

Programmation avancée & Symfony

BUT Informatique - 3ème année

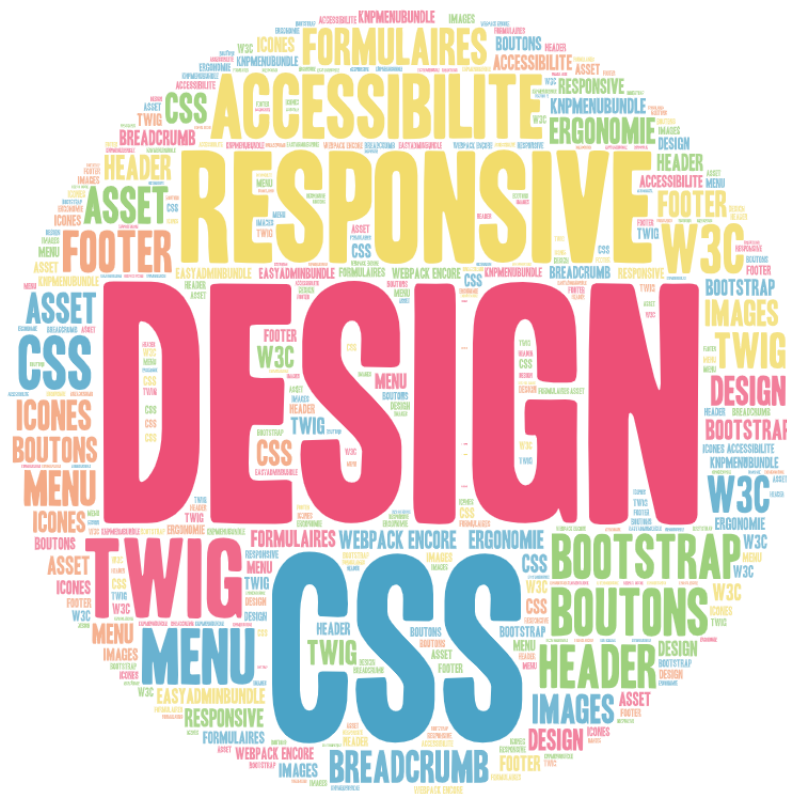
TP 5

Design / CSS

© Claire BOMBEAUX

Tous droits réservés



Version : **15/11/2025**



Objectifs du TP 5

- Mettre en place le **design** général, les styles **CSS**,
- Créer le **menu** de l'application
- Créer le **fil d'ariane (breadcrumb)** de l'application
- Connaître quelques outils, ressources et frameworks CSS

Sommaire

 Objectifs du TP 5	0
 Sommaire	1
1. Design et styles CSS.....	2
1.1 Design général / template de base / TWIG.....	2
1.2 Style CSS	4
CSS « traditionnel »	5
Bootstrap.....	5
Inclure une feuille de style CSS, un javascript ou une image via <i>Asset</i>	6
1.3 Menu	7
1.4 Formulaires	10
[TP 5] Annexe 1 : Ressources & outils.....	12
Outils de test d'accessibilité, ergonomie et performances	16
OPQUAST.....	17

1. Design et styles CSS

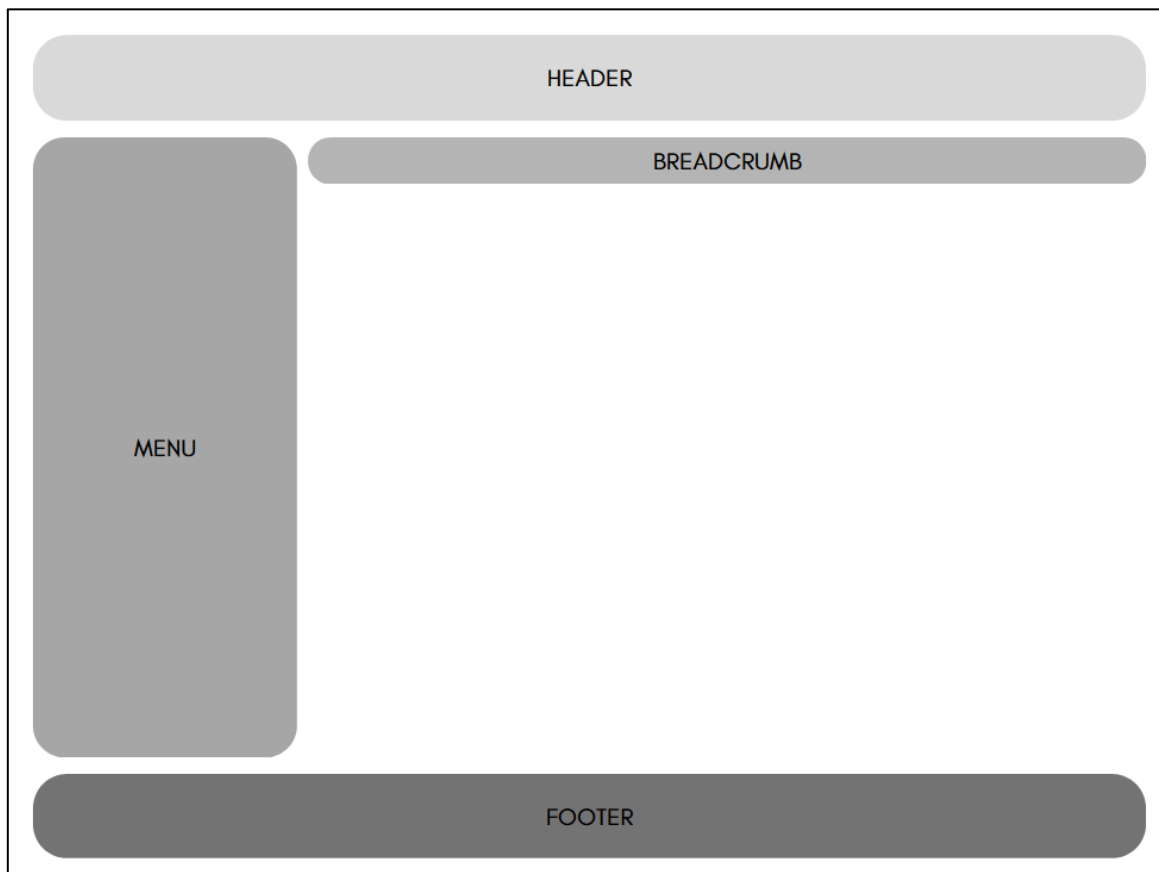
La partie design de votre application est plutôt libre mais elle est requise. Pour vous aider, en plus des directives ci-dessous, la partie **ANNEXES** de ce TP vous présente des **outils et ressources** (outils de validation de code, bibliothèques d'icônes ...)

1.1 Design général / template de base / TWIG

Les éléments suivants sont attendus :

- **Header**
Un entête, commune à toutes les pages
- **Menu**
Commun à toutes les pages.
L'item « courant » c'est-à-dire celui où l'on se situe doit se distinguer des autres.
- **Breadcrumb**
Fil d'ariane, permettant de se repérer, se situer dans l'application
- **Footer**
Un pied de page, commun à toutes les pages

Exemple de structure générale de pages :



Dupliquez le moins de code possible, que ça soit le HTML dans les templates ou les styles CSS, il donc convient de d'utiliser un template « général » : la base (*base.html.twig*) + de l'héritage/extension de templates.

```
base.html.twig
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>{% block title %}JVM{% endblock %}</title>
6
7     <link rel="icon" href="data:image/svg+xml,<svg xmlns=%22http://www.w3.org/2000/svg%22><text y=%221.2em%22 font-size=%2296%22>●</text><text y=%221.3em%22 x=%220.128%22><text y=%221.2em%22 font-size=%2296%22>●</text><text y=%221.3em%22 x=%220.128%22><text y=%221.2em%22 font-size=%2296%22>●</text><text y=%221.3em%22 x=%220.128%22><text y=%221.2em%22 font-size=%2296%22>●</text></svg>" />
8
9     {% block stylesheets %}
10      <link rel="stylesheet" href="{{ asset('style/style.css') }}" />
11      <link rel="stylesheet" href="{{ asset('style/form-style.css') }}" />
12      <link rel="stylesheet" href="{{ asset('style/menu-style.css') }}" />
13      <link rel="stylesheet" href="{{ asset('style/font-awesome.css') }}" />
14      <link rel="stylesheet" href="{{ asset('style/font-awesome-animation.css') }}" />
15    {% endblock %}
16
17    {% block javascripts %}
18    {% endblock %}
19  </head>
20  <body>
21    {{ knp_menu_render('main') }}
22    {% block body %}
23
24
25
26    {% endblock %}
27  </body>
28 </html>
```

Exemple de template général - base.html.twig

```
1 {% extends 'base.html.twig' %}
2
3 {% block title %}Liste des jeux vidéo{% endblock %}
4
5 {% block body %}
6   <h1 class="title1">Liste des jeux vidéo</h1>
7
8   <table class="table">
9     <thead>
10      <tr>
11        <th>Id</th>
12        <th>Image</th>
13        <th>Titre</th>
14        <th>Genre</th>
15        <th>Date de sortie</th>
16        <th>Prix</th>
17        <!--<th>Url Image</th>!-->
18        <th>Editeur</th>
19        <th>Développeur</th>
20      </tr>
21    </thead>
22  </table>
23{% endblock %}
```

Exemple de template étendant base.html.twig (à la ligne 1)

Les directives suivantes sont également à respecter :

- L'ensemble du site doit être en **langue française**, incluant les labels des boutons d'actions des CRUD (Bouton *new*, *create*...)
- Une **cohérence**, une **harmonisation** est attendue :
 - Pour la **forme** : les styles (CSS).
Par exemple : tous les boutons de l'application doivent avoir le même style.
 - Pour le **fond** : informations et textes.
Par exemple, toutes les actions de l'application sont des verbes à l'infinitif : *Ajouter*, *Modifier*...

- Le site doit être « **responsive** » (design réactif/adapté/réceptif, c'est-à-dire s'adaptant au format de l'appareil / au périphérique à partir duquel on le consulte : ordi., tablette, téléphone...)



- Avoir des pages HTML valides **W3C**
- Pensez à l'**ergonomie**, à l'**accessibilité (a11y)** et à l'**inclusivité**

➔ Contraste couleur, taille de police, balises sémantiques HTML (<nav>, <article>, <button>,...), libellés explicites, messages d'erreur explicites, facilité de navigation



- Pensez également à **organiser correctement vos sources** (vos fichiers liés au design)

1.2 Style CSS

Afin de mettre en place le style CSS de votre projet, vous pouvez choisir entre ces deux options : CSS « traditionnel » (développé manuellement) ou via un framework CSS tel que *Bootstrap* ou *Tailwindcss*



- **Rappel** : Une **cohérence**, une **harmonisation** est attendue pour la forme (les styles CSS). Par exemple, tous les boutons de l'application doivent avoir le même style. [...]
- Attention à ne pas « re-designer » le *profil* de Symfony (la *toolbar*)

CSS « traditionnel »

Il convient de placer vos feuilles de style dans `public/css/` :

[Sans utilisation de Webpack Encore]

```
public/  
└─ css/  
    └─ style.css  
    └─ buttons.css
```

```
<!DOCTYPE html>  
<html>  
<head>  
    <link rel="stylesheet" href="{{ asset('css/style.css') }}">  
    <link rel="stylesheet" href="{{ asset('css/buttons.css') }}">  
</head>  
<body>  
    {% block body %}{% endblock %}  
</body>  
</html>
```

Dans un template spécifique :

```
1  {% extends 'base.html.twig' %}  
2  
3  ▼ {% block stylesheets %}  
4      {{ parent() }}  
5      <link rel="stylesheet" href="{{ asset('css/pages/contact.css') }}">  
6  {% endblock %}  
7  
8  ▼ {% block body %}  
9  ▼      <div class="contact-page">  
10         <!-- Contenu -->  
11         </div>  
12  {% endblock %}
```

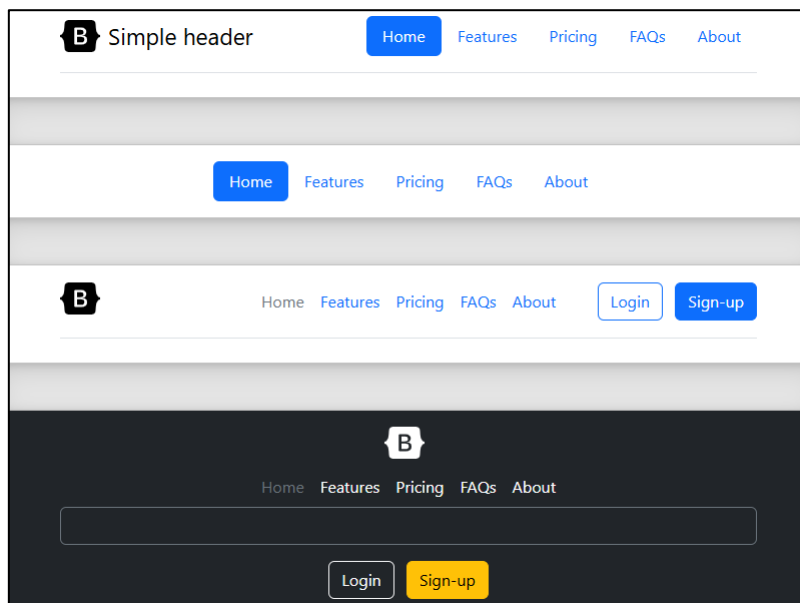
Bootstrap

Bootstrap est un framework CSS open-source qui fournit une collection de composants HTML, CSS et JavaScript préconstruits pour créer rapidement des interfaces web responsives et modernes.

Il inclut une grille flexible, des composants : boutons, formulaires, navigation, modales... et des styles CSS.



Exemples de composants Bootstrap :



En vous aidant de la documentation de *Bootstrap* et de *Symfony*, il vous faudra d'abord ajouter *Bootstrap* à votre projet.

Pour cela plusieurs méthodes sont possibles : avec **Webpack Encore** (recommandé mais non détaillé dans ce projet), via **CDN** (simple et rapide mais nécessite internet pour le fonctionnement dans l'appli. donc pas forcément approprié) ou encore en **téléchargeant l'intégralité des sources** de Bootstrap. Renseignez-vous sur ces différentes méthodes afin d'en choisir une.

Documentation :

- Get started with Bootstrap : <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- Using Bootstrap CSS & JS with Webpack Encore : <https://symfony.com/doc/current/frontend/encore/bootstrap.html>
- Bootstrap 5 Form Theme : <https://symfony.com/doc/current/form/bootstrap5.html>

Exemple d'utilisation de Bootstrap

```
$form = $this->createFormBuilder()
    ->add('name', TextType::class, ['attr' => ['class' => 'form-control']])
    ->add('submit', SubmitType::class, ['attr' => ['class' => 'btn btn-primary']])
    ->getForm();
```

Inclure une feuille de style CSS, un javascript ou une image via *Asset*

Pour inclure des images, des fichiers CSS ou JavaScript, il convient d'utiliser la fonction **asset()** du composant **Asset**.

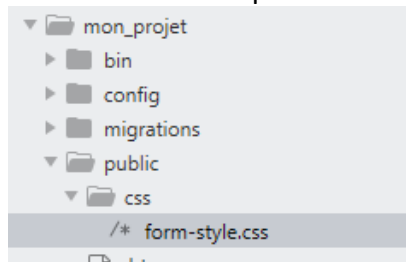
Asset n'est pas installé par défaut. Commande d'install. :

composer require symfony/asset

Exemple d'inclusion d'une feuille de style CSS

```
<link rel="stylesheet" href="{ { asset('css/form-style.css') } }">
```

Avec le fichier CSS présent dans /css :



Exemple d'inclusion d'une image :

```

```

Le path de l'image étant *public/images/logo.png*

1.3 Menu

Pour créer un menu d'application dans Symfony de manière rapide et simple, il existe plusieurs approches possibles, telles que :

KnpMenuBundle (*Recommandé*)

KnpMenuBundle est un bundle connu permettant de gérer des menus de façon relativement simple et rapide. Celui-ci permet de gérer les droits d'accès aux items du menu (restreindre ou non l'accès à des parties).

Cmde d'install. : `composer require knplabs/knp-menu-bundle`

```
PS C:\wamp64\www\mon_projet> composer require knplabs/knp-menu-bundle
./composer.json has been updated
Running composer update knplabs/knp-menu-bundle
Loading composer repositories with package information
Updating dependencies
Lock file operations: 2 installs, 0 updates, 0 removals
  - Locking knplabs/knp-menu (v3.8.0)
  - Locking knplabs/knp-menu-bundle (v3.6.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 2 installs, 0 updates, 0 removals
  - Downloading knplabs/knp-menu (v3.8.0)
  - Downloading knplabs/knp-menu-bundle (v3.6.0)
  - Installing knplabs/knp-menu (v3.8.0): Extracting archive
  - Installing knplabs/knp-menu-bundle (v3.6.0): Extracting archive
Generating autoload files
65 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!

Symfony operations: 1 recipe (7334541447f472c0c37c27188c811678)
  - Configuring knplabs/knp-menu-bundle (>=v3.6.0): From auto-generated recipe
Executing script cache:clear [OK]
Executing script assets:install public [OK]

What's next?

Some files have been created and/or updated to configure your new packages.
```


Exemple de configuration :

```
services.yaml
1  # This file is the entry point to configure your own services.
2  # Files in the packages/ subdirectory configure your dependencies.
3
4  # Put parameters here that don't need to change on each machine where the app is deployed
5  # https://symfony.com/doc/current/best_practices.html#use-parameters-for-application-configuration
6  parameters: {}
7
8
9
10
11
12
13  services:
14      # default configuration for services in *this* file
15      _defaults: {}
16
17      # makes classes in src/ available to be used as services
18      # this creates a service per class whose id is the fully-qualified class name
19      App\: {}
20
21      # add more service definitions when explicit configuration is needed
22      # please note that last definitions always *replace* previous ones
23
24      # KnpmenuBundle - Creating Menu Builders as Services
25      app.menu_builder:
26          class: App\Menu\MenuBuilder
27          arguments: ["@knpmenu.factory"]
28          tags:
29              - { name: knpmenu.menu_builder, method: createMainMenu, alias: main } # The alias is what
```

Exemple d'utilisation « *Creating Menu Builders as Service* » :

Avec uniquement des items de premiers niveaux

```
MenuBuilder.php
1  <?php
2
3  namespace App\Menu;
4
5  use Knp\Menu\Attribute\AsMenuBuilder;
6  use Knp\Menu\FactoryInterface;
7  use Knp\Menu\ItemInterface;
8  /**
9   * Creating Menu Builders as Services
10   */
11  class MenuBuilder
12  {
13      private $factory;
14
15      /**
16       * Add any other dependency you need...
17       */
18      public function __construct(FactoryInterface $factory)
19      {
20          $this->factory = $factory;
21      }
22
23      #[AsMenuBuilder(name: 'main')] // The name is what is used to retrieve the menu
24      public function createMainMenu(array $options): ItemInterface
25      {
26          $menu = $this->factory->createItem('root');
27
28          $menu->addChild('Accueil', ['route' => 'index']);
29
30          $menuJV = $menu->addChild('Jeux vidéo', ['route' => 'app_jeu_video_index']);
31
32          $menuGenre = $menu->addChild('Genres de jeux vidéo', ['route' => 'app_genre_index']);
33
34          $menuEditeur = $menu->addChild('Editeurs de jeux vidéo', ['route' => 'app_editeur_index']);
35
36          // ... add more children
37
38          return $menu;
39      }
40  }
```

- Accueil
- Jeux vidéo
- Genres de jeux vidéo
- Editeurs de jeux vidéo

Avec items de premiers et seconds niveaux

```

32 // $menuGenre = $menu->addChild('Genres de jeux vidéo', ['route' => 'app_genre_index']);
33 $menuGenre = $menu->addChild('Genres de jeux vidéo');
34 $menuGenre->addChild('Liste des genres', ['route' => 'app_genre_index']);
35 $menuGenre->addChild('Création d'un genre', ['route' => 'app_genre_new']);
36
37 $menuEditeur = $menu->addChild('Editeurs de jeux vidéo', ['route' => 'app_editeur_index']);
38
39 // ... add more children
40
41 return $menu;
42 }
43
44

```

- Jeux vidéo
 - Genres de jeux vidéo
 - Liste des genres
 - Création d'un genre
- Editeurs de jeux vidéo

Accueil

Jeux vidéo

Genres de jeux vidéo

Editeurs de jeux vidéo

Liste des éditeurs

Id	Nom	Pays	Description	Site web	Actions
			Sony Interactive Entertainment (ソニ		

Accueil

Jeux vidéo

Genres de jeux vidéo

Editeurs de jeux vidéo

Liste des éditeurs

Id	Nom	Pays	Description	Site web	Actions
3	Sony Interactive Entertainment	Japon	Sony Interactive Entertainment (ソニー・インタラクティブエンターテインメント) est une filiale du conglomérat japonais Sony, spécialisée dans l'industrie vidéoludique et bas	https://www.sie.com	Consulter Modifier

Documentation : <https://symfony.com/bundles/KnpMenuBundle/current/index.html>
https://symfony.com/bundles/KnpMenuBundle/current/menu_builder_service.html

Menu manuel HTML/Twig

Exemple :

```
{# templates/partials/menu.html.twig #}
<nav>
    <ul>
        <li><a href="{{ path('app_home') }}">Accueil</a></li>
        <li><a href="{{ path('app_products') }}">Produits</a></li>
        <li><a href="{{ path('app_contact') }}">Contact</a></li>
    </ul>
</nav>
```

Cette solution n'est pas forcément appropriée car il s'agit d'un menu **statique** donc la mise à jour **manuelle** est nécessaire (à chaque ajout/suppression d'items).

Cependant celle-ci peut être intéressante dans certains cas : dépannage / débogage, utilisation pour des sites et applications de « taille restreinte »...

EasyAdminBundle

Ce bundle permet de créer des menus, spécialement des menus d'administration. Il permet la génération automatique du menu basé sur les entités, et il aucune configuration manuelle n'est nécessaire pour les CRUD basiques

Cmde d'install. : `composer require easycorp/easyadmin-bundle`

📖 Documentation : <https://symfony.com/bundles/EasyAdminBundle/current/index.html>

➔ Créez le menu de votre application en utilisant l'une des solutions précédentes. Il faudra pouvoir justifier de la pertinence de votre choix (➔ notamment dans votre dossier de TP)

1.4 Formulaires

Le design des formulaires est plutôt libre, mais les directives suivantes sont à respecter :

- **Champs et éléments**

Les éléments de formulaire (champs, labels des champs...) doivent être tous de même style (➔ **Harmonisation, Cohérence**).

- **Champs obligatoires**

Faire apparaître (en CSS) un astérisque rouge * si le champ est obligatoire, comme ainsi :

Genre * ACTION

Genre * ACTION

- **Boutons**

Les boutons d'actions / de formulaire (Ajouter, supprimer, créer, retour arrière, ...) doivent être tous de même style (→ **Harmonisation, Cohérence**).

- **Gestion des erreurs et aide :**

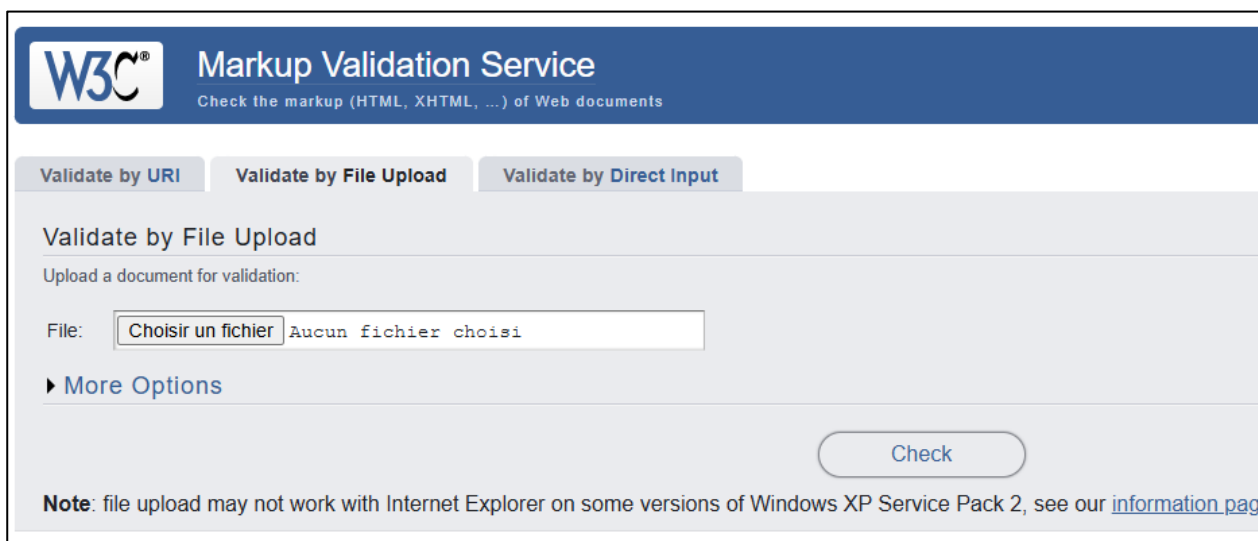
Des éléments doivent faciliter le remplissage et la validation par l'utilisateur : message d'aide, messages d'erreurs, ...

[TP 5] Annexe 1 : Ressources & outils

Validateur HTML du W3C

Le W3C (World Wide Web Consortium) est l'organisme international de normalisation qui développe les standards du Web. Il a été fondé en 1994 par Tim Berners-Lee, l'inventeur du World Wide Web. Il élabore des recommandations techniques qui garantissent l'interopérabilité et l'évolution cohérente du Web, comme HTML, CSS et les normes d'accessibilité. Son rôle est utile pour assurer que le Web reste ouvert, accessible et fonctionne de manière uniforme sur tous les navigateurs et appareils.

<https://jigsaw.w3.org/css-validator/>



The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by File Upload" tab is selected. Under this tab, it says "Upload a document for validation:". There is a "File:" label followed by a text input field containing "Choisir un fichier" and "Aucun fichier choisi". Below the input field is a link "More Options". At the bottom right of the form is a "Check" button. A note at the bottom states: "Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#)."

Validateur CSS du W3C

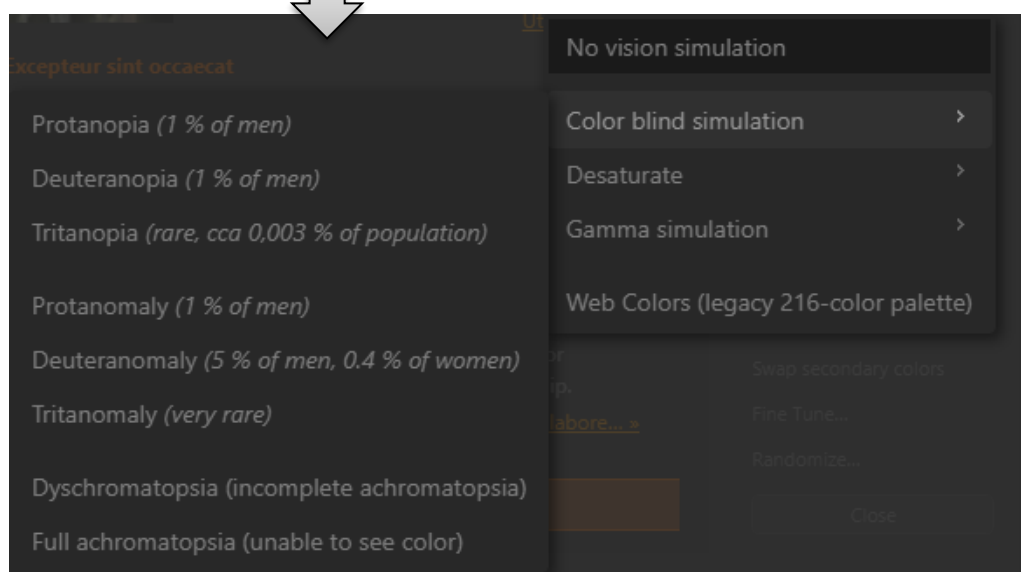
<https://jigsaw.w3.org/css-validator/>



The screenshot shows the W3C CSS Validation Service interface. At the top, there's a blue header with the W3C logo and the text "CSS Validation Service" and "Vérifiez vos feuilles de style CSS et vos documents HTML stylés avec CSS". Below the header, there are three tabs: "par adresse URI", "par chargement de Fichier", and "par Saisie Directe". The "par Saisie Directe" tab is selected. Under this tab, it says "Validation par Saisie Directe". There is a text input field with the placeholder text "Saisissez la feuille de style à valider (ATTENTION : feuilles de style CSS uniquement):".

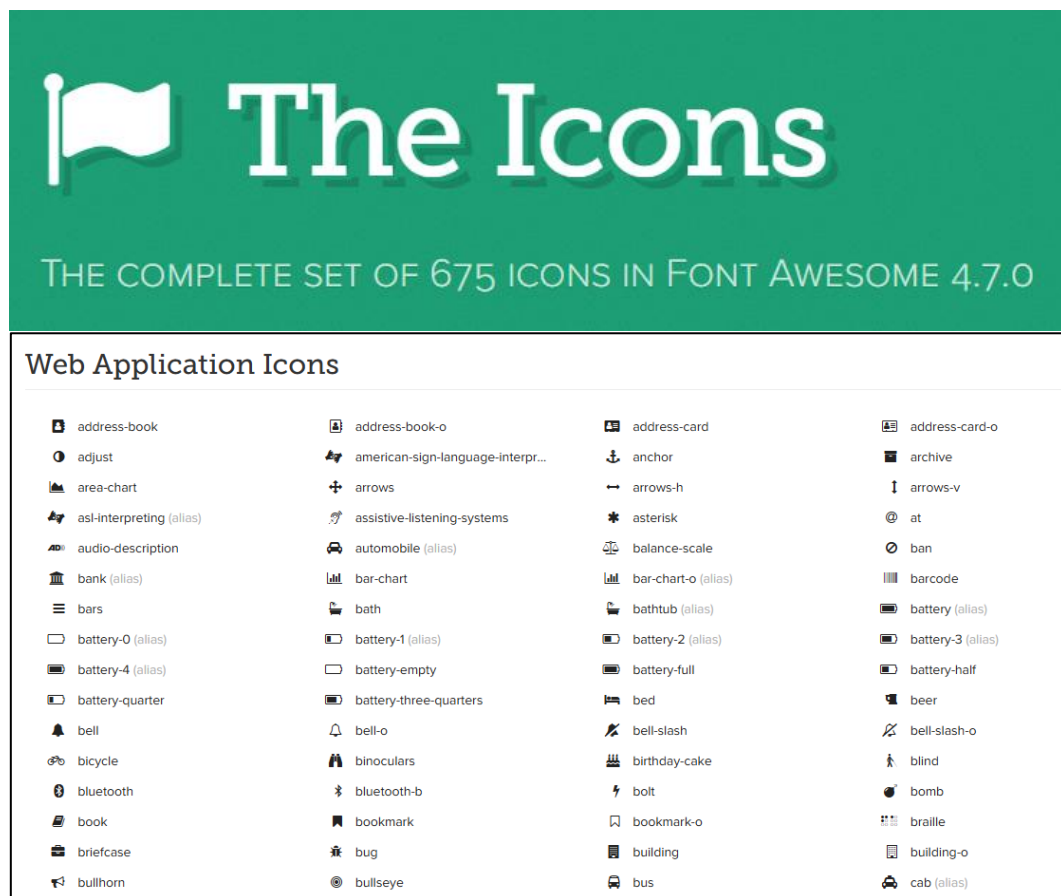
Paletton – Outil de choix de couleur

<https://paletton.com/>



Police d'icônes Fontawesome :

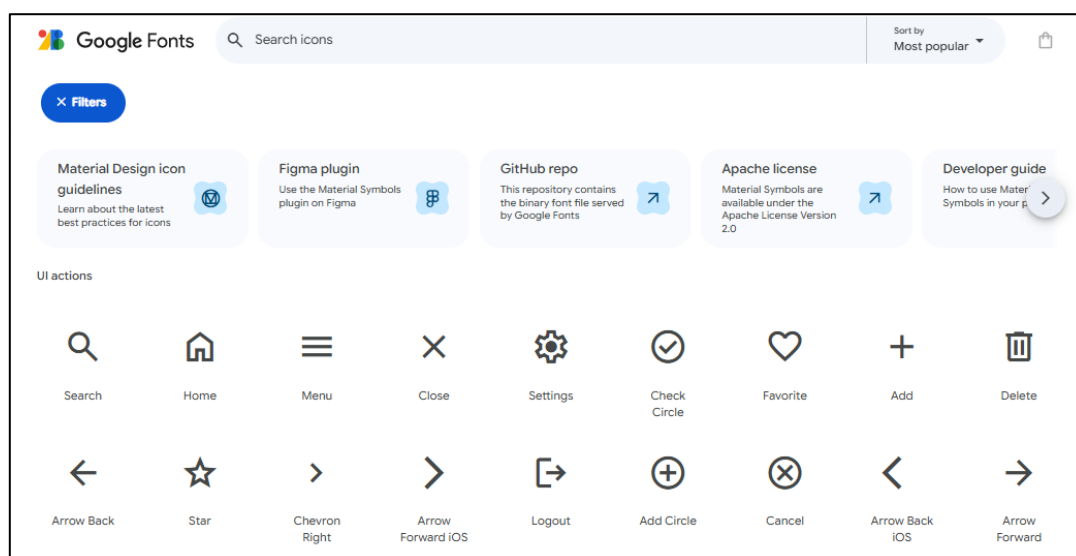
<https://fontawesome.com/v4/icons/>



Police d'icônes de Google (Material Symbols) :

<https://fonts.google.com/icons>

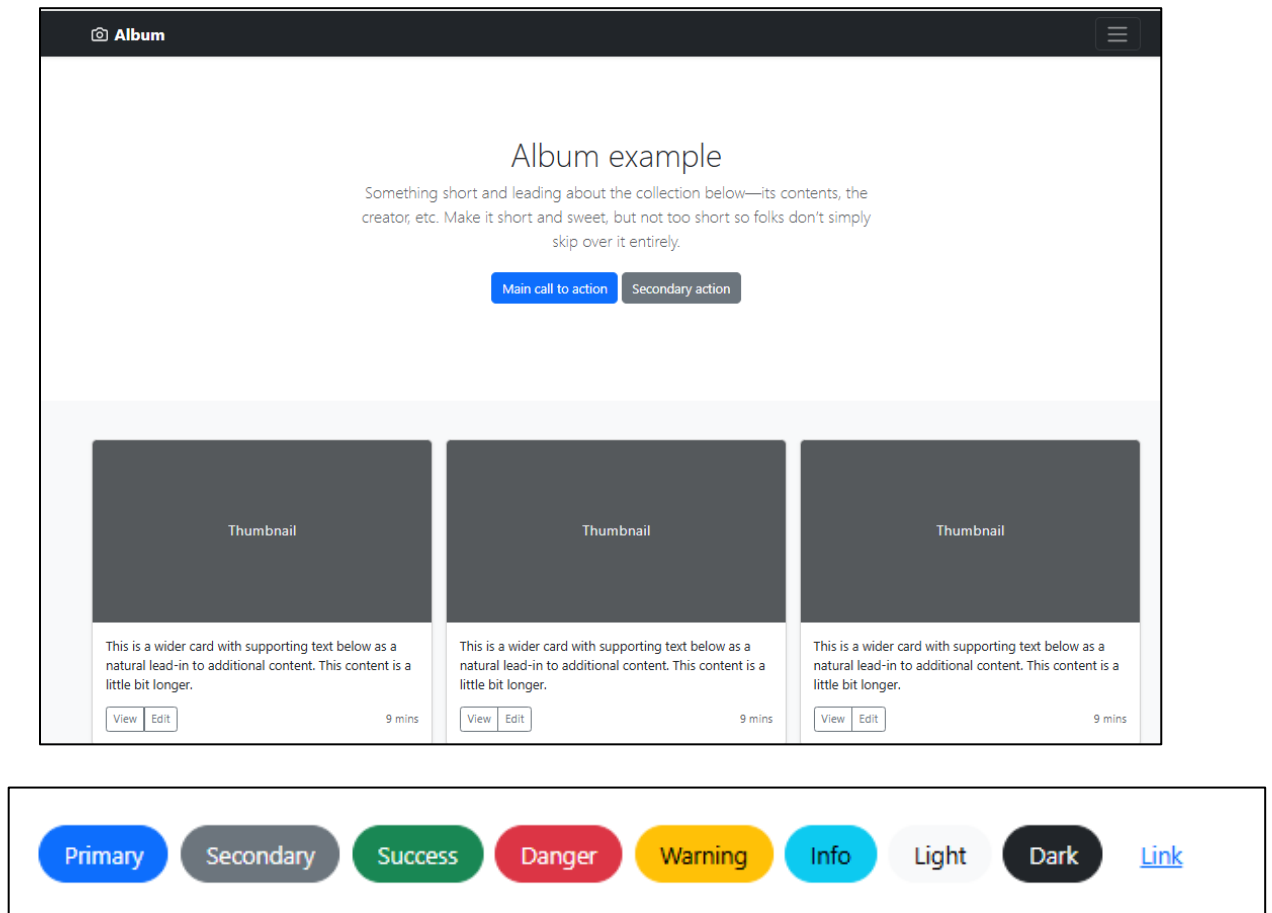
<https://m3.material.io/styles/icons/overview>



Framework CSS *Bootstrap*

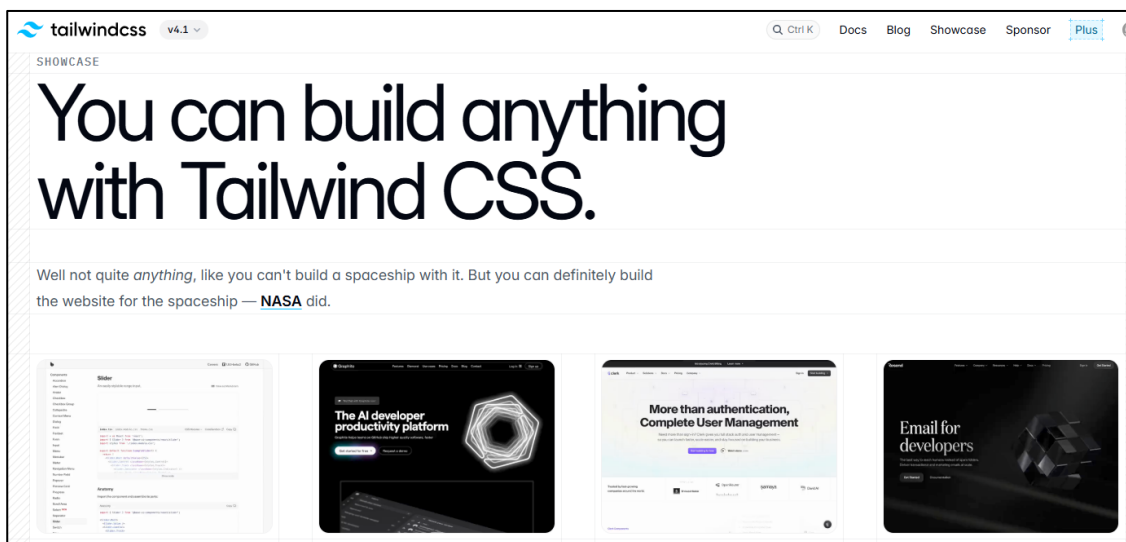
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

<https://getbootstrap.com/docs/5.3/examples/>



Framework CSS *Tailwind*

<https://tailwindcss.com/>

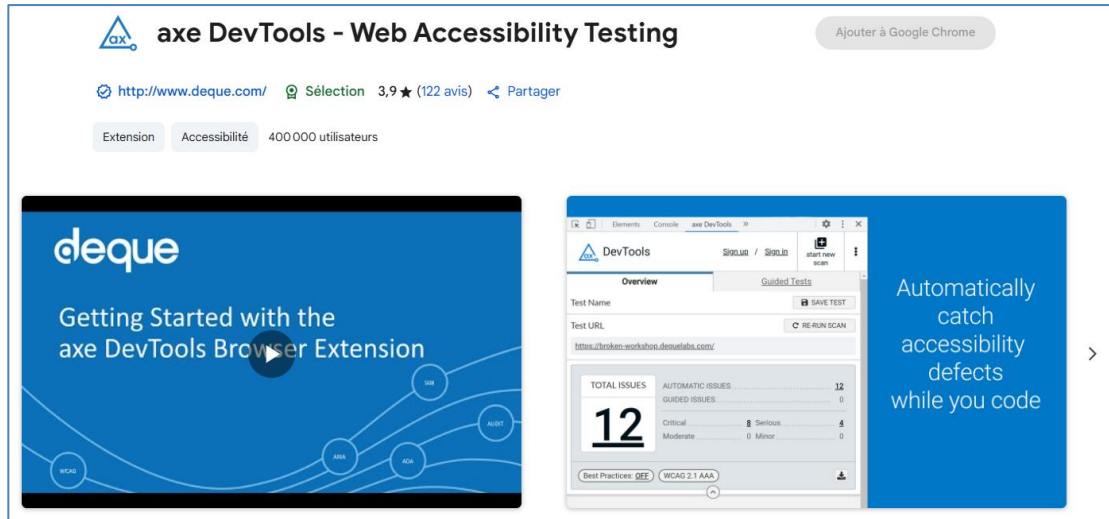


Outils de test d'accessibilité, ergonomie et performances

axe DevTools

Extension pour Chrome : <https://chromewebstore.google.com/detail/axe-devtools-web-accessib/lhdoppojpmngadmndneiefpokejbdd>

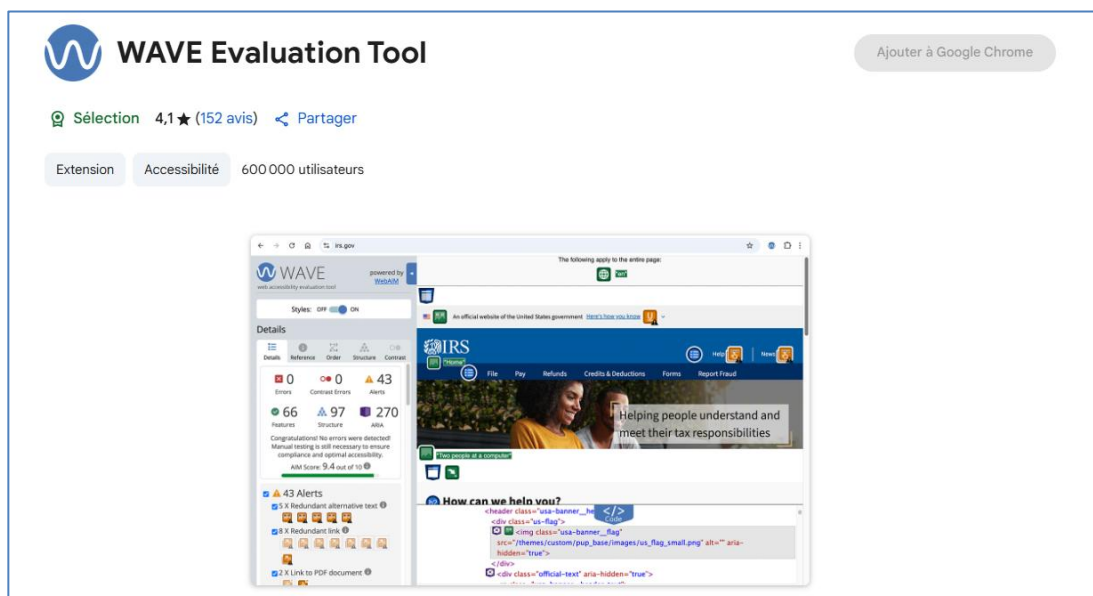
Extension pour Firefox : <https://addons.mozilla.org/fr/firefox/addon/axe-devtools/>



WAVE

Extension pour Chrome : <https://chromewebstore.google.com/detail/wave-evaluation-tool/jbbplnpkjmmebjpijfedlgcdilcofh>

Extension pour Firefox : <https://addons.mozilla.org/fr/firefox/addon/wave-accessibility-tool/>



OPQUAST

OPQUAST est une entreprise et une méthodologie française spécialisée dans la **qualité web**.

OPQUAST propose un **référentiel de bonnes pratiques**. Il s'agit d'un ensemble de règles (checklist) pour améliorer la qualité des sites web, couvrant de nombreux aspects tel que :

- **L'accessibilité**
- **La performance**
- **SEO (*Search Engine Optimization*)** (référencement « naturel »)
- **L'écoconception**
- **La sécurité**
- **Le respect de la vie privée**
- **L'ergonomie**

OPQUAST propose également une **certification professionnelle** (prix : 582€ TTC).

La certification "*OPQUAST Maîtrise de la qualité en projet web*" permet aux professionnels du web (développeurs, chefs de projet, designers, etc.) de valider leurs compétences en qualité web. Cette certification est maintenant plutôt reconnue dans le secteur et est parfois demandée dans les appels d'offres publics.

OPQUAST propose également des **formations** pour sensibiliser les équipes aux bonnes pratiques qualité web.

The screenshot shows the Opquast website interface. At the top, there is a navigation bar with a language selector (FR), links for FAQ, Contact, and Boutique, a search bar labeled 'Chercher dans Opquast', and a user profile icon. Below the navigation bar is the Opquast logo and a secondary navigation bar with links: Découvrir Opquast, Nos formations, Écosystème, and À propos. The main content area has a dark blue header with the breadcrumb 'Accueil → Rendre le Web meilleur → Référentiel Opquast - Qualité Numérique'. The title 'Référentiel Opquast - Qualité Numérique' is prominently displayed in white. Below the title, it states '245 règles pour améliorer vos sites et applications numériques - Version 5 - 2025-2030' and includes a list of hashtags: #qualitéweb #accessibilité #sécurité #performance #privacy #écoconception. At the bottom of the main content area, there are four white buttons: Découvrir, Télécharger, Glossaire, and Licence.

Remarque : Le coût de nombreuses formations et certifications est généralement un frein pour un grand nombre d'acteurs du numérique. Il s'agit en effet d'un business... et l'on peut regretter le peu de ressources gratuites, ou encore la reconnaissance de celles-ci sur le plan professionnel. Cependant, il est intéressant de connaître les offres et tendances, car beaucoup d'entre vous seront en mesure de demander des formations (et leurs prises en charge financières) durant votre carrière.