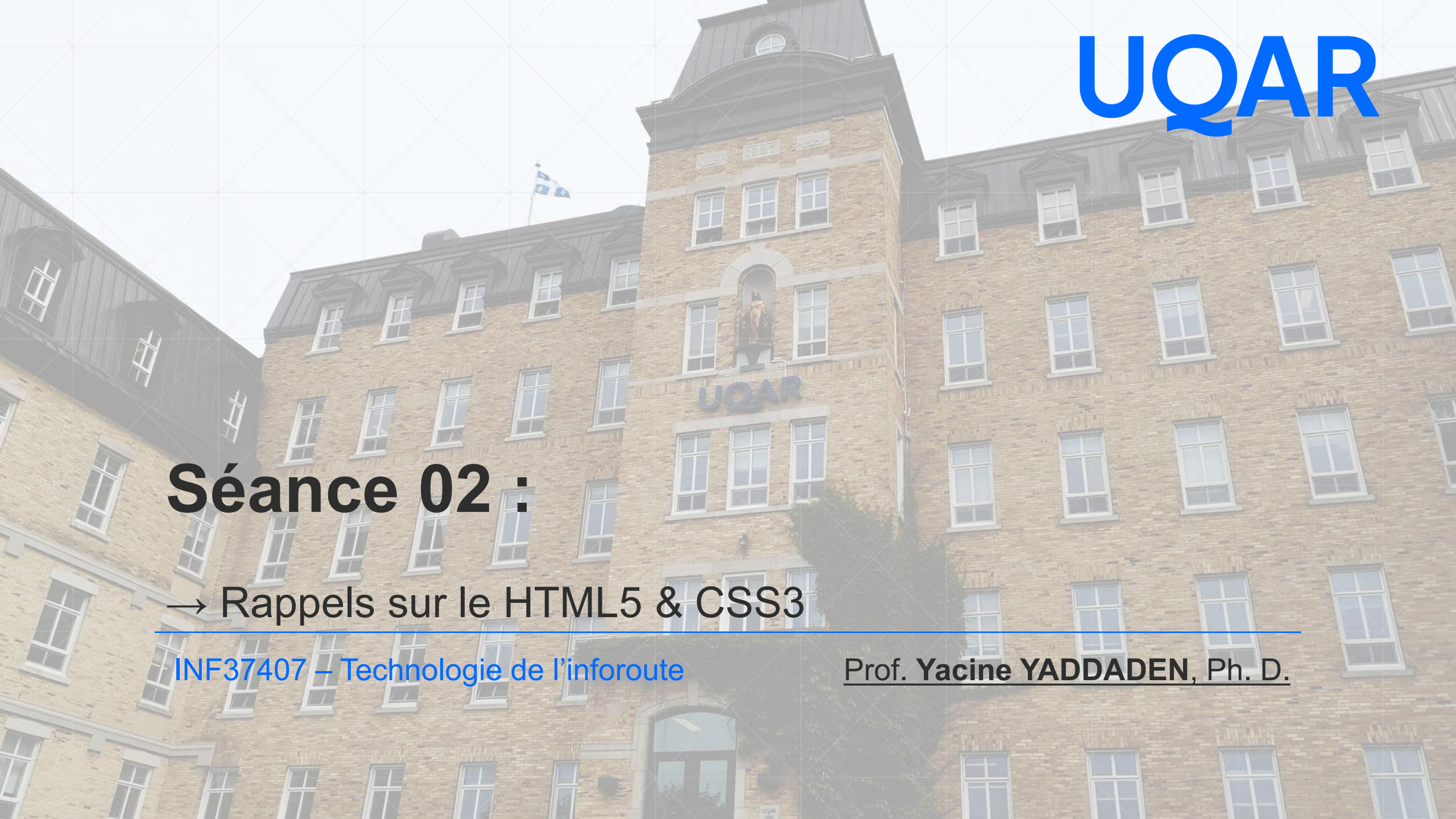


Séance 02 :

→ Rappels sur le HTML5 & CSS3

INF37407 – Technologie de l'inforoute

Prof. **Yacine YADDADEN**, Ph. D.

HTML – Les fondamentaux

UQAR

Plan

1. Introduction et Contexte
2. Qu’est-ce que le HTML ?
3. Représentation minimale
4. Les différents éléments
   a. Le texte
   b. Les listes
   c. Les liens
   d. Les images
   e. Les tableaux
   f. Balises supplémentaires
5. Questions et discussion

UQAR

Introduction et Contexte

- Le moyen utilisé pour la création de *contenu* à partager sur *internet* est via :
  - Les **sites Web** composés de **pages Web**.
- Une pages Web basique est basée sur l'utilisation de :
  - **HTML** : *structure et contenu*.
  - CSS : *mise en forme et esthétique*.
  - JavaScript : *interactivité et programmation*.
- Alors que le CSS et le JavaScript peuvent être *optionnels*, le **HTML** est *nécessaire* et *obligatoire*.

HTML5, quézaco ?

- **Définition :** *« HTML ou HyperText Markup Language est langage de balisage ayant pour principal objectif la représentation et la description des pages Web. »*,

⦸ Ce n'est pas un langage de programmation !

- Il est nécessaire de disposer d'une application ou programme afin de l'interpréter : **Navigateur Web**,
- **HTML5** est la dernière révision ou version majeure du HTML (28 octobre 2014).

Représentation minimale

- La page Web représentée dans un fichier HTML contient différentes **balises**,
- Chacune représente un élément en particulier,
- Elles s'*imbriquent* suivant une structure *arborescente*,
- Le nom des différentes balises est insensible à la casse.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- Cette balise permet de définir le type du document,
- Dans ce cas, c'est une page Web ou HTML,
- Cette représentation est propre au **HTML5**, dans les versions antérieures, ça s'écrit autrement.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- Elle permet d'encadrer et de délimiter le code HTML,
- Elle va permettre d'entourer tout le reste des balises qui vont être ajouter au fur et à mesure,
- Il est possible comme pour toute autres balises d'ajouter des *attributs* : `lang="fr"` permet de définir la langue qui est utilisée dans le document.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- Elle représente l'entête du document et elle contient des informations de configuration :
  - Des métadonnées relatives au document ou page Web,
  - Des liens vers d'autres fichiers requis pour un bon affichage de la page Web.
- Elle ne contient pas de composants visuels qui va s'afficher sur la page Web.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- C'est une **balise** de type *métadonnée* qui a son important, car elle définit le type d'encodage utilisé par le document,
- Généralement, on utilise l'encodage **`UTF-8`** car c'est le plus répandu,
- Il est important de spécifier un encodage qui est compatible avec celui du navigateur afin de de ne pas avoir un mauvais affichage.

```html
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- Elle représente et contient le titre de la page Web,
- Cependant, ce n'est pas affiché à l'intérieur de la page Web,
- Le titre *« Représentation minimale »* est dans le titre de la fenêtre du navigateur Web ou bien sur l'onglet actif.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

Représentation minimale – explication

- Cette **balise** englobe le *contenu* de la page Web,
- À l'intérieur, il est possible d'utiliser d'autres balises *imbriquées* selon le besoin,
- Dans notre cas, nous n'avons pas utilisé de balises. Néanmoins, on aura *« Le corps du document »* affiché sur le navigateur Web.

```
<!DOCTYPE html>

<html lang="fr" >

    <head>
        <meta charset="utf-8" />
        <title>Représentation minimale</title>
    </head>

    <body>
        Le corps du document
    </body>
</html>
```

UQAR

Attributs & Valeurs

- Il est possible d'associer certaines caractéristiques aux différentes balises à travers l'utilisation d'*attributs*,
- Ces *attributs* sont à définir à l'intérieur des balises :

```
<meta charset="utf-8" />
```

Attribut *Valeur*

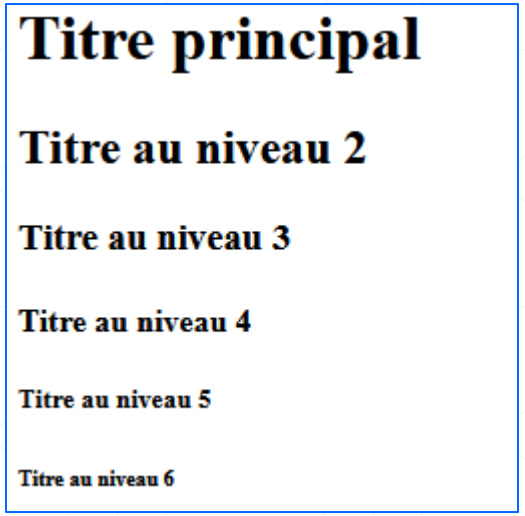
- Il est possible que certains *attributs* ne nécessitent pas une *valeur* :

```
<button disabled></button>
```

Les différents éléments – **Le texte**

- Parmi les *balises* liées au texte, on retrouve celles dédiées aux **titres,**
- Elles sont sur plusieurs niveaux,
- De `<h1>` (*haute*) à `<h6>` (*moindre*) selon l'importance.

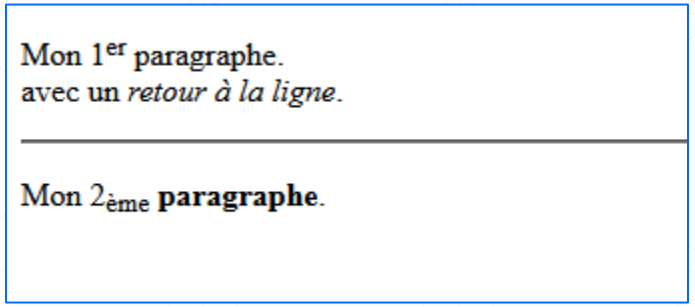
```
<h1>Titre principal</h1>
<h2>Titre au niveau 2</h2>
<h3>Titre au niveau 3</h3>
<h4>Titre au niveau 4</h4>
<h5>Titre au niveau 5</h5>
<h6>Titre au niveau 6</h6>
```



Les différents éléments – **Le texte**

- Afin d'insérer un paragraphe, on utilise la balise `<p>`,
- Il est également possible de mettre du texte en :
  - En italique avec `<i>`,
  - En gras avec `<b>`,
  - En exposant avec `<sup>`,
  - En indice avec `<sub>`.
- Pour un retour à la ligne, on utilise `<br/>` et pour tracer une ligne `<hr/>`.

```
<p>
  Mon 1<sup>er</sup> paragraphe. <br />
  avec un <i>retour à la ligne</i>.
</p>
<hr />
<p>Mon 2<sub>ème</sub> <b>paragraphe</b>.</p>
```



UQAR

Les différents éléments – **Le texte**

- Les balises sémantiques ne sont pas destiné à définir la structure de la pages Web, mais permettent d'*ajouter de l'information supplémentaire*.
- Parmi elles :
  - `<strong></strong>`
    - Donner de l'*importance* à un ou des éléments du texte.
  - `<em></em>`
    - Que le focus sur un certain élément change subtilement.
  - `<blockquote></blockquote>`
    - Pour une *citation* qui prend tout un paragraphe.
  - `<q></q>`
    - Pour une petite et *courte citation*.
  - `<abbr title=""></abbr>`
    - Utilisé pour les *abréviations*.

UQAR

Les différents éléments – **Le texte**

- Et d'autres :
  - `<cite></cite>`
    - Quand on fait *référence* à un livre, film, …
  - `<dfn></dfn>`
    - Quand on veut donner la *définition* d'un certain terme.
  - `<address></address>`
    - Contient les *informations de contact* ou les coordonnées.
  - `<ins></ins>`[1] et `<del></del>`[2]
    - Quand un nouvel élément est inséré[1] ou enlevé[2] du document.
  - `<s></s>`
    - Quand un certain élément est *enlevé* ou *n'est plus considéré*.

Les différents éléments – **Les listes**

- Il y a deux types de listes :
  - **Ordonnées** avec `<ol></ol>`,
  - **Non-ordonnées** avec `<ul></ul>`.
- Pour chaque *élément* `<li></li>`.
- Les listes peuvent *s'imbriquer*.
- Il y a aussi une autre sorte de liste qui consiste en les **définitions** :
  - `<dl></dl>` : le *groupe* de définitions,
  - `<dt></dt>` : le *terme* à définir.
  - `<dd></dd>` : la *définition*.

```
<ol>
  <li>Faire les devoirs</li>
  <li>
    Faire les courses :
    <ul>
      <li>Avocats</li>
      <li>Oranges</li>
    </ul>
  </li>
  <li>Aller à l'université</li>
</ol>
```

1. Faire les devoirs
2. Faire les courses :
   - Avocats
   - Oranges
3. Aller à l'université

UQAR

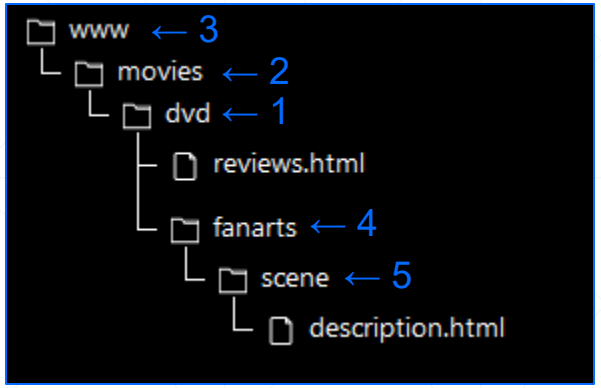
Les différents éléments – **Les liens**

- Afin d'utiliser les liens on fait appel à la balise :

```
<a href="https://www.uqar.ca">UQAR</a>
```

URL *Affiché*

- Chaque page ou image dans un site Web possède un **URL** (*Uniform Resource Locator*) composé du **nom de domaine** + le **chemin** vers la page ou l'image.
- On fait souvent appel aux lien relatifs, il y a cinq cas possibles :
  1. Dans `href="reviews.html"`,
  2. Dans `href="dvd/reviews.html"`,
  3. Dans `href="movies/dvd/reviews.html"`,
  4. Dans `href="../reviews.html"`,
  5. Dans `href="../../reviews.html"`,



UQAR

Les différents éléments – **Les liens**

- Si on veut faire en sorte d'utiliser un lien pour une adresse e-mail, on doit précéder le lien avec `mailto:` :

```
<a href="mailto:yacine_yaddaden@uqar.ca">Mon e-mail</a>
```

- Si on veut ouvrir le lien dans un nouvel onglet, on utilise `target="_blank"` :

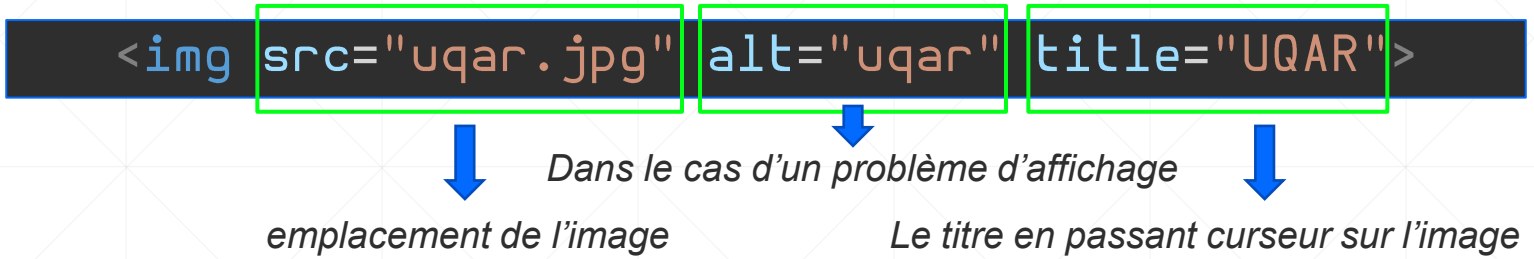
```
<a href="https://www.uqar.ca" target="_blank">UQAR</a>
```

- Si on veut naviguer dans la même page, on utilise une combinaison de :
  - Identification de l'emplacement : `<h1 id="top">Titre en haut !</h1>`
  - Utilisation du lien : `<a href="#top">Lien vers le haut</a>`

UQAR

Les différents éléments – **Les images**

- On pourrait classer les images pour le Web en trois principales catégories :
  - **Images compressées :** JPEG ou PNG (avec transparence),
  - **Images animées :** GIF (plusieurs calques superposés),
  - **Images vectorielles :** SVG (pas de perte de qualité).
- Afin d'insérer une image, on utilise les éléments suivants :



UQAR

Les différents éléments – **Les images**

- Il est possible de changer la taille de l'image afficher avec :
  - `height=""` (*hauteur*) et `width=""` (*largeur*) en **pixel**.
- On peut également entourer l'image avec la balise `<figure></figure>` et y ajouter une légende avec la balise `<figcaption></figcaption>`.

UQAR

Les différents éléments – **Les tableaux**

- Afin de créer un tableau, vous avez besoin des éléments suivants :
  - `<table></table>` elle contient l'ensemble des éléments du tableau.
  - `<tr></tr>` elle représente une *ligne* du tableau.
  - `<td></td>` elle représente une *cellule* contenue dans une ligne du tableau.
- Dans le cas où l'on a un long tableau, on peut délimiter trois parties distinctes :
  - L'**en-tête** du tableau : `<thead></thead>`
  - Le **corps** du tableau : `<tbody></tbody>`
  - Le **pied** du tableau : `<tfoot></tfoot>`

UQAR

Les différents éléments – **Les tableaux**

- Afin de fusionner les cellules d'un tableau, on utilise les *attributs* :
  - `colspan=""` en indiquant comme valeur le nombre de *colonnes* à fusionner,
  - `rowspan=""` en indiquant comme valeur le nombre de *lignes* à fusionner,
- Si on veut définir l'épaisseur de la bordure du tableau, on utilise l'attribut :
  - `border=""` avec l'épaisseur en *pixel* comme valeur.
- Afin de définir la couleur du fond d'une cellule, on utilise l'attribut :
  - `bgcolor="#cccccc"` avec la couleur en code *hexadécimal* (#).

UQAR

Les différents éléments – **Balises supplémentaires**

- Il est possible de commenter du code **HTML** en utilisant :
  - `<!-- commentaire -->` ce qui est à l'intérieur est ignoré par le navigateur.
- Afin d'identifier de façon unique un élément de la page Web :
  - On utilise l'*attribut* `id=""` avec une valeur unique.
- Afin de pointer vers plusieurs éléments de la page Web :
  - On utilise l'*attribut* `class=""` avec une valeur réutilisable.
- Il y a deux catégories de balise :
  1. Balise **inline** : *ne nécessitant pas un retour à la ligne (exemple : `a`, `img`, `em`, …).*
  2. Balise **block** : *nécessitant un retour à la ligne (exemple : `h1`, `p`, `ul`, …).*

Les différents éléments – **Balises supplémentaires**

- Afin de regrouper des éléments en **block**, on utilise la balise `<div></div>`,
- Afin de regrouper des éléments en **inline**, on utilise la balise `<span></span>`,
- La balise `<iframe src="" ></iframe>` permet de réserver un espace dans la page Web afin que le contenu d'une autre page y soit affiché. (*exemple : Google Map*).
- Les balises `<meta http-equiv="" />` sont mises au niveau du `<head></head>` et permettent d'ajouter toute sortes d'information à la page Web :
  - **`author`** : *l'auteur de la page Web,*
  - **`description`** : *une description du contenu,*
  - **`keywords`** : *liste de mots clé séparés par des virgules,*
  - **`robots`** : *définit si les moteurs de recherche doivent ajouter la page aux résultats.*

Plus de détails sur les métadonnées
Appuyer ci-dessus

Questions & Discussion

HTML - Les formulaires

UQAR

Plan

1. Introduction et Contexte
2. Fonctionnement et structure de base d’un formulaire
3. Les différents types de champs :
   a. Champ texte
   b. Champ pour choix
   c. Champ pour téléverser un fichier
   d. Soumettre un formulaire
4. Les boutons et les champs cachés
5. Étiqueter un formulaire et groupe d’éléments
6. Validation des formulaires
7. Questions et discussion

UQAR

Introduction et Contexte

- Les formulaires permettent à l'utilisateur d'**interagir** avec la page Web,
- Ils lui permettent aussi d'**envoyer** des *informations vers le serveur*,
- Il y a différents *types de champs* qui peuvent être utilisé :
  - Le simple champs texte,
  - La liste déroulante,
  - Les cases à cocher,
  - …
- Savoir créer des formulaires en HTML est **primordial** quand on fait du développement Web, exemple : *authentification*, *inscription*, *recherche*, …

UQAR

Comment fonctionne un formulaire

- Le fonctionnement d'un formulaire peut être décrit comme suit :
  1. L'utilisateur remplit les différents *champs* du formulaire :
     - Chaque champ possède un *unique identifiant*.
  2. Les données du formulaire sont envoyées vers le *serveur* :
     - Une **URL** des destinations du serveur est utilisée avec la *méthode* **POST**.
  3. Au niveau du serveur, les données sont *traitées* avec : **PHP**, Python, C#, …
     - Ça inclut, éventuellement, le *stockage* sur une *base de données*.
  4. Une page **HTML** est générée du serveur est *renvoyé* vers l'utilisateur.
- Un exemple de structure d'*identifiant* pour un champs :

identifiant *valeur*

```
Username = Alice
```

UQAR

Structure de base d'un formulaire

- Un formulaire vide est défini comme suit :

```
<form action="/register.php" method="post">
  <!-- contenu du formulaire -->
</form>
```

- La balise utilisée est `<form></form>` et elle entoure les autres champs,
- Il contient les attributs suivants :
  - **`action` :** *contient l'URL de la page sur le serveur permettant de faire le traitement sur les données du formulaire, une fois que ce dernier est soumis.*
  - **`method` :** *Il y a deux possibilités de valeur :*
    1. La méthode **GET** est rapide, mais non sécurisée (*champs de recherche*),
    2. La méthode **POST** est moins rapide, mais plus sécurisée (*plusieurs champs avec téléversement*).

UQAR

Les différents types de champs
Champ texte – Texte simple

- C'est l'un des champs les *plus utilisé*, permettant à l'utilisateur d'entrer une *courte chaîne de caractères*,
- La forme d'un champ texte (*à l'intérieur d'un formulaire*) est défini comme suit :

```
<input type="text" name="username" size="20" maxlength="30" />
```

- On utilise la balise `<input>`, elle contient les attributs suivants :
  - **`type` :** *c'est le type de champs,* ***`text`*** *fait référence à un champs texte sur une seule ligne,*
  - **`name` :** *contient la valeur de l'identifiant unique pour ce champ,*
  - **`size` :** *définit la taille du champ en nombre de caractères,*
  - **`maxlength` :** *définit la taille maximale du texte (en caractères) que peut contenir le champ.*

UQAR

Les différents types de champs
Champ texte – Mot de passe

- C’est une variante du champ de texte simple, la différence réside dans le fait de cacher les caractères en les remplaçant par un *caractère spécial* : « • »,
- Il est utilisé, comme son nom l’indique, pour les mots de passes,
- Sa forme dans un formulaire est la suivante :

```
<input type="password" name="password" size="20" maxlength="30" />
```

- Il contient les *mêmes attributs* qu’un champ texte simple.

UQAR

Les différents types de champs
Champ texte – Paragraphe

- Le dernier type de champ texte permet de recevoir un nombre de caractère plus élevé, il est utilisé pour les *long paragraphe* ou les *commentaires*,
- Sa forme dans un formulaire est la suivante :

```
<textarea name="comment" cols="30" rows="10"></textarea>
```

- Il contient deux attributs qui permettent de définir sa taille :
  - **`cols` :** *sur combien de colonnes s'étendra le texte,*
  - **`rows` :** *sur combien de lignes s'étendra le texte.*

UQAR

Les différents types de champs
Champ pour choix – Bouton radio

- Le *bouton radio* est une des façons de définir plusieurs choix à l'utilisateur,
- Sa spécificité réside dans le fait que l'utilisateur ne peut cocher qu'un bouton radio à la fois, le fait d'appuyer sur un autre décoche le précédent,
- Sa forme dans un formulaire est la suivante :

```
<input type="radio" name="fruits" value="orange" /> Orange
<input type="radio" name="fruits" value="apple" checked /> Apple
<input type="radio" name="fruits" value="banana" /> Banana
```

- Il contient les attributs suivants :
  - **`value` :** *contient la valeur associée au bouton radio (récupérée côté serveur),*
  - **`checked` :** *définit quel bouton radio est coché initialement (un seul à la fois).*

UQAR

Les différents types de champs
Champ pour choix – Cases à cocher

- Les cases à cocher sont une autre façon de définir plusieurs choix à l'utilisateur,
- La différence avec le précédent réside dans le fait qu'on peut *cocher plusieurs cases à la fois*, c'est-à-dire faire des *choix multiples*,
- Sa forme dans un formulaire est la suivante :

```
<input type="checkbox" name="fruits" value="orange" /> Orange
<input type="checkbox" name="fruits" value="apple" checked /> Apple
<input type="checkbox" name="fruits" value="banana" checked /> Banana
```

- C'est les mêmes attributs que le type de champs précédent à la différence où la valeur de l'attribut **`type`** est **`checkbox`**.

UQAR

Les différents types de champs
Champ pour choix – Liste déroulante

- C’est aussi une autre façon de définir plusieurs choix à l’utilisateurs, il ne peut en sélectionner qu’un seul,
- C’est particulièrement utile lorsque la liste de choix trop longue pour être affichée directement sur la page,
- Sa forme dans un formulaire est la suivante :

```
<select name="fruits">
  <option value="orange">Orange</option>
  <option value="apple" selected>Apple</option>
  <option value="banana">Banana</option>
</select>
```

UQAR

Les différents types de champs
Champ pour choix – Liste déroulante *(suite)*

- La liste déroulante est composée des éléments suivants :
  - `<select></select>` : *c'est la balise englobant les différents éléments de choix,*
  - `<option></option>` : *elle permet de définir un élément de choix en particulier.*
- Elle contient l'attribut suivant en plus de ceux définis précédemment :
  - **`selected`** : *permet de définir quel élément de la liste déroulante est sélectionné par défaut.*

UQAR

Les différents types de champs
Champ pour choix – Liste de choix multiples

- C'est une *variante* de la liste déroulante, la différence réside dans le fait qu'il est possible peut *sélectionner plusieurs éléments à la fois*,
- Sa forme dans un formulaire est la suivante :

```
<select multiple name="fruits" size="2">
  <option value="orange" selected>Orange</option>
  <option value="apple" selected>Apple</option>
  <option value="banana">Banana</option>
</select>
```

- Pour sélectionner plus d'un élément, maintenir la touche **CTRL** appuyée (Windows),
- Il y a les deux attributs suivants :
  - **`multiple` :** *permet d'activer le mode sélectionne multiple,*
  - **`size` :** *permet de définir le nombre d'éléments de la liste affichés à la fois.*

UQAR

Les différents types de champs
Champ pour téléverser un fichier

- C'est un champ spécial permettant à l'utilisateur de *téléverser* ou d'*uploader* un *fichier* sur le *serveur*,
- Il est possible de téléverser toute sorte de fichiers : image, vidéo, PDF, son, …
- Sa forme dans un formulaire est la suivante :

```
<input type="file" name="user_pdf" />
```

- C'est comme pour un champ texte simple, sauf que la valeur de l'attribut **`type`** devient **`file`**,
- Un bouton est ajouté permettant à l'utilisateur d'aller (*parcourir*) chercher l'*emplacement* du fichier sur son ordinateur.

UQAR

Les différents types de champs
Soumettre un formulaire

- Chaque formulaire créé a besoin d'avoir un bouton afin que les champs remplis par l'utilisateur soient envoyés vers le serveur,
- Pour cela, on utilise un bouton spécial :

```
<input type="submit" value="Send" >
```

- En appuyant sur le bouton ainsi créé, les données des champs sont envoyées vers l'*URL* indiqué sur le formulaire en utilisant la *méthode* indiquée (*GET* ou *POST*).
- L'attribut **`value`** permet juste de définir le texte qui est affiché sur le bouton.

UQAR

Les boutons et les champs cachés

- Il est possible de créer des boutons à partir d'image :

```
<input type="image" src="button.jpg" alt="button" />
```

- On peut également créer des boutons auxquels on associera une certaine action :

```
<button> <img src="button.jpg" alt="button"> Add</button>
```

- Pour le champ caché, il suffit de changer la valeur de l'attribut **`type`** :

```
<input type="hidden" name="fruit" value="apple" />
```

UQAR

Étiqueter un formulaire

- Afin de mieux *identifier* les champs des formulaires ainsi que leur *signification*, on utilise un *étiquetage* avec `<label></label>` :

```
<label for="username">Username :</label>
<input type="text" name="username" id="username" />
```

- La liaison entre l'étiquette et le champ correspondant se fait à l'aide des attributs :
  - **`for` :** *au niveau de l'étiquette,*
  - **`id` :** *au niveau du champ en question.*

UQAR

Groupe d'éléments

- Afin d'améliorer l'organisation des champs au niveau du formulaire, il est possible de créer des groupes comme :

```
<fieldset>
  <legend>Connect</legend>
  <label for="username">Username</label>
  <input type="text" name="username" id="username" />
  <label for="password">Password</label>
  <input type="password" name="password" id="password" />
</fieldset>
```

- Il y a deux principales balises qu'on utilise :
  - `<fieldset></fieldset>` : *c'est la balise englobante du groupe de champs,*
  - `<legend></legend>` : *c'est un titre qui est affiché en haut du groupe de champs.*

UQAR

Validation des formulaires

- La *validation* des formulaires coïncide avec l'arrivée du **HTML5**,
- La *validation la plus simple* est de faire en sorte qu'un champ soit **obligatoire** :

```
<input type="text" name="username" required />
```

- Une *validation plus avancée* consiste à *définir le type de valeur* que doit contenir un champ en particulier :
  - Les **dates** : *ça nous donne accès à un calendrier, il faut utiliser le type* ***`date`***,
  - Les **e-mails** : *il faut utiliser le type* ***`email`***,
  - Les **URLs** : *il faut utiliser le type* ***`url`***.
- Dans le cas où la validation ne se passe bien, il y a un *message d'erreur en rouge* qui apparaît à côté du champ en question.

UQAR

Champ de recherche

- C’est un autre type de champ qui a été introduit avec l’arrivée de **HTML5**,
- Il ressemble à un champ texte simple, il est défini comme suit :

```
<input type="search" name="search" placeholder="Enter words ...">
```

- Il y a deux particularités (*ci-dessus*) :
  - Le type du champ prend la valeur **`search`**,
  - **`placeholder`** : *permet définir le texte par défaut affiché sur le champ en question.*

Questions & Discussion

CSS – Les fondamentaux

UQAR

Plan

1. Introduction et Contexte
2. Fondamentaux du CSS
3. Les couleurs
4. Le texte
5. Les boites
6. Les listes et les tableaux
7. Les images
8. Positionnement des éléments
9. Framework CSS
10. Questions et discussion

UQAR

Introduction et Contexte

- Alors que le HTML contient l'information utile (texte, images, …), le CSS permet d'assurer une *meilleure présentation* du ce contenu.
  - Il permet d'appliquer *un ensemble de règles* permettant de *contrôler la manière dont sont présentés* les différents *éléments* d'une page Web.
- Une pages Web basique est basée sur l'utilisation de :
  - HTML : *structure et contenu*.
  - **CSS** : *mise en forme et esthétique*.
  - JavaScript : *interactivité et programmation*.
- Même si, techniquement, le **CSS** est *optionnel* lors de la création d'une page Web, mais il est devenu au fil du temps incontournable.

CSS3, quézaco ?

- **Définition :** *« CSS pour Cascading Style Sheets ou encore feuilles de style en cascade est un langage informatique qui décrit la présentation des page Web.»*,

⦸ Ce n’est pas un langage de programmation !

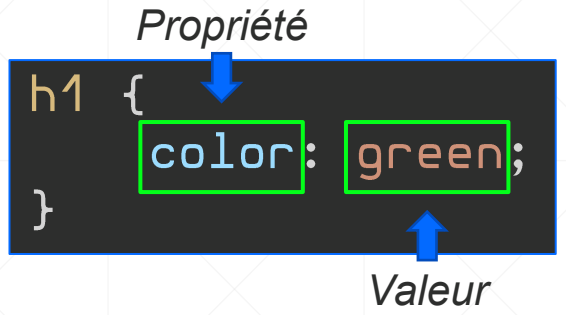
- Il est nécessaire de disposer d’une application ou programme afin de l’interpréter : **Navigateur Web**,
- Un de ces avantage est de pouvoir séparer le contenu de la mise en forme.

UQAR

Structure d'une règle en CSS

- Le CSS permet d'associer une règle de présentation à un élément HTML, il y a deux parties distinctes :
  1. Le **sélecteur :** *indique l'élément ciblé.*
  2. La **déclaration :** *indique comment l'élément va être stylisé.*

- À l'intérieur d'une déclaration, nous avons deux éléments :
  1. La **propriété :** *indique l'aspect de l'élément qui sera affecté par le changement.*
  2. La **valeur :** *indique les paramètres de la propriété.*



UQAR

Comment utiliser le CSS

Il y a deux manières d'appliquer des styles CSS à une page HTML :

1. **En interne :**
   - Pour cela, on fait appel à la balise

```
<style type="text/css">
    /* propriété à appliquer */
</style>
```

2. **En externe :**
   - En utilisant un fichier **`.css`** externe avec :

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

   - **`href`** : *permet de spécifier le chemin vers le fichier de style CSS,*
   - **`type`** : *permet de définir le type de document qui sera importé sur la page HTML,*
   - **`rel`** : *permet de spécifier la relation entre le fichier importé et la page HTML.*

UQAR

Les sélecteurs

- Les **sélecteurs** permettent de *pointer* vers l'élément ou les éléments de la pages HTML ciblés par la modification au niveau de leur présentation ou style,
- Il y a plusieurs *types de sélecteur* :

`* {}` → **Universel :** *Cible toutes les balise de la page HTML,*

`h1, p {}` → **Par type de balise :** *Cible un certain type de balise,*

`.note {}` → **Par classe :** *Cible toutes les balise avec la classe définie,*

`#intro {}` → **Par ID :** *Cible uniquement l'élément avec l'ID défini,*

`li>a {}` → **Par enfant :** *Cible l'ensemble des enfants directs de la balise initiale,*

`p a {}` → **Descendant :** *Cible l'ensemble des éléments contenus à l'intérieur de la balise initiale,*

`h1+p {}` → **Voisin adjacent :** *Cible le premier voisin direct de la balise initiale,*

`h1~p {}` → **L'ensemble des voisins :** *Cible l'ensembles de voisins de la balise initiale.*

UQAR

Les couleurs

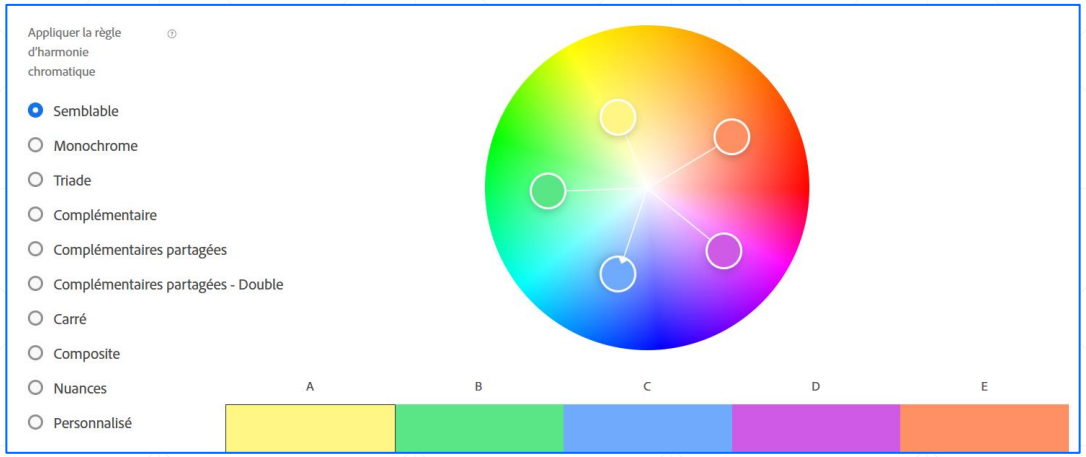
- Lorsqu'on travaille avec les couleurs en CSS, il faut garder à l'esprit qu'il y a :
  1. Couleur du **texte** : `color: ;`
  2. Couleur du **fond** : `background-color: ;`
- Afin de définir la valeur de la couleur à utiliser :
  1. Valeur **RGB** : `rgb(255, 0, 0)`
  2. Valeur *Hexadécimale* (**HEX**) : `#F00`
  3. Nom de la couleur (*en anglais*) : `red`
- Il est également possible de jouer sur la **transparence** avec :

```
rgba(255, 0, 0, 0.5);
```

```
opacity: 0.5;
```

Les couleurs – *Suite*

- **Roue de couleur :**
  - Lien : https://color.adobe.com/fr/create/color-wheel



- **Convertisseur de format de couleur :**
  - Lien : https://www.rapidtables.com/convert/color/rgb-to-hex.html

UQAR

Le texte

- Afin de changer le *type de police* utilisée : `font-family: ;`
  - Parmi les valeurs qu'elle peut prendre : `Arial, Helvetica, sans-serif`
- En ce qui est en lien avec la taille des caractères : `font-size: ;`
- Il y a trois mesures qui peuvent être utilisées :
  1. En **pixels** : *la mesure le plus utilisée par les graphistes et designer* → `12px`
  2. En **pourcentage** : *la taille initiale de la police (100%) est de 16 pixels* → `200%`
  3. En **EMS** : *Elle représente la largeur de la lettre **m*** → `1.3em`

UQAR

Le texte – *Suite*

- Il est également possible de changer le style du texte avec : `font-weight: ;`
  - Mettre en **gras** : `bold`
  - Mettre en **italique** : `italic`
  - La valeur `normal` remet le texte à son style initial.
- En ce qui concerne les transformations *du texte*, il y a : `text-transform: ;`
  - Mettre en **majuscule** : `uppercase`
  - Mettre en **minuscule** : `lowercase`
  - Mettre la **première lettre en majuscule** : `capitalize`

UQAR

Le texte – *Suite*

- Pour tout ce qui décorations du texte : `text-decoration: ;`
  - **Aucune décoration :** `none`
  - **Souligné :** `underline`
  - **Ligne au dessus :** `overline`
  - **Barrer le texte :** `line-through`
- En ce qui concerne l'alignement :
  - **Vertical :** `vertical-align: ;`
    - Elle peut prendre les valeurs suivantes : `top, middle, bottom`
  - **Horizontal :** `text-align: ;`
    - Elle peut prendre les valeurs suivantes : `left, right, center, justify`

UQAR

Le texte – *Suite*

- Afin d'appliquer un *effet d'ombre* au texte, on peut utiliser la propriété suivante :

```
text-shadow: 1px 1px 3px #000;
```

Exemple de paramètres

- Afin d'ajouter une *indentation* avec une certains *valeur en pixel*, on utilise :

```
text-indent: 20px;
```

- Pour avec un minimum d'interaction en utilisant le CSS, on peut utiliser :
  - `:hover` : *lorsque l'utilisateur passe avec le curseur de la souris sur l'élément cible,*
  - `:active` : *lorsqu'un élément du HTML est activé, comme cliquer sur un bouton,*
  - `:focus` : *Quand un élément est ciblé comme dans un champ de formulaire.*
  - Ils sont mis à côté des *sélecteurs*, exemple : `input:focus`

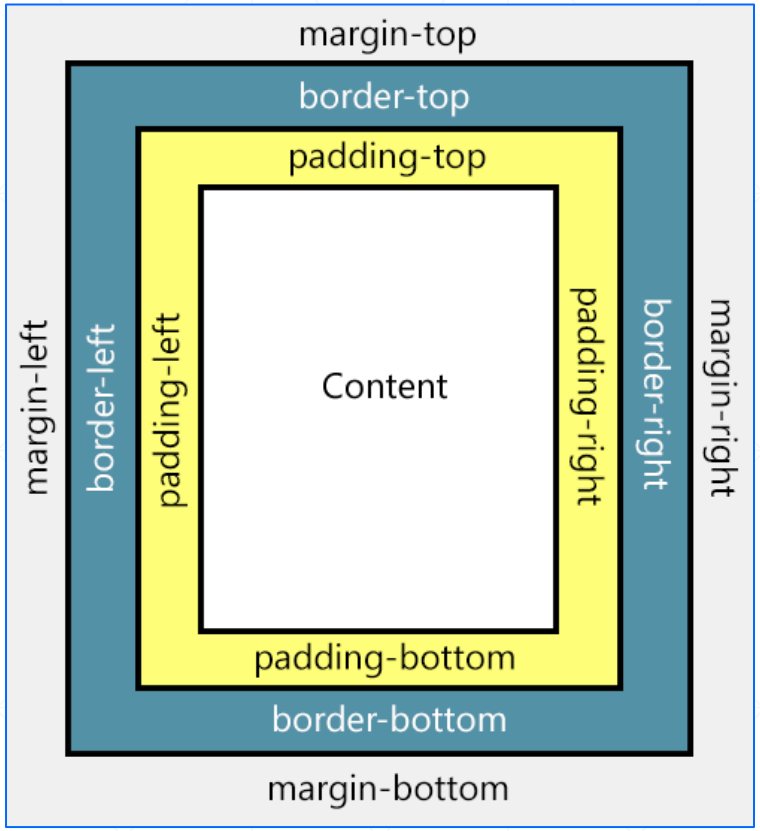
UQAR

Les boites

- Les **boites** sont des *conteneurs* qui incluent des *sous-éléments à l'intérieur*,
- On utilise souvent la balise `<div></div>`
- Voici quelques propriétés qu'il est possible d'utiliser :
  - Fixer la **hauteur** et la **largeur** de la boite avec : `height: ;` `width: ;`
  - Définir les **valeurs maximales** : `max-width: ;` `max-height: ;`
  - Définir les **valeurs minimales** : `min-width: ;` `min-height: ;`
  - Les propriétés suivantes sont particulièrement utiles :

    `margin: ;` `padding: ;` `border: ;`
- Les valeurs sont généralement définies en **pixels** ou **pourcentage**.

Les boites – *Suite*



Source : *http://www.differencebetween.net/technology/difference-between-padding-and-margin/*

UQAR

Les boites – *Suite*

- Il est possible d'appliquer du style des **bordures** :
  - L'*épaisseur* de la ligne : `border-width: ;`
  - Le *style* de la ligne : `border-style: ;` `solid, dotted, dashed`
  - La *couleur* de la ligne : `border-color: ;`
- Il est possible de passer du mode **`inline`** en **`block`** et vice versa avec : `display: ;`
  - Elle prend les valeurs suivantes : `inline` ou `block`
- Il est également possible d'*arrondir les bordures* avec : `border-radius: ;`

UQAR

Les boites – *Suite*

- On peut contrôler le fait qu'un élément soit affiché ou pas avec la propriété : `visibility`
- Elle prend les valeurs suivantes : `hidden` `visible`
- Afin de rajouter un effet d'ombre sur les boites, on peut utiliser : `box-shadow`
  - Exemple : `box-shadow: 5px 5px 5px #777;` ← *Exemple de paramètres*

UQAR

Les listes et les tableaux

Il est possible d'effectuer différentes opérations pour styliser les listes :

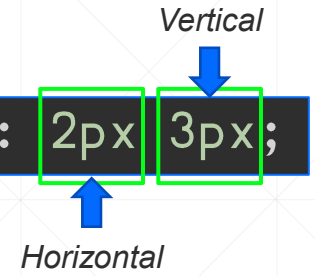
- Changer les *puces* avec des *formes prédéfinies* avec : `list-style-type: ;`
  - Pour les *listes non-ordonnées* : `none, disc, circle, square`
  - Pour les *listes ordonnées* : `decimal, lower-alpha, upper-alpha, lower-roman, upper-roman`
- Il est également possible d'utiliser des *images* avec : `list-style-image: url("");`
- Afin de jouer sur l'*indentation* du texte : `list-style-position: ;`
  - La valeur *par défaut* est *extérieure* : `outside`
  - Il est possible que la seconde ligne *n'ait pas d'indentation* avec : `inside`
- Toutes ces propriétés peuvent être déclarées avec : `list-style: ;`

UQAR

Les listes et **les tableaux**

En plus des autres propriétés déjà présentées, il y a :

- Bordures au niveau des *cellules vides* : `empty-cells: ;`
  - Si *affichées* : `hide`
  - Si *cachées* : `show`
- Gestion de l'espace entre les cellules : `border-collapse: ;`
  - *Aucun espace* avec : `collapse`
  - *Présence d'espace* avec : `separate` *Combine avec* `border-spacing: 2px 3px;`



- Pour changer la forme du *curseur* : `cursor: ;` `pointer, crosshair, move` `url("")`

UQAR

Les images

- Afin de définir une *image de fond*, on utilise `background-image: ;`
  - On doit spécifier le *chemin* vers l'image avec : `url("")`
- Pour spécifier le mode de répétition de l'image de fond : `background-repeat: ;`
  - Elle prend les valeurs suivantes : `no-repeat, repeat-x, repeat-y, repeat`
- Pour déterminer si l'image reste à la même position : `background-attachment: ;`
  - Si elle reste à la *même position* : `fixed`
  - Si elle suit le *défilement* de la page : `scroll`
- En ce qui concerne la *position* de l'image : `background-position: p1 p2;`

`left, center, right` `top, center, bottom`

- On peut également utiliser le raccourci : `background: ;`

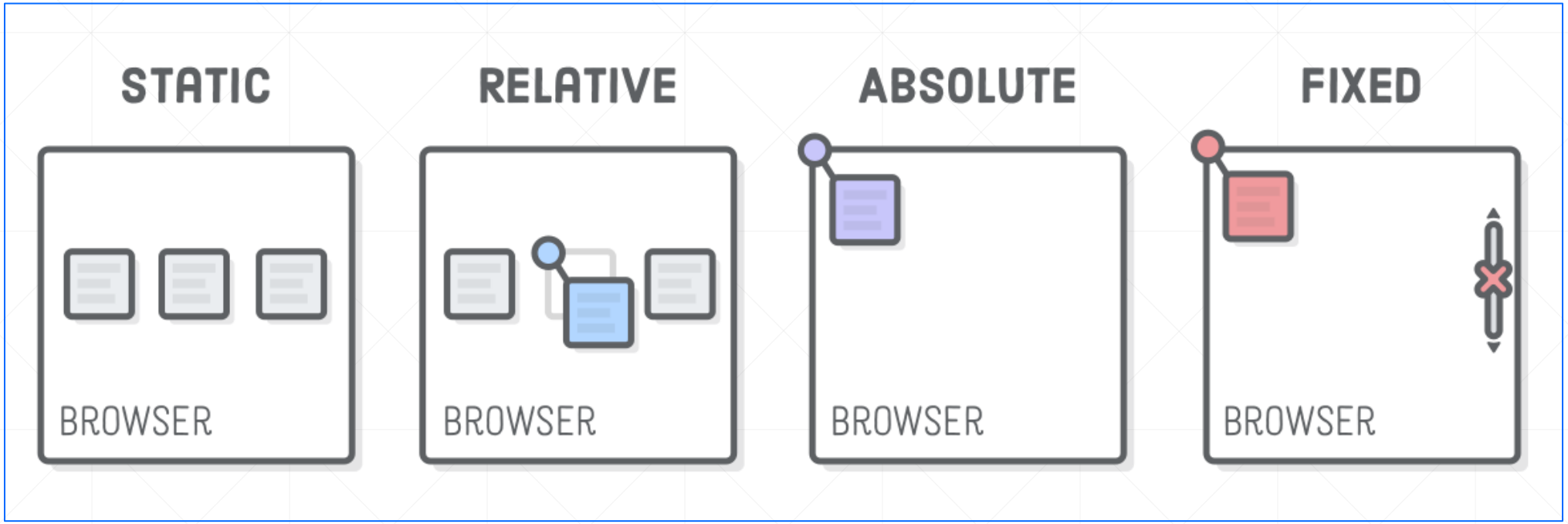
UQAR

Positionnement des éléments

Il y a différents types : `position: ;`

1. **Normal :** `static`
   - *Les éléments apparaissent l'un après l'autre.*
2. **Relatif :** `relative`
   - *Il permet de modifier la position de l'élément cible sans affecter les autres éléments.*
3. **Absolu :** `absolute`
   - *Il permet de positionner l'élément courant par rapport son conteneur.*
4. **Fixe :** `fixed`
   - *Il permet de positionner l'élément par rapport à la fenêtre du navigateur Web.*
5. **Flottant :** `float: ;` `right, left`
   - *Il permet de positionner le l'élément où l'on veut en le retirant du positionnement par défaut.*

Positionnement des éléments



Source : *https://www.internetingishard.com/html-and-css/advanced-positioning/*

UQAR

Framework CSS

- Il est possible d'utiliser **Frameworks CSS** afin de *faciliter* la mise en forme d'une page Web,
- C'est un réel gain de temps pour les développeurs, surtout *Back-End*,
- Il y a différents **Frameworks** qui sont disponibles gratuitement sur internet :

Boostrap 5

https://getbootstrap.com/

Tailwind CSS

https://tailwindcss.com/

Material UI

https://m3.material.io/

Bulma

https://bulma.io/

- Afin de pouvoir les utiliser, il faut inclure les *fichiers CSS* & *JavaScript*.

Questions & Discussion

UQAR

Bibliographie | EN

1. Duckett, J. (2011). *HTML & CSS: design and build websites* (Vol. 15). Indianapolis, IN: Wiley.
2. Duckett, J. (2014). *JavaScript and JQuery: interactive front-end web development*. Wiley & Son.
3. Banks, A., & Porcello, E. (2017). *Learning React: functional web development with React and Redux*. " O'Reilly Media, Inc.".
4. Haverbeke, M. (2014). *Eloquent javascript: A modern introduction to programming*. No Starch Press.

UQAR

Bibliographie | FR

1. Templier, Thierry & Gougeon, Arnaud (2007). JavaScript pour le Web 2.0 Programmation objet, DOM, Ajax, Prototype, Dojo, Script.aculo.us, Rialto. Éditions Eyrolles.
2. Porteneuve, Christophe (2008). Bien développer pour le Web 2.0 : Bonnes pratiques Ajax. Éditions Eyrolles.
3. Engels, Jean (2012). HTML5 et CSS3 : Cours et exercices corrigés. Éditions Eyrolles.
4. Martin, Michel (2014). HTML5, CSS3 & jQuery : Créez votre premier site web. Éditions Pearson.