Travail de groupe - Composants (TQ INFO)

Dans ce mini-projet, tu vas créer un mini-site web en **HTML/CSS** pour expliquer un composant informatique de façon claire, structurée et un peu technique, afin qu'un élève qui "s'y connaît un peu" comprenne réellement son rôle, ses critères de choix et les différences entre modèles.

Création d'un mini-site d'information sur les composants et périphériques informatiques

Pour ce travail, vous allez réaliser **ensemble un site web complet** consacré à différents composants et périphériques informatiques. Chaque élève est responsable d'un composant précis, mais l'ensemble doit former **un seul vrai site cohérent**, avec une identité visuelle commune, une navigation claire et une structure identique.

Le but n'est pas seulement de "faire des pages HTML", mais de produire un site **structuré, lisible, illustré, techniquement propre**, qui montre à la fois :

- vos compétences en **HTML / CSS**
- votre capacité à **comprendre, vulgariser et présenter un sujet technique**

Finalité du projet

À la fin du projet, le site doit permettre à un visiteur de :

- naviguer facilement entre les composants
- comprendre le rôle et le fonctionnement de chaque composant
- découvrir les critères techniques importants
- comparer plusieurs modèles réels
- identifier les grandes marques et les grandes familles de produits
- consulter un site moderne, cohérent et agréable à lire

Objectifs pédagogiques

Objectifs en HTML / CSS

Tu dois montrer que tu es capable de :

- structurer correctement un site avec plusieurs pages HTML
- utiliser des balises adaptées : `header`, `nav`, `main`, `section`, `article`, `footer`, listes, tableaux, images…
- créer une navigation claire
- séparer le contenu, le style et les données
- produire un code propre, indenté, lisible et commenté si nécessaire

- créer une interface moderne avec un style cohérent

Objectifs sur le composant étudié

Tu dois montrer que tu es capable de :

- expliquer clairement à quoi sert le composant
- décrire son fonctionnement de manière vulgarisée mais sérieuse
- présenter les notions techniques importantes
- relier la théorie à des usages concrets
- présenter les marques, gammes et modèles importants
- comparer intelligemment plusieurs modèles

Travail demandé

Chaque élève reçoit **un composant ou périphérique précis**.

Exemples :

- CPU
- RAM
- carte réseau
- LAN
- modem / routeur / Internet
- système d'exploitation
- carte son
- imprimante
- clavier
- souris / trackball
- tablette graphique
- périphérique MIDI
- etc.

Chaque élève doit créer la partie du site correspondant à son sujet, **dans son propre dossier**, tout en respectant la structure commune du projet.

Contraintes générales du site

Le site complet doit respecter les contraintes suivantes :

- plusieurs pages HTML
- structure cohérente et commune à tous
- menu de navigation placé **à gauche**, dans un style type **dashboard**
- design moderne
- thème **dark mode**
- couleurs visibles et harmonieuses
- utilisation de **dégradés**, accents colorés, cartes, blocs, encadrés
- site **amplement illustré**
- contenu lisible, hiérarchisé, aéré
- pas de base de données MySQL

- les données répétitives ou comparatives doivent être séparées du HTML et chargées depuis des fichiers de données

Organisation générale du dépôt GitHub

Un **repo GitHub commun** sera utilisé pour toute la classe.

Chaque élève travaille dans son propre dossier composant.

Exemple d'arborescence possible :

```
  /site-composants/
│
├── index.html
├── composants.html
├── a-propos.html
├── README.md
│
├── assets/
│   ├── css/
│   │   ├── common.css
│   │   ├── layout.css
│   │   ├── theme.css
│   │   └── components.css
│   │
│   ├── js/
│   │   ├── main.js
│   │   ├── navigation.js
│   │   └── data-loader.js
│   │
│   ├── img/
│   │   ├── global/
│   │   └── composants/
│   │
│   └── data/
│       ├── cles-usb.json
│       ├── imprimantes.json
│       ├── claviers.json
│       └── ...
│
├── composants/
│   ├── cpu/
│   │   ├── index.html
│   │   ├── fonctionnement.html
│   │   ├── criteres.html
│   │   ├── marche.html
│   │   ├── comparatif.html
│   │   ├── style.css
│   │   └── images/
│   │
│   ├── imprimantes/
│   ├── claviers/
```

```
42 │   │   ├── souris/
43 │   │   └── ...
```

Structure obligatoire pour chaque composant

Chaque composant doit disposer de son **propre mini-site**, dans son dossier, avec **minimum 4 à 5 pages HTML**.

Structure recommandée :

- `index.html` → présentation générale
- `fonctionnement.html` → comment ça fonctionne
- `criteres.html` → critères techniques importants
- `marche.html` → constructeurs, gammes, évolution
- `comparatif.html` → comparaison de modèles

Selon le composant, une page plus spécifique peut remplacer ou compléter une autre :

- `technologies.html`
- `types.html`
- `usages.html`
- `ergonomie.html`
- etc.

La structure peut être légèrement adaptée **si le composant le justifie**, mais elle doit rester proche du modèle commun.

Structure de page attendue

Chaque page doit avoir une structure claire et similaire.

Exemple attendu :

- en-tête du site
- menu latéral gauche
- zone principale de contenu
- titre clair
- sections avec sous-titres
- images légendées
- encadrés ou blocs d'information
- pied de page

Charte graphique commune

L'ensemble du site doit donner l'impression d'un **même projet**, pas d'un collage de pages indépendantes.

Style commun obligatoire

Le style global doit être commun :

- fond sombre
- texte clair
- couleurs d'accent visibles
- design moderne
- menu latéral fixe ou stable
- cartes / blocs avec coins arrondis
- ombres légères
- dégradés
- effets simples au survol
- typographie cohérente

Styles spécifiques autorisés

Chaque composant peut avoir **quelques adaptations visuelles** si cela apporte du sens :

- une couleur dominante spécifique
- quelques icônes adaptées
- une bannière spécifique
- un schéma personnalisé

Mais l'ensemble doit toujours rester cohérent avec le site global.

Menu de navigation

Le site doit utiliser une **navigation à gauche**, de type dashboard.

Le menu doit permettre d'accéder :

- à la page d'accueil générale du projet
- aux différents composants
- aux pages du composant en cours
- éventuellement à une page "à propos", "sources", "crédits", etc.

Le menu doit être :

- lisible
- bien organisé
- identique sur tout le site
- facile à maintenir

Illustrations obligatoires

Le site doit être **amplement illustré**.

Chaque élève doit intégrer plusieurs éléments visuels pertinents :

- photos du composant

- schémas explicatifs
- captures d’écran si utiles
- illustrations comparatives
- tableaux
- pictogrammes / icônes

Les images ne doivent pas être décoratives uniquement. Elles doivent aider à comprendre.

Obligations minimales

Pour chaque composant :

- au moins 1 schéma explicatif
- plusieurs images de produits ou technologies
- au moins 1 tableau comparatif
- légendes ou explications lorsque nécessaire

Gestion des données : HTML séparé des listes de données

Tout ce qui ressemble à une **liste structurée** ne doit pas être écrit “en dur” directement dans le HTML si cela peut être évité.

Cela concerne par exemple :

- comparatifs de modèles
- listes de marques
- listes de gammes
- tableaux techniques
- fiches produits
- cartes de présentation répétitives

Ces données doivent être stockées dans :

- des fichiers **JSON**
- ou éventuellement **SQLite**
- mais **pas MySQL**

Pour ce projet, le plus simple et le plus recommandé est **JSON**.

Pourquoi utiliser JSON ?

Cela permet :

- de séparer le contenu des données
- de rendre le site plus propre
- de faciliter la maintenance
- de générer automatiquement des tableaux ou cartes
- de montrer une logique plus moderne de développement

Exemple de données JSON avec les clés USB

Voici un exemple simple.

Fichier `cles-usb.json`

```json
[
  {
    "marque": "SanDisk",
    "modele": "Ultra Flair 128 Go",
    "capacite_go": 128,
    "interface": "USB 3.0",
    "debit_lecture": "150 Mo/s",
    "type_usage": "usage général",
    "prix": 14.99
  },
  {
    "marque": "Kingston",
    "modele": "DataTraveler Exodia 64 Go",
    "capacite_go": 64,
    "interface": "USB 3.2 Gen 1",
    "debit_lecture": "100 Mo/s",
    "type_usage": "transport de fichiers",
    "prix": 7.99
  },
  {
    "marque": "Samsung",
    "modele": "Bar Plus 256 Go",
    "capacite_go": 256,
    "interface": "USB 3.1",
    "debit_lecture": "300 Mo/s",
    "type_usage": "stockage rapide",
    "prix": 29.99
  }
]
```

Exemple d’idée d’exploitation

Ce fichier peut servir à générer :

- un tableau HTML
- des cartes produits
- un comparatif
- une liste filtrée ou triée

Exemple de logique JavaScript :

- lire le fichier JSON
- parcourir les objets
- construire dynamiquement le HTML

Ce qui doit être généré à partir des données

Pour chaque composant, il est fortement conseillé de générer dynamiquement :

- le tableau comparatif
- la liste des marques
- la liste des modèles
- les fiches techniques
- éventuellement les cartes de présentation

Contenu technique attendu

Le contenu doit aller **au-delà d'une simple présentation grand public**, mais rester accessible.

On attend :

- une définition claire
- le rôle du composant
- une explication de son fonctionnement
- les notions techniques essentielles
- des usages concrets
- des critères de choix
- des erreurs fréquentes ou idées reçues
- un aperçu du marché
- un comparatif de modèles réels

On n'attend pas :

- du copier-coller de fiches marketing
- un texte trop technique sans explication
- une accumulation de chiffres sans interprétation

Comparatif obligatoire

Chaque composant doit inclure **au minimum 3 modèles comparés**.

On attend généralement :

- un modèle entrée de gamme
- un modèle milieu de gamme
- un modèle plus avancé ou spécialisé

Le comparatif doit comporter :

- un tableau structuré
- des critères techniques
- un prix approximatif
- une analyse rédigée
- un avis personnel argumenté

Qualité rédactionnelle attendue

Le texte doit être :

- clair
- structuré
- reformulé avec vos propres mots
- techniquement correct
- lisible pour un élève ou un visiteur intéressé

Il faut :

- éviter les blocs de texte trop longs
- utiliser des sous-titres
- mettre en valeur les mots importants
- faire des liens entre technique et usage concret

Répartition du travail dans le groupe

Même si chaque élève a son composant, certaines parties doivent être pensées collectivement.

Travail collectif

Le groupe doit se mettre d'accord sur :

- l'arborescence générale
- la charte graphique
- la structure des pages
- le menu latéral
- les fichiers CSS communs
- les conventions de nommage
- la structure des fichiers JSON
- les règles GitHub

Travail individuel

Chaque élève prend en charge :

- la recherche sur son composant
- la rédaction de ses pages
- le choix des images
- la création de ses données JSON
- la génération de ses comparatifs
- le style spécifique éventuel de son composant

Consignes GitHub

Le travail se fait dans un **repo commun sur GitHub**.

Ce qui est attendu

- chacun travaille proprement dans son dossier
- commits réguliers
- messages de commit clairs
- pas d'écrasement du travail des autres
- synchronisation correcte avec le dépôt

Bonnes pratiques recommandées

- faire des commits fréquents
- utiliser des messages explicites Exemples :
  - `ajout structure html composant imprimantes`
  - `ajout json comparatif claviers`
  - `amélioration css menu dashboard`
- éviter les commits énormes en fin de projet
- tester avant de pousser

Selon le niveau du groupe, on peut demander en plus

- branches par élève ou par composant
- pull requests
- revue rapide entre élèves

Qualité technique attendue

Le code doit être :

- indenté
- lisible
- organisé
- sans répétitions inutiles
- avec des noms de fichiers cohérents
- avec des classes CSS compréhensibles

Exemples de bonnes pratiques :

- `common.css` pour le style global
- `style.css` propre au composant
- `data/imprimantes.json`
- `composants/claviers/comparatif.html`

Ce qu'il faut ajouter en plus

Voici ce qu'il est très pertinent d'ajouter au cahier des charges.

Une page d’accueil générale

Le site complet doit avoir une vraie page d’accueil présentant :

- le projet
- ses objectifs
- les composants traités
- l’organisation du site
- un accès rapide aux différentes rubriques

Une page crédits / sources

Chaque groupe ou élève doit indiquer :

- ses sources d’information
- les sources des images
- les crédits éventuels
- les outils utilisés

Cela permet :

- de vérifier le sérieux du travail
- de valoriser la recherche
- d’éviter le copier-coller non assumé

Une convention commune pour les données

Le groupe doit définir au départ :

- les noms de clés JSON
- la structure des objets
- les unités utilisées
- la manière d’écrire les prix
- les noms de champs techniques

Exemple :

- `prix`
- `marque`
- `modele`
- `type`
- `usage`
- `connectivite`

Cela évite le chaos du type :

- `price` dans un fichier
- `prixEuro` dans un autre
- `cost` ailleurs

Une cohérence sur les images

Le groupe doit fixer des règles :

- dossier d'images par composant
- noms de fichiers propres
- format conseillé (`jpg`, `png`, `webp`)
- taille raisonnable
- pas d'images énormes inutiles
- légendes si nécessaire

Une validation technique minimale

Avant la remise, le site doit être vérifié :

- tous les liens fonctionnent
- toutes les images s'affichent
- le menu fonctionne partout
- aucun fichier manquant
- JSON correctement lu
- pages lisibles
- pas d'erreurs HTML évidentes

Une homogénéité des pages

Il faut définir un gabarit commun :

- même placement du titre
- même style de menu
- même style de cartes
- même style de tableaux
- même pied de page
- même logique de navigation

Évaluation équilibrée

Répartition proposée :

HTML / CSS / structure du site : 70 %

On évalue notamment :

- structure HTML
- navigation

- qualité du CSS
- cohérence visuelle
- intégration d'images
- organisation des fichiers
- utilisation des données JSON
- qualité générale du site

Contenu technique sur le composant : 30 %

On évalue notamment :

- justesse des explications
- clarté
- richesse des notions
- lien avec les usages
- qualité du comparatif
- capacité à vulgariser

Ce qui fera la différence

Un bon travail ne sera pas seulement "joli".

Un très bon travail sera :

- cohérent
- lisible
- techniquement propre
- bien illustré
- intéressant à parcourir
- intelligent dans sa manière de présenter les données
- sérieux dans le contenu
- agréable visuellement

Résultat final attendu

À la fin, vous devez remettre :

- le repo GitHub complet
- un site fonctionnel
- une structure claire
- des dossiers bien organisés
- des composants répartis par élève
- des pages reliées entre elles
- des comparatifs générés à partir de JSON
- un design commun moderne en dark mode
- un travail collectif cohérent

En résumé

Votre mission est de produire **un vrai mini-portail web sur les composants informatiques**, pas une simple série de pages séparées.

On doit retrouver :

- une structure commune
- une navigation claire
- un design moderne
- du contenu sérieux
- des illustrations
- des données externes bien exploitées
- un vrai travail d'équipe