Henallux – DA1 – web : principes de base

Laboratoire 4

Arborescence HTML et CSS+

Sommaire

[1. (Exo) Arborescence HTML 2](#_Toc159853134)

[2. (Exo) Bientôt le carnaval ! 8](#_Toc159853135)

[3. (Exo) des tables plus stylées 11](#_Toc159853136)

[4. Éléments de solution de l’exercice 1 15](#_Toc159853137)

# (Exo) Arborescence HTML

Objectifs

* visualiser l’arborescence d’un document HTML
* comprendre des sélecteurs complexes
* utiliser l’inspecteur de Firefox

Cet exercice utilise le document HTML intitulé arborescence.html.

## Étape 1 : l’arborescence du document

Prenez tout d’abord quelques minutes pour lire le document et comprendre sa structure. Pour rendre celle-ci plus apparente, dessinez, sur une feuille de papier, l’arborescence HTML.

Pour plus de simplicité, vous pouvez commencer en prenant la balise <body> comme racine (au lieu de <html>). Pour chaque élément HTML (chaque nœud de l’arbre), indiquez sa balise et, éventuellement, sa classe et/ou son identificateur.

Vous pourrez vérifier votre travail en le comparant avec l’arbre qui se trouve à la fin de ce document.

## Étape 2 : vocabulaire des arborescences

Sur base de l’arbre que vous avez dessiné, répondez aux 7 questions suivantes. Comparez ensuite vos réponses avec celles de vos voisins et discutez en cas de désaccord !

1. Combien de fils possède le premier <div> ?
2. Combien de descendants possède le paragraphe désigné par le sélecteur p#savoir ?
3. Combien d’éléments HTML seront ciblés par le sélecteur p.important ?
4. Indiquez le père de chacun des éléments HTML ciblés par le sélecteur p.important.
5. Indiquez le nombre de frères/sœurs de chacun des éléments HTML ciblés par le sélecteur p.important.
6. Quel est l’élément HTML qui possède le plus de fils ?
7. Quel est l’élément HTML qui possède le plus de descendants ?

## Étape 3 : inspecteur Firefox

Ouvrez le document HTML sous Firefox tout en gardant l’arbre que vous avez dessiné devant vous.

**Ouvrir l’inspecteur**

Effectuez un clic droit sur le titre « Note très importante » puis choisissez l’option « Inspect / Examiner l’élément ».

Cela ouvre l’inspecteur de Firefox. Il s’agit d’une interface qui permet de visualiser le code caché derrière une page web, tant au niveau HTML qu’au niveau CSS. Pour ouvrir l’inspecteur, vous pouvez choisir d’examiner un élément (comme vous venez de le faire) ou encore utiliser le raccourci Ctrl-Shift-I ou encore passer par le menu « Outils » (appuyez sur Alt pour faire apparaître la barre de menu), l’option « Outils du navigateur » et l’option « Outils de développement web ».

L’inspecteur est composé de plusieurs zones contenant de nombreuses informations ; on n’en abordera ici que quelques-unes. Vous pourrez plus d’information au sujet de l’inspecteur sur le site de Firefox (par exemple à l’adresse <https://firefox-source-docs.mozilla.org/devtools-user/page_inspector/>).

**Cibler un élément à inspecter**

Au-dessus de l’inspecteur se trouve une barre présentant un bouton montrant une flèche dans un cadre suivi de plusieurs onglets. Le bouton permet de sélectionner un élément dans la page HTML. Cliquez sur celui-ci puis promenez votre curseur sur la page web.

A black and white screen with white text

Description automatically generated

Notez que diverses informations s’affichent ; entre autres, les éléments survolés sont encadrés et Firefox affiche non seulement la taille de l’élément (en pixels × pixels) mais également sa balise et éventuellement, sa classe et son identificateur.

Pour revenir au titre « Note très importante », cliquez à nouveau sur le bouton (s’il n’est plus actif), puis sur le titre « Note très importante » sur la page HTML.

Pour l’instant, nous ne nous occuperons que de l’Inspecteur… les autres onglets (Console, Débogueur, Éditeur de style, Performances et Réseau) sont laissés de côté.

**L’arborescence HTML dans l’inspecteur**

En-dessous de l’inspecteur, Firefox indique le chemin à parcourir en partant de la racine <html> pour arriver à l’élément examiné.



Dans ce cas-ci :

* l’ancêtre-racine du titre est <html> ;
* il faut ensuite passer par <body> ;
* puis rentrer dans un <div> de classe cadre ;
* pour enfin arriver à la balise <h3>.

Le même chemin (ainsi que toute l’arborescence du document HTML) peut être observé dans la fenêtre principale de l’inspecteur.

A screen shot of a computer program

Description automatically generated

Dans celle-ci, vous verrez apparaître chacune des balises citées ci-dessus : <html>, <body>, le <div> puis <h3>.

Chacune de ces balises peut être « ouverte » (précédée d’une flèche pointant vers le bas) ou « fermée » (précédée d’une flèche pointant vers la droite). La balise <html> est actuellement ouverte ; c’est pour ça que vous voyez qu’elle contient entre autres une balise <head></head> (qui est actuellement fermée) et la balise <body>.

Cliquez sur la flèche à gauche de <head></head> pour ouvrir la balise <head>. Son contenu est alors dévoilé : une balise <meta> sans contenu (donc sans flèche), une balise <title> avec contenu et une balise <style>.

Observez plus attentivement le contenu de cette fenêtre et comparez comment les informations de votre arbre (sur la feuille de papier) y sont représentées.

Si vous déplacez le curseur dans cette fenêtre qui décrit l’arborescence, les éléments HTML correspondants aux nœuds que votre curseur survole sont mis en évidence dans la page web.

## Étape 4 : comprendre des sélecteurs complexes

Il est temps de commencer les choses plus sérieuses, et l’utilisation de sélecteurs complexes. Vous trouverez ci-dessous une liste de sélecteurs. Pour chacun d’eux, effectuez les étapes suivantes :

1. lisez et tentez de comprendre le sélecteur (utilisez éventuellement votre arbre) ;
2. sans utiliser l’ordinateur, déterminez les éléments qui seront ciblés ;
3. pour vérifier votre réponse, ajoutez au document une règle mettant par exemple en couleur jaune sur fond noir les éléments ciblés par le sélecteur puis affichez le document dans Firefox.

Par exemple, pour tester le premier sélecteur, vous pouvez ajouter la règle CSS

p a { color: yellow; background-color: black }

Voici sept séries de sélecteurs à examiner.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Série 1 | | | a | p a | | b | .cadre a | | c | .cadre .important | | d | body > p | | e | p + p | | f | div.important | | g | div .important | | |  |  | | --- | --- | | Série 2 | | | a | .cadre, .important | | b | a, .important | | c | div + p | | d | div ~ p | | e | a + a + a | | f | #savoir + p a | | g | .important a + a | |
| |  |  | | --- | --- | | Série 3 | | | a | a:first-child | | b | p:last-child | | c | p :last-child | | d | p:last-of-type | | e | p :last-of-type | | f | :nth-child(3) | | g | div a:last-child | | h | p:first-child a | | |  |  | | --- | --- | | Série 4 | | | a | p[class] | | b | a[href\*="henallux"] | | c | a[href^="http://"] | | d | a[href\*="wiki"]:last-child | | e | .cadre a[href\*="wiki"] | | f\* | .cadre a:visited |   *(\*) Si la couleur d’arrière-plan ne s’applique pas, ajoutez une seconde règle ciblant a et fixant background-color* |
| |  |  | | --- | --- | | Série 5 | | | a | p:hover a | | b | p::first-letter | | c | p::first-line | | d | .cadre:hover p:first-letter | | e | a:hover + a | | |  |  | | --- | --- | | Série 6 | | | a | a:not(:first-child) | | b | p:not(:hover) a | | c | div.cadre :not(p) | | d | div.cadre > :not(p) | | e | :not(.cadre) a | | f | :not(.cadre) > p > a | | g | :not(.cadre, .important) | | h | body :not(.cadre,.important) | |
| |  |  | | --- | --- | | Série 7 | | | a | p:is(#savoir, .important) a | | b | p:has(a) | | c | p:not(:has(a)) | | d | p:has(+ p) | | e | p:has(a + a)::first-letter | | f | p:is(.important, :has([href\*="com"])) a | | g | p:has(a + a):not(:has(a + a + a)) | | |

## Étape 5 : écrire des sélecteurs complexes

À votre tour d’écrire des sélecteurs afin de cibler les éléments qui sont décrits ci-dessous. Pour tester vos réponses, utilisez naturellement Firefox !

1. tous les liens qui se trouvent dans un paragraphe de classe important ;
2. tous les liens déjà visités qui se trouvent dans le paragraphe d’identificateur savoir ;
3. tous les <div> qui suivent directement un paragraphe de classe important ;
4. tous les paragraphes de classe important qui se trouvent dans un élément de classe cadre ;
5. les premiers paragraphes de chaque <div> de classe cadre ;
6. tous les liens d’ordre pair au sein d’un paragraphe (c’est-à-dire à chaque fois le 2e lien, le 4e lien, le 6e lien, etc.) ;
7. tous les liens qui proviennent de Wikipedia et qui se trouvent dans un <div> de classe cadre ;
8. l’avant-dernier lien dans chacun des paragraphes qui ne sont pas de classe important ;
9. la première lettre des paragraphes qui contiennent au moins un lien provenant de Wikipedia, mais uniquement quand ces paragraphes sont survolés par le curseur ;
10. les paragraphes (dans ce cas-ci, il n’y en a qu’un seul) qui contiennent un lien qui provient de Wikipedia et un lien dont l’adresse comporte « w3 ».

## Étape 6 : after et before

Pour terminer cet exercice, deux règles CSS… Tentez de déterminer leur effet en les lisant puis vérifiez votre réponse en les ajoutant une par une au document HTML.

a::after {

color: green;

content: " (" attr(href) ") ";

}

.important::before {

content: url("warning.jpg") " ";

margin-right: 4px;

}

# (Exo) Bientôt le carnaval !

Objectifs

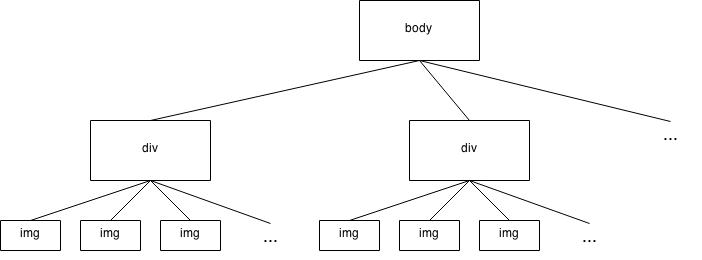
* utiliser des sélecteurs CSS complexes
* Sélecteurs basés sur les attributs et sur l’arborescence HTML

Cet exercice se base sur le fichier carnaval.html disponible dans les ressources pour ce laboratoire.

## Étape 1 : structure du document

Tout d’abord, prenez le temps pour vous familiariser avec la structure du document HTML. Celui-ci est décomposé en plusieurs blocs <div>, chacun comprenant un certain nombre d’images.

Sous forme d’arbre, cela donne la structure suivante.



## Étape 2 : règles CSS

Dans le document HTML, ajoutez une balise <style> et, à l’intérieur, divers styles permettant d’accomplir ce qui est demandé ci-dessous. Utilisez judicieusement les différents types de sélecteurs complexes.

Observez l’effet accompli après l’ajout de chacune des règles CSS.

1. Encadrez chacun des <div> d’une bordure (4 côtés) en trait plein (1 pixel d’épaisseur) de couleur noire. Ajoutez également un rembourrage (padding) de 8 pixels et une marge supérieure (margin-top) de 6 pixels pour aérer la présentation.
2. Encadrez chacune des images d’une bordure (4 côtés) en trait plein de couleur noire de 2 pixels d’épaisseur.
3. Ajoutez un identificateur à l’un des <div>. Utilisez cet identificateur pour rédiger une règle CSS ciblant les images qui se trouvent à l’intérieur de ce <div> et les encadrant d’une bordure (4 côtés) en trait plein de couleur orange de 2 pixels d’épaisseur.  
     
   Observez comment la seconde règle prend le pied sur la première dans le cas des images en question. Testez l’effet d’intervertir l’ordre des deux règles… la priorité est déterminée par la notion de spécificité du sélecteur. Dans les grandes lignes : plus le sélecteur est spécifique/précis et plus la règle sera considérée comme importante (quel que soit l’ordre dans lequel les règles sont citées). C’est un sujet qui sera approfondi dans un module ultérieur.
4. Les images sont trop collées. Pour les séparer, on peut leur ajouter une marge à gauche (margin-left) de 6 pixels. Cependant, si un <div> contient 4 images, seules les 3 dernières ont besoin de cette marge à gauche : inutile de l’ajouter à la première.  
     
   Rédigez une règle CSS qui cible toutes les images qui suivent directement une image et qui leur ajoute une telle marge à gauche.  
     
   Dans un second temps, réécrivez la règle en utilisant :not.
5. Les images sont affichées avec des tailles différentes, leur taille étant spécifiée dans le code HTML (ce qui n’est pas une bonne pratique en général, mais cela permet de s’exercer à la construction de sélecteurs ciblant les attributs).  
     
   Rédigez deux règles ciblant respectivement les images de 100 pixels de hauteur et celles de 150 pixels de hauteur. Pour centrer les images verticalement, ajoutez une marge inférieure (margin-bottom) de 50 pixels aux premiers et de 25 pixels aux seconds.
6. Dans une nouvelle règle, cibles les images qui se basent sur un fichier d’extension png (à l’opposé de jgp). Arrangez-vous pour qu’elles soient affichées sur un fond jaune.
7. Faites en sorte que la dernière image de chaque <div> soit encadrée en rouge (2 pixels d’épaisseur, trait plein).
8. Grâce à une nouvelle règle CSS, ajoutez un fond bleu clair (lightblue) à un <div> sur trois. Les <div> qui auront ce fond seront le 1er, le 4e, le 7e, le 10e, etc. Pensez à trouver une formule mathématique de la forme n qui couvre tous les cas !
9. Dans une nouvelle règle CSS, ciblez les images dont la source est un fichier .png et qui font 150 pixels de hauteur et donnez-leur un fond gris.
10. Finalement, ajoutez une règle pour que toute image survolée par le curseur de la souris soit affichée avec un fond vert et un cadre vert (2 pixels d’épaisseur, trait plein).

# (Exo) des tables plus stylées

Objectifs

* sélecteurs CSS complexes dans le cas des tables
* bordures, arrondis, propriétés composites

Cet exercice se base sur le fichier HTML tables.html.

## Étape 1 : structure du document

Observez le contenu du fichier : un titre et trois tables diverses.

## Étape 2 : titre

Dans un premier temps, ajoutez une section <style> et une règle CSS ciblant les titres <h1> dans l’en-tête du fichier HTML.

Faites en sorte que les titres <h1> soient affichés en caractères gras de taille 200% dans un cadre centré de 450 pixels de large avec un fond vert clair (lime).

Pour que le cadre en question soit centré, ajoutez les propriétés margin-left : auto et margin-right : auto. Pour que le texte soit centré à l’intérieur du cadre, ajoutez text-align : center. Finalement, histoire d’aérer la présentation, ajoutez un rembourrage de 4 pixels.

## Étape 3 : coins arrondis

Observez que le cadre vert est rectangulaire, avec quatre coins en angles droits.

CSS permet d’ajouter divers effets à de tels cadres et, pour bon nombre de ces effets, plusieurs méthodes existent. Testez les unes après les autres les propriétés suivantes (pour plus de clarté, enlevez chacune de ces propriétés avant de tester la suivante).

1. box-shadow: 10px 10px 5px green;  
   ensuite, faites varier les deux premières valeurs de cette propriété, qui indiquent la distance de « l’ombre » par rapport à la boîte originale (testez également des valeurs négatives) ; la troisième valeur, quant à elle, est utilisée pour le floutage de l’ombre.
2. Reprenez la propriété précédente mais en ajoutant également dans la valeur le mot-clef inset
3. box-shadow: inset 0 -1em 1em limegreen, 0 0 0 1px white, .3em .3em 1em green; pour avoir plusieurs ombres/bordures
4. border: 4px solid blue;
5. border-width: 4px;  
   border-style: solid;  
   border-color: blue;
6. border-style: solid;  
   border-color: blue;  
   border-width: 2px 4px 6px 8px;  
   *À quoi correspondent les 4 valeurs données pour border-width ? Notez l’ordre dans lequel les valeurs sont citées ; c’est un ordre utilisé dans plusieurs propriétés CSS.*
7. border-style: solid;  
   border-color: blue;  
   border-width: 6px 2px;  
   *À quoi correspondent les deux valeurs données pour border-width ?*
8. *Pouvez-vous déduire à l’avance l’effet des lignes suivantes ?*  
   border-style: solid dashed;  
   border-color: blue red black orange;  
   border-width: 6px;

Note : le point sur les bordures

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

* CSS permet de préciser trois paramètres pour chaque bord : la couleur (color), le style (style : trait plein, pointillé, tirets…) et l’épaisseur (width).
* On peut préciser les 3 paramètres en même temps, soit pour toutes les bordures (border: 1px solid black) soit de manière individuelle (border-left: 2px dashed red).
* On peut également préciser les paramètres individuellement, soit pour toutes les bordures (border-color: blue), soit pour une seule des bordures (border-bottom-color: red).
* On peut aussi le faire en groupant les bordures top/bottom d’une part et les bordures left/right d’autre part (border-color: blue red) ou en citant les valeurs pour les quatre bordures dans l’ordre top, right, bottom, left (border-color: blue red green orange).

## Étape 4 : coins arrondis

Finalement, CSS permet également de créer des coins arrondis plutôt que des angles droits. Pour ce faire, on indique le rayon de l’arrondi désiré. Comme pour les bordures, cela peut se faire de plusieurs manières différentes :

* soit en précisant le rayon pour chaque coin individuellement :

border-top-left-radius: 2px;  
border-top-right-radius: 4px;  
border-bottom-right-radius: 0; *0 pour un angle droit*  
border-bottom-left-radius: 3px;

* soit en faisant de même en une seule propriété (dans l’ordre top-left, top-right, bottom-right, bottom-left) :

border-radius: 2px 4px 0 3px;

* soit en utilisant la même valeur pour les 4 coins :

border-radius: 2px;

* soit en donnant deux valeurs (la première pour top-left et bottom-right, la seconde pour les deux autres coins) :

border-radius: 2px 3px;

Pour vous entraîner, ajoutez une règle au titre <h1> pour faire en sorte que celui-ci ait des coins arrondis de 6 pixels de rayon.

## Étape 5

Vous connaissez désormais tout ce qu’il faut pour donner un peu de style aux tableaux et, plus particulièrement, pour les afficher avec des coins arrondis.

Dans un premier temps, rédigez une règle CSS donnant un fond de couleur lemonchiffon à tous les <td> et <th> du document. Il s’agit juste d’une règle temporaire pour que vous puissiez bien voir l’effet de la règle suivante.

Ensuite, ajoutez une règle ciblant tous les tableaux (balise <table>) avec la propriété border-collapse: collapse. Observez ce que cette règle modifie : elle permet que les cellules ne soient pas des encadrés distincts en faisant en sorte que les bords communs soient fondus en une seule ligne.

Note : border-collapse

L’autre valeur possible pour border-collapse est separate, qui garde les  
bordures séparées. Avec cette option, vous pouvez également utiliser la propriété border-spacing qui indique l’espace entre les bordures de deux cellules voisines.

Ensuite, pour aérer la présentation, écrivez une règle qui cible tous les <th> et tous les <td> (une seule règle pour les deux) et qui leur ajoute 8 pixels de rembourrage.

Les deux règles suivantes préciseront le format des cellules-titres <th> et des cellules normales <td> et viendront en complément de la propriété padding commune exprimée dans la règle précédente. Utilisez les indications du tableau suivant.

|  |  |  |
| --- | --- | --- |
|  | Cellules titres | Cellules normales |
| Couleur de fond | #333366 | beige |
| Couleur du texte | white | black |
| Autre | gras |  |

## Étape 6 : plus d’arrondis !

Ajoutez des arrondis de 6 pixels de rayon aux quatre coins des tableaux.

Pour ce faire, vous aurez besoin de 4 règles, ciblant chacune des 4 cellules qui forment les coins du tableau.

Construisez les sélecteurs adéquats :

* Ciblez soit la première soit la dernière ligne <tr> d’un tableau.
* Dans cette ligne, ciblez soit la première soit la dernière cellule.
* Cette cellule pourrait être un <th> ou un <td> ; un pseudo-element :is pourrait être utile pour couvrir les deux cas.

Vérifiez ce que ça donne pour chacun des tableaux du document HTML.

## Étape 7 : mise en évidence au survol

Pour faciliter la lecture du tableau, arrangez-vous pour que toutes les cellules <td> de la ligne survolée par le curseur soient affichées avec un fond de couleur burlywood.

## Étape 8 : bordures intérieures

Pourquoi ne pas aller plus loin et ajouter un quadrillage intérieur au tableau, c’est-à-dire un quadrillage qui sépare les cellules mais qui n’existe pas sur les bords extérieurs du tableau (voir l’image ci-dessous).

A table with words on it

Description automatically generated

Cela revient à ajouter

* un bord droit beige à tous les th sauf ceux qui sont en fin de ligne ;
* un bord droit #333366 à tous les td sauf ceux qui sont en fin de ligne ;
* un bord inférieur beige à tous les th sauf ceux qui sont dans la dernière ligne ;
* un bord inférieur #333366 à tous les td sauf ceux qui sont dans la dernière ligne.

Transformez ces indications en 4 règles CSS. Pour traduire les parties exprimées sous la forme « sauf … », pensez à utiliser la pseudo-classe :not !

# Éléments de solution de l’exercice 1

