

CSS3: Responsive Web Design, RWD



CSS3 Media Queries

intro: Le Responsive Web Design (RWD) est une approche de conception web qui permet aux sites internet de s'adapter automatiquement à tous les types d'écrans et d'appareils. Cette technique garantit une expérience utilisateur optimale, que l'on navigue sur un **smartphone**, une **tablette**, un **ordinateur portable** ou un **écran de bureau** (aussi appelé desktop).

Quelques notions de base

Media Query : Condition CSS qui change les styles selon la taille de l'écran

Responsive : Site qui s'adapte automatiquement à tous les écrans

Breakpoint : Taille d'écran où le design change (ex: 600px, 900px)

Viewport : La zone visible dans le navigateur

px (pixel) : Unité de mesure pour les tailles

Balise `<link>` : Élément HTML permettant de relier une feuille de style CSS à une page HTML.

Attribut media : Paramètre ajouté à la balise `<link>` pour préciser dans quel contexte (écran, impression, etc.) appliquer les styles.

Média de destination : Support final sur lequel le contenu sera affiché (écran d'ordinateur, imprimante, tablette, etc.)

Mise en page : Organisation visuelle des éléments sur une page web.

Documentation officielles:

- [Alsacreation - Responsive Design - FR](#)
- [W3C - Doc officielle Type de media queries - EN](#)
- [MDN - Type de Media Queries - FR](#)

Les Media Queries

Type de media queries

Les media queries permettent d'appliquer des styles CSS spécifiques en fonction de certaines conditions, comme la **largeur de l'écran**, **l'orientation de l'appareil**, ou la **Résolution**.

Voici quelques exemples courants :

Par défaut, le type de média utilisé est all.

all:

Correspond pour tous les appareils. Print et Screen

print:

Correspond aux matériaux paginés et aux documents consultés en aperçu avant impression.

screen:

Correspond aux appareils dotés d'un écran.

speech:

Correspond aux outils de synthèse vocale.

Le concept

Imaginez que vous préparez un document : vous ne le présenterez pas de la même manière selon qu'il sera lu sur un écran ou imprimé sur papier. C'est exactement ce principe que CSS permet d'appliquer aux sites web !

L'évolution historique

Dès les versions CSS2 et HTML4 (technologies plus anciennes, **mais toujours fondamentales**), les développeurs pouvaient déjà indiquer pour quel support une feuille de style devait être utilisée. Cette fonctionnalité offrait une flexibilité remarquable pour adapter l'apparence d'un site selon le contexte.

Un cas d'usage concret : l'impression

Prenons l'exemple de l'impression d'une page web.

Lorsqu'un utilisateur imprime votre page, certains éléments deviennent inutiles ou gênants :

- Le menu de navigation : inutile sur papier puisqu'on ne peut pas cliquer dessus
- Les animations : impossibles à reproduire sur une feuille imprimée
- Les couleurs d'arrière-plan : consomment de l'encre inutilement

À l'inverse, vous pourriez vouloir :

Agrandir le texte pour faciliter la lecture

Réorganiser les colonnes pour s'adapter au format A4

Afficher les URL des liens pour que le lecteur sache où ils mènent

La technique : duplication des balises `<link>`

Pour appliquer différents styles selon le contexte, la solution consiste à créer plusieurs feuilles de style et à utiliser l'attribut media pour préciser quand chacune doit s'appliquer.

Voici comment mettre en oeuvre cette technique dans votre HTML :

```

<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8">
    <title>Mon site adaptable</title>

    <!--
        Première feuille de style : pour l'affichage à l'écran
        L'attribut media="screen" indique que ces styles ne s'appliquent
        QUE lorsque la page est consultée sur un écran (ordinateur, tablette, téléphone)
    -->
    <link rel="stylesheet" href="styles-ecran.css" media="screen">

    <!--
        Deuxième feuille de style : pour l'impression
        L'attribut media="print" signifie que ces styles ne seront utilisés
        QUE lorsque l'utilisateur imprime la page (Ctrl+P ou Cmd+P)
        Ces règles CSS remplaceront ou compléteront celles de styles-ecran.css
    -->
    <link rel="stylesheet" href="styles-impression.css" media="print">
  </head>
  <body>
    <!-- Contenu de votre page -->
    <nav class="menu-navigation">
      <!-- Ce menu sera visible à l'écran mais peut être masqué à l'impression -->
    </nav>

    <main>
      <!-- Votre contenu principal -->
    </main>
  </body>
</html>

```

Contenu du fichier `styles-ecran.css`

```

/* Styles pour l'affichage à l'écran uniquement */

.menu-navigation {
    display: flex;          /* Affiche le menu en ligne flexible */
    background-color: #333;  /* Fond sombre pour le menu */
    padding: 20px;          /* Espacement interne */
}

body {
    font-size: 16px;        /* Taille standard pour la lecture à l'écran */
    background-color: #f5f5f5; /* Fond légèrement gris */
}

```

Contenu du fichier `styles-impression.css`

```

/* Styles pour l'impression uniquement */

.menu-navigation {
    display: none;          /* Masque complètement le menu à l'impression */
                           /* Pourquoi ? Car il est inutile sur papier */
}

body {
    font-size: 12pt;         /* Taille optimale pour l'impression (en points) */
    background-color: white; /* Fond blanc pour économiser l'encre */
    color: black;           /* Texte noir pour une meilleure lisibilité */
}

/* Affiche les URLs des liens après leur texte */
a[href]::after {
    content: " (" attr(href) ")"; /* Ajoute l'URL entre parenthèses */
}

```

🔑 Points clés à retenir

- Séparation des contextes (le fond: l'HTML, la forme: le CSS) : Vous pouvez créer autant de feuilles de style que nécessaire pour différents contextes d'affichage
- L'attribut **media** est votre allié : Il permet de cibler précisément quand appliquer vos styles (screen, print, all, etc.)
- Optimisation de l'expérience : Cette approche améliore considérablement l'expérience utilisateur en adaptant le contenu à son contexte d'utilisation
- Économie et efficacité : Pour l'impression, cela permet d'économiser de l'encre et du papier tout en rendant le document plus lisible

Cette technique, bien que **datant de CSS2**, reste **la fondation des approches modernes** de design adaptatif que vous rencontrerez dans ce parcours de développeur !