

# Technologies Web et XML

## TP1 - HTML & CSS

Polytech Nantes - Info 3

### Objectifs du TP

L'objectif de ce TP est de réaliser une page web statique "simple". Pour cela, nous vous proposons d'écrire le code HTML et réaliser la mise en page CSS de la page principale d'un site inspiré d'un site existant : celui de AllMusic (<http://www.allmusic.com/genre/pop-rock-ma0000002613>). Bien entendu, nous allons rebaptiser ce site PolyMusic...

Vous trouverez sur madoc une archive contenant les données nécessaires à la réalisation de ce TP. Celle-ci contient entre autres une image `PolyMusic.png` qui vous donne un aperçu de ce à quoi devra ressembler votre page web une fois finie.

Pour arriver à nos fins et réaliser cette page proprement nous allons procéder en plusieurs étapes, décrites dans la suite de ce document.

### Outils

Pour réaliser ce TP vous aurez besoin de:

- Un éditeur de texte au choix (gedit, sublime text, notepad++, etc.)
- Votre navigateur préféré. Seule contrainte, celui-ci doit proposer des outils d'inspection de votre page (cf. illustration ci-dessous)

Nous allons également utiliser trois outils pour simplifier le développement:

- nodejs qui permet d'utiliser javascript en dehors d'un navigateur. Vous ne vous en servirez pas directement, mais il est nécessaire pour les deux autres outils que l'on va utiliser. **Dans la version actuelles de ce TP nous vous demandons d'installer nodejs version 10.**
- npm (node package manager) nous permettra d'installer les outils de validation de votre code HTML et CSS ainsi qu'un outil permettant de recharger votre page HTML à chaque fichier HTML ou CSS modifié.
- gulp va permettre d'automatiser la réalisation des tâches décrites ci-dessus à chaque modification des fichiers HTML et CSS.

### Installation

- A la racine de votre projet (là où est stocké votre fichier `package.json`) lancez l'installation de tous les outils en tapant la commande suivante dans un terminal : `npm install`

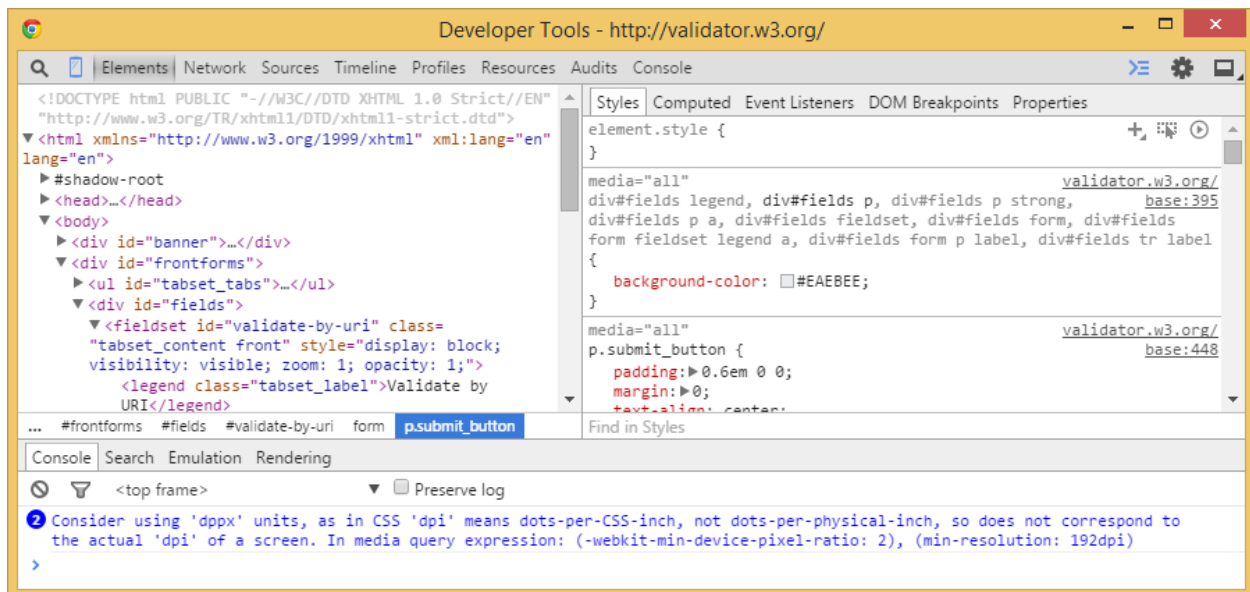


Figure 1: Inspecteur Web

- Maintenant que les outils sont installés vous pouvez lancer gulp. Comme on l'a installé localement, il faut le lancer localement via la commande : `nodejs node_modules/gulp-cli/bin/gulp.js` (si vous êtes en salle de TP sous ubuntu) ou `node node_modules/gulp-cli/bin/gulp.js`
- **Pour ceux qui travaillent sur leur PC:** Il est possible d'installer gulp globalement avec la commande `npm install -g gulp`. Cela permet d'utiliser ensuite gulp plus simplement avec la commande `gulp`.
- Cette commande devrait lancer votre navigateur par défaut (si ce n'est pas le cas, ouvrez l'adresse `http://127.0.0.1:3000` dans votre navigateur) et commencer à surveiller vos fichiers HTML et CSS. A chaque fois que vous effectuerez une modification votre navigateur sera rafraîchi et les validateurs HTML et CSS du w3c utilisés pour valider votre code (les résultats s'afficheront dans votre fenêtre de terminal).

## Le code HTML

Dans cette première partie du TP, le but du jeu est de réaliser un code HTML propre, le plus sémantique possible et respectant bien entendu la norme HTML5. L'image `PolyMusic-sansCSS.png` contenue dans l'archive du TP1 vous donne un aperçu du résultat final que nous devons obtenir. Ce n'est bien entendu pas très beau, mais nous améliorerons ça ensuite.

**Note:** nous allons procéder de manière incrémentale. A chaque étape, vous devrez vérifier que votre code HTML est valide via la vérification automatique effectuée avec gulp.

1. Ouvrez le document `index.html` fourni avec le TP. Celui-ci contient le code HTML minimal pour qu'il soit un document HTML5 valide. Le body de ce document contient pour l'instant le texte brut du site.
2. En regardant le texte contenu dans la balise `body` et le rendu final de la page, déterminez la structure globale celle-ci (entête, pied de page, navigation, sections, etc.).
3. Faites valider votre structure par les enseignants encadrant votre TP.

4. Mettez en place les balises structurantes de la page (entête, corps de la page, sections, pied de page, etc.) que vous avez définies en question 2 et validées en question 3. Vérifiez que votre code HTML est valide avant de passer à la suite.
5. Ajoutez toutes les autres balises manquantes (titres, liste, liens, formulaires, images, etc.), afin de générer un document HTML5 valide qui représente correctement les informations contenues dans la page. Les balise de lien `<a>` renverront sur une URL vide ("").
6. Pensez à l'accessibilité : n'oubliez pas les `label` dans votre formulaire de recherche et les attributs `alt` dans vos images.

**Remarque importante:** a priori à ce stade de l'écriture de votre page, votre document HTML ne doit contenir **aucune** `<div>`. Vous noterez également que le document produit est complètement lisible et compréhensible sans mise en forme (titre, image apportant du sens, etc.).

## La mise en forme CSS

Dans cette seconde partie, nous allons essayer de mettre en forme correctement notre page sans modifier la structure du code HTML. Les seules modifications autorisées sont l'ajout d'identifiants `id`, d'attributs de classe `class` et une seule `<div id="container">` englobant tout le contenu du `body`.

C'est bien entendu un exercice de style : dans la vraie vie on s'autorisera à modifier (un peu) le HTML afin de simplifier la mise en page. L'objectif est cependant de vous montrer qu'il est possible de faire sa mise en page purement en CSS, sans ajout d'un grand nombre de balises `div` (souvent inutiles dans le HTML).

La page une fois mise en forme devra ressembler (au plus proche) à l'image `PolyMusic.png`.

**Remarque :** Les instructions données dans la suite de ce TP effectuent la mise en page sans utiliser les nouvelles propriété CSS3 `flex`, `flow` et `grid`. Vous êtes bien entendus libres de les utiliser si vous êtes plus à l'aise avec elles (ou souhaitez expérimenter).

## Mise en page générale

1. Créez les règles CSS des éléments `body` et du `<div id="container">` afin que:
  - le corps du document ait un fond blanc cassé (`#f6f6f6`), une largeur minimum de 480 pixels et utilise les polices suivantes (par ordre de préférence) `"Helvetica Neue", Helvetica, Arial, sans-serif`;
  - le `<div id="container">` soit large de 980 pixels et soit centré dans la page à l'aide de marges horizontales automatiques (`margin: 0 auto;`). Le contenu de votre page sera donc toujours centré horizontalement.
2. Vous pourrez ajouter quelques règle générale un peu plus tard afin de mettre en forme certains éléments communs aux différentes sections de la page (ex: `h2`, `p`, sections, etc.).

## Le header

1. Le titre `h1` de la page est pour l'instant représenté sous forme de texte. Nous allons le remplacer par une image. Pour cela, donnez à la propriété `text-indent` la valeur de 100%, à `white-space` la valeur `nowrap` et à `overflow` la valeur `hidden` afin de faire sortir disparaître de la fenêtre de votre navigateur. Appliquez ensuite une image de fond (`background-image`) pour remplacer le texte. N'oubliez pas de donner une largeur une hauteur à votre `h1`.

2. L'entête de la page doit faire 80px de hauteur, et avoir une couleur de fond noir. Cette couleur de fond doit recouvrir toute la largeur du body mais le contenu de l'entête doit lui rester dans les 980px de son conteneur `<div id="container">`. Ces deux contraintes sont a priori incompatibles. On va donc insérer un bloc de contenu grâce au pseudo sélecteur `::before` appliqué sur la balise `body`. Ce nouveau contenu sera de type `display: block` et positionné en `absolute` (ce qui implique que le `<div id="container">` soit positionné en `relative`). Il aura la même hauteur de l'entête (80px). Ne pas oublier d'ajouter un attribut `content: ""` à ce bloc afin que le navigateur l'affiche (sans attribut `content`, la pseudo classe `::before` est ignorée).
3. Utilisez la même technique pour insérer un bloc gris (#505050) d'un pixel de haut en dessous du titre pour faire office de séparation avec le reste.
4. Utilisez la propriété `display: inline-block` afin de mettre tous les éléments li de vos menus les uns derrière les autres.
5. Utilisez la technique du `text-indent` vue au dessus afin de remplacer le texte des liens vers les réseaux sociaux par des images provenant de l'image `social-header.png`. Cette image contient les logos de tous les réseaux sociaux (technique dite du 'sprite'). Utilisez la propriété `background-position` ainsi que les informations contenues dans le fichier `sprite.txt` afin d'insérer toutes les images.
6. Positionnez les différents éléments du header (le plus simple est de le faire en `absolute`) et ajuster la couleur du texte et la typographie (gras, taille des polices, mise en forme des liens, etc.).

## Les sections / aside

1. La mise en page des différentes sections n'est pas très complexe. Il faut cependant choisir une technique pour placer correctement la colonne de droite ('Recent Pop/Rock Releases') par rapport aux autres. Celle-ci fera 240px de large, sera positionnée en `absolute`, et collée au bord droit de son conteneur.
2. Pas la peine de préciser la largeur des autres sections, elle prendront l'espace qu'il leur reste dans le `<div id='container'>`.
3. La mise en page des informations des groupes / artistes 'Pop/Rock Artists Highlights' est un peu plus subtile.
  - Il faut positionner les artistes en `inline-block` afin qu'ils se placent les uns à côté des autres.
  - Vous devez également masquer (`display: none`) les informations concernant la période d'existence du groupe ainsi que sa description.
  - Le nom du groupe / artiste doit être positionné en bas de l'image d'illustration (en `absolute`) en blanc sur fond noir.
  - La description du groupe (ainsi que sa période d'existence) doivent s'afficher au survol du nom du groupe. Pour cela il faudra combiner l'utilisation de `:hover` et le sélecteur de voisin `~` !
4. Les liens sont bleu #0089c1 et non soulignés.
5. Effectuez le reste de la mise en page des différentes sections. Il n'y a priori pas de difficulté majeure

## Pop/Rock Artists Highlights

---



Figure 2: Survol groupe/artiste

### Le footer

10. Mettez en page le pied de page. Vous devriez maintenant pouvoir vous débrouiller pour décider de la bonne méthode de positionnement à utiliser. Inspirez vous de la mise en page du header, il vous faudra ici utiliser le pseudo élément `::after`.
11. Si ça n'a pas encore été fait, vérifiez la validité de votre CSS via la sortie de votre terminal. Dans la mesure du possible, testez également votre page HTML sur différents navigateurs afin de vérifier l'homogénéité du rendu.

### Responsive design (bonus)

Ajoutez une media query à votre CSS afin que la mise en page soit "responsive" et adopte la mise en page illustrée par l'image `PolyMusic-small.png` si la largeur de l'écran est inférieure à 880 pixels.

Pour information, la correction officielle n'utilise que 6 règles additionnelles dans cette media query.