TECHNOLOGIES WEB
Partie -1-

Polycopié de cours et de travaux pratiques

Appendre les bases du HTML et du CSS

Créer votre premier site Web

Réalisé par :

Dr. BENKADDOUR F.Z, Maitre de conférences « A » à l'ENS d'Oran Ammour-Ahmed

Année universitaire : 2022-2023

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Oran's Higher Teachers College
AMMOUR Ahmed
المدرسة العليا للأساتذة بوهران
عمور احمد

Département des sciences exactes

Spécialité : Informatique

Polycopié de cours et de travaux pratiques destiné aux :

Étudiants de la 3ème année PEM et PES informatique ainsi qu'aux étudiants de Licence informatique

TECHNOLOGIES WEB Partie 01

Proposé par :

Dr. BENKADDOUR Fatima Zohra, Maitre de conférences « A ».

Année universitaire 2022-2023

À mes étudiants...

Table des matières

Table des figures

Liste des tableaux

Résumé

Ce document propose une introduction aux technologies Web accessible aux étudiants de la troisième année PES (Professeur d'Enseignement Secondaire) et PEM (Professeur d'Enseignement Moyen) à l'ENS d'Oran Ammour-Ahmed ainsi qu'aux étudiants de Licence Informatique.
Il aborde les concepts de base du développement Web. Les langages abordés sont le HTML et le CSS.
Ce document propose une série d'exercices de travaux pratiques qui conduisent l'étudiant à une connaissance approfondie des notions de base de la programmation Web.

Mots clés Technologies Web, HTML, CSS, PES, PEM, Informatique.

Avant-propos

Ce document est le résultat d'un cours dispensé aux étudiants de la troisième (3) année PES[1] et PEM[2] informatique et des Licence informatique pendant plusieurs années. Il présente les éléments de base du développement Web. Les langages applicatifs sous-jacent sont le HTML et le CSS. De ce fait, il constitue une introduction à ces deux langages.

Ce manuscrit s'adresse aux débutants en programmation, qu'ils soient dans les premiers cycles universitaires ou toute personne désireuse d'apprendre à programmer une application Web.

Le document est organisé en deux parties comme suit :
La première partie nommée « Cours » est composée de :

- Un premier chapitre intitulé *« Concepts fondamentaux »*, où l'étudiant aura un aperçu sur toutes les notions importantes sur les technologies du Web.
- Un deuxième chapitre *« Premiers pas en $HTML$ »*, qui permet de maitriser les fondamentaux du langage $HTML$ et de concevoir une page Web minimale.
- Un troisième chapitre *« Cascading Style Sheets (CSS) »*, qui permet de compléter langage $HTML$ en gérant l'apparence des pages Web.

La deuxième partie « Travaux pratiques et examens types » donne un ensemble d'exercices pratiques vus durant les séances de TPs. Aussi, elle regroupe des examens types.

Les corrections erreurs ou suggestions sont acceptées, avec gratitude, à l'adresse suivante : `benkaddour.fatima@ens-oran.dz`.

1. Professeur d'Enseignement Secondaire
2. Professeur d'Enseignement Moyen

Fiche matière

Public cible

Ce cours est destiné aux :
— Étudiants de la troisième année PES et PEM informatique, département des sciences exactes à l'ENS d'Oran ammour-Ahmed.
— Étudiants de Licence informatique, département d'informatique de la faculté des sciences exactes et appliquées.

Pré requis

— Notions générales en informatique.

Objectifs

Les objectifs de ce cours sont :
— Savoir créer, modifier et interpréter le code source d'une page en html ;
— Savoir associer une feuille CSS à une page HTML et savoir écrire une règle CSS ;
— Savoir définir des styles simples pour ajuster la police, les marges ou les bordures ;
— Savoir utiliser les éléments HTML et leur associer des styles CSS ;
— Savoir développer des applications Web riches et ergonomiques.

Fonctionnement du cours

Tout le matériel du cours (polycopié, sujets et TP et de projet) peut-être envoyé par mail.
En cas de question/problème, vous pouvez m'envoyer un mail aux : `benkaddour.fatima@ens-oran.dz` ou sur `benkaddourfatima@gmail.com`.

Environnement informatique

Les séances de TPs ont lieu dans la salle informatique de l'ENS d'Oran Ammour-Ahmed sous Windows.

Volume horaire

Ce programme est annuel, il est constitué de :
— Une séance de **cours** de 1h30 par semaine ;
— Une séance de **TPs** de 1h30 par semaine.

Coefficient

Cette matière est de coefficient **3**.

Assiduité et notation

La présence aux TPs est **obligatoire**.
Toute absence ou un retard, il est important de le justifier auprès du secrétariat.
La note finale se compose des notes suivantes :
— Note de contrôle continu (Test noté) ;
— Un projet de TP noté ;
— Notes des EMD 1 et 2.

Première partie:

Cours

Cette partie présente les concepts de base nécessaires à la programmation Web ainsi que les cours des langages $HTML$ et CSS.

1

Concepts fondamentaux

Le Web est sans nul doute une technologie majeure du 21 ème siècle. Si sa nature, sa structure et son utilisation ont évolué au cours du temps, force est de constater que cette évolution a également profondément modifié nos pratiques commerciales et sociales.

À travers ce chapitre, nous verrons les notions les plus importantes des réseaux informatiques, du Web ainsi que son évolution.

1.1 Les Réseaux

Les réseaux informatiques ont été créés à la fin des années 1950 à des fins militaires et défensives. Ils étaient initialement utilisés pour transmettre des données sur les lignes téléphoniques et avaient des applications commerciales et scientifiques limitées. Avec l'avènement des technologies Internet, un réseau informatique est devenu indispensable aux entreprises.

Les solutions réseau modernes apportent plus que la simple connectivité. Elles sont essentielles à la transformation numérique et au succès des entreprises contemporaines. Les capacités sous-jacentes du réseau sont devenues plus programmables, automatisées et sécurisées.

1.1.1 Définitions

— Dans les technologies de l'information, un réseau est défini par la mise en relation d'au moins deux systèmes informatiques au moyen d'un câble ou sans fil, par liaison radio.
— Un réseau est un groupement de deux ou plusieurs ordinateurs ou autres appareils électroniques permettant l'échange de données et le

partage de ressources communes.

— Le réseau informatique désigne les appareils informatiques inter connectés qui peuvent échanger des données et partager des ressources entre eux. Ces appareils en réseau utilisent un système de règles, appelées protocoles de communication, pour transmettre des informations sur des technologies physiques ou sans fil.

1.1.2 Type de réseaux

Nous distinguons différents types de réseaux selon leur taille (en terme de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue. Il existe généralement trois catégories de réseaux :

1. **LAN** (Local Area Network) : C'est un réseau informatique à une échelle géographique relativement restreinte, il est utilisé pour relier entre eux les ordinateurs.
   Autre **VLAN** (Virtual Local Area Network ou Virtual LAN, en français « Réseau Local Virtuel ») est un réseau local regroupant un ensemble de machines de façon logique et non physique.
   Ainsi dans un réseau local la communication entre les différentes machines est normalement régie par l'architecture physique. Grâce aux réseaux virtuels (VLANs) il est possible de s'affranchir des limitations de l'architecture physique (contraintes géographiques, contraintes d'adressage, ...) en définissant une segmentation logique (logicielle) basée sur un regroupement de machines grâce à des critères (adresses MAC, numéros de port, protocole, etc.).
2. **MAN** (Metropolitan Area Network) : inter connectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants.
   C'est un réseau métropolitain qui désigne un réseau composé d'ordinateurs habituellement utilisés dans les campus ou dans les villes. Ainsi, un MAN permet à deux nœuds (ordinateurs) distants de communiquer comme si ils faisaient partie d'un même réseau local. Un MAN est formé de commutateurs ou de routeurs inter connectés par des liens hauts débits qui utilisent généralement des fibres optiques. Ces réseaux peuvent être placés sous une autorité publique ou privée comme le réseau intranet d'une entreprise ou d'une ville. Il permet donc pour une société, une ville, de contrôler elle-même son réseau. Ce contrôle comprend la possibilité de gérer, surveiller et effectuer des diagnostics à distance, à la différence de la connexion WAN, pour laquelle elle doit se fier à son fournisseur d'accès pour gérer et maintenir

la liaison entre elle et son bureau distant.

3. **WAN** (Wide Area Network) : Le réseau Internet (WAN) est un réseau couvrant une grande zone géographique, à l'échelle d'un pays, d'un continent, voire de la planète entière. Il permet l'interconnexion de réseaux locaux et métropolitains vers l'internet mondial. L'infrastructure est en général publique. Le plus grand réseau *WAN* est le réseau internet.

TABLE 1.1 – Types de réseaux informatiques

Type	Porté	Technologies utilisées	Exemples
LAN	Maison, entreprise	Ethernet (sur câbles de paires torsadées), Wifi	habitation particulière, entreprise, salle informatique, bâtiment
MAN	Campus, ville	Fibre optique, ondes radios (Wi-Fi)	intranet d'entreprise ou d'une ville
WAN	Pays, continent, planète	Câble, fibre optique, satellite et technologie sans fil 4G	internet

1.1.3 Domaines d'utilisation

Les réseaux informatiques locaux ou grande distance, publics ou privés, isolés ou inter connectés, véhiculent les informations correspondant aux différents besoins des professionnels et du grand public. Parmi les applications les plus utilisées du grand public.

— **Transfert de fichiers**
Le transfert de fichiers est l'échange de longs documents entre ordinateurs (par opposition au courrier électronique, qui est plutôt destiné aux messages courts). FTP (file transfer protocol) est un protocole de transfert de fichiers aux fonctionnalités réduites mais simples à gérer. Cette application est surtout employée pour le téléchargement de divers fichiers multimédias. Elle peut servir pour la sauvegarde régulière des données de l'entreprise.

— **Messagerie électronique**
Le courrier électronique est par définition une application qui fonctionne en mode non connecté : le courrier est déposé dans une boîte aux lettres que le destinataire vient consulter à loisir. Avoir une boîte aux lettres signifie être capable de transmettre du courrier, de le recevoir et de le stocker avant de pouvoir le consulter.

— **Services d'annuaire électronique et LDAP**
Un annuaire est un ensemble de données auxquelles un client accède pour trouver rapidement des informations concernant des personnes, des machines, des organisations... Un serveur d'annuaire permet de constituer un carnet d'adresses, d'authentifier des utilisateurs grâce à un mot de passe, de définir les droits de chaque utilisateur, ou encore de recenser des informations sur un parc de matériels (ordinateurs, serveurs, adresses IP et adresses MAC...), de décrire l'organisation de l'entreprise, ses services et ses employés.

1.2 Internet

Internet est le réseau informatique mondial qui rend accessibles au public des services comme le courrier électronique et le **World Wide Web**. Ses utilisateurs sont désignés par le néologisme "internaute". Techniquement, Internet se définit comme le réseau public mondial utilisant le protocole de communication IP (Internet Protocol).

1.2.1 Historique

Internet est né il y a une trentaine d'années aux Etats-Unis, à partir d'un réseau de communication entre ordinateurs. Au fil des années, les informaticiens ont conçu de nombreux protocoles toujours plus performants pour relier davantage d'ordinateurs entre eux. L'agrandissement du réseau Internet provient des recherches d'un informaticien du CERN (Organisation européenne pour la recherche nucléaire), dont le souhait était de partager de nombreuses informations entre scientifiques.

Ainsi en 1991, le réseau relie plus de 1 000 000 ordinateurs entre eux. Aujourd'hui, plus de 4,1 milliards de personnes, soit 53,6% de la population mondiale, peuvent accéder à Internet. De par l'augmentation d'appareils connectés, des problèmes de transmission se sont posés. La recherche d'une solution alternative pour contenter de plus en plus d'internautes était primordiale.

Ces recherches ont permis aux laboratoires du CNET (Centre national d'études des télécommunications) de trouver la technologie qui répond à ces critères en 1994. Cette technologie est celle que nous connaissons aujourd'hui : l'ADSL. Cependant, il faudra patienter jusqu'en 1999 pour profiter de la première commercialisation d'une offre ADSL. Si la fibre optique fait partie des "nouvelles technologies", elle est en fait plus ancienne que l'ADSL.

1.2.2 Word Wide Web (WWW)

Le concept du Web (« la toile » en français) repose sur la notion d'hypermédia, c'est à dire la réunion de documents multimédia (texte, son, image...) par l'intermédiaire de liens préétablis. Le protocole utilisé est le Hyper Text Transfer Protocol (HTTP), qui permet de transférer à partir d'un serveur Web des pages écrites dans le langage de programmation Hyper Text Markup Language (HTML). Pour exploiter l'hypertexte et ainsi passer facilement de pages en pages situées sur des serveurs répartis dans le monde, l'utilisateur doit disposer dans sa machine d'un logiciel dit « navigateur », le plus souvent gratuit et déjà intégré au système d'exploitation de l'ordinateur. Ce logiciel permet de localiser les pages Web et par extension toute ressource disponible sur Internet,grâce à son adresse textuelle nommée Uniform Ressource Locator (URL) (Jean Paul ,2015).
Le Web se définit par son contenu comme un grand réservoir d'informations exploitables sur l'Internet. C'est donc la partie multimédia de l'Internet permettant à la fois la diffusion de textes, de sons, d'images etc (Paluku Vagheni Aloys,2014).

URL (Uniform Resource Locator)

Une URL (Uniform Resource Locator) est un format de nommage universel pour désigner une ressource sur Internet. Il s'agit d'une chaîne de caractères ASCII imprimables qui se décompose en cinq parties (Pillou, 2016d).

- **Le nom du protocole :** le langage utilisé pour communiquer sur le réseau. Le protocole le plus largement utilisé est le protocole HTTP (HyperText Transfer Protocol), le protocole permettant d'échanger des pages Web au format HTML. De nombreux autres protocoles sont toutefois utilisables (FTP, News, Mailto, Gopher,etc).
- **Identifiant et mot de passe :** permet de spécifier les paramètres d'accès à un serveur sécurisé. Cette option est déconseillée car le mot de passe est visible dans l'URL.
- **Le nom du serveur :**il s'agit d'un nom de domaine de l'ordinateur hébergeant la ressource demandée. Notez qu'il est possible d'utiliser l'adresse IP du serveur, ce qui rend par contre l'URL moins lisible.
- **Le numéro de port :** il s'agit d'un numéro associé à un service permettant au serveur de savoir quel type de ressource est demandé. Le port associé par défaut au protocole est le port numéro 80. Ainsi, lorsque le service Web du serveur est associé au numéro de port 80.
- **Le chemin d'accès à la ressource :** permet au serveur de connaître l'emplacement auquel la ressource est située, c'est-à-dire de manière

générale l'emplacement (répertoire) et le nom du fichier demandé.

HTTP (Hyper Text Transfer Protocol)

Le protocole HTTP (Hyper Text Transfer Protocol) est le protocole le plus utilisé sur Internet depuis 1990. Au début, il était uniquement destiné à transférer des données sur Internet (en particulier des pages Web écrites en HTML. La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message en utilisant un codage de type MIME. Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à une chaîne de caractères appelée URL entre un navigateur (le client) et un serveur Web (appelé d'ailleurs httpd sur les machines UNIX) (Pillou, 2016b).

HTML (Hyper Text Markup Language)

Ce langage se compose d'un ensemble d'annotations, appelées étiquettes ou balises, qui permettent de créer et formater un document hypertexte. Un fichier HTML est un fichier texte ce qui a l'avantage de le rendre facilement lisible sur n'importe quelle plateforme/ordinateur. Les balises du HTML sont insérées dans le texte du document et guident son affichage. Le navigateur interprète les commandes HTML contenues dans le document et en déduit le format d'affichage du document (Pillou, 2016a).

1.3 Évolution du Web

Pour comprendre le Web actuel, il faut connaître son évolution car chaque nouvelle version est issue des insuffisances de la version précédente (techniques, usage, etc.). Par ailleurs, elle vient la compléter, et non la remplacer (Waterschoot, 2014, Bruyère et al., 2011).

1.3.1 Web 0.0 (1972)

Le Web 0.0 ou Web militaire symbolise l'origine de l'Internet avec la création en 1972 du premier réseau de données à transfert de paquets (ARPANET) suite à la commande du Pentagone portant sur la création un réseau capable de résister à une attaque militaire.

1.3.2 Web 1.0 (1991-1999)

Le Web 1.0, encore appelé Web traditionnel, est avant tout un Web statique, centré sur la distribution d'informations. Il se caractérise par des sites orientés produits, qui sollicitent peu l'intervention des utilisateurs.
Le Web 1.0 a popularisé le Web auprès du grand public en rendant possible la publication de pages HTML mélangeant du texte, des liens, des images, consultables en ligne dans un navigateur Web via une URL grâce au protocole HTTP.
Le Web 1.0 est caractérisé par :

- La notion de *site Web* qui est comparable dans le monde physique à une bibliothèque où il faut se rendre pour avoir accès à son contenu et dans laquelle un individu ne peut pas modifier une information mais uniquement la consulter (une autre analogie souvent faite pour expliquer le Web 1.0 est la télévision où l'individu est uniquement spectateur de programmes) ;
- La possibilité de publier un contenu Web uniquement par le propriétaire du site Web ;
- Des pages statiques en HTML, l'internaute qui ne peut que consulter les pages.

1.3.3 Web 2.0 (2000-2009)

Le Web 2.0, ou Web social, qui change totalement de perspective privilégie la dimension de partage et d'échange d'informations et de contenus (textes, vidéos, images ou autres). Il voit l'émergence des réseaux sociaux, des Smartphones et des blogs. Le Web se démocratise et se dynamise. Appelé aussi Web collaboratif/participatif, il symbolise le Web interactif où l'internaute n'est plus seulement un consommateur mais aussi un producteur d'informations car son avis est sollicité en permanence.

Le Web 2.0 n'est plus un *site Web* mais plutôt une plateforme Web sur la base d'un ensemble de principes :

- L'internaute devient producteur d'information sur des supports Web qui ne lui appartiennent pas (exemple : blog) ;
- L'information vient à l'internaute grâce à la syndication des flux RSS ;
- Le Web devient une gestion personnalisée par l'internaute de flux de données (lecteur adapté et pas forcément un navigateur Web, affichage personnalisé, etc.) ;
- L'interaction et le partage transforment le Web en média social ;
- Les sites Web deviennent des services Web (exemple : Google Docs).

Parmi les avantages du Web 2.0, nous citons :
— La simplicité et la rapidité d'implémentation et d'utilisation ;
— La gratuité des outils ou des coûts d'utilisation modique ;
— La grande interactivité ;
— Les possibilités de personnalisation des outils.

1.3.4 Web 3.0 (2010-20xx)

Le Web 3.0, aussi nommé Web sémantique, vise à organiser la masse d'informations disponibles en fonction du contexte et des besoins de chaque utilisateur, en tenant compte de sa localisation, de ses préférences, etc. C'est un Web plus portable et qui fait de plus en plus le lien entre monde réel et monde virtuel. Il répond aux besoins d'utilisateurs, qui sont toujours connectés à travers une multitude de supports et d'applications.

1.4 Architecture du Web

De nombreuses applications fonctionnent selon un environnement client/-serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique) (Pillou, 2016c),

1.4.1 Qu'est-ce qu'un serveur ?

Un serveur est un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Le terme serveur s'applique à la machine sur laquelle s'exécute le logiciel serveur. Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec accès permanent et s'exécuter en permanence.

1.4.2 Qu'est-ce qu'un client ?

Un logiciel client est un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut-être raccordé par une liaison temporaire.

1.4.3 Les modèles de client-serveur

les différences entre les modèles sont essentiellement liées aux services qui sont assurés par le serveur.

— **Client-serveur de donnée :** La partie cliente assure à la fois les fonctions de présentation et de logique applicative. Le serveur assure uniquement la gestion des données, le plus souvent à l'aide d'un système de gestion de base de données relationnelle. Dans ce modèle, l'application cliente envoie les requêtes au serveur de données, qui envoie en retour les données demandées, ou informe le client du résultat obtenu. La mise en place d'une application de ce type est facilitée à la fois par une répartition claire des fonctions entre le client et le serveur, et par les produits tels que les SGBD relationnels qui peuvent aussi ajouter le contrôle d'intégrité. Il s'ensuit une sécurité accrue dans la gestion des données, mais aussi une amélioration sensible des performances, car les requêtes de contrôle n'ont pas à transiter.

— **Client-serveur de présentation :** La présentation des pages affichées par le client est intégralement prise en charge par le serveur. Cette organisation présente l'inconvénient de générer un fort trafic réseau.

— **Client-serveur de traitement :** Le serveur effectue des traitements à la demande du client. Il peut s'agir de traitement particulier sur des données, de vérification de formulaires de saisie, de traitements d'alarmes, etc. Ces traitements peuvent être réalisés par des programmes installé sur des serveurs mais également intégrés dans des bases de données (triggers, procédures stockées), dans ce cas, la partie donnée et traitement sont intégrés.

1.4.4 Architecture Client-Serveur

Il existe une multitude d'architectures client-serveur, nous verrons dans ce qui suit les plus répandues (Pillou, 2016c).

— **L'architecture 2 tiers** Dans une architecture deux tiers, encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un

service spécialisé. Le cas typique de cette architecture est une application de gestion fonctionnant sous Windows ou Linux et exploitant un SGBD centralisé. Ce type d'application permet de tirer partie de la puissance des ordinateurs dé- ployés en réseau pour fournir à l'utilisateur une interface riche, tout en garantissant la cohérence des données, qui restent gérées de façon centralisée, voir figure 1.1 suivante.

— **Premier niveau :** L'affichage et les traitements locaux (contrôles de saisie, mise en forme de données, etc.) sont pris en charge par le poste client ;
— **Deuxième niveau :** Les traitements applicatifs globaux sont pris en charge par le service applicatif ;
— **Troisième niveau :**Les services de base de données sont pris en charge par un SGBD.

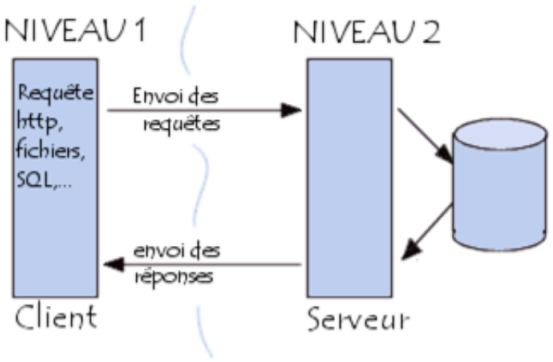


FIGURE 1.1 – Architecture 2 tiers

— **L'architecture 3 tiers** Cette architecture trois tiers, également appelée client-serveur de deuxième génération ou client-serveur distribué sépare l'application en 3 niveaux de services distincts voir figure 1.2.

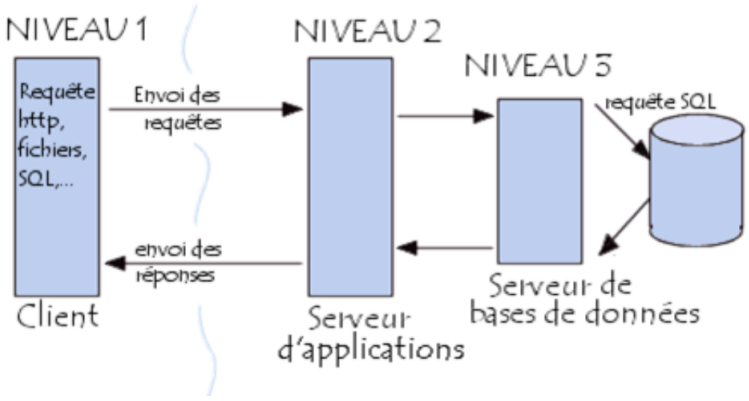


FIGURE 1.2 – Architecture 3 tiers

— **L'architecture n-tiers :** L'architecture n-tiers a été pensée pour pallier aux limitations des architectures trois tiers et concevoir des applications puissantes et simples à maintenir. Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux. Cette évolution des architectures trois tiers met en œuvre une approche objet pour ouvrir une plus grande souplesse d'implémentation et faciliter la réutilisation des développements.

1.5 Ergonomie du Web

Dans les années 50, elle est définie par Alain Wisner, comme étant : « *L'ensemble des connaissances scientifiques relatives à l'Homme nécessaires pour concevoir des outils, des machines et des dispositifs qui puissent être utilisés avec le maximum de confort, de sécurité et d'efficacité.*»
Pour qu'un site Web soit ergonomique, il doit être utile et utilisable :

1. **La notion d'utilité**
   Un site est dit utile lorsqu'il est conçu pour réaliser une tâche pour l'internaute, il offre des services aux internautes ayant un besoin. Pour cela, il faut d'abord déterminer les envies, les besoins et les exigences des visiteurs car elles représentent les objectifs de création de votre site Web.
   Avoir un site utile n'est pas suffisant pour qu'un visiteur reste visiter le site, il doit être aussi utilisable.
2. **La notion d'utilisabilité**
   La norme ISO 9241 définit l'utilisabilité de la façon suivante :

« *Un produit est dit utilisable lorsqu'il peut être utilisé avec efficacité, efficience et satisfaction par des utilisateurs donnés, cherchant à atteindre des objectifs donnés, dans un contexte d'utilisation donné* ».
Un site Web utilisable est un site qui marche, il doit être efficace, efficient, satisfait par le visiteur. Ce dernier a des objectifs, évolués dans un environnement précis.

3. **La notion d'efficacité**
   C'est le critère le plus important à satisfaire, il signifie qu'un utilisateur peut faire ce qu'il veut d'une manière simple et facile.
4. **La notion d'efficience**
   En plus de l'efficacité, avoir un site efficient est essentiel. L'utilisateur peut effectuer des tâches rapidement et quasiment correctement (sans faire beaucoup d'erreurs).
5. **La notion de satisfaction**
   Le troisième critère de l'utilisabilité est la satisfaction des utilisateurs, le site Web doit être mis au service des utilisateurs, répondre aux ses exigences. (Boucher.A,2009)

1.6 Les langages de programmation

Un langage de programmation Web sert à établir des règles et procédures logiques complexes. Là où les langages de balisage comme HTML ne produisent que des documents, un langage de programmation permet de créer n'importe quel programme détaillé pour effectuer des tâches données.
Nous pouvons les diviser en deux catégories :

1. **Langages côté client :** XHTML pour la description du contenu, CSS pour l'apparence, JavaScript pour l'interactivité.
2. **Langages côté serveur :** Python, PHP (Hypertext preprocessor), ASP (Active Server Pages), JSP (Java Server Pages) C#, Perl, etc.

1.7 Autres notions

À travers cette section, nous donnons des notions courantes sur le Web.

1.7.1 Site Web

Un site Web, ou simplement site, est un ensemble de pages Web et de ressources reliées par des hyperliens, défini et accessible par une adresse Web. L'ensemble des sites Web publics constituent le World Wide Web.

Exemples

Le site Web de l'ENS d'Oran Ammour-Ahmed (https ://www.ens-oran.dz)

1.7.2 Navigateur Web

Browser en anglais est un programme qui permet de surfer sur l'Internet. Un logiciel dont la fonction première est d'interpréter les adresses des pages Web, de les afficher et d'exploiter les liens hypertextes a l'intérieur de celles-ci. Ainsi, l'internaute peut se déplacer de page en page, par un simple clic sur un lien.

Exemples

Internet Explorer, Firefox ou Google Chrome.

1.7.3 Application Web

Une application Web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur Web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application Web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide d'un navigateur Web.
La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur les ordinateurs. Les technologies utilisées pour développer les applications web sont les mêmes que celles employées dans la création des sites internet.

Exemples

Les services Google comme Google Maps, Gmail ou le moteur de recherche, Amazon, etc.

1.7.4 Les standards du Web : W3C

Le W3C (World Wide Web Consortium) est un organisme de standardisation à but non lucratif fondé en octobre 2004 par Tim Berners-Lee.

Objectifs

Promouvoir la compatibilité des technologies du World Wide Web en rédigeant et publiant des recommandations ou standards tels que :

- XML : eXtensible Markup Language ;
- XHTML : eXtensible HyperText Markup Language ;
- CSS : Cascading Style Sheet ;
- PNG : Portable Network Graphics ;
- Etc.

1.8 Conclusion

Dans ce chapitre, nous avons abordé les notions de base du Web ainsi que son évolution. Le chapitre suivant donne les premiers pas vers la programmation Web.

2

Premiers pas en HTML

2.1 Introduction

Le HTML est un langage de programmation très simple et constitue une excellente initiation à la programmation Web.
Dans ce chapitre, nous donnerons les éléments essentiels pour débuter la programmation Web avec HTML.

2.2 Définitions

HTML, pour HyperText Markup Language traduit littéralement en langage de balisage d'hypertexte, est le langage conçu pour représenter les pages Internet. C'est un langage de balises permettant d'écrire de l'hypertexte. HTML permet de structurer sémantiquement le contenu de la page, d'inclure des ressources multimédia, des formulaires de saisie et des programmes informatiques.

2.3 Évolution de HTML

HTML (Hypertext Markup Language), un langage de balisage conçu pour afficher des pages Web.

— Vers 1993 et alors appelé SGML (quoique les fichiers de données affichaient le suffixe .html). Cet HTML primitif contient déjà divers éléments comme le titre du document, les hyperliens, la structuration du texte en titres, sous-titres, listes ou texte brut. Tout est prêt mais il manque quelque chose : des instruments qui permettraient au usagers de voir et consulter les pages - il fallait des fureteurs (en anglais :

browsers).
— Vers la même époque, NCSA Mosaic remplit le vide, c'est le premier fureteur : il affiche déjà le texte et les images.
— En 1994, Netscape affiche des cadres (frames).
— Un peu plus tard, Internet Explorer, mis sur le marché lui aussi affiche le texte, les images et les cadres. L'état de HTML correspond alors à ce que l'on pourrait appeler HTML 1.0.
— En octobre 1994, apparition du W3C (World Wide Web Consortium),
— En 1995, Netscape Navigator ajoute le support de nombreux éléments de présentation : attributs de texte, clignotement, centrage, etc. à ce moment, les développeurs ressentent un besoin pour une façon de styliser l'affichage à l'extérieur du HTML.
— Le **W3C** travaille sur le développement d'un modèle d'objets de document qui a rapidement évolué vers une standardisation des fureteurs. il propose alors HTML+, un brouillon de HTML 3.0 avec support des tables, des figures et des expressions mathématiques.
— Au début de 1997, le W3C publie la norme HTML 3.2 qui décrit la pratique courante tels que des styles et des scripts.
— À la fin de 1997, HTML 4.0 apporte différentes améliorations pour l'accessibilité des contenus dont principalement la possibilité d'une séparation plus explicite entre structure et présentation du document.
— En 1999, HTML 4.01, une révision de HTML 4.0, est entré en vigueur.
— HTML 5.0 soit actuellement dans sa phase finale de validation. Elle remplace la version 4.01 et introduire de nouvelles balises, de nouvelles API et de nouveaux attributs qui vont étendre les capacités du langage HTML au delà du simple objectif d'afficher des informations.

2.4 Utilité du HTML

Le langage HTML permet de :
— Définir la structure logique d'un document Web ;
— Composer d'un ensemble de commandes de formatage ;
— Se baser sur la notion d'environnement possédant un début et une fin ;
— Utiliser les délimiteurs : c'est des balises ou marqueurs ;
— Mettre en œuvre du HTML, il suffit d'un éditeur de texte pour taper le code de la page, d'un navigateur WEB pour afficher la page formatée.
— Les balises sont insensibles à la casse et peuvent aussi bien être écrites en minuscules, en majuscules voire avec un mélange des deux.

2.5 Règles ergonomiques

Pour concevoir un site Web, il faut veiller à :

— **Mise en page et styles :** maintenir une certaine uniformité sur l'ensemble du site ;

— **Clarté :** éviter la confusion, présenter de façon simple avec des menus, éviter les textes longs. . . ; l'information doit être trouvée dès les premières secondes ;

— **Hyperliens :** ne pas en abuser, faire en sorte qu'ils indiquent clairement où ils mènent ;

— **Identité :** le visiteur doit savoir facilement sur quel site il est (quelle que soit la page visitée) et savoir à qui s'adresser –> logos, mail, date de dernière mise à jour de la page.

— **Réactivité :** ne pas abuser des images, exécutions de code (temps de chargement/exécution) ;

— **Navigateurs :** faire attention à leurs particularités, s'adapter à la taille de la fenêtre, au type du terminal utilisé par le client, etc.

2.6 Structure d'un document HTML

Les pages Web peuvent sembler assez différentes les unes des autres, mais elles ont toutes tendance à partager des composantes standard similaires.

Pour qu'une page HTML soit déclarée valide, elle doit obligatoirement comporter certains éléments et suivre un schéma précis.En effet, une page non valide ne sera pas comprise par le navigateur qui va alors potentiellement mal l'afficher voire dans certains cas ne pas l'afficher du tout comme montré à la figure 2.1.

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Ma première page en HTML</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8          Le contenu de page
9      </body>
10
11 </html>
```

FIGURE 2.1 – Structure minimal d'un document HTML

1. **Le doctype**
   Comme son nom l'indique, le doctype sert à indiquer le type du document.Dans la balise de l'élément doctype, le langage utilisé est précisé, à savoir le HTML.
2. **L'élément HTML**
   Après avoir renseigné le type du document, le document doit contenir un élément html. Cet élément est composé d'une paire de balises ouvrante <html> et fermante </html>.

   L'élément html représente la page Web où tout le contenu y est inséré (et donc les autres éléments) à l'intérieur de celui-ci.
3. **Les éléments `head` et `body`**
   A l'intérieur de l'élément html, deux éléments sont obligatoirement indiqués qui sont les éléments head et body et qui vont avoir des rôles très différents.

   L'élément head est un élément d'en-tête. Il va contenir des éléments qui vont servir à fournir des informations sur la page au navigateur, comme le titre de la page ou encore le type d'encodage utilisé pour que celui-ci puisse afficher les caractères de texte correctement.

2.6.1 Anatomie d'un élément

Les principales parties d'un élément sont :

1. **La balise ouvrante :** il s'agit du nom de l'élément, encadré par un chevron ouvrant (<) et un chevron fermant (>). Elle indique où l'élément commence ou commence à prendre.
2. **La balise fermante :** c'est la même que la balise ouvrante, sauf qu'elle comprend une barre oblique (/) avant le nom de l'élément. Elle indique la fin de l'élément. Ne pas inclure une balise de fermeture est une erreur qui peut amener des résultats étranges ;
3. **Le contenu :** il s'agit du contenu de l'élément ;
4. **L'élément :** l'ensemble balise ouvrante, balise fermante et contenu constituent l'élément.

Exemple

La figure 2.2 montre un élément textuel "p".

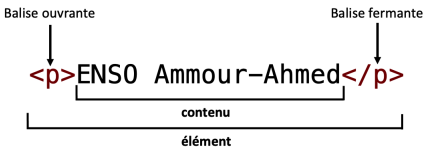


FIGURE 2.2 – Structure d'un élément

2.6.2 Élément imbriqué

Mettre des éléments à l'intérieur d'autres éléments, cela s'appelle **l'imbrication.**

Exemple

La figure 2.3 montre des éléments imbriqués.

```
1   <p>Mon école se situe à <strong>Oran</strong>.</p>
```

FIGURE 2.3 – Éléments imbriqués en HTML

2.6.3 Élément vide

Tous les éléments ne suivent pas le modèle d'ouverture de balise, puis contenu, puis fermeture de balise. Certains éléments ne sont composés que d'une balise. Ils servent généralement à **insérer ou à incorporer** quelque chose dans le document à l'endroit où ils sont mis.

Exemple

La balise **<img/>** insère une image dans une page à l'endroit. Cette balise est auto-fermante.

2.6.4 Attribut

Les attributs contiennent des informations supplémentaires sur l'élément sans qu'elles n'apparaissent dans le contenu réel. Dans ce cas, l'attribut class permet de donner à l'élément un nom d'identification qui peut ensuite être utilisé pour cibler l'élément afin de lui attribuer un style CSS ou un comportement particulier, par exemple. Pour créer un attribut, il faut :

1. insérer un espace entre cet attribut et le nom de l'élément (ou l'attribut précédent, si l'élément possède déjà un ou plusieurs attributs) ;
2. donner un nom à l'attribut, puis ajouter un signe égal ;
3. donner une valeur à l'attribut, entourée par des guillemets d'ouverture et de fermeture.

Exemple

La figure 2.4 montre un exemple d'attribut utilisé dans un élément HTML.

```
1   <p class="class-css">Mon école ENSO</p>
```

FIGURE 2.4 – Exemple d'attribut

Attributs booléens ou attributs sans valeurs définies ; ils ne peuvent avoir qu'une seule valeur, généralement la même que le nom de l'attribut voir l'exemple donné à la figure 2.5.

Exemple

```
3   <input type="text" disabled="disabled">
```

FIGURE 2.5 – Exemple d'attribut booléen

2.6.5 Formatage du texte

L'un des principaux buts de HTML est de structurer du texte et lui donner du sens afin que le navigateur puisse l'afficher correctement.

Il existe d'innombrables balises qui permettent de mettre en page un texte en HTML. Nous nous intéresserons aux balises les plus utilisées. D'abord, nous allons commencer par les balises qui ne nécessitent pas d'attributs tel montré à la table 2.1.

Exemple complet

La figure 2.6 montrée ci-dessous donne un aperçu général des balises simples de formatage du texte en HTML.

TABLE 2.1 – Les balises de formatage du texte en HTML sans attribut

Élément	Description
<b>	Il s'agit d'une balise utilisée pour mettre en gras le texte écrit entre elles.
****	Cette balise indique au navigateur que le texte est important.
<i>	Cette balise est utilisée pour rendre le texte en italique.
<u>	Cette balise est utilisée pour souligner le texte écrit entre elle.
<sup>	Cette balise affiche le contenu légèrement au-dessus de la ligne normale.
<sub>	Cette balise affiche le contenu légèrement en dessous de la ligne normale.
<big>	Cette balise est utilisée pour augmenter la taille de la police d'une unité conventionnelle.
<small>	Cette balise est utilisée pour réduire la taille de la police d'une unité par rapport à la taille de la police de base.
<h1>, <h2>...<h6>	La balise <h1> (heading) désigne un titre de premier niveau. Le texte qu'elle entoure est de grande taille et mis en gras. Il est utile pour créer les grands titres d'une pages. La balise <h2> désigne un titre de deuxième niveau, elle applique le même effet que la balise précédente mais avec une taille de caractères légèrement plus petite. Il existe aussi <h3>, <h4>, <h5> et <h6> avec une taille de caractères de plus en plus petite.
<p>	définit un paragraphe
** **	définit un retour à la ligne (pas besoin d'une balise de fermeture)
<!– –>	Un commentaire HTML est visible dans le code source mais ignoré par le navigateur. Il sert à marquer un bloc de code pour que celui-ci soit facilement trouvé et compris lors de sa prochaine manipulation.

2.6.6 Les caractères spéciaux

Le langage de balisage HTML demande de respecter un codage spécifique pour les caractères spéciaux. Ainsi, tous les caractères accentués doivent être

FIGURE 2.6 – Balises de formatage HTML

"traduits" dans des suites de caractères qui débutent par & et se terminent par un point virgule (;).

Les documents HTML ne devaient à l'origine du HTML être constitués que de caractères ASCII (norme définissant 128 caractères). Les lettres accentuées, les lettres d'autres alphabets, certains symboles et d'autres caractères ne sont pas compris dans cet encodage ; ils sont sont considérés comme des caractères spéciaux. Pour afficher des caractères spéciaux, on utiles des entités HTML qui peuvent se définir ainsi :

$$\& + nom(oucodeISO) +;$$

Exemple

Pour afficher le caractère **é**, nous pouvons écrire :

$$\é$$

ou avec le code ISO :

$$\&233;$$

Encodage UTF-8

Actuellement, il n'est plus obligatoire d'écrire un document HTML uniquement avec des codes ASCII. Nous pouvons utiliser des encodages comprenant plus de caractères tels que : ISO-8859-1, ISO-8859-15, UTF-8,etc. Tous les caractères compris dans ces encodages, l'affichage des caractères non compris doit normalement être fait via les entités correspondantes. Ainsi en ISO-8859-1, il nécessaire d'utiliser les entités pour les caractères Sur ce point l'utilisation de l'encodage UTF-8 présente le grand avantage de n'avoir quasiment pas à recourir aux entités HTML, puisque cet encodage comprend théoriquement tous les caractères voir exemple montré à la figure 2.7 .

Exemple

La figure 2.7 montre un exemple d'utilisation de l'encodage UTF-8.

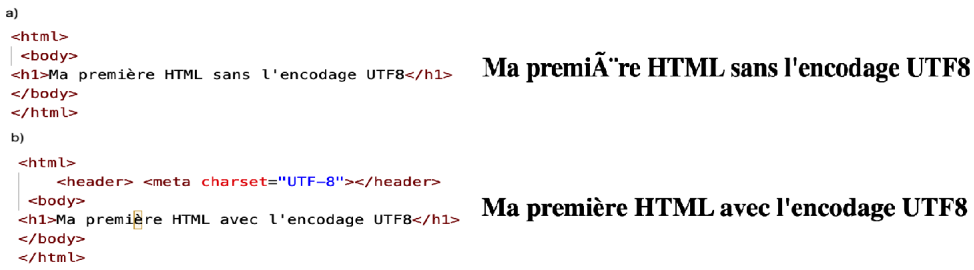


FIGURE 2.7 – Exemple d'utilisation de l'encodage UTF8

2.6.7 Les listes

Les listes HTML permettent d'ordonner du contenu. Ce contenu peut être hiérarchisé ou non.
Les listes sont très utiles pour apporter de la clarté et de l'ordre à un document. Elles permettent de lister plusieurs éléments et de les regrouper. Aussi, elles sont utiliser pour créer des menus de navigation.

2.6.8 Listes non-ordonnées (à puces)

Les listes non-ordonnées, comme leur nom l'indique, n'ont pas de notion d'ordre. Elles se présentent par défaut sous la forme d'une liste à puces comme dans les logiciels de traitement de texte (exp : Word).
La balise HTML correspondant aux listes non-ordonnées (« unordered list » en anglais) est $< ul >< /ul >$ qui va représenter la liste en soi ainsi que d'un élément $< li >< /li >$ (« list item » ou « élément de liste ») pour chaque nouvel élément de liste voir exemple montré à la figure 2.8.

Exemple

La figure 2.8 illustre l'utilisation des listes non-ordonnées en HTML.

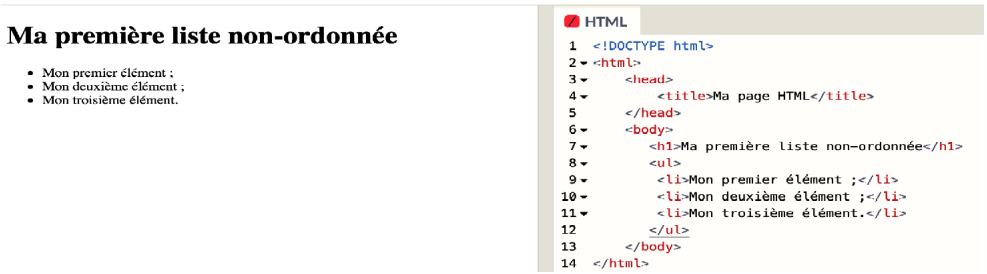


FIGURE 2.8 – Exemple de liste non-ordonnée

2.6.9 Listes ordonnées (numérotées)

Elles concernent des éléments qui doivent être lus dans un ordre donné, ces éléments seront alors numérotés. Ce type de liste est généralement utilisé pour lister des étapes à suivre. Pour créer une liste ordonnée, l'élément utilisé est $< ol >< /ol >$ (pour « ordered list » ou « liste ordonnée ») pour définir la liste en soi et à nouveau des éléments $< li >< /li >$ pour chaque élément de la liste voir exemple montré à la figure 2.9.

Exemple

La figure 2.9 illustre l'utilisation des listes ordonnées en HTML.

FIGURE 2.9 – Exemple de liste ordonnée

2.6.10 Listes de définitions

Le dernier type de liste HTML est la définition. La liste de définitions, qui est créée à l'aide de l'élément $< dl >< /dl >$, consiste généralement en une succession de couples terme $< dt >< /dt >$/définition $< dd >< /dd >$ (même si les listes peuvent avoir d'autres usages) voir exemple montré à la figure 2.10.

Exemple

La figure 2.10 donne un exemple de liste de définition.

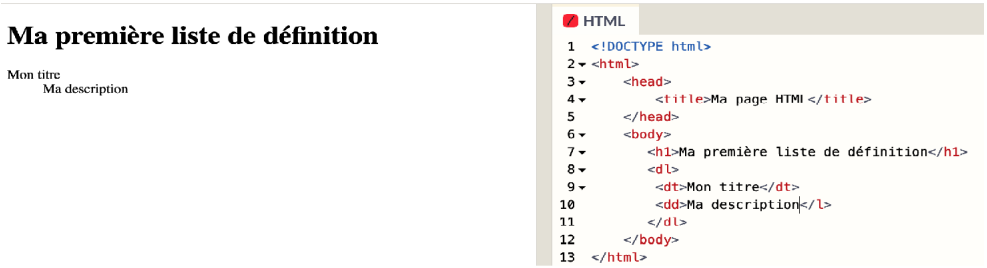


FIGURE 2.10 – Exemple de liste de définition

2.6.11 Les liens hyper-texte

Les Hyperliens sont vraiment importants, ils sont ce qui fait du Web une toile. Afin de créer un lien HTML, il faut utiliser l'élément HTML $< a >< /a >$ (pour ancre ou anchor en anglais) accompagné de son attribut $< href >$. Cet élément crée un lien hypertexte vers des pages web, des fichiers, des adresses e-mail, des emplacements se trouvant dans la même page, ou tout ce qu'une URL peut adresser . En plus de l'attribut href qui est le chemin vers la page à lier montré à l'exemple 2.11, d'autres attributs peuvent êtres utilisés avec l'élément <a> :

title : qui affiche une info bulle sur le lien ;

target :

- $_blank$: Ouverture de la page cible dans une nouvelle fenêtre montré à l'exemple 2.12 ;
- $_parent$: Ouverture de la page cible dans le cadre parent (de niveau immédiatement supérieur) ;
- $_self$: (Valeur par défaut) Ouverture de la page cible dans le cadre d'appel ;
- $_top$: Ouverture de la page cible dans la fenêtre hôte (par-dessus le FRAMESET).

Exemples

Les figures montrées ci-dessous 2.11, 2.12, 2.13 et 2.14 représentent l'utilisation des liens hypertextes.

Créer un lien vers une URL

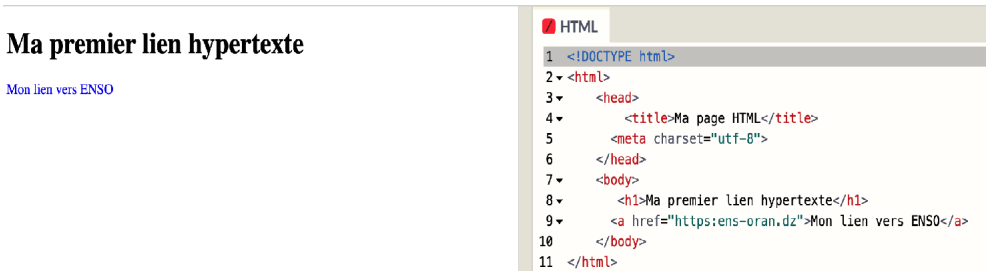


FIGURE 2.11 – Exemple une URL

Un lien qui s'ouvre sur une nouvelle page

FIGURE 2.12 – Exemple une URL qui s'ouvre sur une nouvelle page

Un lien vers la même page

FIGURE 2.13 – Exemple d'un lien vers la même page

Un lien vers une messagerie

FIGURE 2.14 – Exemple d'un lien vers la messagerie

2.6.12 Les images

Pour mettre une image simple sur une page Web,l'élément $< img >$ est utilisé. C'est un élément vide (ce qui signifie qu'il ne contient ni texte ni balise de fermeture) qui demande au moins un attribut pour fonctionner *src*. L'attribut *src* contient un chemin pointant vers l'image à intégrer, qui peut être une URL absolue ou relative, de la même manière que l'élément <a> et l'attribut href vu dans la section précédente 2.6.11. En plus de l'attribut *src*, les attributs suivants peuvent êtres aussi utilisés :

- alt : sa valeur est supposée être un descriptif sous forme de texte de l'image, à utiliser dans les cas où l'image ne peut être affichée ;
- width et height : pour spécifier la largeur et la hauteur de votre image ;
- title : pour fournir un supplément d'information si nécessaire ;
- align : pour définir l'alignement d'une image. L'attribut align peut prendre 5 valeurs : left, right, middle, top et bottom.

Exemple

l'image peut être intégrée par son URL comme montré à la figure 2.15, dans cet exemple les quatre attributs son utilisés.

```html
<!DOCTYPE html>
<html>
    <header>
        <title>Ma page HTML</title>
         <meta charset="UTF-8">
        </header>
 <body>
<h1>Ma première image en HTML</h1>
<img src="images/imagehtml.jpeg"
     alt="Image sur la technologie du Web"
     width="500"
     height="250">
</body>
</html>
```

FIGURE 2.15 – Exemple d'une insertion d'image en HTML

2.6.13 Les tableaux

Un tableau est un ensemble structuré de données (table de données) présentées en lignes et colonnes. Il permet de retrouver rapidement et facilement des valeurs au croisement entre différents types de données.

En HTML, les tableaux permettent de présenter des données de manière organisée et sous une certaine forme pour les structurer et les rendre compréhensibles pour les navigateurs, moteurs de recherche et utilisateurs.

Pour créer un tableau fonctionnel en HTML, il faut utiliser au minimum trois (3) éléments :

— Un élément $< table >$ (« tableau » en français) qui va définir le tableau en soi ;
— Des éléments $< tr >$, pour « table row » ou « ligne de tableau » en français qui permet d'ajouter des lignes dans le tableau ;
— Des éléments $< td >$, pour « table data » ou « donnée de tableau » en français qui permet d'ajouter des cellules dans les lignes et ainsi de créer automatiquement de nouvelles colonnes.

Exemple

Dans cet exemple, le tableau montré à la figure 2.16 est composé de quatre (4) éléments, de deux (2) lignes et deux (2) colonnes.

Très souvent, les tableaux vont posséder une ligne d'en-tête (voir la figure 2.17) dans laquelle des informations sont données au lecteur sur le type des données qui seront renseignées dans chaque colonne.

Cette ligne d'en-tête est différente des autres lignes puisqu'elle ne contient pas le même type d'informations que les autres lignes du tableau.

```html
<!DOCTYPE html>
<html>
    <header>
        <title>Ma page HTML</title>
        <meta charset="UTF-8">
    </header>
<body>
<h1>Ma première table en HTML</h1>
<table>
    <tr>
        <td>1.1</td>
        <td>1.2</td>
    </tr>
    <tr>
        <td>2.1</td>
        <td>2.2</td>
    </tr>
</table>
</body>
</html>
```

FIGURE 2.16 – Exemple de tableau basic en HTML

```html
<!DOCTYPE html>
<html>
    <header>
        <title>Ma page HTML</title>
        <meta charset="UTF-8">
    </header>
<body>
<h1>Ma première table en HTML</h1>
<table>
    <tr>
        <th>Titre 1</th>
        <th>Titre 2</th>
    </tr>
    <tr>
        <td>1.1</td>
        <td>1.2</td>
    </tr>
    <tr>
        <td>2.1</td>
        <td>2.2</td>
    </tr>
</table>
</body>
</html>
```

FIGURE 2.17 – Exemple de tableau avec en-tête en HTML

2.6.14 Les formulaires

Les formulaires HTML permettent aux visiteurs d'envoyer des données qui seront ensuite manipuler et/ou stocker. Ils donnent la possibilité à des utilisateurs de s'inscrire sur des sites (formulaires d'inscription), de se connecter (formulaire de connexion), d'envoyer des messages (formulaire de contact), de laisser des commentaires, etc (voir la figure 2.18).

La balise $< form >$ vient en début du formulaire, (la balise $< /form >$ ferme le formulaire) elle possède deux attributs indispensables.

1. L'attribut *method* indique par quel moyen les données vont être envoyées. Les méthodes les plus utilisées sont *get* et *post*.
2. L'attribut *action* indique l'adresse de la page ou du programme qui va traiter les informations.

Exemple

L'exemple montré à la figure 2.18 présente un formulaire de connexion.

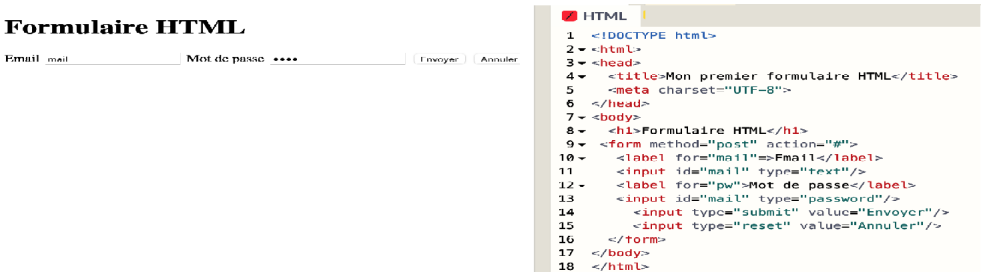


FIGURE 2.18 – Exemple d'un formulaire HTML

Remarques

1. Les valeurs *get* et *post* déterminent la méthode de transit des données du formulaire. En choisissant *get*, les données doivent transiter via l'URL (sous forme de paramètres) tandis qu'en choisissant la valeur *post* les données du formulaire sont envoyés via transaction post HTTP.
2. L'envoi via l'URL (avec la valeur *get*), la quantité de données pouvant être envoyées est limitée et surtout les données vont être envoyées en clair. Cette méthode n'est pas utilisée pour l'envoie des mots de passe ou toute information sensible.

Les éléments d'un formulaire

Le table 2.2 résume les différents éléments que peut contenir un formulaire HTML.

TABLE 2.2 – Les balises de formatage du texte en HTML sans attribut

Élément	Définition
input	Définit un champ de données à l'utilisateur.
label	Définit une légende pour un élément input.
textarea	Définit un champ de texte long.
select	Définit une liste de choix.
fieldset	Permet de regrouper des éléments du formulaire en différentes parties.
legend	Ajoute une légende à un élément fieldset.
option	Définit une option dans une liste.
optgroup	Définit un groupe d'options dans une liste.

L'élément input et les valeurs de l'attribut type

L'élément *input* est l'élément à connaitre dans le contexte de la création de formulaires HTML puisqu'il permet de créer tous types de champs pour récolter des données utilisateurs.
L'attribut type peut prendre de nombreuses valeurs. Les valeurs les plus utiles et les plus courantes sont montrées à la table 2.3.

TABLE 2.3 – Les valeurs que peut prendre l'attribut type de l'élément input

Type	Définition
Text	Définit un champ de texte monoligne qui accepte du texte .
Number	Définit un champ qui accepte un nombre décimal.
Email	Crée un champ qui permet de renseigner une adresse mail.
Date	Permet à l'utilisateur d'envoyer une date.
Password	Crée un champ monoligne dont la valeur va être cachée.
Checkbox	Permet de créer un case à cocher. L'utilisateur peut cocher une ou plusieurs cases en même temps.
Radio	Permet de créer un bouton radio. Un seul bouton radio peut-être coché dans un ensemble.
Url	Crée un champ qui accepte une URL.
File	Permet à l'utilisateur de télécharger un fichier.
Submit	Crée un bouton d'envoi du formulaire.

Attributs des formulaires

La règle à retenir concernant les formulaires HTML les données envoyées par les utilisateurs peuvent-être erronées. Certains utilisateurs font des fautes d'inattention pendant le remplissage des champs du formulaire. La solution est d'ajouter des contraintes directement dans le formulaire pour limiter les données qui vont pouvoir être envoyées. La table 2.4 donne un ensemble d'attributs utilisés pour cette raison.

Table 2.4 – Autres attributs d'un formulaire HTML

Attribut	Définition
size	Permet de spécifier le nombre de caractères dans un champ.
minlenght	Permet de spécifier le nombre minimal de caractères dans un champ.
maxlength	Permet de spécifier le nombre maximal de caractères dans un champ.
min	Permet de spécifier un valeur minimale pour un champ nombre ou date.
max	Permet de spécifier un valeur maximale pour un champ nombre ou date.
autocomplete	Permet d'activer l'auto complétion pour un champ.
required	Permet de forcer le remplissage d'un champ. Le formulaire ne peut-être envoyé si ce champ est vide.
pattern	Permet de préciser une valeur régulière. La valeur de ce champ doit respecter cette forme.

2.7 Les limites du HTML

Les limites du langage HTML peuvent êtres résumées comme suit :

— Absence de toutes structures algorithmiques (conditionnelles ou itératives) ;
— Un langage de création d'interface sans aucune logique de programmation évoluée (notion de sous programmes, gestion des évènements, variables...) ;
— Aucune communication avec la plateforme d'exécution (date système, le navigateur utilisé...) ;
— Absence de toutes fonctionnalités de création d'animations ou de contrôle de saisie dans un formulaire ou d'interfaçage avec une base de données.
— Absence de mécanismes de verrouillage de code source (pour pouvoir le visualiser, il suffit d'utiliser l'option affichage code source de votre navigateur).

2.8 Exercice applicatif

Répondre aux questions suivantes :

1. HTML est considéré comme ?
   (a) Langage de programmation ;
   (b) Langage POO ;
   (c) Langage de haut niveau ;
   (d) Langage de balisage.
2. HTML utilise des ?
   (a) Balises définis par l'utilisateur ;
   (b) Balises prédéfinis ;
   (c) Balises fixes définis par le langage ;
   (d) Balises uniquement pour les liens.
3. Pour définir le style d'un seule élément, quel sélecteur css utiliser ?
   (a) id ;
   (b) text ;
   (c) class ;
   (d) name.

3

Cascading Style Sheets (CSS)

3.1 Introduction

Afin de créer un site, nous devons donner des instructions à l'ordinateur. Il ne suffit plus de taper le texte qui devra figurer à l'écran ; mais aussi indiquer où placer ce texte, insérer des images, faire des liens entre les pages, etc.
Dans le chapitre précédent, nous avons définit le HTML (HyperText Markup Language) paru en 1991 lors du lancement du Web. Il permet de gérer et d'organiser le contenu d'une page Web.
Dans ce chapitre, nous allons définir le CSS (Cascading Style Sheets, aussi appelées feuilles de styles) comme langage permettant de gérer l'apparence de la page Web (agencement, positionnement, décoration, couleurs, taille du texte, etc)qui permet de compléter langage HTML (voir la figure 3.1).

FIGURE 3.1 – Scripts HTML et CSS et leurs correspondance en page Web

3.2 Évolution de CSS

Ces dernières années, le CSS a évolué de plus en plus vite et dans une direction le rapprochant de plus en plus d'un langage de programmation à part entière à l'inverse d'un simple langage de présentation comme cité ci-dessous.

- **CSS 1** : dès 1996, on dispose de la première mouture du CSS. Elle pose les bases pour présenter sa page Web, comme les couleurs, les marges, les polices de caractères, etc.
- **CSS 2** : apparue en 1999 puis complétée par CSS 2.1, cette nouvelle version ajoute de nombreuses options. On peut désormais utiliser des techniques de positionnement très précises, pour afficher les éléments où on le souhaite sur la page.
- **CSS 3** : c'est la version actuelle, qui apporte des fonctionnalités particulièrement attendues comme les bordures arrondies, les dégradés, les ombres, etc.

3.3 Définition et syntaxe générale

Les feuilles de styles en cascade permettent de mettre en forme des documents Web. Le terme **cascade** signifie que la mise en forme d'une page peut faire appel à **plusieurs feuilles de styles**.
Une règle CSS est composée de 2 éléments :

- Le sélecteur : indique l'élément sur lequel vont s'appliquer les déclaration CSS ;
- La déclaration : les propriétés et les valeurs à appliquer à cet élément.

Exemple

$\#style/*sélecteur*/background-color : pink/*propriétés : valeur*/$

3.4 Utilisation du CSS

Les CSS (Cascading Style Sheets en anglais, ou « feuilles de style en cascade ») sont le code utilisé pour mettre en forme une page Web. De la même façon que HTML, le CSS est un langage de feuille de style, il permet d'appliquer des styles sur différents éléments sélectionnés dans un document HTML.

Dans cette section, nous donnons les trois manières existantes pour définir du style.

3.4.1 Définir du CSS spécifique en ligne

Nous pouvons définir individuellement un (ou plusieurs) style(s) pour un mot ou paragraphe de page par exemple. Cela consiste donc à intégrer les styles dans le corps du document HTML tel montré à la figure 3.2.

Exemple

Cet exemple montre la manière d'intégrer du style css dans un document HTML.

```
<!DOCTYPE html>
<html>
    <header>
        <title>Ma page HTML et CSS</title>
         <meta charset="UTF-8">
        </header>
 <body>
<h1>Ma première table en HTML</h1>
<H1 style="font-family: Arial; font-style: italic"> Mon premier css inline </H1>

</body>
</html>
```

FIGURE 3.2 – Scripts CSS inline

Remarque

Cette méthode n'est intéressante que lorsque le style est vraiment spécifique à un élément.

3.4.2 Méthode du CSS interne

Cette deuxième méthode consiste à mettre le style dans l'entête de la page. La syntaxe est incorporée à l'entête du fichier HTML entre les balises $< styletype = "text/css" > .... < /style >$ placées entre les tags $< head >$ et $< /head >$ tel montré à la figure 3.3.
Le style ainsi défini peut être appliqué à tous les endroits de la page, sans avoir à redéfinir les propriétés.

Exemple

Cet exemple donne la méthode pour intégrer du code css dans l'entête de la page HTML.

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Ma page HTML et CSS</title>
        <meta charset="UTF-8">
        <style type="text/css">
        h1 {font-family: Arial; font-style: italic}
        </style>
    </head>
 <body>
<h1>Ma première page en HTML</h1>


</body>
</html>
```

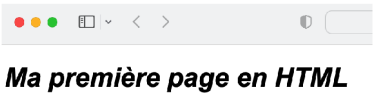


FIGURE 3.3 – Scripts CSS interne

Remarque

Néanmoins, cette méthode nécessite d'inclure le code du style à l'entête de chaque fichier, et de le répéter pour tous les fichiers.

3.4.3 Méthode du CSS externe

Dans cette méthode, un fichier séparé est utilisé, indépendant du code HTML, et qui porte l'extension *.css*. Ce fichier contient tous les styles CSS utilisés dans toutes les pages. Une simple ligne (toujours la même) servira à appeler le fichier *.css* entre les tags $< head >$ et $< /head >$.

Exemple

Dans cet exemple, nous allons utiliser deux fichiers, le premier *Style.css* et le deuxième *Exemple.html*. Dans l'entête du fichier *Exemple.html*, il faut rajouter une ligne pour inclure le fichier *Style.css* voir la figure 3.4.

- La balise $< LINK >$ avertit le navigateur qu'il faudra réaliser un lien ;
- L'attribut classique de lien $href =$ "*Style.css*" donne le chemin d'accès et le nom du fichier à lier ;
- L'attribut $rel = stylesheet$ (sans guillemets) précise qu'il y trouvera une feuille de style externe ;
- L'attribut $type =$ "*text/css*" précise que l'information est du texte et du genre cascading style sheets (css) ;

— L'attribut média permet de spécifier le type de média auquel le style sera appliqué. Le média peut aussi bien être l'ordinateur qu'un téléphone portable, une imprimante, etc.

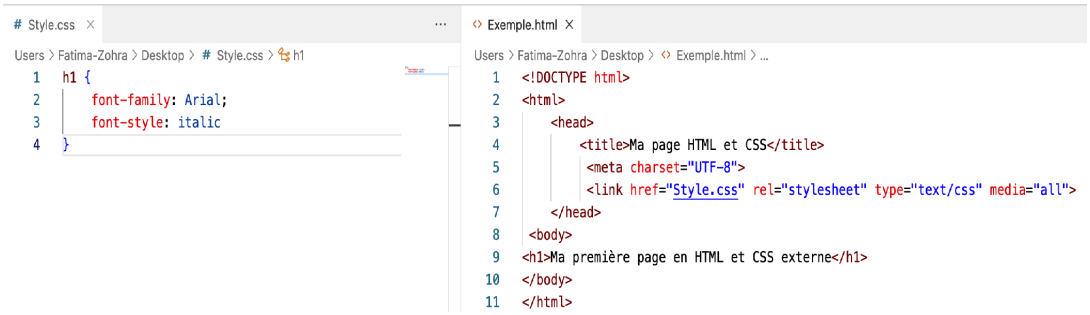


FIGURE 3.4 – Scripts CSS Externe

Remarques

Cette méthode présente divers avantages par rapport au deux précédentes :

— Les définitions de style ne sont faites qu'une seule fois, à un seul endroit, cela vous réduit considérablement la taille de la page HTML ;
— Les navigateurs peuvent mettre en cache le contenu de la feuille de styles qui s'appliquera sur toutes les pages du site. Une fois cette mise en cache effectuée, le navigateur n'aura plus qu'à charger le contenu de la page. D'où une forte réduction du temps de transfert ;
— Cela facilite la maintenance du site ;
— Cela donne une meilleure structuration de la page puisque que le contenu lui-même est séparé de la forme.

3.5 Les bases du CSS

Dans cette section, nous donnons les éléments de bases pour bien débuter en css.

3.5.1 Sélecteurs et propriétés

Le sélecteur détermine ou identifie les parties du document auxquelles sera appliqué le style. Il sert à sélectionner l'élément que l'on a envie de mettre en forme.

Le bloc de déclaration, quant à lui, contient les déclarations des styles à appliquer. Il est toujours mis entre deux accolades voir la figure 3.5.

Dans un bloc de déclaration, nous listons tous les styles que l'on veut utiliser pour l'élément sélectionné. Ce bloc est composé de déclarations qui elles-mêmes contiennent une propriété et une valeur.

La propriété correspond à l'attribut que nous souhaitons changer. A chaque propriété peut être affectée une valeur.Une propriété et une valeur sont toujours séparées par le symbole "deux points" (:),

Exemple

Dans cet exemple, Le style défini entre accolades sera appliqué à la balise body. color est une propriété qui spécifie la couleur, red est une valeur de la propriété qui signifie rouge.

FIGURE 3.5 – Exemple de sélecteur CSS

Remarques

— Si la valeur est composée de plusieurs mots, il serait mieux de le mettre entre guillemets ;
— Un bloc peut contenir plusieurs déclarations. Dans ce cas, les déclarations sont obligatoirement séparées par des points virgules (;).

3.5.2 Les commentaires

Il est très conseillé de mettre des commentaires dans un code CSS. ils permettent à d'autres utilisateurs de comprendre le code facilement en cas de travail en groupe.

Exemple

$$/ * ceciestuncommentaire * /$$

3.5.3 Les attributs HTML class et id

Les sélecteurs de classe CSS permettent de cibler des éléments d'un document en fonction du contenu de l'attribut *class* de chaque élément. L'attribut *class* est une liste de termes séparés par des espaces, il est nécessaire qu'un de ces termes corresponde exactement au nom utilisé dans le sélecteur pour que l'élément soit ciblé.

Syntaxe

$$.nomdeclasse\ \{déclarations\ CSS\}$$

Exemple

La figure 3.6 donne un exemple de classe css.

FIGURE 3.6 – Exemple de classe css

Remarques

— La définition d'un style était : *balisepropriétédestyle* : *valeur* ;
— Elle devient : *balise.nomDeClasse { propriété de style : valeur }* ;
— Ou, comme la mention de la balise est facultative,
.nomDeClassepropriétédestyle : *valeur* ;
— Pour appeler l'effet de style dans le document, nous devons rajouter le nom de la classe à la balise
$< baliseclass = "nomDeClasse" > .... < /balise >$.

Contrairement aux classes, un id s'applique à un objet unique. Il n'est pas possible d'avoir deux éléments de même id dans une même page.

Autrement dit, si un élément HTML possède déjà un id, cet id ne peut plus être utilisé pour un autre élément HTML.

La définition d'un ID en CSS est identique à celle d'une classe, excepté le dièse qui précède le sélecteur.

Syntaxe

#nomDeID {propriété de style : valeur} Et pour l'appeler :

<balise id="nomDeID"> .... </balise>

Les sélecteurs peuvent-être regroupés. Pour cela, les éléments doivent-êtres listés et séparés par une virgule.

Exemple

L'exemple donné à la figure 3.7 montre comment groupés un ensemble de sélecteurs.

```
1  h1, h2, h3, h4, h5, h6 {
2      color: green
3  }
4
```

FIGURE 3.7 – Exemple de sélecteurs groupés

Les sélecteurs peuvent-êtres de types différents. Tel qu'il est possible de définir un style à plusieurs éléments, classes, identifiants.

Exemple

L'exemple donné à la figure 3.8 montre comment groupés un ensemble de sélecteurs de types différents.

```
1  h1, .uneclasse, # idspecial {
2      color : red;
3      font-weight : bold;
4      }
```

FIGURE 3.8 – Exemple de sélecteurs de types différents

Héritage

Chaque élément HTML peut avoir chacun soit sa propre classe, soit son propre style. Les balises HTML peuvent aussi s'emboîter les unes dans les autres (voir les figures 3.9 et 3.10).

Exemple 1

Cet exemple montre un ensemble d'éléments avec leurs différentes classes css.

```
<body class="stylebody">
    <p class= "stylep"> </p>
</body>
```

FIGURE 3.9 – Exemple d'éléments avec leurs différentes classes

Question

Quel style sera appliqué au contenu de la balise <p> ?

Nous appellerons "parent", le style du niveau supérieur (le stylebody dans notre cas) et enfant les styles de la balise en dessous (stylep dans notre cas).

La balise "body", avec ses différents styles, se voit hériter des styles de la classe transmise.

Remarque

S'il y a deux valeurs différentes pour la même propriété, l'enfant garde la sienne.

Exemple 2

À travers l'exemple de la figure 3.10, nous donnons un exemple de code complet HTML et CSS.

Exemple.html

Users > Fatima-Zohra > Desktop > Exemple.html > ...

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Ma page HTML et CSS</title>
5          <meta charset="UTF-8">
6          <link href="Style.css" rel="stylesheet" type="text/css"
7      </head>
8      <body class="stylebody">
9          <p>Voici un paragraphe sans style, il hérite donc entièr
10         <p class= "stylep">Un paragraphe avec style, qui hérite
11
12 </html>
13
```

Style.css

Users > Fatima-Zohra > Desktop > Style.css > ...

```
1  .stylebody {
2      font-size: 10pt;
3      font-family: arial, verdana, sans-serif;
4      color: green;
5      text-align: left;
6      background-color: #FFFFFF ;
7  }
8  .stylep {
9      background-color: yellow;
10     font-style:italic;
11     text-align:center;
12 }
13
```

FIGURE 3.10 – Exemple complet html et css

3.5.4 Ordre des priorités en CSS

Les styles css sont définis à plusieurs niveaux, dans l'ordre, du moins au plus prioritaire :

- Le sélecteur le plus précis imposera ses styles aux sélecteurs moins précis en cas de conflit.
- La « précision » désigne ici le fait d'identifier de manière plus ou moins unique un élément. Les sélecteurs peuvent être rangés dans l'ordre suivant (du plus précis au moins précis) :
  - Un style défini dans un attribut HTML style sera toujours le plus précis et notamment plus précis qu'un style défini avec un sélecteur CSS ;
  - Le sélecteur # id va être le sélecteur le plus précis mais sera moins précis qu'un style défini dans un attribut HTML style ;
  - Un sélecteur *.class* ou un autre sélecteur d'attribut ou un sélecteur de pseudo-classe sera moins précis qu'un sélecteur # id ;
  - Un sélecteur d'élément ou de pseudo-élément sera moins précis qu'un sélecteur d'attribut ou de pseudo-classe.
- Si deux sélecteurs différents sont au même degré de précision, alors c'est le sélecteur le plus «complet» c'est-à-dire celui qui utilisera le plus de combinateurs qui sera jugé le plus précis.
- Le mot clef *!important* sert à forcer l'application d'une règle CSS. La règle en question sera alors considérée comme prioritaire sur toutes les autres déclarations et le style sera appliqué à l'élément concerné.

3.5.5 Les éléments div et span (conteneurs génériques)

L'élément HTML < *div* > (ou division) est le conteneur générique du contenu du flux, en HTML, est un conteneur d'éléments. L'ensemble des éléments renseignés à l'intérieur des balises HTML < *div* >< /*div* > constitue une section à laquelle appliquer une mise en forme spécifique décrite avec du code CSS. Identifier un div HTML à l'aide d'un attribut permet de le cibler avec CSS, pour appliquer systématiquement la mise en forme correspondante. Le conteneur d'éléments permet ainsi de structurer plus facilement la page Web. Le div a un rôle structurant. C'est un élément de type *block*. Le plus souvent il est utilisé comme un conteneur pour encapsuler d'autres éléments.

L'élément HTML < *span* > est un conteneur générique en ligne (inline) pour les contenus phrasés. Il peut être utilisé pour grouper des éléments afin de les mettre en forme (grâce aux attributs class ou id et aux règles CSS) ou parce qu'ils partagent certaines valeurs d'attribut comme lang. Il doit uniquement être utilisé lorsqu'aucun autre élément sémantique n'est approprié. < *span* > est très proche de l'élément < *div* >, mais l'élément < *div* > est un élément de bloc, alors que < *span* > est un élément en ligne.

Exemple

La figure 3.11 donne un exemple d'utilisation des éléments *div* et *span*.

HTML

```
1  <html>
2      <head>
3          <title>Ma page HTML</title>
4          <meta charset="utf-8">
5      </head>
6      <body>
7       <div>Élément div permet de définir un bloc de contenu non typé a
   priori</div>
8          <p><span>L'élément span fonctionne pour les éléments de texte
   inline, à l'intérieur d'un paragraphe de texte (ou d'un div)</span></p>
9      </body>
10 </html>
```

FIGURE 3.11 – Exemple d'utilisation des éléments div et span

3.5.6 Les niveaux block et inline

Il existe deux grands types d'affichage principaux pour les éléments HTML : un élément HTML va pouvoir être soit de niveau (ou de type) block, soit de niveau (ou de type) inline.
Ces types d'affichage vont définir la façon dont les éléments vont se comporter dans une page par rapport aux autres et la place qu'ils vont prendre dans la page.

Le type d'affichage d'un élément va toujours être défini en CSS par la propriété *display*. Si cette propriété n'est pas explicitement renseignée pour un élément, c'est la valeur par défaut de display qui va être appliquée à l'élément c'est-à-dire *display* : *inline*.

Parmi les styles par défaut appliqués par n'importe quel navigateur se trouve la définition du type d'affichage ou du display pour chaque élément. Aujourd'hui, la plupart des navigateurs suivent les recommandations du W3C. Il spécifie pour chaque élément HTML quelle devrait être la valeur de son display.

— display : block : affichage sous forme d'un bloc ;
— display : inline : affichage en ligne ;
— display : none : l'élément n'est pas affiché.

Éléments de type inline

On appellera « élément de type inline » (ou « inline level element » en anglais) un élément auquel a été appliqué un display : inline.
Les éléments de type inline vont posséder les caractéristiques suivantes qui vont les différencier des éléments de type block :

— Un élément de type inline ne va occuper que la largeur nécessaire à l'affichage de son contenu par défaut ;
— Les éléments de type inline vont venir essayer de se placer en ligne, c'est-à-dire à côté (sur la même ligne) que l'élément qui les précède dans le code HTML ;
— Un élément de type inline peut contenir d'autres éléments de type inline mais ne devrait pas contenir d'éléments de type block.

Éléments de type bolck

on appellera « élément de type block » (ou « block level element » en anglais) un élément auquel on va appliquer un display : block.
Les éléments de type block vont posséder les caractéristiques de disposition suivantes :

— Un élément de type block va toujours prendre toute la largeur disponible au sein de son élément parent (ou élément conteneur) ;
— Un élément de type block va toujours « aller à la ligne » (créer un saut de ligne avant et après l'élément), c'est-à-dire occuper une nouvelle ligne dans une page et ne jamais se positionner à côté d'un autre élément par défaut ;
— Un élément de type block peut contenir d'autres éléments de type block ou de type inline.

3.5.7 Notations long hand et short hand

Une notation raccourcie ou notation « short hand » ou encore « propriété raccourcie / short hand » correspond à une propriété à laquelle les valeurs d'un ensemble de propriétés seront passés et qui va donc nous permettre de définir de valeur de plusieurs propriétés en même temps.

La propriété *border* qui correspond en fait à la notation raccourcie des propriétés $border-width$, $border-style$ et $border-color$ et qui permet ainsi de définir d'un coup les valeurs pour ces trois propriétés pour un élément.

De manière générale, il est équivalent de préciser le comportement d'un aspect d'un élément HTML en utilisant une notation raccourcie ou en utilisant les propriétés long hand voir la figure 3.12.

Exemple

Cet exemple donne un aperçu sur l'utilisation de la notation long et short hand.

```
1  /*Notation long hand*/
2  #p1{
3      border-width: 2px;
4      border-style: solid;
5      border-color: red;
6  }
7  /*Notation short hand équivalente*/
8  #p2{
9      border: 2px solid red;
10 }
```

FIGURE 3.12 – Exemple de notations long et short hand

3.5.8 Mise en forme du texte

Le CSS permet dans sa globalité de créer le « design » d'un site ou application Web tandis que le HTML lui s'occupe de l'implémentation de la structure du site. Parmi les diverses possibilités de mise en forme d'un site Web, il existe celles de formatage des textes.

Les propriétés CSS liées au texte peuvent être séparées en deux grandes catégories :

— Les propriétés de type $font-$ qui vont définir l'aspect des caractères en soi en qui agissent directement sur la police d'écriture (choix de la police, de la taille des caractères ou de leur poids entre autres) ;

— Les propriétés de type $text-$ qui ne vont pas impacter directement l'aspect des caractères du texte mais permettent d'ajouter des effets de style autour de celui-ci (alignement du texte, soulignement ou encore ajout d'ombres autour des textes par exemple).

Font-family

L'un des éléments le plus important du webdesign. La propriété CSS $font-family$ permet de définir la police des textes (le rendu graphique de chaque caractère).

Font-weight

La propriété $font-weight$ permet de contrôler le mode gras du texte sans passer par la balise $< span >$ tels que :

— *normal :* Niveau de graisse regular ;
— *bold :* Niveau de graisse gras ;
— *lighter :* Diminue d'un niveau la valeur de graisse ;
— *bolder :* Augmente d'un niveau la valeur de graisse.

Font-style

La propriété $font-style$ permet d'utiliser une version italique de la police de caractère où, à défaut, de pencher artificiellement le texte.

Font-size

La propriété $font-size$ permet de définir une taille au texte. Il est possible d'utiliser plusieurs unités :

— *Les pixels :* ils permettent de définir une valeur fixe, absolue et identique sur tous les navigateurs ;
— *Les ems :* ils permettent de définir une valeur dynamique basée sur la taille de l'élément parent ;
— *Les rems :* ils permettent de définir une valeur dynamique basée sur la taille de l'élément $< html >$.

La propriété $text-$ complète $font-$ en permettant de donner plus de valeur au texte.

Text-indent

La propriété $text-indent$ permet d'ajouter un espace horizontal (retrait) au début de la première ligne d'un paragraphe de texte.

Il est possible d'utiliser plusieurs unités :

— Le pixel ;
— ems ;
— Un pourcentage.

Text-align

La propriété $text-align$ permet de contrôler l'alignement du texte tels que : gauche, droite, centré et justifié.

Text-decoration

La propriété $text-decoration$ permet de décorer le texte avec une ligne placée sous, sur ou en travers le texte.

Text-transform

La propriété $text-transform$ permet de modifier la façon dont les majuscules sont gérée. Le texte peut-être affiché en minuscules, en majuscules ou bien avec une majuscule en début de chaque mot. Les valeurs possibles sont : *capitalize*, *uppercase* et *lowercase*.

Exemple récapitulatif

La figure 3.13 montre un exemple d'utilisation des propriétés $font-$ et $text-$.

FIGURE 3.13 – Exemple d'utilisation des propriétés $font-$ et $text-$

3.5.9 Modèle des boites

Le modèle de boîte définit comment chaque paramètre de la boîte fonctionne pour former l'aspect final affiché à l'écran (voir la figure 3.14).

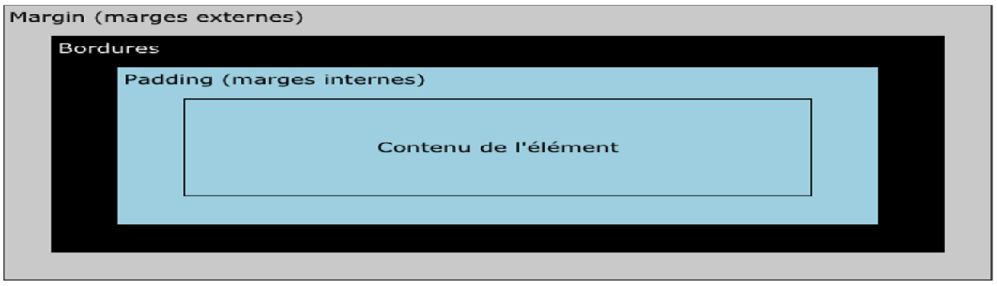


FIGURE 3.14 – Structure générale d'une boite en CSS

Une boîte se décompose en quatre éléments :

— **Le contenu :** Cela correspond au texte, images ou autres éléments HTML enfants présents à l'intérieur de la boîte en question ;

— **Les bordures :** Les bordures correspondent aux traits qui encadrent la boîte. Manipulables avec la propriété CSS *border* et les propriétés associées ;

— **Les marges externes :** Elles permettent de déterminer une zone vierge autour de la boîte qui permet ainsi d'espacer les boîtes les unes par rapport aux autres. Manipulables avec la propriété CSS *margin* et les propriétés associées ;

— **Les marges internes :** Il s'agit d'une d'une zone vierge créée à l'intérieur de la boîte pour espace le contenu des bordures de la boîte en question. Manipulables avec la propriété CSS *padding* et les propriétés associées.

Les dimensions d'une boîte

Les règles de dimensionnement ne s'appliquent qu'au modèle de boîte standard.

Les propriétés *width* et *height* sont utilisés pour dimensionner une boîte, les valeurs indiquées s'appliquent au contenu de la boîte. Ces dimensions ne correspondent donc pas à l'espace réellement occupé par la boîte à l'écran (voir la figure 3.15).

La taille réelle d'une boîte, telle qu'elle sera perçue à l'écran, correspond à la somme des dimensions de trois éléments constituant la boîte : le contenu, la bordure et les marges internes.

L'espace occupé par une boîte correspondrait à la taille réelle à laquelle sera ajouté les marges externes.

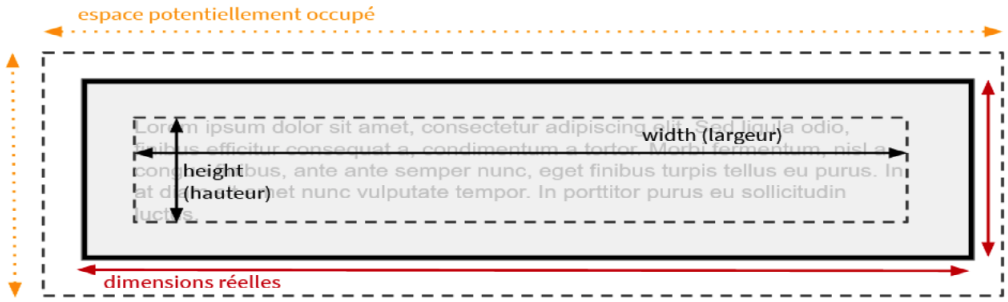


FIGURE 3.15 – Dimensions d'une boîte

Exemple

Le code montré à la figure 3.16 permet de définir les dimensions d'une boite grâce aux paramètres *width*, *height* et *border*.

```
1  .exempleBoite {
2      width: 500px;
3      height: 300px;
4      margin: 15px;
5      padding: 10px;
6      border: 3px solid red;
7  }
```

FIGURE 3.16 – Exemple d'utilisation des boites

L'impact du type d'affichage d'un élément sur ses dimensions

Les dimensions par défaut du contenu des éléments HTML vont avant tout être déterminées par le type d'affichage des éléments : en effet, les éléments de type *block* occuperont par défaut toute la largeur disponible dans leur élément parent tandis que les éléments de type *inline* n'occuperont que la largeur nécessaire à leur contenu.
Le principe de base d'un élément de type *inline* est que l'espace pris par sa boite « contenu » soit toujours égal à l'espace strictement nécessaire à l'affichage de ce contenu.

Gestion des marges internes ou padding

Les marges intérieures se trouvent entre le contenu de l'élément et sa bordure. Ainsi, définir une marge intérieure importante va éloigner la bordure

de l'élément de son contenu. La propriété *padding* est la version raccourcie des propriétés $padding-top$, $padding-left$, $padding-bottom$ et $padding-right$ qui vont servir à définir les marges internes de chaque côté d'un élément. Ces propriétés vont pouvoir accepter deux types de valeurs (voir la figure 3.17).

— Des valeurs de type longueur, généralement en px ou en em ;
— Des valeurs de type pourcentage. Dans ce cas, le % indiqué est calculé par rapport à la taille de l'élément parent.

Exemple

L'exemple donné à la figure 3.16 montre l'utilisation des marges internes paddings.

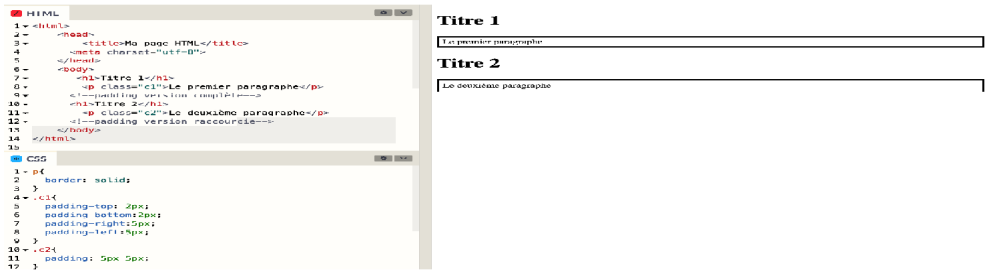


FIGURE 3.17 – Exemple d'utilisation des marges internes paddings

Les caractéristiques des bordures

Les bordures vont être définies par trois caractéristiques en CSS : une épaisseur (ou largeur), un style et une couleur (voir la figure 3.17).

— La propriété $border-width$ permet de définir la largeur (ou « l'épaisseur) d'une bordure ;
— La propriété $border-style$ permet de définir le style d'une bordure ;
— La propriété $border-color$ permet de définir la couleur d'une bordure.

Exemple

Cet exemple à la figure 3.18 montre l'utilisation des bordures en css.

FIGURE 3.18 – Exemple d'utilisation des bordures

La propriété box-sizing

La propriété CSS box-sizing définit la façon dont la hauteur et la largeur totale d'un élément est calculée.

La propriété box-sizing peut être utilisée afin d'ajuster ce comportement (voir la figure 3.18) :

— **content-box** valeur par défaut. Les dimensions définies pour l'élément vont s'appliquer à sa boite de contenu. Toute marge interne ou bordure ajoutées ensuite vont augmenter la taille de l'élément ;

— **border-box** indique au navigateur de prendre en compte la bordure et le remplissage dans la valeur définie pour la largeur et la hauteur.

Exemple

La figure 3.19 donne un exemple d'utilisation de la propriété box-sizing.

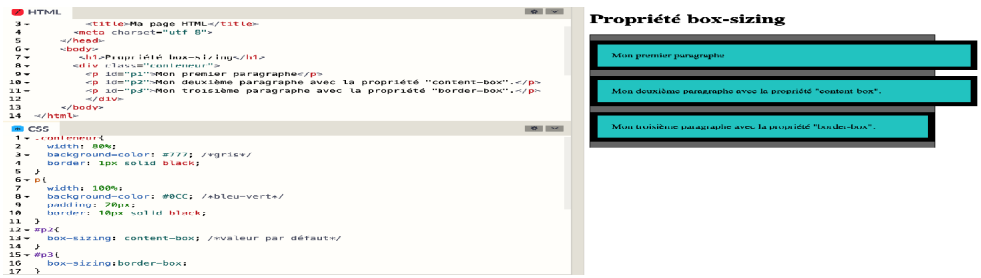


FIGURE 3.19 – Exemple d'utilisation de la propriété box-sizing

3.5.10 Position et affichage des éléments

La propriété *position* est une propriété CSS puissante qui permet de définir un type de positionnement pour les éléments de la page Web. Un élément peut-être positionné relativement à partir de sa position par défaut

ou de façon absolue par rapport à un point donné dans la page en utilisation position conjointement avec les propriétés *top*, *left*, *bottom* et *right*.

La propriété position ne va pas nous permettre de positionner un élément en soi dans une page mais simplement de définir un type de positionnement grâce aux valeurs suivantes :

1. **Static :** La valeur static est la valeur par défaut de la propriété position. Ainsi, par défaut, tous les éléments HTML sont positionnés de manière static. Un élément HTML positionné avec *position* : *static* sera positionné selon le flux normal de la page.
2. **Relative :** Attribuer une position : relative à un élément va positionner l'élément dans le flux normal de la page tout comme la valeur static de la propriété position : static. Cependant, à l'inverse d'un élément HTML positionné avec *position* : *static*, un élément positionné avec position : relative va ensuite pouvoir être décalé par rapport à sa position initiale grâce aux propriétés top, left, bottom et right.
3. **Absolute :** Un élément positionné avec *position* : *absolute* va être positionné par rapport à son parent le plus proche positionné (avec une valeur de position différente de static). Si aucun parent positionné n'est trouvé, alors l'élément sera positionné par rapport à l'élément racine représentant la page en soi.
4. **Fixed :** Le positionnement fixe est très proche du positionnement absolu. Un élément positionné avec *position* : *fixed* va également être retiré du flux de la page et l'espace qui lui était attribué selon le flux normal de la page va également pouvoir être utilisé par d'autres éléments.

Exemple

La figure 3.20 donne un exemple d'utilisation des propriétés de position.

FIGURE 3.20 – Exemple d'utilisation des propriétés de position

Remarque

Le flux d'un document pourrait se définir comme étant le comportement naturel d'affichage des éléments d'une page Web. Les éléments se succèdent dans l'ordre où ils sont déclarés dans le code HTML.

L'ordre d'affichage des éléments en cas de chevauchement avec la propriété z-index

Les éléments HTML vont pouvoir être positionné dans une page en CSS selon 3 dimensions : selon la largeur (axe horizontal ou axe des X en math), la hauteur (axe vertical ou axe des Y) et également selon une épaisseur ou un ordre d'empilement (axe des Z).
La propriété $z-index$ permet de modifier ce comportement et de choisir quel élément doit apparaitre au-dessus de quel ordre en donnant un index sous forme de nombre à un ou plusieurs éléments. Ainsi, lorsque deux éléments se chevauchent, celui possédant la plus grande valeur pour son $z-index$ apparaitra au-dessus de l'autre.
La propriété CSS $z-index$ ne va fonctionner qu'avec des éléments HTML positionnés, c'est-à-dire qu'avec des éléments possédant une propriété position dont la valeur est différente de *static* en CSS.

Exemple

La figure 3.21 donne un exemple d'utilisation des propriétés de position $z-index$.

FIGURE 3.21 – Exemple d'utilisation des propriétés de position avec $z-index$

3.6 Exercice applicatif

Répondre aux questions suivantes :

1. La balise HTML qui spécifie un style CSS intégré dans un élément est appelée ?
   (a) Design ;
   (b) Style ;
   (c) Modify ;
   (d) Define.
2. Quelle propriété css vous utiliserez si vous voulez ajouter une marge entre la bordure d'une DIV et son texte intérieur ?
   (a) Spacing ;
   (b) Margin ;
   (c) Padding ;
   (d) inner-margin.
3. La valeur par défaut de l'attribut « position » est ?
   (a) Fixed ;
   (b) Absolute ;
   (c) Inherit ;
   (d) Relative.

Deuxième partie:

Travaux Pratiques et Examens types

Cette partie contient des exemples d'exercices de travaux pratiques, ainsi qu'un ensemble d'examens types.

Programmation Web

TP1 – HTML

Démarche générale

- Pour ce TP, je vous recommande d'utiliser des outils simples comme Blocnote.
- Une fois que vous aurez intégré ces principes, vous pourrez facilement utiliser des outils plus avancés, ou des logiciels commerciaux tels que Dreamweaver.
- N'utilisez pas un logiciel de traitement de texte, tels que Microsoft Word ou OpenOffice, car ils produisent des fichiers qu'un navigateur Web ne sait pas lire.
- Créez un répertoire « html » dans lequel vous placerez les fichiers de ce TP.
- Créez un fichier index.html que vous sauvegarderez dans votre répertoire et commencez à le remplir, avec sa déclaration de document, son en-tête et le corps du document.
- Vérifiez l'encodage de votre page : attention à déclarer dans votre document Html que l'encodage est UTF-8.
- Pour voir ces changements dans votre navigateur il faut rafraîchir (actualiser) la page ; vous pouvez utiliser une de ces méthodes : (cliquer sur la flèche ronde de votre navigateur, appuyer sur la touche F5, appuyer sur les touches (ctrl+r).
- Testez la validité de votre page (et corrigez votre html si besoin) avec le validateur du W3c (http://validator.w3.org/#validate).

Exercice : Premiers pas en HTML

1. Écrivez l'extrait de code HTML qui réalise ce titrage dans un fichier nommé Pagel.html.

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Titre de niveau 4

2. Rajoutez l'extrait de code HTML qui réalise ceci (utilisez les balises <b>, <i>, <u>).

Texte en **gras**, texte en *italique*, texte souligné.

3. Rajoutez l'extrait de code HTML qui réalise ce lien vers une autre page : actualite.html (créez la page actualite.html dans le même répertoire que la page courante) :

Lien vers l'actualité.

4. Rajoutez l'extrait de code HTML qui réalise ce lien vers une adresse électronique (dans la pratique, cela ouvre le client de courrier électronique, en général Outlook) :

Écrivez-moi à mon adresse contact@mail.com.

5. Rajoutez l'extrait de code HTML qui combine les deux types de liste :

1. Premier élément ordonné
2. Deuxième élément ordonné
    - Premier élément non-ordonné
    - Deuxième élément non-ordonné
3. Troisième et dernier élément ordonné

Programmation Web

TP2 – HTML

Démarche générale

1. Dans votre répertoire, créez un nouveau sous-dossier que vous nommerez « TP2 ».
   En vous basant sur l'exercice du TP1, créez 4 pages html que vous nommerez :
   - index.html
   - presentation.html
   - realisation.html
   - contact.html
2. Pour chacune des pages, rajoutez des titres en exploitant les balises h1 et h2 pour les titres et les sous-titres ;
3. Modifiez les titres et les textes en y insérant des balises des mises en forme (testez les mises en forme proposées ci-dessous). Pensez à enregistrer régulièrement votre travail. Visualisez le résultat dans votre navigateur ;
4. Créez un dossier "images" dans votre sous-dossier « Fiche 2 » ;
5. Téléchargez puis enregistrez des images de votre choix dans le dossier « images » ;
6. Dans chaque page, insérez la balise <img/> en pointant la source vers l'image correspondante ;
7. Enregistrez et visionnez le résultat dans votre navigateur Internet ;
8. La page "index.html" servira de page de regroupement. Dans cette page, ajoutez un menu pointant vers chacune des pages du site. Vous nommerez les titres du menu du même nom que leurs pages correspondantes (Présentation, Réalisations, Contact) ;
9. Sur la page "presentation.html", vous insérerez un lien sur le mot *contactez-nous* vers le page "contact" ;

10. Sur la page "realisation.html", vous insérerez un lien vers le site de l'école (http://ens-oran.dz).
11. Sur la page Contact, insérerez un lien vers le page « realisation.html » sur le texte « nos réalisations » ;
12. Sur chaque page, vous ajouterez un bouton retour, redirigeant le visiteur vers la page d'index du site.
13. Insérez ce tableau sur votre page contact.html.

ADRESSE :	
TELEPHONE :	
MAIL :	

Programmation Web

TP3 – HTML

Exercice 1

1. Dans votre répertoire, créez un nouveau sous-dossier que vous nommerez "TP3".
2. Écrire une page HTML qui affiche :
   - Un titre au niveau du bandeau de la fenêtre du navigateur ;
   - Le même titre dans le document ;
   - Un paragraphe ;
   - Le document devra être valide conformément aux exigences du W3C ;
   - Le document devra être écrit en UTF-8 ;
   - Le document devra contenir des caractères accentués ;
3. À partir de la structure de la page de l'exercice précédent écrivez une page HTML utilisant un autre type d'encodage :
   - L'encodage (charset) sera ISO-8859-15 ;
   - Nous conserverons le mode d'édition en UTF-8 ;
   - Nous ajouterons un mot contenant des caractères accentués sous forme d'entités HTML et sous forme directe de caractères UTF-8 (par exemple é et é). Au chargement de la page que remarquez-vous ?

Exercice 2

Dans le même répertoire, créer un nouveau document « exercice 2_TP2.html »

1. Changer la couleur du fond du document en #00ffff.
2. Placer le titre Corrigé des exercices de HTML centré en haut de la page.
3. Insérer une ligne horizontale de largeur 80% et d'épaisseur 2.
4. Placer le sous-titre Exercice 2 aligné à gauche.

Programmation Web

TP4 – HTML

Démarche générale

1. Dans votre répertoire, créez un nouveau sous-dossier que vous nommerez « TP3 ».
2. En vous basant sur l'exercice du TP1, créez 1 page html que vous nommerez :

   - Form.html

3. Reproduisez l'apparence de la page (formulaire.htm) avec les éléments HTML vus dans le cours.

4. Programmez les boutons *Soumettre formulaire* pour envoyer le contenu du formulaire par email en utilisant la méthode (METHOD="POST"), le bouton *Effacer* qui permet l'initialisation du formulaire (vider les champs de saisies).
5. Enregistrez votre travail, puis visualiser le résultat sous un navigateur Web.

Programmation Web

TP5 – CSS

Démarche générale

1. En se basant sur le code en bas ;
2. Dans votre répertoire, créer puis nommer le nouveau sous-dossier « TP5 » ;
3. Changer la couleur du texte de tous les paragraphes ;
4. Changer la couleur de fond et ajouter une marge intérieure aux éléments de classe *maclasse*;
5. Ajouter une bordure en trait continu, une marge intérieure et une marge extérieure à l'élément d'identifiant *monid* ;
6. Définir la couleur du texte des sections en vert et de marge d'au moins 100 pixels ;
7. Définir la taille de police des titres 1 à 2.5em et petites majuscules ;
8. Définir la taille de police des paragraphes à 14px ;
9. Définir la taille de police des titres 2 et la police de caractère en Helvetica ;
10. Changer la couleur du texte de tous les paragraphes en bleu ;
11. Changer la couleur du texte des éléments de classe *maclasse* en vert ;
12. Changer la couleur du texte des éléments d'identifiant *superid_en* rouge.

Code

```
<section>
  <h1>Mon titre 1</h1>
  <article>
    <h2>Mon Titre 2</h2>
    <p>Premier paragraphe</p>
    <p class="maclasse">Deuxième paragraphe</p>
    <p>Troisième paragraphe</p>
    <p id="monid">Quatrième paragraphe</p>
  </article>
  <article>
    <h2 class="maclasse">Mon Titre 3</h2>
    <p id="superid" class="maclasse">Cinquième paragraphe</p>
    <p>Sixième paragraphe</p>
  </article>
</section>
```

Programmation Web

TP6 – CSS

Démarche générale

1. Dans votre répertoire, créer puis nommer le nouveau sous-dossier « TP6 » ;
2. Écrire une page HTML avec titre « Formulaire CSS » ;
3. Cette page contient un formulaire avec les contraintes suivantes :
   - Utilisation de la méthode Get ;
   - Le nom de la page cible est défini grâce à l'attribut action ;
   - Présente une zone de saisie pour un identifiant (input de type text qui aura automatiquement le focus dès l'ouverture de la page) ;
   - Présente une zone de saisie pour un mot de passe ;
   - Présente un champ caché nommé opérations ;
   - Présente un champ numéro de téléphone qui commence obligatoirement avec le chiffre 9 et n'accepte les chiffres de zéro à 9 ;
   - Présente un bouton pour télécharger les fichiers ;
   - Présente un bouton d'envoi ;
   - Présente un bouton annuler.
4. Les textes du formulaire sont en Helvetica, la largeur globale du formulaire est de 400px, il doit être centré dans la page et écarté du bord supérieur de la page de 100px.
5. Des inputs avec une bordure noire continue de 1 pixel, une largeur fixée à 200px et faire en sorte que les champs soient espacés les uns des autres.
6. Des labels de 100px de large à gauche des champs de saisie.
7. Le bouton d'envoi doit avoir une couleur de fond modifiée, faire 400px de large et l'écriture doit être en gras au centre du bouton.

Programmation Web

TP7 – CSS

Démarche générale

1. Dans votre répertoire, créer puis nommer le nouveau sous-dossier « TP7 » ;
2. Créer une page en HTML **index.html** ;
3. Respecter l'encodage UTF-8 ;
4. Complétez la page « index.html » afin qu'elle contienne les éléments suivants :

- Le titre de la page "Accueil" doit se trouver en titre sur le navigateur ;
- Un paragraphe de présentation de la page expliquant son objectif, une partie de ce texte devra être mise en valeur en italique ;
- Un tableau de trois colonnes et au moins 4 lignes présentant plusieurs éléments.
- Ce tableau doit comporter la légende "**Cours Technologies Web**", et avoir une ligne d'entêtes comportant "**Prénom & Nom des étudiants**" en première colonne, "**HTML**" en deuxième colonne, et "**CSS**" en troisième colonne.
- Remplissez le tableau par les informations correspondantes ;
- Un lien vers une autre page, page « notes.html » ;
- Créer une feuille de style CSS "**style.css**" qui sera appelée dans les fichiers Html afin de prendre en compte les éléments suivants de présentation :
  i. Les titres principaux doivent être de couleur bleue (#58acfa par exemple) et au centre de la page ;
  ii. Le tableau doit :
    a. Comporter une bordure visible bleue ;
    b. Occupera toute la zone horizontale d'affichage,

c. Les cellules d'entête seront centrées et écrites en gras et en blanc sur fond bleu foncé ;
d. Les cellules du corps seront en gris clair et en texte justifié,
e. La légende sera positionnée en bas du tableau avec une couleur noire positionnée au centre de la page.

iii. Faire en sorte que la police utilisée dans le paragraphe à l'aide de la règle @fontface).

a. La règle @font-face permet de définir les polices d'écriture à utiliser pour afficher le texte de pages web. Cette police peut être chargée depuis un serveur distant ou depuis l'ordinateur de l'utilisateur.
b. Si la fonction local() est utilisée, elle indique à l'agent utilisateur de prendre en compte une police présente sur le poste de l'utilisateur.
c. Exemple :

```
@font-face { font-family:'ma-police-ttf'; src:local('ma-police'), url('ma_police.ttf')

format('truetype');

}
```

iv. Les liens doivent s'afficher en **vert** lorsque l'utilisateur passe sa souris dessus avec une taille de police double ;
v. Le texte mis en valeur doit s'afficher en rouge.

Programmation Web
TP8 – BOOTSTRAP

Bootstrap est une collection d'outils HTML, CSS et JavaScript destinés à aider les développeurs de sites et applications web.

- Faciliter la maintenance du site (code normalisé donc très facile à maintenir);
- Garantir un aspect unique au site quel que soit le navigateur utilisé (compatibilité entre navigateurs) ;
- D'avoir un site responsif qui s'adapte automatiquement en fonction de l'écran/périphérique (Smartphone, Tablette, Ordinateur).

Exercice 1

Créer un projet Bootstrap simple:

1. Télécharger Bootstrap4 ;
2. Organiser votre projet ;
3. Écrire la page index.html de base avec les différents liens vers les fichiers :
   <link href="css/bootstrap.min.css" rel="stylesheet">

- La balise meta permettant à Bootstrap 4 d'analyser la largeur de l'écran :
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

- Avant de fermer la balise </body> ajouter :

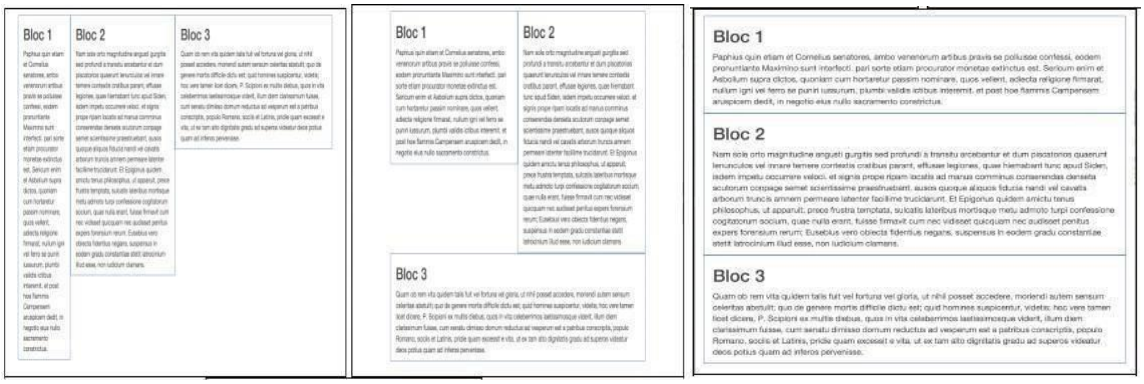
<script src="js/jquery.js"></script> , <script src="js/bootstrap.min.js"></script>

Remarque : Si vous disposez d'un accès à internet, vous pourriez utiliser les liens directs sans avoir à télécharger le dossier Bootstrap3.

Exercice 2

Dans le projet, ajouter 3 divisions.

1. Utiliser Bootstrap pour gérer des positions différentes selon la résolution de l'écran ;
2. Afficher les bordures des divisions ayant des classes colonnes ;



Exercice 3

1. Créer les mises en page présentées ci-dessous à l'aide du système de grille de Bootstrap:

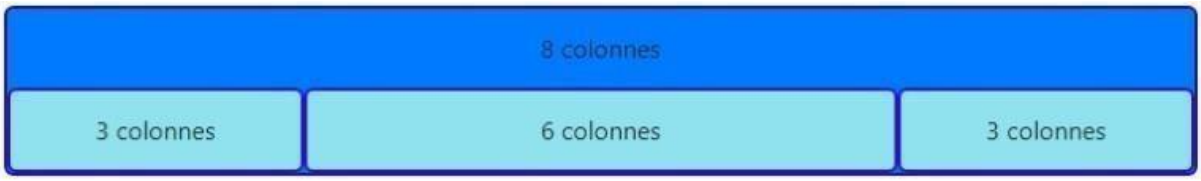


Figure 1: Grille1.

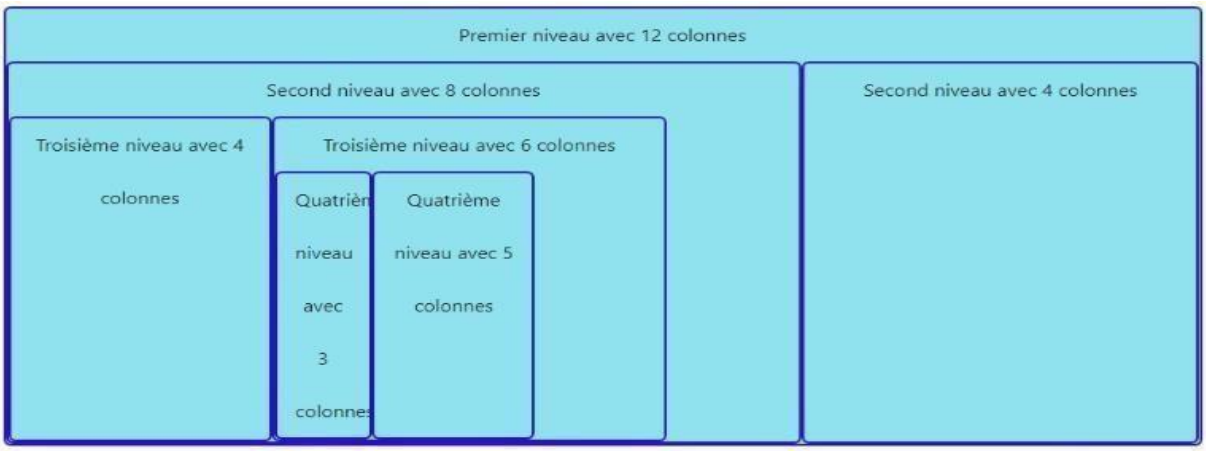


Figure 2: Grille 2.

2. Rajouter les styles suivants à votre page HTML :
   - Background-color : #90e3ee ;
   - Border : 2px solid #2b09ee ;
   - Border-radius : 6px ; *// propriété définit le rayon des coins de l'élément.*
   - Line-height : 50px ; *// propriété spécifie la hauteur d'une ligne.*
   - Text-align : center ;

Programmation Web

TP9 – BOOTSTRAP-Avancé

Flexbox *(pour flexible box)* est un mode de mise en page prévoyant une disposition des éléments d'une page de telle sorte que ces éléments possèdent un comportement prévisible lorsqu'ils doivent s'accommoder de différentes tailles d'écrans/appareils. Dans de nombreux cas, le modèle de boîte **Flexbox** offre une amélioration du modèle block dans lequel les flottements float ne sont pas utilisés, pas plus que la fusion des marges du conteneur flex avec ses éléments.

Exercice 01 : Tableaux

Dark table

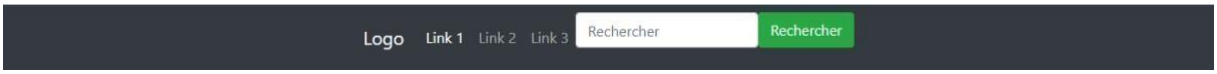
#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Light table

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

1. Réaliser le tableau donné ci-dessus.
2. Modifier les données du tableau pour qu'il devienne un tableau contenant les descriptifs d'un produit donné.
3. Rajouter les classes Bootstrap 4 applicable sur l'élément table.
4. Rajouter une légende à ce tableau.

Exercice 02 : Barre de navigation

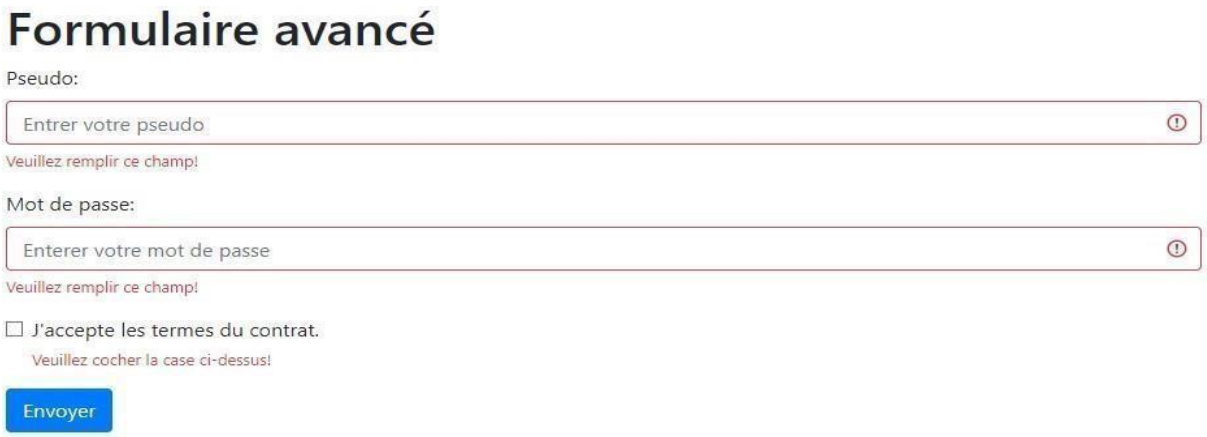


Une barre de navigation standard est créé avec la .navbar classe, suivie d'un effondrement sensible classe: .navbar-expand-xl|lg|md|sm (empile verticalement sur la barre de navigation extra large, grand, moyen ou petit écran).

1. Ajouter la classe « .justify-content-center » au centre de la barre de navigation.
2. Utiliser l'une des « .bg-colorclasses » pour changer la couleur de fond de la barre de navigation(.bg-primary, .bg-success, .bg-info, .bg-warning, .bg-danger, .bgsecondary, .bgdarket .bg-light)
3. Ajouter à votre barre de navigation La classe « .navbar-brand » pour mettre en évidence le logo ;
4. Ajouter un logo sous forme d'image ;
5. Ajouter une barre de recherche à votre barre de navigation à l'aide de :

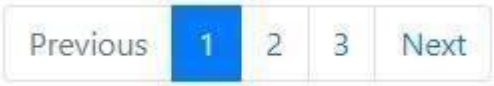
   ```
   <form class="form-inline" action="#">
   <input class="form-control" type="text" placeholder="Rechercher">
   <button class="btnbtn-success" type="submit">Rechercher</button>
   </form>
   ```
6. Fixer la barre de navigation en haut de la page à l'aide de « .fixed-top ».

Exercice 03 : Formulaire avancé



1. Reproduire le formulaire donné dans la figure ci-dessus.
2. Indiquer ce qui manque avant de soumettre le formulaire.

Exercice 04 : Pagination



1. Reproduire le contenu donné dans la figure ci-dessus ;
2. Utiliser les classes : page-item et page-link pour les éléments de la liste ainsi que les liens
3. ;
4. Activer la page numéro 1 tel qu'il est montré à la figure ;
5. Il faut que lorsque l'utilisateur navigue sur la page numéro 1 ne puisse retourner à la page précédente. Le même principe pour la page numéro 3 et next.

Programmation Web

TP10 – BOOTSTRAP-Avancé

Exercice 1 : Card

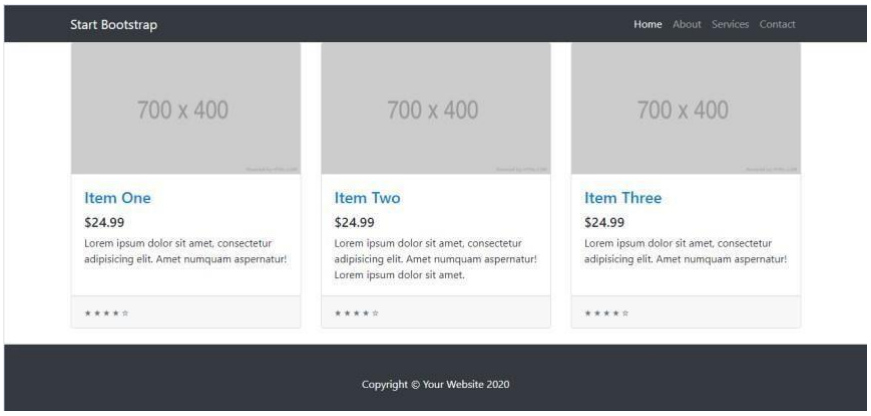


Figure 1: Exemple de l'élément card.

Une carte est un conteneur flexible et extensible. Il comprend des options pour les en-têtes et pieds de page, une grande variété de contenus, des couleurs d'arrière-plan contextuelles et de puissantes options d'affichage.

Sachant que les titres de carte sont utilisés en ajoutant **.card-title** à une balise **<h *>**. De la même manière, les liens sont ajoutés et placés les uns à côté des autres en ajoutant .**card-link** à une balise **<a>**.

Les sous-titres sont utilisés en ajoutant un sous-titre **.card** à une balise **<h *>**. Si les éléments **.card-title** et **.card-subtitle** sont placés dans un élément **.card-body**, le titre et le sous-titre de la carte sont bien alignés.

- Reproduire l'image ci-dessus qui représente un ensemble de cartes bootstrap4 ;
- Recopier la barre de navigation vue dans le TP10 ;
- Regrouper les cartes en ajoutant un ***.card-group*** ;
- Ajouter un footer tel qu'il est montré à la figure à l'aide de la classe .***card-footer.***

Exercice 2 : Thumbnail

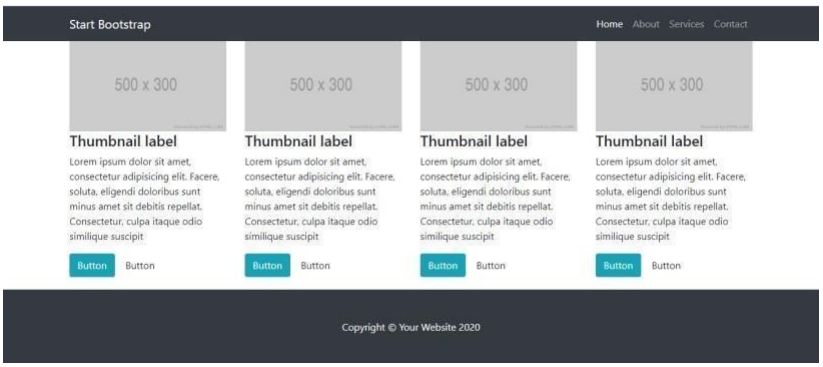


Figure 2: Exemple de l'élément Thumbnail.

- Réaliser l'image ci-dessus avec la classe ***.thumbnail*** vue au cours.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Oran's Higher Teachers College
AMMOUR Ahmed
المدرسة العليا للأساتذة بوهران
عمور احمد

Examen Type -1-

Questions de cours (04 points)

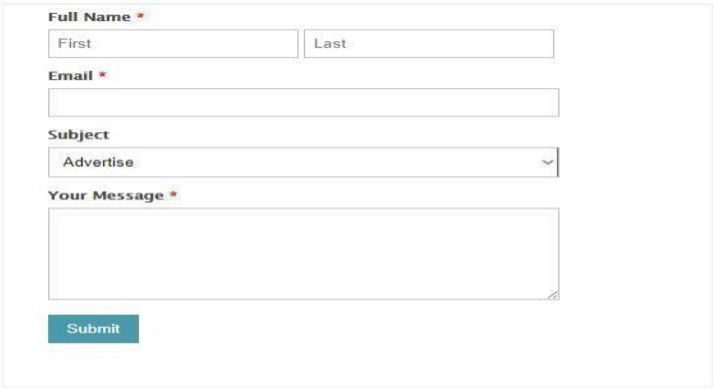
(1 point) HTML et CSS ont deux rôles différents. Lesquels ?

(1 point) Dans une page HTML, l'élément html contient généralement trois éléments fils. Quelssontils, et à quoi servent-ils ?

(1 point) Écrivez un élément td ayant comme attribut colspan la valeur 2, et comme contenu le texte « Ma colonne ».

(1 points) Écrivez du code CSS qui permet de faire en sorte tous les titres de second niveau soient écrits en vert et en gras.

Exercice 01(10 points) Soit le formulaire ci-dessous :



1. A quoi correspond ce formulaire, où le trouve-t-on ? quelle est son utilité ?
2. Que signifie l'étoile (*) mise après les champs Full Name, Email, et Your Message ?
3. Donner le code HTML/ CSS de ce formulaire ;
4. Ajouter les informations et les éléments HTML/CSS correspondant afin d'inscrire un nouvel étudiant à l'ENS d'Oran ;
5. Quelles sont les contraintes indispensables que vous devriez ajouter à ce formulaire d'inscription ?

Exercice 02(06 points)

Trouver les erreurs dans le code donné ci-dessous :

1. lien
2. bla <img src="t.png" alt="t"> bla
3. bla<h3>mon titre</h3> bla
4. bla bla
5. bla bli
6. ligne1
ligne2

Bon courage!

Examen Type -2-

Questions

1) **Lequel des styles de bouton Bootstrap suivants indiquent que cette action doit être prudente ?**
   a) .btn-warning
   b) .btn-danger
   c) .btn-link

2) **Quelle est la différence entre .container et .container-fluid ?**
   a) .container-fluid conserve les marges à gauche et à droite du contenu, tandis que .container remplit toute la largeur de la fenêtre.
   b) .container-fluid est sensible à la taille de la fenêtre, tandis que container crée une largeur fixe qui ne change jamais, quelle que soit la taille de la fenêtre.
   c) .container doit être utilisé dans les en-têtes, tandis que .container-fluid est utilisé pour les balises p.

3) **Vrai ou faux: si vous utilisez le framework Bootstrap pour votre site, vous ne pouvez pas ajouter vos propres règles CSS.**
   a) Vrai
   b) Faux

4) **Lequel des éléments suivants crée un jumbotron réactif (responsive) sur votre site Web?**
   a) <jumbotron>
      <h1>This text will be in a jumbotron</h1>
      </jumbotron>
   b)
      <h1>This text will be in a jumbotron</h1>

   c)
      <jumbotron>
      <h1>This text will be in a jumbotron</h>
      </jumbotron>

   d)

      <h1>This text will be in a jumbotron</h>

5) Combien de colonnes tiennent dans une seule ligne dans le système de grille Bootstrap?

a) 8
b) 12
c) Cela dépend de la taille de l'écran.

**6) **
** 1 **
** 2 **

7) , Quel est le plus petit appareil qui affichera ces colonnes côte à côte ?

a) Ordinateur de bureau
b) Tablette
c) Téléphone intelligent (smart phone)

8) Lequel des énoncés suivants est correct à propos des wells Bootstrap ?

a) Un well est un conteneur dans qui fait apparaître le contenu enfoncé ou un effet d'insertion sur la page.
b) Pour créer un well, enveloppez simplement le contenu que vous souhaitez voir apparaître dans le well avec un contenant la classe de .well.
c) Les deux ci-dessus.
d) Aucune de ces réponses.

9) Dans le cas d'envoi d'informations plus ou moins sensibles par formulaire, quelle méthode utilisera-t-on de préférence ?

a) get
b) mailto
c) post

Bon courage !

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Oran's Higher Teachers College
AMMOUR Ahmed
المدرسة العليا للأساتذة بوهران
عمور احمد

Examen Type -3-

Questions de cours (08 points)

a. Que signifie la balise < ! DOCTYPE HTML> dans un document WEB ?
b. Quelle est la différence entre Web et Internet ?
c. Quelle est la différence entre http et www dans l'URL http://enso.dz ?
d. Qu'est-ce qu'une adresse IP ? Pourquoi est-ce que l'on n'utilise pas uniquement des adresses MAC pour identifier les machines sur Internet ?
e. Quelle est la spécificité des noms de fichier index.html ou index.php par rapport à d'autres noms de fichiers HTML ou PHP ?
f. Quelle est la différence entre une requête HTTP POST et GET ? Donner un exemple où utiliser une requête POST a un avantage par rapport à une requête GET.
g. Quelle est la différence entre WEBM, MP4 et Vorbis ? Ou pouvez-vous les trouver ?

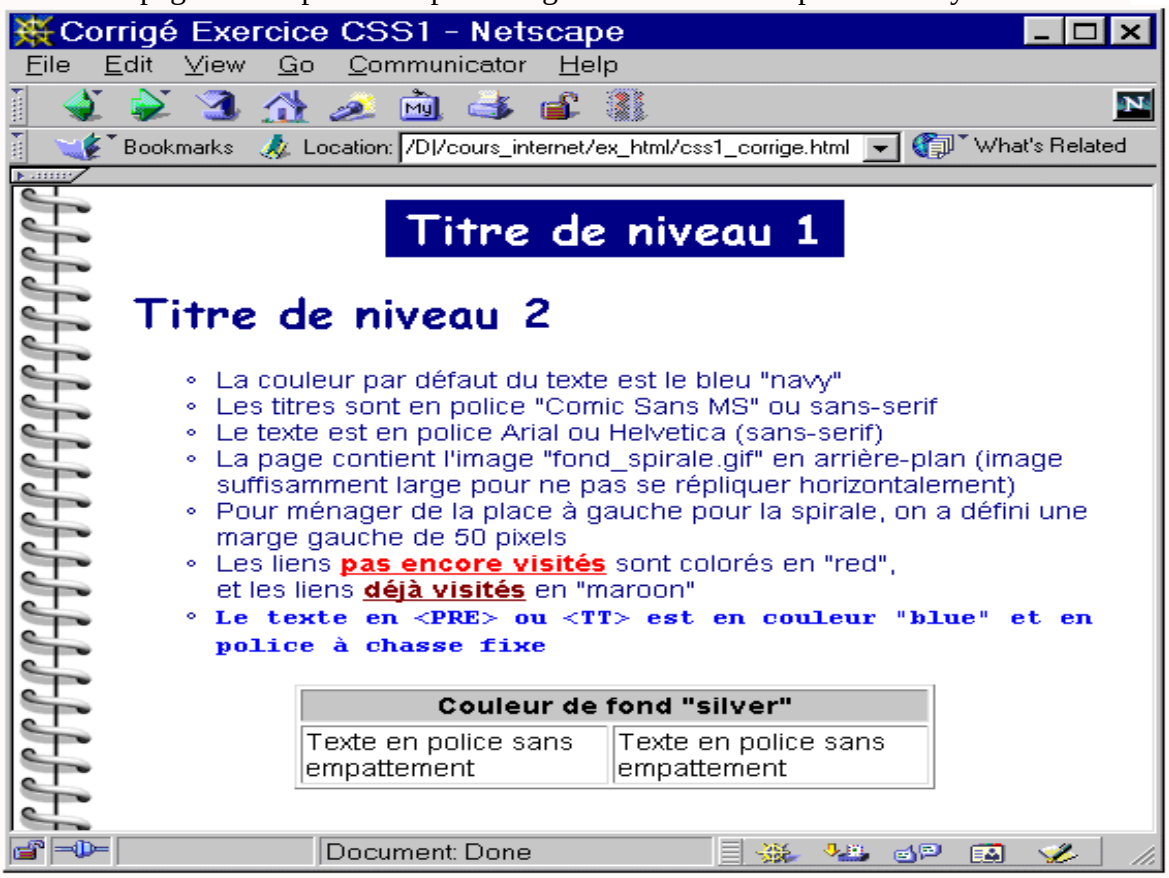
Exercice 01 (05 points)

Le document ci-dessous comporte de nombreuses erreurs qui empêchent sa validation par les validateurs du W3C (HTML et CSS). Lesquelles ? Indiquer les corrections à apporter pour qu'il le devienne (la ligne où la correction s'applique).

```
1  <!doctype html>
2  <head>
3  <meta charset="utf-8"/>
4  <meta langue="français"/>
5  </head>
6  <body>
7      <header>
8          <h2>si t'es valide, c'est bien, si tu l'es pas, tant pis !</h2>
9      </header>
10
11         <h3>Définition</h3>
12         <dl>
13             <dt><strong>Valide</strong>
14                 <dd>Sain <span>et</span> vigoureux</dd></dt>
15             </dl>
16
17
18             <table>
19                 <tr><th>Coquillages</th> <th>Crustacés</td></tr>
20                 <tr> <td>Epitonium turtoni</td> <td>Stenopus spinosus</td> </tr>
21                 <tr> <td>Asphorais pespelecani </td> <td>Palinurus elphas</td></tr>
22                 <caption>Sur la page abandonnée</caption>
23             </table>
24
25         <footer>
26             <h7>Conclusion</h7>
27             <p>Alors, alors ?<p>
28         </footer>
29     </body>
30 </html>
```

Exercice 02 (05 points)

Réaliser la page Web représentée par l'image ci-dessous en respectant le style demandé :



Exercice 03 (02 points)

Étant donné le html suivant, donnez le résultat attendu :

```
<p class="cadre">un premier paragraphe</p>
<div class="cadre" id="a">
  <p>un texte un texte un texte un texte....</p>
  <div id="b" class="cadre">
    <p> bla bla bla bla</p>
    <div class="message_cadre">note: 8</div>
  </div>
</div>
```

et les CSS :

```
.cadre {border: 1px solid black;}
#a {position: relative; left: 10ex;}
.message {position: absolute; right: 10ex; top: 0ex;}
```

Bon courage !

Examen Type -4-

Questions : Les questions numérotées de 1 à 8 comptent pour le test

1) **Quelle organisation définit les standards Web ?**
   a) IBM Corporation
   b) World Wide Web Consortium
   c) Microsoft Corporation
2) **Si nous souhaitons définir le style d'un seul élément, quel sélecteur CSS utiliserons-nous ?**
   a) id
   b) class
   c) name
3) **Supposons que nous souhaitons organiser trois DIV de sorte que DIV 3 soit placé au-dessus de DIV1. Maintenant, quelle propriété CSS nous allons utiliser pour contrôler l'ordre de pile?**
   a) d-index
   b) s-index
   c) z-index
4) **Le système de grille Bootstrap fonctionne sur plusieurs appareils.**
   a) vrai
   b) faux
5) **En bootstrap, quelle classe ajoute des zébrures à une table ?**
   a) .table-zebra
   b) .table-striped
6) **Quelle classe est utilisée pour créer une grande boîte pour attirer d'avantage l'attention de l'utilisateur ?**
   a) .bigbox
   b) .jumbotron
   c) .container
7) **Un onglet de navigation standard est créé avec:**
   a) <ul class="navigation-tabs">
   b) <ul class="nav nav-navbar">
   c) <ul class="nav nav-tabs">
8) **Quelle classe indique le texte en majuscule?**
   a) .text-uppercase
   b) .uppercase
   c) .text-capitalize
9) **Quelle classe de bouton indique un danger?**
   a) .btn-danger-button
   b) .btn-danger
   c) .btn-warning
10) **Par défaut, un formulaire prend la forme :**
   a) horizontale
   b) verticale
   c) Formulaire dans une ligne (inline)
   d) Aucun d'eux

11) Laquelle des classes suivantes crée une image arrondie ?

a) .img-cercle-coin
b) .img-crl
c) .img-cercle

12) Afin de fournir des informations sur la validité des champs avant qu'un utilisateur ait envoyé son formulaire, on ajoutera la classe :

a) .was-validated
b) .needs-validation

13) Que signifie PHP?

a) Personal Home Page
b) Hypertext Preprocessor
c) Pretext Hypertext Processor
d) Preprocessor Home Page

14) La classe lg de bootstrap indique :

a) phones
b) tablets
c) desktop
d) larger desktops

15) Quelle classe doit-on utiliser pour créer une navigation simple entre les pages web avec Précèdent et Suivant ?

a) pagination
b) pager
c) nav
d) carousel

16) La classe .btn-group organise les boutons de manière :

a) Verticale
b) Horizontale
c) Menu
d) Dropdown

Bon courage !

Références

Balises de formatage du texte - Partie 1. (s. d.). Consulté sur https://www.chiny.me/balises-de-formatage-du-texte-partie-1-3-5.php

Base CSS - Exercices CSS. (s. d.). Consulté sur https://www.codingame.com/playgrounds/36092/exercices-css/base-css (Visité le : 01/03/2023)

Benkaddour, F. Z. (2017). *Intégration d'une ontologie et des technologies Web 2.0 dans la conception d'un Système d'Aide à la Décision Collaborative (SADC) : Cas de l'industrie du non-tissé.* université d'Oran1 Ahmed Ben Bella.

CSSdébutant.com : découvrir le CSS ! (s. d.). Consulté sur https://www.cssdebutant.com/debuter-en-css-integrer-du-css-page-HTML.html (Visité le : 16/01/2023)

Exercice HTML : Page Formulaire - Exercices HTML. (s. d.). Consulté sur https://www.cours-gratuit.com/exercices-html/exercice-html-page-formulaire (Visité le : 28/02/2023)

Formation, W. (2019, 4). *Listes en HTML.* Consulté sur https://juliencrego.com/cours/listes-en-html/ (Visité le : 18/12/2022)

Formation, W. (2022, 7). *Le modèle de boîte en CSS.* Consulté sur https://juliencrego.com/cours/le-modele-de-boite-en-css/ (Visité le : 05/02/2023)

Giraud, P. (2019, 12). *Créer des liens en HTML.* Consulté sur https://www.pierre-giraud.com/html-css-apprendre-coder-cours/lien-interne-externe-ancre/ (Visité le : 18/12/2022)

Hajji, R. (2021, 11). *Exercices les formulaires en HTML série 01.* Consulté sur https://apcpedagogie.com/exercices-les-formulaires-en-html-serie-01/ (Visité le : 01/03/2023)

Hajji, R. (2022, 12). *Exercices en CSS le positionnement Série 01.* Consulté sur https://apcpedagogie.com/tp-03-css/ (Visité le : 01/03/2023)

- *HTML (HyperText Markup Language) | MDN.* (2022, 11). Consulté sur https://developer.mozilla.org/fr/docs/Web/HTML/Element/span (Visité le : 04/02/2023)

Les caractères spéciaux et entités HTML. (s. d.). Consulté sur http://zone47.com/xhtml/caracteres-entites.php (Visité le : 20/11/2022)

Les réseaux Internet en France. (2022, 10). Consulté sur https://selectra.info/telecom/guides/technologies/reseaux-internet

Mercier, C., Bird, A., & Swick, R. (s. d.). *World Wide Web Consortium (W3C).* Consulté sur https://www.w3.org

Otto, M. J. T. (s. d.). *Bootstrap.* Consulté sur https://getbootstrap.com

PUShAUNE. (s. d.). *Apprendre HTML et CSS // Autres propriétés.* Consulté sur https://www.apprendre-html-et-css.com/la_propriete_text.html (Visité le : 24/02/2023)

Qu'est-ce qu'une mise en réseau informatique ? – La mise en réseau informatique expliquée – AWS. (s. d.). Consulté sur https://aws.amazon.com/fr/what-is/computer-networking/ (Visité le : 19/10/2022)

Qu'est-ce qu'un réseau informatique ? définition et exemples. (s. d.). https://www.ionos.fr/digitalguide/serveur/know-how/reseau-informatique-definition. (Visité le : 19/10/2022)

Structure de Site Web et de document | MDN. (2022, 11). Consulté sur https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure

Universalis, E. (s. d.). *RÉSEAUX INFORMATIQUES, Grandes applications des réseaux informatiques - Encyclopædia Universalis.* Consulté sur https://www.universalis.fr/encyclopedie/reseaux-informatiques/7-grandes-applications-des-reseaux-informatiques/ (Visité le : 21/10/2022)