**UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE D’ORAN**

**Département Mathématques et Information**

**Année : 3ème LMD Option :ISIL Date :2021/2022**

**LA PROGRAMMATION AVANCEE DU WEB**

**INTRODUCTION ET RAPPELS**

Le Web a été conçu par Tim Berners-Lee. Le WEB (www ou World Wide Web) appelé aussi « la toile mondiale » est un ensemble de pages web reliées entre elles par des liens hypertextes. Ces pages sont stockées dans un serveur web. La communication entre le serveur web et les navigateurs se fait grâce au protocol HTTP (S) (HyperText Transfer Protocol) (Security).

Plusieurs standards du web ont vu le jour et ils sont toujours en évolution. Parmi ces standards, il ya HTML 5(HyperText Markup Language), CSS 3 (Cascading Style Sheet), Javascript, PHP, Ajax etc...

Ces standards sont utilisés pour créer des sites et des applicatiions web.

Ces standards sont promus par le W3Consortium fondé par Tim Berners-Lee.

**I -LES SITES WEB**

Un site web peut être une page ou plusieurs pages . Les sites sont statiques, ils ont la plupart un aspect informatif c’est à dire qu’ils affichent uniquement les informations et les données de la page.

**II-LES APPLICATIONS WEB**

Les applications web se concentrent sur l’intéractivité. Une application web peut par exemple proposer des transactions ou utiliser des logiciels pour un but précis. Exemple, certains services de Google comme Google Maps, GMAIL sont des applications web.

Pour concevoir des sites web ou bien adapter des applications web, il faut commencer par les bases comme HTML et CSS,etc.... Mais pour concevoir des applications web comme des jeux , voire des programmes de retouche photo , il convient de miser en plus sur d’autres bases. Parmi celles-ci, on trouve notamment C++/C# et Java.

**III- HTML 5**

En général, HTML sert à structurer le contenu de la page web. Pour la mise en forme , on utilise un autre langage qui est le CSS.

Plusieurs développements de HTML ont vu le jour, de la première spécification HTML jusqu’au HTML 5. Comme pour CSS nous sommes aujourd’hui à CSS3 et prochainement CSS4, CSS5. Tous ces développements sont motivés par l’évolution des navigateurs.

**III-1 HISTORIQUE**

HTML 5 a été conçu au sein de W3C et WHATWG(Web Hypertext Application Technology Working Group).

HTML 5 a succédé à XHTML(Xml HTML) qui lui même a succédé à HTML 4.

Donc le passage du HTML4 à XHTML a apporté une écriture plus précise. Ces règles de XHTML sont simples, il faut :

* Chaque balise doit être fermée même celle qui ne contient pas de contenu :

exemple <br> et <img>

* Les balises s’écrivent en minuscule.
* Les valeurs des attributs sont entre guillements doubles:

**exemple** <**img src="usto.dz**">

**III-2 LES AVANTAGES DE HTML 5**

* Lorsque vous codez une page en HTML 5 théoriquement elle s’affichera de la même façon quelque soit le navigateur (Firefox, Opéra, Chrome, Internet Explorer), l’IOS(Windows, Linux, Mac) et le supprot (PC, Smartphone, Tablette) car on dit qu’il est compatible.
* L’autre avantage est sa simplicité (au niveau des balises comme <**!DOCTYPE**> et on peut ne pas respecter les règles qu’on a décrit dans l’historique)

***Remarque***

Mais si on veut avoir un code lisible, respecter les règles de XHTML.

**III-3 LES OUTILS NECESSAIRES POUR ECRIRE DES DOCUMENTS HTML /CSS3**

*A- Les navigateurs*

Si un seul navigateur suffit pour afficher le résultat de votre code HTML 5 avec CSS3 , il vous en faudra plusieurs sous la main pour tester la compatibilité de votre page web. Parmi les plus connus, nous avons :

-Chrome

- Internet Explorer

- Safari

-Opéra

- Firefox

Il reste le problème des versions car deux versions différentes d’un même navigateur peut visualiser la page différamment.

*A-1 Les solutions*

1. La première solution  est les tests en ligne. Il existe différents services en ligne qui permettent de tester l’affichage de votre site web sur plusieurs versions des principaux navigateurs.

Exemple d’un service : BrowserShots et (http ://html5demos.com)

L’inconvenient de ce service est qu’il faut que notre site soit hébergé et non pas local.

1. La deuxième solutiin c’est d’ajouter des plug-in (extensions) aux navigateurs.
2. La troisième solution est d’utiliser des scripts . **Exemple** : **Modernizr** qui est une biblithèque Javascript. Elle indique simplement si oui ou non le navigateur prend en charge la fonctionnalité que vous voulez essayer.

*B- Les editeurs de texte*

Pour écrire un document HTML 5, nous avons

- Bloc-note

- Notepad++

- etc.....

**III-4 LA VALIDATIION DES DOCUMENTS HTML 5**

Avant d’interpréter le document HTML 5 par le navigateur, nous pouvons le valider grâce à un analyseur syntaxique en ligne de w3c.

.

Ressouce : validateur officiel de W3C : http ://validator.w3c.org

**III-5 LA SYNTAXE DE HTML5**

Nous trouvons,

A- Les éléments (Balises) qui structurent le contenu du document en délimitant des blocs qui seront amenés à contenir des paragraphes, des titres, différents types de média(sons, vidéos, images), des formulaires ou encore des liens hypertextes.

Ces éléments peuvent se succéder par exemple plusieurs paragraphes dans un bloc <DIV> qui va former une hiéarchie de balises. On parlera de balise parente ou enfant (notion d’héritage)

**Exemple**

<**ul**>

<**li**> HTML<**/li**>

<**li**>CSS<**/li**>

<**li**> Javascript<**/li**>

<**/ul**>

<**ul**> est la balise parente et <**li**> sont les balises enfants.

Mais les différentes balises <**li**> sont des balises frères.

**A-1 Les diférents types d’éléments**

**A-1-1 Les éléments de type « ligne »**

* Element <span> qui sert à délimiter une partie de texte ayant une mise en forme commune.
* Elment<a> pour le lien
* Element <img> et objet <object>
* Elment <em> mise en relief ou encore plus en évidence<strong>
* Element <q> extraits de citation(apparaît entre guillemets et <cite> (en italique)
* Elmenet <code>
* Elmeent <samp> et <var>
* Elment <abbr>
* Elment <ins>

Ces éléments peuvent être imbriqués mais ils ne peuvent pas contenir des éléments de type Bloc

**A-1-2 Les éléments de type Bloc**

* Elément <div>
* Titre<h1> à<h6>
* Paragraphe <p>
* Liste et élements de liste <ul>,<ol> et <li>
* Liste de définition <dl> ainsi que ses éléments <dd> et <dt>
* Citation <blockquote>
* Texte préformaté<pre> (affichage fidèle de tous les espaces et retours à la ligne).

**B-Les attributs**

Un élement peut avoir zero ou plusieurs attributs

**Exemple**

Pour image en HTML

<img src= "cirl.jpeg "/>

la balise <**img**> a un attribut « **src**» qui est le nom du fichier de l’image qui sera affichée dans la page.

**C-Les commentaires**

**Syntaxe**

< !-- commentaire-->

**D-L’encodage des caractères**

Il ya plusieurs codes

* Le code ASCII (code sur 7 bits)
* Le code labellisé ISO-8859-1 ‹(code sur 8 bits :Latin 1)
* Le code UTF-8(UCS Transformation Format 8 bits) c’est ce dernier qui est

utilisé de nos jours

**E- Les entités**

Pour écrire les caractères spéciaux qui ne peuvent s’afficher, on utlise une syntaxe précise

**Exemple**

&grave ; correpond à (à)

&#169 ;  ©

**F-Le type MIME**

Le type MIME( Multipurpose Internet Mail Extension) est constitué de deux parties normalisées : Le type et le sous-type séparé par un caractère slash(/)

**Exemple**

**application/msword** correspond à un fichier Word(Microsoft)

**text/css** feuille de style CSS

**text/html**HTML

**III-6 STRUCTURE GENERALE D’UN DOCUMENT HTML 5**

La structure est comme suit :

<**!DOCTYPE html**>

<**html**>

<**head**>

........

<**/head**>

<**body**>

......

<**/body**>

<**/html**>

*Remarque*

Pour chaque élément et attribut nouveaux, on mentionne ces derniers avec le mot clè (HTML 5) pour la suite du document.

Nous remarquons la simpcité de HTML 5 au niveau de la balise < !DOCTYPE>.

Par comparaison avec les autres déclarations de cettte balise pour les autres normes de HTML

**Exemple**

**HTML 4.01**

<**!DOCTRYPE HTML PUBLIC** "**-//W3C//DTD HTML 4.01//EN**"

"**http ://www.w3.org/TR/html4/strict.dtd**">

**XHTML 1.0**

< **!DOCTYPE html PUBLIC** "**-//W3C//DTD XHTML 1.0 Strict//EN**"

"**http ://www.w3.org/TR/xhtml-strict.dtd**">

**III-6-1 L’élement <html>**

Les attributs associés à cet élément sont :

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| lang | "fr", "en",etc.. | indique la langue du document. |
| manifest (HTML 5) | URL | indique l’adresse du manifeste contrôlant le cache des fichiers pour une utilisation en mode déconnecté (hors ligne). |

**Le style par défaut**

**html**{

**display**:block ;

}

**html :focus**{

outline :none ;

}

**III-6-2 L’élément <head>**

L’en-tête du document est important car il fournit des informations sur le document lui-même que le contenu n’indique pas.

Les éléments inclus de <head> sont :

| **Elements** | **Occurrences** | **Fonction** |
| --- | --- | --- |
| <title> | 1 | Titre du document. |
| <meta> | 0 ou + | Méta-informations. |
| <link> | 0 ou+ | Relation vers des ressources externes (par exemple des CSS). |
| <style> | 0 ou + | Styles CSS embarqués dans le document. |
| <script> | 0 ou + | Scripts embarqués dans le document. |
| <base> (facultatif) | 0 ou 1 | Base d’adresses par défaut et de cible pour tous les liens du document. |

**Exemple**

<**head**>

<**title**> HTML<**/title**>

<**/head**>

**Stytle par défaut de <head>**

**head**{

**display**:none ;

}

**III-6-2-1 L’élement <meta>**

Les attributs sont :

| **Attributs** | **Valeurs( Type)** | **Fonction** |
| --- | --- | --- |
| name | application-name (texte)  author(texte)  description(texte)  generator(texte)  keywords(texte) | Méta-informations relatives à la page : nom de l’application web, auteur, une description du contenu, programme de conception utilisé, mots-clès relatifs au contenu. |
| http-equiv | refresh (entier positif ou entier positif ;url=URL)  default-style (valeur de l’attribut « **title** » d’un élément <style> ou  <link rel= "stylesheet">  content-type (text/html ;charset=code correspond à l’encodage des caractères) | Rafraîchissement de la page.  Feuille de style préférée.  Déclaration du code de la page et du type MIME. |
| content | texte | Cet attribut est obligatoire. En combinaison avec les attributs « **name**» et « **http-equiv** » donne sa valeur à la balise. |
| charset | encodage des caractères | Déclarartion du code de la page pour l’interprétation si celui-ci est différent de l’ASCII. |

**Exemples**

< !-- l’attribut content est obligatoire pour l’attribut name-->

<**meta name=**"**author**" **content=**"**Tim**"**/**>

<**meta name=**"**generator**" **content=**"**Bloc-note**"**/**>

< !-- Rafraîchissemnt de la page toutes les minutes-->

<**meta http-equiv=**"**refresh**" **content=**"**60**"**/**>

< !-- Rediretion vers une autre URL au bout de 5 secondes-->

<**meta http-equiv=**"**refresh**" **content=**"**5 ;http ://www.usto.dz**"**/**>

< !-- Déclaration du style préféré-->

<**style title=**"**Monstyle**">< !--CSS---><**/style**>

<**meta http-equiv=**"**defaukt-style**" **content=**"**Monstyle**"**/**>

**Style par défaut de <meta>**

**meta**{

**display** :none ;

}

**III-6-3 L’élément <link>**

Les attributs sont :

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| href | URL | Adresse de la cible. |
| hreflang | code de la langue | langue de base du document cible. |
| rel | alternate,  archives,author,  first,help,icon,  index,last,  license,next,  pingback,prefetch,  prev, search,  sidebar, stylesheet,  tag, up | Indique la relation existante entre le document courant et celui de la cible (valable uniquementlsi l’attribut **« href** » est précisé). |
| media(HTML 5) | media query | Spécifie une requête de média pour laquelle la cible est optimisée. |
| type | type MIME | Spécifie le type MIME de la cible, par exmple text/javascript pour javascript. |
| sizes(HTML 5) | any ou  hauteur x largeur | Dans le cas d’une relation icône (**rel**= "**icon**"), indique les dimensions de l’image en pixels. Plusieurs valeurs sont possibles, séparées par des espaces.  **any** indique que l’image convient à plusieurs résolutions, par **exemple** lorsqu’elle est au format vectoriel SVG. |

**Exemples**

< !-- Icône associée au document-->

<**link rel=**"**icon**" **href=**"**icone.gif**" **type=**"**image/gif**" **sizes=**"**32x32**">

< !-- Autre icône associée au document-->

**<link rel=**"**icon**" **href=**"**icone.gif**" **type=**"**image/svg**" **sizes=**"**any**">

< !-- Feuile de style pour l’impression-->

**<link rel=**"**stylesheet**" **href=**"**css/print.css**" **type=**"**text/css**" **media=**"**print**">

**Style par défaut**

**link**{

**display** :none ;

}

**III-6-4 L’élément <style>**

Les attributs de <style> sont

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| type | type MIME | Type MIME d’un langage de style. En général c’est text/css. |
| media | media query | Spécifie une ou plusieurs requêtes de média à laquelle l’instruction de style sera appliquée. |
| scoped (HTML 5) | scoped | Indique que les instructions de style s’appliquent uniquement à l’élément parent qui conteint la balise <style> ainsi qu’à ses descendants.  Si cet attribut n’existe pas , les propriètés css seront appliqueés sur l’ensemble du document. |

**Exemple**

<**head**>

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

<**title**> Exemple</**title**>

<**style type=**"**text/css**" **media=**"**screen**">

**body** {

**background**:#EFECCA ;

}

**p** {

**color** :red ;

**font-size**:2em ;

**text-align** :center ;

}

<**/style**>

<**/head**>

< !--Utilisation de l’attribut Scoped (Portée)-->

**Remarque**:

l’élément <**style**> est également permise dans les éléments <**div**>, <**noscript**>, <**section**>, <**article**> et <**aside**> d’où l’intérêt de l’attribut « **scoped**» pour rendre sa portée locale à l’élément qui la contient.

**Exemple**

<**div**>

<**p**> paragraphe dispose du style par défaut<**/p**>

<**article**>

<**style scoped**>

**p** {**color**:red ;}

<**/style**>

<**p**> ce paragarhe sera en rouge<**/p**>

<**/article**>

<**/div**>

**Style par défaut**

**style**{

**display** :none ;

}

**III-6-5 L’élément <base>**

Représente une base commune pour tous les éléments du document. C’est un élément vide (pas de contenu).

Les attributs de <base> sont :

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| href | URL | Indique l’adresse à utiliser comme base de réference (préfixe) pour tous les hyperliens du document. |
| target (HTML 5) | \_blank  \_parent  \_self  \_top | Cible par défaut pour l’ouverture de tous les liens de la page. Cette indication générale est contournée individuellement en utilisant un attribut target dans tout lien. |

**III-7 L’élément <body>**

**Style par défaut**

**body**{

**display**:block ;

**margin** :8px ;

}

**body :focus**{

**outline** :none ;

}

**III-7-1 Les éléments de <body>**

On trouve :

1. **Les éléments de regroupements**
2. **Les Titres et éléments de texte**
3. **Listes**
4. **Lien**
5. **Contenu embarqué**
6. **formulaire**
7. **Tables**

**III-7-1 Les élements de regroupements**

Il ya deux sortes de regroupements : ceux qui possédent une valeur sémantique particulière et d’autres qui n’ont aucun sens seulement leur rôle est de regrouper des éléments ayant des propriètés communes

**a- Balises non sémantiques**

| **Elements** | **Fonction** |
| --- | --- |
| <div> | Conteneur générique ; Elle représente une division du document. |
| <span> | A l’intérieur d’une phrase ,cette balise entoure les mots, groupes de mots ou valeurs qui doivent avoir une mise en forme spécifique. |

**Exemple**

<**div**>

<**h1**> HTML 5 <**/h1**>

<**p**>cours lmd <**/p**>

<**/div**>

**Style par défaut**

**div**{

**display**:block ;

}

**Exemple**

<**p**> HTML signifie <**span lang=**"**en**"> HyperTexte Markup Language <**/span**> en anglais<**/p**>

<**span**> ne possède pas de style par défaut et reste un élément de type en ligne.

**Remarque**

Ces balises ne doivent être utilisées qu’en dernier recours.

**b- Les balises sémantiques**

Parmi ces éléments, on trouve :

| **Elements** | **Détails** |
| --- | --- |
| <section>(HTML 5) | Element fondamental de la découpe sémantique des documents. Il regroupe un contenu relatrif à une même thématique ou une même fonctionnalité. Il peut englober les éléments suivants : <header>, <hgroup>, <h1> à <h6>,<article> etc... |
| <article> (HTML 5) | Cet élément est une spécialisation de <section>. |
| <nav> (HTML 5) | Cet élément abrite des liens de navigation majeurs, permettant de naviguer au sein du document ou vers d’autres pages. |
| <aside> (HTML 5) | Une page Web est bien souvent dotée d’un contenu principal et d’un contenu annexe qui n’est pas essentiel à sa compréhension comme pa**r exemple** de la publicité. |
| <header> (HTML 5) | C’est un élément d’introduction (d’un document, d’une autre section <section>, d’un article <article>) pouvant contenir- selon le contexte- titres, navigation, formulaire de recherche, table des  matières, logo, etc.... |
| <footer> (HTML 5) | Pied de page de section ou d’article pouvant contenir des informations connexes et une navigation annexe. |
| <hgroup> (HTML 5) | Il s’agit de regrouper ensemble des titres successifs de type h1,h2,etc.... Cette balise peut être incluse dans un <header>. Son utilisation n’est pas obligatoire. |
| <address> | Un bloc contenant des informations de contact. Celui-ci s’applique à son plus proche parent de type <body> ou <article>. Si l’élément <address> est unique pour le document (dans <body>), alors il contient les informations de contact pour l’ensemble . S’il se trouve dans <article> alors il contient les informations de contact relatives à l’auteur de l’article uniquement.  Ces informations concernent (pas seuelement une adresse postale) les noms de personne à l’origine du document ou de l’article, à leurs adresses E-mail, leurs coordonnées téléphoniques ou postales. |

**Exemple du code HTML 5**

< **!DOCTYPE html**>

<**html lang=**"**fr**">

<**head**>

<**title**> HTML <**/title**>

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

<**/head**>

<**body**>

< !--Début-->4

<**header**>

<**nav**><**/nav**>

<**/header**>

<**section**>

<**article**><**/article**>

<**article**><**/article**>

<**/section**>

<**aside**><**/aside**>

<**footer**><**/footer**>

< !-- Fin -->

<**/body**>**</html>**

**Remarque**

* On peut avoir plusieurs <section> dans <article>.
* On peut avoir un <header> et un <footer> dans <article>
* Ces éléments sont de type bloc.

**Style par défaut de <section>, <article>, <aside>, <header>, <hgroup>,<nav> et**

**<footer>**

**section**{

**display**:block ;

}

**Style par défaut de**

**address**{

**display** :block ;

**font-style** :italic ;

}

**III-7-2 Les titres et les éléments texte**

Nous avons les éléments de <h1> à <h6> pour les titres.

**Style par défaut**

**h1**{

**display** :block ;

**font-size**:2em ;

**margin**: .67em 0 .67em 0 ;

**font-weight** : bold ;

}

**h2**{

**display**:block ;

**font-size** :1.5em ;

**margin** : .83em 0 .83em 0 ;

**font-weight** : bold ;

}

Les nouveaux éléments de texte pour HTML 5

| **Elements** | **Détails** |
| --- | --- |
| <mark> | cette balise surligne les mots, expressions ou les phrases. Elle signale et met en valeur une portion du document pour attirer l’attention dans un contexte particulier. |
| <time>  Les attributs de <time> sont  - datetime : la date et l’heure en même temps.  le format : datetime= "**YYYY-MM-DDThh :mm :ss**"  **ex** : <**p**> c’est le <**time datetime=**"**2021-01-16T14 :00+02 :00**">de l’écriture de cet exemple. <**/time**><**/p**>  - pubdate : si cet attribut est présent il devient de facto un repère de temps pour l’élément <**article**> qui le contient, ou à défaut le document entier <**body**>s’il n’est pas enfant d’un article  **ex**: <**article**>  <**p**> cet article a été mis en ligne le  <**time pubdate**>2021-01-16  </**time**>.  <**/p**>  <**/article**> | A partir d’une date ou d’une heure écrite dans un format standard, la balise <time> permettra au navigateur de l’afficher au  format « régional » défini par l’internaute sur son ordinateur. |
| <b>, <i> et <s> pour gras, italique et barré | la balise<b> met le texte en gras et sera  utilisée lorsque le texte concerné n’a pas  davantage d’importance que les mots qui  l’entourent, sinon c’est <strong> qui s’applique.  La balise <i> pourra remplacer <em> pour placer des mots en italique, dans le cas où cela ne correpond pas à une mise en relief.  la balise <s> permet de rayer un texte qui est incorrect ou qu’il n’est plus approprié.  **ex**: <**h3**> Solde sur les vestes <**/h3**>  <**p**> Prix fixe <**s**> 20.000DA <**s**><**/p**>  <**p**> Prix de ce mois <**strong**> 16000DA <**/strong**></**p**> |
| <wbr> | Cette balise autorise des coupures de lignes de mots qui sont trop longs.  **ex**: <**p**> le mot le plus long de la langue française est anti<**wbr**>constitu<**wbr**>tionnellement <**/p**> |
| <details> | Elle donne la possibilté d’afficher sur demande des précisions sur un texte. Elle est initialement masquée et il suffira de cliquer sur le titre associé, qui est affiché à l’aide de la balise <summary>  **ex**: <**details**>  <**summary**> Le W3><**/summary**>  <**p**> W3C signifie World Wide Web  Consor tium.<**/p**>  <**p**> il a été cré en 1994 par Tim Berners-Lee.<**/p**>  <**details**> |
| <ruby> | La prise en compte des caractères annotés de certaines langues étrangères, notamment asiatiques. Très peu de navigateurs supportent nativement cette balise pour le  moment. |

**Style par défaut**

**mark**{

**background-color** :yellow ;

**color** :black ;

}

**s**{

**text-decoration** :line-through ;

}

**ruby**{

**text-indent**:0 ;

}

b{

**font-weight** :bolder ;}

**i**{

**font-style** :italic ;

}

**III-7-3 Les éléments des listes HTML 5**

La balise <**ol**> crée automatiquement une liste numérotée, attribuant un nouveau numéro à chacune des lignes <**li**>.

Les attributs spécifiques à <ol> sont

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| start | nombre entier | Valeur de départ pour la numérotation. |
| reversed (HTML 5) | reversed ou " " ou sans valeur | Indique que la numérotation est descendante. |
| type | 1 ou a ou A ou i ou I | Type de numérotation (numérique classsique, lettrée, chiffres romains. |

**Exemple**

<**ol start=**"**2**" **type=**"**A**">

<**li**> Premier Element<**/li**>

<**li**> Deuxième Elment <**/li**>

<**li**> Troisième Element <**/li**>

<**/ol**>

<**ol reversed=**"**reversed**">

<**li**> Troisième Element<**/li**>

<**li**> Deuxième Element <**/li**>

<**li**> Premier Elment <**/li**>

<**/ol**>

**Remarque**

Il ya aussi les balises <**ul**> (liste non ordonnée), <**dl**> (liste de définition), <**dt**> (Terme de liste de définition) et <**dd**> (élément de description de la liste de définition)

**III-7-4 Les Liens**

l’élément <**a**> (anchor en anglais ou ancre )est un hyperlien.

Les attributs de <a> sont

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| href | URL | cible du lien. |
| hreflang | code de la langue | langage de base de la cible du lien. |
| media (HTML 5) | media query | Spécifie une requête de média pour laquelle la cible est optimisée. |
| rel | alternate, archives, author,  bookmark, external, first, help,  index, last, license, nextnofollow,  noreferrer, prefetch,, prev,search,sidebar,tag,up | Spécifie la relation établie par le lien, entre le document courant et la cible. Plusieurs valeurs peuvent être combinées séparées par des espaces. |
| target | \_blank,  \_parent,  \_self,  \_top | Attribue un contexte de navigation dans lequel suivre le lien. |
| type (HTML 5) | type MIME | Le type MIME de la destination du lien. |
| download (HTML 5) | nom du fichier après téléchargement ou (vide) | Indique que la ressource liée est prévue pour être téléchargée. Si une valeur est donnée à l’attribut, elle représente le nom du fichier après téléchargement. |

**Remarque**

Le lien peut être absolu, c’est à dire formuler une adresse complète, notamment le cas d’un lien menant vers un autre site hébergé sur un autre nom de domaine.

Il peut être relatif, si l’on souhaite se baser sur l’adresse courante- affichée dans la barre d’adresses- et faire réference à un document stocké au même niveau, ou dans un niveau différent( sous-répartoire, répertoire parent, répertoire racine).

**Style par défaut**

**a :link,a :visited**{

**color** :valeur interne ;

**text-decoration** :underline ;

**cursor**:auto ;

}

**a :link :active, a :visited :active**{

**color** :valeur interne ;

}

**III-7-5 Contenu embarqué**

Les éléments du contenu sont :<**img**>, <**audio**>,<**video**>, <**source**>, <**track**> et <**canvas**>

**III-7-5-1 L’élément <img>**

Il ya différents formats de l’image : GIF, PNG et JPEG.

Les attributs de <img> sont

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| src | URL | Adresse du fichier de l’image. |
| alt | texte | Texte alternatif à utiliser en cas d’absence de la resssource. |
| width | nombre positif entier | Largeur de l’image en pixels. |
| height | nombre entier positif | Hauteur de l’image en pixels. |
| ismap | ismap(true or false) ou " " ou  (vide) | Indique que l’élément <**image**> donne accès à une image map côté serveur. |
| usemap | chaîne de texte | Nom de l’élément <**map**> à utiliser pour la définition des zones cliquables de l’image map (appelation courante en anglais) côté client |

La valeur booléenne (true ou false) de l’attribut «**ismap** » indique que le navigateur doit transmettre les coordonnées du point cliqué sur l’image au serveur, lorsque celle-ci est bien cliquable (le lien <**a**>). Ces coordonnées sont envoyéesd sous la forme ?x,y et peuvent être analysées par le langage interprété côté serveur.

**a- Liens sur image**

Les liens hypertextes appliqués aux images revêtent les mêmes caractéristiques que les liens sur le contenu texte.

**Exemple**

<**a href=**"**photo/vase.png**" **title=**"**vase plein de roses**"> <**img src=**"**vase.png**" **alt=**"**vase en porcelaine**"**/**><**/a**>

**Remarque**

Les autres éléments qui ont une correspondance avec l’élément <**img**>

<**map**> en conjonction avec <**img**> est utilisé pour concevoir une carte cliquable appliquée sur l’image. L’attribut « **name** » est requis pour cet élément, car il permet de l’associer à l’élément image <**img**> via son attribut « **usemap**».

**Exemple**

<**img src= "afrique.jpg" alt= "continent-afrricain" usemap="#afrimap"**>

<**map name="afrimap"**>

<area shape="rect" coords="264,230,400,422" alt="n-afrique" href="Nord.htm">

<area shape="rect" coords="400,300,450,230" alt="s-afrique" href="Sud.htm">

<area shape="rect" coords="264,230,400,422" alt="e-afrique" href="Est.htm">

<area shape="rect" coords="400,300,450,230" alt="o-afrique" href="Ouest.htm">

<**/map**>

<**area**> est une zone cliquable individuelle faisant partie d’une image map.

<f**igure**> (HTML 5) cet élément n’est pas proprement parlé un contenu embarqué, mais il accompagne l’élément <**img**>. Il petrmet d’associer une légende optionnelle <**figcaption**> à du contenu regroupé en un bloc, qui peut être indépendant du contenu principal du document

**Exemple :** une image avec une légende

<**body**>

< !-- Une image avec une légende-->

<**figure**>

<**img src=**"**Front-de-mer.png**" **width=**"**600**">

<**figcaption**> Photo du front de mer d’Oran <**/figcaption**>

<**/figure**>

**III-7-5-2 L’élément <video> et l’élément <audio> (HTML 5)**

Un média de type vidéo et un média de type audio

**Remarque**

Les attributs des balises <**audio**> et <**video**> sont :

- src : indique la source du contenu c’est à dire le nom du fichier audio ou video à lire

de la page.

- autoplay= "autoplay" entraîne un démarrage automatique du son ou du film dès

l’affichage de la page

- preload permet d’anticiper ou non le chargement du contenu à l’ouverture de la pa

ge. Les valeurs possibles sont : preload="none" (le fichier ne se télécharge qu’à la

lecture), preload="metadata" (seules quelques données associées au fichier sont

téléchargées :taille , durée, liste de lecture, première image pour une vidéo, etc...)

preload="auto"chargement automatique du fichier à lire.

- loop="loop" sert à reboucler en permanence la diffusion.

- controls="controls" affiche les boutons de contro^le proposés par l’interface du na

vigateur, comme volumle, pause etreprise

-mediagroup= " ... "permet d’associer entre eux plusiers fichiers audio et vidéo en

les reliant au même nom de groupe écrit entre guillemets.

De plus la balise <**video**> accepte les attributs « **width** » et « **height** » et l’attribut

«**poster**» qui indique l’adresse de l’image qui s’affichera en attente de la diffusion de la

vidéo.

**Exemple**

<**audio src=**"**mamusique.mp3**" **loop=**"**loop**" **controls=**"**controls**">

votre navigateur ne peut affixher ce contenu audio <**/audio**>

<**video src=**"**film.webm**" **poster=**"**affiche.jpg**" **autoplay=**"**autoplay »**>

Ce navigateur ne peut afficher cette vidéo

<**/video**>

**III-7-5-3 L’élément <source> (HTML 5)**

Une ressource média pour les balises <audio> et <video>. Son avantage est de proposer plusieurs formats de fichiers possédant un attribut « type » différent, mentionnant le type MIME.

**Exemple** : Sources multiples <**video**>

<**video**>

<**source src= "video.mp4" type= "video/mp4"**>

<**source src= "video.ogv" type="video/ogg ; codecs=theora, vorbis"**>

<**source src="video.webm" type= "video/webm ;codecs=vp8, vorbis"**>

cet élément ne supporte pas cette video.

<**/video**>

Dans ce cas, trois sources sont proposées :

* Une vidéo MP4 pour les navigateurs connaissant ce format.
* Une vidéo Ogg(**un conteneur est une structure de fichiers permettant de regrouper des flux vidéo et audio, ayant chacun leur codec, ainsi que des meta-informations telles que les sous-titres, chapitres, etc...**) Theora (**un codec ou code-decode est un procédé de compression et décompression d’un signal numérique permettant de générer un flux respectant une norme.**) à destination des navigateurs supportant le type MIME video/ogg et équipés des codecs  *theora* (video) et *vorbis*(audio).
* Une video WebM à destination des navigateurs supportant le type MIME video/webm et équipés des codecs vp8 (video) et vorbis (audio.

**III-7-5-4 L’élément <track> (HTML 5)**

Une ressouce texte pour média <audio> et <video>. L’intérêt de cet élément est multiple :

* Fournir des sous-titres et légendes aux personnes malentendants.
* Fournir des audio-descriptions aux personnes mal-voyants.
* Afficher des traductions aux personnes ne comprenant pas la langue parlée.
* Afficher les paroles d’une chanson, par exemple pour un karaoké.
* Donner la possibilité de naviguer par chapitres ou de rechercher « dans » la vidéo.

Le type de données ajoutées est précisé avec l’attribut « **kind** » qui peut prendre les valeurs suivantes :

| **Valeurs** | **Signification** | **Description** |
| --- | --- | --- |
| subtitles | Sous-titres | Transcription ou traduction des dialogues lorsque le son est disponible mais non compris (par exemple lorsque l’utilisateur ne comprend pas le langage original). |
| captions | Légendes | Transcription ou traduction des dialogues, effets sonores et musicaux lorsque l’audio n’est pas disponible ( désactivée lorsque l’utilisateur est un malentendant). |
| description | Descriptions | Descriptions textuelles de la vidéo, destinées à la synthèse vocale (utilisée lorsque l’utilisateur navigue sans écran). |
| chapters | Chapitres | Titres de chapitres destinés à la navigation dans la ressource, affichés sous forme de liste par le navigateur. |
| metadata | Méta-informations | Données à destination du langage de script, non affichées apr le navigateur. |

**Exemple**

<**video src=**"video.webm" **id=**"mavideo">

<**track kind=**"subtitles" **src=**"video.fr.vtt" **srclang="fr**" **label=**"Français" **default>**

<t**rack kind=**"captions" **src=**"**video.fr.cap.vtt**" **srclang=**"fr" **label=**"Français pour malentendant">

<**track kind=**"subtitles" **src=**"video.en.vtt" **srclang=**"en" **label=**"English">

<**track kind=**"descriptions" **src=**"video.fr.desc.vtt" **srclang=**"fr" **label=**"Description Texte">

<**/video**>

**III-7-5-5 L’élément <canvas> (HTML 5)**

Une zone de dessin bitmap pilotable avec un script. Cette API de dessin 2D pour le Web est une alternative puissante à flash ou java pour concevoir en ligne et dynamiquement des images fixes ou animées, à l’aide de formes ou via des tracés pixel par pixel. Elle peut êtrevinséré dans le flux de la page HTML par une balise éponyme.

La zone active relève d’un élément dont les dimensions sont définies dans le code de la page. Un ensemble de fonctions Javascript permettent un accès à la zone de dessin, en modification ou en lecture.

Parmi ces possibilités figurent :

- des méthodes de tracé de formes géométriques (cercles, rectangles,...) ;

- des méthodes de tracé de polygones ;

- des méthodes de choix de styles de couleurs et de remplissages ;

- des méthodes de tracé de texte ;

- des méthodes d’import et de manipulation d’images ;

- des méthodes de transformation (échelle, rotation, déplacement) qui affectent toute la matrice.

**Exemple**

<canvas id= »dessin » width= »640px » height= »480 »>

Texte à destination des navigateurs qui ne supportent pas canvas.

</canvas>

A l’aide de Javascript, quelques appels de fonctions suffisent pour desssiner :

<script>

**var moncanvas=document.getElementById( 'dessin') ;**

**var ctx=moncanvas.getContext('2d') ;**

**ctx.fillStyle="red" ;**

**ctx.fillRect(20,30,100,50) ;**

</script>

Pour l’élément <canvas> , il faut

***1- une base de départ et un contexte graphique***

**Exemple**

<**canvas id= »dessin » width= »640px » height= »480 »>**

**Texte à destination des navigateurs qui ne supportent pas canvas.**

**</canvas>**

**<**script>

var moncanvas=document.getElementById( 'dessin') ;

var ctx=moncanvas.getContext('2d') ;

// D’autres fonctions de dessin appliquées au contexte ...

***2- Les coordonnées***

Elles sont définies dans un sytème cartésien et débutent dans le coin supérieur gauche de la zone (0,0). Toutes les valeurs sont exprimées en pixels. Deux axes sont pris en compte : horizontal (x) et vertical (y).

Exemple : Un point situé à (3,2) sera situé à 3 pixels de la gauche et 2 pixels du haut.

**(0,0)**

**X**

**Y**

**3-** Les formes géométriques

Les formes géométriques sont les fonctions de dessin les plus basiques.

Les méthodes de dessin des formes sont :

| Fonction | Rôle | Détails des arguments |
| --- | --- | --- |
| fillRect(x,y,w,h) | Rectangle plein | x,y :coin supérieur gauche du rectangle.  w,h :largeur et hauteur |
| strokeRect(x,y,w,h) | Rectangle surligné | (similaire) |
| clearRect(x,y,w,h) | Rectangle vide(efface) | (similaire) |

**Exemple**

// Un rectangle plein (rouge)

**ctx.fillStyle= »red » ;**

**ctx.fillRect(10,10,200,,100) ;**

**4- Chemins**

Canvas n ’embarque que très peu de primitives géométriques. Donc pour tracer des autres formes, il se base sur les chemins. Ces derners sont définis étape par étape et point par point.

Les méthodes de tracé sont :

| **Fonction** | **Rôle** | **Détails des arguments** |
| --- | --- | --- |
| beginPath() | Débute un nouveau chemin | (aucun) |
| closePath() | Ferme le chemin | (aucun) |
| moveTo(x,y) | Débute un nouveau sous chemin avec le point donné | x,y : point de départ |
| lineTo(x,y) | Ajoute le point au sous-chemin, connecté à une ligne. | x,y :point d’arrivée |
| rect(x,y,w,h) | Ajoute un rectangle au chemin | x,y :point de départ (coin supérieur gauche)  w,h :largeur,hauteur |
| arctTo(x1,y1,x2,y2,r) | Ajoute le point au sous-chemin, connecté par un arc | x1,y1 :point de contrôle n°1  x2,y2 :point de contrôle n°2  r :rayon |
| arc(x,y,r,a1,a2,c) | Ajoute un arc au sous-chemin, connecté au point précédent par une ligne | x,y :point de départ  r :rayon  a1,a2 :angle de départ,fin  c :sens de rotation |
| bezierCurveTo(ctx1,cty1,ctx2,cty2,x,y) | Ajoute le point au sous-chemin, connecté par une courbe de Bézier | ctx1,cty1 :point de contrôle n°1  ctx2,cty2 :point de contrôle n°2  x,y :point d’arrivée |
| quadraticCurveTo(ctx,cty,x,y) | Ajoute le point au sous-chemin, connecté par une courbe de Bézier quadratique. | ctx,cty :point de contrôle  x,y :point d’arrivée. |

Les méthodes remplissage et surlignage sont :

| Fonction | Rôle | Détails des arguments |
| --- | --- | --- |
| fill() | Applique un remplissage au chemin pour créer une forme pleine | (aucun) |
| stroke() | Applique un style de trait (surlignage) au chemin | (aucun) |

**5- Styles de traits, remplissage et couleurs**

Les méthodes fill() et stroke() vues précédemment emploient des valeurs de style, courantes au contexte, telles que fillStyle() pour la couleur de remplissage et strokeStyle pour la couleur des traits.

Les propriétés graphiques sont :

| **Propriétés** | **Rôle** | **Valeurs** |
| --- | --- | --- |
| fillStyle | Style de remplissage à l’intérieur des formes | Code couleur CSS, dégradé ou motif |
| strokeStyle | Style pour les lignes autour des formes | Code couleur CSS, dégradé ou motif |
| lineWith | Largeur de lignes | Nombre positif |
| lineJoin | Style de jointure des lignes | bevel,round, ou miter |
| lineCap | Forme de fin de ligne | butt,round, ou square |
| miterLimit | Limite de fin de ligne avec miter | Nombre positif |
| globalAlpha | Transparence générale | Nombre positif ou nul (0 à 1) |

**Exemple :** context.fillStyle= »red »

**6- Transformations et états du contexte**

Tous les tracés sont admissibles à des opérations géométriques : translations, rotations, mises en échelle et transformations.

Les méthodes de transformation sont :

| **Fonction** | **Rôle** | **Détails des arguments** |
| --- | --- | --- |
| scale(x,y) | Modification de l’échelle (réduction ou agrndissement) | x,y :coefficients appliqués horizontalement et verticalement (1= Pas de redimensionnement) |
| rotate(angle) | Rotation | angle :angle de rotation en radians |
| translate(x,y) | Translation | x,y : déplacements appliqués horizontalement et verticalement |
| transform(m11,m12,m21,m22,dx,dy) | Multiplie la matrice de transformation courante par  m11 m21 dx  m21 m22 dy  0 0 1 | Matrice de transformation |
| setTransform(m11,m12,m21,m22,dx,dy) | Réinitialise la transformation courante à la matrice d’indentité, et invoque transform() avec les mêmes arguments | Matrice de transformation |

**7- Images (dans canvas)**

L’intérêt majeur de canvas est de pouvoir importer des images et les manipuler.

**8- Texte (dans canvas)**

**9- Vidéo et Audio (dans canvas)**

**III-7-6 Les formulaires en HTML 5**

Le HTML 5 introduit de nombreux attributs pour mieux définir les formulaires et vérifier les types de données sans passer par des scripts. En plus de ces attributs, quelques nouvelles balises de formulaires ont été crées pour enrichir ces derniers.

Nous allons donné un tableau récapitulatif des éléments avec leurs attributs des formulaires (les nouveaux éléments sont indiqués par HTML 5).

| **ELEMENTS** | **ATTRIBUTS** |
| --- | --- |
| <form> | accept-charset,  action, autocomplete (HTML 5), enctype,  method, name, novalidate(HTML 5), target |
| <input> | type, name, form(ou for), autofocus (HTML 5), form, disabled, readonly,required(HTML 5) dirname (HTML 5), placeholder (HTML 5), size (HTML 5), pattern (HTML 5), min (HTML 5), max (HTML 5), step (multiple du nombre donné pour le pas ex :step= »5 » muliple de 5)  (HTML 5), autocomplete (HTML 5), multiple (HTML 5),value,list |
| <fieldset> | for, name, disabled |
| <legend> |  |
| <label> | form(ou for) |
| <textarea> | rows, cols, name, wrap (HTML 5), maxlength (HTML 5) |
| <select> | size, name, multiple |
| <option> | selected (ou disabled) |
| <optgroup> | label, disabled |
| <datalist> (HTML 5) | id |
| <button> | for, name, value,type,disabled |
| <output> (HTML 5) | for, name, disabled |
| <keygen> (HTML 5) | challenge, keytype, name |
| <progress> (HTML 5) |  |
| <meter> (HTML 5) | name, value, max, min, low, high, optimum |

**Exemples :**

<**label**> Livre préferé<**/label**>

<**input type =**"**text**" **list=**"**livres**" **name=**"**Livre**">

<**datalist id=**"**livres**">

<**option value=**"**html**">HTML </**option**>

<**option value=**"**css3**">CSS 3<**/option**>

<**option value=**"**javascript**"> JAVASCRIPT <**/option**>

<**/datalist**>

<**input type=**"**text**" **id=**"**recherche**" **name=**"**int**" **size=**"**30**" **placeholder=**"**adressse, ville, code postal....**">

L’élément <**input**> est particulier car il présente diverses fonctionnalités fournies par son principal attribut « **type** »

Les valeurs possibles pour l’attribut « type » de la balise <input> sont

| **Valeurs** | **Fonction** |
| --- | --- |
| text | champ de texte |
| password | champ de mot de passe |
| hidden | champ caché invisible |
| radio | bouton radio(un seul choix) |
| checkbox | case à cocher (choix multiples) |
| button | Bouton |
| reset | Remise à zéro du formulaire |
| submit |  |
| image | Image cliquable |
| file | Fichier |
| tel (HTML 5) | Numéro de téléphone |
| url (HTML 5) | Adresse URL/URI |
| email (HTML 5) | Adresse E-mail |
| search (HTML 5) | Champ de recherche |
| date (HTML 5) | Date |
| time (HTML 5) | Heure |
| datetime (HTML 5) | Date et heure |
| datetime-local (HTML 5) | Date et heure (locales) |
| month (HTML 5) | Mois |
| week (HTML 5) | Semaine |
| number (HTML 5) | Valeur numérique |
| range (HTML 5) | Valeur numérique d’un intervalle |
| color (HTML 5) | Une couleur RVB (RGB) |

**Exemple**

<**input type=**"**search**" **name=**"**Recherche**" **autofocus**>

**Remarque**

La page ne doit contenir qu’un seul élément portant cet attribut.

**III-7-7 Les tables**

Les éléments imbriqués dans la balise <table> sont

| **Elements** | **Occurrences** | **Fonction** |
| --- | --- | --- |
| <caption> | 0 ou 1 | Titre du tableau. |
| <colgroup> | 0 ou + | Propriétés pour un groupe de colonnes. |
| <thead> | 0 ou 1 | En-tête du tableau. |
| <tfoot> | 0 ou 1 | Pied du tableau. |
| <tbody>  ou  <tr> | 0 ou +  1 ou + | Corps du tableau(tr, th, td).  Lignes du tableau (contenant th, td). |

**Exemple**

<**table**>

<**tr**>

<**td**> Cellule 1<**/td**>

<**td**> Cellule 2 <**/td**>

<**/tr**>

<**/table**>

**Style par défaut de <table>**

**table**{

**display** : table ;

**border-collapse**: separate ;

**border-spacing** : 2px ;

**border-color**: gray ;

}

L’atribut de <**table**> est

| **Attribut** | **Valeur** | **Fonction** |
| --- | --- | --- |
| summary | texte | Un résumé du contenu de la table hérité des précédentes spécifications HTML, il est conseillé d’utiliser à sa place <**caption**> ou d’autres éléments texte (comme <**details**>). |

**Exemple** complet pour <**thead**>, <**tfoo**t> et <**tbody**>.

<table>

<**thead**>

<**tr**>

<**th**> Participant <**/th**>

<**th**> Score <**/th**>

<**/tr**>

<**/thead**>

<**tfoot**>

<**tr**>

<**td**> Total <**/td**>

<**td**> 463 <**/td**>

<**/tr**>

<**/tfoot**>

<**tbody**>

<**tr**>

<**td**> Mohamed <**/td**>

<**td**> 333 <**/td**>

<**/tr**>

<**tr**>

<**td**> Noureddine <**/td**>

<**td**> 130 <**/td**>

<**tr**>

<**/tbody**>

</table>

**Les styles par défaut de :**

**thead**{

**display** :table-header-group ;

**vertical-align :middle ;**

**border-color** :inherit ;

}

**tfoot**{

**display** :table-footer-group ;

**verftical-align** :middle ;

**border-color** :inherit ;

}

**tbody**{

**display** :table-row-group ;

**vertical-align**:middle ;

**border-color** :inherit ;

}

L’élément <**tr**> représente une ligne de la table. Il contient un ou plusieurs élements <**td**> et/ou <**th**>

**Style par défaut**

tr{

**display** :table-row;

**vertical-align**:inherit;

**border-color** :inherit ;}

L’élément <**td**> peut contenir du texte et autres éléments comme des liens, des images, des listes , etc...

Les attributs associés à <td> sont :

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| headers | identifiants | Une liste d’identifiants (attribut « i**d** ») séparés par des espaces de cellules en-têtes qui relèvent de cette cellule. Utilisé par les navigateurs non visuels. |
| colspan | nombre entier  positif | Nombre de colonnes occupées et fusionnées par cette cellule. |
| rowspan | nombre entier  positif | Nombre de lignes occupées et fusionnées par cette cellule. |

**Exemple**

<table>

<tr>

<th scope="col" colspan= "2"> Animal et groupe </th>

<th scope= "col"> Classe</th>

</tr>

<tr>

<**td**> Chat <**/td**>

<**td**> Chien<**/td**>

<**td** rowspan= "2"> mammifère </**td**>

</tr>

<tr>

<**td**>Singe<**/td**>

<**td**> Baleine<**/td**>

< !-- on omet ce td car il est occupé par la cellule de la précédente ligne-->

</tr>

</table>

**Le style par défaut**

td{

**display** :table-cell ;

**vertical-align**:middle ;

}

L’élement <**th**> est une balixse spécialisée qui joue le rôle d’en-tête de colonne (ou ligne) dans un tableau. Il doit être imbriqué dans <**tr**>.

Les attributs de <th> sont :

| **Attributs** | **Valeurs** | **Fonction** |
| --- | --- | --- |
| scope | row  col  rowgroup  colgroup | Spécifie les cellules concernées par cette cellule en-tête :  row : la ligne courante ;  col : la colonne courante ;  rowgroup : le groupe de lignes courant ;  colgroup : le groupe de colonnes courant. |
| headers | Identifiants | Une liste d’identifiants (attribut « **id** » ) séparés par des espaces de cellules en-têtes qui relèvent de cette cellule. |
| colspan | Nombre entier positif | Nombre de colonnes occupées et fusionnées par cette cellule. |
| rowspan | Nombre entier positif | Nombre de lignes occupées et fusionnées par cette cellule. |

**Exemple**

<table>

<tr>

<**th scope=**"**col**"> Animal <**/th**>

<**th scope=**"**col**">Groupe <**/th**>

<**th scope=**"**col**"> Classe <**/th**>

<tr>

<td> Chat </td>

<td> Chien</td>

<td> Mammifère </td>

</tr>

<tr>

<td>Singe</td>

<td> Baleine</td>

<td>Mammifère </td></tr></table>

**Style par défaut**

th{

**display** :table-cell ;

**vertical-align**:inherit ;

**font-weight**:bold ;

**text-align** :center ;

}

L’élément <**caption**>

Son intérêt est de donner un titre ou une légende à un tableau de données.

<table>

<**caption**>Classification animale<**/caption**>

<tr>

<th scope= »col »> Animal </th>

<th scope= »col »>Groupe </th>

<th scope= »col »> Classe </th>

<tr>

<td> Chat </td>

<td> Chien</td>

<td> Mammifère </td>

</tr>

<tr>

<td>Singe</td>

<td> Baleine</td>

<td>Mammifère </td>

</tr>

</table>

L’élément <**colgroup**> placé en début de <table> avant l’usage de <thead>, <tbody>, <tfoot> et <tr>. Il est utilisé pour spécifier des propriétés communes à un groupe de colonnes. Ces propriétés seront inspirées depuis les attributs affectés à <colgroup>, tels que « **style** ». Elles seront affectées au nombre de colonnes indiqué par l’attribut « **span** ».

<table>

<**colgroup span=**"**2**" **style=**"**background :lightgray**"></colgroup>

<tr>

<td> Cellule 1 de la colonne 1 </td>

<td> Cellule 1 de la colonne 2 </td>

<td> Cellule 1 de la colonne 3 </td>

</tr></table>

**Style par défaut**

colgroup{

**display** :table-column-group ;

}

Une énumération d’un ou plusieurs éléments <**col**> définit des propriétés individuelles pour chacune des colonnes, prenant la priorité sur celles indiquées par le parent <**colgroup**>

L’attribut associé à <col> est :

| **Attribut** | **Valeur** | **Fonction** |
| --- | --- | --- |
| span | Nombre entier positif | Nombre de colonnes auxquelles appliquer les propriétés. |

**Exemple**

<table>

<**colgroup span=**"**2**" **style=**"**background :lightgray**">

<**col style=**"**width :50px ;background=**"**darkgray**">

<**col style=**"**width :100px**">

<**/colgroup**>

<tr>

<td>Cellule 1 de la colonne 1 </td>

<td> Cellule 1 de la colonne 2 </td>

<td> Cellule 1 de la colonne 3 </td>

</tr>

</table>

**Un autre exemple**

<table>

<**colgroup**>

<**col style=**"**width :50px**">

<**col style=**"**width :200px ; background :lightgray**" **span=**"**2**">

<**/colgroup**>

<tr>

<td>Cellule 1 de la colonne 1 </td>

<td> Cellule 1 de la colonne 2 </td>

<td> Cellule 1 de la colonne 3 </td>

</tr>

</table>

**Style par defaut**

col{

**display** :table-column ;

}

**LES FEUILLES DE STYLE CSS3 (Cascading Style Sheets)**

**I- DIFFERENTS TYPES DE STYLE**

* style interne
* style en ligne
* style externe : c’est ce dernier type qui est recommendé.

**Exemple**

<head>

.....

<**style type="text/css"**>

**@import url( "styles.css") ;**

<**/style**>

....

</head>

*Remarque*

La règle «**@import**» peut être utilisée dans une feuille de style externe, ce qui permet d’importer une première feuille de style à l’intérieur d’une deuxième. Dans ce cas, elle doit figurer au début de la feuille.

**II- LES SELECTEURS**

| **Sélecteur** | **Eléments concernés** | **Exemples** |
| --- | --- | --- |
| p{ } | Tous les éléments <p> | <**p**>.... |
| p span{ } | Tous les éléments <span> situés dans l’élément <p> | <**p**>...<**span**>... |
| p,span{ } | Tous les éléments <p> et <span> | <**p**>...<**/p**>  <**span**>....<**/span**> |
| div p span { } | Tous les éléments <span> situés dans des éléments <p>, eux-même situés dans des éléments <div> | <**div**><**p**><**span**>.... |
| . intro { } | Tous les éléments de classe="intro" | <**p class= "intro"**>...  <**span class="intro"**>... |
| p.intro { } | Tous les éléments <p> de  classe ="intro" | <**p class="intro"**>.... |
| #titre { } | L’élément portant l’identifiant="titre" | <**div id="titre"**>....  <**h1 id="titre"**>.... |
| p#titre { } | L’élément <p> portant l’id ="titre" | <**p id= "titre"**>... |
| div#titre  p.intro { } | Tous les éléments <p> de classe "intro" situés dans l’élément <div> portant l’id "titre" | <**div id="titre**>  <**p class="intro">.......</p**>  <**p class="intro**">....<**/p**>  <**/div**> |
| a :hover { } | Tous les liens survolés | <**a**> .....<**/a**> |
| input[type=submit] { }  [alt] { } | Tous les éléments <input> d’attribut type="submit"  Tout élément possédant l’attribut "alt" | <**input type="submit"**> |
| h1+p { } : sélecteur d’élément adjacent | Tous les éléments <p> suivants directement les éléments <h1> | <**h1**>......<**/h1**>  <**p**>......<**/p**> |
| ul>li { } :sélecteur d’enfant direct | Tout élément <li> enfant direct d’un <ul> | <**ul**>  // le style sera appliqué sur le 1er <li>  <**li**>.....<**/li**>  <li>.......</li>  <li>...</li>  </ul> |
| h2~p { } : sélecteur d’éléments frères | Un ou plusieurs éléments <p> précédés par un élément <h2> | <h2>.....  <**p**>.......<**/p**>  <**p**>.....<**/p**>  <h2> |

**III- LES PROPRIETES**

Les propriétés sont des paires de clès et valeurs. La clè est le nom de la propriété, suivie par deux points, terminée par la valeur qui peut prendre différentes formes : mots-clès, valeur numérique simple, valeur numérique avec unité, chaîne de caractères, etc...

Les unités les plus pertinentes sont :

| **Unité** | **Description** | **Type d’unité** |
| --- | --- | --- |
| px | Pixels | Relative |
| % | Pourcentage (en général par rapport à l’élément parent) | Relative |
| em | Cadratin (relatif à la taille de la police des caractères) | Relative |

Les autres unités de taille fixe comme pt(point), pc (pica), cm,mm et in(inch) sont à éviter.

**III-1 Les propriétés du Texte**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| color | Couleur du texte | héritée | **body{color :#0000ff ;}**  **h1.menu{color :#6e05c3;}**  **.utile{color :rgb(30%,80%,20%) ;}**  **a{color :rgb(255,0,0) ;}**  **em {color :green ;}** |
| font | (Délaration groupée raccourci pour **font-style, font-variant, font-weight, font-size, line-height et font-family**) | héritées | **h3{ font :bold 1.2em Verdena, sans-serif ;}** |
| font-weight | Poids de la police | héritée | **p{font-weight :bold ;}**  les valeurs possibles : **bold, normal, lighter et bolder.** |
| font-size | Taille de police | héritée | **h1 {font-size :150% ;}**  **p{font-size :large ;}** /\* **mot clè** \*/ |
|  |  |  |  |
| font-family | Famille de police | héritée | **p{font-family : Arial, « Times New Roman », sans-cherif ;}**  On doit toujours ajouter une police « générique » à la fin parmi : **serif, sans-serif ou monospace.** |
| font-variant | Variante | héritée | **.ville {font-variant : small-caps ;}**  l’autre valeur est :**normal.** |
| font-stretch | Etirement du texte | héritée | p.note{font-stretch :condensed ;}  Les valeurs : **normal, ultra-condensed, extra-condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded** |
| text-decoration | Décoration du texte | non | **a :hover{text-decoration none ;}**  les valeurs : **underline, overline, line-through et blink.** |
| text-transform | transformation du texte | héritée | .pays{text-transfom :uppercase ;}  les valeurs : **capitalize, lowercase, none** |
| text-align | Alignement | héritée | **.auteur{text-align :justify ;}**  les valeurs : **left, right, center.** |
| text-justify ? | Texte justifié |  |  |
| text-indent | Indentatioin | héritée | **p{text-intend :5em ;}**  valeur positive ou négative.   1. **positif**: retrait vers la droite de la première ligne, 2. **négatif**: retrait à gauche de la première ligne, |
| line-height | Hauteur de ligne | héritée | **p{line-height :2em ;}**  Valeurs :  **normal**, ou **valeur positive** pour régler l’espacement entre les lignes.  Même unité que les tailles des polices, ou % |
| **word-spacing** | Espacement de mot | héritée | **.petitespace{word-spacing :-2px ;}**  valeurs :  **normal,**  **valeur positive** pour augmenter l’espace entre les mots,  **valeur négative** pour diminuer l’espace entre les mots.  Même unités que les tailles de polices, sauf % |

**III-2 Les propriétés du fond**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| background | (Déclaration groupée ou propriété raccourcie pour  **background-color,**  **background-image, background-repeat, background-attachment**  **et**  **background-position dans un ordre quelconque.**) | non | **h1 {**  **background :blue url(usto.png) 50% repeat-x fixed ; }** |
| background-color | Couleur de fond | non | **p.relief{**  **background-color :red ;}** |
| background-image | Image de fond | non | **body{**  **background-image :url(photos/usto.png );}**  les valeurs : **url ou none** |
| background-repeat | Répetition de l’image de fond | non | **body{**  **background-repeat :repeat-y ;}**  les valeurs :  **repeat :**  répetition horizontale et verticale,  **repeat-x**: répetition horizontale seulement,  **repeat-y** : répititiin verticale seulement,  **no-repeat** : pas de répetition. |
| background-position | Position de l’image de fond | non | **body{background-position :center top ;}**  les valeurs : Une ou deux valeurs données par des noms ou des nombres.  La 1ère valeur pour l’alignement horizontal :  **left, center, right ou nombre (0%=left,100%=right),**  La 2ème valeur pour l’alignement vertical :  **top, center, bottom ou nombre**  **(0%=top, 100%=bottom).** |
| background-attachment | Attache de l’image de fond par rapport à la page | non | **body{**  **background-attachment :sroll ;}**  les valeurs :  **scroll**: l’image défile avec le contenu ;  **fixed** : l’image reste fixe lors du défilement, seul défile le contenu qui est en premier plan. |
| background-origin | Position du fond par rapport à la boîte de l’élément | non | **div.titre{**  **background-origin :content-box ;}**  les valeurs :  **padding-box** : l’arrière-plan est placé par rapport au début de la marge interne du bloc ;  **content-box** : l’arrière-plan se place par rapport au début du contenu du bloc, après la marge interne. |
| background-size | Taille de l’image de fond par rapport aux dimensions de lélément | non | **#pub{background-size :100px 300px ;}**  les valeurs :  largeur de l’imge de fond en unité de mesure ou en % ;  largeur et hauteur de l’image de fond en unité de mesure ou % ;  **auto** : dimensions réelles de l’image de fond ;  **contain** : dimensions calculées pour que l’image atteigne le bord du bloc , sans le dépasser ;  **cover** : dimensions calculées pour que l’image recouvre entièrement le bloc . |
| background-clip | Prolongement de l’arrière-plan | non héritée | **p.note{**  **background-clip : padding-box ;}**  les valeurs :  **border-box** : l’arrière-plan se prolonge jusque sous la bordure ;  **padding-box** : l’arrière-plan s’étend sous la marge interne , mais pas sous la bordure ;  **content-box** : l’arrière-plan n’apparaît que sous le contenu, mais pas sous les marges internes ni sous la bordure. |

**III-3 Les propriétés des ombrages et de la transparence**

Avant d’aborder ces propriétés, nous allons définir l’opacité et les différents codes de couleurs.

L’opacité (opacity) permet des effets de semi-transparence

Le codage RGB (standard) RGBA (Red Green Blue Alpha), le A désigne la couche alpha, c’est à dire le niveau de transparence. Il est donné par une valeur comprise entre 0 (transparence complète) et 1 (opacité totale)..

Le codage HSL (Hue, Saturation, Lightness, soit en français TSL : Teinte, Saturation, Luminosité) et le codage HSLA(Alpha :opacité).

Le HSL se rapproche de la perception humaine des couleurs :

- La teinte représente la couleur , indiquée par un angle (en degrès) dans le cercle chromatique des couleurs ex :0 :rouge, 60=jaune, 120=vert,240=bleu et 360 à nouveau rouge.

- la saturation est un pourcentage qui donne la pureté de la couleur choisie , du gris à 0% à la couleur vive 100%.

- La luminosité est également en %, où 0% correspond au noir et 100% au blanc.

| **Propriétés** | **Rôle** | **Hériatge** | **Exemples** |
| --- | --- | --- | --- |
| text-shadow | Ombrage du texte | héritée | **h2{**  **text-shadow :2px -2px 2px blue ;}**  Les valeurs :  **none** : aucun ombrage ;  **1ère valeur** : décalage en x de l’ombre (**positif**= vers la droite),  **2ème valeur** : décalage en y de l’ombre ( **positif**=vers le bas)  **3ème valeur** : étendue du flou, aucun flou si cette valeur est absente ou =à0  -**couleur** : couleur de l’ombre (code couleur ou mot clè)  Pluseiurs ombres peuvent être définies à la suite dans une même propriété text-shadow, séparées par des virgules, dans ce cas les différents effets d’ombre se superposent. |
| text-outline ? | Contour du texte |  |  |
| box-shadow | Ombrage d’un élément boîte | non | **ul{**  **box-shadow :2px 2px 2px grren ;}**  les valeurs :  **none** : aucun ombrage  **1ère valeu**r : décalage en x de l’ombre ;  **2ème valeur** : décalage en y de l’ombre ;  **3ème valeur** : étendue du flou ;  **4ème valeur** : étendue de l’ombre qui définit l’épaisseur d’ombre de couleur unie avant l’application du flou, égale à 0 si absente. |
| opacity | Opacité | non | **div.pub{**  **background-color :red ; opacity=0.5}**  **div{**  **background-color : rgba(183,232,255,0.7) ;}** |

**III-4 Les propriétés des bordures**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| border | (déclaration groupée :raccourci pour  **border-width,border-style et border-color)** |  |  |
| border-color | Couleur du bord | non | **div{border-color :gray ;}**  valeurs : **nom de la couleur ou code RVB** |
| border-spacing  (pour les tables) | Espacement du bord | héritée | **table {border-spacing : 5px ;}**  un ou deux nombres de même unité que les tailles de police sauf %  **Une valeur** : espacement pour toutes les bordures ;  **Deux valeurs** : espacement horizontal et vertical |
| border-collapse  (les tables) | Fusion du bord | héritée | **table{**  **border-collapse :collapse ;}**  les valeurs :  **separate** : séparation des bordures ; |
| border-style | Style du bord | non | **h2{border-style :solid}**  Les valeurs :  **none** : valeur par défaut ;  **hidden** : aucune bordure ;  **solid** : trait plein ;  **dotted** : pointillés,  **dashed** : tirets,  **double** : trait plein double, **groove** : en creux,  **ridge** : en relief,  **inset** : creux ombré,  **outset** : relief ombré,  **dot-dash** :alterne points et tirets,  **dot-dot-dash** : affiche deux points puis un tiret,  **wave**: crée une vague sous forme d’un trait ondulé. |
| border-width | Largeur de la bordure | non | **p.note{border-width : 2px ;}**  Les valeurs :  **thin** : bordure fine,  **medium** : moyenne,  **thick** : bordure épaisse  ou valeur numérique en em, px (sauf %). |
| border-radius | Rayon du bord arrondi | non | **p.annonce{**  **border-radius :20px 30 px ;}**  Les valeurs :  **none** : pas d’arrondi ;  soit une ou deux valeurs de longueur : rayon vertical et rayon horizontal.  Si une valeur est donnée alors ces deux rayons sont égaux. |
| border-image | Style de bord avec image | non | **h1{**  **border-image :url(fleur.gif) 35% 35% 35% round stretch ;}**  Les valeurs :  - none ou  1) **url**  : fichier contenant l’image  2) Pour chaque partie du cadre (**haut, droite, bas, gauche**), la partie correpondante du cadre image à utiliser. La dimension est en pixel ou %.  3)**round ou repeat**  4)**stretch.** |
| outline | Contour(raccourci pour **outline-width,outline-style et outline-color**)  Elle affiche une bordure qui se superpose à l’élément sans augmenter ses dimensions |  | **:focus{**  **outline thick solid #fc0 ;}** |
| outline-style | Style du contour | non | Mêmes valeurs que border-style sauf hidden |
| outline-color | Couleur du contour | non | Mêmes valeurs que border-color |
| outline-width | Largeur du contour | non | Mêmes valeurs que border-width |

**III-5 Les propriétés de positionnement et dimensionnemet**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| display | Mode d’affichage : elle permet le changement du type de l’élément par exemple transforme un élément en ligne en bloc (pour le lien)ou l’inverse | non | **span.bloc{display :block) ;}**  **p.note{display :none ;}**  Les valeurs :  **inline** : élément en ligne,  **block** : bloc,  **list-item** : élément de liste,  **inline-block** : élément en ligne remplacé,  **run-in** : bloc ou élément en ligne, suivant le contexte,  **table** :tableau,  **inline-table** :tableau en ligne,  **table-cell** : cellule de tableau,  **table-row** : ligne du tableau,  **table-column** :colonne de tableau. |
| float,clear | Mode flottant  pour clear : oblige le navigateur à afficher un contenu à la ligne, sous le plus bas des éléments flottants | non | **img{float :left ;}**  Les valeurs :  **none**: pas de flottement,  **left** :élément flottant à gauche,  **right** :élément flottant à droite.  **h1{clear :both ;}**  Les valeurs :  **none** : éléments flottants autorisés à droite et à gauche ,  l**eft** :pas d’élément flottant à gauche,  **right** :pas d’élément flottant à droite,  **both** :aucun élément flottant, ni à gauche, ni à droite. |
| position | Positionnement | non | **p.note {**  **position :relative ;left :-5px ;}**  Les valeurs :  **static** : positionnement dans le flux normal,  **relative** : décalage par rapport à la position du flux,  **absolute** : positionnement par rapport au bloc conteneur,  f**ixed**: positionnement par rapport à l’écran. |
| z-index | Ordre de recouvrement « vertical ».  Lorsque plusieurs blocs sont superposés, ils s’entassent suivant l’ordre de leur arrivée dans le code HTML. Cet ordre peut être modifié en utilisant cette propriété | non | **#logo{z-index :-5 ;}**  Les valeurs :  **auto** : même niveau d’empilement que la boîte parente ou  nombre entier positif, nul ou négatif, |
| overflow | Mode de dépassement de bloc | non | **p{overflow :hoidden ;}**  Les valeurs :  **visible** :le débordement est visible,  **hidden** :le débordement est masqué,  **scroll** : affichage dans tous les cas d’une barre de défilement qui permet d’accéder à un débordement éventuel,  **auto** :une barre de défilement apparaît mais seulement en cas de débordement. |
| overflow-y | Mode de dépassement de bloc vertical |  |  |
| overflow-x | Mode de dépassement de bloc horizontal |  |  |
| width | Largeur | non | **div{width :300px ;}**  Les valeurs :  **auto**,  taille relative en em,ex, px ou % |
| height | Hauteur | non | **div{height :50% ;}**  Les valeurs :  % de la hauteur du bloc conteneur si celle-ci est fixée, sinon c’est la valeur **auto** qui est appliquée. |
| top | position par rapport au haut | non | **p.note{**  **position :relative ; top :5px ;left :10px ;}**  Les valeurs  **none** : pas de décalage  **valeur relative**:en em, ex, px, % |
| left | positioin par rapport à la gauche | non | idem |
| right | Position par rapport à la droite | non | idem |
| bottom | Position par rapport au bas | non | idem |
| padding | Marge interne | non | **p{padding :5ex ;}**  Les valeurs :  **Une valeur** : définit toutes les marges intérieures  **Deux valeurs** :  1ère-marges du haut et du bas égales,  2ème- marges de gauche et de droite égales,  **trois valeurs** :  1ère-marge du haut,  2ème-marges de gauche et droite égales,  3ème-marge du bas,  **quatre valeurs** :  1ère- marge du haut,  2ème-marge de droite,  3ème- matge du bas,  4ème- marge de gauche  Les unités sont :em,px, ex et % |
| margin | Marge externe | non | **p{margin :3em 5em ;}**  Les valeurs :  **Une valeur**: définit toutes les marges extérieures  **Deux valeurs** :  1ère-marges du haut et du bas égales,  2ème- marges de gauche et de droite égales,  **trois valeurs** :  1ère-marge du haut,  2ème-marges de gauche et droite égales,  3ème-marge du bas,  **quatre valeurs** :  1ère- marge du haut,  2ème-marge de droite,  3ème- marge du bas,  4ème- marge de gauche |
| min-height | Hauteur minimale |  | **h1{min-widtrh :50% ;}**  les valeurs :  **none** =0 ,dimension (nombre positif) :  **taille relative** en em, ex,%,px  **taille fixe**en pt,pc,cm,mm |
| min-width | Largeur minimale |  | idem |
| max-height | Hauteur maximale |  | idem |
| max-width | Largeur maximale |  | idem |
| vertical-align | Alignement vertical |  | **.exposant{**  **vertical-align :super ;}**  Les valeurs :  **baseline** :sur la base de la ligne,  **sub** :indice,  **super** :exposant,  **middle** :au milieu de la ligne (centrage vertical sur la ligne),  **text-top ou text-bottom**  :alignement avec le haut ou le bas de la boîte parente,  **valeur ou pourcentage** : valeur positive pour un décalage vers le haut, négative un décalage vers le bas. |
| visibility | Visibilité | non | **.note{visibility :hidden}**  Les valeurs  :  **visible** : l’élément est visible,  **collapse** : hidden, sauf dans les tableaux, où l’espace est libéré lorsqu’il s’agit d’une ligne entière ou d’une colonne entière. |
| rotation | Rotation | non | **img.illustration{**  **rotation :10deg ;}** |
| rotation-point | Point de rotation | non | **img.illustration{**  **rotatioin-point :bottom left ;}** |

*Remarque*

1. Position normale : Les blocs sont affichés les uns sous les autres, les éléments en ligne sont placés côte à côte
2. Position relative : Pour décaler ou superposer un élément par rapport à ses frères.
3. Position absolue :

Elle sert à

* découper la page en zones, sans utiliser des tableaux
* pour disposer à un endroit précis un encadré, une image,etc...

**III-6 Les propriétés des listes**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| list-style | (Déclaration groupée c-à-d raccourcie pour  **list-style-type, list-style-image et list-style-position**) | héritée |  |
| list-style-type | Type de puce pour les éléments de la liste | héritée | **ul{**  **list-style-type :square ;}**  **ol{**  **list-style-type :upper-roman ;}**  Les valeurs :  listes à puces :**disc** (cercle plein-valeur par défaut),  **circle** (cercle vide),  **square**(carré plein).  liste numérotée :  **decimal**(1,2...valeur par défaut),  **decimal-leading-zero**(01,02,....),  **lower-roman**(i,ii,...),  **upper-roman**(I,II,...)  **georgian**,**armenian**,  **lower-lati**n=**lower-alpha**(a,b,c,...)  **upper-latin**=**upper-alpha**(A,B,C,...)  **lower-greek**(α,γ,β,..)  **none** : pas de puce ni numéro |
| list-style-image | Image de puce pour les éléments de la liste | héritée | **ul{**  **list-style-image :url(image/puce.gif) ;}**  **li{**  **list-style-image :url(http ://www.photes.org/logo.png) ;}**  Les valeurs :  **url** (nom d’image avec chemin relatif ou absolu)  **none** :aucune image (valeur par défaut) |
| list-style-position | Position de la puce | héritée | **ul{list-style-position :outside ;}**  **ol{list-style-position :inside ;}**  Les valeurs :  **outside**:la puce est dans la marge (valeur par défaut)  **inside** :la puce fait partie de la première ligne du paragraphe. |

**III-7 Les propriétés d’animation**

Une règle **@keyframes**  permet de créer en CSS3 des animations qui s’afficheront lors du chargement de la page en leur attribuant un nom qui permet de les utiliser dans la feuille de style avec la propriété « animation ».

**Régle@keyframes**

Pour créer une nouvelle animation, il suffit de définir un nom et des styles qui au minimum correspondront à deux états , le départ et l’arrivée

**Exemple**

**@keyframes descente{**

**from{margin-top :50px ;}**

**to{margin-top :200px ;}**

Le nom attribué à l’animation est ici « **descente** ». La situation de départ est une marge de haut de 50 px, qui à l’arrivée est de 200 px.

**Un exemple d’utilisation**

**div#titre{animation :descente 3s ;}** (forme simple)

l’animation définie par @keyframes ne se limite pas à cette (forme simple) :

1. elle peut utiliser plusieurs propriétés, chacune ayant des valeurs différentes au départ et à l’arrivée ;
2. il est possible de définir des étapes intermédiaires, en les repérant par des pourcentages du temps écoulé

**Exemple**

**@keyframes apparition{**

**0% {width :0 ;height :0 ; opacity :0 ;}**

**20% {width :50px ;height :70 ; opacity :0.2 ;}**

**50% {width :100px ;height :140px ; opacity :0.3 ;}**

**80% {width :220px ;height :310px;opacity :0.5 ;}**

**100% {width :250px ;height :350px ; opacity :1 ;}**

**img.logo{animation :apparition 5s linear ;}**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| animation | (déclaration groupée c-à-d raccourci pour **animation-name,animation-duration,animation-timing-function,animation-delay,animation-iteration-count, animation-direction et animation-fill-mode** |  |  |
| animation-delay | Délai d’animation | non | **div#bandeau{**  **animation-delay :2.5s ;}**  **0**:pas de retard de démarrage (valeur par défaut),  un délai indiqué en secondes. |
| animation-direction | Sens d’animation | non | **img.logo{animation-direction :normal ;}**  Les valeurs  **normal** (valeur par défaut): si l’animation doit toujours recommencer à partir de sa forme initiale,  **reverse** : une animation dans le sens inverse par rapport à celui défini initialement par la règle @keyframes,  **alternate** :si les itérations paires doivent s’effectuer dans le sens contraire de celui défini par la règle keyframes,  **alternate-reverse**: si ce sont les itérations impaires qui doivent s’effectuer dans le sens contraire de celui défini par la règle keyframes. |
| animation-duration | Durée d’animation | non | **img.logo{**  **animation-duration :5s ;}**  Les valeurs :  **0** : aucune animation (valeur par défaut) ou une durée en secondes suivie de la lettre « s ». |
| **animation-iteration-count** | Nombre d’itérations | non | **img.log{**  **animation-iteration-count :3 ;}**  Les valeurs :  **1** (valeur par défaut) :l’animation ne se produit qu’une seule fois,  Un nombre quelconque d’itérations,  **infinite** :pour une répétition permanente. |
| **animation-name** | Nom d’animation | non | **img.logo{animation-name :apparition ;}**  Les valeurs :  **none** :aucune animation (valeur par défaut) ou le nom d’une animation qui a été définie par @keyframes  *Rmq* : plusieurs noms d’animation peuvent être indiquées séparés par des virgules. |
| animation-fill-mode | Modifie les états initial et final de l’élément animé | non | **img.logo{animation-fill-mode :both ;}**  Les valeurs :  **none**(valeur par défaut) :avant et après l’animation, l’élément animé conserve des propriétés de la feuille de style, si elles sont en contradiction avec celles de l’animation,  **forwards** :après l’animation, l’élément conserve les propriétés finales qui lui ont été attribuées par la règle@keyframes, dans l’étape to{...} ou 100%,  **backwards** :avant le début de l’animation (si un délai d’attente a été défini), l’objet prend les propriétés définies pour l’instant initial dans la règle @keyframes, à l’étape from{...} ou 0%.,  **both** :l’élément utilise toujours les propriétés initiales et finales qui lui ont été attribuées par la règle @keyframes, même si une règle de style les contredit. |
| animation-play-state | Mettre en pause l’animation, par exemple à l’aide d’un bouton auquel on a affecté une fonction javascript | non | i**mg.log{animation-play-state :running ;}**  Les valeurs :  **running** (valeur par défaut) :si l’animation doit continuer à s’exécuter,  **paused** :pour interrompre momentanément l’animation. |
| animation-timing-function | Fonction d’accélération | non | **img.logo{animation-timing-function :linear ;}**  Les valeurs :  **linear** :linéaire (vitesse contante),  **ease** (valeur par défaut, démarrage progressif mais rapide avec un ralentissement à la fin),  **ease-in** :lent au début (rapide ensuite),  **ease-out** : (rapide dès le début (ralenti à la fin),  **ease-in-out** :démarrage et fin en douceur (vitesse plus rapide entre les deux),  **cubic-bezier(x1,y1,x2,y2)** :paramétrage de la courbe à partir des coordonnées de deux points intermédiairs,  **step-start ou step-end** :pas de transition , valeur finale attribuée instantanément au début ou à la fin de la durée de la transition,  **steps(n,start)** ou **steps(n,end)** :transition par n sauts successifs, avec start le premier saut dès le début de la transition et pour end le dernier saut à la fin de la transition, |

**III-8 Les propriétés de transformations**

La norme CSS3 propose des transformations en deux et trois dimensions. Elles s’appliquent aux blocs comme aux éléments en ligne.Comme transformations, il y a : déplacements, changements d’échelle, rotations et déformations :

| **Propriétés** | **Rôle** | **Hériatge** | **Exemples** |
| --- | --- | --- | --- |
| transform | Transformations | non | **img.logo{**  **transform :rotate(45deg) ;}**  La propriété transform utilise une palette de fonctions, parmi celles-ci, il y a :  **1**- tarnsformation **2D** :  **1-1** Transformations sous forme de décalage :  - **matrix(a,b,c,d,e,f)** :transformation matricielle à partir des six nombres donnés  - **translateX(nombre)**:décalage horizontal de la valeur fournie  - **translateY(nombre)**:décalage vertical de la valeur fournie  -**translate(x,y)** : décalage horizontal de x et vertical de y  **1-2** Pour des changements d’échelle, il y a :  - **scaleX(nombre)** :changement d’échelle horizontale,  -**scaleY(nombre)** :changement de l’échelle verticale,  -**scale(x,y)** :changement de l’échelle verticale et horizontale,  **1-3** Pour une rotation ou déformation, il y a :  -**rotate(angle)** :rotation de l’angle indiqué dans le sens des aiguilles d’une montre pour un angle positif,  -**skewX(angle)** :déformation le long de l’axe horizontal,  -**skewY(angle)** :déformation le long de l’axe vertical,  **skew(angle1,angle2)** :déformation le long de l’axe horizontal et vertical  **2-** Transformations **3D**  **2-1** Pour le décalage :  **matrix(a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p)**  -**translateX(nombre)**  -**tranlateY(nombre)**  -**tranlateZ(nombre)** :décalage en profondeur de la valeur fournie  -**translate3d(x,y,z)**  **2-2** changement d’échelle :  -**scaleX(nombre)**  -**sacleY(nombre)**  **scaleZ(nombre)** :changement d’échelle dans le sens de la profondeur  -**scale3d(x,y,z)**  **2-3** Rotations et perspective  -**rotateX(angle)**  -**rotateY(angle)**  -**rotateZ(angle)**  -**rotate3d(x,y,z,angle)**  -**perspective(profondeur)** :place en perspective le contenu du bloc concerné, suivant la profondeur indiquée, celle-ci doit être positive ou nulle et contenir une unité de mesure, par exemple  :500px, la valeur 0 indique l’absence de perspective. |
| transform-origin | Point d’origine de la transformatioin | non | **img.logo {**  **transform-origin :0 0 ;}**  Les valeurs :  deux valeurs correspondent au couple de coordonnées définissant l’origine géométrique de la transformation.  chaque valeur peut être :  -soit un nombre, pourcentage ou longueur(unité de dimension)  -soit des mots clès, combinant position horizontale(left,center,right) et position verticale(top, center,bottom).  Pour les transformations en 3D, trois valeurs numériques peuvent être données correspondantes aux coordonnées dans l’respace. |
| transform-style | Style de transformation( pour 3D) |  | **transform** :valeur précise pour un élément qui a été transformé, si les éléments qu’il contient doivent être aplatis avant la transformation 3D (valeur par défaut ,flat) ou si ce contenu doit rester en 3 dimensions |
| perspective | Effet de perspective |  | **perspective** :distance  place en perspective, avec la profondeur indiquée, les éléments contenus dans le bloc concerné, mais uniquement s’ils ont été positionnés ou transformés. Valeur par défaut :none |
| perspective-origin | Point d’origine de la perspective |  | **perspective-origin** :x y  définit les coordonnées de l’origine de la perspective, à partir des valeurs exprimées en pixel, pourcentage ou mot clè(left,center ou right pour x, top,center ou bottom pour y) la valeur par défaut est 50% 50% (équivalent de center center). |
| backface-visibility | Précise si la face arrière d’un objet doit être visible (valeur par défaut, visible) ou cachée (valeur hidden) |  |  |

**III-9 Les propriétés des transitiions**

Avant les animations de nos pages étaient créées soit en javascript, par exemple à l’aide des bibliothèques jQuery et Moo Tools, soit en flash, en utilisant un logiciel externe pour toute modification (Flash, Swish, E-anim,etc..)

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| transition | (Déclaration groupée c-à-d raccourci pour **transition-property, transitiion-duration, transition-timing-function et transition-delay** ) |  |  |
| transition-delay | Délai de la transition | non | **a.menu{transition-delay :0 ;}**  Les valeurs :  (valeur par défaut, pas de retard au démarrage)  un délai indiqué en secondes et suivi par le symbole « **s** » sans espace avant. |
| **transition-duration** | Durée de la transition | non | **a.menu{**  **transition-duration :1.5s ;}**  les valeurs :  **0** (valeur par défaut) :pas de transition,  une durée indiquée en secondes. |
| transition-property | Propriété à laquelle on applique la transition | non | **a.menu{**  **transition-property :all ;}**  les valeurs :  **none**:aucune propriété concernée par les transitions  **all**(valeur par défaut- toutes les propriétés sont concernées) un ou plusieurs noms de propriétés séparés par des virgules. |
| transition-timing-function | Fonction d’accélération pour la transition | non | **a.menu{**  **transition-timing-function :ease-out ;}**  Les valeurs :  **linear** :linéaire (vitesse contante),  **ease** (valeur par défaut, démarrage progressif mais rapide avec un ralentissement à la fin),  **ease-in** :lent au début (rapide ensuite),  **ease-out** : (rapide dès le début (ralenti à la fin),  **ease-in-out** :démarrage et fin en douceur (vitesse plus rapide entre les deux),  **cubic-bezier(x1,y1,x2,y2)**:paramétrage de la courbe à partir des coordonnées de deux points intermédiairs,  **step-start ou step-end** :pas de transition , valeur finale attribuée instantanément au début ou à la fin de la durée de la transition,  **steps(n,start)** ou **steps(n,end)** :transitions par n sauts successifs, avec start le premier saut dès le début de la transition et pour end le dernier saut à la fin de la transition, |

**III-10 Autres**

| **Propriétés** | **Rôle** | **Héritage** | **Exemples** |
| --- | --- | --- | --- |
| **cursor** | Apparence du curseur | héritée | **.aide{cursor :help ;}**  Les valeurs :  **auto** (valeur par défaut) : la forme est fonction du contexte,  **défault** :généralement une flèche blanche  **crosshair** :croix noire,  **pointer** :main,  **move**:quadruple-flèche de déplacement,  et double flèches de redimensionnement orientées (nord n-sud s-est-e-ouest-w) :**n-resize,s-resize,e-resize,,w-resize,ne-resize,sw-resize,nw-resize,se-resize**. |
| direction | Direction d’écriture | non | **body{direction :ltr ;}**  Les valeurs :  **ltr** :de gauche à droite (left to right) valeur par défaut,  **rtl** :de droite à gauche (right to left). |

**III-11 Multicolonnage**

columns

column-count

column-width

column-gap

column-rule

column-rule-style

column-rule-width

column-rule-color

column-span

column-fill

**III-12 Les propriétés pour Média paginé**

page-break-before

page-break-after

page-break-inside

size

page

orphans

widows

image-orientation

fit

fit-position

**III-13 Les propriétés pour Média sonore**

voice-volume

voice-balance

speak

speak as

voice-family

voice-stress

voice-duration

cue

cue-before

cue-after

pause

pause-before

pause-after

rest

rest-before

rest-after

**IV- Pseudo-classes et pseudo-éléments**

| **Pseudo** | **Rôle** |
| --- | --- |
| :link | Lien |
| :visited | Lien visité |
| :hover | Elément survolé |
| :active | Elément actif |
| :focus | Elément ayant le focus |
| :first-child | Premier enfant |
| :last-child | Dernier enfant |
| :nth-child(n) | n-ième enfant |
| nth-last-of-child(n) | n-ième dernier enfant |
| nth-of-type(n) | n-ième type d’élément |
| :nth-last-of-type(n) | n-ième dernier type d’élément |
| :first-of-type | Premier type d’enfant |
| :last-of-type | Dernier type d’enfant |
| :only-child | Enfant unique |
| only-of-type | Elément de type unique |
| :checked | Etat coché |
| :enabled | Etat activé |
| :indeterminate | Etat indéterminé |
| :not(expr) | Négation de l’expression |
| :first-letter | Première lettre |
| :first-line | Première ligne |

**V- Règles**

| **Propriétés** | **Rôle** |
| --- | --- |
| @import | Importer une autre feuille de style |
| @charset | Déclaration de l’encodage des caractères |
| @page | Définir des règles générales pour les médias paginés |
| @font-face | Importer un fichier de polices |
| @media | Définir des requêtes média |

**Exemple**

@import « autresstyle.css » ;

@page {

size :15 cm 20cm ;

margin :2cm ;

marks :cross ;

}

@font-face{

font-family :mapolice ;

src : »ToutsaufComic.otf » ;

font-weight :bold ;

}

p{

font-family :mapolice ;

}

@media print{

body{

font-size :2em ;

background :white ;

}

nav{

display :none ;

}

}

**JAVASCRIPT**

**I Rappels sur Javascript**

Le véritable nom de javascrip est ECMAscript dasn le cadre de la spécification ECMA-262.

Javascript est un langage développé par Netscape en 1995, dérivé d’un projet de Brendan Eich, et ajouté à Netscape Navigator, puis par la suite aux autres navigateurs..

Ce langage interprété par l’agent utilisateur (navigateur) doit permettre d’ajouter de l’intéraction avec l’utilisateur au lieu d’un simple chargement de page statique.

Il doit permettre aussi au travers du DOM (Document Object Model) de manipuler les éléments HTML présents dans la page : les créer dynamiquement, les modifier, les supprimer, agir sur leurs propriétés CSS, etc..

C’est aussi Javascriot qui est mis en oeuvre au sein de la technologie Ajax (Asynchronous Javascrip and XML) pour effectuer des chargements de contenu dynamique en relation avec le serveur, ou encore JSON(Javascript Object Notation) en tant que format de données.

**II- Frameworks Javascrip**

Un « framework » Javascript est un ensemble de fonctions (écrites elles-même en javascript), regroupées dans un fichier externe à inclure dans une page HTML grâce à la balise <script>. Un framework définit un environnement permettant d’étendre les possibilités déjà offertes par la base du langage.

JQuery (Licence GPL/MIT) est un framework très connu. Il comprend des fonctions de modification dynamique du DOM, des styles CSS, des effets d’animation, et des fonctions pour l’écriture de gestionnaires d’évènements ou d’appels Ajax.

Il y a d’autres frameworks comme : MooTools, Dojo Toolkit, Scriptaculous

**III- Le placement du script**

<head>

<**script src= "test.js"**></script>

...........

</head>

OU avant <**/body**> ou dans le contenu de la page.

**IV- Les bases de Javascript**

Dans la majorité des cas, on utilise Javascript dans une page Web pour manipuler le DOM.

C’est quoi le DOM ?

Le DOM est une interface de programmation pour tous les documents et pages Web ou XML. Il s’agit d’une structure représentant ces documents tels qu’ils ont été interprétés en mémoire, permettant de parcourir et modifier leur contenu, ainsi que leur présentation visuelle si des règles CSS sont appliquées.

Au chargement d’un document HTML, un arbre de la page est élaboré via le DOM. Chacun des éléments est considéré comme un objet et peut jouer le rôle de noeud, auquel on peut attacher les qualités de parent, enfant,orphelin ou frère. Cette représentation permet d’accéder aux propriétés de chacun de ces objets ou groupe d’objets et aux méthodes qui les équipent.

Les opérations les plus courantes sont l’ajout, la modification et la suppression de noeud.

**Exemple** l’arborescence DOM d’un document HTML est comme suit :

HTML

HEAD

BODY

SCRIPT

HEADER

SECTION

ARTICLE

FOOTER

H1

P

P

texte

AUDIO

texte

*1- Variables*

Comme tout langage de programmation, on retrouve de nombreuses structures connues et la notion de variable.

Avec HTML, on retrouve la déclaration de script dans une balise <script>

<script>

alert("Script") ;

</script>

Toute instruction est séparée par « ; ».

Une variable est déclarée par le mot-clé « **var** » suivi de « **=**»

**Exemple**

var a=3 ;

var b=4 ;

alert(a\*b) ;

*Remarque*

Pour profiter au mieux de la découverte de javascipt et surtout de meilleures conditions de développement et débogage, utiliser un navigateur moderne équipé d’une console Javascript et utiliser plutôt la fonction console.log().

**2- Types simples**

Des chaînes de caractères peuvent être déclarées :

var proverbe= "rien ne sert de courir il faut partir à point" ;

et concaténées grâce au signe «**+** »

console.log(proverbe+ "claire fontaine") ;

Des tableaux de données sont initialisés rapidement :

var tableau=Array("Pommes ","ananas", "orange") ;

Chaque élément de tableau est accessible par son index (commence par zero) placé entre crochets «**[]** ».

console.log(tableau[0]) ; //*Affiche Pommes*

console.log(tableau[2]) ; //*Affiche orange*

La longueur d’une chaîne de texte ou d’un tableau peut être connue grâce à sa propriété length.

proverbe.lenght ;//**45**

tableau.lenght ; //**3**

tableau[2].lenght ; //**6**

**3- Objets**

Les objets sont crées avec des paires de clés/valeurs, séparées par des virgules, le tout regroupé entre accolades « { } »

**Exemple**

var voiture={

roues :4,

couleur :"rouge",

achat :{

prix :20000,

année :2011,

}

} ;

Un objet peut contenir un sous-objet, et ainsi de suite. Dans notre cas l’objet « **voiture**» contient trois propriétés (**roues,couleur et achat**) dont la dernière est un autre objet avec deux propriétés (**prix, année**).

Les propriétés des objets sont accessibles en « naviguant » dans l’arborescence de l’objet par des points « . »

voiture.achat.année ; //*Affiche 2011*

Un appel à console.log permet d’obtenir une visualisation de la variable objet sous forme d’arbre.

console.log(voiture) ;

Certains objets sont célèbres dans un navigateur, ils servent d’interface à un ensemble d’informations réunies au chargement du document. Parmi eux, window qui « symbolise » la fenêtre active.

window ; //*interface DOMWindow*

window.document ; //*interface Document*

window.navigator ; // *interface Navigator*

window.navigator.userAgent ; /*/de type String*

En programmation orientée objet, l’objet lui-même est une instance de classe.

var o=new Object() ;

Le mot clé « new » intervient, suivi de l’appel au constructeur (le nom de la classe de l’objet).

var w=new Worker() ;

Il s’agit d’un objet implémentant l’interface Worker.

Tout est objet en Javascript, avec l’exception de « null » et « undefined » qui correspondent respectivement à une valeur nulle(différente de zero), et une valeur indéfinie.

**Exemple**

var s=new String("Rose") ;

s.toString() ; // *Affiche "Rose"*

**4- Fonctions**

Les fonctions servent à regrouper des instructions. Elles sont déclarées avec le mot-clé « **function** » suivi du nom de la fonction et «  **( )** ».

**Exemple**

function ident\_navigateur(){

console.log(window.navigator.userAgent) ;}

cette function est invoquée comme suit :

ident\_navigateur() ; // *Affiche le navigateur*

**Une variante**

ident\_navig=function() {

console.log(window.navigator.userAgent) ;}

Un autre exemple de fonction avec paramètres

function addition(a,b){

return a+b ;

var c=addition(2,3) ;

console.log(c) ;

**5- Boucles**

- **For**

for(var i=0 ;i<10 ;i++){

console.log(i) ; // Affiche 0,1,2,....9}

- **while**

var i=0 ;

while(i<10){

console.log(i) ;

i++ ;}

**V- Méthodes de sélection DOM**

Les fonctions pour la selection de noeuds dans le DOM sont getElementById et getElementsByTagName(depuis DOM Level 2). Elles permettent de cibler un élément par son identifiant, l’attribut id, ou par son nom. Une fois l’objet obtenu par retour de ces fonctions, on peut accéder en lecture ou en modification à ses propriétés (attributs, style,...).

getElmentById()

Renvoie un élément dont on spécifie l’identifiant (attribut "id") ou null sinon

**Exemple**

<h1 id= "titre">HTML5</h1>

<script>

var element=document.getElementById("titre") ;

alert(elment.textcontent) ; //*Affiche "HTML"*

</script>

getElmentsByTagName()

Renvoie un ensemble d’éléments dont le type est spécifié.

var elements=document.getElmentsByTagName("img") ;

La variable elements obtenue est de type nodeList. Chacun de ses éléments est accessible de deux manières :

elements.item(1) ;

elements[1] ;

D’autres fonctions ont été crées

getElementsByClassName()

Renvoie un ensemble d’éléments dont on spécifie la classe.

var elements=document.getElementsByClassName("maclasse ") ;

querySelector() (Nouveau HTML 5)

Retourne le premier élément qui correspond au sélecteur spécifié(CSS)

var el=document.querySelector("#menu ul li.sousmenu") ;

querySelectorAll() (HTML 5)

Retourne tous les éléments qui correspondent au sélecteur spécifié.

var titres=document.querySelectorAll(" article>h1") ;

**VI- Propriétés et Méthodes DOM**

Les éléments DOM renvoyés par les fonctions précédentes sont de type HTMLElement lorqu’on a un élément, et HTMLDOMChildrenCollection lorqu’on obtient une énumération d’un ou plusieurs éléments.

Quelques propriétés ou méthodes de l’interface HTMLElement

| **Propriétés ou Méthodes** | **Description** | **Type** |
| --- | --- | --- |
| innerHTML | Contenu HTML d’un élément | string |
| outerHTML | Contenu HTML et l’élément lui-même | string |
| textContent ou innerText | Contenu Texte simple | string |
| style | Ensemble des propriétés CSS de l’élément | CSSStyleDeclaration |
| style.\* | Propriétés CSS individuelles, accessibles par des membres :  style.width, style.fontSize, style.listStyleType | string |
| tabindex | Index de tabulation | long |
| id | Valeur de l’attribut « **id** » | string |
| lang | Valeur de l’attribut « **lang** » | string |
| title | Valeur de l’attribut « **title** » | string |
| tagName | Type de l’élément (nom de la balise) | string |
| ClassName | Valeur de l’attribut de classe affecté à l’élément | string |
| classList (HTML 5) | Enumération des classes individuelles présentes dans l’attribut « class » | DOMTokenList |
| childrenNodes | Liste des nodes (noeuds) enfants contenus dans l’élément | NodeList |
| firstChild | Premier Node enfant | HTMLElement |
| lastChild | Dernier Node enfant | HTMLElement |
| parentNode | Node parent | HTMLElement |
| nextSibling | Node frère suivant | HTMLElement |
| prevSibling | Node frère précédent | HTMLElement |
| attributes | Attributs de l’élément | NamedNodeMap |
| getAttribute(attr) | Renvoie la valeur de l’attribut «**attr**» |  |
| setAttribute(attr,val) | Modifie la valeur de l’attribut « **attr** » par **val** |  |
| removeAttribute(attr) | Supprime l’attribut « **attr** » s’il existe |  |
| focus() | Donne le focus à l’élément |  |
| blur() | Retire le focus. |  |

**VII-Manipulation DOM**

createElement()

Crée dynamiquement un noeud DOM. Le nom de l’élément HTML est indiqué (sans chevrons).

var nouveauDiv=document.createElement("div") ;

nouveauDiv.innerHTML= "<p> C’est un <b>nouveau </b> paragraphe </p>" ;

Cet élément est crée mais il devient visible qu’après insertion à l’aide d’une des fonctions suivantes

appendChild()

Insère un noeud DOM enfant dans un parent

var paragraphe=document.createElement("p") ;

paragraphe.innerHTML= "je suis<b>un nouveau</b>" ;

document.body.appendChild(paragrahpe) ;

removeChild()

Retire un enfant à son parent.

var paragraphe=document.querySelector(" body>p") ;

document.body.removeChild(paragraphe) ;

insertBefore()

Insère un noeud DOM avant un autre.

var nouveauLi=document.createElement("li") ;

nouveauLi.innerHTML= "je suis <b> nouveau </b> ";

var uneListe=document.getElementByTagName("ul")[0] ;

uneListe.insertBefore(nouveauLi,uneListe.lastchild) ;

createTextNode()

Crée un noeud Texte

var nouveauTexte=document.createTextNode("beau temps") ;

document.querySelector("body").appendChild(nouveauTexte) ;

classList (HTML 5)

cette propriété manipule l’attribut « **class** » avec des méthodes appropriées qui manquaient auparavant et devaient être comblées par des bibliothèques Javascript supplémentaires.

var monElment=document.getElemmentById("résultata") ;

monElement.classList.add("positif") ;

**A- Les propriétés et méthodes de classList**

| **Propriétés ou Méthodes** | **Description** | **Retour** |
| --- | --- | --- |
| length | Nombre de classes appliquées à l’élément | integer |
| add(classe) | Ajoute une classe à l’ensemble |  |
| remove(classe) | Supprime une classe |  |
| toggle(classe) | Ajoute ou supprime une classe, alternativement, selon son état initial |  |
| contains(classe) | Retourne true en cas de présence d’une classe, ou false sinon | boolean |
| toString() | Retourne une chaîne de texte contenant toutes les classes appliquées | string |

insertAdjacentHTML() (HTML 5)

Cette méthode insert du contenu dans un document HTML. Elle prend comme premier paramètre l’emplacement d’insertion :beforebegin (avant l’élément), afterbegin (dans l’élément au début, avant son premier enfant), beforeend (à la fin de l’élément, après son dernier élément), afterend (après l’élément), et en deuxième paramètre le bout de code HTML à ajouter.

**Exemple**

<ul id= "maliste ">

<li> Premier </li>

<li> Deuxième </li>

</ul>

<script>

var uneListe=document.getElementById("maliste") ;

uneListe.insertAdjacentrHTML("beforeend", "<li> je suis le dernier </li> ") ;

**VIII- Méthodes pour Formulaires**

Il existe des méthodeq propres aux API d’éléments particuliers. Parmi eux , les champs d’entrée de type <**inpu**t>

**VIII-1 Quelques propriéts et méthodes**

| **Propriétés et Méthodes** | **Description** | **Retour** |
| --- | --- | --- |
| value | Retourne la valeur du champ  Peut être modifié (avec=) pour changer la valeur. |  |
| checked | Retourne true si coché, ou false si décoché | boolean |
| files | Liste des fichiers sélectionnés. Renvoie null si ce n’est pas un champs d’entrée de fichiers. | FileList |
| valueAsDate | Peut être modifié (avec=) pour changer la valeur | Date |
| valueAsNumber | Peut être modifié (avec=) pour changer de valeur | Number |
| list | Retourne l’élément <**datalist**> indiqué par l’attribut « **list**» | HTMLElement |
| selectedOption | Retourne l’option de liste sélectionnée | HTMLElement |
| stepUp(n) | Incrémente la valeur de n |  |
| stepDown(n) | Décrémente la valeur de n |  |

**IX- Gestionnaire d’évènements**

Ces gestionnaires sont des fonctions Javascript associées à un évènement utilisateur ou navigateur. Une fois l’évènement déclenché (par exemple un clic, une touche pressée, ou une page chargée), la fonction est exécutée.

**IX-1 Les attributs évènements**

Les attributs gestionnaires jouent un rôle particulier car ils associent un type d’évènement DOM pouvant survenir pour un élément à une fonction de script.

Cela permet de déclencher un bout de code Javascript sur des évènements tels que « lorsque la page a été chargée », ou « lorsque l’utilisateur presse sur une touche ». La plupart d’entre eux concernent :

- les entrées utilisateur (souris,clavier) ;

- la navigation au clavier (focus) ;

- les formulaires (changement de valeur, validation) ;

- la lecture de médias (audio,vidéo) ;

- les intéractions riches (glisser-déposer, molette souris) ;

- le chargement des pages et des ressources.

**Déclenchement d’une fonction au clic**

<input type= "button" value="Cliquez-ici" onclick= "faireQuelquechose() ; ">

<script>

function faireQuelquechose(){

.........

alert("Un clic ") ;

}

</script>

Certains sont nouveaux en HTML 5, surtout ceux qui concernent la lecture des médias (audio,vidéo). Tous débutent par le terme « **on** » suivi du nom de l’évènement DOM auquel ils correspondent.

**IX-2 Les évènements DOM sont :**

| **Evènement** | **Correspondance** | **Catégorie** |
| --- | --- | --- |
| onabort | Chargement annulé de l’utilisateur | Réseau |
| onblur | Perte de focus : l’élément détenait le focus et l’a perdu, car l’utilisateur a navigué vers un autre élément, au clavier ou grâce à son dispositif de pointage | Navigation |
| oncanplay | L’agent utilisateur peut reprendre de l’audio ou la vidéo, mais estime que s’il débute à cet instant, la totalité du média ne pourra être lue au rythme actuel sans avoir à effectuer d’autre(s) pause(s) pour mettre en cache les données | Médias |
| oncanplaythrough | L’agent utilisateur peut reprendre la lecture de l’audio ou de la vidéo en estimant qu’à cet instant la totalité du média pourra être lue au rythme de lecture actuel sans avoir à effectuer d’autre(s) pause(s) pour mettre en cache les données | Médias |
| onchange | L’utilisateur a validé le changement de la valeur d’un élément (de formulaire) | Formulaires |
| onclick | L’utilisateur a cliqué sur l’élément et a relâché le bouton de la souris ou a utilisé un autre dispositif de pointage qui émule cette action | Souris,pointage |
| oncontextmenu | L’utilisateur a déclenché le menu contextuel | Navigation |
| ondblclick | L’utilisateur a double-cliqué sur l’élément ou a utilisé un autre dispositif de pointage qui émule cette action | Souris,pointage |
| ondrag | L’utilisateur est en train de déplacer l’élément | Drag&drop |
| ondragend | L’utilisateur a fini de déplacer l’élément | Drag&drop |
| ondragenter | L’opération de glisser-déposer initiée par l’utilisateur est entrée dans l’élément | Drag&drop |
| ondragleave | L’opération glisser-déposer initiée par l’utilisateur a quitté l’élément | Drag&drop |
| ondragover | L’opération glisser-déposer survole l’élément | Drag&drop |
| ondragstart | Début de glisser-déposer | Drag&drop |
| ondrop | Fin de glisser-déposer | Drag&drop |
| ondurationchange | L’attribut DOM «**duration** » sur l’élément <audio> ou <video> a été modifié | Médias |
| onemptied | L’élément <audio> ou <video> a été modifié | Médias |
| onended | La fin de l’audio ou de la vidéo a été atteinte | Médias |
| onerror | Le chargement de l’élément a échoué | Réseau |
| onfocus | L’élément a reçu le focus | Navigation |
| onformchange | L’utilisateur a validé le changement de valeur d’un élément dans le formulaire dont cet élément dépend | Formulaires |
| onforminput | L’utilisateur a modifié la valeur d’un élément dans le formulaire dont cet élément dépend | Formulaires |
| oninput | L’utilisateur a changé la valeur de l’élément formulaire | Formulaires |
| oninvalid | l’élément de formulaire n’a pas respecté les contraintes de validation | Formulaires |

**Les évènements (suite)**

| **Evènements** | **Correpondance** | **Catégorie** |
| --- | --- | --- |
| onkeydown | L’utilisateur a enfoncé une touche | Clavier |
| onkeypress | L’utilisateur a enfoncé une touche correspondant à un code caractère | Clavier |
| onkeyup | L’utilisateur a relâché une touche | Clavier |
| onload | L’élément a fini son chargement | Réseau |
| onloadeddata | L’agent utilisateur peut rendre le contenu audio ou vidéo à la position de lecture courante, pour la première fois | médias |
| onloadedmetadata | L’agent utilisateur a pu déterminer la durée et les dimensions de l’élément <audio> ou<video> | Médias |
| onloadstart | L’agent utilisateur a commencé à charger des données médias dans l’élément <audio> ou<video> | Médias |
| onmousedown | L’utilisateur a pressé le bouton de la souris ou du dispositif de pointage sur l’élément | Souris,pointage |
| onmousemove | L’utilisateur a effectué un mouvement avec la souris ou le dispositif de pointage | Souris,pointage |
| onmouseout | L’utilisateur a déplacé le curseur de la souris ou du dispositif de pointage hors des limites de l’élément | Souris,pointage |
| onmouseover | L’utilisateur a déplacé le curseur de la souris ou du dispositif de pointage à l’intérieur des limites de l’élément, ou d’un de ses descendants | Souris,pointage |
| onmouseup | L’utilisateur a relâché le bouton de la souris ou du dispositif de pointage sur l’élément | Souris,pointage |
| onmousewheel | L’utilisateur a actionné la molette de défilement de la souris ou du dispositif de pointage équivalent | Souris,pointage |
| onpause | L’utilisateur a mis en pause la lecture de l’élément <audio> ou<video> | Médias |
| onplay | L’agent utilisateur a débuté la lecture de l’élément <audio> ou<vidéo> | Médias |
| onplaying | La lecture de l’élément <audio> ou<video> a débuté | Médias |
| onprogress | L’agent utilisateur télécharge les données médias pour l’élément <audio> ou <vidéo> | Médias |
| onratechange | L’attribut DOM « **defaultPlaybackRate** » ou »**playbackRate** » a été modifié sur l’élément <audio> ou <vidéo> | Médias |
| onreadystatechange | L’élément et toutes ses ressources ont fini leur chargement | Médias |
| onreset | Le formulaire a été remis à zéro | Formulaires |
| onscroll | L’élément qui a fait l’objet d’un défilement | Souris,pointage |

**Les évènements (Suite)**

| **Evènements** | **Correspondance** | **Catégorie** |
| --- | --- | --- |
| onseeked | La valeur de l’attribut « **seeking** » a été modifiée à false (l’opération de navigation temporelle sur l’audio ou la vidéo a pris fin) | Médias |
| onseeking | La valeur de l’attribut «**seeking** » a été modifiée à true (l’opération de navigation temporelle sur l’audio ou la vidéo a eu lieu et est d’une durée suffisamment pertinente pour déclencher l’évènement | Médias |
| onselect | L’utilisateur a sélectionné du texte | Souris,pointage |
| onshow | L’utilisateur a requis l’affichage de l’élément comme menu contextuel | Navigation |
| onstalled | L’agent utilisateur essaie de télécharger les données pour la lecture de l’élément média <audio> ou <video>, mais celles-ci ne sont pas reçues | Médias |
| onsubmit | Le formulaire a été validé | Formulaires |
| onsuspend | L’agent utilisateur ne télécharge actuellemnt pas le contenu du média, mais ne dispose pas encore de toutes les données | Médias |
| ontimeupdate | La position de lecture de l’élément <audio> ou<video> a changé | Médias |
| onvolumechange | L’attribut DOM « **volume**» ou « **muted** » a été modifié sur un élément média | Médias |
| onwaiting | La lecture de l’élément <audio> ou<video> a été mise en pause, car les données pour la suite sont attendues | Médias |