Technologies du web

Tutoriel : « Mettre en forme une page web »

Mots-clés : maquette graphique, gabarit, template, mise en page, règle CSS, sélecteur, déclaration, conteneur, identifiant, propriété, type bloc, type en ligne.



OBJECTIF

Présenter une page web avec des feuilles de style en cascade pour des applications de supervision et de contrôle de systèmes techniques.

CONDITIONS DE DÉROULEMENT DE L'ACTIVITÉ

Étapes	Objectifs pédagogiques	Activités pratiques
1 - Mise en situation	***Découvrir** les feuilles de style en cascade (CSS) et le cahier des charges de la page web à réaliser.*	***Lecture** de la mise en situation.*
2 – Découverte du CSS	***Concevoir** le gabarit d'un site web.*	***Écriture** de règles CSS.*
3 - Synthèse	***Concevoir** une page web conformément à un cahier des charges.*	***Rédaction** du contenu de la page domotique.* ***Écriture** de règles CSS.*

SIGLES UTILISES

Durée conseillée	: Important	: Information	Visual Studio Code

PRÉPARATION DU DOSSIER DE TRAVAIL

Organisez votre dossier de travail en suivant les indications données dans l'**annexe 1** située à la fin de ce document.

RESSOURCES DOCUMENTAIRES, LOGICIELS ET MATÉRIELS

PC, navigateur (**Mozilla**, Chrome, Edge, etc.), dossier du TP, liste des balises xHTML et des propriétés CSS2, éditeur de code.

Table des matières

1 Mise en situation 3
1.1. Page web dans le contexte industriel (Rappel) 3
1.2. Les feuilles de style en cascade 3
1.3. Le modèle client-serveur dans l'industrie 3
1.4. Fil conducteur du TP : conception d'une IHM pour une installation Domotique 4
1.5. Cahier des charges 4
2 Découverte du CSS 5
2.1 Comprendre le modèle de boîte 5
2.1.1 Le modèle de boîte 5
2.1.2 Calcul de la place occupée par une boîte 5
2.2 Concevoir le gabarit du site *Domotique* pas à pas 5
2.2.1 Organisation du projet *Domotique* 5
2.2.2 Étape 1 : afficher les blocs 5
2.2.3 Étape 2 : placer les blocs 7
2.2.4 Étape 3 : éloigner les blocs 8
2.2.5 Étape 4 : simuler le contenu textuel 8
2.2.6 Étape 5 : faire les finitions 9
3 Synthèse 10
3.1 Préparation 10
3.2 Conception de la page domotique 10
3.3 Amélioration du tableau des grandeurs physiques 10
4 Compléments : les balises de type bloc et en ligne 11
Annexe 1 : installation du dossier du TP dans votre compte sur le serveur Nas_SIN 12

Table des illustrations

Figure 1: Interface Homme Machine 3
Figure 2 : feuille de style (CSS) appliquée à un site web 3
Figure 3 : les domaines de la domotique 4
Figure 4 : maquette graphique d'un site web 4
Figure 5 : gabarit du site "Domotique" (fichiers template.html et style.css) 4
Figure 6 : page réalisée à partir du gabarit (fichiers domotique.html, style.css et tableau.css) 4
Figure 7 : accès aux dimensions d'une boîte avec la touche [F12] dans le navigateur 5
Figure 8 : analyse de <p> 5
Figure 9 : calcul de la place occupée par une boîte 5
Figure 10 : dossier de travail 5
Figure 11 : Domotique2 6
Figure 12 : template.html et style.css côte à côte 6
Figure 13 : maquette graphique du site "Domotique" 7
Figure 14 : les blocs sont positionnés 7
Figure 15 : logo a été éloigné de banner 8
Figure 16 : logo et banner n'ont plus assez de place à l'intérieur du bloc page 8
Figure 17 : les blocs sont positionnés et éloignés les uns des autres 8
Figure 18 : le gabarit du site domotique en bonne voie ! 9
Figure 19 : paint.net 9
Figure 20 : gabarit du site domotique 10
Figure 21 : page domotique 10

1 Mise en situation

[🕐 5mn]

1.1. Page web dans le contexte industriel (Rappel)

Une page web est une interface homme–machine (*IHM*) consultable à l'aide d'un navigateur internet. Dans le domaine industriel, elle peut servir à :

- **Superviser à distance** le fonctionnement de systèmes automatisés
- **Afficher en temps réel** des données issues de capteurs
- **Contrôler des équipements** grâce à des commandes accessibles en ligne
- **Configurer les paramètres** d'un système ou d'une installation

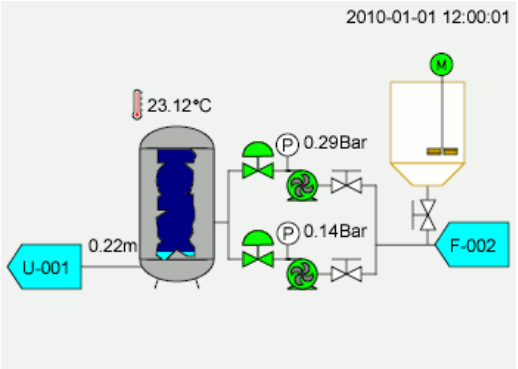


Figure 1: Interface Homme Machine

Vous avez découvert les balises HTML dans le tutoriel : « **Structurer une page web** ». Certaines d'entres-elles, comme ** ** et **** permettent de **diviser un document et de fixer la position des objets**.

Plus généralement, une page HTML est conçue en séparant sa **structure** de sa **présentation**.

1.2. Les feuilles de style en cascade

*« Le design d'un site évolue au fil du temps. Le **problème**, lorsqu'on n'utilise pas de feuilles de style, c'est qu'il faut reprendre toutes les pages HTML une à une pour modifier une police de caractère ou une couleur de fond...*

*Avec les "**Cascading Style Sheets" (CSS)**, ce lourd handicap est résolu. »*
developpez.com

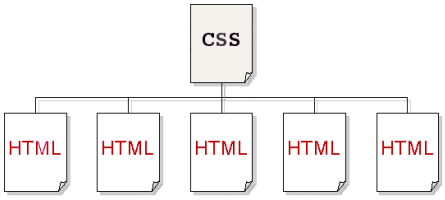


Figure 2 : feuille de style (CSS) appliquée à un site web



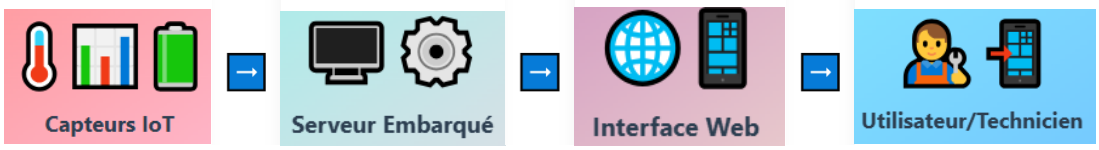
*« **CSS** (Cascading Style Sheets : **feuilles de style en cascade**) est un **langage informatique** qui sert à décrire la **présentation** des documents HTML.*

*L'un des objectifs majeurs des CSS est de permettre la **mise en forme** hors des documents. Il est par exemple possible de ne décrire que la **structure** d'un document en HTML, et de décrire toute la **présentation dans une ou plusieurs feuilles de style CSS** séparées.*

*Ainsi, les **avantages des feuilles de style** sont multiples :*
*- La structure du document et la présentation peuvent être gérées dans des **fichiers séparés**.*
*- La conception d'un document se fait dans un premier temps **sans se soucier de la présentation**, ce qui permet d'être plus **efficace**.*
*- Dans le cas d'un site web, la **présentation est uniformisée** : les documents (pages HTML) font référence aux mêmes feuilles de styles.*

Cette caractéristique permet de plus une remise en forme rapide de l'aspect visuel.
*- Le **code HTML** est considérablement **réduit en taille et en complexité,** puisqu'il ne contient plus de balises ni d'attributs de présentation. » (Wikipédia)*

1.3. Le modèle client-serveur dans l'industrie



Flux de données industriel :

1. Les **capteurs** collectent les mesures
2. Le **serveur embarqué** traite les données
3. Le **protocole HTTP** transmet les informations
4. L'**interface web** affiche les résultats

Consultez la fiche
« Chaîne IoT dans la spécialité STI2D-SIN »
sur https://edurl.fr/NPGptQYk

1.4. Fil conducteur du TP : conception d'une IHM pour une installation Domotique



Figure 3 : les domaines de la domotique

Ce tutoriel sur les **technologies du web** est le second d'une série en comprenant trois :
- Tutoriel 1 : **«** Structurer une page web »
- **Tutoriel 2 : « Mettre en forme une page web »**
- Tutoriel 3 : **«** Interaction client-serveur »

Au cours de cette série, vous allez réaliser un site web sur le thème de la **domotique.** Ce site présentera les grandeurs physiques mesurées dans la salle de classe.

Le site **« Domotique »** doit être composé de **deux pages** :
- Une page affiche les **grandeurs physiques** mesurées dans la salle de classe (température, humidité, pression et luminosité).
- Une page assure la **connexion** au site à partir d'un identifiant et d'un mot de passe.

Lors de la précédente activité (**Tutoriel : « Structurer une page web »**), vous avez implémenté la **structure** du **gabarit** du site "Domotique" dans le fichier ***template.html***.

Rappel : gabarit d'un site web

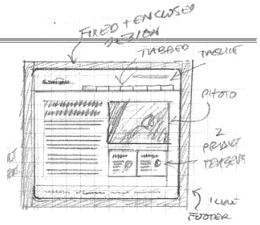


Figure 4 : maquette graphique d'un site web

Lors de la conception d'un site web, il est utile de disposer d'une **page unique**, un modèle ou **gabarit** qui sera utilisé par toutes les autres pages du site.
La conception de ce modèle unique :
- facilite les mises à jour ultérieures du site,
- facilite la navigation des visiteurs,
- assure une cohérence structurelle sur toutes les pages du site.

Avant d'écrire le code d'un gabarit, on élabore une **maquette graphique** du site au crayon ou avec des outils logiciels.
Le **gabarit**, aussi nommé template, est donc une **mise en page formatée**, qui sert de **modèle aux pages d'un site internet**.

1.5. Cahier des charges

Concevoir le gabarit du site "Domotique" représenté ci-dessous (Figure 5).

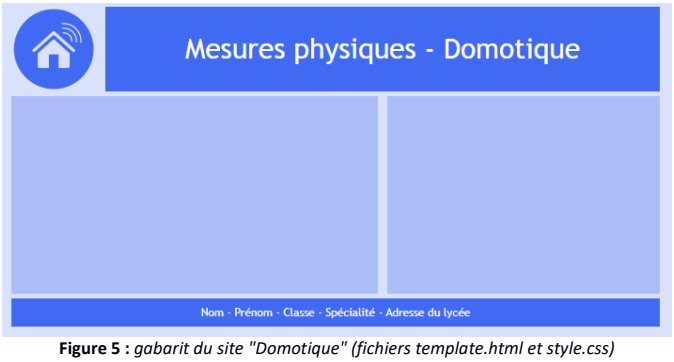


Figure 5 : *gabarit du site "Domotique" (fichiers template.html et style.css)*

Créer la page suivante à partir de ce gabarit.

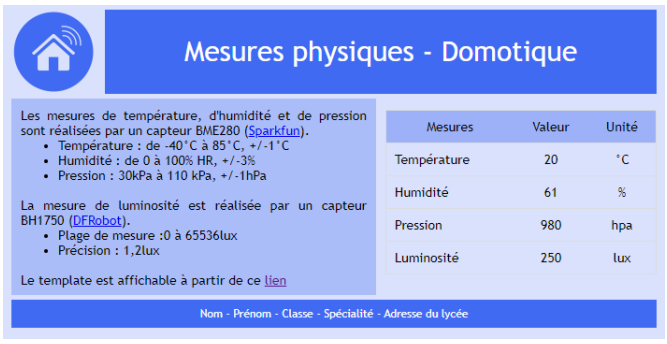


Figure 6 : page réalisée à partir du gabarit *(fichiers domotique.html, style.css et tableau.css)*

La réalisation d'un site nécessite de déterminer, **au préalable**, le public auquel il s'adresse et les informations qu'il aura à présenter. Dans le cas présent, ce travail a été réalisé !

2 Découverte du CSS

[🕐 1h]

2.1 Comprendre le modèle de boîte

En CSS2 comme en CSS3, la mise en page repose sur le **modèle de boîte** : chaque élément d'une page est considéré comme une boîte que l'on peut imbriquer, juxtaposer ou superposer. Ce principe permet de structurer et d'organiser textes, images et autres contenus de manière cohérente.

2.1.1 Le modèle de boîte

En **HTML**, chaque élément est considéré comme une **boîte**. Aux dimensions induites par son **contenu** s'ajoutent souvent les espaces de **marges externes (margin)** ou **internes (padding)** et une **bordure (border)**.

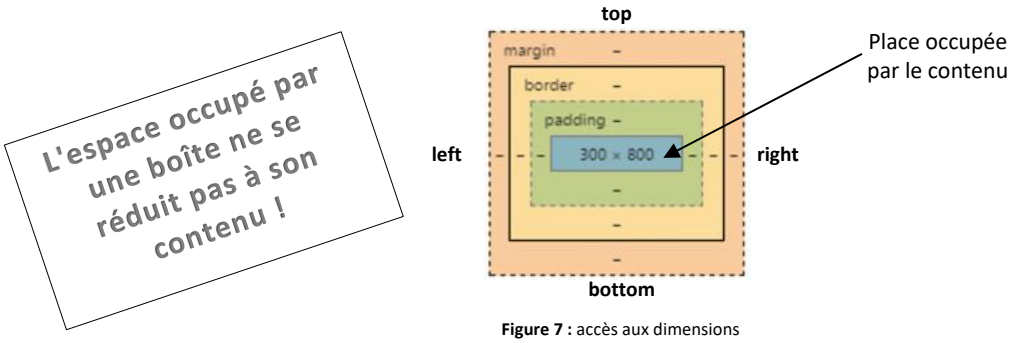


Figure 7 : accès aux dimensions d'une boîte avec la touche [**F12**] dans le navigateur

Les propriétés **border**, **margin** et **padding** ne sont pas obligatoires. Non renseignées, elles prendront la valeur par défaut, toujours **nulle** pour les éléments en ligne. En revanche **parmi les éléments de type bloc, seul n'a pas de marge par défaut**.

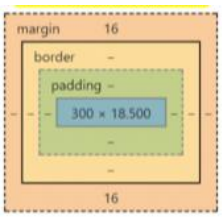


Figure 8 : analyse de <p>

💡 Exemple : La balise de paragraphe <p> possède par défaut une marge supérieure (*margin-top*) et une marge inférieure (*margin-bottom*) de **16px**.

👉 Remarque : Il est fréquent de supprimer ou de réinitialiser ces marges par défaut afin de garder un contrôle précis sur la mise en page.

2.1.2 Calcul de la place occupée par une boîte

La place occupée par une boîte est **calculée** en **ajoutant** les marges internes (**padding**), externes (**margin**) et l'épaisseur de la bordure (**border**).

Exemple : La boîte ci-contre occupe :
- une largeur l = 2*2 + 146 = 150px et
- une hauteur h = 2*5 + 2*2 + 126 = 140px

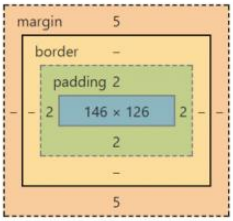


Figure 9 : calcul de la place occupée par une boîte

2.2 Concevoir le gabarit du site *Domotique* pas à pas

2.2.1 Organisation du projet *Domotique*

Le projet est regroupé dans le dossier **Domotique2**, qui contient :

- un fichier principal ***template.htm,***
- deux sous-dossiers : *img* et *styles*. L'image du logo (***domo.png)*** est dans *img* et la feuille de style (***style.css***) dans *styles*.

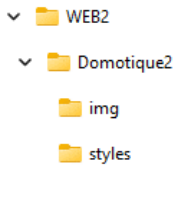


Figure 10 : dossier de travail

2.2.2 Étape 1 : afficher les blocs

Le fichier *template.html* contient six balises ** ** identifiées par *page*, *logo*, *banner*, *content*, *aside* et *footer*.



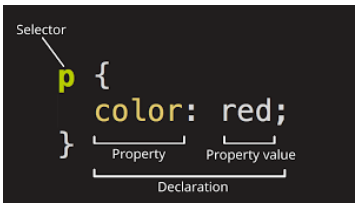
*« L'élément HTML ** ** (pour **div**ision du document) est un **conteneur** générique qui permet d'organiser le contenu sans représenter rien de particulier. Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs **class** ou **id**) »*

Exemple : **<div id**="page"> ** **

Les **id**entifiants page, *logo*, *banner*, *content*, *aside* et *footer* sont repris dans le fichier *style.css* pour déclarer les **sélecteurs** assurant la mise en forme des boîtes qu'ils référencent.

Définition

Sélecteur simple *: toute balise HTML privée des signes* **<** et **>**. *Le sélecteur "* ***accroche****" un* ***effet de style*** *à un élément ou une balise HTML. Chaque sélecteur (body, p, etc.) possède un ensemble de* ***propriétés*** *qui peuvent être définies par une ou plusieurs feuilles de style.*



La **déclaration d'un style** s'effectue par le couple **propriété : valeur ;**

Identifie la **nature** de l'effet souhaité.

Identifie ce qui sera défini dans le style adopté.

Syntaxe

RÈGLE CSS
=
UN Sélecteur
+
une ou plusieurs
déclarations de style

```
sélecteur {  propriété1 : valeur1 ;
             propriété2 : valeur2 ;
                    ...
             propriétén : valeurn ;
}
```

Sélecteur

Exemples : body {backgroud-color: black;}
div#page{ width: 700px;} /* dans la page css, # fait le lien avec id dans la page HTML */

Remarques : Il est possible de définir une ou plusieurs propriétés pour un groupe de sélecteurs.
Exemple : h1, h2 {text-align : right ;}

Chaque couple ***propriété : valeur*** se termine par un point-virgule " **;** " et l'ensemble est encadré par des accolades **{ }.**

Les cibles des principales propriétés

- Propriétés liées aux **polices de caractères**
- Propriétés liées à la **mise en page** des textes
- Propriétés liées aux **arrière-plans**
- Propriétés liées aux **bordures**
- Propriétés liées à la **définition des marges**

Activité 1

1. Ouvrez l'**IDE VSCode** en cliquant sur l'icône placée dans la barre des tâches.
2. **Placez** le dossier **WEB2** dans **VSCode** et dépliez **Domotique2** (fig. 11).
3. Placez les fichiers *template.html* et *styles.css* l'un à côté de l'autre à l'écran en cliquant sur l'icône . (fig.12)
4. Ouvrez la page *template.html* dans un navigateur par un clic droit et sélectionnez *"Open In Default Browser" (**Alt-B**).*
5. Identifiez les boîtes en les nommant sur le **document réponse 1** et **notez** leurs dimensions dans le tableau.

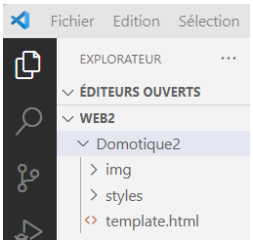


Figure 11 : Domotique2

Étape 1 : afficher les blocs

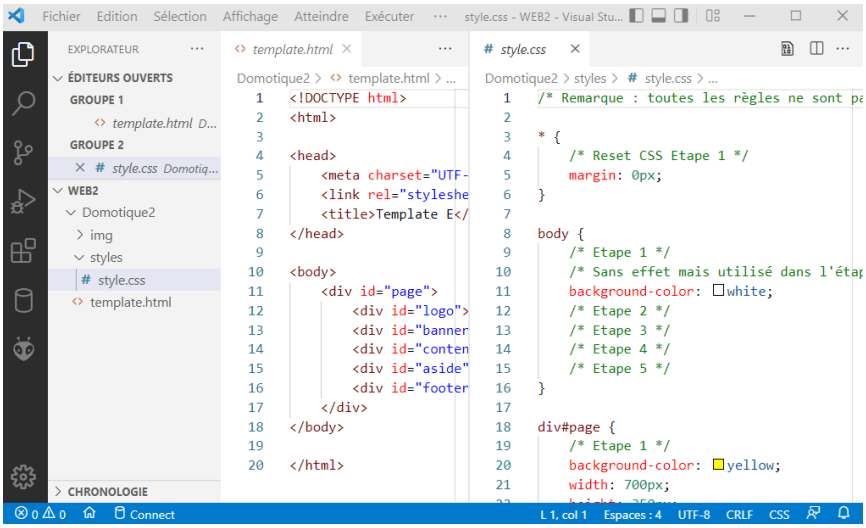
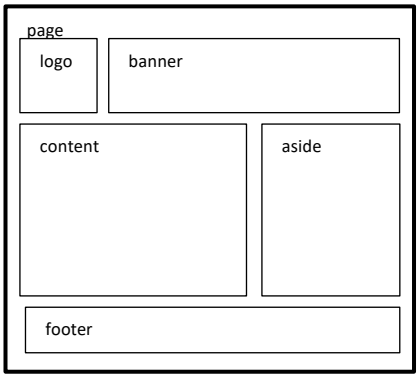


Figure 12 : template.html et style.css côte à côte

Ces **dimensions** représentent l'**encombrement total** de chacune des boîtes à l'écran. Elles ne devront pas être modifiées par les **marges** ou le **bord** ajoutés dans les étapes suivantes. C'est pourquoi il est utile d'en conserver une trace afin de s'y référer si l'affichage ne se fait pas correctement !

2.2.3 Étape 2 : placer les blocs

Le positionnement attendu est représenté par la **maquette graphique** du site "Domotique" ci-dessous.



Étape 2 : placer les blocs

Figure 13 : maquette graphique du site "Domotique"

Comme vous pouvez le constater, on souhaite placer le bloc *banner* à droite du bloc *logo* et *aside* à droite de *content*. Le langage CSS dispose d'une propriété pour cela : **float**.

Activité 2

a) Placez la déclaration de style ***float:left;*** sous **les commentaires /* Étape 2 */) des règles** *logo et banner*.
Remarque : pour le moment, les blocs sont joints. Nous les éloignerons à l'étape suivante.
b) En suivant la même démarche, **placez** *aside* à côté de *content*.

Tout semble bien se passer, ou presque !
En effet, le bloc *footer* a certainement disparu. Si c'est le cas, mettez la déclaration *float:left;* sous le commentaire /* Étape 2 */ dans la règle CSS *footer* pour obtenir la copie d'écran ci-dessous.

Résultat attendu

float

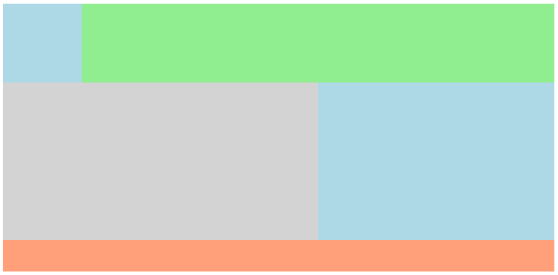


Figure 14 : les blocs sont positionnés

L'étape 2 est-elle terminée ?
Oui et non. Oui, car on obtient bien le positionnement souhaité et non, car la **modification ultérieure des dimensions du bloc** *page* sera problématique.

⚠ En effet lorsque l'on place des déclarations de style ***float:left;*** dans une règle CSS, on demande à la boîte de se positionner (**flotter**), **si elle le peut**, à droite de l'élément précédent (**qu'il y en ait ou pas !**). Donc pour le moment, si le bloc *content* ne flotte pas à droite de *banner* c'est uniquement parce qu'il n'a pas la place **à l'intérieur** du bloc *page*.

Désactivez la largeur du bloc *page* en plaçant une étoile * devant *width:700px;*. Le bloc *page* prend maintenant toute la largeur de l'écran. Les blocs *content* et *aside* flottent à droite de *banner* !!!

Pour "dire au bloc" *content* de ne plus flotter à droite de *banner* mais de se placer en dessous tout en continuant à flotter avec *aside* : placez la déclaration ***clear:both;*** sous /* Étape 2 */ dans la règle CSS *content*.

Activité 3

En suivant la même démarche, **placez** le pied de page footer puis **supprimez** l'étoile * devant *width:700px;*.

Le rendu visuel n'a pas changé et les blocs restent en place lorsque l'on redimensionne le bloc *page* !

2.2.4 Étape 3 : éloigner les blocs

Le positionnement étant terminé, nous allons maintenant créer un effet visuel en éloignant les blocs les uns des autres comme sur la figure 15 ci-dessous. Pour cela, nous allons mettre un peu de ***margin*** dans les règles CSS *logo* et *banner*.

margin



Figure 15 : *logo* a été éloigné de *banner*

Placez la déclaration de style ***margin:5px;*** sous /* Étape 3 */ dans les règles CSS *logo* et *banner*. Testez le résultat dans le navigateur. Le bloc l*ogo* s'est bien éloigné du bord de la page, mais *banner* n'a plus la place pour rester à côté comme sur la figure 16 !

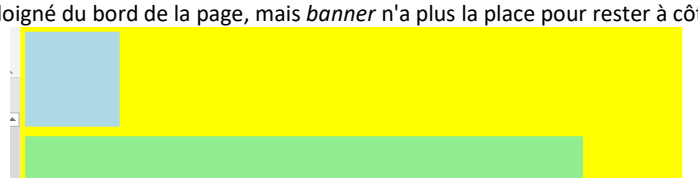


Figure 16 : *logo* et *banner* n'ont plus assez de place à l'intérieur du bloc *page*

En effet, le bloc *logo* "a grossi" de **5px** sur chacun de ses côtés. Pour redonner à logo **ses dimensions initiales** (100px * 100px) et préserver la mise en page, il faut calculer les nouvelles valeurs de *width* et *height* :

width = $100px_{(initiale)} - 2*5px_{(margin)}$ = **90px** height = $100px_{(initiale)} - 2*5px_{(margin)}$ = **90px**

Activité 4

a) Redimensionnez *logo* et *banner* pour qu'ils se positionnent comme sur la figure 15.
⚠ Pour b) et c), placez les **déclarations de style sous les commentaires /* Étape 3 */) des règles concernées.**
b) **Éloignez** *content* de *aside* et du bord de la page en utilisant des **margin-left** et **margin-right** (**5px**).
c) **Placez** une marge de **5px** autour de *footer***.**

Résultat attendu



Étape 3 : éloigner les blocs

Figure 17 : les blocs sont positionnés et éloignés les uns des autres

Maintenant que la structure des boîtes est correctement positionnée et espacée, nous allons pouvoir tester la **mise en forme du contenu textuel**.
L'objectif de l'étape suivante est de **simuler du contenu** (texte et image) afin de **vérifier le rendu visuel** et d'expérimenter différentes propriétés typographiques (taille, police, alignement).

2.2.5 Étape 4 : simuler le contenu textuel

Nous avons du texte à placer dans les blocs *banner*, *content* et *footer* et une image dans le bloc *logo*. Dans un premier temps, nous allons utiliser du faux texte appelé ***Lorem ipsum*** pour voir l'effet des propriétés (**font-size**, **font-family**, **text-align**, etc.) que l'on souhaite appliquer au texte définitif.

Lorem ipsum

*« Le **lorem ipsum** (également appelé **faux texte** ou **lipsum**) est, en imprimerie, une suite de mots sans signification **utilisée à titre provisoire** pour **calibrer une mise en page**, le texte définitif venant remplacer le faux texte dès qu'il est prêt ou que la mise en page est achevée.*
*Il circule des centaines de versions différentes du Lorem ipsum, mais ce texte aurait originellement été tiré de l'ouvrage écrit par Cicéron en 45 av. J.-C., De finibus bonorum et malorum (Liber Primus, 32), texte populaire à cette époque, dont l'une des premières phrases est : « Neque porro quisquam est qui do**lorem ipsum** quia dolor sit amet, consectetur, adipisci velit… » (« Il n'existe personne qui aime la souffrance pour elle-même, ni qui la recherche ni qui la veuille pour ce qu'elle est… ») ». (Wikipédia)*

Activité 5 – retour sur le fichier *template.html*

a) **Placez** l'image *domo.png* dans le bloc logo**.**
b) **Placez** une ligne de faux texte dans les blocs *banner* et *footer* (touche F1, écrivez lorem dans la barre de recherche de l'éditeur et sélectionnez "*Lorem Ipsum: **Insert a line***"). Réduisez la taille de la ligne quatre mots.
c) **Placez** un paragraphe de faux texte dans le bloc *content* (touche F1, écrivez **Lorem** dans la barre de recherche de l'éditeur et sélectionnez "*Lorem Ipsum: **Insert a paragraph***").

Indications pour l'activité 6

On remplace ***height*** par ***line-height*** pour **centrer verticalement** un texte. La déclaration de style ***text-align:center;*** permet de le **centrer horizontalement**. La propriété ***font-size*** est utilisée pour le dimensionner et **text-align:justify**; le **justifie**.

Activité 6

⚠ Placez les **déclarations de style sous les commentaires /* Étape 4 */) des règles concernées.**
a) **Centrez** et **dimensionnez** le texte situé dans *banner* (**30px**) et *footer* (**12px).** Régler le texte dans *body* à 14px.
b) **Justifiez** le texte situé dans *content.*
c) **Définissez** la police de caractère de la page en ajoutant la déclaration de style ***font-family: "Trébuchet MS", Arial, sans-serif;*** à la règle body.

Indications pour l'activité 7

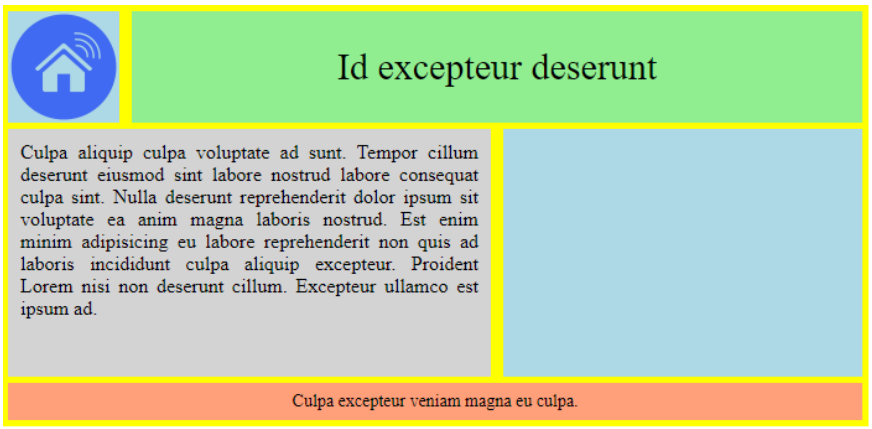
On utilise la propriété **padding** pour éloigner le texte du bord d'un blo. Mais comme pour *margin*, si on ajoute du padding, la place occupée par la boîte augmente !
La déclaration ***box-sizing: border-box;*** corrigera automatiquement les dimensions !

padding

Activité 7

Placez les déclarations ***padding:10px;*** et ***box-sizing: border-box;*** sous le commentaire /* Étape 4 */ dans le bloc *content*. Visualisez le résultat dans le navigateur. Au contenu textuel près, vous devriez obtenir le rendu ci-dessous.

Résultat attendu



Étape 4 : simuler le contenu textuel

Figure 18 : le gabarit du site domotique en bonne voie !

2.2.6 Étape 5 : faire les finitions

Notre gabarit est presque terminé. Nous allons maintenant modifier les couleurs des blocs. Pour "faire simple" la couleur de l'image *domo.png* a été identifiée. Elle sera utilisée avec des transparences pour obtenir le rendu de la figure 5.

Des logiciels comme **paint.net, Gimp, etc.** offrent un outil appelé "**sélecteur de couleurs**" ou "**pipette**" pour déterminer le code **R**ed **G**reen **B**lue (RVB en français) d'une couleur.



Figure 19 : paint.net

Les couleurs peuvent être définies en CSS de plusieurs manières :

- **en notation RGB :** rgb(65, 106, 242)
- **en notation hexadécimale :** #416AF2
- **avec une valeur de transparence (*alpha*) :** rgba(65, 106, 242, 0.5)

RGB

Dans ce tutoriel, nous utiliserons la notation RGB avec transparence pour harmoniser le design avec le logo.

Exemple : ***background-color: rgb(65, 106, 242, 0.5);*** */* Correspond à la couleur du logo avec une transparence égale à 50% */*

Activité 8

a) **Supprimez** le faux texte, le remplacer par :
- "Mesures physiques - Domotique" dans le bloc *banner*
- "Nom – Prénom – Classe – Spécialité - Adresse du lycée" dans le bloc *footer*

⚠ Placez les **déclarations de style sous les commentaires /* Étape 5 */) des règles concernées.**
b) **Changer** la couleur du texte des blocs *banner* et *footer* en leur appliquant *la déclaration* ***color:white;***
c) **Appliquer** la valeur ***rgb(65, 106, 242)*** à la propriété *background-color* des différents blocs avec les transparences suivantes :
- 100% pour *banner* et *footer*,
- 30% pour *content* et *aside*,
- 20% pour *body* (fond d'écran).

d) **Centrer** le bloc *page* en lui ajoutant la déclaration ***margin: 0px auto;***
e) **Supprimer** la déclaration *background-color* dans *les conteneurs logo* et *page.*
f) **Faites** un copier-coller du fichier *style.css* et renommez-le ***template.css***.

Résultat attendu

gabarit

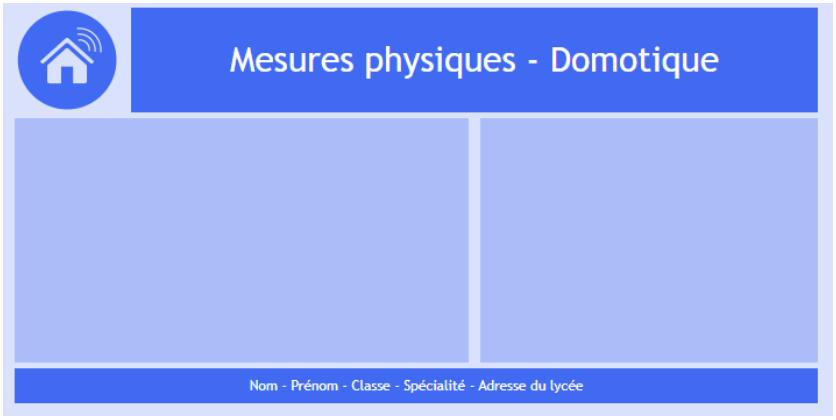


Figure 20 : gabarit du site domotique

3 Synthèse

[🕐 20mn]

3.1 Préparation

1. Faites un copier-coller du fichier *template.html* et renommez-le *domotique.html.*
2. Remplacez le lien vers *style.css* dans *template.html* par *template.css*.

Le gabarit peut maintenant être utilisé pour créer la page "**Mesures physiques – Domotique**" représentée ci-dessous.

3.2 Conception de la page domotique

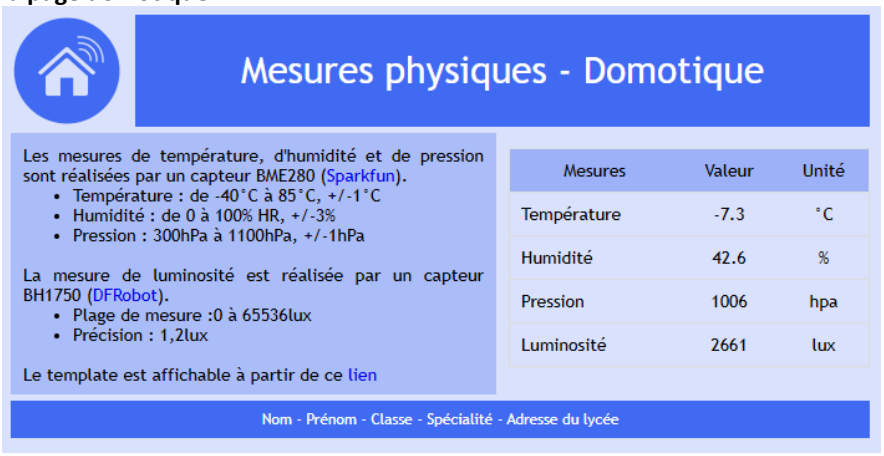


Figure 21 : page domotique

Ne modifiez plus ***template. html*** et ***template. css***

Travaillez uniquement sur **domotique.html et style.css** pour réaliser la page finale

Activité 9

a) **Complétez** le bloc *content* avec le texte de la figure ci-dessus.
b) **Ajoutez** des liens s'ouvrant dans de nouvelles fenêtres pour accéder à la page dédiée au capteur BME280 sur le site **Sparkfun** et au capteur BH1750 sur le site **DFRobot**.
c) **Créez** le tableau des valeurs dans le bloc *aside* et positionnez-le (width :100%;) dans le CSS pour qu'il occupe toute la largeur du bloc, etc. (recherches Internet nécessaires)
d) **Placez** un lien pour afficher le gabarit (*template.html*) !

3.3 Amélioration du tableau des grandeurs physiques

Voir prof

4 Compléments : les balises de type bloc et en ligne

- **HTML**
  La plupart des balises peuvent se ranger dans l'une ou l'autre des **catégories :** bloc ou en ligne**.**

bloc

Une balise de type **bloc** crée automatiquement un **retour à la ligne** avant et après son contenu. Une page web est constituée d'un ensemble de blocs que l'on peut mettre **les uns à l'intérieur des autres**.

Exemples : <hx></hx>, ,

En ligne

Une balise de type **en ligne** se trouvera **obligatoirement à l'intérieur** d'une balise de type *bloc*. Une balise de type en ligne ne crée pas de retour à la ligne.

Exemples : <a>, , …

- **CSS**
  La propriété **CSS *display*** permet de transformer n'importe quel élément d'un type dans un autre. Cette propriété prend les valeurs ci-dessous.

display

Valeur	Description
inline	Les éléments se placent les uns **à côté** des autres.
block	Les éléments sont en forme de blocs. Ils se placent les uns **en dessous** des autres et peuvent être redimensionnés.
inline-block	Les éléments sont positionnés **les uns à côté des autres** (comme inline) et peuvent **être redimensionnés** (comme block).
none	Les éléments ne seront pas affichés.

Exemple : positionnement de liens les uns en dessous des autres et redimensionnement possible.

```
a {
   display : block ;
}
```

Pour aller plus loin…





Apprenez à créer votre site web avec HTML5 et CSS3

https://edurl.fr/s22IpKhT

Webographie

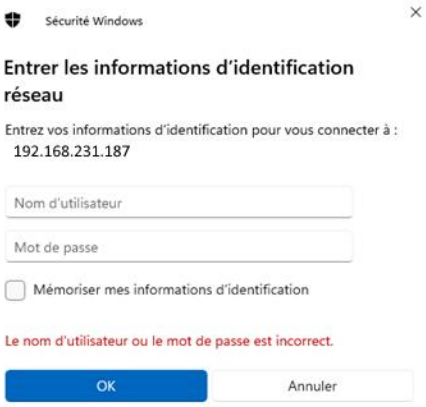
- Référence CSS sur mdn web docs
  https://mzl.la/3MDMpKu
- Compatibilité des navigateurs sur "Can I use"
  https://caniuse.com
- Validateur de code
  https://validator.w3.org/

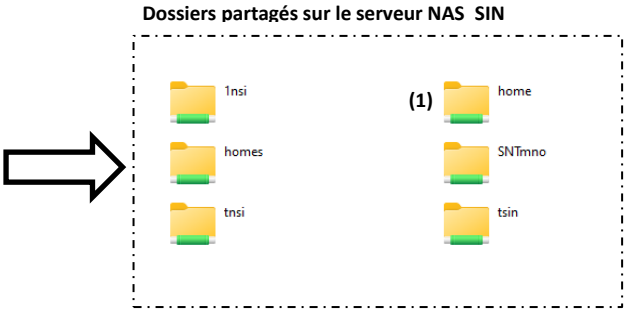


Annexe 1 : installation du dossier du TP dans votre compte sur le serveur Nas_SIN

1. Connexion au serveur NAS_SIN pour accéder au dossier partagé du cours.

- Sélectionnez  et entrez **\\192.168.231.187** dans l'invite de commande. 
- Complétez la fenêtre de sécurité Windows représentée ci-dessous avec l'**identifiant** et le **mot de passe** de votre compte sur le serveur **NAS_SIN pour accéder aux dossiers partagés.**





(1) Votre dossier personnel

- Double-cliquez sur le dossier **du cours**.

2. Dossier à utiliser dans le TP

Copiez le dossier **WEB2** situé dans ***DOSSIER_du_COURS/5_IHM_WEB/*** et collez-le dans ***home/www/***.

