

CSS



Introduction à Modernizr

Modernizr est une bibliothèque JavaScript open-source qui permet de détecter les fonctionnalités HTML5 et CSS3 des navigateurs. Elle aide les développeurs à créer des expériences utilisateur adaptées en fonction des capacités du navigateur.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js">
```

Installation de Modernizr

Il existe plusieurs façons d'installer Modernizr, y compris via CDN, téléchargement direct ou intégration avec des outils de gestion de paquets comme npm.

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js">
```

Détection des fonctionnalités HTML5

Modernizr peut détecter une large gamme de fonctionnalités HTML5, comme le support pour les éléments vidéo, audio et canvas.

```
if (Modernizr.video) {  
  console.log('HTML5 Video supporté');  
} else {  
  console.log('HTML5 Video non supporté');  
}
```

Détection des fonctionnalités CSS3

Modernizr teste les fonctionnalités CSS3 comme les animations, transitions, et transformations 3D.

```
if (Modernizr.csstransforms3d) {  
  console.log('CSS3 3D Transforms supporté');  
} else {  
  console.log('CSS3 3D Transforms non supporté');  
}
```

Utilisation des classes CSS

Modernizr ajoute des classes CSS à l'élément <html> pour chaque fonctionnalité détectée, ce qui permet d'appliquer des styles conditionnels.

```
.no-js .feature { display: none; }  
.js .feature { display: block; }
```

Polyfills et Modernizr

Modernizr peut être utilisé conjointement avec des polyfills pour ajouter la prise en charge des fonctionnalités non supportées dans les anciens navigateurs.

API JavaScript de Modernizr

Modernizr fournit une API JavaScript simple pour tester des fonctionnalités et réagir en conséquence.

```
if (Modernizr.canvas) {  
  console.log('Canvas supporté');  
} else {  
  console.log('Canvas non supporté');  
}
```


Tests personnalisés

Il est possible de créer des tests personnalisés dans Modernizr pour détecter des fonctionnalités spécifiques non incluses par défaut.

```
Modernizr.addTest('customFeature', function() {  
    return 'customFeature' in document;  
});
```

Utilisation avec des frameworks

Modernizr peut être intégré avec des frameworks comme React, Angular, et Vue.js pour améliorer la compatibilité des applications.

```
import Modernizr from 'modernizr';  
if (Modernizr.flexbox) {  
  // Code spécifique pour flexbox  
}
```

Introduction aux iframes

Les iframes (inline frames) sont des éléments HTML qui permettent d'inclure un autre document HTML à l'intérieur d'une page web. Ils sont souvent utilisés pour intégrer des contenus externes tels que des vidéos, des cartes ou des applications web.

```
<iframe src="https://www.example.com" width="600" height="400"></iframe>
```

Attributs des iframes

Les iframes ont plusieurs attributs importants tels que src, width, height, frameborder, et allowfullscreen. Ces attributs contrôlent la source, les dimensions et d'autres propriétés de l'iframe.

```
<iframe src="https://www.example.com" width="800" height="600" frameborder="0"
```

Sécurité et Sandboxing

Pour des raisons de sécurité, les iframes peuvent être "sandboxed" pour limiter les actions qu'elles peuvent effectuer. L'attribut sandbox peut prendre plusieurs valeurs telles que allow-scripts, allow-same-origin, et allow-popups.

```
<iframe src="https://www.example.com" sandbox="allow-scripts allow-same-origin"></iframe>
```

Communication entre l'iframe et la page parente

Les iframes peuvent communiquer avec leur page parente en utilisant l'API `postMessage`. Cela permet de passer des messages de manière sécurisée entre les deux documents.

```
<!-- Code dans l'iframe -->  
<script>  
  window.parent.postMessage('Hello from iframe', '*');  
</script>  
  
<!-- Code dans la page parente -->  
<script>  
  window.addEventListener('message', function(event) {  
    console.log('Message from iframe:', event.data);  
  });  
</script>
```

Utilisations courantes des iframes

Les iframes sont couramment utilisées pour intégrer des vidéos (par exemple, YouTube), des cartes (par exemple, Google Maps), des widgets de réseaux sociaux, et d'autres contenus interactifs. Elles permettent d'incorporer facilement du contenu externe sans rediriger l'utilisateur.

```
<!-- Intégration d'une vidéo YouTube -->  
<iframe width="560" height="315" src="https://www.youtube.com/embed/dQw4w9WgXcQ"  
  
<!-- Intégration de Google Maps -->  
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3151.835434509
```

Introduction à Less

Less est un préprocesseur CSS qui permet d'écrire des styles plus dynamiques et maintenables. Il étend les fonctionnalités de CSS avec des variables, des mixins, des fonctions, et plus encore.

```
@primary-color: #4D926F;  
body {  
  color: @primary-color;  
}
```


Installation de Less

Less peut être installé de plusieurs façons, notamment via npm ou en incluant un script dans le HTML pour le compiler en temps réel dans le navigateur.

```
<script src="https://cdn.jsdelivr.net/npm/less@4.1.1"></script>
```

Utilisation de Variables

Les variables dans Less permettent de stocker des valeurs que vous pouvez réutiliser tout au long de votre feuille de styles.

```
@font-size: 16px;  
@line-height: 1.5;  
body {  
  font-size: @font-size;  
  line-height: @line-height;  
}
```

Mixins de Base

Les mixins permettent de réutiliser des groupes de styles. Ils sont comme des fonctions dans Less.

```
.border-radius(@radius) {  
  border-radius: @radius;  
}  
.box {  
  .border-radius(10px);  
}
```

Mixins avec Paramètres

Les mixins peuvent accepter des paramètres, ce qui les rend encore plus puissants et réutilisables.

```
.border-radius(@radius) {  
  border-radius: @radius;  
}  
  
.box {  
  .border-radius(10px);  
}
```

Nesting (Imbrication)

Less permet d'impliquer les sélecteurs, ce qui rend le code CSS plus propre et plus lisible.

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    li {  
      display: inline-block;  
    }  
  }  
}
```

Opérations et Calculs

Less permet d'effectuer des calculs directement dans vos feuilles de styles.

```
@base: 5%;  
@width: @base * 2;  
.container {  
  width: @width;  
}
```

Fonctions intégrées

Less inclut de nombreuses fonctions intégrées pour manipuler les couleurs, les chaînes, et plus encore.

```
@base-color: #111;  
@light-color: lighten(@base-color, 20%);  
@dark-color: darken(@base-color, 20%);
```

Imports

Vous pouvez diviser votre code Less en plusieurs fichiers et les importer quand nécessaire.

```
// main.less
@import "variables.less";
@import "mixins.less";
@import "styles.less";
```


Héritage de Sélecteurs

Less permet de faire hériter un sélecteur d'un autre, ce qui simplifie la gestion des styles.

```
.button {  
  padding: 10px;  
  border: none;  
}  
.primary-button {  
  &:extend(.button);  
  background-color: blue;  
}
```

Gestion des Boucles

Less permet d'utiliser des boucles pour générer des styles répétitifs.

```
.loop(@i) when (@i > 0) {  
  .class-@{i} {  
    width: (@i * 10%);  
  }  
  .loop(@i - 1);  
}  
.loop(10);
```

Fonctions définies par l'utilisateur

Vous pouvez créer vos propres fonctions pour une réutilisation et une modularité accrues.

```
.square(@n) {  
  @return: @n * @n;  
}  
  
.size {  
  width: square(4);  
  height: square(4);  
}
```

Guarded Mixins (Mixins Conditionnels)

Les guarded mixins vous permettent d'appliquer des styles conditionnellement.

```
.mixin(@width) when (@width > 10px) {  
  width: @width;  
}  
.mixin(@width) when (@width <= 10px) {  
  width: 10px;  
}  
div {  
  .mixin(5px);  
}
```

Interpolation

Less permet d'utiliser l'interpolation pour créer des noms dynamiques de classes ou d'identifiants.

```
@base: "width";  
.{@base}-10 {  
  width: 10px;  
}
```

Utilisation des Mises à l'échelle

Less permet de créer des mises à l'échelle dynamiques pour les éléments.

```
.scale(@ratio) {  
  transform: scale(@ratio);  
}  
  
.element {  
  .scale(1.5);  
}
```

Manipulation des Couleurs

Less offre des fonctions pour manipuler les couleurs, comme la teinte, la saturation et la luminosité.

```
@base-color: #5a5;  
.color-adjusted {  
  color: saturate(@base-color, 20%);  
}
```

Introduction au RGAA

Le Référentiel Général d'Amélioration de l'Accessibilité (RGAA) est un ensemble de règles et de critères visant à garantir l'accessibilité des sites web publics en France. Il s'assure que les sites sont utilisables par tous, y compris les personnes handicapées.

```
<a href="https://accessibilite.numerique.gouv.fr/">En savoir plus sur le RGAA</a>
```


Définition d'un site Web "accessible"

Un site web accessible est un site conçu pour être utilisé par tout le monde, indépendamment des handicaps physiques ou mentaux. Cela inclut des fonctionnalités comme la navigation au clavier, les lecteurs d'écran, et des contrastes de couleurs adaptés.

```
<body>
  <header>
    <nav aria-label="Main navigation">
      <ul>
        <li><a href="#content">Skip to content</a></li>
        <li><a href="/about">About</a></li>
      </ul>
    </nav>
  </header>
</body>
```

Les outils dédiés à l'accessibilité

Divers outils peuvent être utilisés pour tester et améliorer l'accessibilité des sites web, tels que les validateurs automatiques, les lecteurs d'écran, et les extensions de navigateur.

```
<!-- Outil de validation -->  
<a href="https://wave.webaim.org/">WAVE Web Accessibility Evaluation Tool</a>
```

```
<!-- Outil de test de contraste -->  
<a href="https://color.ally.com/Contrast/">Contrast Checker</a>
```

Bonnes pratiques de construction HTML

Utiliser des balises HTML correctement et de manière sémantique est crucial pour l'accessibilité. Cela inclut l'utilisation appropriée des titres, des listes, et des formulaires.

```
<h1>Title of the page</h1>
<section>
  <h2>Section Title</h2>
  <p>This is a paragraph within the section.</p>
</section>
```

La sémantique des éléments

La sémantique HTML permet de donner un sens au contenu pour les navigateurs et les technologies d'assistance. Utiliser des balises comme `<header>`, `<nav>`, `<main>`, et `<footer>` aide à structurer le contenu.

```
<main>
  <article>
    <header>
      <h1>Article Title</h1>
    </header>
    <section>
      <p>This is the article content.</p>
    </section>
    <footer>
      <p>Author: John Doe</p>
    </footer>
  </article>
</main>
```

Les attributs fondamentaux pour l'accessibilité

Utiliser des attributs comme aria-label, aria-hidden, et role améliore l'accessibilité en fournissant des informations supplémentaires aux technologies d'assistance.

```
<button aria-label="Close">X</button>
<nav aria-label="Main navigation">
  <ul>
    <li><a href="/home">Home</a></li>
    <li><a href="/services">Services</a></li>
  </ul>
</nav>
```

Couleurs et Contrastes

Les couleurs doivent être choisies avec soin pour assurer un contraste suffisant entre le texte et l'arrière-plan, facilitant ainsi la lecture pour tous les utilisateurs.

```
body {  
  color: #333;  
  background-color: #fff;  
}
```

Introduction aux Frameworks de Style

Les frameworks de style sont des bibliothèques CSS qui facilitent la création de sites web réactifs et attrayants. Ils offrent des composants prédéfinis, des grilles de mise en page et des thèmes personnalisables.

Bootstrap

Bootstrap est l'un des frameworks CSS les plus populaires, développé par Twitter. Il offre une vaste gamme de composants prêts à l'emploi, une grille flexible, et des plugins JavaScript.

```
<div class="container">  
  <div class="row">  
    <div class="col-md-6">Colonne 1</div>  
    <div class="col-md-6">Colonne 2</div>  
  </div>  
</div>
```


Foundation

Foundation, créé par Zurb, est un framework CSS robuste et flexible. Il est conçu pour être responsive et mobile-first, avec des outils puissants pour la typographie, les formulaires, les boutons et bien plus.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.6.3/css/
<div class="grid-container">
  <div class="grid-x grid-padding-x">
    <div class="cell small-6">Colonne 1</div>
    <div class="cell small-6">Colonne 2</div>
  </div>
</div>
```

Bulma

Bulma est un framework CSS moderne basé sur Flexbox. Il est simple à utiliser et à personnaliser, offrant des composants réactifs et une typographie élégante.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.3/css/bulma.min.css">
<div class="columns">
  <div class="column is-half">Colonne 1</div>
  <div class="column is-half">Colonne 2</div>
</div>
```

Tailwind CSS

Tailwind CSS est un framework utilitaire-first qui permet de créer des designs sans écrire de CSS personnalisé. Il offre des classes utilitaires pour la mise en page, la typographie, les couleurs, et plus encore.

```
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css"
<div class="flex">
  <div class="w-1/2 p-4">Colonne 1</div>
  <div class="w-1/2 p-4">Colonne 2</div>
</div>
```

Semantic UI

Semantic UI utilise des classes CSS basées sur le langage naturel, facilitant la compréhension et l'utilisation du code. Il offre une grande variété de thèmes et de composants.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/semantic-ui/2.4.1/  
<div class="ui grid">  
  <div class="eight wide column">Colonne 1</div>  
  <div class="eight wide column">Colonne 2</div>  
</div>
```

Materialize

Materialize est un framework CSS basé sur Material Design de Google. Il offre des composants modernes et réactifs qui suivent les principes de Material Design.

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css
<div class="row">
  <div class="col s6">Colonne 1</div>
  <div class="col s6">Colonne 2</div>
</div>
```

Introduction aux Éléments Audio et Vidéo en HTML5

HTML5 introduit les éléments `<audio>` et `<video>` pour intégrer des médias directement dans les pages web sans nécessiter de plugins externes comme Flash. Ces éléments permettent une intégration plus fluide et une meilleure accessibilité.

L'Élément <audio>

L'élément <audio> permet d'intégrer des fichiers audio. Il prend en charge plusieurs formats comme MP3, WAV, et Ogg. Il inclut des attributs pour contrôler la lecture, le volume et plus encore.

```
<audio controls>
  <source src="audiofile.mp3" type="audio/mpeg">
  <source src="audiofile.ogg" type="audio/ogg">
  Votre navigateur ne supporte pas l'élément audio.
</audio>
```

L'Élément <video>

L'élément <video> permet d'intégrer des fichiers vidéo. Il prend en charge plusieurs formats comme MP4, WebM, et Ogg. Il inclut des attributs pour les contrôles, les sous-titres, et les dimensions.

```
<video width="600" controls>
  <source src="videofile.mp4" type="video/mp4">
  <source src="videofile.webm" type="video/webm">
  Votre navigateur ne supporte pas l'élément vidéo.
</video>
```


Attributs et Contrôles de l'Élément Audio

L'élément <audio> comprend des attributs comme controls, autoplay, loop, et muted pour personnaliser la lecture audio.

```
<audio controls autoplay loop muted>  
  <source src="audiofile.mp3" type="audio/mpeg">  
  Votre navigateur ne supporte pas l'élément audio.  
</audio>
```

Attributs et Contrôles de l'Élément Vidéo

L'élément `<video>` comprend des attributs comme `controls`, `autoplay`, `loop`, `muted`, `poster` (image d'affiche), et des dimensions (`width`, `height`).

```
<video width="600" height="400" controls poster="posterimage.jpg">  
  <source src="videofile.mp4" type="video/mp4">  
  Votre navigateur ne supporte pas l'élément vidéo.  
</video>
```