Avant-propos

Ce manuel scolaire, que nous souhaitons compréhensible, pratique et claire pour les élèves, est aussi un manuel de recueil pour les enseignants. Ce manuel est conforme au programme d'enseignement secondaire. Il prend en compte la grande évolutivité de l'informatique en termes de matériel et de logiciel. C'est pourquoi ce manuel met plus l'accent sur les invariants matériels et logiciels.

Ce manuel concerne les classes de **Premières ACD** de l'enseignement secondaire. Pour mieux utiliser ce support, l'élève devra d'abord lire attentivement les différentes leçons avant de traiter les exercices d'applications associés. A la fin de ce manuel, il est proposé quelques anciens sujets d'examens et quelques sujets type examen avec leurs corrigés. On retrouve également à la fin de chaque chapitre des exercices d'approfondissement.

Nous sommes convaincus qu'en utilisant correctement ce document les élèves n'auront pas de soucis à affronter l'épreuve d'informatique au Probatoire. Nous vous souhaitons beaucoup de courage.

Dans l'attente des observations, des critiques et d'éventuelles propositions visant à améliorer ce manuel «**LE LEADER en informatique** », nous encourageons chaque élève à l'acquisition d'un exemplaire. Par cette acquisition, vous nous encouragez à la recherche scientifique.

Toute la reconnaissance revient au seigneur **Jésus-Christ** qui nous a donné le succès sur le péché.

Copyright Septembre 2017

AUTEUR PRINCIPAL : GEORGES CYRILLE TCHUENTE NJIMFO

PLET et Ingénieur en informatique

Tel : +237 679637622 / 697870683 **E-MAIL** : georgtchuente@yahoo.fr

version complète = 1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

SOMMAIRE

CHAPITRE I = ALGORITHME 4

Leçon 1 : Définitions des concepts et les notions d'algorithme 4

Leçon 2 : Structure générale d'un algorithme 8

Leçon 3 : Les instructions simples et les tableaux 11

Leçon 4 : Les instructions alternatives 14

Leçon 5 : La notion de fonction 21

Leçon 6 : La notion de procédure 24

10 exercices d'algorithme types examens + corrigés 27

CHAPITRE 2 = LANGAGE HTML 32

Leçon 7 : Les pages Web 32

Leçon 8 : Structure d'un document HTML 34

Leçon 9 : Les notions de balises HTML 36

Leçon 10 : les formulaires 40

Leçon 11 : les tableaux 45

Leçon 12 : les liens hypertextes, les images et les sons 49

Leçon 13 : Assemblage des dossiers du site 53

QUELQUES EXERCICES de HTML TYPES EXAMEN 57

CHAPITRE 3 : PRODUCTION DES FEUILLES DE STYLE 60

Leçon 14 : Généralités sur les feuilles de style 60

Leçon 15 : Enrichir son document 63

Leçon 16 : Création d'un fichier de styles et Utilisation de l'attribut style 65

CHAPITRE 4 = INFOGRAPHIE 69

Leçon 17 : Généralités sur l'infographie et le traitement d'image 69

Leçon 18 : Acquisition et numérisations d'images 73

Leçon 19 : présentation et prise en main de Photoshop 75

Leçon 20 : Transformation de textes 80

Leçon 21 : Publication assistée par Ordinateur (PAO) 82

Leçon 23 : Quelques opérations de retouche de photos 86

version complète =1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

Leçon 24 : le Biseautage d'une photo 88
Leçon 25 : Généralités sur la notion de multimédia 89
Quelques exercices de multimédia+ Infographie types examen + Correction 93
Leçon 26 : Maintenance préventive et optimisation d'un ordinateur 97
Quelques exercices de maintenance types examen + Correction 100
anciens sujets d'examen + corrections ... 109
PROBATOIRE 2014 109
CORRECTION PROBATOIRE 2014 109
PROBATOIRE SESSION 2015 110
CORRECTION PRABATOIRE 2015 111
PROBATOIRE SESSION 2016 113
CORRECTION PRABATOIRE 2016 114
PRABATOIRE 2017 115
CORRECTION PRABATOIRE 2017 117
PR0BATOIRE 2018 119
CORRECTION DU PRABATOIRE 2018 121
Sujets types d'examen + corrections ... 122
PROBATOIRE BLANC 2018 COLLEGE JEAN TABI 122
CORRECTION PROBATOIRE BLANC 2018 COLLEGE JEAN TABI .. 124
PROBATOIRE BLANC VOGT 2018 126
CORRECTION PROBATOIRE BLANC VOGT 2018 127
Probatoire blanc collège saint joseph de Bandjoun 128
CORRECTION PROBATOIRE C.ST.BANDJOUN 130
Probatoire blanc Régional session 2015 131
CORRECTION PROBATOIRE BLANC 2015 132
PROBATOIRE BLANC : COLLEGE SAINT THOMAS D'ACQUIN 133
CORRECTION PROBATOIRE BLANC ST JOSEPH 134
PROBATOIRE BLANC CONODIC 135
CORRECTION PROBATOIRE BLANC CONODIC 136
PROBATOIRE BLANC REGIONAL SESSION 2014 138
CORRECTION PROBATOIRE BLANC REGIONAL SESSION 2014 139

Version complète =1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

CHAPITRE I : ALGORITHME

Leçon 1 : Définitions des concepts et les notions d'algorithme

Objectifs :

L'élève doit être capable de :

- Définir les différents concepts de l'algorithme (algorithme, mot clé, programme, …)
- Enoncer les étapes de résolution d'un problème algorithmique,
- Enoncer les caractéristiques ou les qualités d'un algorithme
- Enoncer les étapes de la mise en place d'un programme informatique

INTRODUCTION

L'algorithme est un moyen pour le programmeur de présenter son approche du problème à d'autres personnes.

I. LES ETAPES DE RESOLUTION D'UN PROBLEME

Face à un problème quelconque, nous devons nous poser au préalable un certain nombre de questions. La réponse à ces questions facilitera la résolution du problème c'est-à-dire aboutir à un résultat.

1. Les données d'un problème

Elles peuvent être définies comme un ensemble d'éléments nécessaires à résolution d'un problème. C'est ainsi qu'un maçon pour construire une maison doit savoir quels éléments lui faut-il pour résoudre son problème. Pour réaliser son projet il lui faut du sable béton ; des taules… Ces éléments sont les données d'entrée de son problème (construire une maison).

2. Les résultats d'un problème

Ils peuvent être définis comme les attentes ou objectifs visés dans la résolution du problème. Déjà les éléments d'entrée nous devons nous fixer les objectifs à atteindre car le but ici est de résoudre un problème donné.

3. La procédure de résolution du problème

Ce sont les étapes permettant d'aboutir à un résultat donné.et le résultat qu'on souhaite obtenir il est question ici d'énoncer les étapes de résolution du problème c'est-à-dire comment partir des éléments dont on dispose pour obtenir le résultat souhaité.

II. LES ETAPES DE RESOLUTION D'UN PROBLEME EN LANGAGE ALGORITHMIQUE

version complète=1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

Certains de nos problèmes demandent des énormes efforts aussi bien cognitifs que psychomoteurs. Alors, pour alléger ce problème l'Homme a souvent recourt à la machine afin de lui faciliter certaines tâches. Mais pour y arriver nous devons nous poser un certains nombres de question d'où la notion d'**algorithme.**

1. Définition d'un algorithme

C'est une séquence d'instructions qui décrit comment résoudre un problème particulier **ou** encore une suite d'instruction ordonnées permettant de résoudre un problème donné**.** Il doit respecter les règles suivantes **:**

- il est défini sans ambigüité.
- Il se termine après un nombre fini d'opérations.
- Il manipule les objets qui doivent être définis de façon très précis.

2. Etapes de résolutions d'un problème en langage algorithmique ou démarche algorithmique

L'algorithmique est une science qui recherche les procédés d'automatisation de certaines taches afin de simplifier le travail humain. La démarche algorithmique basique est la suivante :

- Enoncer clairement les résultats que l'on veut obtenir
- Remonter des résultats aux données
- Décomposer le traitement dans une démarche naturelle
- Réaliser l'algorithme en respectant la syntaxe
- Construire un jeu d'essai pour vérifier si l'algorithme fonctionne

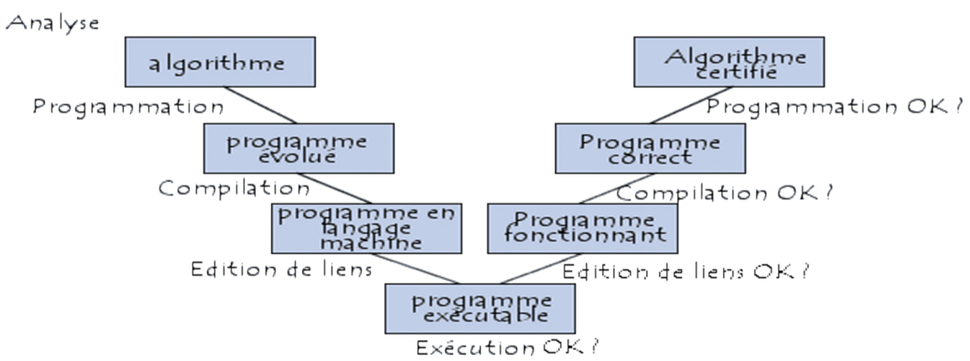
3. Langage de programmation

Un langage de programmation est un moyen d'expression d'instructions à une machine. On distingue ainsi des langages évolués tel Java, C++,… qui sont construit autour d'un vocabulaire et d'une syntaxe stricte.

Le vocabulaire est constitué des mots bien défini ayant un sens particulier dans le langage. Ces mots sont appelés des **mots réservés** ou **mots clés** (ils doivent toujours être soulignés).

4. Etapes de mise au point d'un programme informatique

La mise au point d'un programme informatique se fait en plusieurs étapes.



Version complète = 1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

Il s'agit de fournir la solution à un problème, la première étape consiste donc à analyser le problème, c'est-à-dire en cerner les limites et le mettre en forme dans un langage descriptif, on parle généralement d'**analyse** pour décrire le processus par lequel le problème est formalisé. Le langage de description utilisé pour écrire le résultat de l'analyse est appelé **algorithme**. L'étape suivante consiste à traduire l'algorithme dans un **langage de programmation** spécifique, il s'agit de la phase de **programmation**. Le ***langage de programmation*** est l'intermédiaire entre l'humain et la machine, il permet d'écrire dans un langage proche de la machine mais intelligible par l'humain les opérations que l'ordinateur doit effectuer. Ainsi, étant donné que le langage de programmation est destiné à l'ordinateur, il doit donc respecter une syntaxe stricte. Un algorithme peut toutefois aboutir à plusieurs programmes.

Le programme est ensuite transformé en **langage machine** lors d'une étape appelée **compilation**. La *compilation* est une phase réalisée par l'ordinateur lui-même grâce à un autre programme appelé **compilateur**. La phase suivante s'appelle l'**édition de liens**, elle consiste à lier le programme avec tous les éléments externes (généralement des librairies auxquelles il fait référence).

III. CARACTERISTIQUES D'UN ALGORITHME

Un algorithme est l'énoncé dans un langage bien défini d'une suite d'opérations permettant de répondre au problème. Un algorithme doit donc être :

- **lisible**: l'algorithme doit être compréhensible même par un non-informaticien
- **de haut niveau**: l'algorithme doit pouvoir être traduit en n'importe quel langage de programmation, il ne doit donc pas faire appel à des notions techniques relatives à un programme particulier ou bien à un système d'exploitation donné
- **précis**: chaque élément de l'algorithme ne doit pas porter à confusion, il est donc important de lever toute ambiguïté
- **concis:** un algorithme ne doit pas dépasser une page. Si c'est le cas, il faut décomposer le problème en plusieurs sous-problèmes
- **structuré:** un algorithme doit être composé de différentes parties facilement identifiables.
- **Modularité :** Segmentation en petits problèmes
- **Robustesse :** Adaptation aux situations diverses et voire imprévues

CONCLUSION

L'algorithme est un langage formel et structurer permettant de résoudre de manière optimale un problème donné.

BILINGUAL GAME

FRENCH	ENGLISH

version complète -1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

Algorithme	Algorithm
Mots clés	Keys words
Programme	Program
Langage de programmation	Programming language
Etapes de résolution des problèmes	Steps of problems solving

EXERCICES D'APPLICATION THEORIQUE

1. Donner la définition des mots suivants : **Algorithme, mot clé, programme**
2. Quels encore les caractéristiques d'un algorithme?
3. Etablir la différence entre les termes « Algorithme » et « Algorithmique » ?
4. Quels sont étapes de réalisation d'un programme informatique ?
5. Qu'est-ce qu'un langage de programmation ? citer au moins deux exemples

Devoirs à faire à la maison

1. Quel est selon vous l'apport des algorithmes pour le quotidien des hommes ?
2. En quoi remarque-t-on qu'un algorithme est modulaire ?
3. Présenter les étapes nécessaires à la réalisation d'un algorithme qui permettent de calculer la surface d'un carré
4. Une recette de cuisine est un algorithme VRAI ou FAUX ? pourquoi ?
5. Citer deux avantages d'un langage structuré

Corrections

EXERCICES D'APPLICATION THEORIQUE

1. **Algorithme** : c'est une suite ordonnée d'instructions permettant de résoudre un problème donné. **Mot clé** : c'est un mot réservé d'un langage
   Programme : c'est un algorithme écrit dans un langage de programmation
2. Lisible, Concis, Précis, Haut niveau, Structuré, Modulaire, robuste
3. **Un algorithme** : c'est une suite ordonnée d'instructions permettant de résoudre un problème donné alors que l'algorithmique est la science qui étudie l'algorithme.
4. **Analyse (algorithme), Programmation (programme évolué), Compilation (programme en langage machine), Edition des liens (programme exécutable)**
5. **Un langage de programmation** est un langage permettant d'écrire un programme informatique. **Exemple** : C++, JAVA

Devoirs à faire à la maison

1. Quel est selon vous l'apport des algorithmes pour le quotidien des hommes ?
   - Souvent utilisés par les médecins pour prédire des résultats ou choisir des traitements

version complète=1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

10 exercices d'algorithme types examens + corrigés

Exercice 1 : Algorithme

```
Algorithme Traitement
    Var  a, b, somme, diff : Réel ;
    Const  p = 2 ;
  Fonction addition (nb1 : Réel, nb2 : Réel) : Réel
    Début
      Retourner (nb1 + nb2) ;
    Fin
  Fonction soustraction (nb1 : Réel, nb2 : Réel) : Réel
    Début
      Retourner (nb1 - nb2) ;
    Fin
  Début
  ecrire (" Veuillez saisir deux nombres ") ;
  Lire(a) ;   lire(b) ;
  Somme ← addition (a, b) ;
  Diff  ← soustraction (a, b) ;
  Si (a > b) alors
    ecrire ("le résultat est : ", somme * p);
  Sinon
    ecrire ("le résultat est : ", diff  *  p);
  finsi
FinAlgo
```

1- Quel est l'entête de cet algorithme ?
2-Combien de sous programmes possède cet algorithme ?
3- citez les paramètres de la fonction « addition »
4- donner la syntaxe de déclaration d'une procédure
5- Quelle différence faites-vous entre une procédure et une fonction ?
6- Citez les différentes instructions de base utilisées dans l'algorithme ci-dessus
7- Faire un tableau récapitulatif des objets manipulés par l'algorithme ci-dessus. Ce tableau devra préciser le nom de l'objet, sa nature et son sens (objet d'entrée ou de sortie).
8- Exécutez l'algorithme ci-dessus et donnez le message qui sera affiché à l'écran dans les cas suivants :

a- a = 10 et b = 9 ;
b- a = 9 et b = 10 ;
c- a = 10 et b = 10

EXERCICE 2

```
Algorithme :.. Inconnu2 ...
var a, b, c : entier
Début
a=3; b=2 ;
tantque(a>0) faire
c=a-2*b;
a=a-1;
fintanque
b=a+c ;
fin
```

a) Identifiez les structures algorithmiques utilisées dans cet algorithme.
b) quelle est la valeur respective de a, b et c à la fin de l'exécution de cet algorithme?

EXERCICE 3

On veut concevoir un algorithme qui calcule le périmètre d'un rectangle.
Rappel : Périmètre = 2 × (longueur + largeur)

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

a) Quelles sont les données en entrée ?
b) Quel est le traitement à effectuer sur ces données ?
c) Ecrire un algorithme qui prend en entrée les dimensions du rectangle, calcule et affiche à l'écran, le périmètre du rectangle
d) Donnez la valeur de la variable ***result*** *à la première exécution de la boucle si on entre 5*

EXERCICE 4

```
1   Algorithme quelconque ;
2   var nombre, result: entier ; var reponse : chaine de caractère ;
3   Début
4   reponse ←"OUI" ;
5   TANTQUE (reponse = "OUI") FAIRE
6   Ecrire ("Entrer un nombre") ;
7   lire (nombre) ;
8   nombre ← nombre + 1 ;
9   result ← 4 * nombre ;
10  Ecrire (result) ;
11  lire (reponse) ; }
12  FinTANTQUE
```

Exercice 5

Soit l'algorithme ci-contre. Identifier :
1.1 Le type de données des variables a et n.
1.2 La structure de contrôle
1.3 Une instruction de lecture
1.4 Une instruction d'initialisation

```
Algorithme sage
Var a, n : ... ;
Début
a ← 0 ;
Répéter
Ecrire("Entrer un nombre") ;
Lire(n) ;
a ← a+n ;
jusqu'à (a>10)
fin
```

EXERCICE 6

Soit l'algorithme suivant :

```
Algorithme Exercice1
Var i : Entier ;
Début
Pour i allant de 1 à 5 Faire
 Ecrire ("Entrer un nombre") ;
 Lire (t[i]) ;
FinPour
FinAlgo
```


Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

1.1 Quel est le type de données composé utilisé dans cet algorithme ?
1.2 Ce type a été omis d'être déclaré, donner la syntaxe de sa déclaration.
1.3 Que faire cet algorithme ?

EXERCICE 7 : algorithme

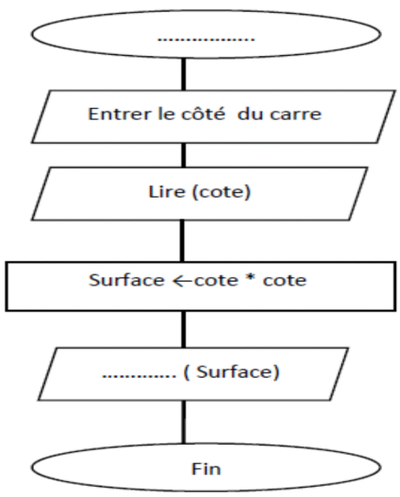
Soit l'algorithme ci-dessous. La fonction est déclarée dans la partie déclarative de l'algorithme

Algorithme Valeur_Absolue **Var** a ,b: **Entier ;** **Début** **écrire**("Entrez un entier :") ; **lire**(a) b ←abs(a); **écrire**("la valeur absolue de ",a," est ",b) ; **fin.**	**Var** a ,b: **Entier ;** **fonction** abs **(**unEntier : **Entier) : Entier ;** **var** valeurAbsolue : **Entier ;** **si** unEntier >= 0 **alors** valeurAbsolue ←unEntier ; **sinon** valeurAbsolue ← unEntier ; **fsi ;** **retourner** valeurAbsolue ; **finFonction ;**

1. Donnez le nom de la fonction utilisée dans cet algorithme et le type de données qu'elle doit renvoyer
2. Identifiez la structure de contrôle utilisée dans cet algorithme. Donner sa nature
3. Relevez dans cet algorithme, l'instruction permettant d'appeler la fonction et de ranger son résultat dans une variable.
4. Relever dans cet algorithme deux instructions simples

Exercice 8 : Algorithme

Donner l'algorithme qui a permis d'obtenir l'algorithgramme ci-dessous



version complète -1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

CHAPITRE 3 : PRODUCTION DES FEUILLES DE STYLE

Leçon 14 : Généralités sur les feuilles de style

Objectifs :

L'élève doit être capable de :

- Définir Feuille de style,
- Savoir quand appliquer les feuilles de style,

INTRODUCTION

Lorsque nous ouvrons la page d'accueil d'un site web, nous sommes en général émerveillés par sa mise en forme (alignements, couleurs, images, etc.). Il est par ailleurs facile de constater que les pages d'un site présentent les mises en formes presque identiques. Comment d'opère cette mise en forme dynamique ?

1. définition

Le concept de **feuilles de style** est apparu en 1996 avec la publication par le W3C d'une nouvelle recommandation intitulée « *Cascading StyleSheets* » (*feuilles de style en cascade*), notée **CSS**. C'est l'application de mises en forme à des éléments de texte. Le principe des feuilles de style consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte. Il est ainsi possible de créer un groupe de titres en police Arial, de couleur verte et en italique.

2. Intérêt des feuilles de style

Les feuilles de style ont été mises au point afin de compenser les manques du **langage HTML** en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier ! Elles sont appellées « *feuille**s** de style en **cascade*** » (en anglais « **Cascading Style Sheets** ») car il est possible d'en définir plusieurs et que les styles peuvent être hérités en cascade.

Les feuilles de style permettent notamment :

- d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ;

version complète - 1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

- de permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes ;
- une plus grande lisibilité du HTML, car les styles sont définis à part ;
- des chargements de page plus rapides, pour les mêmes raisons que précédemment ;
- un positionnement plus rigoureux des éléments.

3. **URL** *Uniform Resource Locator – Littéralement « Localisateur uniforme de ressources »*

C'est l'adresse « écrite » (suite de caractères) d'une ressource sur Internet. Par usage, on parle aussi d'adresse Web. Une ressource peut être une page Web, une image, un fichier de n'importe quel type (texte, calcul, son, vidéo, archive...). Elle est composée du protocole, le plus souvent « http », suivi par « **://** », du nom de domaine et, s'il y a lieu, du chemin vers le fichier (la ressource) ciblé. Par exemple : http://dite-ens.cm/annuaire/promo_2009

http://	est le type de service, ici le Web ;
dite-ens.cm	est le nom de domaine ;
/annuaire/promo_2009	est le chemin vers la page prom_2009

BILINGUAL GAME

FRENCH	ENGLISH
Domaine	Domain

EXERCICES D'APPLICATION THEORIQUE

1. Définir feuille de style
2. Pourquoi utilise-t-on les feuilles de style ?
3. Donner 04 avantages des feuilles de styles
4. Soit le site : **http://**www.minesup.com identifier les partis de ce site
5. Quelle est la différence entre les feuilles de style et le langage html ?

EXERCICE A FAIRE A LA MAISON

Que signifie les sigles suivants : FTP, URL, http, CSS

Corrections

EXERCICES D'APPLICATION THEORIQUE

1. Une **feuille de style** est une application qui consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments.

Version complète = 1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

2. Les feuilles de style ont été mises au point afin de compenser les manques du **langage HTML** en ce qui concerne la mise en page et la présentation.
3. Donner 04 avantages des feuilles de styles
   - d'obtenir une présentation homogène sur tout un site en faisant appel sur toutes les pages à une même définition de style ;
   - de permettre le changement de l'aspect d'un site complet entier par la seule modification de quelques lignes ;
   - une plus grande lisibilité du HTML, car les styles sont définis à part ;
   - des chargements de page plus rapides, pour les mêmes raisons que précédemment ;
   - un positionnement plus rigoureux des éléments.
4. Soit le site : **http://**www.minesup.com identifier les partis de ce site

http://	est le type de service, ici le Web ;
www.minesup.com	est le nom de domaine ;

5. le **langage HTML** offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte, toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce **aux feuilles de style**, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier !

EXERCICE A FAIRE A LA MAISON

FTP: File Transfer Protocol
URL: Uniform Resource Locator
HTTP: HyperText Tranfer Protocol
CSS: **Cascading Style Sheets**

Version complète =1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

Leçon 15 : Enrichir son document

Objectifs :

L'élève doit être capable de :

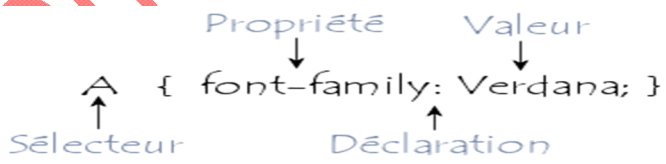
- Savoir comment appliquer les feuilles de style
- Connaitre la syntaxe générale
- Modifier des caractères (font-family, font-size, font-weight, color, …)
- Modifier le texte (text-decoration, text-align, text-transform

I. Définition d'un style

La définition d'un style se fait à l'aide de règles en texte simple permettant de décrire l'aspect des éléments de la page. Une règle CSS est caractérisée par deux principaux éléments :

- Un **sélecteur de balises**, permettant de préciser à quelles balises du document le style s'applique ;
- Une **déclaration** de style, définie entre accolades, permettant de préciser le style à appliquer aux balises sélectionnées. La déclaration est elle-même constituée :
  - d'une ou plusieurs **propriété**(s), suivie du caractère « : » (double point),
  - d'une ou plusieurs **valeur**(s) associée(s) à chaque propriété, entourée de guillemets s'il s'agit de plusieurs mots ou séparés par des virgules s'il y en a plusieurs, suivie d'un point-virgule.

La syntaxe est ainsi la suivante : **Sélecteur {propriété 1 : valeur ; …. Propriété n : valeur n}** exemple :



La syntaxe suivante définit par exemple le style à appliquer aux liens hypertextes (balise <A>), c'est-à-dire une police Verdana de taille 18 pixels, en gras et de couleur jaune :

```
A {
        font-family: Verdana;
         font-size: 18px;
         font-style: bold;
          color: yellow
                        }
```

Les sélecteurs de balise

version complète =1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

On appelle « sélecteur de balise » (ou « sélecteur d'éléments ») le ou les mots clés précédant l'accolade et servant à indiquer le ou les balises du document auxquelles le style entre accolades s'applique.

Pour définir le style d'une balise HTML spécifique, il suffit de mettre le nom de la balise (sans les caractères < et >. Par exemple :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
        <HEAD>
                <STYLE type="text/css">
                <!--  balise {propriétés}    -->
                </STYLE>
        </HEAD>
<BODY>
<balise> ... </balise>
...
</BODY>
</HTML>
```

II. COMMENT APPLIQUER UN STYLE SUR UNE BALISE

➢ Application simple

balise {propriété1 **:** valeur1**;** propriété2 **:** valeur2**;** ...}

Exemple de code CSS : **p { font-family: Arial, sans-serif; color: purple; }**

Tous les textes inclus entre les balises (paragraphes) seront de police de caractère « Arial » et de couleur violette.

➢ Application multiple

balise1, balise2 {propriété1 **:** valeur1**;** propriété2 **:** valeur2**;** ...}

Exemple de code CSS : **h1, h2, h3 {font-family: Arial, sans-serif; font-style: italic;}**

Tous les titres de niveau 1, 2 et 3 seront de police de caractères « Arial » et en italique.

BILINGUAL GAME

FRENCH	ENGLISH
Style	Style

EXERCICES D'APPLICATION THEORIQUE

1. Donner la syntaxe générale d'un style
2. Comment peut-on appliquer un style à une balise ? donner les différentes syntaxes
3. Donne la syntaxe d'un style appliqué à une balise disposant des propriétés suivantes : une police arial, un style italic et une couleur yellow. Puis applique au texte « hello world » dans une page web

version complète =1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM

CHAPITRE 4 : INFOGRAPHIE

Leçon 17 : Généralités sur l'infographie et le traitement d'image

Objectifs :

L'élève doit être capable de :

- Définir infographie, image, résolution d'une image
- Distinguer les différents logiciels de traitements d'image
- Distinguer les différents types d'images
- Distinguer les formats d'images

I. GENERALITES SUR LA NOTION D'IMAGE

1. Définition

Une **image** est un fichier contenant un dessin, une photographie, une icône. Une image est un fichier binaire.

2. Quelques formats

Un format d'image donne des informations (résolution, algorithme de compression, le codage des couleurs, des teintes et des transparences) sur l'image.

Les formats servant à stocker les images (photos, dessin, etc.) sont nombreux. Les plus courants sont : **BMP, GIF, ILBM, EPS, JPEG** (extension JPG), **PCX, TIFF, TGA**, **WMF**…

3. Les types d'images

On distingue deux types d'images : les images matricielles (ou images bitmap) et les images vectorielles

a) Image matricielle (ou image bitmap)

Une image matricielle est décomposée en éléments de base appelés points. Exemples : **BMP, GIF, ILBM, JPEG** (extension JPG), **PCX, TIFF, TGA** (formats bitmap) Les images bitmap sont divisées en plusieurs sous types.

- Les images 2D : elles sont représentées
- Les images 2D animées ou animations 2D
- Les images 3D : elles sont représentées dans un espace à trois dimensions
- Les images 3D animées ou animations 3D

Remarque : rappelons-nous qu'une image bitmap est composée d'un ensemble de points. En dimension 2 (2D) ces points sont appelés **pixels**. En dimension trois par contre (3D) ces points sont des appelés **voxels** ou **boxel.**

4. Image vectorielle

Version complète = 1500

Créé par la version démo de Universal Document Converter. La version complète n'ajoute pas ce texte.
FR.PRINT-DRIVER.COM