

HTML / CSS : exercices

Exercice 1 : mises en forme basiques

Les balises <h1>, <h2>, <h3>, etc. permettent respectivement d'insérer un titre 1, 2, 3, etc.

Exemple :

```
<h1>Mon titre 1</h1>
<h2>Mon titre 2</h2>
```

Les balises et permettent de mettre le texte en gras. Les balises <i> et permettent de mettre le texte en italique. Exemple :

```
<b>Texte en gras</b>
<strong>Texte en gras</strong>
<i>Texte en italique</i>
<em>Texte en italique</em>
```

Les balises et permettent respectivement d'insérer une liste ordonnée (numérotée) et non ordonnée (non numérotée). Les éléments qui composent une liste doivent être placés dans une balise . Exemple :

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ul>
```

- Element 1
- Element 2
- Element 3

```
<ol>
  <li>Element 1</li>
  <li>Element 2</li>
  <li>Element 3</li>
</ol>
```

1. Element 1
2. Element 2
3. Element 3

Sur base de ce que tu as appris précédemment, rédige un document HTML permettant d'obtenir le résultat suivant :

The Weeknd

Abel Tesfaye, dit *The Weeknd*, est un chanteur, auteur-compositeur-interprète et producteur musical canadien, né le 16 février 1990 à Toronto. Il commence sa carrière musicale en 2009 en publiant anonymement de la musique sur YouTube.

Sa discographie

- 2013: *Kiss Land*
- 2015: *Beauty Behind the Madness*
- 2016: *Starboy*
- 2020: *After Hours*
- 2022: *Dawn FM*

Exercice 2 : tableaux

Un tableau permet de structurer des données. Il se compose :

- d'une **ligne d'en-tête** : il s'agit de la première ligne du tableau ; elle contient le nom des colonnes et apparaît généralement en gras
- de plusieurs **lignes de contenu** : chaque ligne contient des données
- d'une **légende** : elle se place généralement en haut ou en bas du tableau

La création d'un tableau en HTML se fait de la manière suivante :

1. on insère une balise `<table>` (« table » signifie « tableau » en français) qui contiendra l'ensemble du tableau
2. dans cette balise, on insère plusieurs balises `<tr>` (« Table Row » qui signifie « ligne de tableau » en français) ; chacune représente une ligne dans le tableau
3. dans la première balise `<tr>`, on insère une balise `<th>` (« Table Header » qui signifie « en-tête de tableau » en français)
4. dans les balises `<tr>` suivantes, on insère plusieurs balises `<td>` (« Table Data » qui signifie « Donnée de tableau » en français) ; chacune représente une cellule dans le tableau
5. enfin, on insère une balise `<caption>` (« caption » signifie « légende » en français) pour ajouter une légende au tableau

Voici un petit exemple :

```
<table>
  <caption>
    Palmarès masculin des championnats de Belgique de cyclo-cross
    (2016-2018)
  </caption>
  <tr>
    <th>Année</th>
    <th>Or</th>
    <th>Argent</th>
    <th>Bronze</th>
  </tr>
  <tr>
    <td>2016</td>
    <td>Wout van Aert</td>
    <td>Laurens Sweeck</td>
    <td>Sven Nys</td>
  </tr>
  <tr>
    <td>2017</td>
    <td>Wout van Aert</td>
    <td>Kevin Pauwels</td>
    <td>Laurens Sweeck</td>
  </tr>
  <tr>
    <td>2018</td>
    <td>Wout van Aert</td>
    <td>Laurens Sweeck</td>
    <td>Daan Soete</td>
  </tr>
</table>
```

Palmarès masculin des championnats de Belgique de
cyclo-cross (2016-2018)

| Année | Or | Argent | Bronze |
|-------|---------------|----------------|----------------|
| 2016 | Wout van Aert | Laurens Sweeck | Sven Nys |
| 2017 | Wout van Aert | Kevin Pauwels | Laurens Sweeck |
| 2018 | Wout van Aert | Laurens Sweeck | Daan Soete |

Sur base de ce que tu as appris précédemment, rédige un document HTML permettant d'obtenir le résultat suivant :

| Année | Cours | Cours d'informatique à l'INDBG | |
|-------|-----------------------------------|--------------------------------|---------------------------------------------------------------|
| | | Matière | Exemples |
| 3TT | Programmation séquentielle Python | | Les structures répétitives/conditionnelles, les variables,... |
| 4TT | Programmation impérative | Le langage Javascript | Progra événementielle |
| 5TT | Programmation procédurale | Le langage C | Les types, les structures, les tableaux,... |
| 6TT | Base de données | Le langage SQL | Les clauses SELECT, FROM, WHERE,... |

Il s'agit bien d'un tableau, même s'il n'en a pas vraiment l'apparence... Pour ajouter des bordures, des couleurs, etc. il faut utiliser le CSS. Tu découvriras comment mettre en forme un tableau plus loin dans les exercices.

Exercice 3 : récap

Fabriquer une pioche

On peut fabriquer des outils, et entre autres des **pioches** avec différents matériaux. Du plus faible au plus fort, on trouve:

1. le bois, récolté sur les arbres;
2. la pierre, récoltée dans le sol;
3. le fer, miné dans le sous-sol;
4. les diamants, minés à très grande profondeur;

Que faire de cette pioche ?

La pioche permet alors de miner diverses ressources, qu'on trouve à des profondeurs (des couches) différentes. Cependant, *certaines ressources ne peuvent être récoltées qu'avec certains types de pioches !*

Ressources qu'il est possible de miner

Ressource Couche Qualité minimale

charbon de 5 à 128 bois

fer de 5 à 64 pierre

Chacune des ressources minées peut être utilisée sous deux formes:

- la forme *unitaire*: un morceau de charbon, un lingot de fer, un diamant...;
- la forme de *bloc*: un bloc constitué de 9 morceaux de charbon, un bloc de fer.

Exercice 4 : liens et images

Les liens permettent d'accéder à un autre document HTML (par exemple, une autre page internet).

L'insertion d'un lien se fait avec la balise `<a>` à laquelle on ajoute :

- l'attribut `href` : spécifie la destination du lien
- l'attribut `target` : spécifie l'endroit où doit être affiché la destination du lien

| | |
|-------------------------------------|-----------------------------|
| Lorsque <code>target</code> vaut... | L'ouverture se fait dans... |
| <code>_self</code> | le même onglet |
| <code>_blank</code> | un nouvel onglet |

Par exemple, le lien suivant ouvrira une page Wikipedia dans un nouvel onglet :

```
<a href="https://fr.wikipedia.org" target="_blank">Clique ici !</a>
```

L'insertion d'une image se fait avec la balise `` à laquelle on ajoute :

- l'attribut `src` : spécifie le chemin de l'image
- l'attribut `alt` : spécifie le texte alternatif à afficher si l'image ne peut pas être affichée

Par exemple :

```

```

Il est possible de se servir d'une image comme lien : lorsque l'utilisateur clique sur l'image, le lien s'ouvre. Pour cela, il suffit de placer la balise `` dans la balise `<a>`.

Par exemple :

```
<a href="https://fr.wikipedia.org" target="_blank">
  
</a>
```

Sur base de ce que tu as appris précédemment, rédige un document HTML permettant d'obtenir le résultat suivant :

Les liens HTML

En cliquant sur [ce lien](#), la page sera redirigée vers la page Wikipedia de mon dessert préféré

En cliquant sur [ce lien](#), une nouvelle page s'ouvrira vers la page Wikipedia de mon artiste préféré

En cliquant sur [ce lien](#), la page sera dirigée vers la photo de mon sportif préféré (plus bas dans la page)

Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde consequuntur eligendi quia beatae dignissimos maxime nostrum, eaque aliquam praesentium reiciendis totam labore in quae distinctio illum. Eius voluptatem laborum nisi? Lorem ipsum dolor sit amet consectetur adipisicing elit. Ipsam qui, animi, aspernatur reiciendis accusamus sint, praesentium exercitationem a omnis earum officiis similique consequatur dignissimos voluptate impedit eos iste. Facere, eveniet. Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti at, modi repellendus impedit praesentium placeat exercitationem! Laudantium libero impedit asperiores ut voluptatem eligendi dolor, excepturi sed unde nisi, nesciunt soluta! Lorem, ipsum dolor sit amet consectetur adipisicing elit. Omnis iste earum dolores necessitatibus quas obcaecati eos debitis quod accusamus corrupti, aliquid molestias, voluptas et quidem veritatis repellat voluptates corporis? Eum?

En cliquant sur cette image :



la page sera redirigée vers la page Wikipedia de mon sportif préféré.

L'image peut se pas s'afficher correctement (trop grande, trop petite, etc.). À l'aide de quel langage pourrait-on résoudre le problème ?

Exercice 5 : récap

La Coupe du Monde de Football



Quelques généralités

- **Sport** : football
- **Création** : 1928
- **Organisateur** : FIFA
- **Périodicité** : 4 ans

Introduction

La **Coupe du monde de football**, ou Coupe du monde de la FIFA, est le championnat du monde des équipes nationales masculines de football. Décidée le 28 mai 1928 par la Fédération internationale de football association (FIFA) sous l'impulsion de son président *Jules Rimet*, elle a été ouverte à toutes les équipes des fédérations reconnues par la FIFA, professionnelles y compris, se distinguant en cela du tournoi olympique de football, à l'époque réservé aux amateurs. Le tableau des palmarès se trouve à la [fin de cette page](#).

Histoire

Origine de la compétition

Le projet d'organiser une compétition internationale commence dès la création de la [Fédération internationale de football association](#) (FIFA) en 1904. En 1906, la première édition initiée par le dirigeant néerlandais Carl Hirschman est programmée en Suisse et quatre poules de quatre équipes en guise de premier tour sont mises en place. Mais lors de la clôture des confirmations d'inscriptions pour les seize sélections invitées, le 31 août 1905, aucune fédération ne confirme sa participation et le projet est enterré.

Première Coupe du monde (1930)

Pour la première édition de la Coupe du monde, la compétition se déroule en Uruguay à *Montevideo* en 1930. La FIFA lance les invitations et table sur l'inscription de seize équipes, mais seulement treize nations répondent favorablement pour participer au tournoi. Seuls quatre pays européens traversent l'océan Atlantique en bateau pour disputer la compétition.

Palmarès (depuis 2006)

Palmarès de la Coupe du monde de football

Année Vainqueur Finaliste Troisième

| | | | |
|------|-----------|-----------|-----------|
| 2006 | Italie | France | Allemagne |
| 2010 | Espagne | Pays-Bas | Allemagne |
| 2014 | Allemagne | Argentine | Pays-Bas |
| 2018 | France | Croatie | Belgique |
| 2022 | Argentine | France | Croatie |

Exercice 6 : la Coupe du Monde (version colorée)

Dans cet exercice, tu vas reprendre le document HTML de l'exercice précédent et lui ajouter un peu de couleurs grâce à CSS.

Tout d'abord, crée un fichier CSS nommé `style.css` dans le même dossier que le fichier HTML que tu as créé précédemment.

Pour « lier » le fichier CSS au fichier HTML, il faut ajouter la balise suivante dans la balise `<head>` de ton document :

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

Retourne ensuite dans ton fichier `style.css`. Ce fichier doit contenir un ensemble de **règles** de style qui s'appliqueront à des éléments (titre, lien, paragraphe, etc.) bien spécifiques de ton document. Ainsi, une règle CSS est toujours composée :

1. d'un **sélecteur**, qui spécifie quels éléments sont concernés par la règle (par exemple, uniquement les titres)
2. d'un **ensemble de propriétés** : le style qui vont s'appliquer sur ces éléments (par exemple, le texte en gras)

Par exemple, voici une règle CSS qui cible uniquement les paragraphes et qui fait en sorte que le texte soit affiché en gras :

```
p{  
    font-weight: bold;  
    text-align: center;  
}
```

Attention: les propriétés doivent être séparées par un point-virgule !

Tu viens de découvrir un premier sélecteur : le **sélecteur de type** qui permet de sélectionner des éléments sur base du nom de la balise. Il en existe un second qui te sera également utile pour la suite : le **sélecteur de groupement** qui permet de cibler plusieurs éléments pour une même règle (en les séparant par une virgule). Par exemple, voici une règle CSS qui cible uniquement les paragraphes et les titre 1 :

```
p, h1{  
    font-weight: bold;  
    text-align: center;  
}
```

Il existe encore un troisième sélecteur : le **sélecteur de pseudo-classe**. Il s'agit d'un mot-clé qui peut être rajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel l'élément doit être pour être ciblé par la règle CSS. Par exemple, voici une règle CSS qui cible uniquement les paragraphes qui sont survolés (*hover* en anglais) par la souris :

```
p:hover{
    font-weight: bold;
    text-align: center;
}
```

Prends connaissance de la liste des sélecteurs et des propriétés CSS et ajoute les règles CSS suivantes dans ton fichier `style.css` :

1. Pour le contenu entier (balise `<body>`) :
 1. le texte doit être de couleur blanche (`color: white`)
 2. le fond de couleur `#335799` (`background-color: #335799`)
 3. un rembourrage de `20px` des 4 cotés (`padding: 20px`)
2. Pour les titres 1 (balise `<h1>`) :
 1. le texte doit être de couleur `#f2c832` (`color: #f2c832`) et centré (`text-align: center`)
 2. une bordure inférieure pleine de `2px` d'épaisseur de couleur orange (`border-bottom: 2px solid orange`)
 3. la police doit être en petites majuscules (`font-variant: small-caps`) de taille `300%` (`font-size: 300%`)
3. Pour les titres 2 (balise `<h2>`) :
 1. le texte doit être de couleur `#f2c832` (`color: #f2c832`)
 2. un rembourrage à gauche de `4px` (`padding-left: 4px`)
 3. une bordure inférieure pleine de `1px` d'épaisseur de couleur orange (`border-bottom: 1px solid orange`)
 4. pour l'effet dégradé, tu peux ajouter la règle
`background: linear-gradient(#335799, #335799, #335799, orange)`
4. Pour les titres 3 (balise `<h3>`) : le texte doit être de couleur orange
5. Pour les liens (balise `<a>`) : le texte doit être de couleur orange
6. Pour les images (balise ``) : elles doivent être entourées d'une bordure noire pleine de `5px` (`border: 5px solid`)
7. Pour les cellules du tableau (balises `<th>` et `<td>`) :
 1. elles doivent être entourées d'une bordure pleine blanche d'`1px`
 2. un rembourrage de `3px` (4 cotés)

**Attention : tu ne dois rédiger qu'une seule règle pour ces deux éléments !
(utilise le sélecteur de groupement)**

8. Pour les cellules d'en-tête (balise <th>) :
 1. Le texte doit être de couleur #f2c832
 2. Le fond doit être de couleur noire
 3. La police doit être en petites majuscules
9. Pour les listes (balise) : les puces doivent avoir un style de cercle
(list-style-type: circle)
10. Pour les éléments de liste (balise) : un rembourrage inférieur de 3px
(padding-bottom: 3px)
11. Enfin, on souhaite cibler uniquement les liens qui sont survolés par le curseur de la souris et faire en sorte que le texte devienne de couleur #f2c832 à ce moment là. Pour cela, il faut utiliser une pseudo-classe. Il en existe énormément : essaie de trouver dans le tableau fourni celle qui pourrait t'être utile ! Vérifie ensuite sur ton document que ça fonctionne en passant ta souris sur un des liens.

Le résultat final se trouve à la page suivante.

LA COUPE DU MONDE DE FOOTBALL



Quelques généralités

- **Sport** : football
- **Création** : 1928
- **Organisateur** : FIFA
- **Périodicité** : 4 ans

Introduction

La **Coupe du monde de football**, ou Coupe du monde de la FIFA, est le championnat du monde des équipes nationales masculines de football. Décidée le 28 mai 1928 par la Fédération internationale de football association (FIFA) sous l'impulsion de son président *Jules Rimet*, elle a été ouverte à toutes les équipes des fédérations reconnues par la FIFA, professionnelles y compris, se distinguant en cela du tournoi olympique de football, à l'époque réservé aux amateurs. Le tableau des palmarès se trouve à la [fin de cette page](#).

Histoire

Origine de la compétition

Le projet d'organiser une compétition internationale commence dès la création de la [Fédération internationale de football association](#) (FIFA) en 1904. En 1906, la première édition initiée par le dirigeant néerlandais Carl Hirschman est programmée en Suisse et quatre poules de quatre équipes en guise de premier tour sont mises en place. Mais lors de la clôture des confirmations d'inscriptions pour les seize sélections invitées, le 31 août 1905, aucune fédération ne confirme sa participation et le projet est enterré.

Première Coupe du monde (1930)

Pour la première édition de la Coupe du monde, la compétition se déroule en Uruguay à *Montevideo* en 1930. La FIFA lance les invitations et table sur l'inscription de seize équipes, mais seulement treize nations répondent favorablement pour participer au tournoi. Seuls quatre pays européens traversent l'océan Atlantique en bateau pour disputer la compétition.

Palmarès (depuis 2006)

Palmarès de la Coupe du monde de football

| ANNÉE | VAINQUEUR | FINALISTE | TROISIÈME |
|--------------|------------------|------------------|------------------|
| 2006 | Italie | France | Allemagne |
| 2010 | Espagne | Pays-Bas | Allemagne |
| 2014 | Allemagne | Argentine | Pays-Bas |
| 2018 | France | Croatie | Belgique |
| 2022 | Argentine | France | Croatie |

Exercice 6 : températures

Tu as à ta disposition un fichier nommé `temperature.txt` qui contient du texte brut à propos des températures de cinq villes.

Dans un premier temps, crée un fichier `temperature.html` et copie le contenu de `temperature.txt` dans ce fichier. Place ensuite les balises HTML adéquates au bon endroit pour obtenir le résultat affiché à la page 20.

Ensuite, rédige les règles CSS suivantes :

1. Pour les titres 1, le texte doit être en vert (couleur green), la taille de la police doit être de 200% et le texte doit être encadré en par une bordure grise en pointillés (dotted) de 4px.
2. Pour les titres 2, le texte doit avoir une double (double) bordure inférieure verte de 4px et avoir la police Verdana (font-family: Verdana).
3. Pour les cellules d'en-tête, le texte doit être de couleur vert, en majuscule (text-transform: uppercase) et ne dois pas être en gras (par défaut, les cellules d'en-tête s'affichent en gras ; utilise font-weight: normal pour l'enlever)
4. Les cellules de contenu (balise `<td>`) doivent avoir un fond de couleur gris-clair (lightgray), une largeur de 100px avec un texte centré.

Jusqu'à présent, tu n'as utilisé que deux types de sélecteur : le sélecteur de type et le sélecteur de groupement. Il est temps d'en découvrir deux autres : le **sélecteur de classe** et le **sélecteur d'identifiant**.

Le sélecteur de classe permet de cibler des éléments d'un document qui appartiennent à une même classe. Pour faire en sorte qu'un élément appartienne à une classe, il faut le spécifier grâce à l'attribut `class`. Voici un exemple, où plusieurs éléments `<td>` appartiennent à la classe « chaud » :

```
<td class="chaud">21,3</td>
<td>10</td>
<td class="chaud">20,1</td>
```

Un élément peut appartenir à plusieurs classes à la fois. Il suffit de séparer les classes par des espaces. Voici un exemple, où l'élément `<td>` appartient à la classe « chaud » et à la classe « humide » :

```
<td class="chaud humide">21,3</td>
```

On peut ensuite cibler cet élément en CSS en plaçant un point avant le nom de la classe. Dans cet exemple, tous les éléments ayant la classe « chaud » auront leur texte en rouge :

```
.chaud{  
    color: red;  
}
```

À toi de jouer ! Modifie ton fichier HTML de telle manière que :

- toutes les cellules qui contiennent une température chaude (c'est-à-dire au dessus de 20 degrés) possèdent la classe « chaud »
- fais de même pour celles contenant une température négative (c'est-à-dire en dessous de 0 degré) pour qu'elles possèdent la classe « négatif »
- fais également en sorte que le nom des villes possèdent la classe « ville » ; pour cela, utilise la balise `` qui permet d'« emballer » un mot ou un groupe de mots

Modifie ensuite ton fichier CSS de telle sorte que :

- les éléments de classe « chaud » affichent leur texte en rouge , gras, avec un fond de couleur #eeeeee
- les éléments de classe « négatif » affichent leur texte en bleu, gras, avec un fond de couleur #eeeeee
- les éléments de classe « ville » affichent leur texte en couleur « olive » et en gras

Tu devrais obtenir le résultat suivant :

| | | | | |
|------|------|------|------|------|
| 7,9 | 3,5 | -7,9 | 4,8 | 7,2 |
| 8,5 | 3,8 | -7,6 | 4,9 | 7,8 |
| 10,9 | 6,4 | -2,8 | 6,7 | 10,6 |
| 13,5 | 9,9 | 5,7 | 9,4 | 13,8 |
| 17 | 13,5 | 13,2 | 12,7 | 17,9 |
| 21,3 | 16,6 | 17 | 15,7 | 22,5 |
| 23,8 | 18,5 | 20 | 17,8 | 25,2 |
| 23,9 | 18 | 18 | 17,3 | 25,4 |
| 20,9 | 15,3 | 12,3 | 15 | 21,1 |
| 17,4 | 11,8 | 5,7 | 11,8 | 17,1 |
| 12,1 | 7,3 | -0,5 | 7,8 | 12,6 |
| 8,6 | 4,2 | -4,5 | 5,3 | 8,5 |

Barcelone. Capitale adn comarque du Barcelonès

Bruxelles. Capitale de la cette ville est entourée d Bruxelles-Capitale.

Moscou. Capitale de la superficie de 2 510 km2

Londres. Capitale et plu de la Tamise dans le sud

Rome. Capitale de l'Itali Tyrrhénienne. Elle est ég

Il reste encore un sélecteur à découvrir : le **sélecteur d'identifiant**. Il permet de cibler un élément grâce à son identifiant (la valeur de l'attribut `id`). Voici un exemple où un paragraphe possède l'identifiant « `intro` » :

```
<p id="intro">Le tableau ci-dessous présente les températures moyennes relevées au fil des mois dans cinq villes.</p>
```

Attention : un même identifiant ne peut être attribué qu'à un seul élément HTML !

Dans l'exemple, il ne peut pas exister un deuxième élément qui possède l'identifiant « intro ».

On peut ensuite cibler cet élément en CSS en plaçant une dièse (#) avant le nom de l'identifiant. Dans cet exemple, l'élément identifié par l'identifiant « intro » aura son texte en gris :

```
#intro {  
    color: gray;  
}
```

C'est encore à toi de jouer ! Modifie ton fichier HTML de telle sorte que :

- Le paragraphe contenant le texte « Le tableau ci-dessous présente les températures moyennes relevées au fil des mois » soit identifié par l'identifiant « intro »
- La ligne du tableau (balise <tr>) contenant les températures moyennes soit identifié par l'identifiant « moyenne »

Modifie ensuite ton fichier CSS de telle sorte que :

- L'élément identifié par « intro » affiche son texte en gris et en italique (font-style: italic)
- Toutes les cellules (en-tête et contenu) qui se trouvent dans l'élément identifié par « moyennes » affichent leur texte en gras et soient entourées par une bordure grise pleine de 2px. Il faut utiliser ici un sélecteur particulier que tu découvriras en détail plus tard : le sélecteur de descendant. Pour l'instant, contentes-toi d'ajouter cette règle :

```
#moyennes td,  
#moyennes th {  
    font-weight: bold;  
    border: 2px solid gray  
}
```

Pour clôturer cet exercice, voici un petit tableau qui résume les sélecteurs déjà rencontrés :

| Sélecteur | pour cibler... | on écrit... | par exemple... |
|------------------|-----------------------------------------------|-----------------------------------|------------------|
| de type | tous les éléments de même balise | nomBalise | h1 |
| de groupement | plusieurs éléments de mêmes balises | nomBalise1, nomBalise2, ... | h1, p, div |
| de classe | tous les éléments ayant une classe donnée | .nomClasse | .chaud |
| d'identifiant | un élément identifié par un identifiant donné | #nomIdentifiant | #intro |
| de pseudo-classe | tous les éléments dans un état particulier | nomBalise:nomPseudoClasse | p:hover |

Le résultat final se trouve à la page suivante.

Températures mensuelles moyennes pour quelques villes

Présentation des villes

Barcelone. Capitale administrative et économique de la Catalogne, de la province de Barcelone, de la comarque du Barcelonès ainsi que de son aire et de sa région métropolitaines, en Espagne.

Bruxelles. Capitale de la Belgique, de la Communauté française de Belgique et de la Communauté flamande, cette ville est entourée de 18 autres communes étroitement imbriquées entre elles et formant la région de Bruxelles-Capitale.

Moscou. Capitale de la Russie, elle compte environ 12 600 000 habitants intra muros (2017) sur une superficie de 2 510 km², ce qui en fait la ville la plus peuplée à la fois du pays et d'Europe.

Londres. Capitale et plus grande ville d'Angleterre et du Royaume-Uni. La ville est située près de l'estuaire de la Tamise dans le sud-est de l'Angleterre.

Rome. Capitale de l'Italie, cette ville est située au centre-ouest de la péninsule italienne, près de la mer Tyrrhénienne. Elle est également la capitale de la région du Latium.

Les températures

Le tableau ci-dessous présente les températures moyennes relevées au fil des mois dans cinq villes.

Températures mensuelles moyennes dans ces cinq villes

| | BARCELONE | BRUXELLES | MOSCOU | LONDRES | ROME |
|-----|-----------|-----------|--------|---------|------|
| JAN | 7,9 | 3,5 | -7,9 | 4,8 | 7,2 |
| FÉV | 8,5 | 3,8 | -7,6 | 4,9 | 7,8 |
| MAR | 10,9 | 6,4 | -2,8 | 6,7 | 10,6 |
| AVR | 13,5 | 9,9 | 5,7 | 9,4 | 13,8 |
| MAI | 17 | 13,5 | 13,2 | 12,7 | 17,9 |
| JUI | 21,3 | 16,6 | 17 | 15,7 | 22,5 |
| JUI | 23,8 | 18,5 | 20 | 17,8 | 25,2 |
| AOÛ | 23,9 | 18 | 18 | 17,3 | 25,4 |
| SEP | 20,9 | 15,3 | 12,3 | 15 | 21,1 |
| OCT | 17,4 | 11,8 | 5,7 | 11,8 | 17,1 |
| NOV | 12,1 | 7,3 | -0,5 | 7,8 | 12,6 |
| DÉC | 8,6 | 4,2 | -4,5 | 5,3 | 8,5 |
| MOY | 15,5 | 10,7 | 5,7 | 10,8 | 15,8 |

Exercice 7 : quelques énigmes

Prends connaissance du document HTML fourni et observe sa structure. Tu remarqueras qu'il est découpé en plusieurs parties :

- un en-tête
- une liste d'énigmes qui contient une liste de question/solution
- un pied de page

As-tu remarqué la présence d'une nouvelle balise HTML ? La balise `<div>` !

Cette balise, tout comme la balise ``, n'a pas de signification particulière : elles permettent toutes les deux de cibler du contenu dans un document HTML pour lui appliquer des règles CSS.

Dans l'exemple précédent, tu as ciblé certains mots (le nom des villes) grâce à la balise `` pour leur assigner une classe particulière :

```
<span class="ville">Barcelone</span>
```

Ici, on ne souhaite pas cibler un ensemble de mots, mais bien un ensemble de balises. On utilise alors la balise `<div>` :

```
<div>
  <h1>Quelques énigmes</h1>
  <p>
    Vous trouverez ci-dessous une poignée d'énigmes de
    logique, de
    bon sens ou mathématiques qui devraient être à la portée
    de
    tous. Chacune des énigmes est suivie d'une solution que
    vous
    pourrez rendre visible en la survolant avec le curseur de
    la
    souris.
  </p>
</div>
```

À toi de jouer ! Place les balises <div> au bon endroit dans ton document HTML pour faire en sorte que :

- le titre 1 et le paragraphe qui constituent l'**en-tête** soient « emballés » dans un seul et même <div>
- tous les paragraphes qui constituent une **question** soient « emballés » dans un seul et même <div>
- les titres 2, les questions et les solutions qui constituent une **énigme** soient « emballés » dans un seul et même <div>
- la **liste des énigmes** (ensemble de toutes les énigmes) soient « emballés » dans un seul et même <div>
- le paragraphe et le lien qui constituent le **pied de page** soient « emballés » dans un seul et même <div>

En-tête

Quelques énigmes

Vous trouverez ci-dessous une poignée d'énigmes de logique, de bon sens ou mathématiques qui devraient être à la portée de tous. Chacune des énigmes est suivie d'une solution que vous pourrez rendre visible en la survolant avec le curseur de la souris.

Enigme

💡 Les trois comprimés

— Je vous prescris trois comprimés. Vous en prendrez un toutes les demi-heures.
— Pendant combien de temps, docteur ?
— Calculez vous-même !

Question

Réponse

La plupart de ces énigmes sont directement tirées ou, tout au moins, inspirées du site [Rustrel](#)

Pied de page

Maintenant, modifie tes fichiers CSS et HTML pour faire en sorte que :

- **le corps du document** (balise <body>) ait un fond noir et un texte noir
- **l'en-tête, la liste des énigmes et le pied de page** aient un fond de couleur « lemonchiffon », une largeur de 800px, et un rembourrage de 6px. Pour centrer le tout, tu peux ajouter les deux propriétés margin-left:auto et margin-right:auto.
- **l'en-tête** ait une marge inférieure de 4px.
- **le pied de page** ait une marge supérieure de 4px, un texte gris en italique et aligné au centre.
- **chaque énigme** ait une bordure noire en pointillés de 2px, une marge supérieure et inférieure de 12px et un rembourrage de 4px
- **chaque solution** ait un texte de couleur « lemonchiffon » (même couleur que le fond afin de la cacher), une bordure orange de 1px, un rembourrage de 4px et une marge inférieure de 0px (utilise une classe pour ne cibler que les paragraphes qui constituent une réponse !). Lorsque le pointeur de la souris passe dessus, le texte devient noir et la solution est ainsi révélée (utilise la bonne pseudo-classe !)
- **l'image du gâteau** ait une largeur de 200px (utilise un identifiant !)
- **les chiffres de la question « Tour de nombres »** soient centrés et de police « Courier » (utilise un identifiant et une <div> !)
- **chaque question** soit affichée en italique
- **les titres 1** aient un texte de taille 200%, un fond de couleur orange et un rembourrage de 3px.
- **les titre 2** aient un fond de couleur « palegoldenrod », un rembourrage à gauche de 40px et une marge supérieure de 0px. Pour ajouter l'image de l'ampoule, tu peux ajouter ces 3 règles :

```
background-image: url('ampoule.png');  
background-repeat: no-repeat;  
background-size: auto 100%;
```

Le résultat final se trouve à la page suivante.

Quelques énigmes

Vous trouverez ci-dessous une poignée d'énigmes de logique, de bon sens ou mathématiques qui devraient être à la portée de tous. Chacune des énigmes est suivie d'une solution que vous pourrez rendre visible en la survolant avec le curseur de la souris.

Le trou

Combien de mètres cubes de terre contient un trou de 6m de diamètre et de 3m de profondeur ?

Les trois comprimés

— Je vous prescris trois comprimés. Vous en prendrez un toutes les demi-heures.

Au survol de la souris, les réponses apparaissent :

Quelques énigmes

Vous trouverez ci-dessous une poignée d'énigmes de logique, de bon sens ou mathématiques qui devraient être à la portée de tous. Chacune des énigmes est suivie d'une solution que vous pourrez rendre visible en la survolant avec le curseur de la souris.

Le trou

Combien de mètres cubes de terre contient un trou de 6m de diamètre et de 3m de profondeur ?

Zéro, car il s'agit d'un trou : il ne contient donc pas de terre !

Les trois comprimés

— Je vous prescris trois comprimés. Vous en prendrez un toutes les demi-heures.

commues. Y participent un humain, un martien, un klingon et un cthulhien. Malgré leurs physionomies différentes, les quatre races en présence ont un point en commun : tous possèdent des mains à cinq doigts.

Selon le protocole, l'humain doit serrer la main de chacun des trois autres participants... seulement, le problème est qu'il ne peut y avoir de contact direct ou indirect entre des races différentes, car cela pourrait entraîner la propagation de virus potentiellement mortels pour l'une ou l'autre des races.

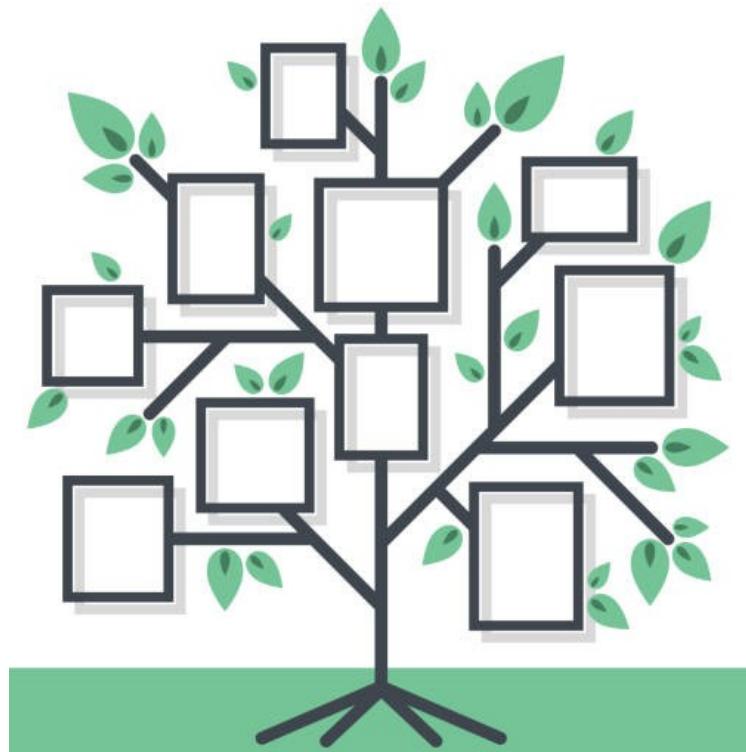
L'humain ne possède qu'une paire de gants... comment peut-il s'en sortir pour serrer la main de chacun des trois extra-terrestres sans mettre en danger qui que ce soit ?

La plupart de ces énigmes sont directement tirées ou, tout au moins, inspirées du site [Rustrel](#)

Exercice 8 : arborescence

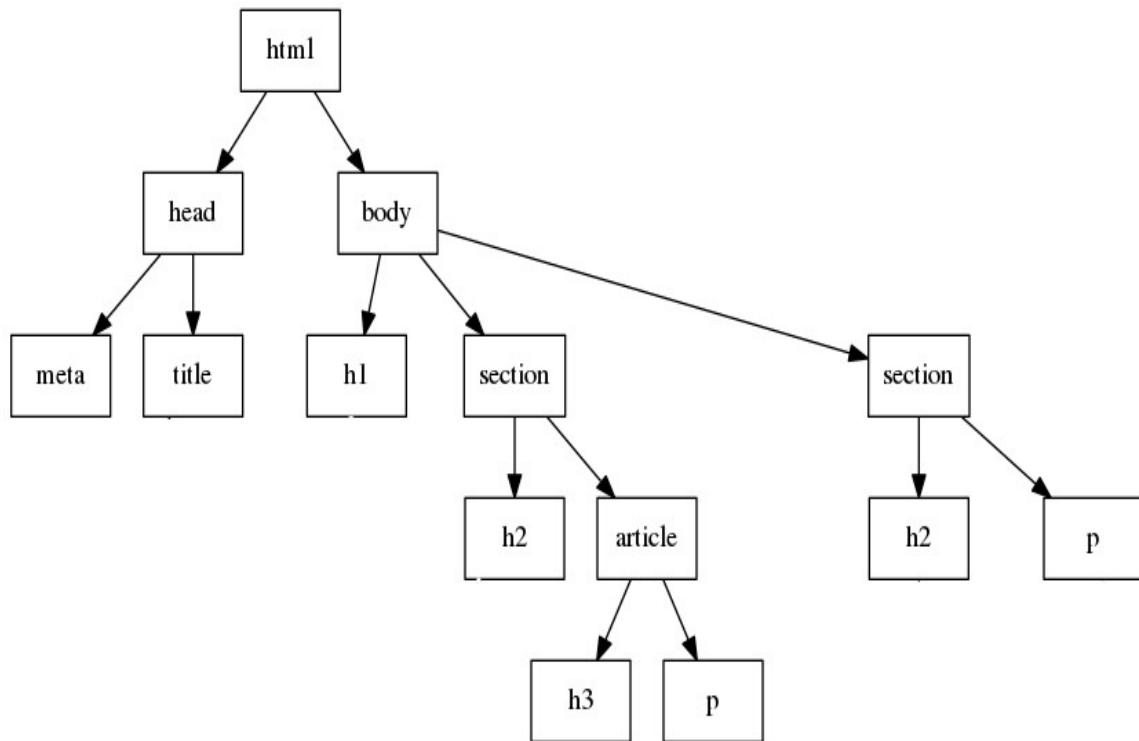
As-tu déjà observé la structure d'un arbre ? Il est composé de **nœuds**, qui relient les branches entre elles, et de **feuilles**, qui sont les points terminaux des certaines branches.

Complète les cases blanches du schéma¹ ci-dessous en indiquant « F » s'il s'agit d'une feuille et « N » s'il s'agit d'un nœud.



¹ <https://istockphoto.com>

Un document HTML est structuré de la même manière : on appelle ça une **arborescence**.



Dans cet exemple :

- **html** est la **racine**
- **body** et **head** sont les **enfants** (*child* en anglais) de **html**
- **meta** et **title** sont les enfants de **head** et les **descendants** de **html**
Attention, **body** et **head** sont eux aussi les descendants de **html** !
- **h1** et **section** (le plus à droite) sont **frères** (*sibling* en anglais)
- **meta** et **title** sont **frères adjacents** (il sont l'un à côté de l'autre)

Selon toi :

- combien d'enfants possède **section** (le plus à droite) ?
- combien de descendants possède **html** ?
- combien de frères possède **html** ?
- **head** et **body** sont-ils des frères adjacents ?
- qui est le parent de **article** ?
- les deux **p** sont-ils des feuilles ?
- **h3** est-il une feuille ?

Maintenant que tu en sais plus sur l'arborescence d'un document HTML, il est temps de découvrir quatre nouveaux sélecteurs CSS.

Le premier est le **sélecteur de descendant** qui permet de cibler les éléments qui sont les descendants d'un autre élément. Il suffit pour cela de séparer les sélecteurs par un espace blanc. Par exemple, voici une règle qui cible tous les paragraphes qui sont les descendants d'un titre 1 :

```
h1 p{  
    text-align:center  
}
```

Il est possible de « chaîner » plusieurs sélecteurs. Par exemple, voici une règle qui cible tous les paragraphes qui sont les descendants d'un titre 1 qui est lui-même le descendant d'un élément identifié par « chiffres » :

```
#chiffres h1 p{  
    text-align:center  
}
```

Un deuxième sélecteur est le **sélecteur d'enfant** qui permet de cibler les éléments qui sont les enfants (descendant direct) d'un autre élément. Il suffit pour cela de séparer les sélecteurs par un chevron (symbole >). Par exemple, voici une règle qui cible tous les paragraphes qui sont les enfants d'un titre 1 :

```
h1 > p{  
    text-align:center  
}
```

Le troisième sélecteur est le **sélecteur de frère** qui permet de cibler les éléments qui sont les frères d'un autre élément. Il suffit pour cela de séparer les sélecteurs par un tilde (symbole ~). Par exemple, voici une règle qui cible tous les paragraphes qui sont les frères d'un titre 1 :

```
h1 ~ p{  
    text-align:center  
}
```

Enfin, le quatrième sélecteur est le **sélecteur de frère adjacent** qui permet de cibler les éléments qui sont les frères adjacents d'un autre élément. Il suffit pour cela de séparer les sélecteurs par un plus (symbole +). Par exemple, voici une règle qui cible tous les paragraphes qui sont les frères adjacents d'un titre 1 :

```
h1 + p{  
    text-align:center  
}
```

À toi de jouer ! Maintenant que tu en sais plus sur l'arborescence d'un document HTML, prends connaissance de la structure du document fourni et dessine ci-dessous son arborescence.

Sur base de ton dessin, réponds aux questions suivantes :

- Combien d'enfants possède le premier <div> ?
- Combien de descendants possède le paragraphe désigné par le sélecteur `#savoir` ?
- Combien d'éléments HTML sont ciblés par le sélecteur `.important` ?
- Indiquez le père de chacun des éléments HTML ciblés par le sélecteur `.important`
- Indiquez le nombre de frères/sœurs de chacun des éléments HTML ciblés par le sélecteur `.important`
- Quel est l'élément HTML qui possède le plus de fils ?
- Quel est l'élément HTML qui possède le plus de descendants ?

Sur base de ton dessin, détermine les éléments qui seront ciblés par les sélecteurs suivants :

- p a
- .cadre a
- body > p
- p + p
- div + p
- div ~ p
- div.important
- div .important
- a:first-child
- p:last-child
- p:nth-child(3)
- p:hover

Écris les sélecteurs permettant de cibler les éléments décrits ci-dessous.

- tous les liens qui se trouvent dans un paragraphe de classe « important »
- tous les liens qui se trouvent dans le paragraphe identifié par « savoir »
- tous les div qui suivent directement un paragraphe de classe « important »
- tous les paragraphes de classe « important » qui se trouvent dans un élément de classe « cadre »
- les premiers paragraphes de chaque div de classe « cadre »

Exercice : créatures de Minecraft

Prend connaissance de la structure du document HTML et dessine ci-dessous son arborescence.

Observe les effets après l'ajout de chacune des règles CSS suivantes :

- Encadre chacun des div d'une bordure (4 cotés) en trait plein (1px d'épaisseur) de couleur noire. Ajoute également un rembourrage (padding) de 8px et une marge supérieure (margin-top) de 6px.
- Encadre chacune des images d'une bordure (4 cotés) en trait plein de couleur noire et de 2px d'épaisseur.
- Ajoute un identificateur à l'un des div. Utilise cet identificateur pour ajouter une règle CSS qui cible les images qui se trouvent à l'intérieur de ce div. Encadre ces images d'une bordure (4 cotés) en trait plein de couleur orange de 2px d'épaisseur.
- As-tu remarqué que les images sont trop collées ? Pour les séparer, tu peux leur ajouter une marge à gauche (margin-left) de 6px, **sauf à la première** (indice : c'est l'élément qui n'est pas frère adjacent).
- Les images sont affichées avec des tailles différentes. Rédige deux règles qui ciblent les images de 100px de hauteur (height) ainsi que celles de 150px. Pour les centrer verticalement, ajoute une marge inférieure (margin-bottom) de 50px à celles de 100px et de 25px pour celles de 150px.
- Crée une nouvelle règle CSS qui cible les images avec une extension .png. Arrange-toi pour qu'elles soient affichées sur un fond jaune. As-tu que remarqué que les sorcières et les zombies ne sont pas affichées sur fond jaune ?
- Fais en sorte que la dernière image de chaque div soit encadrée en rouge (2px d'épaisseur, trait plein).
- Grâce à une nouvelle règle CSS, ajoute un fond bleu clair (lightblue) au 2ième div (et uniquement celui-là).
- Dans une nouvelle règle CSS, cible les images dont la source est un fichier .png **et** qui font 150 pixels de hauteur. Donne-leur un fond gris.
- Pour finir, ajoute une règle pour que toute les images survolées par le curseur de la souris soit affichée avec un fond vert et un cadre vert (2px d'épaisseur, trait plein).

Exercice : capacités de Salamèche

Prends connaissance de la structure du document et rédige les règles CSS adéquates.

Règles pour le titre

- Le titre doit être affiché en caractère gras de taille 200% dans un cadre centré de 450px de largeur avec un fond bleu clair (lightblue).
- Pour centrer le cadre, utilise les propriétés `margin-left` et `margin-right` et donne-leur la valeur `auto`.
- Pour center le texte à l'intérieur du cadre, utilise la propriété `text-align` et donne-lui la valeur `center`.
- Pour aérer la présentation (éviter que le texte soit collé contre le cadre), ajoute un rembourrage (padding) de 4px de chaque coté.

Observe le cadre : il est rectangulaire avec 4 coins en angles droits. Teste chacune des règles les unes après les autres (enlève à chaque fois la règle avant de tester la suivante) :

```
box-shadow: 10px 10px 5px green
```

```
border: 4px solid blue
```

```
border-width: 4px;  
border-style: solid;  
border-color: blue;
```

```
border-style: solid;  
border-color: blue;  
border-width: 2px 4px 6px 8px;
```

```
border-style: dashed;  
border-color: blue red black orange;  
border-width: 6px;
```

Avec CSS, on peut également créer des coins arrondis plutôt que des angles droits. Pour cela, on indique le rayon (radius) de l'arrondi désiré. Comme pour les bordures, cela peut se faire de plusieurs manières différentes :

- en précisant le rayon de chaque coin individuellement :

```
border-top-left-radius: 2px;  
border-top-right-radius: 4px;  
border-bottom-right-radius: 0;  
border-bottom-left-radius: 3px;
```

- en faisant de même en une seule propriété :

```
border-top-left-radius: 2px 4px 0 3px;
```

- en utilisant la même valeur pour les 4 coins :

```
border-radius: 2px;
```

Pour t'entraîner, ajoute une règle au cadre du titre pour faire en sorte que celui-ci ait des coins arrondis de 6px de rayon.

Règles pour le tableau

- Fais en sorte que la couleur de fond du tableau soit du rose clair (`lightpink`). Ajoute-lui un rembourrage de 5px et arrondis ses coins (3px de rayon)
- Ajoute un fond blanc aux cellules de contenu (balise `<td>`) et fais en sorte que le texte soit de couleur bleue (`blue`)
- Fais en sorte que toutes les cellules (balises `<td>` et `<th>`) aient des bords arrondis (3px de rayon) et que le texte contenu à l'intérieur de celles-ci soit centré
- Les cellules d'en-tête (balise `<th>`) ainsi que la colonne des niveaux doivent être de fond corail clair (`lightcoral`), de couleur noire, en gras avec un rembourrage de 5px. Comment faire pour cibler les cellules de cette colonne ? Indice : il s'agit des cellules qui sont les premiers enfants de chaque ligne
- Les cellules contenant « Niveau » et « Attaques » doivent avoir un fond de couleur rouge. Comment faire pour cibler ces cellules d'en-tête sans prendre en compte les autres (par exemple celle contenant « Griffes ») ? Indice : ces deux cellules sont les éléments descendants de la première ligne (balise `<tr>`) du tableau

Capacités de Salamèche

Exercice : quizz sur HTML

Dans cet exercice, tu réaliseras une page HTML de A à Z qui contiendra un Quizz sur HTML.

Partie 1

1. Crée deux fichiers : `quizz.html` et `quizz.css`
2. Rédige les parties standards du document
3. Dans la balise `<body>`, insère un formulaire en utilisant la balise `<form>`
4. À l'intérieur du formulaire, ajoute les uns après les autres les étiquettes (balise `<label>`) des champs (balises `<input>`, `<textarea>`, `<select>`) suivants
 - a) Un champ textuel de nom et d'identificateur « `prenom` »
 - b) Crée une étiquette indiquant « Prénom » et lie cette étiquette au champ textuel grâce à l'attribut `for`
 - c) Fais de même pour le nom. Comment faire pour éviter que tout n'apparaisse sur une seule ligne ?
 - d) Pour vérifier que les étiquettes sont bien liées aux champs, tu peux cliquer sur l'une de ces étiquettes : le curseur doit normalement se positionner sur le champ correspondant
 - e) Ajoute une troisième étiquette et un troisième champ qui contiendra le mail de l'utilisateur
 - f) Ajoute un nouveau paragraphe commençant par les mots « Mon groupe ». Après viendront plusieurs options (bouton radio) qui permettent de choisir une option de l'INDBG . Chaque option sera composée d'un bouton radio et d'une étiquette associée à ce bouton indiquant le code de l'option. Tous les boutons radios doivent partager le même groupe mais avec des identificateurs différents.
g) Insère un bouton de soumission qui contient le texte « Envoyer »
5. Fais en sorte que les données de ton formulaire soient envoyées vers un serveur. Pour cela, modifie la balise du formulaire de cette manière :
`<form action="https://httpbin.org/post" method="POST">`
6. Remplis le formulaire et appuie sur le bouton "Envoyer". Observe ce qu'il se passe.
7. Essaie maintenant de soumettre le formulaire avec des données incorrectes:
 - a) adresse mail invalide
 - b) champs videsRemarques-tu les messages d'erreur ?

Partie 2

Avant d'ajouter de nouvelles informations dans ton formulaire, il faut le structurer. Le formulaire sera structuré de cette manière :

- « Mes informations », qui contiendra les champs insérés lors de la partie 1 (nom, prénom, etc.)
- « Question 1 », qui contiendra une question sur les balises HTML
- « Question 2 », qui contiendra une question sur les hyperliens

Pour « découper » un formulaire en plusieurs parties, il faut utiliser la balise **<fieldset>** et la balise **<legend>** de cette manière :

```
<form>
  <fieldset>
    <legend>Ma section 1</legend>
    <!-- On insère ici les champs -->
  </fieldset>
  <fieldset>
    <legend>Ma section 2</legend>
    <!-- On insère ici les champs -->
  </fieldset>
</form>
```

Utilise adéquatement ces deux balises pour structurer le formulaire comme demandé et place au bon endroit les champs insérés lors de la partie 1.

Partie 3

Ajoute les informations nécessaires pour répondre à la question 1 :

1. Dans la section « Question 1 », ajoute un paragraphe qui indique la question « Parmi les mots suivants, indique lesquels sont des balises HTML »
2. Toujours dans cette section, ajoute cinq cases à cocher : head, style, margin, href et div. Ces 5 cases doivent être cochées automatiquement au chargement de la page.

Attention, puisqu'il s'agit d'un choix multiple, chacune des cases doit avoir un identifiant (id) et un nom (name) différent !

Quelle est la différence entre des boutons radio et des cases à cocher ?

Ajoute les informations nécessaires pour répondre à la question 2 :

1. Dans la section « Question 2 », ajoute un paragraphe qui indique la question « Quel est l'attribut permettant de créer un lien hypertexte ? »
2. Ajoute une liste déroulante permettant de choisir entre href, url, name et www

Quelle est la différence entre une liste déroulante, des boutons radio et des cases à cocher ?

Avant le bouton « Envoyer », ajoute un second bouton permettant de réinitialiser le formulaire.

Partie 4

Ajoute les règles CSS permettant d'obtenir le résultat proposé à la page suivante :

1. Largeur de 600px pour le formulaire (cible la balise <form>).
2. Tout le contenu du document doit avoir la couleur de fond #cad006 et avoir la couleur de texte #005ca8
3. Les sections ont la couleur de fond #fcce0d, une bordure de 2 pixels d'épaisseur en traits solides de couleur #de0546, un arrondi de coin de 6 pixels. Ajoute-leur également une ombre grise (avec la propriété box-shadow : 4px 4px 2px gray) et une marge en bas de 12px.
4. Faits en sorte que les titres des sections soient écrits en utilisant une fonte de type cursive (propriété font-family), à 250% de la taille normale et en gras. Ajoute-leur aussi une ombre grâce à la propriété text-shadow : 2px 2px #de0546.
5. Ciblez les champs dans lesquels il est possible d'écrire (c'est-à-dire la liste, et les champs de type text ou email). Donne-leur une couleur de fond « wheat », une bordure de couleur #cad006 en trait plein de 1px d'épaisseur et un arrondi de coin de 6 pixels.
6. Finalement, cible les boutons. Faits en sorte que leur texte utilise une police de type cursive, ajoute-leur une bordure de 2px d'épaisseur en trait plein de couleur #de0546, un arrondi d'angle de 6 pixels et une ombre box-shadow : 2px 2px 1px #de0546.

Attention. Pour les points 4 et 5, tu devras sans doute utiliser un sélecteur sur la balise <input> en ne ciblant que les éléments qui possèdent une valeur particulière de l'attribut « type ».

Comment faire pour faire en sorte que les deux boutons soient affichés à droite du formulaire plutôt qu'à gauche ?

Indice : il faut les « emballer » dans un div et faire en sorte que l'alignement de ce div soit à droite.

Mes informations

Mon prénom :

Mon nom :

Mon e-mail : Entre ton adresse mail

Mon option :

- Informatique
- Sciences-économiques
- Sciences-sociales
- Sciences

Question 1

Parmi les mots suivants, indiquez lesquels sont des balises HTML

- Head
- Style
- Margin
- Href
- Div

Question 2

Quel est l'attribut permettant de créer un lien ?

name

Exercice : Pokédex

Dans cet exercice, tu vas réaliser un formulaire permettant l'enregistrement d'un Pokémon dans un Pokédex (l'encyclopédie des Pokémons). Le résultat à obtenir se trouve à la page suivante.

Instructions pour le contenu (HTML) :

1. Le formulaire doit comprendre **un champ textuel obligatoire** permettant d'enregistrer le nom du Pokémon. Ce champ doit comporter l'instruction « Nom du Pokémon ».
2. Ensuite, le formulaire doit comprendre une **section nommée « Informations générales »** qui contient quatre champs :
 1. Un **champ de type nombre** qui contient la taille du Pokémon. La valeur minimale de ce nombre est 0.
 2. Un **champ de type nombre** qui contient le poids du Pokémon. La valeur minimale de ce nombre est 0.
 3. Une **liste déroulante** qui représente la catégorie du Pokémon (un Pokémon ne peut se trouver que dans une seule catégorie) proposant les choix suivants : « Graine », « Lézard », « Flamme », « Tortue » et « Autre »
 4. Deux **boutons radio** représentant le sexe du Pokémon : mâle ou femelle.

Attention : pour faire en sorte que ces 4 champs soient bien alignés, utilise un tableau !

3. Ensuite, le formulaire doit comprendre une **section nommée « Types »** qui contient un seul champ : **des cases à cocher** où chaque case représente un type (un Pokémon peut être de plusieurs types à la fois). Les types disponibles sont : « Plante », « Poison », « Feu », « Vol », « Eau », « Insecte », « Ténèbres », « Electrik », « Sol » et « Glace » (tu peux en ajouter d'autres si tu le souhaites)
4. Ensuite, le formulaire doit comprendre une **section nommée « Description »** qui contient un seul champ : une zone de texte avec l'indication « Ajoutez une description ». La zone de texte doit comporter exactement 44 colonnes et 4 lignes.
5. Enfin, le formulaire doit comprendre un bouton de soumission avec le texte « Enregistrer dans le Pokédex »

Instructions pour la forme (CSS) :

1. Le formulaire doit avoir une couleur de fond beige, une marge à droite et à gauche de 400px ainsi qu'un rembourrage (4 cotés) de 4px.
2. Les sections doivent avoir un fond de couleur #30a7d7, des coins arrondis (5px), une marge supérieure et inférieure de 10px.
3. Les légendes doivent être de couleur blanche, avec un fond de couleur #30a7d7, des coins arrondis (5px), une bordure (4 cotés) de 2px de couleur #c0c0c0 ainsi qu'une police de taille 120%.
4. Les champs de type nombre ainsi que la liste déroulante doivent avoir une largeur de 100px.
5. Tous les champs, la liste déroulante et le champ de texte libre doivent avoir une couleur de fond « powderblue » ainsi que des bords arrondis (5px)
6. Le champ permettant d'insérer le nom du Pokémon doit être aligné au centre de la page (utilise un identifiant pour cibler l'élément). Sa hauteur doit être de 25px et sa largeur de 200px
7. Le bouton de soumission doit être aligné au centre de la page. Lorsque le curseur de la souris passe dessus, la couleur du texte doit devenir beige et la couleur de fond doit devenir #30a7d7. Pour cela, utilise la bonne pseudo-classe CSS.
8. La zone contenant l'ensemble des cases à cocher est un peu grande en hauteur car il y a beaucoup d'éléments à l'intérieur. Pour faire en sorte que cette zone prenne moins de place, on peut lui imposer une hauteur, par exemple 100px. Le problème, c'est que certains éléments se trouvent alors en dehors de cette zone : on dit qu'ils « débordent » (*overflow* en anglais). Quelle propriété CSS pourrais-tu utiliser pour faire en sorte que le débordement vertical (axe des Y) devienne défilable ?

Nom du Pokémon

Informations générales

Taille (en m) :

Poids (en kg) :

Catégorie : Graine ▾

Sexe : Femelle Male

Types

- Plante
- Poison
- Feu
- Vol
-



Description

Ajoutez une description

Enregistrer dans le Pokédex

Exercice : le flux naturel en HTML

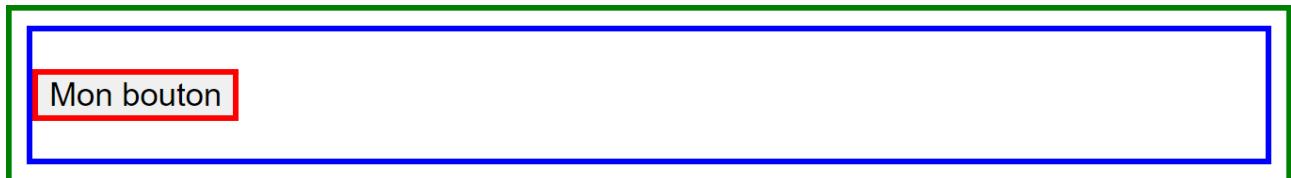
L'objectif de cet exercice est d'observer le comportement normal (c'est-à-dire le mode d'affichage) de quelques balises HTML.

Il existe deux grands modes d'affichage en HTML :

- **l'affichage « block »** : l'élément prend le plus d'espace possible sur sa ligne



- **l'affichage « inline »** : l'élément prend uniquement l'espace nécessaire sur sa ligne



Certaines balises s'affichent en mode « inline » alors que d'autres s'affichent en mode « block » par défaut.

Sur base du document fourni, classe les balises dans le tableau ci-dessous.

| Affichage « inline » | Affichage « block » |
|----------------------|---------------------|
| | |

Il est possible de changer le mode d'affichage par défaut grâce à la propriété CSS `display`.

Ainsi, la règle suivante fait en sorte que les paragraphes s'affichent en mode « inline » :

```
p{  
  display: inline;  
}
```

Et celle-ci fait en sorte que les images s'affichent en mode « block » :

```
img{  
    display: block;  
}
```

Ajoute la règle suivante et observe ce qu'il se passe :

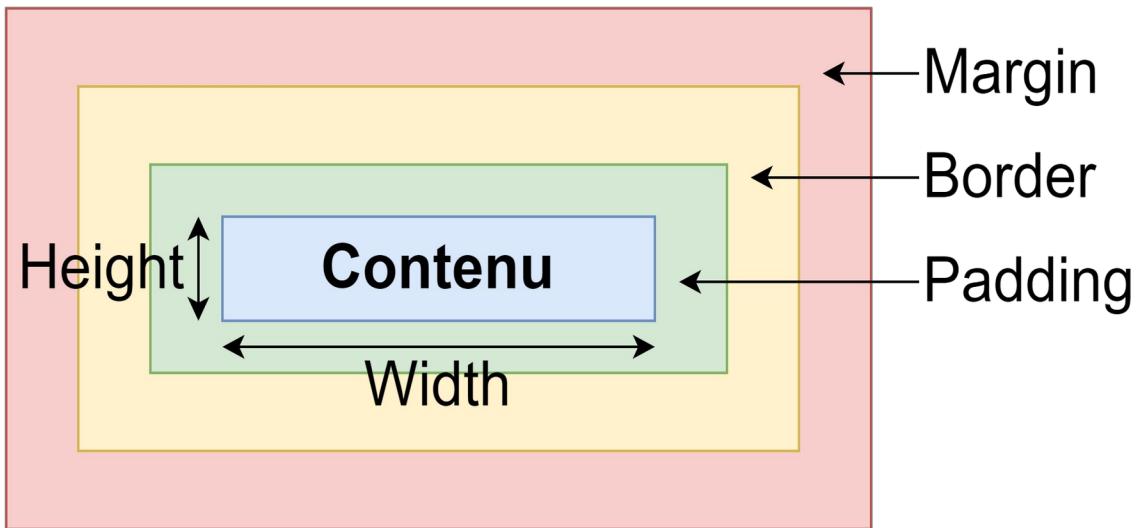
```
body * {  
    display: inline  
}
```

Pour terminer ce petit exercice introductif, voici un petit récapitulatif de la situation...

| Affichage « block » | Affichage « inline » |
|-------------------------------------|------------------------------------------|
| Empilés les uns sur les autres | Affichés à la suite les uns des autres |
| Précédés d'un passage à la ligne | Retour à la ligne si plus assez de place |
| Largeur = toute la place disponible | Largeur = dépend du contenu |
| Occupent un espace rectangulaire | |

Exercice : le box model

Tous les éléments occupent un espace rectangulaire (en mode « inline » ou « block »). La taille de cet espace et l'espacement qui l'entoure sont déterminés par le **box model**.



Padding est le rembourrage autour du contenu. Elle utilise la couleur de fond (background-color).

Border est la bordure qui encadre le contenu avec son rembourrage.

Margin correspond à la marge c'est-à-dire l'espace autour de la bordure qui sépare l'élément des autres éléments. Elle n'a pas de couleur de fond : elle est transparente !

Height et **Width** correspondent respectivement à la hauteur et la largeur du contenu. Ces deux propriétés ne peuvent être utilisées qu'en mode « block ».

Attention, il ne s'agit pas de l'espace total occupé !

$$\text{Espace total} = \text{contenu} + \text{padding} + \text{border} + \text{margin}$$

À toi de jouer ! Commence par créer un document HTML dans lequel tu ajoutes le code suivant :

```
<div id="div1">DIV1</div>
<div id="div2">DIV2</div>
<div id="div3">DIV3</div>
```

Pour rendre l'espace occupé par chaque div plus facilement repérable, fait en sorte que les trois div aient une couleur de fond différente.

Par défaut, comment sont affichés les div ? En mode « inline » ou « block » ?



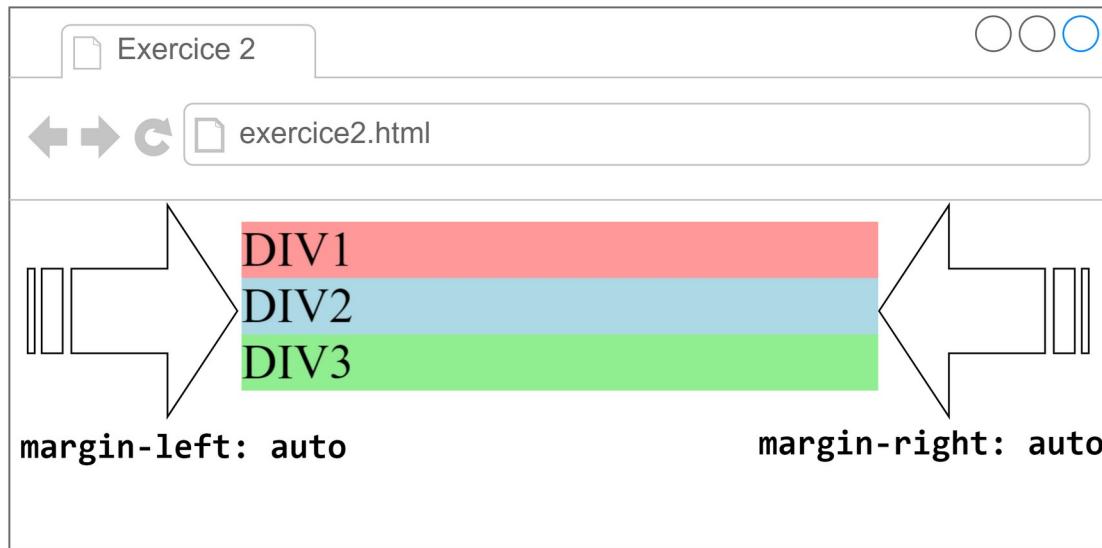
Pour certains sites web, il peut être dérangeant de laisser les div occuper toute la largeur de la fenêtre car dans ce cas, l'apparence change selon que la fenêtre est plus ou moins étirée. Fait le test sur ton navigateur : si tu réduis la fenêtre, les div adaptent leur largeur automatiquement (car ils prennent toute la place disponible).

Un astuce pour régler ce problème consiste à emballer tout le contenu dans un div dont la largeur sera imposée (grâce à la propriété width). On appelle ce div un « emballeur » ou un « wrapper » en anglais.

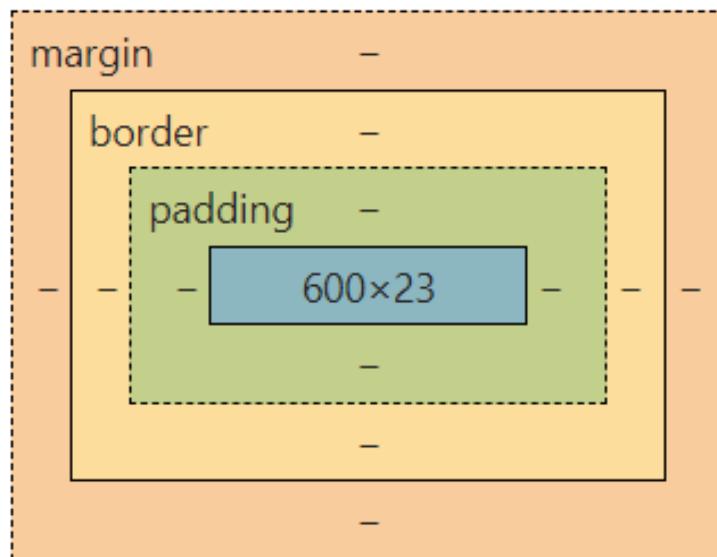
À toi de jouer ! Modifie ton fichier HTML de telle sorte que les trois div soient emballés dans un « wrapper ». Fait en sorte que ce wrapper ait une largeur de 600px. Observes-tu la différence ? Le wrapper a maintenant une largeur fixe : peu importe la taille de la fenêtre, le contenu fera toujours 600px de large (sauf si la fenêtre est trop petite).



Si l'on souhaite centrer cet élément par rapport à la fenêtre, on peut utiliser les propriétés `margin-left` et `margin-right` en leur mettant la valeur `auto`. Ainsi, les marges à gauche et à droite feront exactement la même taille et pousseront l'élément suffisamment vers la gauche et la droite pour le centrer.

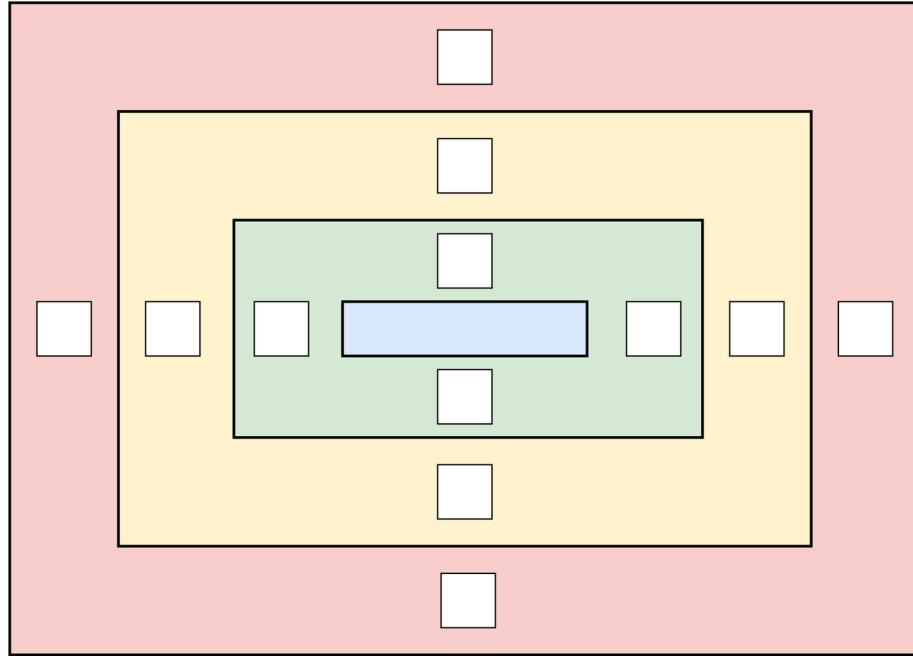


Sur ton navigateur internet, effectue un clic droit dans DIV2 puis choisis l'option « Inspecter l'élément » pour ouvrir **l'inspecteur**. La boîte suivante devrait apparaître :

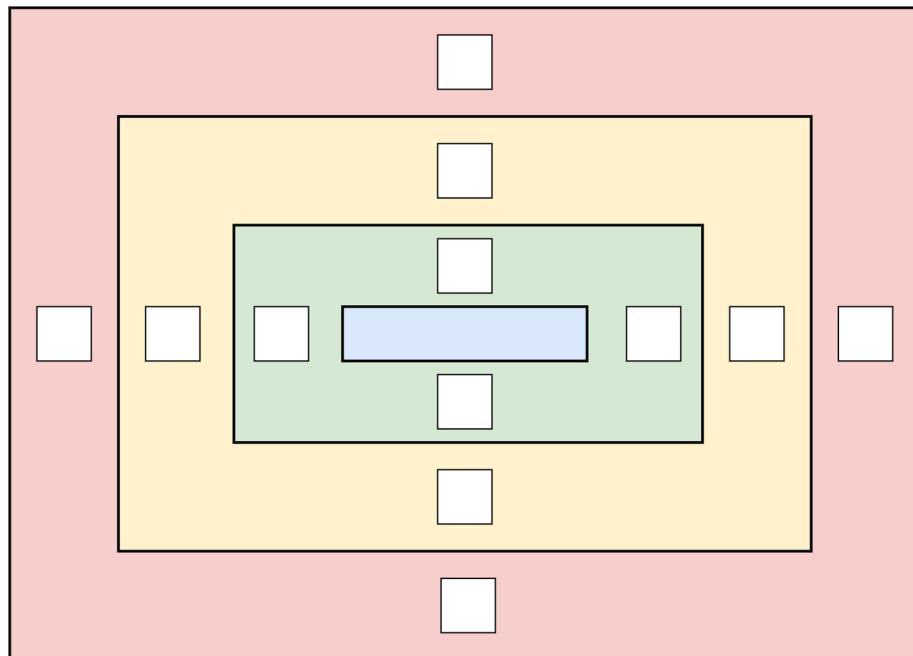


Cette boîte représente le box model d'un élément HTML. On remarque que la largeur du contenu fait bien 600px. La hauteur (23px) peut être différente.

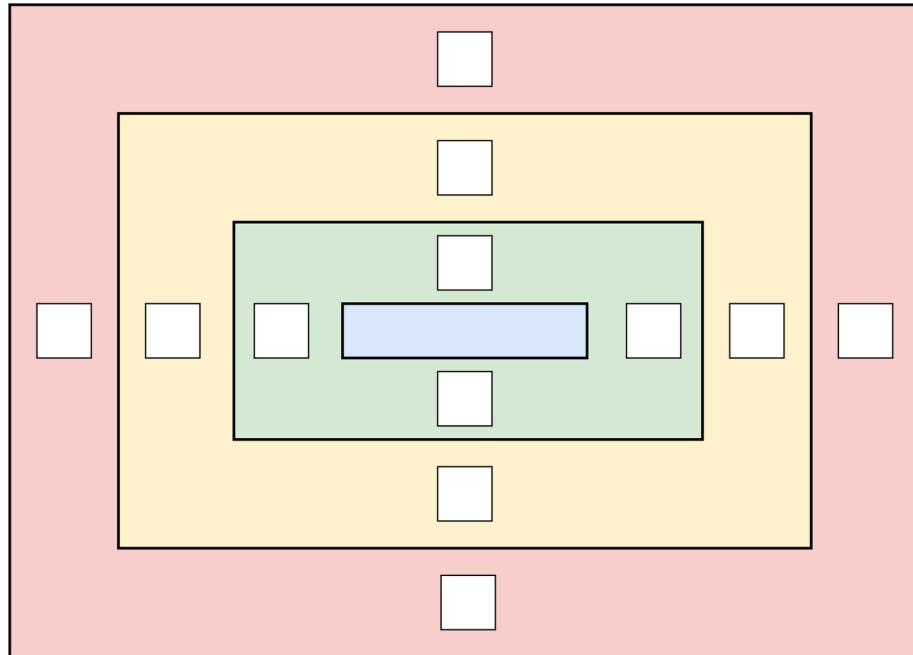
Fait maintenant en sorte que le DIV2 ait un rembourrage de 6px dans les quatre directions. Observe l'effet de cette modification sur le rendu visuel et sur la boîte de l'inspecteur. Que devient la taille du contenu (rectangle bleu) ? Pourquoi ce changement ?



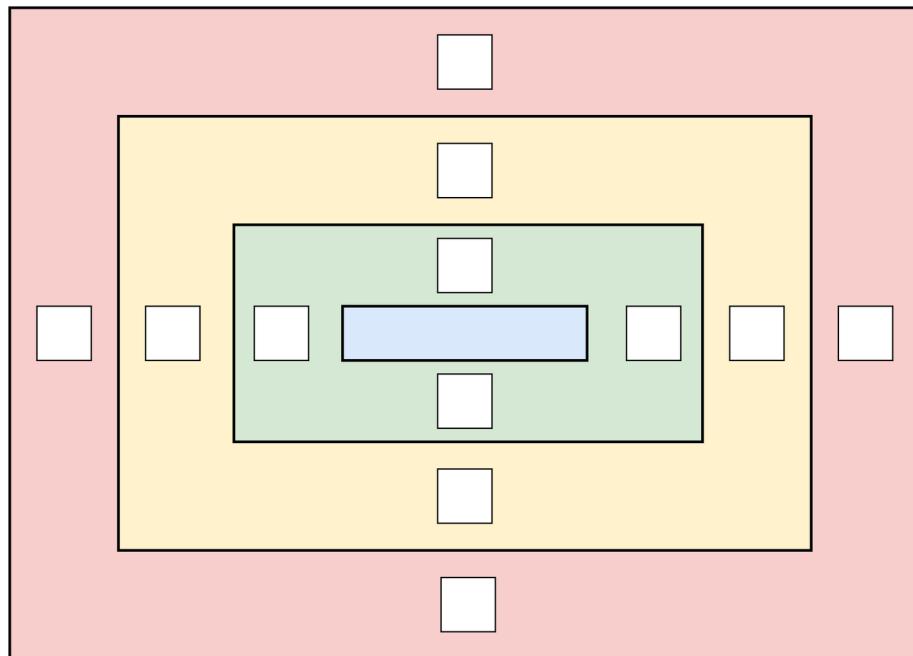
Ajoute maintenant une bordure en pointillés autour du DIV2 avec une épaisseur de 4px. Que devient la taille du contenu ?



Enfin, ajoute une marge de 12px dans les quatre directions autour du DIV2. Que devient la taille du contenu ?

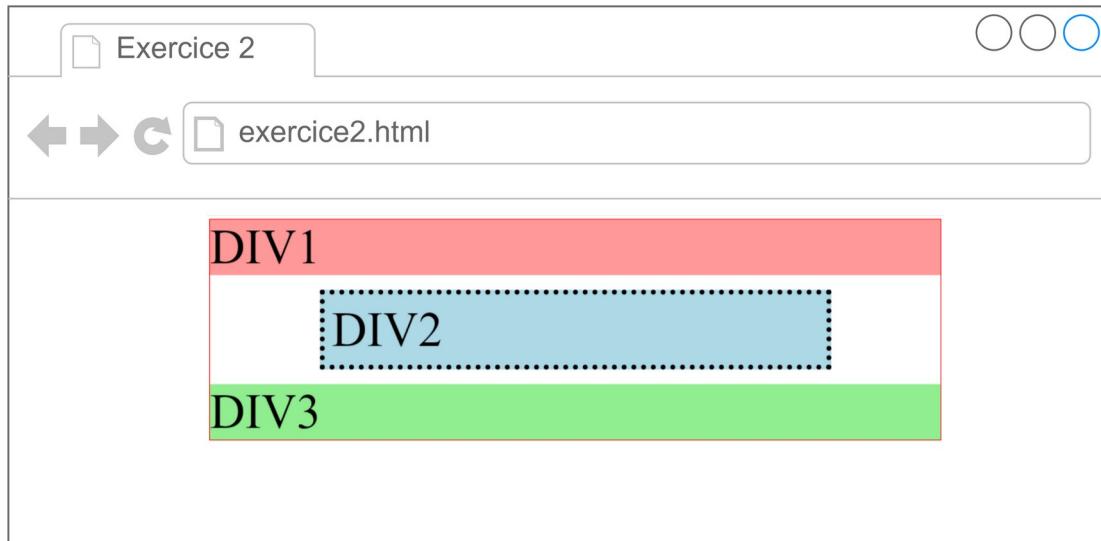


Tu vas maintenant travailler sur l'emballeur (« wrapper »). Pour rendre ses dimensions plus visibles, ajoute-lui une bordure en trait plein d'1px. Ajoute lui également une marge inférieure et supérieure de 12px. Que deviennent ses dimensions ?

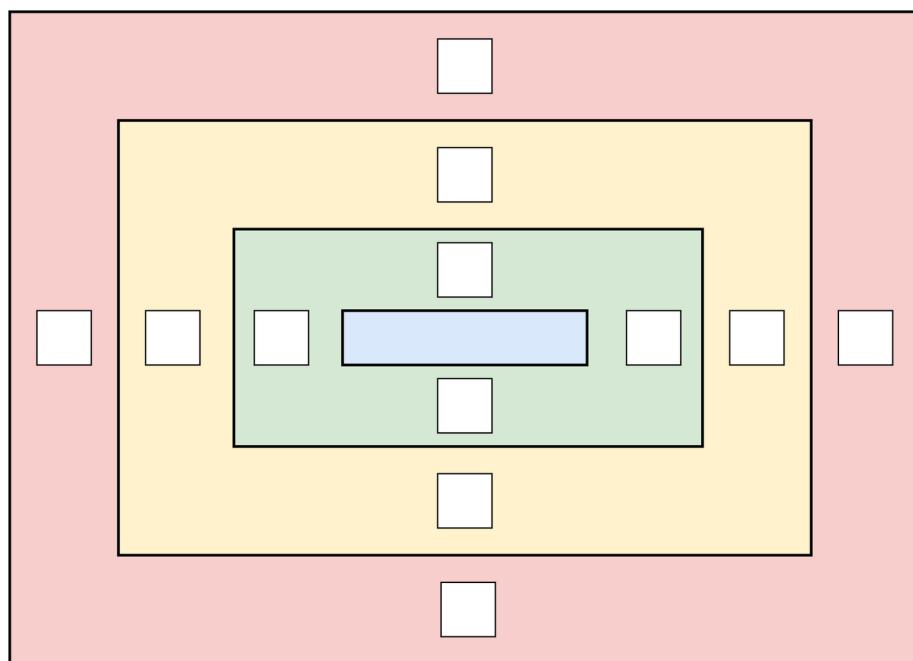


Rappel : le width et le height ne correspondent pas à l'espace total !

Fais maintenant en sorte d'obtenir le résultat suivant :

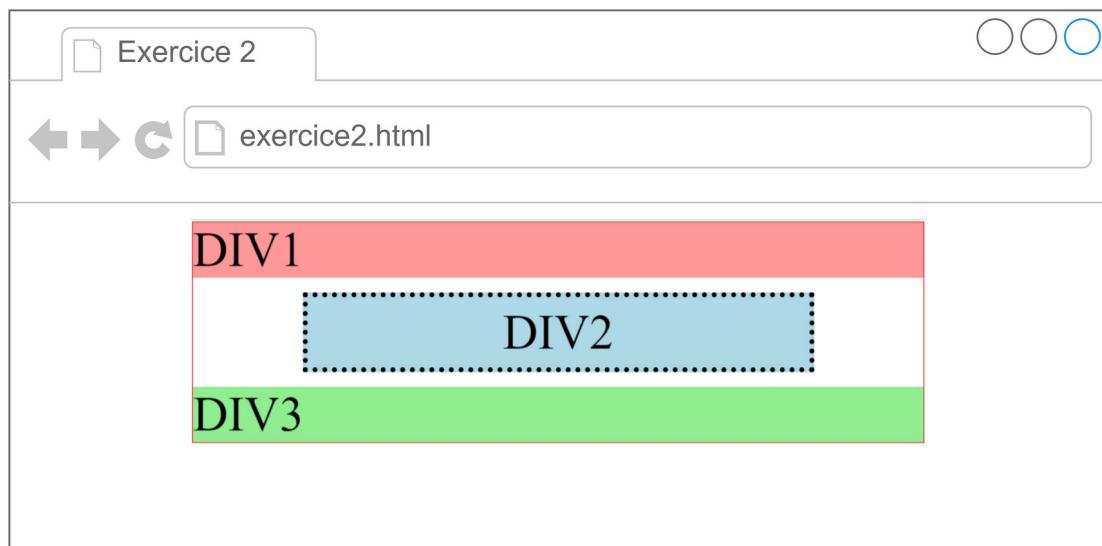


Sans utiliser l'inspecteur, complète le box model du DIV2. Quelles seront ses marges à gauche et à droite ?



Si tu as répondu 100px à gauche et à droite pour les marges, ce n'est pas correct...

Enfin, fait en sorte d'obtenir le résultat suivant. Quelle propriété dois-tu utiliser pour center le texte « DIV2 » ?

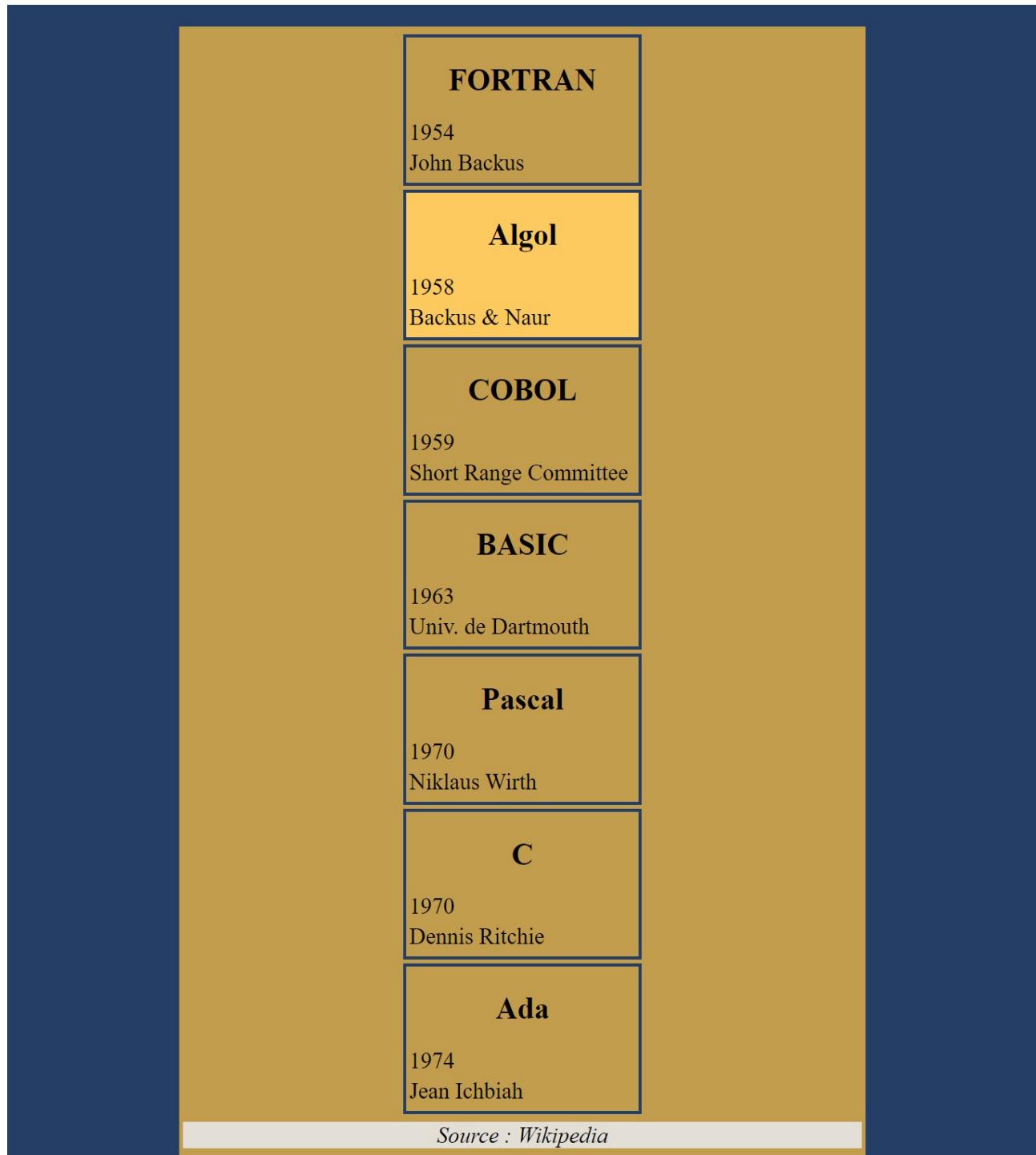


Pour centrer horizontalement...

| Un élément en affichage « block » | Un élément en affichage « inline » |
|---------------------------------------------------------------|---------------------------------------------------------------------------|
| Il faut cibler l'élément que l'on souhaite centrer | Il faut cibler le parent de l'élément que l'on souhaite centrer |
| Il faut donner la valeur auto aux marges à gauche et à droite | Il faut utiliser donner la valeur center à la propriété text-align |

Exercice : langages de programmation

Crée un document HTML ainsi qu'un fichier CSS (n'oublie pas de faire le lien entre les deux). Fais en sorte d'obtenir le résultat suivant :



Lorsque la souris passe au dessus d'un langage, la couleur de fond de la case qui le représente devient plus claire.

Exercice : langages de programmation (version 2)

Demande à ton professeur de te fournir la correction de l'exercice précédent pour partir sur les mêmes bases.

Cet exercice consiste à améliorer la présentation du document de l'exercice précédent. Cette amélioration consiste à afficher trois langages de programmations sur une même ligne (les uns à la suite des autres).

| | | |
|--------------------------------------------|----------------------------------------|-----------------------------------------------|
| FORTRAN 1954 John Backus | Algol 1958 Backus & Naur | COBOL 1959 Short Range Committee |
| BASIC 1963 Univ. de Dartmouth | Pascal 1970 Niklaus Wirth | C 1970 Dennis Ritchie |
| Ada 1974 Jean Ichbiah | | |
| Source : Wikipedia | | |

Dans un premier temps, réponds aux questions suivantes :

1. Quel est le mode d'affichage par défaut des div (ceux qui encadrent chaque langage) ?
2. Quel est le mode d'affichage nécessaire pour faire en sorte que les éléments s'affichent les uns à la suite des autres ?
3. Pour quel mode d'affichage les propriétés width et height sont-elles disponibles ?
4. Dans ton fichier CSS, utilises-tu les propriétés width ou height pour définir la taille des div (ceux qui encadrent chaque langage) ?
5. En répondant à ces quatre questions, tu devrais te rendre compte d'un problème. Lequel ?

Lorsque l'on souhaite afficher les éléments en mode « inline » (c'est-à-dire les uns à la suite des autres) et pouvoir bénéficier des avantages du mode « block » (notamment utiliser les propriétés `height` et `width`), il faut utiliser un mode d'affichage un peu particulier : le mode « **inline-block** ».

Dans ce mode d'affichage, les éléments s'affichent les uns à la suite des autres (comme avec le mode « inline ») mais il est possible d'utiliser les propriétés `width` et `height` (comme avec le mode « block »).

À toi de jouer ! Fais en sorte que les div qui encadrent chaque langage s'affichent en mode « inline-block » (te souviens-tu de la propriété à utiliser ?) et observe le résultat. Les langages devraient s'afficher les uns à la suite des autres. Si les langages sont collés, c'est probablement parce que tu as oublié d'enlever les propriétés `margin-left: auto` et `margin-right: auto`.

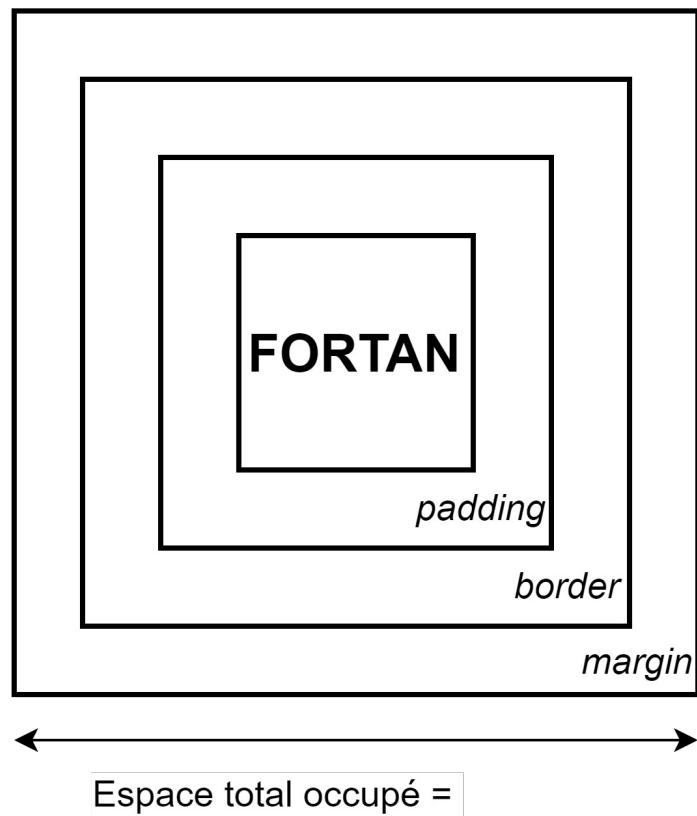
À ce stade de l'exercice, tu devrais obtenir le résultat suivant :

| | | |
|-----------------------------------------------|--------------------------------------------|--|
| FORTRAN 1954 John Backus | Algol 1958 Backus & Naur | |
| COBOL 1959 Short Range Committee | BASIC 1963 Univ. de Dartmouth | |
| Pascal 1970 Niklaus Wirth | C 1970 Dennis Ritchie | |
| Ada 1974 Jean Ichbiah | Source : Wikipedia | |

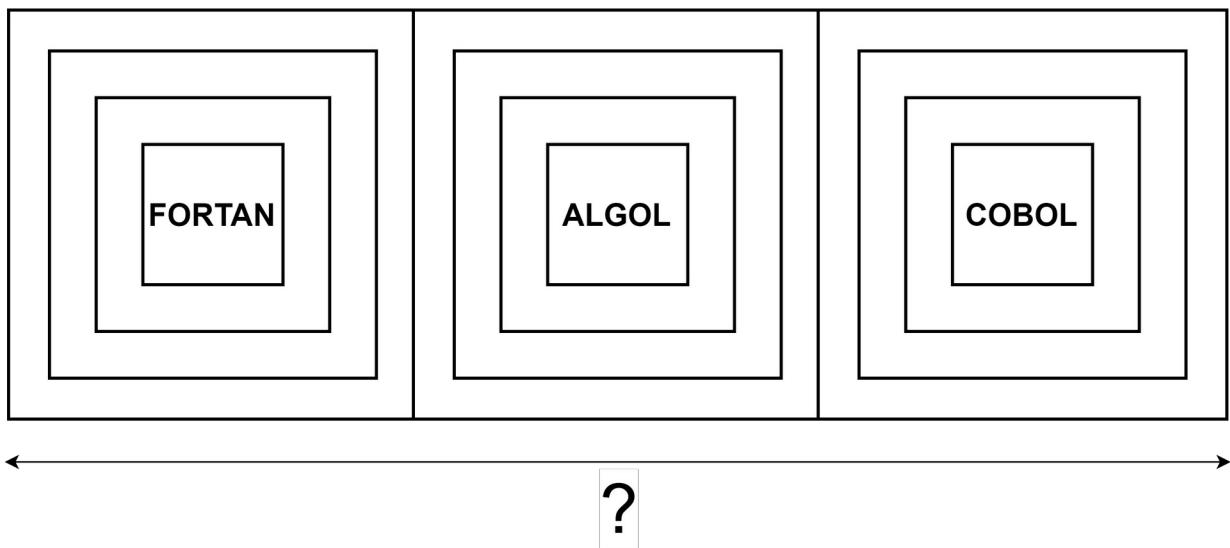
Maintenant, nous souhaiterions afficher trois langages sur une même ligne et pas deux...

Il est temps de faire un peu de math ! Quelle est la largeur exacte que doit faire l'emballageur (wrapper) pour pouvoir accueillir trois langages sur une même ligne ?

Pour répondre à cette question, complète dans un premier temps le box model d'un div représentant un langage :



Maintenant que tu connais l'espace total occupé par un langage, tu devrais être capable de calculer l'espace total occupé par trois langages sur la largeur.



Dans ton fichier CSS, ajuste la largeur de l'emballeur (wrapper) pour faire en sorte qu'il y ait trois langages par ligne. Attention, cette largeur doit être la plus optimale possible : il ne doit pas y avoir d'espace en trop !

Tu devrais obtenir un résultat assez étrange... La largeur que tu as calculée te semble correcte et pourtant il y a toujours deux langages sur une même ligne et non trois.

Pour comprendre le problème, rends-toi sur l'inspecteur HTML. Tu devrais observer ceci :

```
▼ <div id="wrapper">
  ▶ <div class="langage"> ...
    espaces
  ▶ <div class="langage"> ...
  ▶ <div class="langage"> ...
    espaces
  ▶ <div class="langage"> ...
  ▶ <div class="langage"> ...
    espaces
```

Des espaces (retours à la ligne) se sont glissés entre chaque langage. Pourtant, dans ton code HTML, il n'y a aucune balise `
` !

La raison est simple : en mode « inline » et « inline-block », les retours à la ligne présent dans le code HTML sont ne sont pas ignorés (comme avec le mode « block ») mais affichés à l'écran comme des espaces (de 4px de large environ) ! C'est ce qui rend ton calcul incorrect.

```
<div class="langage">
    <h1>FORTRAN</h1>
    <p>1954</p>
    <p>John Backus</p>
</div> [----- RETOUR À LA LIGNE -----]
<div class="langage">
    <h1>Algol</h1>
    <p>1958</p>
    <p>Backus & Naur</p>
</div> [----- RETOUR À LA LIGNE -----]
<div class="langage">
    <h1>COBOL</h1>
    <p>1959</p>
    <p>Short Range Committee</p>
</div> [----- RETOUR À LA LIGNE -----]
```

Le seul moyen de se débarrasser de ces espaces inutiles, c'est de **supprimer ces retours à la ligne dans ton code...**

Tu devrais obtenir le résultat suivant une fois ton code mis à jour :

| | | |
|--------------------------------------------|----------------------------------------|-----------------------------------------------|
| FORTRAN 1954 John Backus | Algol 1958 Backus & Naur | COBOL 1959 Short Range Committee |
| BASIC 1963 Univ. de Dartmouth | Pascal 1970 Niklaus Wirth | C 1970 Dennis Ritchie |
| Ada 1974 Jean Ichbiah | Source : Wikipedia | |

Voici un petit récapitulatif des modes d'affichages vus jusqu'à présent :

| Affichage « block » | Affichage « inline » | Affichage « inline-block » |
|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| Empilés les uns sur les autres | Affichés à la suite les uns des autres | |
| Précédés d'un passage à la ligne | Retour à la ligne si plus assez de place | |
| Largeur = toute la place disponible | Largeur = dépend du contenu | |
| Propriétés width et height disponibles | Propriétés width et height non disponibles | Propriétés width et height disponibles |
| Les retours à la ligne dans le code HTML entre deux « block » sont ignorés | Les retours à la ligne dans le code HTML entre deux « inline » ou « inline-block » s'affichent comme des espaces | |
| Occupent un espace rectangulaire | | |

