

Activité 2 - 2

Traiter l'information numérique

Volet connecté

Problématique : Après vérification des fonctions du produit, l'installateur devra mettre en fonctionnement le produit en mode automatique et manuel soit in-situ soit à distance par l'intermédiaire d'IHM : une page web.

1. Présentation

1.1. Généralités

Les objectifs :

- Tester les capteurs et les actionneurs du produit.
- Tester le fonctionnement automatique.
- Tester le fonctionnement à distance.
- Tester le réglage du capteur de luminosité.



Une IHM pour tester le produit :

Comment développer une page web en html pour piloter le produit ?

Première page Web

HTML



1 Squelette d'une page Web

1.1 Une page minimale en HTML

Le langage permettant d'écrire le contenu d'une page web est le HTML (*HyperText Markup Language* ou langage de balisage d'hypertexte). Voici le code d'une page Web minimale en HTML :

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Titre de la page</title>
</head>
<body>
<h1>Premier titre de paragraphe</h1>
<p>Voici le texte du premier paragraphe</p>
</body>
</html>
```



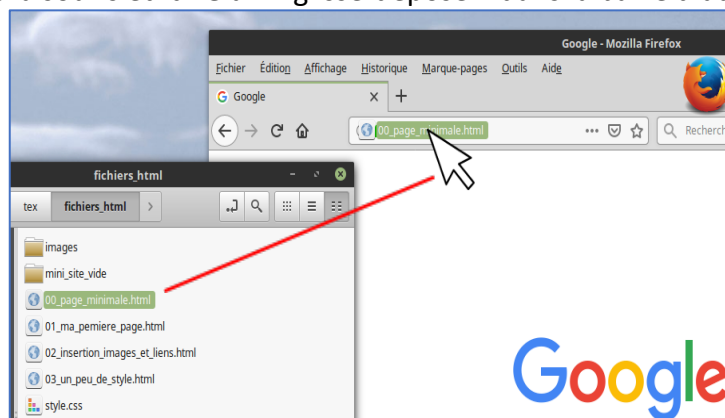
Lors de son interprétation par un navigateur, nous obtenons :

Exercice 1 : afficher une page Web dans le navigateur

Copier le répertoire « fichier_html » dans votre répertoire de travail.

Celui-ci contient les fichiers HTML du T.P., **double-cliquer** sur le fichier *00_page_minimale.html*

OU attraper le avec la souris et **faire** un « glissé-déposé » dans la barre d'adresse du navigateur.



1.2 HTML : un langage à balises

Comme nous pouvons le constater dans le code ci-dessus, le HTML est un **langage à balises**.

`<>` : balise ouvrante,

`</>` : balise fermante.

Les différents éléments constitutifs de la page sont insérés dans des balises permettant de déterminer la structure de la page et le rendu graphique à appliquer à chaque élément.

Par exemple, une portion de texte à afficher en gras sera ainsi notée :

**** Ma portion de texte en gras ****

Dans la pratique, nous écrivons toujours nos pages web à partir d'un fichier "minimal" fourni dans le répertoire *fichiers html*. Il n'est donc pas nécessaire de connaître par cœur tout le code présenté ici. Il est par contre indispensable d'en comprendre les principaux éléments pour pouvoir le modifier. Par exemple :

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre de la page</title>
  </head>

  <body>
    <h1>Premier titre de paragraphe</h1>
    <p>Voici le texte du premier paragraphe</p>
  </body>
</html>

```

Titre de la page

Premier titre de paragraphe

Voici le texte du premier paragraphe

<body>

Détaillons la structure de cette page et les principales balises :

- ☐ Le contenu de notre page est délimité par les balises `<html>` et `</html>`, viennent ensuite **deux sous-parties** dans le code :
 - ☐ `<head>` ... `</head>` : spécifie certaines informations nécessaires et permet de définir le titre de la page entre les balises : `<title>` et `</title>`.
 - ☐ `<body>` ... `</body>` : définit la partie de la page qui sera affichée dans le navigateur.
 - La balise de titre `<h1>` ... `</h1>` : définit un titre de paragraphe.
 - La balise paragraphe `<p>` ... `</p>` : délimite un paragraphe.

Ouvrir côte à côte sur le même écran (ou sur deux écrans) :

- ☐ un **éditeur de texte** pour écrire la page en HTML (Notepad++ au lycée),
- ☐ un **navigateur** pour visualiser le résultat.

Exercice 2 : écriture de notre première page Web sur le thème de la domotique :

Ressource : <https://www.domo-blog.fr/13-bonnes-raisons-de-passer-a-domotique/>

1. Afficher le fichier **01_ma_premiere_page.html** dans un explorateur type Chrome ou autre.

2. Ouvrir ce même fichier **01_ma_premiere_page.html** dans Visual studio code.

Dans les questions suivantes, les parties du code à modifier sont signalées dans le fichier par des "`< ! - A FAIRE ... ->`". Vous pouvez bien entendu effacer ces commentaires une fois le "`< ! - A FAIRE... ->`" réalisé.

Après chaque modification du fichier HTML dans l'éditeur, vérifier le résultat en actualisant l'onglet dans Firefox.

3. Choisir le titre de la page : `< ! - A FAIRE 1 ->`.

4. Choisir le titre du premier paragraphe : `< ! - A FAIRE 2 ->`.

5. Écrire un premier paragraphe : `< ! - A FAIRE 3 ->`.

6. Écrire un second titre de paragraphe et un second paragraphe : `< ! - A FAIRE 4 ->`.

2 Insertion d'images et de liens

2.1 Insertion d'images

De nos jours, presque toute page Web contient des images. Ces images sont fournies par le serveur au client lors des échanges par le protocole HTTP. Voyons comment les intégrer à une page Web.

Exercice 3 : intégrer une image dans une page Web

1. Copier et coller dans votre répertoire de travail le fichier *02_insertion_images_et_liens.html*. Une pratique indispensable dès qu'un site grossit consiste à placer toutes les images dans un même répertoire qui ne recevra que les images.

2. Créer un répertoire "images" dans le même répertoire que votre fichier html.

3. Choisir et télécharger une image de volet roulant à partir du Web et **placer** cette image dans le répertoire *images*.

4. Insérer cette image dans votre page Web : < ! - A FAIRE 1 ->, à l'aide de l'instruction :

```

```

Remarque : src signifie source.

Il est probable que l'image ne soit pas à la taille souhaitée.

5. Pour faire en sorte que l'image occupe 50% de la largeur disponible dans le navigateur client, écrire :

```

```

6. Centrer l'image horizontalement dans la page, ce que la balise <center> nous permet de faire :

```
<center></center>
```

7. Insérer maintenant une seconde image d'une télécommande de volet : < ! - A FAIRE 2 ->.

Remarque : la balise est une balise **orpheline** : elle s'ouvre et se ferme avec une seule balise.

2.2 Insertion de liens

Nous en venons ici à la balise la plus emblématique du Web, celle qui permet d'interconnecter les hypertextes numériques, donnant ainsi naissance à la gigantesque toile du Web : **le lien hypertexte !**

Très naturellement en somme, cette balise, au fondement du web, porte le nom : <a>.

Elle s'utilise ainsi :

```
<a href="http://www.adresse_du_lien">texte_du_lien</a>
```

Remarquons que cette balise se décompose en deux parties :

Dans la première partie, l'attribut href de la balise <a> permet de spécifier l'adresse vers laquelle pointe le lien. Nous pourrions retenir href comme "hypertext reference".

La seconde partie fournit le texte contenu dans la balise, sur lequel l'utilisateur pourra cliquer.

Exercice 4 : insérer un lien hypertexte dans une page Web

Reprendre le fichier *02_insertion_images_et_liens.html* de l'exercice précédent et l'ouvrir dans votre navigateur et dans Visual studio code

1. Insérer un lien hypertexte vers la page web du volet : < ! - A FAIRE 3 ->.

2. Insérer un second lien hypertexte vers la page manuel du volet : < ! - A FAIRE 4 ->.

Exercice 5 : image cliquable

Nous souhaiterions maintenant créer une "image - lien cliquable" : au lieu d'avoir un texte à cliquer pour suivre le lien, nous souhaiterions avoir une image à cliquer. En imbriquant judicieusement les balises et <a>, cela fonctionne très bien : < ! - A FAIRE 5 ->.

Un peu de style, avec CSS

1 Séparation de la forme et du fond

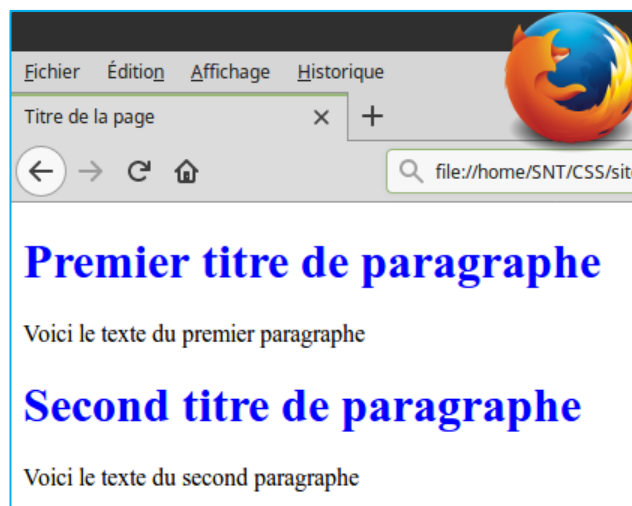
Vous l'avez remarqué, les pages écrites en HTML dans le chapitre précédent sont très rudimentaires, manquent de couleurs et adoptent une mise en page désastreuse. Au début du Web, dans les années 1990 - 2000, il était courant de voir des pages écrites de la façon suivante :

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre de la page</title>
  </head>
  <body>
    <font color="blue"><h1>Premier titre de paragraphe</h1></font> <p>Voici le texte du premier
    paragraphe</p>
    <font color="blue"><h1>Second titre de paragraphe</h1></font>
    <p>Voici le texte du second paragraphe</p>
  </body>
</html>
```

Ce qui produit le résultat ci-contre.

Si cela peut sembler faisable, bien qu'inconfortable, avec seulement deux titres à mettre en bleu, cela devient très compliqué avec un site comprenant des milliers ou plus de titres et de paragraphes.

D'autant que les modifications de couleurs sont loin d'être les seuls éléments à gérer pour obtenir une belle mise en page (taille des caractères, gras, italique, espacement entre les différents composants de la page . . .).



Cela aboutit très vite à un code illisible, impossible à maintenir et ré-exploiter, avec un mélange permanent :

- du fond, le texte à afficher dans notre page,
 - et de la forme : couleurs, tailles, espacements et autres aspects graphiques.
- Le langage CSS (*Cascading Style Sheets* ou feuilles de style en cascade) permet de remédier à ce problème :
- la description des éléments de la pages et du fond est décrite dans un fichier HTML,
 - les propriétés graphiques à leurs appliquer (la forme), est décrite dans un second fichier CSS.

Nous pourrions ainsi, en à peine une ligne de CSS, imposer à tous les titres compris dans des balises <h1>, de s'afficher en bleu. **L'économie de travail** devient majeure si l'on imagine un site comprenant des milliers, voire plus, de balises <h1>.

2 Première feuille de style CSS

Voici une feuille de style ultra-minimaliste :

```
h1{
  color: black;}
p{
  color: green; }
```



Une telle feuille de style va permettre, avec le fichier HTML associé, d'afficher :

- en noir, le contenu des balises `<h1>` (titres de niveau 1),
- en vert, le contenu des balises `<p>` (paragraphes).

Pour appliquer les propriétés définies dans une feuille **style.css**, à un fichier **03_un_peu_de_style.html** :

- **placer** le fichier *style.css* dans le même répertoire que le fichier *03_un_peu_de_style.html*,

Pour indiquer à *03_un_peu_de_style.html*, où se trouve le fichier de style :

```
<link rel="stylesheet" href="style.css" />
```

Nous pouvons maintenant, définir nos couleurs et propriétés graphiques, dans le fichier *style.css*.

Exercice 2 : modification d'une feuille de style

Nous travaillerons sur les deux fichiers *style.css* et *03_un_peu_de_style.html* que vous trouverez dans le répertoire précédent.

- **Ouvrir** ces deux fichiers dans Visual studio code
- **Ouvrir** le fichier html dans Firefox.
Vous avez dû observer que le rendu des couleurs n'est pas très esthétique.
- **Modifier** le fichier CSS de façon à faire apparaître les titres en bleu et les paragraphes en noir (ou encore aux couleurs de votre choix !).

Exercice 3 : des marges : margin !

Notre texte est pour l'instant beaucoup trop proche des bords de la fenêtre : nous aimerions le doter de marges (*margin* en anglais).

CSS permet d'appliquer des marges aux éléments de notre choix. Par exemple, pour obtenir des titres `<h1>`, de couleur bleu, avec des marges de 20 pixels :

```
h1{
  color: blue;
  margin: 20px; }
```

- **Modifier** le fichier CSS de l'exercice précédent, pour appliquer des marges de 20 pixels, aux titres de niveau h1 et aux paragraphes p.

Exercice 4 : ajouter une couleur de fond à la page

Pour obtenir un fond de couleur, nous devons, dans la feuille de style, définir la propriété background de la balise correspondant à l'intégralité de la page : body .

Par exemple :

```
body{
background: Cornsilk; }
```

- **Choisir** une couleur de fond à l'adresse suivante
https://en.wikipedia.org/wiki/Web_colors#X11_color_names.
- **Modifier** votre fichier CSS de façon à obtenir un fond coloré.

Exercice 5 : des images et des liens

- **Insérer** une image et un lien dans votre page.
- **Définir** les propriétés graphiques des balises et <a> dans la feuille de style pour obtenir un rendu harmonieux.
Nous aimerions que le lien devienne rouge lorsqu'il est survolé par la souris.
- **Définir** la propriété CSS suivante, et observer l'effet après avoir ré-actualisé l'onglet dans Firefox :

```
a:hover {
color: red; }
```

3 Mini - Projet, niveau Entraînement.

A partir des fichiers de votre **page Web** sur les volets **apporter** du « style » à votre page.