, 7vw, env(safe-area-inset-left))`, car les retraits sont nuls sur un téléphone sans découpe et un rem laissait le texte trop près du bord.

Les sections plein écran prennent leur hauteur dans une variable commune fixée à `100dvh`, avec `100vh` en repli. Le commentaire voisin en consigne la raison : `100vh` peut dépasser la hauteur visible lorsque la barre d'adresse est affichée et avait laissé apparaître la section suivante sous le bandeau d'ouverture. Deux

éléments ignorent délibérément cette variable. Le symbole de l’en-tête est dimensionné en `vh` statiques, car un symbole lié à la zone d’affichage dynamique changerait de taille lorsque l’interface du navigateur se rétracte et recomposerait le bandeau au milieu du défilement. Le fond fixe sur écran tactile mesure `100lvh` de haut : il ne change donc jamais de taille lorsque la barre d’adresse apparaît ou disparaît. Il dispose de son propre calque de composition, car les navigateurs tactiles le déchiraient en bandes claires et sombres lors du défilement inertiel.

Dans une ancienne mise en page d’accueil, chaque bloc de prestations avait une hauteur minimale de `100dvh`, occupant ainsi un écran tant que son contenu était moins haut que celui-ci. Les commentaires de la feuille de style, vérifiés en novembre 2026, consignent un essai à 852 par 393 pixels : le contenu d’un bloc atteignait 412, et les blocs se touchaient. Dans cette mise en page, une marge intérieure incluse dans les `100dvh` grâce à `box-sizing: border-box` ne coûtait rien sur un écran haut et n’ouvrait un espace que lorsque le contenu débordait.

Au stade d’implémentation du contrôle qualité documenté en août 2026, les tests suivaient les requêtes média. Le contrôle qualité du site piloté par navigateur avait étendu ses vérifications à une table commune de vingt-deux formes de fenêtres, tirées de chaque requête média réellement présente dans la feuille de style. Ce seul élargissement avait porté le contrôle de mise en page de 95 secondes à 405.

L’impression a révélé les différences entre moteurs. La feuille de style d’impression consigne des unités de zone d’affichage calculées par rapport à la boîte de page dans un moteur et à la fenêtre dans un autre. Dans le second, une page imprimée depuis une fenêtre large perdait donc un tiers de chaque ligne. La feuille de style de l’en-tête consigne l’autre face du même écart, celle des requêtes média : l’impression conservait l’orientation, la hauteur et la largeur de la fenêtre. Les feuilles imprimées depuis une fenêtre large et une fenêtre étroite avaient ainsi des en-têtes décalés de 47 pixels, jusqu’à ce que `screen and` en tête de requête limite la règle aux fenêtres.

Le piège du bloc conteneur est lui aussi consigné. Le travail sur la réduction des animations a établi que cinq propriétés différentes font chacune d’un élément le bloc conteneur de tout ce qui est positionné par rapport à la zone d’affichage. Cela avait silencieusement réduit un calque de fermeture à une fraction de l’écran.

Où les problèmes surviennent

Le téléphone affiche une minuscule page d'ordinateur. Sans élément `meta` de zone d'affichage, le navigateur compose la page à une largeur d'ordinateur, puis la réduit. Les requêtes média voient cette largeur. Ajoutez `width=device-width, initial-scale=1` et laissez le zoom actif : `user-scalable=no` retire au lecteur son propre moyen d'agrandir les petits caractères.

Une section plein écran dépasse le bas de l'écran. `100vh` correspond à la grande zone d'affichage. Lorsque la barre d'adresse est affichée, le bas de la section et tout bouton qui s'y trouve passent donc derrière elle. Utilisez `svh` pour tenir un élément à l'écart des barres d'outils déployées du navigateur, et `dvh` seulement là où un redimensionnement en plein défilement est acceptable.

Un calque plein écran se réduit à une boîte dans la page. Un ancêtre doté d'un `transform`, d'un `will-change` qui annonce une transformation, ou de certains réglages de confinement peut devenir le bloc conteneur pour `position: fixed`. Le calque prend alors les dimensions de cet ancêtre. Placez les calques en dehors de tout élément transformé.

Une page imprimée perd la fin de chaque ligne. Les unités de zone d'affichage et les requêtes média dimensionnées selon la fenêtre se glissent dans l'impression. Un moteur qui les calcule par rapport à la fenêtre met alors la feuille en page pour un écran. Écartez les unités de zone d'affichage de tout ce qui s'imprime et placez `screen and` devant les requêtes destinées aux fenêtres.

Pour aller plus loin

- CSS Viewport Module Level 1 — l'élément `meta` de zone d'affichage et les trois comportements d' `interactive-widget`.
- CSS Values and Units Level 4 : longueurs relatives à la zone d'affichage — les petites, grandes et dynamiques zones d'affichage et toutes les unités qui en dérivent.
- CSS Positioned Layout Level 3 — le positionnement fixe, les propriétés qui en changent le bloc conteneur et son comportement sur papier.
- CSS Environment Variables Level 1 — les retraits de zone sûre et la règle qui les rend nuls sur un écran rectangulaire.
- Media Queries Level 4 — `width`, `height` et `orientation`, mesurés sur la zone d'affichage à l'écran et sur la boîte de page à l'impression.

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 qu'une zone d'affichage ? » 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/viewport/