

Les sélecteurs CSS

Table des matières

I. Contexte	3
II. Le système de cascade	3
III. Exercice : Appliquez la notion	6
IV. L'héritage	7
V. Exercice : Appliquez la notion	9
VI. Sélecteurs et combinateurs	11
VII. Exercice : Appliquez la notion	15
VIII. Pseudo-classes	16
IX. Exercice : Appliquez la notion	17
X. Pseudo-éléments	18
XI. Exercice : Appliquez la notion	20
XII. Auto-évaluation	22
A. Exercice final	22
B. Exercice : Défi	23
Solutions des exercices	26
Contenus annexes	33

I. Contexte

Durée : 1 h

Environnement de travail : Visual Studio Code (cf. p.33)

Pré-requis :

- Connaître la notion de balise
- Avoir des notions de base en CSS
- Savoir rédiger une page HTML valide

Contexte

Nous allons voir les différentes façons de cibler des éléments pour leur appliquer un style particulier, et comment le style peut se transmettre de parent à enfants.

Le "C" de CSS signifie "*cascading*", « cascade » en français. Cela signifie que l'ordre d'apparition des règles dans le CSS a aussi son importance.

Nous verrons également comment le CSS gère les conflits de règles.

II. Le système de cascade

Objectifs

- Comprendre le système de cascade de CSS
- Comprendre comment sont gérés les conflits de style

Mise en situation

L'ordre de déclaration des règles CSS a une importance dans l'application de ces dernières.

Lorsque l'on souhaite appliquer une même propriété à un élément, le moment où il a été ciblé et la façon dont il l'a été sont des paramètres qui entrent en compte afin d'établir les priorités d'application.

Définition

L'ordre de déclaration

Lorsqu'un élément a été ciblé de la même façon à deux endroits différents, l'ordre de déclaration des règles CSS a une importance.

En effet, si l'on déclare deux fois une propriété sur le même élément, mais avec une valeur différente, le style qui sera appliqué sera celui contenu dans la règle arrivant en dernier dans le fichier CSS.

De même, si l'on importe plusieurs fichiers CSS avec la même règle sur un même élément, la règle qui sera prise en compte sera la dernière chargée par le navigateur. Il s'agira alors de celle contenue dans le fichier importé en dernier dans le <head>.

Exemple

Si nous déclarons deux fois une propriété sur la couleur du fond du paragraphe :

```
1 <p>Je suis un paragraphe</p>
2
3
4
5 p {
6   background-color: green;
7 }
```

Le style déclaré en dernier sera appliqué.

Je suis un paragraphe

Fondamental La spécificité

Si une même règle est appliquée avec une valeur différente sur une classe et sur un élément, celle qui sera prise en compte sera celle de la classe.

Une classe est en effet un sélecteur plus spécifique avec un poids plus important : l'ordre de déclaration n'est pas pris en compte dans ce cas.

```
1 <p class="ma-classe">Je suis un paragraphe</p>
2
3
4
5 .ma-classe {
6   background-color: red;
7 }
8
9 p {
10  background-color: green;
11 }
```

Je suis un paragraphe

Méthode Déterminer la spécificité : le poids des règles

Afin de déterminer quelles règles appliquer à un élément, le navigateur se base sur un poids, c'est-à-dire un score calculé en fonction de la spécificité du sélecteur.

De façon générale, plus un sélecteur est précis, plus celui-ci sera considéré comme prioritaire.

- Un sélecteur d'élément est peu spécifique : il cible tous les éléments d'un type donné dans la page, son score est donc faible.
- Un sélecteur de classe est plus spécifique : dans la page, il ne cible que les éléments dont l'attribut `class` a la valeur choisie, son score est donc plus important.

Par exemple, si nous ciblons un élément en utilisant son nom de balise et sa classe, la règle CSS prioritaire sera la plus précise. Ce sera donc celle dont le sélecteur est composé du nom de la balise et de la classe. En effet, son poids sera plus important que si nous avons seulement donné la classe ou le nom de l'élément.

Pour comprendre dans le détail comment sont effectués ces calculs, cet article¹ résume particulièrement bien la façon dont est calculé le poids.

Remarque

Il est possible d'assigner aux éléments HTML de multiples classes en les listant dans l'attribut class en séparant chaque classe d'un espace.

Si la même propriété est déclarée dans les deux règles, le conflit est résolu de la manière suivante : premièrement selon la règle de spécificité, ensuite selon l'ordre de déclaration du CSS. L'ordre des classes dans l'attribut class n'est pas pris en compte.

```
1 <p id="identifiant" class="classe1 classe2">Je suis un paragraphe</p>
```

```
1 p {  
2     color: white;  
3 }  
4  
5 .classe2 {  
6     text-decoration : underline ;  
7     background-color: blue;  
8 }  
9 .classe1 {  
10     background-color: green;  
11 }
```

Je suis un paragraphe

Attention

Il existe le mot-clé `!important` qu'il n'est pas recommandé d'utiliser, car il comporte le poids le plus fort, c'est-à-dire que, s'il est utilisé, nous ne pourrions pas surcharger la propriété CSS sur laquelle il est appliqué.

Cela pose un problème si nous souhaitons réutiliser notre code et modifier des éléments par surcharge.

Syntaxe À retenir

- En cas de conflit sur une règle CSS à appliquer au même élément ciblé, c'est le style déclaré en dernier qui sera appliqué.
- En cas de conflit sur des éléments ciblés de façons différentes, c'est la règle comportant le plus gros poids (c'est-à-dire la plus spécifique) qui sera appliquée.

Complément

Cascade et héritage²

1 <https://blog.organicweb.fr/comprendre-le-poids-des-regles-css/>

2 https://developer.mozilla.org/fr/docs/Apprendre/CSS/Building_blocks/Cascade_et_heritage#Comprendre_la_cascade

III. Exercice : Appliquez la notion

Question

[solution n°1 p.27]

Vous disposez du code HTML ci-dessous.

Travaillez la façon dont sont affichés les différents éléments dans le but d'obtenir l'affichage suivant :

- Par défaut, le contenu d'un paragraphe doit être justifié. Les paragraphes doivent également disposer d'un padding de 10 px. Leur couleur de fond doit être beige.
- Les paragraphes d'exemple (.sample) uniquement doivent être affichés sur fond gris clair (lightgray) et en italique.
- Les paragraphes de méthode (.method) uniquement doivent être affichés sur fond gris (gray).
- Quant aux titres disposant de cette même classe, ils doivent être soulignés.
- Le paragraphe de définition (#definition) doit quant à lui être spécifiquement affiché avec un fond orangé (sandybrown par exemple), un padding de 12 px et être encadré en noir.

Le code que vous produirez devra être externalisé dans un fichier .css.

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title>Le système de cascade</title>
6 </head>
7 <body>
8
9 <h1>Le système de cascade</h1>
10 <p>
11     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
12     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
13     ex ea commodo consequat.
14 </p>
15 <h2 class="definition">Définition</h2>
16 <p id="definition" class="method">
17     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
18     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
19     ex ea commodo consequat.
20 </p>
21 <h3 class="italic">Exemple</h3>
22 <p class="sample">
23     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
24     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
25     ex ea commodo consequat.
26 </p>
27 <h3>Remarque</h3>
28 <p>
29     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
30     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
31     ex ea commodo consequat.
32 </p>
33 <h2>Le poids des règles</h2>
34

```

```
35 <h3 class="method">Méthode</h3>
36 <p class="method">
37     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
38     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
39     ex ea commodo consequat.
40 </p>
41 <h3>Exemple</h3>
42 <p class="sample italic">
43     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
44     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
45     ex ea commodo consequat.
46 </p>
47 <h3>Remarque</h3>
48 <p>
49     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
50     ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip
51     ex ea commodo consequat.
52 </p>
53 </body>
54 </html>
55
```

IV. L'héritage

Objectif

- Comprendre le fonctionnement de l'héritage

Mise en situation

Les éléments peuvent recevoir du style sans avoir été précisément ciblés. En effet, le style peut se propager d'un élément parent vers ses enfants.

Définition	Héritage de styles
------------	--------------------

Certains styles peuvent se transmettre d'éléments parents à éléments enfants : il s'agit d'héritage de styles CSS.

Par exemple, la couleur du texte, la police ou encore l'alignement sont des propriétés qui vont se transmettre de parents vers enfants.

C'est-à-dire que, si le parent possède une couleur d'écriture verte, alors les enfants qu'il contient seront également affichés en vert, sauf si nous réécrivons le style de l'enfant.

Toutefois, toutes les propriétés CSS ne sont pas héritable : en effet, les règles sur les boîtes notamment (`padding` et `margin` par exemple) ne seront pas transposées aux enfants et ne s'appliqueront que sur le parent sélectionné.

Méthode

Pour savoir si une propriété peut être héritée ou non, rendez-vous sur la documentation¹ de cette propriété.

Pour chacune, il est systématiquement indiqué si celle-ci peut ou non être héritée.

Si l'on prend par exemple la documentation de la propriété `color`², on constatera qu'elle l'est.

Valeur initiale	Varie d'un navigateur à l'autre
Applicabilité	tous les éléments. S'applique aussi à <code>::first-letter</code> et <code>::first-line</code> .
Support	oui
Média	visuel
Valeur calculée	si la valeur est transsidente, la valeur calculée est la fonction <code>rgba()</code> correspondante. Sinon, la fonction <code>rgb()</code> correspondante. Le mot-clé transparent devient <code>rgb(0,0,0)</code> .
Type d'animation	une couleur
Ordre canonique	l'ordre unique et non-ambigu défini par la grammaire formelle

Exemple

Si nous appliquons la propriété de changement de couleur du texte sur la `<div>` parente qui contient des paragraphes `<p>`, cette propriété sera transmise aux enfants.

```

1 <div>
2   <p>
3     Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus
4     nisl, sed tincidunt enim scelerisque at.
5   </p>
6   <p>
7     Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies
8     libero vel suscipit.
9   </p>
10 </div>
11
12 div {
13   color: #c01b15;
14 }
```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus nisl, sed tincidunt enim scelerisque at.

Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies libero vel suscipit.

Il est toujours possible de surcharger cette propriété, mais il faudra être plus spécifique :

```

1 <div>
2   <p>
3     Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus
4     nisl, sed tincidunt enim scelerisque at.
5   </p>
6   <p class="green-text">
7     Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies
8     libero vel suscipit.
9   </p>
10 </div>
```

1 <https://developer.mozilla.org/fr/docs/Web/CSS/Reference>

2 <https://developer.mozilla.org/fr/docs/Web/CSS/color>


```

1 div {
2   color: #c01b15;
3 }
4
5 .green-text {
6   color: green;
7 }

```

Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus nisl, sed tincidunt enim scelerisque at.

Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies libero vel suscipit.

Syntaxe À retenir

- Certaines propriétés CSS appliquées aux parents peuvent, par héritage, s'appliquer aux enfants. Cependant, toutes les propriétés ne peuvent pas être transmises aux enfants.

V. Exercice : Appliquez la notion

Question

[solution n°2 p.27]

Vous disposez du code HTML et CSS ci-dessous.

Travaillez la façon dont sont affichés les différents éléments, dans le but d'obtenir l'affichage suivant :

- Par défaut, la police utilisée pour le corps du document est Calibri, bleue foncée (#000634)
- La police utilisée dans la section `.context` sera Tahoma, dans un bleu plus clair (#2657cf)
- Par défaut, les titres h2 et h3 seront de couleur bleue foncée (*darkblue*)
- Les éléments disposant de la classe `warning` seront affichés en rouge pâle (#cf3e4f) et en gras

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>L'héritage</title>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9
10 <h1>L'héritage</h1>
11
12 <section class="context">
13   <p>
14     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
15     incididunt ut labore et dolore
16     magna
17     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
18     aliquip ex ea commodo
19     consequat.
20   </p>
21 </section>

```

```

20
21 <section class="presentation">
22   <h2>Présentation</h2>
23   <p>
24     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
25     magna
26     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
27     consequat.
28   </p>
29 </section>
30
31
32 <section class="course-content">
33   <h2 class="definition">Définition</h2>
34   <p id="definition" class="method">
35     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
36     magna
37     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
38     consequat.
39   </p>
40
41   <h3 class="italic">Exemple</h3>
42   <p class="sample">
43     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
44     magna
45     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
46     consequat.
47   </p>
48
49   <h3>Remarque</h3>
50   <p>
51     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
52     magna
53     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
54     consequat.
55   </p>
56
57   <h2>Le poids des règles</h2>
58
59   <h3 class="method">Méthode</h3>
60   <p class="method">
61     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
62     magna
63     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
64     consequat.
65   </p>
66
67   <h3>Exemple</h3>
68   <p class="sample italic">
69     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
70     magna

```

```

71      aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo
72      consequat.
73      </p>
74
75      <h3>Remarque</h3>
76      <p>
77      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore
78      magna
79      aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo
80      consequat.
81      </p>
82
83      <h3 class="warning">Attention</h3>
84      <p class="warning">
85      Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore
86      magna
87      aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo
88      consequat.
89      </p>
90 </section>
91 </body>
92 </html>
93

```

```

1 p {
2   padding: 10px;
3   background-color: beige;
4 }
5
6 p.sample {
7   background-color: darkgray;
8   font-style: italic;
9 }
10
11 h3.method {
12   text-decoration: underline;
13 }
14
15 p.method {
16   background-color: gray;
17 }
18
19 #definition {
20   background-color: sandybrown;
21   padding: 12px;
22   border: 1px solid black;
23 }
24

```

VI. Sélecteurs et combineurs

Objectif

- Apprendre de nouveaux modes de sélection

Mise en situation

Il est possible de sélectionner plus précisément des éléments grâce à différents types de combineurs, qui permettent de réaliser des sélections avancées.

Définition Les sélecteurs

Avant de styliser un élément, il est important de le cibler en utilisant la méthode la plus juste.

Voici les principaux sélecteurs possibles et la manière de les utiliser :

- Sélecteur universel : `*` va sélectionner tous les éléments du document.
- Sélecteur d'élément : le nom de l'élément sans les chevrons va sélectionner tous les éléments de ce type.
- Sélecteur par identifiant : l'attribut `id=""` d'un élément va cibler l'élément possédant cet identifiant.
- Sélecteur par classe : l'attribut `class=""` d'un élément va cibler tous les éléments possédant cette classe.
- Sélecteur d'attribut : va cibler tous les éléments possédant un attribut particulier.

De manière générale, il est recommandé de ne pas cibler un élément en utilisant son identifiant, car le code CSS ne pourra pas être réutilisé.

La sélection la plus pérenne et la plus maintenable en cas de besoin de personnalisation particulier est l'utilisation des classes. De plus, il sera possible de donner plusieurs classes à un élément afin de lui appliquer plusieurs styles.

Méthode Sélecteur d'enfants

Il est possible de combiner des sélecteurs afin de sélectionner des éléments de façon plus précise.

À l'aide du combineur `>`, il est possible de sélectionner un élément qui sera directement issu du premier élément ciblé dans le sélecteur : *mon-élément* `>` *la-cible*.

Le style s'appliquera alors uniquement aux éléments *la-cible* directement rattachés à l'élément *mon-element*.

Attention, les sous-éléments d'enfants ne seront pas pris en compte.

Exemple

Voici du style appliqué grâce au sélecteur d'enfants.

On remarque que tous les enfants directs `<p>` de `<div>` seront colorés :

```
1 div > p {
2   color: #2cc0b4;
3   font-style: italic;
4 }

1 <div>
2   <p>
3     Premier enfant direct de div.
4   </p>
5   <span><p>
6     Deuxième enfant indirect de div.
7   </p></span>
8 </div>
9 <p>
10  Un paragraphe hors de div. Mais premier voisin de div.
11 </p>
12 <p>
13  Un paragraphe, second voisin de div.
14 </p>
```

Premier enfant direct de div.

Deuxième enfant indirect de div.

Un paragraphe hors de div. Mais premier voisin de div.

Un paragraphe, second voisin de div.

Méthode Sélecteur descendant

Ce sélecteur est représenté par un espace blanc entre les sélecteurs.

Contrairement au sélecteur d'enfants, le sélecteur descendant permet d'appliquer un style à tous les enfants d'un type descendant d'un parent.

Par exemple, si nous sélectionnons "*mon-élément .la-cible*", le style s'appliquera sur tous les éléments possédant la classe `.la-cible` et étant les descendants de l'élément "*mon-élément*".

Les enfants d'enfants seront pris en compte.

Exemple

Voici le même style appliqué après sélection via l'utilisation du sélecteur descendant, tous les enfants `<p>` de `<div>` sont colorés :

```
1 div p {  
2   color: #2cc0b4;  
3   font-style: italic;  
4 }
```

Premier enfant direct de div.

Deuxième enfant indirect de div.

Un paragraphe hors de div. Mais premier voisin de div.

Un paragraphe, second voisin de div.

Méthode Combinateur adjacent et combineur général "siblings"

Il existe également deux autres sélecteurs permettant de sélectionner les voisins d'un élément :

- Combinateur adjacent : matérialisé par le `+`, il est possible de sélectionner un élément se trouvant juste après un autre élément voisin ciblé : "*mon-élément + la-cible*", l'élément sélectionné sera l'élément "*.la-cible*" se trouvant juste après "*mon-élément*".
- Combinateur général : représenté par le `~`, il permet de sélectionner tous les éléments frères qui suivent un élément ciblé : "*mon-élément ~ la-cible*", va sélectionner tous les éléments "*.la-cible*" situés après "*mon-élément*".

Exemple

Si nous appliquons du style grâce au combinateur adjacent, on remarque que seul le voisin direct `<p>` de `<div>` est coloré :

```
1 div + p {
2   color: #2cc0b4;
3   font-style: italic;
4 }
```

Premier enfant direct de div.

Deuxième enfant indirect de div.

Un paragraphe hors de div. Mais premier voisin de div.

Un paragraphe, second voisin de div.

Si nous utilisons le combinateur général, tous les éléments `<p>` après l'élément `<div>` sont colorés en bleu :

```
1 div ~ p {
2   color: #2cc0b4;
3   font-style: italic;
4 }
```

Premier enfant direct de div.

Deuxième enfant indirect de div.

Un paragraphe hors de div. Mais premier voisin de div.

Un paragraphe, second voisin de div.

Remarque

Pour appliquer le même style à plusieurs éléments ou plusieurs classes, il est possible d'enchaîner les sélecteurs avec des virgules.

```
1 <h1>Lorem ipsum dolor sit amet</h1>
2 <p>Consectetur adipiscing elit. Vestibulum volutpat purus nisl, sed tincidunt enim scelerisque
   at.</p>
3 h1 {
4   color: #e67e22;
5   text-decoration: underline;
6 }
7 p, h1 {
8   text-align: center;
9 }
```

Ici, l'élément `<h1>` profitera de la propriété `text-align: center;` qu'il aura en commun avec `<p>`, en plus de ses règles propres.

Lorem ipsum dolor sit amet

Consectetur adipiscing elit. Vestibulum volutpat purus nisl, sed tincidunt enim scelerisque at.

Syntaxe À retenir

- Il est possible de préciser la sélection d'éléments en utilisant les combineurs, soit en sélectionnant l'enfant direct d'un élément avec ">", soit en sélectionnant tous les enfants d'un type descendant d'un élément avec un espace blanc.
- Il est également possible de sélectionner le voisin direct d'un élément à l'aide du "+", ou de sélectionner les voisins d'un même type grâce au "~".

VII. Exercice : Appliquez la notion

Question

[solution n°3 p.28]

Vous disposez du code HTML ci-dessous.

Travaillez la façon dont sont affichés les différents éléments dans le but d'obtenir les caractéristiques suivantes :

- Les liens présents dans l'élément `nav` ne doivent pas être soulignés et doivent être affichés en gras
- Les éléments `li` de l'élément `nav` doivent être affichés en ligne, sans style (`display: inline; list-style: none;`), de couleur verte (*green*), avec une taille de police de 14 px et une marge de 10 px
- Les liens présents dans les éléments `li` de l'élément `nav` doivent être affichés avec la couleur #2b2b2b
- Le lien de déconnexion présent dans l'élément `nav` doit quant à lui être affiché en marron (*brown*)

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Sélecteurs et combineurs</title>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9 <nav>
10   <a href="#">Déconnexion</a>
11   <ul class="site-navigation">
12     <li><a href="#">Accueil</a></li>
13     <li><a href="#">Page précédente - L'héritage</a></li>
14     <li>Sélecteurs et combineurs</li>
15     <li><a href="#">Page suivante - Pseudo - class</a></li>
16   </ul>
17 </nav>
18 </body>
19 </html>
20

```

VIII. Pseudo-classes

Objectif

- Découvrir l'utilisation des pseudo-classes

Mise en situation

Nous allons voir comment appliquer du style à des éléments lorsque ceux-ci se trouvent dans un état particulier.

Définition Pseudo-classe

Une **pseudo-classe** est une classe permettant de sélectionner un élément quand il est dans un état particulier.

Par exemple, nous remarquons que, lorsque l'on survole un lien, le style change. En effet, c'est le style de la pseudo-classe `:hover` qui sera appliqué à l'élément.

Il nous est possible de sélectionner ces pseudo-classes et d'en modifier le style. Pour cela, il faut sélectionner l'élément ou la classe, puis ajouter la pseudo-classe précédée de `:`.

Il existe de nombreuses pseudo-classes, nous ne citerons que les principales :

- `:active` sélectionne l'élément lorsqu'il est activé, notamment par le clic utilisateur.
- `:hover` cible l'élément au survol de la souris.
- `:checked` cible l'élément lorsqu'il est coché.
- `:focus` cible l'élément lorsque celui-ci est sélectionné par l'utilisateur.
- `:visited` détermine le style d'un lien lorsque l'utilisateur l'a déjà visité.

```
1 a:hover {
2
3 }
```

Exemple

Si nous modifions le style d'un élément lien, pour qu'au survol de la souris le style de ce dernier change :

```
1 <a href="#">Cliquez ici</a>
2
3 a:hover {
4   color: orange;
5   text-decoration: none;
6   font-weight: bold;
7   letter-spacing: 1px;
8 }
```

Cliquez ici

Complément

Il existe d'autres pseudo-classes permettant de sélectionner des éléments parents ou enfants, ou répondant à certains critères :

- `:first-child` sélectionne le premier élément d'un type par rapport à son parent.
- `:first-of-type` sélectionne le premier élément de son type dans le parent.
- `:not()` sélectionne un élément qui n'est pas l'argument entre parenthèses (il peut s'agir d'un autre élément).
- `:last-of-type` sélectionne un élément qui est le dernier de son type dans le parent.

Syntaxe **À retenir**

- Il est possible de sélectionner des éléments dans des états particuliers grâce au sélecteur : la pseudo-classe.
- Le style ne s'appliquera que lorsque l'élément se trouvera dans l'état ciblé.

Complément

Liste des pseudo-classes¹

IX. Exercice : Appliquez la notion**Question**

[solution n°4 p.28]

Vous disposez du code HTML et CSS ci-dessous.

Travaillez la façon dont sont affichés les différents éléments dans le but d'obtenir l'affichage suivant :

- Lorsque l'on maintient le clic sur le contenu d'un paragraphe, la couleur de fond suivante devra y être appliquée : `rgba(0, 255, 85, 0.15)`
- Les liens survolés devront apparaître en bleu (`#2657cf`) et soulignés

```

1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Sélecteurs et combinateurs</title>
6   <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9 <nav>
10  <a href="#">Déconnexion</a>
11  <ul class="site-navigation">
12    <li><a href="#">Accueil</a></li>
13    <li><a href="#">Page précédente - L'héritage</a></li>
14    <li>Sélecteurs et combinateurs</li>
15    <li><a href="#">Page suivante - Pseudo - class</a></li>
16  </ul>
17 </nav>
18
19 <h1>Sélecteurs et combinateurs</h1>
20
21 <section class="context">
22   <p>
```

¹ <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

```

23     Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
24     magna
25     aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
26     consequat.
27     </p>
28 </section>
29
30 <section class="presentation">
31     <h2>Présentation</h2>
32     <p id="presentation">
33         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
    incididunt ut labore et dolore
34         magna
35         aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
    aliquip ex ea commodo
36         consequat.
37     </p>
38
39     <h2>Objectifs</h2>
40     <ul>
41         <li>Lorem ipsum</li>
42         <li>Lorem ipsum</li>
43         <li>Lorem ipsum</li>
44     </ul>
45 </section>
46 </body>
47 </html>
48

```

```

1  /* Code déjà présent */
2  nav a {
3      font-weight: bold;
4      text-decoration: none;
5  }
6
7  nav > ul.site-navigation > li {
8      display: inline;
9      list-style: none;
10     color: green;
11     font-size: 14px;
12     margin: 10px;
13 }
14
15 nav > ul.site-navigation > li > a {
16     color: #2b2b2b;
17 }
18
19 nav > a {
20     color: brown;
21 }
22
23 body {
24     font-family: Calibri, sans-serif;
25     color: #000634;
26 }

```

X. Pseudo-éléments

Objectifs

- Découvrir les pseudo-éléments
- Savoir les utiliser pour styliser des éléments

Mise en situation

Nous allons voir comment appliquer du style à certaines parties d'un élément, grâce aux pseudo-éléments.

Définition Pseudo-élément

Le pseudo-élément n'est pas un élément à proprement parler, il s'agit d'un mot-clé permettant de sélectionner une certaine partie d'un élément.

Les plus utilisés sont les suivants :

- `::after` permet de créer un élément interne juste après l'élément sélectionné. Il faut associer la propriété CSS `content` à ce pseudo-élément pour ajouter des éléments, comme une icône qui va suivre un lien pour indiquer qu'il est externe au site. Attention, le contenu de ce pseudo-élément n'est pas sélectionnable par l'utilisateur.
- `::before` pareil que `::after`, mais pour `::before` l'élément créé précédera l'élément sélectionné.
- `::first-line` permet de sélectionner la première ligne d'un paragraphe.
- `::first-letter` sélectionne la première lettre d'une phrase.
- `::selection` permet d'appliquer des règles CSS à une portion du document qui a été sélectionnée par l'utilisateur (via la souris ou un autre dispositif de pointage).

Ces pseudo-éléments s'utilisent après le sélecteur souhaité, suivi de `: :pseudo-élément`.

```
1 p::first-letter {  
2  
3 }
```

Exemple

Par exemple, il est possible de sélectionner la première ligne d'un paragraphe en utilisant le pseudo-élément `::first-line` afin de colorer la première ligne en bleu et de passer le texte en italique :

```
1 <p>  
2   Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus nisl,  
3   sed tincidunt enim scelerisque at.  
4 </p>  
5 <p>  
6   Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies  
7   libero vel suscipit.  
8 </p>  
9  
10 p::first-line {  
11   color: #2cc0b4;  
12   font-style: italic;  
13 }
```

Tous les éléments `<p>` verront leur première ligne modifiée :

Lorem ipsum dolor sit amet consectetur adipiscing elit. Vestibulum volutpat purus nisl, sed tincidunt enim scelerisque at.

Nam pretium justo a purus vehicula, non feugiat mi faucibus. Nam malesuada ultricies libero vel suscipit.

Grâce au pseudo-élément `::after` associé à la propriété `content`, il est possible de styliser des liens de façon plus avancée, à l'aide de caractères Unicode par exemple.

```
1 <a class="travel-link" href="#">Mon blog de voyage</a>
1 .travel-link::after {
2     content: "\2708";
3 }
```

[Mon blog de voyage✈](#)

Syntaxe À retenir

- Il est possible de sélectionner une certaine partie d'un élément grâce aux pseudo-éléments : le style ne s'appliquera pas à l'élément entier, mais seulement à la partie de l'élément ciblée.
- Pour utiliser un pseudo-élément, il faut ajouter à l'élément ciblé : `pseudo-élément`.

Complément

Liste des pseudo-éléments¹

XI. Exercice : Appliquez la notion

Question

[solution n°5 p.29]

Vous disposez du code HTML et CSS ci-dessous.

Travaillez la façon dont sont affichés les différents éléments dans le but d'obtenir ceci :

- Le premier lien de la navigation devra être suivi du contenu suivant, permettant d'afficher un espace et une maison : `"\0020 \2302"`
- Les titres h2 seront précédés par une flèche, au moyen du contenu suivant : `"\21B3 \0020"`

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title>Sélecteurs et combinateurs</title>
6     <link rel="stylesheet" href="style.css">
7 </head>
8 <body>
9 <nav>
10     <ul class="site-navigation">
```

¹ <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-%C3%A9l%C3%A9ments>

```

11     <li><a href="#">Accueil</a></li>
12     <li><a href="#">Page précédente - L'héritage</a></li>
13     <li>Sélecteurs et combineurs</li>
14     <li><a href="#">Page suivante - Pseudo - class</a></li>
15     <li><a href="#">Déconnexion</a></li>
16 </ul>
17 </nav>
18
19 <h1>Sélecteurs et combineurs</h1>
20
21 <section class="presentation">
22     <h2>Présentation</h2>
23     <p id="presentation">
24         Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
25         incididunt ut labore et dolore
26         magna
27         aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
28         aliquip ex ea commodo
29         consequat.
30     </p>
31
32     <h2>Objectifs</h2>
33     <ul>
34         <li>Lorem ipsum</li>
35         <li>Lorem ipsum</li>
36         <li>Lorem ipsum</li>
37     </ul>
38 </section>
39 </body>
40 </html>

```

```

1 /* Code déjà présent */
2 nav a {
3     font-weight: bold;
4     text-decoration: none;
5 }
6
7 nav > ul.site-navigation > li {
8     display: inline;
9     list-style: none;
10    color: green;
11    font-size: 14px;
12    margin: 10px;
13 }
14
15 nav > ul.site-navigation > li > a {
16    color: #2b2b2b;
17 }
18
19 nav a:hover, nav > ul.site-navigation > li > a:hover {
20    color: #2657cf;
21    text-decoration: underline;
22 }
23
24 body {
25    font-family: Calibri, sans-serif;
26    color: #000634;
27 }

```

XII. Auto-évaluation

A. Exercice final

Exercice 1

[solution n°6 p.30]

Exercice

Si l'on déclare plusieurs fois une même propriété avec une valeur différente pour un même élément, laquelle sera appliquée ?

- ☐ La première propriété déclarée
- ☐ La dernière propriété déclarée

Exercice

Lorsqu'une même règle est appliquée à la fois sur une classe et sur un élément, qu'importe dans quel ordre, quelle règle sera appliquée ?

- ☐ Celle définie sur l'élément
- ☐ Celle définie sur la classe

Exercice

Parmi ces déclarations, laquelle dispose du poids le plus élevé ?

- ☐ *
- ☐ div
- ☐ div.classe
- ☐ #mon-element
- ☐ !important

Exercice

Lorsque l'on applique un style à un élément parent, ses enfants hériteront de l'ensemble des propriétés.

- ☐ Vrai
- ☐ Faux

Exercice

La déclaration suivante permet de cibler...

```
1 div p {}
```

- ☐ N'importe quel élément `p` contenu dans un élément `div`
- ☐ Les éléments `p` étant les premiers enfants d'un élément `div`

Exercice

La déclaration suivante permet de cibler...

```
1 nav > ul {}
```

- ☐ N'importe quel élément `ul` contenu dans un élément `nav`
- ☐ Les éléments `ul` étant les premiers enfants d'un élément `nav`

Exercice

La déclaration suivante permet de cibler...

```
1 div ~ p {}
```

- ☐ N'importe quel élément `p` contenu dans un élément `div`
- ☐ N'importe quel élément `p` situé au même niveau qu'un élément `div`

Exercice

La déclaration suivante permet de cibler...

```
1 a:visited {}
```

- ☐ N'importe quel lien survolé
- ☐ N'importe quel lien disposant de la classe `visited`
- ☐ N'importe quel lien visité par l'utilisateur

Exercice

Quelle déclaration convient-il d'utiliser lorsque l'on souhaite ajouter du contenu après un élément ?

- ☐ `element.after`
- ☐ `element:after`
- ☐ `element::after`
- ☐ `element#after`

Exercice

Quelle propriété convient-il d'ajouter lorsque l'on souhaite ajouter du contenu à un élément, grâce à une pseudo-élément par exemple ?

B. Exercice : Défi

Une agence de voyage souhaite revoir la présentation de sa page présentant ses séjours à destination de l'Europe et vous êtes chargé de préparer la charte graphique de cette page. Pour cela, vous disposez d'un rendu de cette page au format HTML représentant les données qui seront affichées sous leur forme brute.

Question

[solution n°7 p.32]

Voici les consignes dont vous disposez.

Concernant la page :

- Sauf contre-indication, la police utilisée sera Calibri
- La taille de la police sera 12 px
- Lorsqu'un élément est sélectionné, il disposera d'un fond vert transparent `rgba(134,183,31,0.2)`

Concernant le menu :

- Les éléments du menu seront affichés sur une seule ligne avec une marge de 10 px entre les éléments `li`
- Les liens seront colorés avec la couleur #B9482B et ne seront pas soulignés par défaut
- Au survol, ces liens seront soulignés
- Un lien visité sera coloré avec la couleur #FF00FF
- Les puces de la liste à puces ne seront pas affichés
- La police utilisée sera Tahoma de taille 14 px et le texte sera affiché en gras
- Une maison (content: "\0020 \2302") sera affichée après le premier lien

Concernant les titres :

- Les titres de premier niveau seront colorés avec la couleur #BC8E14
- Les titres de niveau 2/3/4 seront colorés avec la couleur #729E0E
- Les titres de niveau 4 seront soulignés
- Un avion sera affiché avant les titres de niveau 2 (content: "\2708") lorsqu'ils sont contenus dans un article

Concernant les paragraphes

- Les paragraphes étant directement des enfants d'une section disposeront d'un padding de 10 px et seront encadrés avec une bordure d'1 px grise
- Les paragraphes ne disposant pas de la classe `description` ou `information` seront écrits en italique

```

1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="UTF-8">
5     <title>Voyager en europe</title>
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <nav>
10      <ul class="menu">
11        <li><a href="#accueil">Accueil</a></li>
12        <li><a href="#voyages">Nos voyages</a></li>
13        <li><a href="#hebergements">Nos hébergements</a></li>
14      </ul>
15    </nav>
16    <section>
17      <h1>Destination l'europe</h1>
18      <p>
19        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
20        incididunt ut labore et
21        ullamco laboris nisi ut aliquip ex
22        ea commodo consequat
23      </p>
24      <h2>L'avis de Catherine</h2>
25      <p>
26        Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
27        incididunt ut labore et
28        ullamco laboris nisi ut aliquip ex
29        ea commodo consequat
30      </p>
31      <article>
32        <h2>Partir à Paris</h2>
33        <h3>Pourquoi cette destination</h3>

```



```

32         <p class="description">
33             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et
34             dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex
35             ea commodo consequat
36         </p>
37         <h3>Bon à savoir</h3>
38         <div class="informations">
39             <h4>Tarif</h4>
40             <p class="tarif">XXX €</p>
41             <h4>Avantages</h4>
42             <ul>
43                 <li>Lorem ipsum</li>
44                 <li>Lorem ipsum</li>
45                 <li>Lorem ipsum</li>
46             </ul>
47             <h4>Transports disponibles</h4>
48             <ul>
49                 <li>Lorem ipsum</li>
50                 <li>Lorem ipsum</li>
51                 <li>Lorem ipsum</li>
52             </ul>
53         </div>
54     </article>
55     <article>
56         <h2>Partir à Berlin</h2>
57         <h3>Pourquoi cette destination</h3>
58         <p class="description">
59             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et
60             dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex
61             ea commodo consequat
62         </p>
63         <h3>Bon à savoir</h3>
64         <div class="informations">
65             <h4>Tarif</h4>
66             <p class="tarif">XXX €</p>
67             <h4>Avantages</h4>
68             <ul>
69                 <li>Lorem ipsum</li>
70                 <li>Lorem ipsum</li>
71                 <li>Lorem ipsum</li>
72             </ul>
73         </div>
74     </article>
75     <article>
76         <h2>Partir à Rome</h2>
77         <h3>Pourquoi cette destination</h3>
78         <p class="description">
79             Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et
80             dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex
81             ea commodo consequat
82         </p>
83         <h3>Bon à savoir</h3>
84         <div class="informations">
85             <h4>Tarif</h4>

```

```

86         <p class="tarif">XXX €</p>
87     </h4>Avantages</h4>
88     <ul>
89         <li>Lorem ipsum</li>
90         <li>Lorem ipsum</li>
91         <li>Lorem ipsum</li>
92     </ul>
93 </div>
94 </article>
95 </section>
96 </body>
97 </html>

```

Voici le rendu attendu :

[Accueil](#)
[Nos voyages](#)
[Nos hébergements](#)

Destination l'europe

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

L'avis de Catherine

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

Partir à Paris

Pourquoi cette destination

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

Bon à savoir

Tarif

XXX €

Avantages

- Lorem ipsum
- Lorem ipsum
- Lorem ipsum

Transports disponibles

- Lorem ipsum
- Lorem ipsum
- Lorem ipsum

Partir à Berlin

Pourquoi cette destination

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat

Bon à savoir

Tarif

Solutions des exercices

p. 6 Solution n°1

Voici le fichier CSS que vous avez dû produire :

```
1 p {
2     text-align: justify;
3     padding: 10px;
4     background-color: beige;
5 }
6
7 p.sample {
8     background-color: lightgray;
9     font-style: italic;
10 }
11
12 h3.method {
13     text-decoration: underline;
14 }
15
16 p.method {
17     background-color: gray;
18 }
19
20 #definition {
21     background-color: sandybrown;
22     padding: 12px;
23     border: 1px solid black;
24 }
```

p. 9 Solution n°2

Voici le code que vous avez dû produire :

```
1 body {
2     font-family: Calibri, sans-serif;
3     color: #000634;
4 }
5
6 section.context {
7     font-family: Tahoma, sans-serif;
8     color: #2657cf;
9 }
10
11 h2, h3 {
12     color: darkblue;
13 }
14
15 .warning {
16     color: #cf3e4f;
17     font-weight: bold;
18 }
19
20 /* Code déjà présent */
21
22 p {
23     padding: 10px;
```

```

24     background-color: beige;
25 }
26
27 p.sample {
28     background-color: darkgray;
29     font-style: italic;
30 }
31
32 h3.method {
33     text-decoration: underline;
34 }
35
36 p.method {
37     background-color: gray;
38 }
39
40 #definition {
41     background-color: sandybrown;
42     padding: 12px;
43     border: 1px solid black;
44 }
45

```

p. 15 Solution n°3

Voici le code que vous avez dû produire :

```

1 nav a {
2     font-weight: bold;
3     text-decoration: none;
4 }
5
6 nav > ul.site-navigation > li {
7     display: inline;
8     list-style: none;
9     color: green;
10    font-size: 14px;
11    margin: 10px;
12 }
13
14 nav > ul.site-navigation > li > a {
15     color: #2b2b2b;
16 }
17
18 nav > a {
19     color: brown;
20 }
21

```

p. 17 Solution n°4

Voici le code que vous avez dû produire :

```

1 nav a:hover, nav > ul.site-navigation > li > a:hover {
2   color: #2657cf;
3   text-decoration: underline;
4 }
5
6 p:active{
7   background-color: rgba(0, 255, 85, 0.15);
8 }
9
10 /* Code déjà présent */
11 nav a {
12   font-weight: bold;
13   text-decoration: none;
14 }
15
16 nav > ul.site-navigation > li {
17   display: inline;
18   list-style: none;
19   color: green;
20   font-size: 14px;
21   margin: 10px;
22 }
23
24 nav > ul.site-navigation > li > a {
25   color: #2b2b2b;
26 }
27
28 nav > a {
29   color: brown;
30 }
31
32 body {
33   font-family: Calibri, sans-serif;
34   color: #000634;
35 }

```

p. 20 Solution n°5

Voici le code que vous avez dû produire :

```

1 nav > ul.site-navigation > li:first-child::after {
2   content: "\0020 \2302";
3 }
4
5 h2::before {
6   content: "\21B3 \0020";
7 }
8
9 /* Code déjà présent */
10 nav a {
11   font-weight: bold;
12   text-decoration: none;
13 }
14
15 nav > ul.site-navigation > li {
16   display: inline;

```

```

17 list-style: none;
18 color: green;
19 font-size: 14px;
20 margin: 10px;
21 }
22
23 nav > ul.site-navigation > li > a {
24 color: #2b2b2b;
25 }
26
27 nav a:hover, nav > ul.site-navigation > li > a:hover {
28 color: #2657cf;
29 text-decoration: underline;
30 }
31
32 body {
33 font-family: Calibri, sans-serif;
34 color: #000634;
35 }
36

```

Exercice p. 22 Solution n°6

Exercice

Si l'on déclare plusieurs fois une même propriété avec une valeur différente pour un même élément, laquelle sera appliquée ?

- ☐ La première propriété déclarée
- ☒ La dernière propriété déclarée

Exercice

Lorsqu'une même règle est appliquée à la fois sur une classe et sur un élément, qu'importe dans quel ordre, quelle règle sera appliquée ?

- ☐ Celle définie sur l'élément
- ☒ Celle définie sur la classe
- ☐ Un sélecteur de classe est un sélecteur plus spécifique avec un poids plus important, l'ordre de déclaration n'est pas pris en compte dans ce cas.

Exercice

Parmi ces déclarations, laquelle dispose du poids le plus élevé ?

- ☐ *
- ☐ div
- ☐ div.classe
- ☐ #mon-element
- ☒ !important

🔍 Même s'il est fortement déconseillé, c'est le mot-clé `!important`.

Exercice

Lorsque l'on applique un style à un élément parent, ses enfants hériteront de l'ensemble des propriétés.

☐ Vrai

☒ Faux

🔍 C'est faux, certaines règles, comme `padding` ou `margin`, ne seront pas transposées aux enfants.

Exercice

La déclaration suivante permet de cibler...

```
1 div p {}
```

☒ N'importe quel élément `p` contenu dans un élément `div`

☐ Les éléments `p` étant les premiers enfants d'un élément `div`

Exercice

La déclaration suivante permet de cibler...

```
1 nav > ul {}
```

☐ N'importe quel élément `ul` contenu dans un élément `nav`

☒ Les éléments `ul` étant les premiers enfants d'un élément `nav`

Exercice

La déclaration suivante permet de cibler...

```
1 div ~ p {}
```

☐ N'importe quel élément `p` contenu dans un élément `div`

☒ N'importe quel élément `p` situé au même niveau qu'un élément `div`

Exercice

La déclaration suivante permet de cibler...

```
1 a:visited {}
```

☐ N'importe quel lien survolé

☐ N'importe quel lien disposant de la classe `visited`

☒ N'importe quel lien visité par l'utilisateur

Exercice

Quelle déclaration convient-il d'utiliser lorsque l'on souhaite ajouter du contenu après un élément ?

- ☐ element.after
- ☐ element:after
- ☒ element::after
- ☐ element#after

Exercice

Quelle propriété convient-il d'ajouter lorsque l'on souhaite ajouter du contenu à un élément, grâce à une pseudo-élément par exemple ?

content

p. 23 Solution n°7

Voici le contenu que vous avez dû produire :

```

1 /* Page */
2 *::selection {
3     background-color: rgba(134, 183, 31, 0.2);
4 }
5
6 body {
7     font-family: Calibri, sans-serif;
8     font-size: 12px;
9 }
10
11 /* Menu */
12 nav > ul.menu > li {
13     display: inline;
14     list-style: none;
15     margin: 10px;
16     font-size: 14px;
17     font-weight: bold;
18     font-family: Tahoma, sans-serif;
19 }
20
21 nav > ul.menu > li > a {
22     text-decoration: none;
23     color: #B9482B;
24 }
25
26 nav > ul.menu > li > a:hover {
27     text-decoration: underline;
28 }
29
30 nav > ul.menu > li > a:visited {
31     text-decoration: underline;
32     color: #FF00FF;
33 }
34
35 nav > ul.menu > li:first-child::after {
36     content: "\0020 \2302";
37 }
38
39 /* Titres */
40 h1 {

```



```

41     color: #BC8E14
42 }
43
44 h2, h3, h4 {
45     color: #729E0E;
46 }
47
48 h4 {
49     text-decoration: underline;
50 }
51
52 article > h2::before {
53     content: "\2708";
54 }
55
56 /* Paragraphes */
57 section > p {
58     padding: 10px;
59     border: 1px solid gray;
60 }
61
62 p:not(.description):not(.informations) {
63     font-style: italic;
64 }

```

Contenus annexes

1. Visual Studio Code

Objectifs

- Comprendre pourquoi utiliser un éditeur de texte
- Découvrir et installer notre premier éditeur de texte

Mise en situation

Afin de créer notre premier site web, il va nous falloir télécharger les bons outils et comprendre en quoi ils nous sont utiles.

Définition Présentation

Les pages web HTML sont écrites dans des fichiers textes avec l'extension *.html*.

Ce sont des fichiers contenant du texte, au même titre qu'une image peut se trouver sous l'extension *.jpg* ou encore les chansons que l'on trouve dans des fichiers *.mp3*.

Il faut un logiciel pour les créer et les éditer : pour cela, nous utiliserons un éditeur de texte.

Conseil

On pourrait utiliser un éditeur de texte standard (Bloc-Notes, vi, emacs...), mais, pour des raisons de confort, nous utiliserons un éditeur de texte plus avancé : Visual Studio Code. Il permettra notamment de colorer automatiquement les éléments du code, afficher des aides, etc.

Installation

Maintenant, installons notre premier logiciel qui nous permettra d'écrire du HTML : <https://code.visualstudio.com/>.

À la fin de l'installation, ouvrons le logiciel Visual Studio Code :

- Cliquez sur le menu "File" puis "New file", un nouveau fichier vierge apparaît au centre.
- Sauvegardez le nouveau fichier au format *.html*. Par exemple, appelons-le *premiere-page.html*.
- Écrivez une phrase (par exemple, "Voici un fichier HTML") et sauvegardez.
- Double-cliquez sur le fichier pour l'ouvrir dans le navigateur ou effectuez un glisser-déposer vers votre navigateur préféré, le texte va s'afficher !

Les pages web HTML sont des fichiers textes presque comme les autres !

Syntaxe	À retenir
----------------	------------------

- Un éditeur de texte est un logiciel qui permet de créer et d'éditer des fichiers notamment des fichiers textes.
- Pour écrire du code HTML, des éditeurs plus adaptés que les éditeurs de texte standard existent, offrant de multiples fonctionnalités.