

Exercices CSS

Déclaration et validation

1. Ouvrez dans un éditeur de texte le fichier **css-premier.html**. Chargez simultanément ce document dans le navigateur et visualisez-le. Lisez en le contenu.
2. Consultez le code source et déduisez-en le rôle (càd la sémantique) de la balise HTML `<pre>`. Quelle est ce rôle ?
3. Vous avez certainement remarqué dans ce code source, la déclaration en commentaire de l'utilisation d'une feuille de style au sein de l'élément `<head>`. Dans l'éditeur supprimez ces marques de commentaire, sauvegardez et rechargez la page dans le navigateur pour constater l'effet de l'application de cette feuille de style au document.
4. Affichez le code source de la page HTML, vous pouvez alors accéder directement au code de la feuille de style utilisée **style1.css** en cliquant sur le nom du fichier dans sa déclaration dans l'entête de ce code source. Faites-le.
5. Rendez-vous sur la page du [validateur CSS du W3C](#), choisissez l'onglet permettant de proposer un fichier par chargement de fichier et vérifiez la validité CSS3 de la feuille de style **style1.css**.

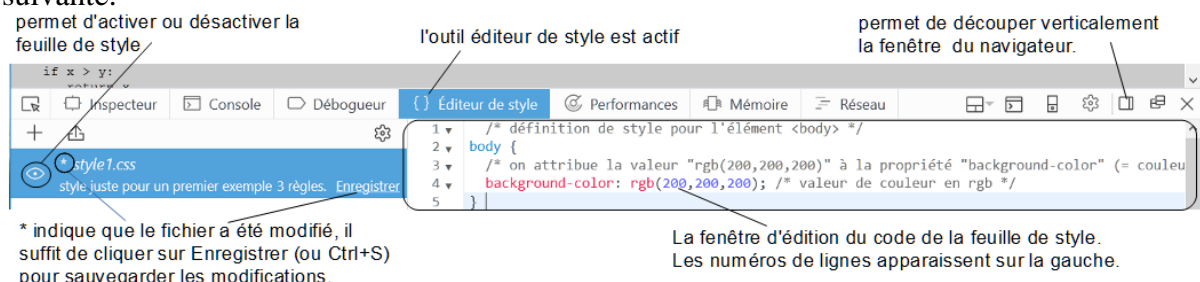
Premières propriétés et éditeur de style

S'il ne l'est plus, ouvrez à nouveau dans le navigateur le fichier **css-premier.html**. Nous allons découvrir des propriétés CSS et leur rôle.

1. Une fois un document HTML qui déclare l'utilisation d'une feuille CSS chargé dans le navigateur. Il est possible d'utiliser l'outil appelé **éditeur de styles** disponible par **Outils** → **Développeur Web** → **Editeur de styles** (ou le raccourci `Shift+F7`). Celui-ci permet de consulter et modifier les feuilles de style et de constater en direct l'impact des modifications avant de les enregistrer.

Il peut être plus pratique de diviser la fenêtre du navigateur verticalement plutôt qu'horizontalement. Il y a un bouton qui permet cela dans la barre d'outils de l'inspecteur.

Ouvrez cet outil, vous devriez obtenir quelque chose qui ressemble à l'image suivante.



Vue de l'outil **Editeur de style**.

Nous allons utiliser cet outil pour découvrir quelques propriétés CSS. Utilisez le bouton indiqué pour diviser verticalement la fenêtre du navigateur, vous devriez avoir sur la gauche le document HTML affiché et sur la droite l'éditeur de style actif.

2. Cliquez sur l'œil à côté du nom du fichier CSS pour activer/désactiver la feuille de style. Constatez l'effet dans la fenêtre du navigateur. A la fin laissez la feuille active.
3. Dans l'éditeur de style, repérez les déclarations des règles CSS qui s'applique à l'élément `<body>` et aux éléments `<h1>`. Faites le lien entre ces règles et les effets visibles de l'application de la feuille de style dans la zone d'affichage du navigateur.
4. Nous allons nous intéresser aux règles CSS qui s'appliquent aux éléments `<h2>`. Il s'agit des lignes 16 à 21 de la feuille de style. Pour le moment les déclarations des règles sont mises en commentaire entre les couples de signes `/*` et `*/`.

Supprimez les commentaires (`/*` et `*/`) à la ligne 17 pour activer la règle. Constatez le changement dans la zone d'affichage pour tous les éléments `<h2>`. Déduisez en l'effet (la sémantique) de la propriété CSS **color**.

Procédez de la même manière pour chacune des autres lignes 18 à 20 pour identifier le rôle des propriétés utilisées.

Vérifiez vos réponses en étudiant la documentation sur ces différentes propriétés à partir de [cette page](#) le site du MDN.

Sauvegardez la feuille de style ainsi modifiée pour conserver les changements.

5. Supprimez les marques de commentaires aux lignes 25 et 32 pour activer les règles s'appliquant aux éléments `<p>`. Déduisez-en le sens des propriétés qui apparaissent dans ces règles.

Si dans l'outil éditeur de style vous survolez à la souris le nom de l'élément auquel s'applique la règle (le sélecteur), alors dans la zone d'affichage du navigateur les éléments correspondants, c'est-à-dire ceux auxquels d'applique la règle, sont mis en évidence. Essayez en plaçant votre souris quelques instants au-dessus du sélecteur de règle `h2` puis `p`.

Continuez ensuite à découvrir de nouvelles propriétés ou valeurs CSS en décommentant progressivement le code des règles dans la suite de la feuille de style qui s'appliquent aux éléments `<pre>`, `` et `<abbr>`.

6. Sauvegardez la feuille de style CSS modifiée puis rendez-vous sur la page du [validateur CSS du W3C](#), choisissez l'onglet permettant de proposer un fichier par chargement de fichier et vérifiez la validité CSS3 de la feuille de style **style1.css** (il se trouve dans le dossier **style**).

Ayez le réflexe console

Lorsque les choses ne semblent pas se passer comme ce que vous attendiez, il est bon d'avoir pour premier réflexe d'aller regarder si des erreurs sont signalées dans la console web.

Prenons un exemple...

1. Ouvrez dans un éditeur de texte et dans le navigateur le fichier **css-premier-probleme.html**. Vous constatez qu'aucun élément de style ne s'affiche malgré la déclaration d'une feuille de style.
2. Ouvrez l'outil de développement Console web : `Shift+Ctrl+K` ou dans le menu **Outils** choisissez **Développeur web** et ensuite **Console web**. Vous pouvez constater que deux messages d'erreur sont signalés dans cette console (il sera peut-être nécessaire d'activer l'affichage de ces messages en cochant dans la liste des options du bouton **CSS** de la console les choix **Erreurs** et **Avertissements**). Lisez-les puis corrigez les erreurs dans le fichier **style1-probleme.css** une par une (pensez à utiliser l'éditeur de style pour faire ces corrections).

Premières règles de calcul des valeurs de propriétés.

Poursuivons avec le fichier **css-premier-bis.html** et sa feuille de style **style1-bis.css**.

Pour accéder aux valeurs par défaut des propriétés de style, il suffit de désactiver la feuille de style utilisée en utilisant le bouton représentant un œil à côté du nom du fichier de la feuille de style dans l'outil éditeur de style.

1. Quelle est couleur par défaut (c'est-à-dire sans utilisation de la feuille de style) du texte pour l'élément `<h2>` ?
Quelle est cette valeur lorsque l'on utilise la feuille de style ?
Que peut-on en déduire pour les propriétés pour lesquelles aucune valeur n'est fixée dans la feuille de style ?
2. Quelle est la valeur de la couleur d'arrière-plan affichée par défaut pour l'élément `<p>` ?
Quelle est cette valeur lorsque l'on utilise la feuille de style ?
Pourquoi cette valeur ? Où est-elle définie ?
Que peut-on en déduire pour les valeurs de propriétés définies dans la feuille de style ?
3. Quelle est la valeur par défaut de la couleur de texte pour l'élément `<abbr>` ?
Quelle est cette valeur lorsque l'on utilise la feuille de style ?

Pourquoi cette valeur ? Où est-elle définie ?

Que peut-on en déduire pour les valeurs de propriétés quand elles ne sont pas redéfinies au niveau d'un élément emboîté ?

4. Vérifiez que dans le document, les éléments `` sont toujours emboîtés dans des éléments `<p>`.

Quelle est la valeur de la couleur de texte définie pour l'élément `<p>` dans la feuille de style ?

Quelle est la valeur de la couleur définie pour l'élément `` dans la feuille de style ?

Quelle est la valeur de la couleur affichée ?

Pourquoi cette valeur ? Que peut-on déduire pour les valeurs de propriétés quand elles sont redéfinies au niveau d'un élément emboîté ?

5. Vous allez modifier le style défini dans **style1-bis.css** en utilisant l'outil *Editeur de styles*. N'oubliez pas de sauvegarder vos modifications.

Soit le bloc de déclaration suivant :

```
p {  
    background-color : #F5F5DC;  
}
```

1. Ajoutez ce bloc dans le fichier **style1-bis.css** avant l'actuel bloc portant sur l'élément `<p>` et testez.
 2. Recommencez en le plaçant **après** l'actuel bloc et testez à nouveau.
 3. Que constatez-vous ? Que peut-on déduire sur la valeur d'une propriété utilisée quand il existe plusieurs définitions pour un même élément ?
6. A partir des conclusions des différentes questions précédentes, rédigez un texte qui présentent les règles qui semblent gérer les valeurs des propriétés utilisées pour l'affichage.

Ces règles seront précisées et affinées au fur et à mesure.

Couleurs

On trouve dans le fichier **style1.css** les valeurs des couleurs exprimées de trois manières : un *nom*, une *valeur hexadécimale*, un *triplet rgb* (parfois appelée *valeur fonctionnelle*).

Il existe un ensemble de noms de couleur définis. Chacun correspond à une valeur de couleur hexadécimale. Vous en trouverez la liste avec le code hexadécimal correspondant [ici](#).

Dans la valeur hexadécimale, les deux premiers chiffres correspondent à la valeur hexadécimale de la composante *rouge*, les deux suivants à celle de la composante *verte* et les deux derniers à celle de la composante *bleu*.

Les trois valeurs entières d'un triplet *rgb* correspondent également dans l'ordre aux valeurs des trois composantes *rouge*, *verte* et *bleu* dans cet ordre.

Les valeurs hexadécimales présentes dans le fichier sont sur six caractères. Il existe aussi pour les valeurs des couleurs des valeurs hexadécimales sur trois caractères. Elles correspondent à la valeur hexadécimale à six caractères dans laquelle chacun des caractères est répété deux fois. Ainsi la couleur de valeur #1A0 est la même que celle de valeur #11AA00.

*En plus des valeurs *rgb*, on trouve des valeurs sous forme de quadruplet avec la fonction *rgba* qui ajoute au triplet *rgb* une composante *a* (pour canal alpha) qui exprime l'opacité de la couleur. Sa valeur va de 0.0 (totalement transparent) à 1.0 (totalement opaque). Pour une couleur exprimée par triplet *rgb* le canal alpha vaut 1.0. Donc la valeur *rgb(x,y,z)* est la même que *rgba(x,y,z,1.0)*.*

1. Sachant que les notations hexadécimales et *rgb* sont équivalentes. Quel est l'intervalle de valeurs autorisées pour les éléments d'un triplet *rgb* ?
Par des tests, étudiez comment agit le navigateur si l'on donne des valeurs en dehors de cet intervalle.

Les valeurs dans le triplet peuvent également être exprimées sous forme de pourcentage. 0% correspond à la borne inférieure de l'intervalle et 100% à sa valeur maximale.

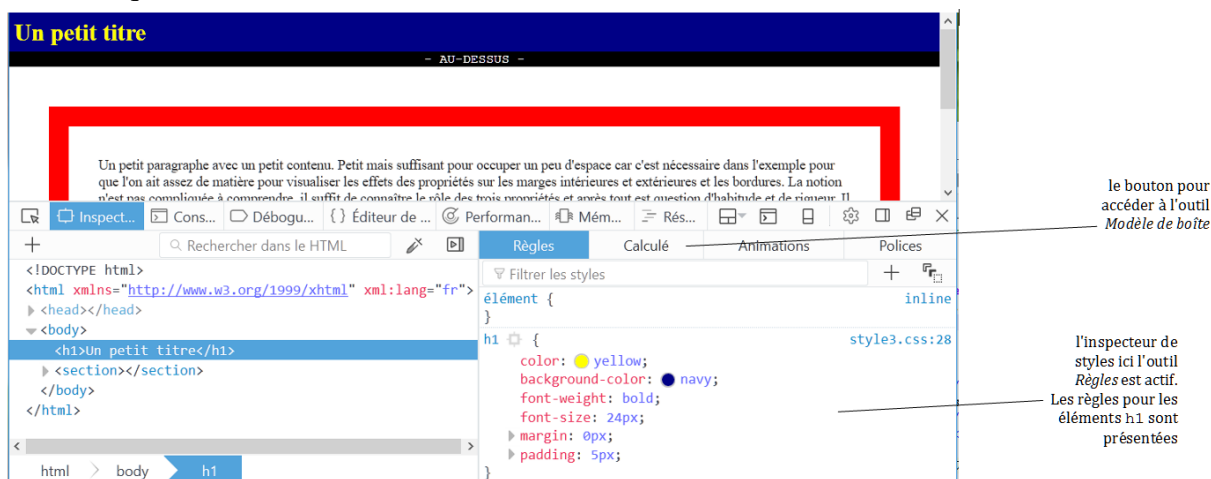
2. Calculez la valeur hexadécimale de toutes les couleurs présentes dans le fichier **style1-bis.css**.
Vérifiez visuellement vos valeurs en utilisant l'outil *Editeur de styles* pour remplacer dans le fichier **style1-bis.css** les valeurs de toutes les couleurs par ces valeurs hexadécimales.
3. Faites de même pour les valeurs *rgb* des couleurs et testez à nouveau grâce à l'éditeur de style.

Inspection de style

Chargez le fichier **css-premier-bis.html** dans le navigateur.

Nous allons de nouveau utiliser l'outil d'inspection du document déjà découvert pour explorer la structure HTML. Activez-le.

L'écran se sépare à nouveau en deux zones. Nous avons utilisé sa partie gauche, cette fois nous allons découvrir la partie droite de la zone de l'inspecteur. Dans un premier temps nous nous concentrerons sur les deux premiers outils appelés *Règle*, *Calculé*. Les autres onglets tel que *Polices*, permet simplement de connaître les différentes polices utilisées par le navigateur dans l'élément sélectionné. L'onglet *Mise en page* contient une rubrique *Modèle de boîte* qui sera abordée ultérieurement.



La zone CSS de l'inspecteur dans le navigateur. L'outil *Règles* est actif.

1. Commençons avec l'outil *Règles*.
Sélectionnez-le puis dans la zone de gauche de l'inspecteur cliquez sur différents éléments de la structure du document. Vous pouvez aussi utiliser le bouton de sélection d'élément pour pointer directement un élément dans la zone d'affichage. Examinez à chaque fois ce qui se passe dans l'inspecteur de styles.
Que représentent les informations affichées ?
2. Sélectionnez l'élément `<abbr>` situé dans le premier bloc `<p>`. Que remarquez-vous dans les règles affichées pour cet élément, en particulier dans la partie inférieure de l'affichage de ces règles ?
Idem pour l'élément `` emboîté dans ce premier élément `<p>` ?
Qu'indique les informations affichées ? Quel lien avec les *règles de calcul* établies lors d'un précédent exercice ?
3. Affichez dans l'outil les règles qui s'appliquent pour l'élément `<p>`. Ensuite, comme dans un exercice précédent modifiez la feuille de style **style1.css** pour lui ajouter la règle :

```
p {
    background-color : #F5F5DC;
}
```

Déplacez cette à différentes positions dans le fichier **style1-bis.css**, en particulier avant et après la règle présente initialement pour `p` et observer l'évolution de l'affichage des règles de l'élément `p` dans l'outil.

A nouveau faites le lien avec une des *règles de calcul* découverte précédemment.

4. Sélectionnez un des éléments ``. Dans la zone **Règles**, déplacez la souris sur les définitions des propriétés CSS. Des cases, actuellement cochées, apparaissent sur la gauche. Décochez celle en face de la propriété `font-weight`. Que constatez-vous dans la zone d'affichage ? Pourquoi ?

Faites de même pour la propriété `color` active. Que remarquez-vous ? Vous pouvez également désactiver cette propriété sur l'élément `<p>`. Justifiez le comportement observé.

Faites différents essais d'activation/désactivation de propriétés.

5. Nous allons voir qu'il est possible de tester en direct les impacts de modification de la feuille de style avec cet outil.

Après avoir à nouveau coché toutes les cases. Cliquez cette fois sur la valeur de la propriété `color` de l'élément ``. Une zone d'édition s'affiche, vous pouvez alors modifier la valeur, actuellement `yellow`, en la remplaçant par exemple par `lightgreen`. Observez le comportement dans la zone d'affichage. Faites plusieurs manipulations similaires et observez leurs effets.

Sélectionnez maintenant l'élément `<p>` et dans la zone des règles cliquez dans la zone blanche de sa définition (c'est-à-dire ni sur une case à cocher, ni sur une propriété, ni sur une valeur), une nouvelle ligne apparaît. Elle permet la saisie d'une nouvelle propriété et de sa valeur. Par exemple ajoutez : `font-style: oblique;` et observez le résultat.

Ajoutez une couleur d'arrière-plan aux éléments `<pre>`.

6. Poursuivons avec l'outil **Calculé**. Il permet pour un élément de connaître le *style calculé* qui s'applique à lui et d'avoir une explication sur ce calcul.

Sélectionnez-le puis dans la zone de gauche de l'inspecteur examinez son contenu lors de la sélection de différents éléments du document.

Par exemple sélectionnez l'élément ``.

Quelle est la valeur indiquée pour la propriété `color` ?

Tout en appuyant sur la touche `Shift`, cliquez sur le disque contenant la couleur à gauche de la valeur pour afficher différentes représentations de la couleur. Expliquez les informations fournies.

Explorer les styles calculés pour différents éléments.

Première feuille de style

1. Lisez la [page de référence](#) pour la propriété `text-align`.
2. Lisez la [page de référence](#) pour la propriété `font-family`. Consultez en particulier les [exemples](#).
3. Nous allons créer une première feuille de style pour les documents **premier.html** et **second.html** utilisés dans les exercices HTML. Commençons par préparer le travail en commençant par **premier.html** :
 1. Reprenez donc ce fichier et placez en une copie dans un nouveau dossier.
 2. Dans le dossier que vous venez de créer, créez un sous-dossier **style**. Ce dossier est destiné à accueillir les fichiers liés aux feuilles de style. C'est une bonne pratique de les placer dans un dossier à part.
 3. Créez maintenant dans ce dossier **style** un fichier vide de nom **premier-style.css** (par exemple). C'est ce fichier qui contiendra votre première feuille de style.
 4. Cette feuille de style va s'appliquer au document **premier.html**. Ouvrez donc ce document dans un éditeur de texte et modifiez son entête pour y déclarer à l'aide de la balise `<link>` l'utilisation de la feuille de style **premier-style.css**. Utilisez un chemin relatif pour faire référence à la feuille de style. Inspirez-vous de la syntaxe des fichiers fournis.
4. Vous êtes maintenant prêt à créer le contenu de votre première feuille de style. Ouvrez dans le navigateur le document **premier.html** et activez l'outil **Editeur de styles** (rappel du raccourci : `Shift+F7`).

Vous pouvez maintenant construire le contenu de ce fichier en y définissant les règles CSS qui doivent s'appliquer à votre document.

Faites-les en utilisant les différentes propriétés découvertes dans les différents exemples et exercices précédents (*color*, *background-color*, *font-size*, *font-style*, etc.). Inspirez-vous pour la syntaxe des exemples fournis dans les exercices précédents.

Vous utiliserez dans votre feuille chaque propriété au moins deux fois, avec des valeurs différentes.

Essayez de faire des choix esthétiques raisonnables...

Il vous manque certainement encore certaines propriétés, notamment sur les marges, pour avoir un résultat réellement satisfaisant. Nous les découvrirons très bientôt. Vous pourrez alors compléter votre feuille de style.

N'oubliez pas de sauvegarder votre feuille de style régulièrement pour conserver votre travail.

5. Vérifiez la validité CSS3 de la feuille de style que vous avez créée.
6. Vous allez maintenant appliquer votre feuille de style au document **second.html**. Placez une copie de ce fichier dans le dossier créé en début d'exercice et modifiez-le pour qu'il déclare lui aussi l'utilisation de la feuille de style **premier-style.css**. Visualisez ensuite le résultat dans votre navigateur.
7. Un document HTML peut utiliser plusieurs feuilles de style. Pour le vérifier, créez un fichier **second-style.css** que vous placerez dans le dossier **style**, puis déclarez en l'utilisation dans l'entête du fichier HTML **premier.html** à l'aide de la balise `link`.

Ouvrez ce document HTML dans le navigateur et utilisez l'éditeur de styles pour ajouter dans cette seconde feuille de style une règle CSS qui ne vient pas en conflit avec l'une de celles écrites dans la première feuille de style. Vérifiez alors que les règles des deux feuilles de style s'appliquent bien simultanément au document HTML.

Définir la dimension des éléments

La propriété CSS *width* permet de définir la largeur de la zone d'affichage du contenu d'un élément (`<p>`, ``, `<table>`, `<td>`, etc.). Sa valeur peut s'exprimer par une longueur fixe, le plus souvent en pixels, ou par un pourcentage. Il s'agit alors d'un pourcentage de la taille de l'élément conteneur (celui dans lequel l'élément est emboîté).

Dans cet exercice nous allons étudier le comportement de cette propriété.

1. Récupérez le fichier **sites-touristiques.html** et ouvrez le dans un navigateur. Examinez sa structure à l'aide de l'outil inspecteur. En particulier prenez connaissance des éléments emboîtés dans les différents éléments `<section>`. Notez également qu'aucun attribut *width* ni *height* n'est défini pour les éléments `` dans le code HTML et que dans la feuille de style utilisée et fournie il n'y aucune règle qui porte sur la propriété *width*.

Dans la feuille de style vous pouvez constater que des couleurs d'arrière-plan ont été définies pour les différents éléments. Ces couleurs vont nous permettre de bien visualiser ces éléments. En effet la couleur d'arrière-plan s'applique sur toute la largeur et toute la hauteur du contenu d'un élément. On peut donc visuellement constater les dimensions d'un élément en observant les zones couvertes par sa couleur d'arrière-plan.

Pour information le fichier **chenonceau.jpg** contient une image de 1024 pixels de large et les deux autres, **colisee.jpg** et **pont-du-gard.jpg**, contiennent des images larges de 800 pixels.

2. Faites varier la largeur de la fenêtre de votre navigateur et observez les variations de la valeur de la largeur des éléments affichés (`section`, `header`, `p` et `img`). Que constatez-vous ?
 1. Quelle est la valeur de cette largeur pour les éléments `section`, `header`, `p` ?
 2. A l'aide de l'outil inspecteur, zone **Mise en page**, déterminez la valeur d'affichage de chacune des images. Déduisez-en la valeur de la largeur d'affichage des images

quand aucune règle CSS n'est présente (donc la valeur par défaut de cette largeur pour les éléments `img`).

3. Utilisez l'outil Editeur de style pour ajouter à la feuille CSS une règle qui fixe à `600px` la valeur de la propriété `width` des éléments `section`. Que constatez-vous au niveau de l'affichage ?

Essayez d'autres valeurs, `400px` puis `700px` par exemple. Quelle est dans chaque cas la valeur de la largeur d'affichage pour les éléments `header` et `p` (on peut l'observer grâce aux couleurs d'arrière-plan de ces éléments) ? Utilisez l'outil inspecteur pour le vérifier. Faites varier la largeur de la fenêtre du navigateur et constatez que la largeur d'affichage des éléments `section`, `header`, `p` ne change pas.

Que constatez-vous au niveau des éléments images ? Quelle est leur largeur dans les différents cas ? Il s'agit d'un cas particulier des éléments `img` dont la largeur d'affichage par défaut est la largeur intrinsèque de l'image qui s'impose quand rien n'est indiqué par la feuille de style.

Ajoutez une règle qui fixe à `600px` la valeur de la propriété `width` pour les images.

4. On peut fixer une valeur pour la largeur de tous les éléments de flux. Cette largeur s'impose aux éléments. Ajoutez une règle qui fixe la largeur des éléments `p` (uniquement eux) à `500px` et observez le résultat. Recommencez ensuite avec une valeur de `800px`.

Qu'observe-t-on ?

Définissez une règle qui fixe la largeur des éléments `strong` à `300px` et constatez qu'elle est sans effet. Il n'est en effet (par défaut) pas possible d'agir sur la largeur des éléments de phrases.

Il s'agit d'un comportement par défaut, on peut modifier ce comportement en adaptant la valeur de la propriété `display`. Nous verrons cela ultérieurement.

Notons que si les `img` sont des éléments de phrase, ce sont également des éléments de flux ce qui explique qu'il est possible d'agir sur leur largeur. Il en est de même des éléments `a` lorsque leur contenu est un élément de flux.

5. Supprimez dans la feuille CSS la règle qui fixe la largeur des éléments `p` et modifiez celle pour les éléments `section` en attribuant à la propriété `width` la valeur `80%`.

Dans l'outil inspecteur, sélectionnez l'un des éléments `section` de la page et dans la zone **Calculé** consultez la largeur de cet élément.

Maintenant faites varier la largeur de la fenêtre du navigateur tout en surveillant dans l'outil inspecteur la valeur de la largeur affichée. Que peut-on en déduire sur la signification de la valeur `80%` ?

6. Dans la feuille CSS, fixez maintenant la largeur de l'élément `body` à `1000px`. Définissez ensuite une règle qui fixe la largeur des éléments `section` à `80%` puis à `60%`. Dans chaque cas utilisez l'outil inspecteur pour connaître la valeur exacte de la largeur de l'un des éléments `section`. Pourquoi cette valeur ?

Repositionnez dans la feuille CSS la largeur des éléments `section` à `80%` puis définissez une nouvelle règle qui fixe la largeur des éléments `p` à `80%`. Quelle est la valeur indiquée par l'inspecteur pour la largeur de l'un des éléments `p` ? Pourquoi cette valeur ?

Que peut-on en déduire sur la signification de la définition d'une valeur en pourcentage de la largeur d'un élément ? A quelle quantité s'applique ce pourcentage ?

7. Dans le code source du fichier `sites-touristiques.html` vous pouvez constater que des éléments `figure` qui encadrent les éléments `img` sont actuellement placés en commentaire. Supprimez ces marques de commentaire dans le fichier HTML puis ajoutez dans le fichier CSS une règle qui fixe à `550px` la largeur des éléments `figure`.

Déduisez de la question précédente une règle CSS qui définit la largeur des éléments `img` de telle sorte qu'ils occupent la totalité de la largeur des éléments `figure` sans en déborder.

Définir la dimension des éléments (suite)

De manière analogue à la propriété CSS `width` la propriété CSS `height` permet de définir la hauteur de la zone d'affichage du contenu d'un élément. Elle peut à nouveau s'exprimer en pixels ou en pourcentage. On prolonge le travail de l'exercice précédent avec le fichier **sites-touristiques.html**. Ouvrez le dans un navigateur.

1. Testez la propriété `height` en ajoutant une règle qui fixe la hauteur des éléments `p` à `300px` puis à `30px`. Dans chacun des cas constatez ce qu'il se passe. En particulier comparez le comportement du contenu des éléments `p` (les textes) avec leurs dimensions perceptibles grâce à la couleur d'arrière-plan.

On le constate agir sur la propriété `height` peut être plus délicat que pour la propriété `width` en ce qui concerne l'adaptation au contenu.

2. On va s'intéresser à nouveau aux images.

Supprimez dans la feuille de style toutes les définitions de valeur pour la largeur ou la hauteur des images.

Fixez maintenant la valeur de la propriété `width` à `400px` et visualisez. Tout est normal ? (A priori oui...)

Supprimez la définition précédente et fixez cette fois la valeur de la propriété `height` à `500px` et visualisez. Tout est normal ? (A priori, encore oui...)

Conservez cette définition et remplacez la valeur de la propriété `width` à `400px`. Visualisez le résultat au niveau des images. Qu'en pensez-vous ?

Quelle conclusion en tirer sur la définition simultanée des propriétés `width` et `height` pour les images ?

Dans la mesure où le plus souvent on souhaite conserver le rapport d'affichage des images. Lorsque l'on définit des valeurs pour les deux propriétés `width` et `height`, on donne une valeur numérique à l'une (le plus souvent `width`) et à l'autre la valeur `auto` pour indiquer que l'on souhaite conserver ce ratio. `auto` est la valeur par défaut de ces propriétés.

3. Supprimez toute référence à `height` et fixez la valeur de la propriété `width` des éléments `` à `50%`.

Réduisez la largeur de la fenêtre du navigateur, y compris jusqu'à avoir une très petite largeur. Que constatez-vous sur les tailles d'affichage des images ?

Ajoutez maintenant à votre déclaration les définitions suivantes :

```
min-width : 200px;
max-width : 450px;
```

Puis, à nouveau, réduisez et agrandissez la fenêtre du navigateur. Que constatez-vous concernant la taille d'affichage des images ?

Concluez-en le rôle de ces deux propriétés et vérifiez/complétez votre réponse à l'aide du MDN.

4. Complétez votre feuille de style avec des règles portant sur l'ensemble des éléments du document : titre, paragraphe, hyperlien.

Les propriétés `width` et `height` (et leurs déclinaisons `min-` et `max-`) peuvent s'appliquer à la plupart des éléments HTML de flux avec le même fonctionnement que ce qui a été étudié ici.

Boîtes, marges et bordures

Ouvrez le fichier **boites.html** dans le navigateur. L'affichage de ce fichier se traduit par un damier de 3 lignes de 4 tuiles de couleur.

1. Etudiez la structure du code HTML (soit dans un éditeur de texte soit grâce à l'outil inspecteur) et constatez que les lignes sont représentées par des éléments `div` qui contiennent quatre éléments `section`. Le contenu de ces éléments `section` est un élément `article`.

L'utilisation de la règle `display : inline-block` a pour conséquence que les éléments de flux `section` s'affichent les uns après les autres en ligne mais ce n'est pas l'important dans cet exercice.

2. L'étude de la feuille CSS vous permet de constater :
- les règles CSS des lignes 36 à 46 du fichier **style-boite.css** ont pour rôle la définition des couleurs d'arrière-plan des éléments `article`. Elles produisent la présentation en damier en gérant l'alternance des éléments. La suite du cours permettra de comprendre la signification des sélecteurs mais ce qui compte ici c'est de constater que les couleurs `#FFC107` et `#8BC34A` sont les couleurs d'arrière-plan des éléments `article`. La couleur d'arrière-plan des éléments `section` est `#0288D1` (ligne 20).
 - les valeurs des largeur et hauteur des éléments `section` et de celles de leur contenu, les éléments `article`. Ces valeurs fixées à `100%` justifient que les couleurs visibles correspondent à celles des éléments `article` qui occupent la totalité du contenu des éléments `section`

Ces constatations faites nous pouvons commencer à étudier les propriétés qui concernent la boîte CSS des éléments.

3. Commençons par revenir rapidement sur le point abordé dans l'exercice précédent : les propriétés `width` et `height` définissent les dimensions du contenu de l'élément.
- Pour le vérifier il suffit de modifier les valeurs de ces propriétés pour les éléments `section` et de constater que les dimensions des éléments `article` qu'ils contiennent s'adaptent. Faites quelques essais avant de revenir à la valeur initiale (`150px`).
4. La propriété `border` permet sans surprise de définir une bordure (ou cadre) qui est positionnée autour du contenu des éléments. Une bordure est caractérisée par trois valeurs chacune associée à une propriété :
- son épaisseur, `border-width`,
 - sa couleur, `border-color`,
 - son style de trait, `border-style`.

L'étude de la [page de référence du MDN](#) pour la propriété `border` vous permettra d'en connaître tous les détails et vous pouvez en profiter pour découvrir également les détails des propriétés `border-width`, `border-style` et `border-color`

- Il est possible de définir les trois valeurs en une seule fois. Ajoutez pour les éléments `section` la règle :

```
border : 2px white solid;
```

La signification des différentes valeurs est assez évidente. Notez que l'ordre est sans importance. Vous pouvez le vérifier en modifiant l'ordre des valeurs sans changement dans l'affichage, par exemple

```
border : solid white 2px;
```

- Il est également possible d'agir sur chacune des propriétés individuellement. Ajoutez, après la précédente, la règle :

```
border-width : 5px ;
```

pour constater qu'elle ne modifie que l'épaisseur de la bordure. Les autres propriétés opèrent de façon similaire.

La valeur de l'épaisseur de la bordure peut aussi être fixée par quelques constantes prédéfinies. Par exemple `thin` (mince en anglais) correspond à une épaisseur de `1px`.

5. La propriété `margin` permet d'ajouter un espacement autour d'un élément. Cet espace est défini à l'**extérieur** de la zone délimitée par la bordure, on parle donc de marge extérieure. Pour la tester ajoutez aux élément `section` la règle

```
margin : 10px;
```

Vous constatez que les éléments `section` "s'écartent" les uns des autres. La propriété `margin` permet donc de créer un espace entre un élément et ses voisins.

Comme vous pouvez le vérifier avec l'outil inspecteur, l'élément `body` a une valeur par défaut fixée à `8px` pour la propriété `margin`. Ceci explique l'espace entre le contenu du document et le bord de la fenêtre du navigateur.

6. La propriété `padding` permet d'ajouter un espacement autour du contenu de l'élément. Cet espace est défini à l'intérieur de la zone délimitée par la bordure, on parle donc de marge intérieure. Pour la tester ajoutez aux éléments `section` la règle

```
padding : 15px;
```

Vous constatez que le contenu des éléments `section` s'éloigne des bordures, sans que les dimensions de ce contenu, définies par `width` et `height`, ne changent.

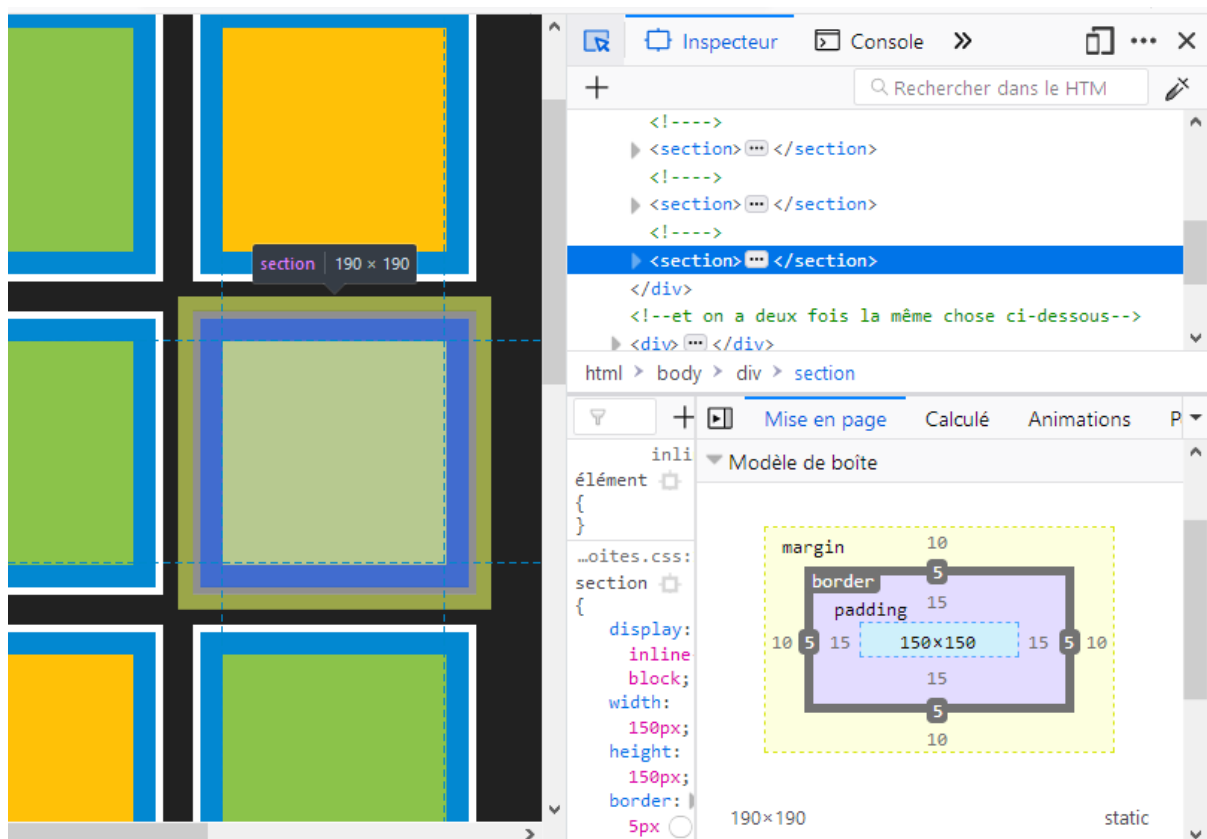
7. Quelle est la couleur d'arrière-plan de la zone due à la propriété `margin` liée à la propriété `padding` qui est apparue entre la bordure et le contenu ? D'où provient cette valeur ?

De même quelle est la couleur d'arrière-plan de la zone due à la propriété `margin` ?



Vérifiez votre réponse

8. Dans l'outil inspecteur, activez l'onglet **Calculé** puis sélectionnez l'un des éléments `section`. Vous devriez obtenir des informations similaires à ceci :



Les valeurs des propriétés de la boîte CSS d'un élément `section` dans l'outil inspecteur.

L'outil permet de retrouver l'ensemble des valeurs des propriétés qui régissent la boîte CSS d'un élément.

On peut constater que, dans la fenêtre du navigateur, la zone mise en évidence lors de la sélection de l'élément `section` comprend bien la partie `margin` même si celle-ci est "invisible".

A quoi correspondent les différentes valeurs qui apparaissent dans l'outil **Calculé** ? En particulier la valeur `190x190` ?

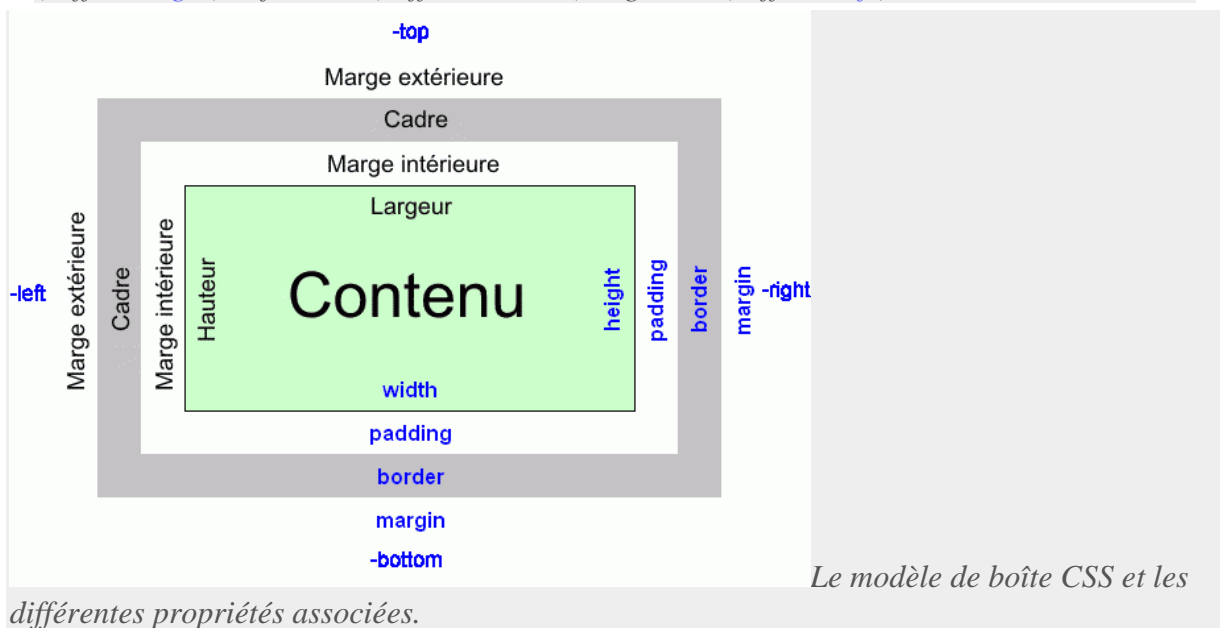
Les éléments occupent dans la page un espace de forme rectangulaire. La taille de l'espace occupé par un élément dans la page est définie à la fois par la taille de son

contenu et par l'espacement autour de ce contenu. Les dimensions d'affichage du **contenu** d'un élément sont définies par les propriétés **width** et **height**. L'espace autour du contenu est géré grâce à trois paramètres :

- la **marge intérieure** (**padding**) gère les dimensions de la zone entre le contenu de l'élément et sa bordure
- la **bordure** (**border**) désigne un cadre situé autour du contenu de l'élément
- la **marge extérieure** (**margin**) gère les dimensions de l'espacement entre la bordure et le reste de la page (les boîtes des autres éléments affichés autour), donc l'espacement à l'extérieur de la bordure

Notons que ces propriétés ne s'héritent pas. C'est-à-dire que si l'on définit une marge (par exemple) pour un élément, cette marge ne s'applique pas pour ses éléments enfants (ceux qui sont emboîtés dedans).

Pour chacune de ces propriétés (voir exercice suivant) il est possible de définir séparément des valeurs de propriétés pour les parties supérieure (suffixe **-top**), droite (suffixe **-right**), inférieure (suffixe **-bottom**) et gauche (suffixe **-left**) de la boîte.



Boîtes, marges et bordures (suite)

Nous continuons avec le fichier **boîtes.html** de l'exercice précédent. Ouvrez-le dans le navigateur.

1. Si vous n'avez pas conservé les modifications de l'exercice précédent, ajoutez dans la feuille de style les règles pour :
 - définir une bordure blanche, en trait continu et épaisse de **5px**
 - définir une marge extérieure de **10px**
 - définir une marge intérieure de **20px**
2. Dans l'exercice précédent, à chaque fois, nous avons modifié les propriétés **margin**, **padding** et **border** en agissant simultanément sur leurs valeurs au-dessus, à droite, en-dessous ou à gauche du contenu.

En ajoutant un suffixe au nom de ces propriétés il est cependant possible de définir séparément des valeurs pour les parties supérieure (suffixe **-top**), droite (suffixe **-right**), inférieure (suffixe **-bottom**) et gauche (suffixe **-left**) de la boîte.

Testez-le ajoutant pour les éléments **h1** la règle :

```
padding-left : 20px;
```

Profitez de l'occasion pour préciser que les dimensions que nous avons jusque maintenant toujours exprimé de manière fixe en pixels, peuvent s'exprimer également en pourcentage.

Essayez :

```
padding-left : 3%;
```

Comme vous pouvez le vérifier avec l'outil inspecteur, les éléments `h1` ont une valeur par défaut pour les marges supérieure et inférieure fixée à `20px`. Ceci explique l'espace au-dessus et en-dessous de cet élément dans la page.

3. Ajoutez pour les éléments `h1` la règle CSS :
`padding : 4px;`
en essayant de la placer après puis avant la règle définie dans la question précédente, vérifiez que dans un cas comme dans l'autre pour la marge gauche c'est bien la dernière règle écrite qui l'emporte.
4. Ajoutez une règle pour donner aux bordures inférieures des éléments `section` la couleur `#F44336`. Les trois autres côtés de la bordure doivent rester en blanc.
5. Définissez des règles pour ajouter au-dessus et au-dessous des éléments `h1` une bordure en trait continu de couleur `#FFEB3B` et épaisse de `5px`. Il ne doit pas y avoir de bordure à gauche et à droite.
6. Pour les marges, intérieures ou extérieures, il est possible de définir les valeurs des tailles des différents côtés directement avec `padding` ou `margin`. Il suffit de fournir plusieurs valeurs à ces propriétés. Les côtés affectés dépendent du nombre de valeurs fournies. Par exemple en fournissant deux valeurs on agit sur les tailles verticales et horizontales, essayez pour les éléments `section` :
`padding : 30px 10px`
En fournissant quatre valeurs on agit (dans cet ordre) sur les tailles au-dessus, à droite, en-dessous et à gauche, essayez pour les éléments `section` :
`padding : 20px 10px 5px 0px;`
7. Pour les éléments `h1` définissez une marge intérieure de `5px` au-dessus et en-dessous et de `3%` à gauche et à droite.
8. Jusque maintenant nous n'avons agi que sur des éléments de flux mais les propriétés `margin`, `padding` et `border` s'appliquent également aux éléments de phrasé. Pour le vérifier ajoutez pour les éléments `span` (il y en a dans le titre) des règles pour
 - définir une bordure au-dessus et en-dessous blanche, d'un pixel et en style `dashed`,
 - définir un `padding` horizontal de `5px` et vertical de `1px`.
9. Nous allons voir comment centrer horizontalement un élément (pas un contenu qui peut se centrer grâce à la propriété `text-align : center`).
Commencez par fixer la largeur des éléments `h1` à `800px`. On pourrait chercher les valeurs de `margin-left` et `margin-right` à définir pour centrer l'élément mais la difficulté est que l'on ne connaît pas la largeur d'affichage du document HTML. Pour contourner ce problème il faut utiliser la valeur `auto`. Essayez-la en définissant une règle CSS qui attribue aux éléments `h1` une marge (extérieure) supérieure et inférieure de `10px` et la valeur `auto` aux marges gauche et droite.
Constatez le résultat puis vérifiez qu'il reste valable lorsque vous changez la valeur de la propriété `width` des éléments `h1`.
10. Finissons par une étude du comportement des marges entre éléments contigus. Commencez par supprimer les valeurs de la propriété `margin` pour les éléments `section`.
 - Ajoutez une règle pour définir la marge gauche des éléments `section` à `10px`, puis, après avoir observé le résultat, ajoutez une autre règle pour définir la marge droite des éléments `section` à `10px`.

Vous devez constater que les espaces créés par les marges gauche et droite se sont additionnés. A l'aide de l'outil inspecteur vérifiez que les boîtes (avec leur marge) des éléments `section` voisins se juxtaposent.
 - Ajoutez maintenant une règle pour définir la marge supérieure des éléments `div` à `10px`. Vous constatez que logiquement un espacement a été créé entre les éléments `div` qui représentent les lignes.

Ajoutez maintenant une autre règle pour définir la marge *inférieure* de ces mêmes éléments `div` à **10px**. Que constatez-vous au niveau de l'espace vertical entre ces éléments ? A-t-il changé ?

La réponse est non : contrairement aux marges gauche et droite, les marges supérieure et inférieure ne se juxtaposent pas mais elles **fusionnent**. Vous pouvez le vérifier avec l'inspecteur où vous pouvez constater le chevauchement des boîtes de deux éléments `div` successifs.

Modifiez la règle qui définit la marge inférieure pour modifiez sa valeur à **20px**. Que constatez-vous ? Quelle est la valeur de l'espace vertical entre deux éléments `div` ?

Passez la valeur de la marge supérieure à **35px** et à nouveau déterminez la valeur de l'espace vertical entre les éléments `div`. Pourquoi cette valeur ?

Déduisez-en la règle qui définit l'espace vertical entre deux éléments en fonction des valeurs de leurs marges inférieure et supérieure.

Lorem Ipsum

Ouvrez le fichier `lorem-ispum.html` dans le navigateur. Comme vous le constatez les textes sont un peu à l'étroit et il est difficile de visuellement distinguer les différents éléments de la structure du document, nous allons y remédier dans cet exercice en complétant la feuille de style **`style/style-lorem-ispum.css`**. L'image ci-dessous vous donne un aperçu du résultat à obtenir à la fin de cet exercice.

Les éléments HTML utilisés dans le code de ce fichier ont été choisis uniquement afin de permettre de travailler sur les propriétés qui gèrent les boîtes CSS, leur utilisation n'est pas nécessairement totalement adaptée ici à leur sémantique.

Lorem ipsum

Origine

Qu'est-ce que le Lorem Ipsum ?

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker.

Pourquoi l'utiliser ?

On sait depuis longtemps que travailler avec du texte lisible et contenant du sens est source de distractions, et empêche de se concentrer sur la mise en page elle-même. L'avantage du Lorem Ipsum sur un texte générique comme 'Du texte. Du texte. Du texte.' est qu'il possède une distribution de lettres plus ou moins normale, et en tout cas comparable avec celle du français standard. De nombreuses suites logicielles de mise en page ou éditeurs de sites Web ont fait du Lorem Ipsum leur faux texte par défaut, et une recherche pour 'Lorem Ipsum' vous conduira vers de nombreux sites qui n'en sont encore qu'à leur phase de construction. Plusieurs versions sont apparues avec le temps, parfois par accident, souvent intentionnellement (histoire d'y rajouter de petits clins d'oeil, voire des phrases embarrassantes).

D'où vient-il ?

Contrairement à une opinion répandue, le Lorem Ipsum n'est pas simplement du texte aléatoire. Il trouve ses racines dans une oeuvre de la littérature latine classique datant de 45 av. J.-C., le *rendant vieux* de 2000 ans. Un professeur du Hampden-Sydney College, en Virginie, s'est intéressé à un des mots latins les plus obscurs, *consectetur*, extrait d'un passage du Lorem Ipsum, et en étudiant tous les usages de ce mot dans la littérature classique, découvrit la source incontestable du Lorem Ipsum. Il provient en fait des sections 1.10.32 et 1.10.33 du "De Finibus Bonorum et Malorum" (Des Suprêmes Biens et des Suprêmes Maux) de Cicéron. Cet ouvrage, très populaire pendant la Renaissance, est un traité sur la théorie de l'éthique. Les premières lignes du Lorem Ipsum, "Lorem ipsum dolor sit amet...", proviennent de la section 1.10.32.

Exemple

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sed enim ut sem viverra aliquet eget. Lorem mollis aliquam ut porttitor leo. Orci nulla pellentesque dignissim enim sit. Netus et malesuada fames ac turpis egestas maecenas pharetra. Pretium viverra suspendisse potenti nullam. Nibh sed pulvinar proin gravida hendrerit. Ut morbi tincidunt augue interdum velit. Ac turpis egestas integer eget aliquet. Quam nulla porttitor massa id neque aliquam. Rutrum tellus pellentesque eu tincidunt tortor aliquam. Malesuada fames ac turpis egestas sed tempus urna et pharetra. Quisque sagittis purus sit amet volutpat consequat.

Elementum pulvinar etiam non quam lacus. Turpis nunc eget lorem dolor sed. Elementum eu facilisis sed odio morbi quis commodo odio aenean. Euismod elementum nisi quis eleifend quam. Arcu cursus euismod quis viverra nibh cras pulvinar mattis nunc. Libero enim sed faucibus turpis in eu mi bibendum neque. At volutpat diam ut venenatis. Congue mauris rhoncus aenean vel elit scelerisque. Et malesuada fames ac turpis. Nibh ipsum consequat nisl vel pretium lectus quam id leo. Non blandit massa enim nec dui nunc. Condimentum vitae sapien pellentesque habitant morbi tristique. Ullamcorper eget nulla facilisi etiam. Mattis rhoncus urna neque viverra justo nec ultrices dui. Varius sit amet mattis vulputate enim nulla aliquet porttitor. Aliquam ultrices sagittis orci a scelerisque purus. Adipiscing diam donec adipiscing tristique risus nec feugiat in. Vitae proin sagittis nisl rhoncus mattis rhoncus urna.

Enim ut sem viverra aliquet. Ante metus dictum at tempor commodo ullamcorper a lacus. Fermentum iaculis eu non diam phasellus vestibulum lorem sed. Lacus laoreet non curabitur gravida arcu. Consequat id porta nibh venenatis cras. Sit amet massa vitae tortor condimentum lacinia quis vel. Donec massa sapien faucibus et molestie. Risus nec feugiat in fermentum posuere urna. Est pellentesque elit ullamcorper dignissim. Facilisi cras fermentum odio eu feugiat. Scelerisque viverra mauris in aliquam. Diam quam nulla porttitor massa id neque aliquam vestibulum morbi. Mattis nunc sed blandit libero volutpat. Scelerisque in dictum non consectetur a erat nam. Ut pharetra sit amet aliquam id diam maecenas ultricies mi. Nisl nunc mi ipsum faucibus vitae aliquet nec. Vestibulum lectus mauris ultrices eros in cursus.

Mattis nunc sed blandit libero volutpat sed cras. Integer vitae justo eget magna fermentum iaculis eu. Aliquet lectus proin nibh nisl. Volutpat commodo sed egestas egestas fringilla. Eget felis eget nunc lobortis mattis aliquam faucibus. Phasellus faucibus scelerisque eleifend donec pretium vulputate sapien nec. Sapien eget mi proin sed libero enim sed. Aenean euismod elementum nisi quis eleifend quam adipiscing. Suscipit adipiscing bibendum est ultricies integer quis auctor elit sed. Sit amet consectetur adipiscing elit duis tristique sollicitudin nibh. Ut porttitor leo a diam sollicitudin. Ornare quam viverra orci sagittis eu volutpat odio. Et odio pellentesque diam volutpat commodo sed egestas egestas fringilla. Iaculis nunc sed augue lacus viverra vitae congue eu.

Consectetur nunc ut faucibus pulvinar elementum integer. Aliquam malesuada bibendum arcu vitae. Dolor magna aenean est lorem ipsum. Vulputate enim nulla aliquet porttitor lacus. Aenean

Aperçu du résultat à obtenir

1. Ajouter des couleurs, d'arrière-plan notamment, est un premier moyen de faire apparaître la structure d'un document et ses différents éléments. Faites-le en enlevant du fichier CSS les marques de commentaires (lignes 1 et 27). Il reste néanmoins nécessaire "d'aérer" un peu le document.
2. Définissez les règles CSS nécessaires pour que l'élément de titre `h1` occupe les trois-quarts de la largeur de la page et soit centré. Faites également le nécessaire pour que son texte soit centré dans l'élément.
3. Faites-en sorte qu'il n'y ait pas d'espace vertical entre un élément `h2` et l'élément `section` qui le suit.
4. Ajoutez une bordure fine de couleur `#BDBDBD` en trait continu aux éléments `h2` et `section`.
5. Sans modifier les autres bordures, faites le nécessaire pour que la bordure gauche des éléments `h2` fasse `10px` d'épaisseur. Complétez en éloignant le texte contenu dans cet élément de `20px` vers la droite.
6. Ajoutez sous les éléments `header` une bordure de `3px` de couleur `#BDBDBD` en trait continu.
7. Ajoutez pour les sections une règle qui crée un espace autour du contenu de ces éléments de `5px` au-dessus et en-dessous et de `10px` à gauche et à droite. Vous devez réaliser cela sans écarter ces éléments `section` de leurs éléments voisins (on veut par exemple qu'il n'y ait toujours pas d'espace vertical avec les éléments `h2` qui les précède quand il y en a un). Essayez de l'écrire en une seule règle.
8. Les éléments `article` sont trop proches des autres éléments, éloignez verticalement (uniquement) de `10px` des éléments voisins.
9. Dans les éléments `article` le texte est trop proche du bord de la zone définie par la couleur d'arrière-plan (ici blanc). Définissez pour les éléments `div` en une seule règle une marge intérieure verticale de `5px` et horizontale de `20px`.
10. Terminez en "aérant" aussi un peu le contenu des éléments `header` qui est encore un peu à l'étroit.

Première feuille de style (reprise)

1. Reprenez la feuille de style que vous avez créé dans le cadre de l'exercice intitulé *Première feuille de style* et complétez votre fichier **premier-style.css** avec des règles qui agissent sur les dimensions, marges et/ou bordures des différents éléments des documents.
2. Appliquez cette feuille de style (ou une copie de celle-ci) au document sur Tim Berners Lee que vous avez créé dans un exemplaire précédent et complétez ou modifiez la pour obtenir un résultat visuel qui vous convienne pour ce document.

Des tableaux plus stylés

Dans cet exercice nous allons réutiliser les fichiers manipulés lors des exercices sur les tables HTML : celui sur les données d'insertion des masters du FIL et **evol-pop.html**.

1. Explorez la [page de référence](#) pour la propriété `text-align`.
Cette propriété peut, entre autres, être appliquée aux éléments d'une table HTML.
2. Créez deux feuilles de style qui portent sur tout ou partie des différents éléments présents dans une table HTML (`<table>`, `<th>`, `<tbody>`, `<caption>`, etc.).

Évidemment l'idée est de mettre en valeur les entêtes et pied de tables par rapport au contenu.

Vous veillerez entre les deux feuilles à varier les éléments sur lesquels vous définissez une valeur de propriété. Par exemple dans l'une des feuilles définissez une valeur pour `color` sur l'élément `<th>` et pas sur l'élément `<thead>`. Faites le contraire dans la seconde feuille, puis constatez les différences.

A nouveau, un peu de patience, nous verrons assez vite comment faire mieux.

3. Appliquez chacune de ces feuilles de style aux deux documents sur les tables et visualisez les résultats.
4. Compte-tenu de ce qui a été vu, que faudrait-il modifier dans le document **evol-pop.html** pour que les cellules contenant les années soient considérées comme des entêtes de ligne et puissent recevoir un style différent des autres
Si cela n'avait pas été fait dans le document initial, faites-le, au moins sur quelques-unes des premières lignes, puis étudiez l'impact de cette modification lors de l'application de la feuille de style.
5. Comme toujours, vérifiez la validité CSS3 de vos feuilles.

Les tableaux ont plusieurs couches

Ouvrez le fichier **evol-pop-couches.html**.

Consultez sa feuille de style pour constater que pour le moment toutes les couleurs d'arrière-plan définies pour les éléments `<table>`, `<thead>`, `<th>` et `<td>` le sont avec une composante *alpha* à 0 (c'est-à-dire la quatrième composante de la valeur *rgba*). Elles sont donc transparentes.

1. A l'aide de l'éditeur de style ou de l'outil inspecteur, modifiez une par une ces couleurs en fixant leur composante alpha à 1 (= opaque), et en laissant les valeurs alpha des autres couleurs à 0. Vous pouvez ainsi constater visuellement sur quels éléments de la table les propriétés agissent.
2. En repartant de la feuille initiale (avec toutes les composantes alpha à 0), modifiez à nouveau une par une les composantes alpha à 1 en procédant élément par élément, d'abord `<table>`, puis `<thead>`, puis `<th>` et `<td>`, mais cette fois sans remettre à 0 les composantes alpha des autres éléments. Observez entre chaque modification l'évolution visuelle de la table.
Que constatez-vous ?
3. Recommencez ce travail en fixant cette fois une par une les composantes alpha à 0.5. Observez et interprétez le résultat.
4. Déduisez des questions précédentes quelle est la superposition des différentes couches qui gèrent le modèle d'affichage des tables.

Tableaux stylés (suite)

1. Reprenez la feuille de style créée pour les tableaux et complétez-la avec des propriétés de taille, marges, bordures...

Vous pouvez par exemple fixer la largeur des cellules, créer une bordure autour des `<thead>`, agrandir la marge intérieure des `<th>`, ajouter une marge autour des `<table>`, définir des propriétés sur `<caption>`, etc.

Il peut être intéressant d'étudier les propriétés *border-collapse* et *border-spacing* qui s'appliquent à l'élément `<table>`.

Il est certainement plus pertinent de définir les largeurs des cellules en % plutôt qu'en pixels.

*Supprimez de **evol-pop.html** les informations de style définies initialement dans l'entête.*