Henallux – DA1 – web : principes de base

Laboratoire 3

Le langage CSS

Sommaire

[1. (Exo) Températures à travers le monde 2](#_Toc159323124)

[2. (exo) mini-recueil d’énigmes 12](#_Toc159323125)

# (Exo) Températures à travers le monde

Objectifs

* définir un style de manière inline ou interne
* utiliser quelques sélecteurs CSS simples
* rédiger des règles CSS (typographie, bordures, tailles…)

Cet exercice utilise le document HTML intitulé temp.html.

Au fil des diverses étapes de l’exercice, vous allez ajouter quelques spécifications de style aux divers éléments de ce document qui présente les températures mensuelles moyennes dans cinq villes.

## Étape 1 : style inline

Ouvrez le document temp.html à la fois dans l’éditeur de texte et dans Firefox.

Commencez par associer un style au titre principal de la page (à savoir le contenu de la balise <h1>). Pour ce faire, ajoutez un attribut « style » à la balise, comme indiqué ci-dessous.

<h1 style="color:red; font-size:250%; border-bottom:1px solid red">

Sauvegardez le fichier et observez la différence en rafraichissant l’affichage dans Firefox.

Une remarque générale pour cette étape et toutes les suivantes : quand l’énoncé vous demande d’ajouter plusieurs propriétés de style, n’hésitez pas à les ajouter une par une en testant la page sur Firefox à chaque fois, pour bien identifier ce que chacune des propriétés fait et pourquoi les autres propriétés ajoutées sont nécessaires.

## Étape 2 : couleurs

Les étapes suivantes vont examiner de manière plus détaillée chacune des trois propriétés CSS ajoutées dans l’exemple ci-dessus.

**color** : il s’agit de la couleur du texte ; ici, on a indiqué la valeur désirée sous la forme d’un mot-clef (red). Ce n’est pas la seule manière de définir une couleur (il existe d’autres méthodes comme : un code hexadécimal, des composantes rgb, des composantes hsl, éventuellement avec un paramètre d’opacité).

Note : indiquer une couleur en CSS

*Si c’est votre première utilisation de CSS, dans un premier temps, vous  
pouvez passer cette note sur la manière d’écrire les couleurs en CSS et  
y revenir plus tard !*

Comme le CSS vise à préciser la manière dont le contenu d’une page web  
sera affiché, il est important de savoir comment indiquer la couleur exacte désirée. Plusieurs syntaxes sont possibles. En voici quelques-unes.

* *Via un mot-clef.* Plusieurs couleurs prédéfinies sont associées à des termes anglais qui peuvent être utilisés directement en CSS (comme red dans l’exemple ci-dessus). Pour une liste plus complète des mots autorisés, voir par exemple <https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>.
* A pink rectangle with a white border

  Description automatically generated*Via un code hexadécimal.* Une manière standard de définir une couleur consiste à indiquer ses composantes en termes des couleurs primaires rouge, vert et bleu (Red-Green-Blue ou RGB). Chacune de ces composantes est généralement représentée par un nombre entre 0 et 255, c’est-à-dire entre 00 et FF en écriture hexadécimale.  
    
  Par exemple, on peut obtenir du rose en utilisant les  
  composantes Rouge 255, Vert 0, Bleu 153. En hexadécimal,  
  cela correspond aux nombres FF, 00 et 99.  
    
  Le code hexadécimal d’une couleur se note avec le symbole #, qui peut être suivi d’une ou plusieurs valeurs hexadécimales accolées :  
  - #FF0099 en indiquant les composantes RGB ;  
  - #F09 en abrégé quand chaque composante est composée d’un chiffre dédoublé ;  
  - #FF009977 en ajoutant une quatrième composante hexadécimale pour la transparence (composante alpha) : 00 pour une couleur complètement transparente, FF pour une couleur complètement opaque.
* *Via ses coordonnées RGB.* Les composantes RGB peuvent également être écrites en « notation fonctionnelle » (l’écriture ressemble à un appel de fonction) : rgb(255 0 153) pour le rose donné en exemple ci-dessus, ou encore rgb(255 0 153 / .2) si on désire préciser une valeur d’alpha/de transparence (entre 0 et 1). Les composantes peuvent aussi se noter en pourcentages ; par exemple : rgb(100% 0 60% / 20%).
* *A diagram of a color spectrum

  Description automatically generatedVia ses coordonnées HSL.* Le système HSL (Hue – Saturation – Lightness) est une autre méthode permettant de décrire une couleur  
  via sa teinte (hue), sa saturation (couleur vive vs  
  couleur terne et grise) et sa luminosité (couleur claire  
  vs couleur foncée).

Il existe encore d’autres méthodes, y compris des  
options pour calculer le mélange de deux couleurs ou  
encore obtenir un gradient de couleurs. Vous  
pourrez trouver plus d’information à ce sujet sur le  
site MDN, entre autres à l’adresse suivante :  
<https://developer.mozilla.org/en-US/docs/Web/CSS/color_value>.

Deux autres sites intéressants pour l’utilisation des couleurs sur le web :

* <https://htmlcolorcodes.com/>
* <https://coolors.co/> (générateur de palettes de couleurs)

## Étape 3 : Taille de la police d’écriture

**font-size** : il s’agit de la taille d’écriture ; ici, on a indiqué qu’on désirait une taille correspondant à 250% de la taille d’écriture standard.

Pour cette propriété aussi, on peut exprimer la valeur de différente manière ; entre autres :

* pourcentage de la taille standard (ou plutôt, de la taille d’écriture du parent),
* taille absolue (par exemple en pixels, comme 12px, mais il existe également d’autres unités de mesure),
* taille relative en utilisant l’unité em pour se baser sur la taille d’écriture du parent ou l’unité rem pour se baser sur la taille d’écriture de base du document (root em) ; par exemple : 1.5rem pour 1,5 fois la taille d’écriture de base,
* en utilisant un mot-clef qui décrit une taille absolue calculée sur base de la taille par défaut du document :   
   xx-small x-small small medium large x-large xx-large
* en utilisant un mot-clef qui décrit une taille relative par rapport au parent (élément HTML englobant) :   
   smaller larger

Pour bien comprendre les différences, intéressez-vous à la structure <dl> (description/definition list) qui présente les cinq villes. Notez qu’elle utilise une balise <dl> englobant plusieurs éléments, dont des <dt> (definition term, termes définis) et des <dd> (definition description). La balise / l’élément <dl> est donc le parent des éléments <dt> et <dd> qu’elle contient.

1. Ajoutez un style inline pour le texte descriptif d’Anchorage avec la propriété font-size : larger.
2. Faites de même pour le texte descriptif d’Auckland avec la propriété font-size : large.
3. Les deux textes descriptifs ont la même taille. La taille large correspond à une taille supérieure à la taille standard (medium) des lettres du document. La taille larger indique une taille agrandie par rapport à la taille d’écriture du parent… et, comme cette taille est également medium, le résultat est identique dans les deux cas.
4. Pour mettre en évidence la différence, ajoutez cette fois-ci un style inline à la balise <dl>, à savoir : font-size : x-small. Observez le résultat et assurez-vous de bien comprendre pourquoi les tailles sont différentes désormais. Une fois que c’est fait, vous pouvez enlever toutes les modifications apportées par cette étape.

## Étape 4 : bordures

**border-bottom** : il s’agit de la bordure inférieure (ce qui peut correspondre à un effet de souligné) ; ici, on a demandé une bordure de 1 pixel de large, en trait plein de couleur rouge.

Il s’agit d’une propriété composée : elle accepte plusieurs valeurs, à savoir l’épaisseur du trait, son style et sa couleur.

Chacun de ces paramètres pourrait être indiqué individuellement. Ainsi, on aurait pu obtenir le même style en écrivant

border-bottom-width:1px;

border-bottom-style:solid;

border-bottom-color:red

La propriété border-bottom offre simplement le moyen d’écrire tout cela en une seule fois. On retrouve assez souvent ce système de « propriété composite » dans la syntaxe CSS.

La valeur de l’épaisseur du trait peut s’exprimer sous la forme d’une quantité de pixels (1px) ou encore en utilisant un mot-clef tel que thin (fin), medium (moyen, valeur par défaut) ou thick (large).

Pour indiquer la couleur, on dispose des mêmes options que ci-dessus pour la couleur du texte.

Finalement, pour le style, on peut utiliser les mots-clefs suivants : none (aucune bordure), solid (trait plein), dotted (en pointillé), dashed (en tirets), double (ligne double)

Avant de passer à la suite, modifiez le style inline de la balise h1 pour obtenir les résultats suivants :

1. un titre écrit en bleu, avec des caractères de taille xx-large, avec une bordure inférieure en tirets de couleur olive et de 3 pixels d’épaisseur ;
2. un titre écrit en marron, avec des caractères de 32 pixels et avec deux traits oranges pleins de 2 pixels d’épaisseur, un au-dessus du texte et un en-dessous (utilisez la propriété border-top) ;
3. un titre écrit en vert, avec des caractères dont la taille est le double de la taille standard, encadré sur les 4 côtés par des traits en pointillés de 4 pixels d’épaisseur et de couleur grise (utilisez border pour fixer l’apparence des 4 bordures en une fois).

## Étape 5 : style interne

Pour diverses raisons explicitées lors de la présentation théorique, il vaut mieux réserver le style « inline » pour des petits tests lors de l’élaboration d’un document. Une fois le style décidé, il est préférable de rassembler toutes les informations soit dans une balise <style> (version interne) soit dans un fichier séparé (version externe).

Dans le document HTML, et plus précisément dans la section <head>, ajoutez une balise <style> (ainsi que la balise fermante correspondante). Dans celle-ci, copiez la règle CSS suivante.

h2 { text-transform:uppercase; font-style:italic; color:olive }

Observez la manière dont le document est désormais affiché sous Firefox.

## Étape 6 : CSS et typographie

Examinez la règle ajoutée dans la balise <style>. La règle commence par un sélecteur (ici : h2) qui indique à quels éléments HTML elle correspond. Dans cet exemple, la règle cible tous les titres de balise <h2>. Ensuite vient, entre accolades, la liste des propriétés à appliquer.

**text-transform** : cette propriété permet d’imposer une transformation des caractères au niveau des majuscules et minuscules. La valeur associée peut être entre autres

* uppercase (tout en majuscules, comme dans l’exemple),
* lowercase (tout en minuscules),
* capitalize (la première lettre de chaque mot en majuscule) ou
* none (aucune transformation).

**font-style** : cette propriété permet de choisir le style de police utilisé. Les valeurs les plus utilisées sont

* normal (style normal) et
* italic (italique).

Pour demander l’affichage de caractères en gras, il faut utiliser la propriété **font-weight** ; celle-ci accepte (entre autres) comme valeurs

* normal (épaisseur normale),
* bold (gras),
* un nombre parmi 100, 200, 300, 400 (normal), 500, 600, 700 (bold), 800 et 900,
* un adjectif parmi lighter et bolder pour se baser sur l’épaisseur du parent.

Avant de passer à la suite, effectuez les 5 transformations suivantes.

1. Arrangez-vous pour que tous les titres <h2> soient soulignés par une ligne double de 4 pixels d’épaisseur et de couleur bleue.
2. Arrangez-vous pour que tous les titres <h2> soient écrits en utilisant une police de type Verdana (utilisez la propriété font-family:Verdana).
3. Créez une nouvelle règle grâce à laquelle tous les titres de colonnes ou de lignes dans les tableaux seront écrits en bleu, non gras, et entièrement en majuscules.
4. Créez une nouvelle règle grâce à laquelle toutes les cellules normales des tableaux (c’est-à-dire pas les titres) auront un fond gris clair (utilisez la propriété background-color et la couleur lightgray).
5. Modifiez la règle précédente pour que chacune des cellules ait une largeur (propriété width) de 100 pixels ; faites également en sorte que le texte y soit centré (propriété text-align:center).

Note : les polices d’écriture

Si c’est votre première utilisation de CSS, dans un premier temps, vous  
pouvez passer cette note et y revenir plus tard !

La propriété font-family permet d’indiquer quelle police d’écriture utiliser pour un texte. Elle peut recevoir trois types de valeurs : le nom d’une police bien précise, un type générique de police ou encore une liste d’options séparées par des virgules (dans ce cas, la première option utilisable sera choisie, en fonction des polices disponibles au navigateur).

* *Nom de police.* Si le nom comporte plusieurs mots séparés par des espaces, il faut l’entourer de guillemets.  
  Par exemple : font-family: "brush script mt"
* *Type de police générique.* Les polices d’écriture sont classées en plusieurs grandes catégories. On distingue notamment les polices  
  - serif et sans-serif : les polices serif comportent des caractères avec des empattements (voir illustrations) alors que ce n’est pas le cas des sans-serif ;  
  - monospace : les polices généralement utilisées pour du code, où chaque caractère occupe la même largeur (un i et un m par exemple) ;  
  - cursive : les polices ressemblant à une écriture manuscrite.  
  Chacun des mots écrits en monospace dans la liste ci-dessus peut être utilisé comme valeur pour la propriété font-family.
* *Liste d’options.* Pour couvrir tous les cas et résoudre les problèmes d’accessibilité aux polices d’écriture, on peut citer plusieurs valeurs. Par exemple, font-family: Georgia, serif indique d’utiliser la police Georgia de préférence et, si elle n’est pas disponible, n’importe quelle police serif.

A black background with red circles

Description automatically generated

A close-up of a logo

Description automatically generated

Notez que Google met à disposition plusieurs polices d’écriture (les « Google Fonts ») qui peuvent être assez facilement intégrées dans une page web. La liste des polices est disponible sur <https://fonts.google.com/>. Pour pouvoir utiliser une telle police dans un document web, il faut ajouter une base <link> dans l’en-tête du document HTML. Un exemple est des explications plus précises sont disponibles sur <https://developers.google.com/fonts/docs/getting_started>.

## Étape 7 : identificateur

On peut avoir besoin de créer une règle CSS qui ne cible pas forcément tous les éléments HTML qui utilisent une balise donnée. Dans la section « Introduction » de la page HTML se trouve par exemple un premier paragraphe (« Le tableau… ») qui présente le tableau en bas de page. Que faire pour cibler ce paragraphe de présentation et uniquement lui ?

Tout d’abord, ajoutez au paragraphe en question un attribut id (pour identificateur) qui lui associe l’identificateur intro. Ensuite, créez une règle CSS ciblant cet élément unique. Dans la règle, le sélecteur correspondant à « l’élément dont l’identificateur est intro » s’écrit #intro.

Rédigez la règle CSS pour que ce paragraphe d’introduction soit affiché en italique et en gris.

## Étape 8 : classe

Au lieu de cibler un unique élément HTML (via son identificateur), une règle CSS peut également cibler plusieurs éléments HTML appartenant à la même classe. Dans le tableau des températures se trouvent plusieurs valeurs négatives… on pourrait les mettre en évidence en les écrivant en gras et en rouge.

Tout d’abord, ajoutez à chacune des cellules contenant un nombre négatif la classe negatif (il y en a 8 en tout) :

<td class="negatif">-9,5</td>

*Remarque* : ce genre de modifications dans un fichier texte peut s’effectuer de manière très rapide si on maîtrise bien les divers outils offerts des éditeurs. Par exemple, vous pourriez vous contenter d’effectuer un « Replace » pour remplacer toutes les occurrences de « <td>- » en « <td class="negatif">- » dans le document.

Rédigez ensuite une règle CSS ciblant tous les éléments ayant la classe negatif et imposant qu’ils soient écrits en rouge et en gras. Le sélecteur correspondant à « tous les éléments ayant la classe negatif » s’écrit en CSS .negatif.

## Étape 9 : sélecteurs CSS

Un point de la situation sur les sélecteurs…

|  |  |  |
| --- | --- | --- |
| Pour cibler… | on écrit… | exemple |
| tous les éléments de même balise | nomBalise | h2 |
| un élément d’identificateur donné | #identificateur | #intro |
| tous les éléments ayant une classe donnée | .nomClasse | .negatif |

## Étape 10 : CSS et contrôle des tailles

La section « Introduction » contient une description des cinq villes dont on a relevé les températures. Le premier objectif est d’afficher les noms des villes en gras et en couleur olive.

Notez que, si cette partie du contenu avait été encodé en HTML sous la forme de simples paragraphes, il aurait fallu ajouter des balises <span> autour des noms des villes pour pouvoir les cibler avec une règle CSS (comme dans le laboratoire précédent). Ici, ce n’est pas nécessaire car le contenu a déjà été structuré avec des balises <dl>, <dt> et <dd>. Si vous êtes curieux, n’hésitez pas à aller jeter un coup d’œil sur la documentation de ces balises sur le site MDN (Mozilla Developer Network), entre autres à la page <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dl>.

1. Ajoutez une règle CSS ciblant les noms des villes afin qu’elles soient écrites en couleur olive, en gras et avec une taille agrandie par rapport à la normale.
2. Ajoutez une règle CSS ciblant les descriptions des villes afin qu’elles possèdent une bordure gauche de 1,5 pixels en trait plein de couleur olive.
3. Pour séparer la bordure du texte auquel elle est collée, ajoutez à la règle la propriété suivante : padding-left:4px.
4. Pour réduire l’espacement à gauche des descriptions, ajoutez également la propriété margin-left:20px.
5. Finalement, vous pouvez également influencer la largeur sur laquelle le texte est écrit en ajoutant la propriété width:600px. Observez son effet en réduisant la largeur de la fenêtre Firefox.
6. Modifiez la dernière propriété en remplaçant width par max-width et, à nouveau, observez l’effet quand vous augmentez/réduisez la largeur de la fenêtre d’affichage.

## Étape 11 : aperçu de sélecteurs plus avancés

*Note.* Cette étape et la suivante donnent un aperçu de certains sujets qui sont approfondis dans les cours suivants.

Dans le tableau des températures, la dernière ligne présente les moyennes annuelles pour chacune des villes. Pour plus de clarté, il faudrait distinguer ces valeurs du reste du tableau. Dans un premier temps, ajoutez un identificateur moyenne à la dernière ligne du tableau (donc, sur la balise <tr>).

Supposons qu’on désire cibler toutes les cellules de cette dernière ligne, pour imposer que leur texte soit écrit en gras et pour leur ajouter une bordure supérieure (de 2 pixels d’épaisseur, en trait plein de couleur grise).

Il faudrait donc une règle qui cible tous les <td> (ainsi que le <th>) qui se trouvent à l’intérieur du <tr> dont l’identificateur est moyenne.

À cette fin, on peut utiliser le sélecteur #moyenne td. Quand deux sélecteurs (#moyenne et td ici) sont séparés par un espace, cela signifie qu’on cible tous les <td> (le second sélecteur) qui se trouvent à l’intérieur d’un #moyenne (le premier sélecteur).

Ajoutez la règle adéquate.

#moyenne td { … (ici les propriétés) … }

Pour que ce style s’applique également à la cellule-titre « MOY », il faudrait aussi cibler le <th> à l’intérieur de #moyenne. Une première solution consiste à ajouter une autre règle (avec les mêmes propriétés) qui commencerait par #moyenne th.

Mais on peut également écrire une règle qui s’applique à plusieurs sélecteurs. Cela se fait en séparant les sélecteurs par des virgules :

#moyenne td, #moyenne th { … (ici les propriétés) … }

## Étape 12 : ordre des règles CSS (premier aperçu)

Dans les règles CSS de votre document devraient se trouver, entre autres, une règle ciblant tous les <td> (et imposant une couleur de fond gris clair) et une règle ciblant tous les .negatif.

Modifiez la règle ciblant les .negatif pour imposer que les cellules comportant des températures négatives aient un fond gris très clair. Il n’y a pas de mot-clef de couleur correspondant à « encore plus clair que lightgray ». Aussi, utilisez plutôt le code hexadécimal #EEEEEE.

… background-color:#EEEEEE …

Enregistrez et observez le résultat.

Dans votre fichier HTML, intervertissez l’ordre des règles sur td et sur .negatif et observez le résultat.

Dans les deux cas (et donc quel que soit l’ordre des règles), le navigateur est suffisamment « malin » pour comprendre que « colorer le fond en gris clair » est la règle générale et « colorer le fond des températures négatives en gris très très clair » est une règle particulière et que c’est cette dernière qui doit prendre le pas sur la première en cas de conflit.

La manière exacte dont le navigateur résout ces conflits de règles sera étudiée dans un cours ultérieur mais, pour l’instant, on peut se contenter de retenir que « plus la règle concerne un cas précis » et plus elle sera considérée importante.

# (exo) mini-recueil d’énigmes

Objectifs

* définir un style de manière externe
* structurer le code HTML pour pouvoir construire des règles CSS
* mettre son site en ligne

Cet exercice se base sur le document puzzles.html. Le but est à nouveau de définir divers styles pour rehausser la présentation mais, dans cet exercice, tous les styles seront définis dans un fichier séparé.

## Étape 1 : fichier de style externe

Ouvrez le fichier puzzles.html (ou une copie) dans l’éditeur et dans Firefox.

Créez également un second fichier texte, puzzles.css, qui contiendra les règles CSS et les définitions de style. Liez le fichier puzzles.css au document puzzles.html en ajoutant une balise <link> dans le document HTML.

Organisez vos fichiers en dossiers :

* À la racine : puzzle.html ;
* Dans un dossier style : puzzle.css ;
* Dans un dossier images : gateau.jpg.

Sauf mention contraire, tous les styles demandés ci-dessous devront être définis dans ce fichier puzzles.css (même si vous pouvez, dans un premier temps, tester l’écriture des propriétés de manière inline si vous le désirez).

## Étape 2 : structure du document

Prenez quelques minutes pour parcourir le document HTML, comprendre son contenu et sa structure. Observez qu’il se décompose assez naturellement en trois parties : une introduction (avec un titre <h1> et un paragraphe descriptif), le contenu principal (avec une série d’énigmes, chacune composée d’un titre, d’un énoncé puis d’une solution), puis finalement un dernier paragraphe citant les sources.

Pour faire apparaître cette structuration en trois parties dans le code HTML, utilisez les balises <header>, <main> et <footer> pour encadrer chacune des sections. À l’intérieur de la balise <main>, placez chaque énigme dans une section <article>.

Tous ces changements techniques ne devraient rien changer à l’apparence du document ; les balises ajoutées sont là pour (a) rendre plus visible la structure du document et (b) permettre la construction de règles CSS ciblant les sections ainsi identifiées.

N’oubliez pas de peaufiner l’indentation !

## Étape 3 : header, main et footer

Construisez des règles CSS pour effectuer la mise en page au niveau « global » (c’est-à-dire pour les 3 parties principales <header>, <main> et <footer>). Par exemple :

* Créez tout d’abord une règle ciblant la balise <body>. Dans celle-ci, vous demanderez un fond noir (propriété background-color).
* Créez ensuite des règles pour les balises <header>, <main> et <footer> imposant à chacune de ces parties d’avoir une largeur de 800 pixels et un fond de couleur jaunâtre (couleur lemonchiffon). Pour centrer horizontalement chacun de ces trois blocs, ajoutez les propriétés suivantes : margin-left:auto; margin-right:auto.  
  *Note.* Le procédé décrit ci-dessus pour fixer la largeur permet de s’assurer que l’affichage du contenu se fera sur une largeur donnée, quelle que soit la taille de la fenêtre. Il comporte toutefois un inconvénient important : si la taille convient à la vision du site web sur un PC, il y a de grandes chances pour que le rendu soit problématique sur un GSM ou une tablette. Ces considérations seront abordées plus tard dans le cours. Pour le moment, vous pouvez vous contenter de prendre en compte uniquement l’affichage sur écran de PC.
* Si une règle CSS comporte de nombreuses propriétés CSS, pensez à utiliser une indentation plus claire en ne plaçant qu’une seule propriété par ligne. Par exemple, remplacez la première version ci-dessous par la seconde.

selecteur { prop1:val1; prop2:val2; prop3:val3 }

selecteur {

prop1:val1;

prop2:val2;

prop3:val3

}

* Pour éviter que le texte ne se retrouve trop près des bords des cadres, vous pouvez ajouter la propriété padding:6px.  
  *Rappel.* Vous pouvez écrire un ensemble de propriétés qui s’appliquent à plusieurs sélecteurs à la fois en séparant ces sélecteurs par des virgules. Dans ce cas-ci, plutôt que d’écrire 3 règles (une pour header, une pour main et une pour footer), vous pouvez utiliser le sélecteur header, main, footer.
* Écrivez une règle pour que le titre principal (<h1>) soit écrit avec une taille de caractères de 200% sur un fond (background-color) de couleur #B1CF5F. Là aussi, pour éviter que les lettres ne collent aux bords du cadre orange, vous pouvez ajouter un padding/rembourrage de 3 pixels.
* Pour séparer légèrement la partie header du contenu principal, utilisez la propriété margin-bottom:4px pour la section <header>. Cette « marge » demande une séparation entre ce bloc et le reste de la page, ce qui laissera apparaître la couleur noire de l’élément <body>.  
  *Note.* Si vous avez rassemblé en une seule règle toutes les propriétés communes à header, main et footer, vous pouvez tout simplement créer une nouvelle règle (ne ciblant que header) pour ajouter la propriété margin-bottom :

header, main, footer { … propriétés communes aux trois … }

header { … propriétés supplémentaires pour header seulement … }

* Occupez-vous ensuite de la partie footer. Pour la séparer, ajoutez une marge supérieure (margin-top) de 4 pixels. Assurez-vous également que le texte soit écrit en gris, italique et centré horizontalement.
* Transformer également l’URL contenue dans le footer en lien… vous vous souvenez comment faire, n’est-ce pas ? Et ajoutez une règle CSS pour faire en sorte que le texte du lien soit écrit en #1B512D.

## Étape 4 : les titres des énigmes

Intéressons-nous maintenant aux différentes énigmes et à leur titre.

* Commençons par la balise <article>. Pour séparer les différentes énigmes, tentez tout d’abord d’utiliser la même propriété que pour <header> et <footer>, à savoir margin-bottom:4px. Pourquoi cela n’ajoute-t-il pas des lignes noires entre les énigmes comme avant ?
* Pour distinguer les énigmes, ajoutez une règle ciblant la balise article et faisant en sorte que chaque énigme soit encadrée (sur les 4 côtés) par des traits en pointillés noirs de 2 pixels d’épaisseur. Pour que les cadres ainsi créés ne soient pas collés les uns aux autres, ajoutez une marge supérieure (margin-top) et une marge inférieure (margin-bottom) de 12 pixels. Pour que le texte à l’intérieur ne soit pas collé contre les bords, ajoutez un rembourrage (padding) de 4 pixels.



* Écrivez maintenant une règle CSS pour les titres des énigmes (<h2>). L’objectif est d’afficher chacun des titres sur une bande de couleur #1B512D avec le symbole de l’ampoule qui se trouve dans le fichier ampoule.png (à ranger dans le dossier images). Au passage, pourquoi a-t-on utilisé ici le format png plutôt que le format jpg ?  
    
  Pour ce faire, ajoutez les propriétés suivantes.

background-color:#1B512D;  
color:lemonchiffon;

background-image:url('ampoule.png');

background-repeat:no-repeat;

background-size:auto 100%;

padding-left:40px;

La propriété **background-image** permet d’indiquer l’image à utiliser comme arrière-fond. La valeur peut être none (pour aucune image) ou l’url de l’image à incruster. Dans les règles CSS, les url s’écrivent en les encadrant par url('…').  
  
Par défaut, l’image de fond est répétée de manière à recouvrir toute la zone. Comme, ici, on ne désire qu’une seule ampoule, on utilise la propriété **background-repeat** avec la valeur no-repeat. Les autres valeurs acceptées par cette propriété sont :

* + repeat-x (répétition uniquement vers la gauche/droite),
  + repeat-y (répétition uniquement vers le bas/haut) et
  + repeat (répétition dans toutes les directions).

Finalement, la propriété **background-size** permet d’indiquer la taille de l’image à incruster. Dans cet exemple, la largeur est calculée automatiquement mais la hauteur est fixée à 100% du cadre. Cela permet d’avoir une ampoule qui s’étend sur toute la hauteur de la bande colorée.

La dernière propriété, **padding-left**, permet de repousser le texte du titre vers la droite, de manière à ce qu’il ne soit pas au-dessus de l’ampoule.  
  
Pour rapprocher les titres du haut des cadres en pointillés, vous pouvez également ajouter la propriété margin-top:0 si vous le désirez.

## Étape 5 : les solutions des énigmes

Intéressons-nous aux solutions. Pour l’instant, ces solutions sont de simples paragraphes en fin d’énigme et ils sont toujours visibles. Le but est de les rendre invisibles jusqu’à ce que l’utilisateur passe le curseur de la souris au-dessus d’elles.

* Dans un premier temps, il faut se donner les moyens de cibler les solutions via une règle CSS. Les solutions sont de simples paragraphes… mais tous les paragraphes ne sont pas des solutions. Une première option consiste à utiliser une classe. Ajoutez à tous les paragraphes concernés la classe solution.
* Écrivez ensuite une règle CSS visant ces paragraphes et s’assurant qu’ils soient écrits en couleur lemonchiffon (la même couleur que le fond, pour les rendre invisibles). Pour qu’on puisse tout de même « voir » où les solutions se trouvent, ajoutez un cadre (sur les 4 côtés) en trait plein orange de 1 pixel d’épaisseur. Afin que le texte (lorsqu’il apparaîtra) ne soit pas collé aux bords, ajoutez un rembourrage de 4 pixels. Et, finalement, pour rapprocher le cadre de solution du bord inférieur de l’encadré d’énigme, ajoutez la propriété margin-bottom:0.
* Désormais, toutes les solutions devraient être masquées… Pour les rendre visibles lorsque le curseur survole (hovers over) le cadre, il faut rédiger une nouvelle règle CSS. Celle-ci utilisera le sélecteur .solution:hover, qui cible tous les éléments de classe solution mais n’est appliquée que lorsque le curseur se trouve au-dessus d’eux. Dans cette règle, imposez simplement que la couleur du texte soit le noir. Testez le tout sous Firefox.
* Testez l’affichage des solutions en les survolant avec le curseur.
* Ajoutez la propriété transition: color 2s à la règle ciblant les paragraphes-solutions. Celle-ci indique que les changements de la propriété color devraient être effectués sous la forme d’une transition qui prend 2 secondes. Observez son effet sur Firefox.

## Étape 6 : les énoncés des énigmes

Il ne nous reste plus qu’à nous occuper des énoncés des questions.

* Dans un premier temps, repérez l’image du gâteau… celle-ci est bien trop grande. Pour pouvoir modifier son apparence via la feuille de styles CSS, attribuez-lui tout d’abord un identificateur.
* Ensuite, utilisez cet identificateur pour créer une règle CSS tout spécialement pour cette image. Dans la règle, imposez une largeur de 200 pixels ; cela devrait être suffisant.
* Finalement, supposons qu’on veuille écrire tous les énoncés dans un style particulier (par exemple, en italique). Comment construire une règle CSS ? On pourrait, dans un premier temps, penser à donner une classe enonce à chacun des paragraphes en question. Cela risque cependant de poser un problème : on pourrait mettre le texte en italique mais, si on désirait également l’encadrer, on encadrerait en fait chaque paragraphe individuellement.  
    
  Pour faire les choses correctement, il faut donc rassembler chaque énoncé dans un « groupe ». Comme il s’agit d’un groupe de paragraphes, <span> ne convient pas ; il faut utiliser la balise générique <div>.  
    
  Dans un premier temps, emballez donc chaque énoncé (c’est-à-dire chaque groupe de paragraphes… et d’images) dans un <div> ayant enonce comme classe. Attention à bien conserver une indentation correcte ! Vérifiez également que vous n’avez pas oublié de balise fermante en vous assurant que le résultat n’a pas changé dans Firefox.
* Ensuite, rédigez une règle CSS ciblant les énoncés et imposant qu’ils soient écrits en italique.
* Ciblez les sept lignes de chiffres de l’avant-dernière énigme avec une règle CSS et faites en sorte qu’elles soient écrites de manière centrée, dans une fonte de type monospace, comme « Courier » et sans être en italique.

## Étape 7 : DivulgÂchage/spoiler (dépassement, optionnel)

Une autre manière de présenter les énigmes et leurs réponses consiste à utiliser les balises de divulgâchage/spoiler details et summary.

La balise details permet de créer une portion de page web constituée d’un bout de texte toujours affiché (le contenu de summary) et d’un symbole permettant d’afficher ou de cacher le reste du texte, qui est caché par défaut. L’idée ici serait d’afficher les énoncés par défaut et de n’afficher les réponses que quand l’utilisateur clique sur le symbole pour les dévoiler.

Explorez la documentation sur ces balises à l’adresse <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/details> si vous le désirez et transformez votre page web pour qu’elle utilise ce type de structure.

## Étape 8 : Mise en ligne (optionnel)

Pour rendre un site accessible, il faut placer les fichiers correspondants sur un serveur en ligne. La qualité et les performances (ainsi que les prix) des serveurs en ligne varient grandement. L’objectif ici n’est pas d’étudier dans le détail tous ces aspects de l’hébergement en ligne mais juste de pratiquer au moins une fois cette opération.

Vous pouvez utiliser n’importe quelle option d’hébergement gratuit ; une solution possible est 000webhost (<https://www.000webhost.com/>). C’est celle qui est décrite ci-dessous mais le fonctionnement d’autres services identiques devrait être similaire.

1. Commencez par vous rendre sur le site, connectez-vous avec votre compte Google (ou créez-en un si vous n’en avez pas encore). Choisissez l’option pour créer un site web gratuit.
2. Choisissez un nom de sous-domaine unique pour votre site (celui-ci fera partie de l’URL pour y accéder) – ne reprenez pas celui de la capture d’écran !  
     
   A screenshot of a facebook page

   Description automatically generated
3. Choisissez l’option permettant d’uploader un site (plutôt qu’un site partiellement préfabriqué) – l’option de droite sur l’image suivante.  
     
   Screens screenshot of a screen

   Description automatically generated
4. Vous arrivez sur la page permettant de gérer le contenu de votre site et d’y téléverser des fichiers (via la protocole FTP).
5. Entrez dans le répertoire public\_html et placez-y votre fichier HTML (a priori puzzles.html) ainsi que les autres dossiers nécessaires.  
     
   A screenshot of a computer

   Description automatically generated
6. Une fois que c’est fait, ouvrez un nouvel onglet et accédez à votre site en ligne. L’URL à utiliser est composée du nom de domaine que vous avez choisi à l’étape 3 suivi de .000webhostapp.com/ suivi du nom de votre fichier HTML.  
     
   
7. Revenez à l’onglet de gestion du site via FTP. Si vous l’avez fermé, vous pouvez y accéder en vous connectant sur le site 000webhost.com puis en sélectionnant votre site et en cliquant sur « Manage ». L’interface FTP se trouve sous l’option « File Manager » dans la section « Dashboard » « Tools » du menu de gauche.  
     
   A screenshot of a phone

   Description automatically generated
8. Rendez-vous dans le répertoire public\_html et renommez votre fichier puzzles.html en index.html (clic droit sur le fichier puis Rename). Si, dans une URL, aucun nom de fichier html n’est précisé, c’est par défaut un fichier nommé index.html qui va être chargé.
9. Rafraîchissez l’onglet dans lequel votre site était ouvert. Cela devrait produire une erreur 404 (fichier introuvable), vu que puzzle.html n’existe plus.
10. Modifiez l’URL en supprimant la dernière partie, pour qu’il ne reste plus que votre nom de domaine et 000webhostapp.com. Vous devriez ainsi retrouver votre site, vu que le fichier index.html sera chargé automatiquement.  
      
    