

## **Ecrire des pages web avec le HTML et le CSS (d'après : <http://isn-marchal.com>)**

L'objet de cette page est d'arriver à écrire des pages pour un site internet, avec un nombre restreint d'instructions et de comprendre le fonctionnement de base du html et du css.

Pour des explications plus évoluées, je vous recommande le site suivant : Open classroom

Voici les différents points abordés:

Pourquoi deux langages associés

Principes de base du html

Les blocs et les balises

Les différentes sortes de blocs

L'entête d'un fichier html

Les "in line"

Les classes

Les liens

Les images

Bien davantage

Principes de base du css

Comment relier le fichier html et le fichier css

Les couleurs

Les propriétés de la police

Alignement du texte

Les bordures

Les arrière-plans

Taille des blocs

Position des blocs

Les marges et les paddings

Travaux pratiques

### **1) Pourquoi deux langages associés ?**

Dans une page internet, on distingue le fond (le contenu lui même) ainsi que la structure de la page d'une part et la forme (taille, police, disposition etc...) d'autre part.

Le fait de distinguer ces deux opérations, permet de changer l'un, sans changer l'autre et de réaliser séparément les deux opérations (peut-être par des personnes différentes).

Le html est le langage qui s'attache au fond, le css à la forme.

### **2) Principes de base du html**

#### **Les blocs et les balises**

Le contenu de la page internet est créé en le répartissant dans diverses boîtes, pour que les propriétés d'affichage de ces boîtes puissent ensuite être décrites grâce au fichier css associé.

Ces boîtes sont délimitées par des balises d'entrée et de sortie. < p > est la balise d'entrée dans un bloc de type p (pour paragraphe) et < /p > est la balise de sortie correspondante.

Le paragraphe s'écrit alors entre ces deux balises : c'est le contenu de la boîte.

Exemple :

< p > Ceci est le paragraphe < /p>

Il existe deux sortes de boîtes, les blocs et les "in-line".

#### **Les blocs**

Il existe différents noms (indicatifs de l'objet du bloc) pour les blocs, de manière à rendre le code plus intelligible. Ces blocs n'ont que peu de propriétés prédéfinies. Il faut toutefois savoir que, sans

modification, l'affichage d'un bloc est suivi d'un retour à la ligne. Dans le flux normal de l'affichage, les blocs s'affichent donc les uns en dessous des autres.

Voici quelques uns de ces différents types:

p pour un paragraphe

Attention le "p" est particulier : on ne peut mettre d'autre bloc dans un bloc de type "p". C'est le plus petit bloc possible. De plus, si on utilise <p> tout seul (sans balise de sortie), cela provoque une ligne vide.

header pour une entête

footer pour un pied de page

div pour une division

section pour une section

aside pour un élément à placer sur le côté

ul pour une liste non numérotée (unordered list)

ol pour une liste ordonnée

li pour un item d'une liste

h1 pour une section de première importance

h2 pour une section de seconde importance (et ainsi de suite jusqu'à h8)

nav pour mettre des liens

article

Pour utiliser un bloc, procéder comme précédemment :

< h1 > Ceci est un titre très important < /h1 >

### **L'entête d'un fichier html**

Un fichier html commence toujours par une entête ("head"). Elle explique, entre autres, que le code est du html, que l'on utilise (par exemple) l'unicode pour ne pas avoir de problème avec les accents, quel est l'auteur etc...

Ces informations qui donnent des renseignements sur la page s'appellent des métadonnées. On en trouve dans les entêtes de presque toutes les sortes de fichiers.

Voici un exemple, qui vous permettra de tester aussi les différents blocs.

```
<!DOCTYPE html >
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<title > Essai</title>
```

```
</head>
```

```
<body>
```

```
<header> Ceci est un header</header>
```

```
<h1> Ceci est un h1</h1>
```

```
<h2> Ceci est un h2</h2>
```

```
<aside> Ceci est un aside</aside>
```

```
<footer> Ceci est un footer</footer>
```

```
<p> Ceci est un paragraphe</p>
```

```
<ol> liste ordonnée</ol>
```

```
<li> li pour un item d'une liste</li>
```

```
</body>
```

```
</html>
```

Pour essayer ce code, copier le dans un éditeur (notepad++ en choisissant dans "Affichage" , puis "syntaxe" le html), et enregistrer le sous "essai.html" par exemple. Ce qui est important, c'est de mettre l'extension ".html" !.

Il ne vous reste plus qu'à ouvrir votre fichier.

A savoir : pour modifier votre fichier, ouvrir d'abord l'éditeur choisi, puis le fichier (s'il n'est pas resté ouvert).

On constate que tout ce qui est affiché entre les balises < head> et < /head > n'est pas affiché sur la page. Ce qui est affiché est placé entre les balises du "body".

Les blocs sont tous affichés les uns en dessous des autres sans distinction. Dans les blocs de listes le texte est décalé vers la droite, et dans les blocs de titres, la taille de la police est variable.

Tout ceci, et bien d'autres propriétés, peut être modifié dans le fichier css.

Attention, le texte ne s'écrit pas sur la page comme dans votre fichier...Pour passer à la ligne, il faut taper <br > (qui est appelée balise orpheline car elle n'est pas associée à une balise de sortie).

### **Les "in line"**

On peut vouloir attacher des propriétés particulières à des éléments de texte (souligner, mettre en italique, en gras etc...). Les blocs ne sont pas appropriés pour cela, car ils impliquent un passage à la ligne...On utilise alors des éléments "inline" qui se mettent également entre des balises d'entrée et de sortie comme les blocs.

Ces "in-line" sont affichés dans le flux normal du texte, à l'endroit où on aurait affiché le caractère suivant. Voici les principaux :

- em pour mettre en valeur
- strong pour mettre très en valeur
- mark pour marquer quelque chose
- span pour faire ce que l'on veut

Je vous recommande de faire des essais....

### **Les classes**

Les noms des différents blocs et inline (c'est à dire des différentes boîtes) sont souvent insuffisants pour distinguer les différents types d'objets. On a donc la possibilité de distinguer des boîtes en leur donnant des noms. Voici un exemple :

```
<p class="mon_paragraphe">Bla bla bla </p>.
```

On y met ce qu'on veut, la seule différence est que l'on pourra, grâce au css, traiter ce paragraphe, et les autres qui porteront le même nom, de manière particulière...

### **Les liens**

Ce sont des cas particuliers de "inline", c'est à dire que si l'on ne change rien, les liens , comme les images se positionnent dans le flux normal, sans passage à la ligne

Pour insérer un lien on utilise la balise <a> et la balise de sortie correspondante. Cette balise est utilisée aussi bien pour des liens vers d'autres pages du même site, que pour des liens vers des pages externes.

Il est recommandé de créer un dossier pour y stocker tous les fichiers concernant un site internet, appelé par exemple "monsiteweb".

Dans la balise, on met l'adresse du lien après l'attribut "href=", en l'écrivant entre des guillemets. Cette adresse peut être soit :

- absolue c'est à dire que l'on écrit l'adresse complète "http://...."
- ou "C:\Users\toto\Desktop\monsiteweb\page3"
- relative .

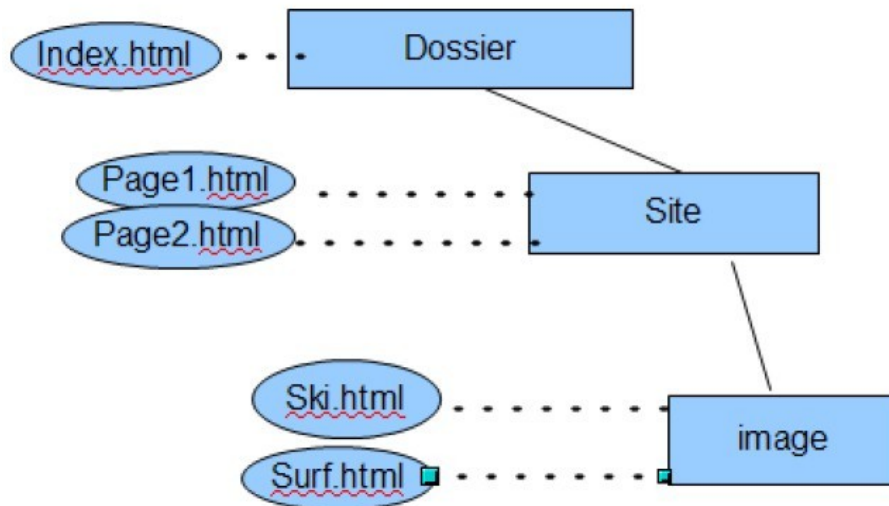
Voici une organisation fictive de fichiers :

Si on est sur la page "index.html" et qu'on veut pointer vers "Page1.html", on écrira : href="site/Page1.html" car le dossier "site" est dans le même dossier que "index.html"

Si on est sur la "Page1.html" et qu'on veut pointer vers "index.html", on écrira href="../index.html" car "../" permet de remonter au dossier parent du dossier dans le quel se trouve "Page1.html".

Si on est dans "Ski.html" et qu'on veut pointer vers "index.html", on écrira donc href="../../index.html" car on remonte deux fois au dossier parent.

Si on est sur "Page1.html" et qu'on veut pointer vers "Page2.html", on écrira simplement href="Page2.html".



Par exemple, si vous créez la page2, stockée dans le dossier "monsiteweb" et que vous voulez créer un lien vers la première page "index.html" (stockée dans le même dossier) vous allez écrire :

```
<a href="index.html" >Accueil </a >
```

Ce qui apparaîtra à l'écran c'est le mot accueil souligné (le titre du lien est ce qui est entre la balise d'entrée et celle de sortie).

Si vous voulez créer un lien vers ce site web vous écrirez:

```
<a href="http://isn-marchal.com" >Site de la spécialité isn </a >
```

### **Les images**

Une image est également du type "inline", c'est à dire qu'elle se positionne (si l'on ne fait rien) dans le flux normal, à l'endroit où serait écrit le mot suivant et au-dessus de la ligne courante.

On la met donc en général dans un paragraphe, ce qui permettra de la placer plus facilement à l'endroit souhaité.

Voici un exemple :

```
>p></p>
```

Après src on écrit le chemin de l'image avec les mêmes principes que pour les liens.

Je recommande de mettre tous les fichiers images dans le même dossier que les fichiers html (au moins dans un premier temps).

Après alt, on met un petit texte qui sera affiché en cas d'impossibilité d'afficher l'image (utile aussi pour des handicapés).

Le point sur les formats images :

Le format jpg (JPEG) est un format qui permet de réduire le poids des photos

Il est recommandé pour afficher les photos, qui dans les autres formats mettent beaucoup plus longtemps à se charger

Le format png est beaucoup plus lourd, mais il est recommandé pour afficher des graphiques etc...

Le gif a largement été remplacé par le png, mais sert encore pour les gifs animés

Le bmp est beaucoup trop lourd pour des pages internet

On peut changer le format d'une image en l'ouvrant avec paint par exemple, puis en utilisant "enregistrer sous".

Choisissez des noms parlants pour vos images, sans accents et sans espaces...

Attention, vous ne pouvez publier que des photos libres de droits d'auteur !!

On en trouve très facilement, par exemple à partir du lien suivant (qui date un peu...):

50 banques d'images libres et gratuites (<http://www.fredzone.org/50-banques-d-images-gratuites-et>)

Il est aussi très pratique de savoir utiliser, pour cela, la recherche avancée de Google :

taper les mots clés de votre recherche dans Google

cliquer sur images

cliquer sur l'écrou en haut à droite, puis sur recherche avancée

dans la rubrique "droits d'usage" choisissez ce qui vous convient

### **Bien davantage**

Il est clair qu'il existe de nombreuses autres méthodes et balises....

Ne sont décrites dans cette page que les premières balises, qui peuvent toutefois suffire à créer un site assez basique.

Il faut savoir que l'on peut :

créer des tableaux, voir Les tableaux sur open classroom

créer des ancres qui permettent de se déplacer à un point précis d'une page, voir Les ancres open classroom

faire un lien pour télécharger un fichier :

`<a href="monfichier.zip">Télécharger le zip</a>`

après avoir mis " monfichier.zip" (ou ".pdf", ".doc" etc...) dans le même dossier que la page.

faire un lien pour envoyer un mail :

`<a href="mailto:votrenom@free.fr">Envoyer un mail à votrenom</a>`

ajouter des sons, des formulaires etc....

Il est également très instructif de regarder le code source des pages que l'on affiche sur internet : pour cela, il suffit de faire un clic droit sur la page, puis de sélectionner "Afficher le code source de la page".

### **3) Principes de base du css**

Le css va permettre d'indiquer les propriétés choisies pour les différentes boîtes.

Comme le HTML, le CSS est un langage à balise, avec toutefois une syntaxe différente :

nom\_de\_la\_boite

{

attribut\_a\_modifier:valeur de l'attribut;

autre\_attribut:valeur;

etc...

```

}
ou
.ma_classe
{
    attribut_a_modifier:valeur de l'attribut;
    autre_attribut:valeur;
    etc...
}

```

pour attribuer des propriétés à une boîte appelée `ma_classe` ( sans oublier le "." devant).

### **Comment relier le fichier html et le fichier css**

Aller dans Notepad++, dans "Affichage" et "Syntaxe", choisir css (on peut aussi utiliser un autre éditeur).

Il est recommandé d'enregistrer le fichier css dans le même dossier que le fichier html.

Supposons que ce fichier s'appelle "style.css".

Pour que ces styles soient pris en compte dans le fichier html, il faut ajouter dans le "head" les lignes suivantes :

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

### **les couleurs**

On utilise les couleurs pour différents attributs :

color : pour la couleur de la police

background-color : pour la couleur de l'arrière plan

pour la couleur des bordures etc...

Il est donc important de savoir indiquer la couleur choisie. Ceci peut se faire de différentes manières:

en donnant le code RGB de la couleur (c'est à dire les quantités de rouge (Red), de vert (Green) et de bleu (Blue) sous la forme suivante :`rgb(255,0,0)` pour un rouge pur par exemple.

en donnant le code de la couleur en hexadécimal, par exemple :`#7278c2` (avec un dièse devant) qui vous donnera un violet.

en nommant l'une des seize couleurs standards :

Nom de la couleur	Triplet RGB	en français
aqua	#00FFFF	Vert d'eau
black	#FFFFFF	Noir
blue	#0000FF	Bleu
fuchsia	#FF00FF	Fuchsia
gray	#808080	Gris
green	#008000	Vert
lime	#00FF00	Citron vert
maroon	#800000	Maron
navy	#000080	Bleu marine
olive	#808000	Olive
purple	#800080	Pourpre
red	#FF0000	Rouge
silver	#C0C0C0	Argent
teal	#008080	Sarcelle
white	#FFFFFF	Blanc
yellow	#FFFF00	Jaune

Voici un exemple :

```
header
{
  color:blue;
  background-color:#dcf0e5;
}
```

ce qui aura pour effet d'afficher une police en couleur bleu dans l'entête (header) avec un fond d'un bleu très clair.

Vous pouvez essayer, après avoir ajouté le lien vers le fichier de style (fichier css) décrit au paragraphe précédent dans votre fichier "index.html".

Et l'on peut fixer ainsi les propriétés de couleur de toutes les sortes de boîtes.

Reste à trouver les codes des couleurs choisies...Pour cela , vous pouvez installer sur votre ordinateur le logiciel suivant:

La Boite à Couleurs,

ou aller dans Paint (par exemple) en cliquant sur "modifier les couleurs".

### **Les propriétés de la police**

Pour la police, outre sa couleur on peut choisir :

font-size qui est la taille de la police.

Cette taille peut-être exprimée en pixel (px), en taille du m (em), ou par des expressions (xx-small, x-small,small,medium,large, x-large,xx-large).

Par exemple:

```
h1
{
  font-size:large;
}
h2
{
  font-size:1.3em;
}
header
{
  font-size:15px;
}
```

Vous pouvez faire des essais en faisant varier les tailles et les couleurs. Note : on peut mettre plusieurs propriétés d'une même boîte dans une même accolade (les unes en dessous des autres)

font-family qui est la police elle-même

Attention, tous les navigateurs ne possèdent pas toutes les polices. On indique donc plusieurs polices, les unes derrière les autres (séparées par des virgules), en finissant toujours par une police qui est partout ("serif" ou "sans serif").

Voici un exemple:

```
h1
{
```

```
font-family:"comic sans ms","verdana","serif";
}
h2
{
font-family:"arial black","sans serif";
}
```

font-style

qui peut être italic ou normal  
font-weight

qui peut être bold (en gras) ou normal  
text-decoration

qui peut être blink (clignotant), underline (souligné), line-through(barré) ,overline(ligne au-dessus)

Je vous laisse le soin de faire des essais.

### **Alignement du texte**

On utilise l'attribut text-align qui peut prendre les valeurs center, left, right, justify. Attention, cet attribut ne peut s'appliquer qu'à des blocs (et pas à des "inline").

### **Les bordures**

Un exemple facilitera les explications :

```
footer
{
border:3px blue solid;
}
```

On voit que l'attribut border a trois propriétés, la première est l'épaisseur de la bordure (en pixels), la deuxième sa couleur (avec toutes les possibilités évoquées dans le paragraphe "Couleur") et la dernière son style qui peut être : dashed (avec des tirets), solid (trait simple), dotted (pointillés), double, groove (relief), inset ou outset (effets 3D).

On peut faire la même chose avec border-left (ou right, top, bottom).

### **Les arrière-plans**

Il est fréquent de placer une image en arrière plan. Cela nécessite aussi de la positionner, de savoir si on la répète et d'autres propriétés que nous n'aborderons pas. Là encore un exemple vaudra mieux que de nombreuses explications :

```
body
{
background-image: url("soleil.png");
background-attachment: fixed; /* Le fond restera fixe */
background-repeat: no-repeat; /* Le fond ne sera pas répété */
background-position: top right; /* Le fond sera placé en haut à droite */
}
```



Ceci vous permettra d'afficher en arrière-plan de votre "body" un soleil (à condition d'avoir mis le fichier "soleil.png" dans le même dossier que vos fichiers html et css).

Les valeurs que peut prendre background-repeat sont no-repeat, repeat (en mosaïque), x-repeat (motif répété horizontalement) et y-repeat.

Comme on va le voir ci-dessous, les problèmes de positionnement sont souvent complexes. L'attribut background-position peut prendre des valeurs du type:

30px 50px :ce qui signifie à 30 pixels du haut et à 50 pixels de la gauche  
top (ou bottom ou right ou left)  
top left

Le point de référence est le coin en haut à gauche du conteneur de votre boîte. C'est à dire que si votre bloc est par exemple dans une section c'est le coin supérieur gauche de cette section qui est le point "origine".

### **Taille des blocs**

On peut définir la taille des blocs de deux manières  
de manière absolue

Voici un exemple:

```
header
{
  width:500px;
  height:200px;
}
```

de manière relative (en pourcentage)

Autre exemple (avec des bordures pour bien voir l'espace occupé):

```
header
{
  width:50%;
  height:70%;
  border:3px blue solid;
}
header p/*ceci signifie les "p" du "header*/
{
  width:50%;
  height:50%;
  border: 1px black solid;
}
```

Si vous testez ceci, vous allez commencer à comprendre la complexité des problèmes de dimensions et de positions.

On voit que le "header" occupe bien 50% du "body" en largeur, mais que le "header p" n'occupe que 50% du "header".

Il faut donc se souvenir que les pourcentages sont ici des pourcentages du précédent conteneur !

### **Position des blocs**

Cette question est vraiment difficile, mais je vais vous expliquer une méthode basique qui sera suffisante dans un premier temps. Les puristes trouveront certainement que ce n'est pas satisfaisant, mais cela a au moins le mérite de permettre un positionnement correct assez facilement. L'inconvénient majeur est que ce positionnement absolu n'aura le rendu attendu que sur des écrans de taille comparable au vôtre, alors que les écrans peuvent être de tailles très variables (smartphones, tablettes etc...).

Ceux qui ont un peu de temps, peuvent essayer de positionner leurs éléments en utilisant le positionnement flex.

Voir pour cela le site : Openclassrooms : le flex

Pour d'autres techniques, voir aussi Positionnement sur Openclassroom

Nous allons parler uniquement de positionnement absolu. Pour chaque boîte, on va donner sa position en pixels par rapport au coin en haut et à gauche de son conteneur. Ceci impose de commencer par dire, pour chaque boîte que l'on utilise la position absolue, puis de fixer les dimensions des boîtes (en position normale, dans le flux, les blocs occupent 100 % de la largeur, mais ceci cesse en position absolue):

```
header
{
  position:absolute;
  top:15px;/*on peut aussi exprimer ces longueurs en
  pourcentage du bloc conteneur, càd ici de l'écran*/
  left:50px;
  height:800px;
  width:900px;
}
header p
{
  position:absolute;
  top:10px;/*à 10 pixels du haut du "header" qui est son conteneur*/
  left:30px;/*si on prend 10% par ex ici, ce sera 10% de la
  largeur du conteneur qui est le header*/
  height:500px;
  width:600px;
}
```

De cette manière, quand vous concevez une page web, faites un dessin, puis positionnez tous vos éléments dessus, et reportez ceci dans votre fichier css.

Attention, les éléments en position absolue sont sortis du flux. C'est à dire que les éléments du flux sont positionnés, les uns derrière les autres (pour les 'inline') ou en dessous des autres (pour les blocs) et indépendamment les éléments en position absolue sont positionnés: des recouvrements sont alors presque certains !

Essayez de réaliser (à peu près) l'exemple suivant, après avoir fait un dessin sur une feuille pour essayer de fixer à peu près les dimensions (prenez par exemple des pourcentages).

Voici la page à obtenir: appelons la "proj1.html"

Pour voir un corrigé, afficher le code source de la page (clic droit sur la page), puis cliquer sur le lien du head "style1.css", pour voir le fichier ".css".

Une fois les blocs principaux positionnés, on continue un positionnant les éventuels blocs secondaires (c'est à dire ceux qui sont à l'intérieur des blocs principaux) et ainsi de suite..

Ensuite seulement, on s'occupe du contenu, des couleurs, des polices etc.

### **Les marges et les paddings**

Supposez que votre bloc soit un cadre rectangulaire, il y a les marges extérieures au cadre (distances avec les bords du conteneur) qui sont les "margin" et les marges entre le cadre et le début du contenu qui sont les "padding". On peut spécifier "margin-bottom", "padding-left" ou donner des valeurs qui s'appliquent aux quatre côtés.

Exemple :

```
header p
{
    margin-top:10%;
    margin-left:50px;
    padding:10em;/*s'applique aux quatre côtés*/
}
```

Si on a spécifié la largeur d'un bloc, la commande "margin:auto;", permettra de centrer en largeur ce bloc.

Pour plus de détails, et aussi pour gérer les dépassements de boîtes (overflow) :

Les marges sur open classroom

### **Travaux pratiques**

Nous allons compléter la page "proj1.html" commencée au paragraphe sur les positions.

On voudrait obtenir :

proj2.html

Rappelez vous que vous pouvez, en cas de problème consultez les codes des fichiers html et css.

Ajout d'une image dans p

Choisissez une image libre de droit et mettez la dans le "p".

Attention, on ne veut pas que l'image déborde du cadre. Pour cela deux méthodes : utiliser un logiciel pour modifier la taille de votre image, ou utiliser la propriété "overflow:hidden;" qui masque tout ce qui dépasse.

Essayez de réaliser le aside comme dans proj2.html

Les encadrés avec "page1" etc....peuvent être des blocs, et leur position n'a alors pas besoin d'être absolue, les blocs se positionnant automatiquement les uns en dessous des autres. Jouez avec les marges....

Ajoutez deux sections dans "article" comme sur l'exemple

On veut maintenant obtenir la page finale de ce TP :

proj3.html

Choisissez d'autres couleurs pour les backgrounds de l'article et des sections

Changez la couleur du background du body, et les couleurs de toutes les bordures ou polices qui ne peuvent rester noires

Cherchez une image pour le background du header, éclaircissez la, au besoin avec un logiciel du type Picassa par exemple

Changez la taille de la police du header, et choisissez une autre police. Centrez le texte.

Dans la section de gauche, essayez de mettre une partie du texte en évidence, en utilisant une boîte "inline"

Dans le footer, mettez un lien qui permet d'envoyer un mail

Dans la section de droite, essayez de mettre un texte avec des espaces et des lignes vides....

Vous remarquerez l'ancre (dans la section de droite) et le lien qui y mène dans l'aside (optionnel)

Dans la section de droite, notez le "overflow:auto;" qui est bien utile pour que le texte ne déborde pas..

Il est clair qu'il est facile de remplacer "page1" etc..par des liens vers les éventuelles autres pages...

Il ne vous reste qu'à vous lancer dans votre site personnel !!

N'hésitez pas à chercher des solutions en utilisant un moteur de recherche, ou en posant des questions sur un forum (celui de Openclassrooms fonctionne très bien).