

Sciences Numériques et Technologie

Seconde

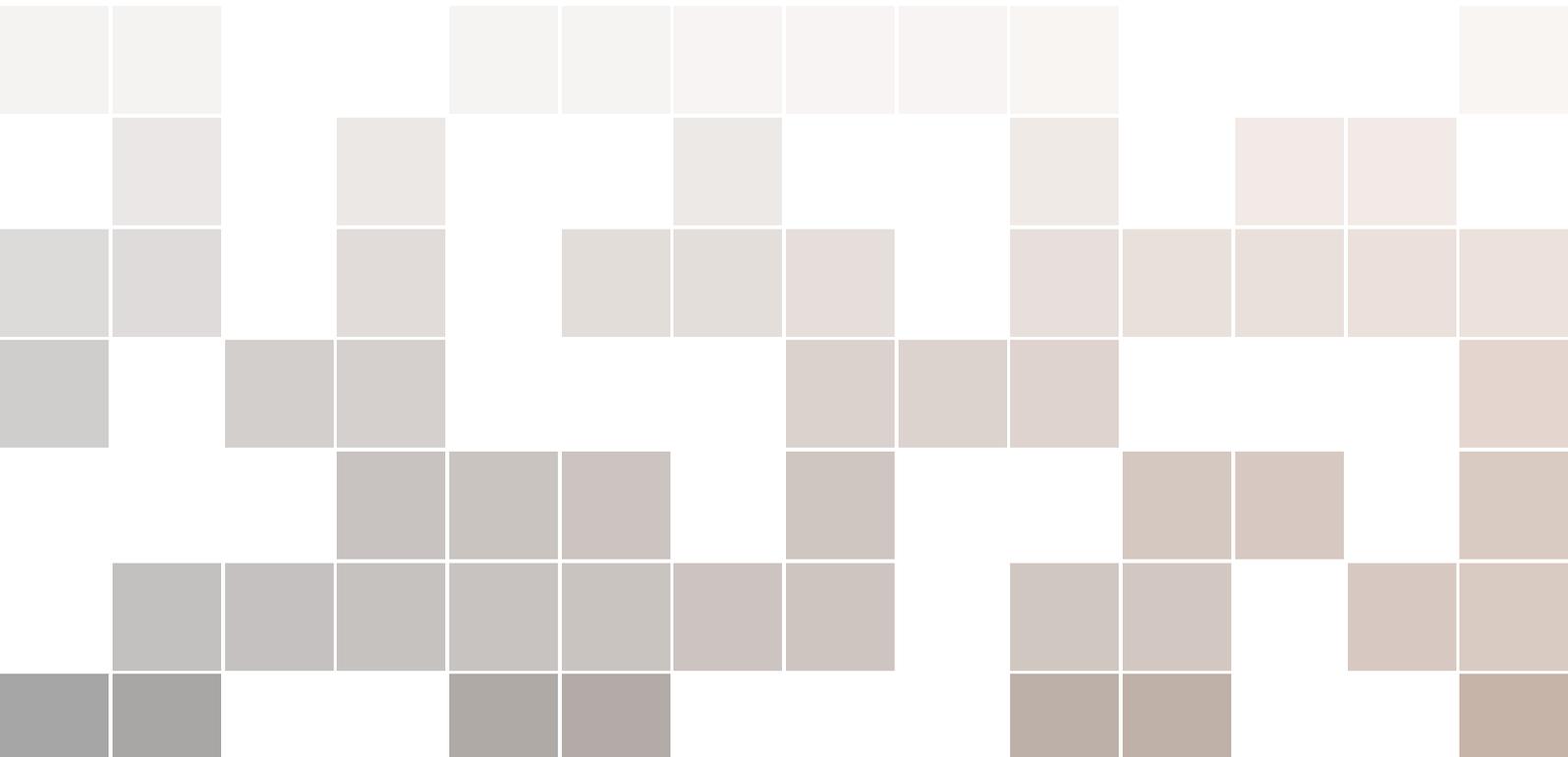


Table des matières

1	Le langage Python	5
1.1	Les variables	5
1.1.1	Définitions	5
1.1.2	Manipulation	5
1.2	Les boucles	7
1.2.1	La boucle for	7
1.2.2	La boucle while	8
1.3	Les conditions	8
1.4	Les fonctions	9
1.5	Indentation du code	10
2	Image Numérique	13
2.1	Tâche finale	13
3	Le Web	15
3.1	Le langage HTML	15
3.2	Mise en forme avancée : CSS	16
3.2.1	CSS : Style intégré à une balise HTML	17
3.3	Séparer contenu et mise en forme : la bonne solution	18
3.4	Mise en forme complexe	20

4	Internet et réseaux	21
4.1	Adresse IP	21
4.2	Les composants d'un réseau	22
4.3	Routage et DNS	25
4.4	P2P	26
5	Structure de donnée	29
5.1	Base de données	29
5.2	Manipulation d'une base de données	30
5.2.1	Importation d'une table	30
5.2.2	Sélection de données	31
5.3	La jointure	33
5.4	Regroupement et graphiques	34
5.5	Tâches finales	37
6	Projet Final	39
6.1	Principe général du projet	39
6.2	Les étapes du projet	40
6.2.1	Étape 1 : le parsing	40
6.2.2	Étape 2 : Object Oriented Programming	41
6.2.3	Étape 3 : L'algorithme de Dijkstra	41
6.3	Étape 1 : Le parsing	42

1. Le langage Python

1.1 Les variables

1.1.1 Définitions

Une variable est une information stockée par un ordinateur. Une variable possède un nom, un type et une valeur. Le nom est celui que vous avez choisi. Les types les plus communs sont les suivants :

- Entier (int)
- Décimaux (float)
- Chaîne de caractère (string)
- Tableau (array)

Python reconnaît automatiquement le type d'une valeur. Vous n'avez donc pas à déclarer (signifier) le type d'une variable.

Exemple 1.1.1 Voici quelques exemples de variables.

```
# a est un entier dont la valeur est 2
a = 2
# b est un decimal (float) de valeur 5,2
b = 5.2
# txt est une chaine de caractere
txt = "Bonjour"
# tab est un tableau contenant les nombre entiers 7-12-15
tab = [7, 12, 15]
```

Faites attention à l'usage de la virgule. Nous sommes sur des normes anglo-saxonnes. La virgule est considérée comme un séparateur entre deux nombres. Pour faire la part entre la partie entière et la partie décimale d'un nombre, on utilisera le point. Pour assigner une valeur à une variable on utilise le =.

1.1.2 Manipulation

Comme son nom l'indique, une variable peut varier. Pour ce faire on a besoin d'opérateur. Un opérateur est un symbole (ou un mot réservé) utilisé pour effectuer une opération entre des opérandes. Une opérande est une variable, un littéral ou bien une expression. Une expression est une suite valide d'opérateurs et d'opérandes.

Opérateurs mathématiques

Symboles	Types	Exemples
+	int, float, string, array	6 + 4 → 10 "a" + "b" → "ab" [1, 2, 3] + [2, 4, 6] → [1, 2, 3, 2, 4, 6]
+ =	int, float	6 + = 4 → 10
-	int, float	6 - 4 → 2
*	int, float, string	6 * 4 → 24 1.2 * 4 → 4.8 3 * "he" → "hehehe"
**	int, float	4 ** 2 → 16
/	enter, réel	11 / 4 → 2.75
//	enter, réel	10 // 4 → 2
%	enter, réel	11 % 4 → 3

Opérateurs de comparaison

Un opérateur de comparaison renvoie un booléen. Un booléen est une variable qui a soit la valeur vrai (True) ou la valeur faux (False).

Opérateur	Signification
<	Inférieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
==	Est égal à
!=	Est différent de

Exemple 1.1.2 Voici quelques exemples :

```
>>> 2 == 3
False
>>> 2 != 3
True
>>> 3 == 3
True
>>> 3 == '3'
False
>>> 2 < 3
True
>>> "a" < "aa"
True
```

Opérateurs logiques

Les opérateurs logiques ont pour fonction de combiner des tests logiques entre eux afin d'en faire des tests logiques plus complexes.

Opérateur	Signification	Fonctionnement
not	non booléen	Renvoie l'opposé de l'expression. Si a = true alors not a = false
and	et booléen	Renvoie vrai si les deux expressions sont vraies. (a<10) and (a>2) renvoie true si a est compris entre 2 et 10
or	ou booléen	Renvoie vrai si une des deux expressions est vraie. (a<10) or (a>2) renvoie true si a est supérieur à 2 ou inférieur à 10

1.2 Les boucles

Une boucle for parcourt une séquence et effectue une action pour chaque élément de cette séquence. Une séquence est une structure de donnée ordonnée de longueur finie. Un tableau ou une chaîne de caractères sont des séquences.

1.2.1 La boucle for

Exemple 1.2.1 Ce sera plus clair avec quelques exemples

```
txt = "Bonjour"
compteur = 0
for i in txt:
    print i, "C'est la lettre", compteur
    compteur += 1
```

```
B C'est la lettre 0
o C'est la lettre 1
n C'est la lettre 2
j C'est la lettre 3
o C'est la lettre 4
u C'est la lettre 5
r C'est la lettre 6
```

```
tab = range(10)
print tab
for i in tab:
    print i*8
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
0
8
16
24
32
40
48
```

56
64
72

1.2.2 La boucle while

Une boucle while tourne tant que la condition n'est pas vérifiée. Cette boucle est à manier avec précaution. En effet si la condition ne se vérifie jamais, le programme tourne indéfiniment.

Exemple 1.2.2 Voici quelques exemples :

```
i = 0
while i < 10 :
    print i
    i+=1
```

0
1
2
3
4
5
6
7
8
9

1.3 Les conditions

Les conditions permettent d'exécuter un morceau de code si une condition est vérifiée. Le mot clé if permet de mettre en place une condition.

Exemple 1.3.1 Dans cet exemple, nous allons compter le nombre de lettres e dans la phrase.

```
txt = "Le mal vient de ce que l'homme se trompe au sujet du bien."

#Nombre de lettre e
nbe = 0

for i in txt:
    if i == "e" :
        nbe += 1

print "Le nombre de e est",nbe
```

Le nombre de e est 10

Le mot clé else permet d'exécuter un morceau de code dans le cas où la condition n'est pas vérifiée.

Exemple 1.3.2 En reprenant l'exemple précédent

```
txt = "Quand tu prends confiance en la confiance tu deviens confiant."

#Nombre de lettre e
nbe = 0
#Compte le nombre de lettres autre que e
nbautres = 0
for i in txt:
    if i == "e" :
        nbe += 1
    else:
        nbautres += 1

print "Le nombre de e est",nbe
print "Le nombre des autres lettres est",nbautres
```

Le nombre de e est 6

Le nombre des autres lettres est 56

1.4 Les fonctions

Les fonctions permettent d'encapsuler un morceau de code afin de le réutiliser plus tard. Les fonctions favorise aussi la lisibilité de votre code. Le mot clef `def` permet de définir une fonction. Une fonction possède :

- Un nom (sans espace et sans accent)
- Des arguments (valeurs données à la fonction permettant l'exécution du code)
- Une valeur de retour (valeur renvoyée par la fonction après traitement)

Exemple 1.4.1 Voici un exemple de fonction convertissant un temps exprimé en heure, minutes et secondes en seconde.

```
# La duree exprimee en h,m,s est stockee dans un tableau a trois valeurs.
def conversionseconde(temps) :
    nbseconde = temps[0]*3600
    nbseconde += temps[1]*60
    nbseconde += temps[2]
    return nbseconde

t = [12,24,32]

print t[0],"heures",t[1],"minutes",t[2],"secondes est egale a"
print conversionseconde(t),"secondes"
```

12 heures 24 minutes 32 secondes est égale à 44672 secondes

1.5 Indentation du code

Les fonctions Python n'ont pas de begin ou end explicites, ni d'accolades qui pourraient marquer là où commence et où se termine le code de la fonction. Le seul délimiteur est les deux points (« :») et l'indentation du code lui-même.

Les blocs de code (fonctions, instructions if, boucles for ou while etc.) sont définis par leur indentation. L'indentation démarre le bloc et la désindentation le termine. Il n'y a pas d'accolades, de crochets ou de mots clés spécifiques. Cela signifie que les espaces blancs sont significatifs et qu'ils doivent être cohérents.

Cette particularité de Python rend ce langage très lisible mais capricieux en phase d'écriture. Plus de la moitié des erreurs d'un programmeur Python débutant sont des erreurs d'indentation.

Voici plusieurs exemples du même code. Repérer les erreurs d'indentation.

```
#longueur : arg un tableau - renvoie la longueur du tableau

def longueur(tableau):
    longueur = 0
    for elt in tableau:
        print("prise en compte de l'élément",elt)
        longueur = longueur + 1
        print("actualisation de la longueur",longueur)
        print("\n")

    if longueur == 0:
        print("Le tableau est vide")
    else :
        print("La longueur du tableau est",longueur)

    return longueur

tab = ['a','b','c','d','c','d','o','f']

longueur(tab)

#longueur : arg un tableau - renvoie la longueur du tableau

def longueur(tableau):
    longueur = 0
    for elt in tableau:
        print("prise en compte de l'élément",elt)
        longueur = longueur + 1
        print("actualisation de la longueur",longueur)
        print("\n")

    if longueur == 0:
        print("Le tableau est vide")
    else :
        print("La longueur du tableau est",longueur)

    return longueur

tab = ['a','b','c','d','c','d','o','f']

longueur(tab)
```

```
#longueur : arg un tableau - renvoie la longueur du tableau

def longueur(tableau):
    longueur = 0
    for elt in tableau:
        print("prise en compte de l'élément",elt)
        longueur = longueur + 1
        print("actualisation de la longueur",longueur)
        print("\n")

    if longueur == 0:
        print("Le tableau est vide")
    else :
        print("La longueur du tableau est",longueur)

    return longueur

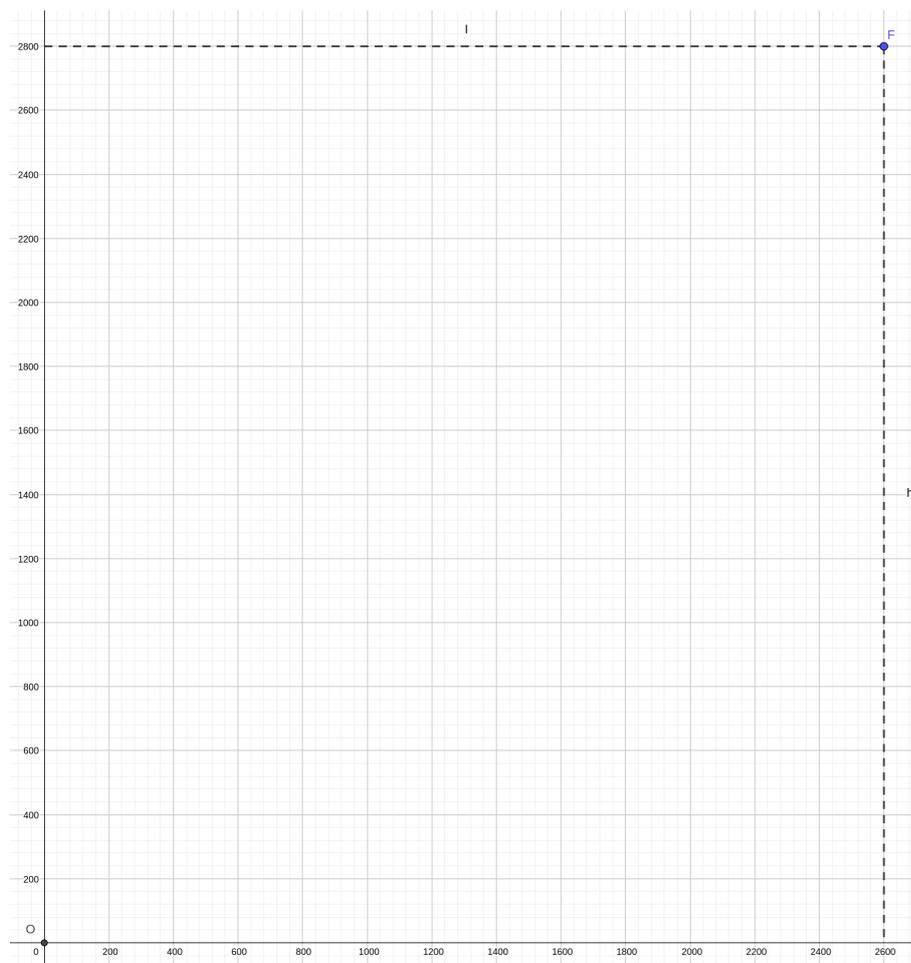
tab = ['a','b','c','d','c','d','o','f']

longueur(tab)
```


2. Image Numérique

2.1 Tâche finale

Exercice 2.1 On considère un rectangle défini par deux points O l'origine du repère et $F(l, h)$ avec l et h positifs.



1. Etude géométrique

- Déterminer en fonction de l et h , la distance OF
- Soit $M(x, y)$ avec $x \leq l$ et $y \leq h$. Déterminer, en fonction de x et y , la distance OM
- On note d , la distance OM et d_{max} , la distance OF . Déterminer un encadrement de d .

2. Etude fonctionnelle

Soit f une fonction affine, d'équation $f(x) = ax + b$ tel que :

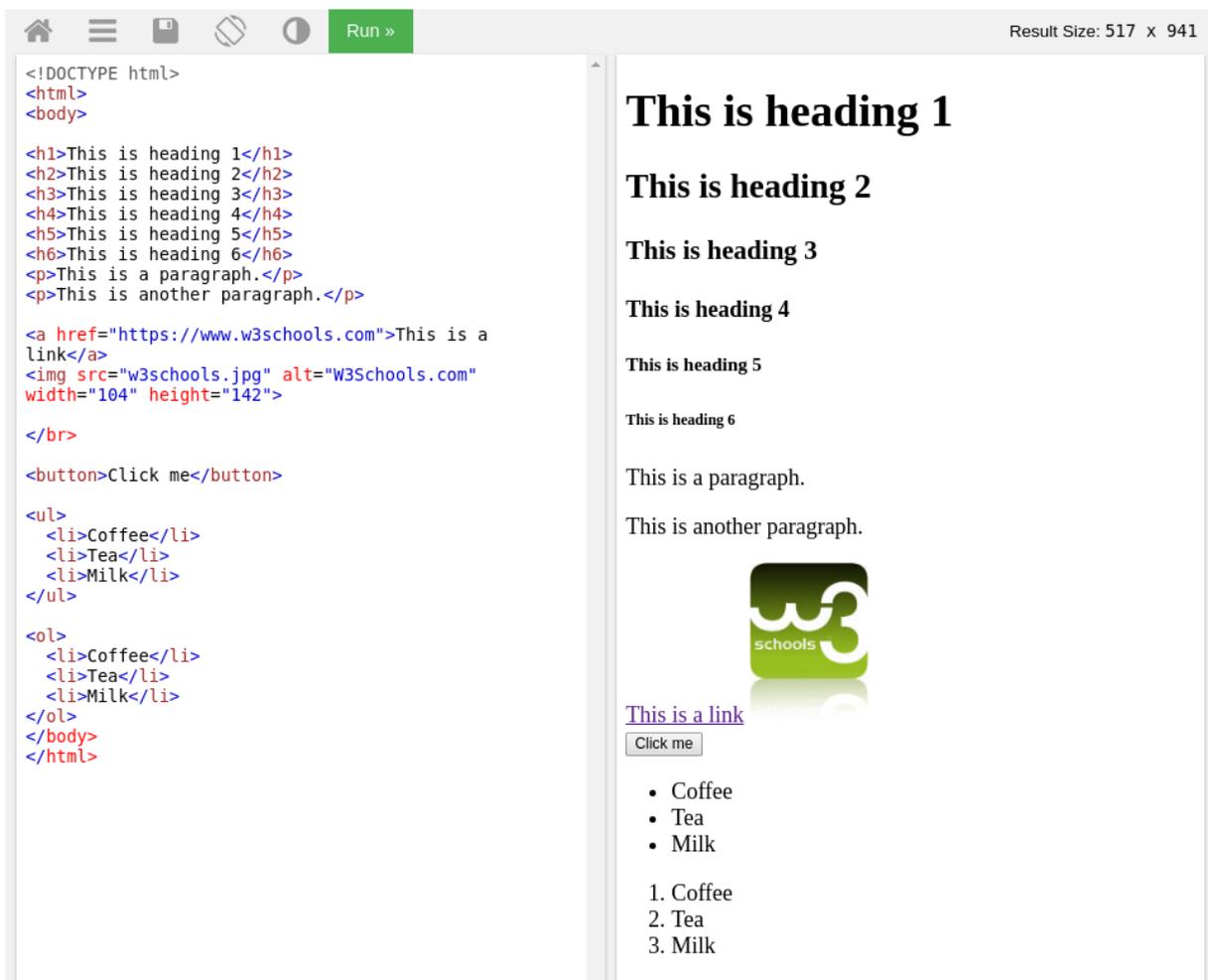
- $f(0) = 255$
 - $f(d_{max}) = 0$
- Déterminer en fonction de d_{max} , l'expression de a et de b .
 - Faire le lien avec cette fonction f et la tâche finale.

3. Le Web

3.1 Le langage HTML

Définition 3.1.1 — HTML : L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

Exemple 3.1.1 Voici un exemple d'une page et de son code html.



```
<!DOCTYPE html>
<html>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>

<a href="https://www.w3schools.com">This is a
link</a>


</br>

<button>Click me</button>

<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>

<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
</body>
</html>
```

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

This is heading 6

This is a paragraph.

This is another paragraph.



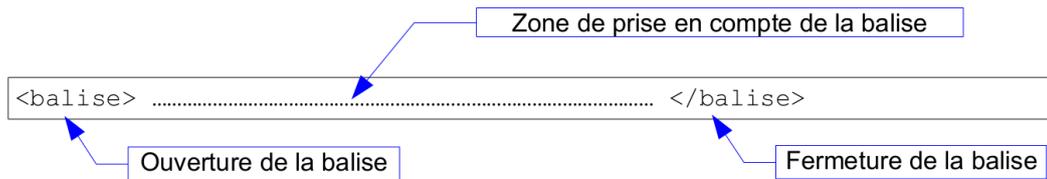
[This is a link](#)

Click me

- Coffee
- Tea
- Milk

1. Coffee
2. Tea
3. Milk

Définition 3.1.2 — Balise : On pourrait définir une balise comme un marqueur donnant un attribut particulier au texte qui l'entoure. Pour être prise en compte, une balise doit donc être ouverte (début de son action) puis refermée (fin de son action).



Exercice 3.1 Dans la définition de l'HTML, il est stipulé que ce langage est **balisable**. A l'aide de l'exemple ci-dessus, repérer ces balises et donner leur rôle.

Travaux Pratiques 3.1 Rendez-vous sur lms.zone pour effectuer le TP1. Les instructions sont disponibles sur la plateforme.

3.2 Mise en forme avancée : CSS

Les balises HTML permettent une mise en forme sommaire. En 1996, apparaît le CSS (Cascading Style Sheets). Ce langage permet un contrôle total de la mise en forme d'un site et facilite la mise à jour de cette dernière. Historiquement la mise en place du CSS a été complexe due à la guerre entre les navigateurs. Certaines fonctionnalités n'étant pas compatibles avec certains navigateurs. Depuis le naufrage d'Internet Explorer (aujourd'hui Edge), le CSS est à présent compatible pour quasiment tous les navigateurs.

Le CSS est obligatoire pour avoir un site ergonomique et un tant soit peu esthétique.

Exemple 3.2.1 Voici le votre site favori avec et sans CSS.

- Accueil
- Tableau de bord
- Calendrier
- Facteurs personnels
- Mes cours
- jériso-20nc
- snt01CA
- snt02CA
- snt03CA
- snt04CA
- snt05CA
- snt06CA
- snt07CA
- snt08CA
- snt09CA
- snt10CA
- snt11CA
- snt12CA
- snt13CA
- snt14CA
- snt15CA
- snt16CA
- snt17CA
- snt18CA
- snt19CA
- snt20CA
- snt21CA
- snt22CA
- snt23CA
- snt24CA
- snt25CA
- snt26CA
- snt27CA
- snt28CA
- snt29CA
- snt30CA
- snt31CA
- snt32CA
- snt33CA
- snt34CA
- snt35CA
- snt36CA
- snt37CA
- snt38CA
- snt39CA
- snt40CA
- snt41CA
- snt42CA
- snt43CA
- snt44CA
- snt45CA
- snt46CA
- snt47CA
- snt48CA
- snt49CA
- snt50CA
- snt51CA
- snt52CA
- snt53CA
- snt54CA
- snt55CA
- snt56CA
- snt57CA
- snt58CA
- snt59CA
- snt60CA
- snt61CA
- snt62CA
- snt63CA
- snt64CA
- snt65CA
- snt66CA
- snt67CA
- snt68CA
- snt69CA
- snt70CA
- snt71CA
- snt72CA
- snt73CA
- snt74CA
- snt75CA
- snt76CA
- snt77CA
- snt78CA
- snt79CA
- snt80CA
- snt81CA
- snt82CA
- snt83CA
- snt84CA
- snt85CA
- snt86CA
- snt87CA
- snt88CA
- snt89CA
- snt90CA
- snt91CA
- snt92CA
- snt93CA
- snt94CA
- snt95CA
- snt96CA
- snt97CA
- snt98CA
- snt99CA
- snt100CA

Lms

Gestion des cours

1. Accueil

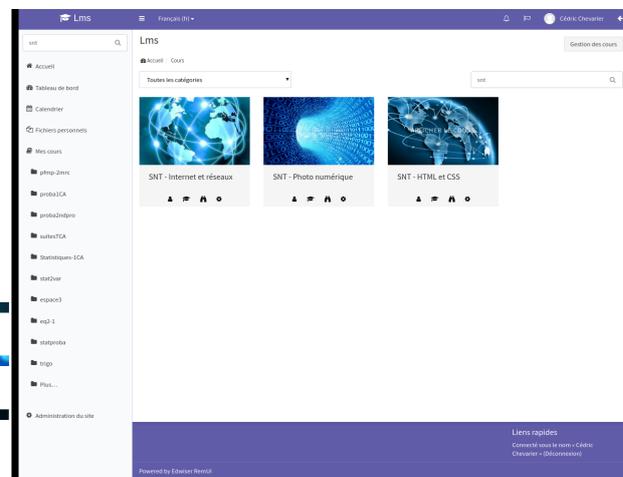
2. Cours

Tous les catégories

SNT - Internet et réseaux

SNT - Photo numérique

SNT - HTML et CSS



3.2.1 CSS : Style intégré à une balise HTML

Le mot clef `style` permet de rajouter ou de redéfinit des éléments de mise en forme des balises HTML existante.

Exemple 3.2.2 Voici un exemple d'élément de style CSS inclus dans des balises HTML.

The screenshot shows a web browser window. On the left, the HTML source code is visible, featuring inline styles for various elements. On the right, the rendered page is shown, demonstrating the effects of these styles.

HTML Code (Left):

```
<!DOCTYPE html>
<html>
  <head>
    <title>Welcome</title>
    <meta charset="utf-8" />
  </head>
  <body style="background : rgb(25, 51, 77);">
    <h1 style="text-decoration-line: underline;color:white;font: bold 32px Courier;">Title 1</h1>
    <h2 style="text-decoration-line: underline;color:white;font: bold 24px Courier;">Section 1</h2>
    <p style="color:white;font: 16x Courier;"> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris ornare urna sed neque accumsan sagittis. Donec scelerisque arcu ut risus consequat mattis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin dapibus pharetra sem vel ultricies. Aenean volutpat tempus felis ut elementum. Vestibulum mollis a nulla id lacinia. Nam arcu augue, ullamcorper sed elit non, suscipit convallis sem. Nam hendrerit, ex eu porta consectetur, velit ipsum tincidunt dolor, non aliquam elit justo vel massa. Duis at pellentesque sem, vitae gravida erat. Cras dui nulla, euismod nec efficitur id, porttitor ac nibh. Etiam pellentesque bibendum sapien, ut consectetur enim pharetra ut. Duis eu nunc nulla. Proin imperdiet hendrerit nibh tristique lobortis. In pharetra odio eu augue feugiat, at molestie leo elementum.</p>
    <h2 style="text-decoration-line: underline;color:white;font: bold 24px Courier;">Section 2</h2>
    <p style="color:white;font: 16x Courier;">Ut tempor, quam nec elementum tincidunt, purus diam tincidunt magna, ut interdum nisi ligula et mi. Maecenas at imperdiet purus. Ut semper metus mi, ac tincidunt purus elementum vitae. Curabitur vestibulum sed erat sed mattis. Nam a dolor et ante commodo pulvinar. Etiam tempor lacinia sollicitudin. Maecenas dapibus est quam, quis finibus urna congue at. Proin turpis turpis, bibendum sed tempus et, fermentum vitae purus. Sed tristique nibh purus, id luctus libero iaculis non. Vestibulum et tortor non lectus maximus laoreet non fringilla turpis. Maecenas eget lorem sit amet quam faucibus pharetra nec non arcu. Morbi sed elementum nisi, in vestibulum lectus. Nullam convallis sem augue, vitae consectetur tortor volutpat id.</p>
    <h2 style="text-decoration-line: underline;color:white;font: bold 24px Courier;">Section 3</h2>
    <p style="color:white;font: 16x Courier;">Aenean molestie ante iaculis velit pellentesque, a facilisis urna ornare. Morbi pretium, nulla non accumsan ultricies, tellus eros fringilla orci, a malesuada augue dolor id lorem. Donec non dolor a enim lacinia porta. Aliquam erat volutpat. Duis neque enim, pulvinar in volutpat eu, pretium a sapien. Integer tincidunt magna lorem, et feugiat risus accumsan vel. Aliquam rutrum eleifend metus a venenatis. Proin semper luctus pellentesque.</p>
    <h2 style="text-decoration-line: underline;color:white;font: bold 24px Courier;">Section 4</h2>
    <p style="color:white;font: 16x Courier;">Integer eget est ac turpis malesuada scelerisque a vitae diam. Phasellus maximus porta dolor a ullamcorper. Sed quis interdum libero. Nam interdum velit quis elit imperdiet scelerisque. Curabitur at arcu a nibh venenatis ultrices non tempus lorem. Pellentesque mauris nisi, posuere sit amet mattis eget, consectetur at nisi. Nulla nunc lacus, vestibulum quis neque eget, iaculis congue erat. Suspendisse ac metus ex. Nulla condimentum arcu a ipsum cursus hendrerit. Nulla a libero vel ex congue placerat ut id nisi. Curabitur eleifend nunc vel iaculis condimentum. Fusce eu arcu tortor.</p>
    <h2 style="text-decoration-line: underline;color:white;font: bold 24px Courier;">Section 4</h2>
```

Rendered Output (Right):

The rendered page shows a dark blue background. The title "Title 1" is underlined and in a large, bold, white font. Below it, "Section 1" is also underlined and in a bold, white font. The text of Section 1 is in a white font. Section 2 and Section 3 are also underlined and in a bold, white font. The text of Section 2 and Section 3 is in a white font. Section 4 is underlined and in a bold, white font. The text of Section 4 is in a white font.

Exercice 3.2 Rendez-vous sur lms.zone pour faire cet exercice puis répondre aux questions suivantes :

1. Quels sont les principaux inconvénients à rajouter du code CSS dans les balises HTML ?

2. Proposer une solutions aux inconvénients précédents.

3.3 Séparer contenu et mise en forme : la bonne solution

Cette solution vise à créer deux fichiers :

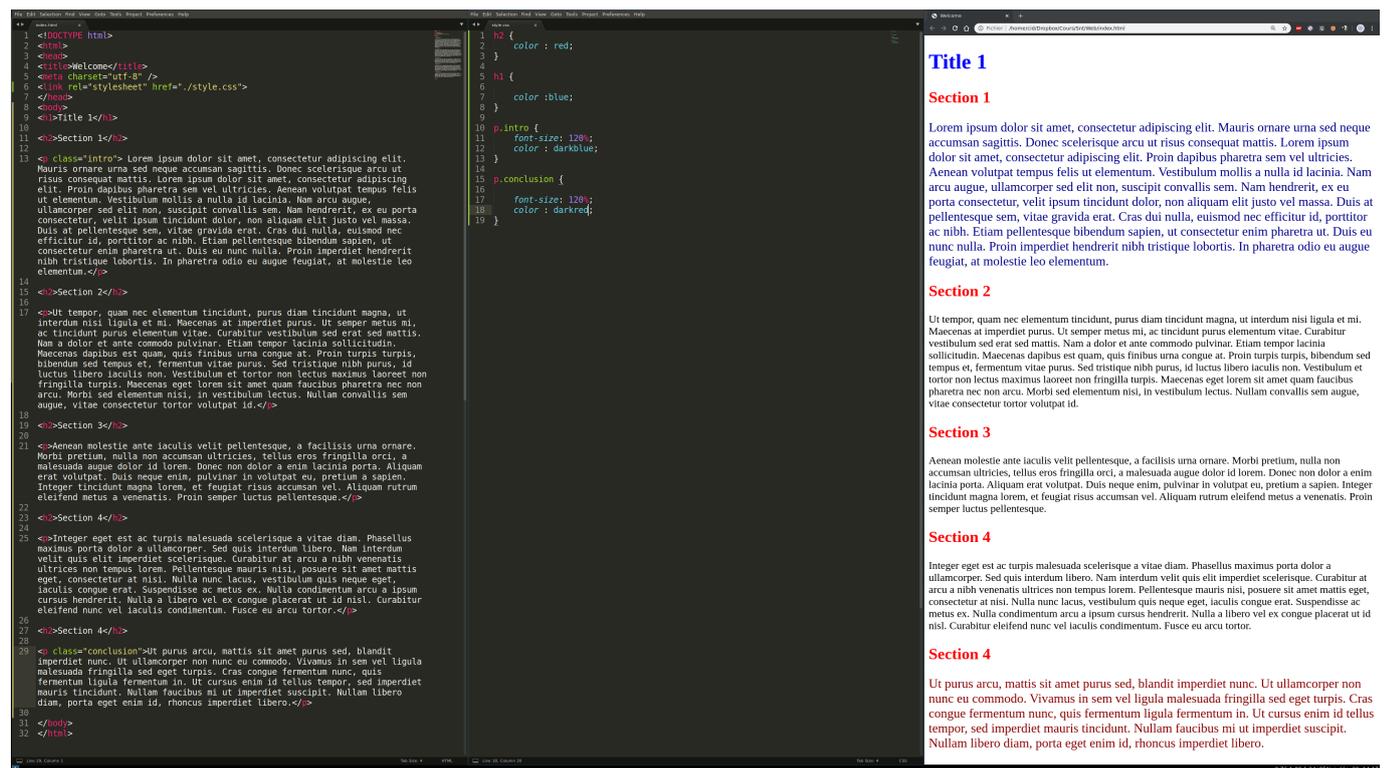
- Le fichier de contenu contient le texte brut. Ce texte est simplement balisé pour identifier le type de contenu grâce aux classes qu'on définira un peu plus tard.
- Le fichier mise en forme contient toutes les règles de mise en page de chacune des classes.

Cette solution permet à un créateur de se focaliser sur le contenu en ajoutant uniquement des balises pour le typer. Un fichier de mise en forme regroupe toutes les instructions afin que cette dernière soit uniforme. Cette technique est particulièrement utile dans le cas d'un site possédant plusieurs rédacteurs et permet une uniformité de la mise en page du site.

La mise en oeuvre est la suivante :

- Un fichier `index.html` regroupe le contenu de la page avec un contenu balisé.
- Un fichier `style.css` regroupe les informations de mise en page pour les balises personnalisées. Ce fichier est appelé dans la tête (head) du fichier HTML.

Exemple 3.3.1 Voici un exemple : à gauche le fichier HTML, au centre le fichier CSS et à droite le rendu dans un navigateur



Les détails de cette mise en oeuvre vous sont donnés sur lms.zone.

La séparation contenu / mise en forme n'est pas réservée au site web, elle peut être aussi utilisée pour des documents standards.

Exemple 3.3.2 Voici votre cours : à gauche le contenu balisé, à droite le fichier de mise en forme (langage : LaTeX).

```

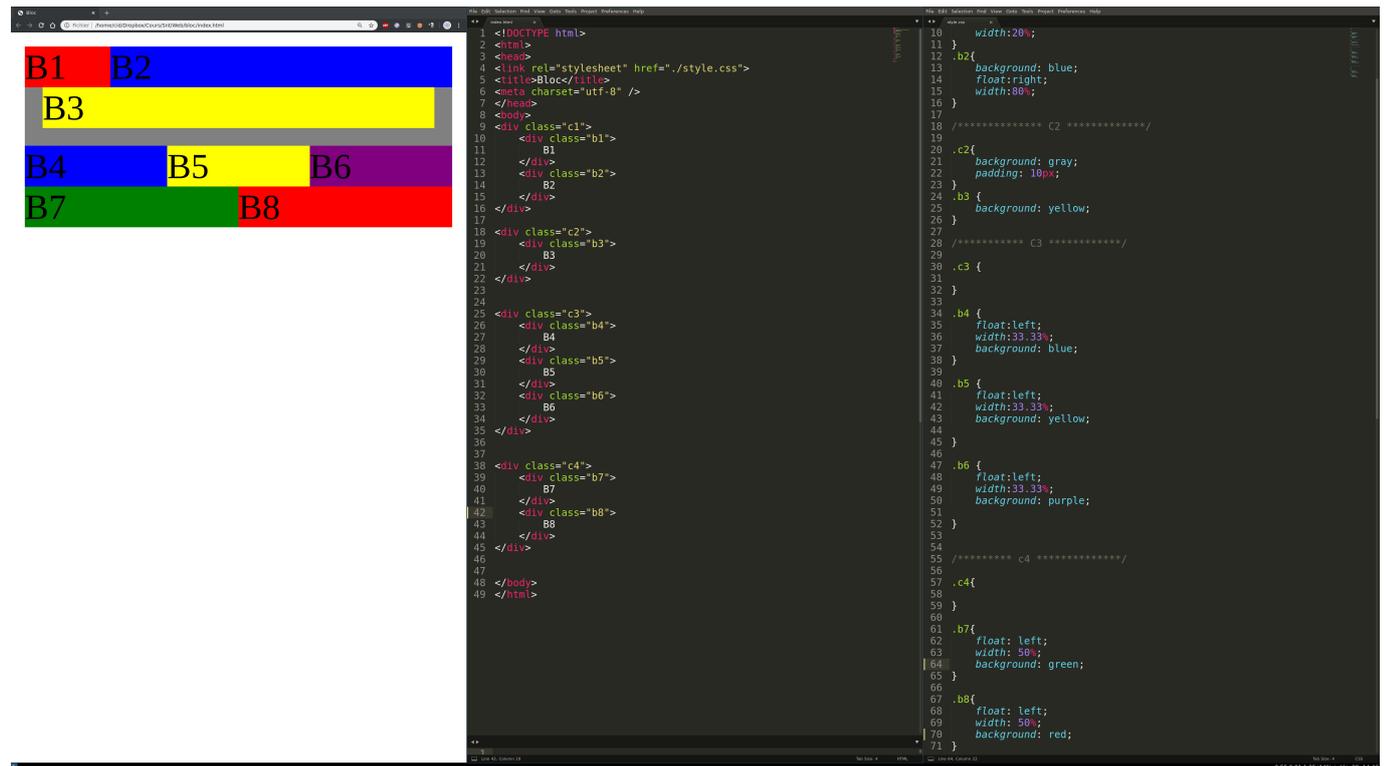
577 pour être prise en compte, une balise doit être ouverte (debut de son action) puis fermée (fin
578 de son action).
579 \begin{center}
580 \includegraphics[width=14cm]{html2.png}
581 \end{center}
582 \end{definition}
583
584 \begin{exercice}
585 Dans la définition de L'HTML, il est stipulé que ce langage est \textbf{balisable}. À l'aide de l'exemple ci-dessus,
586 repérez ces balises et donnez leur rôle.\\
587 \lignereponse{5}
588 \end{exercice}
589
590 \begin{travail}
591 Rendez-vous sur lms.zone pour effectuer le TP1. Les instructions sont disponibles sur la plateforme.
592 \end{travail}
593
594 \section{Mise en forme avancée : CSS}
595
596 Les balises HTML permettent une mise en forme sommaire. En 1996, apparaît le CSS (Cascading Style Sheets). Ce langage
597 permet un contrôle total de la mise en forme d'un site et facilite la mise à jour de cette dernière. Historiquement la
598 mise en place du CSS a été complexe due à la guerre entre les navigateurs. Certaines fonctionnalités n'étant pas
599 compatibles avec certains navigateurs, depuis le naufrage d'Internet Explorer (aujourd'hui Edge), le CSS est à présent
600 compatible pour quasiment tous les navigateurs.
601
602 Le CSS est obligatoire pour avoir un site ergonomique et un tant soit peu esthétique.
603
604 \begin{exemple}
605 Voici le votre site favori avec et sans CSS.
606 \end{exemple}
607
608 \begin{center}
609 \includegraphics[width=16cm]{ascss.png}
610 \end{center}
611 \end{exemple}
612
613 \subsection{CSS : Style intégré à une balise HTML}
614
615 Le mot clef \verb|style| permet de rajouter ou de redéfinir des éléments de mise en forme des balises \verb|HTML|
616 existante.
617
618 \begin{exemple}
619 Voici un exemple d'élément de style \verb|CSS| inclus dans des balises \verb|HTML|.
620 \end{exemple}
621
622 \begin{center}
623 \includegraphics[width=18cm]{htmlcss.png}
624 \end{center}
625 \end{exemple}
626
627 \begin{exercice}
628 Rendez-vous sur lms.zone pour faire cet exercice puis répondre aux questions suivantes :
629
630 \item Quels sont les principaux inconvénients à rajouter du code \verb|CSS| dans les balise \verb|HTML| ? \\
631 \lignereponse{4}
632 \item Proposer une solutions aux inconvénients précédents.\\
633 \lignereponse{4}
634 \end{exercice}
635
636 \end{exercice}
637
638 \section{Séparer contenu et mise en forme : la bonne solution}
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

3.4 Mise en forme complexe

La balise `<div></div>` de type block : elle délimite des éléments en bloc de la page web. Cette balise permet de définir des containers et de définir le placement et la mise en forme des éléments de votre site.

Exemple 3.4.1 Voici un exemple de l'utilisation de divisions. De gauche à droite, le rendu, le fichier HTML et le fichier CSS.



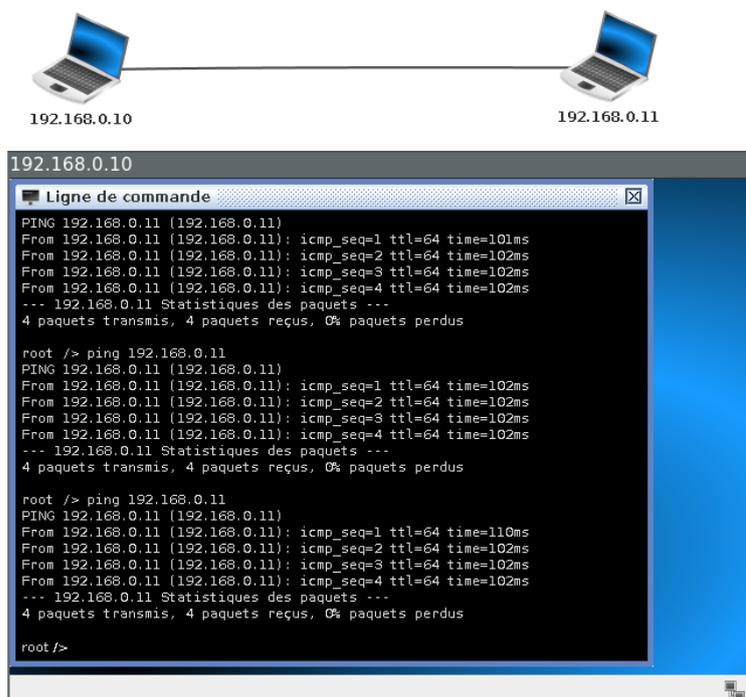
La mise en œuvre complète de cet exemple et les explications relatives au `<div></div>` sont disponibles sur lms.zone

4. Internet et réseaux

4.1 Adresse IP

Définition 4.1.1 — Adresse IP. Adresse IP numéro d'identification de chaque appareil connecté à un réseau.

Exemple 4.1.1 Voici l'exemple de deux portables reliés entre eux par un câble réseau. Celui de gauche possède l'adresse IP 192.168.0.10 et celui de droite 192.168.0.11.



A partir de l'ordinateur de gauche (192.168.0.10) on effectue la commande ping 192.168.0.11. Cette commande envoie un paquet de données à l'ordinateur dont on a précisé l'adresse IP et mesure le temps mis par le paquet pour faire l'aller-retour. On peut aussi voir le pourcentage de paquets perdus dans l'opération. Cette commande est très utile pour vérifier la qualité de la connexion entre deux ordinateurs.

L'adresse IP est un outil complexe ; pour ce cours nous utiliserons la décomposition standard pour les réseaux domestiques et d'entreprises de taille moyenne.

Exemple 4.1.2 Voici l'adresse IP d'une machine.

192.168.2.54

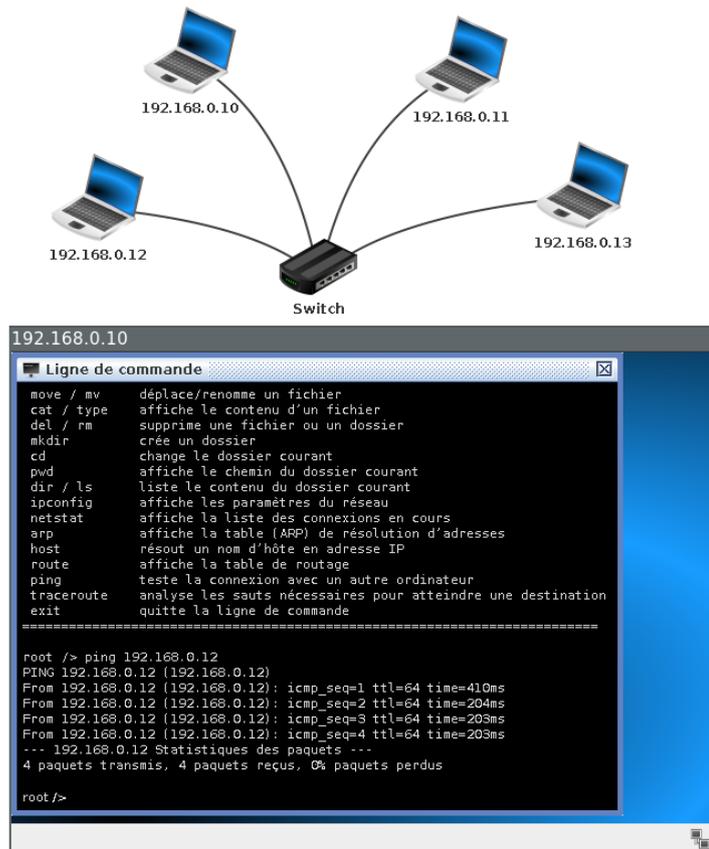
Adresse du réseau hôte

Cette machine est le numéro 54 du réseau 192.168.2.

4.2 Les composants d'un réseau

Définition 4.2.1 — Switch. Switch réseau (commutateur) Un switch Ethernet (en Français, un commutateur) est un concentrateur qui permet d'augmenter le nombre d'ordinateurs connectés sur un réseau Ethernet (topologies en étoile) en local (ce qu'on appelle en jargon informatique un LAN (Local Area Network)).

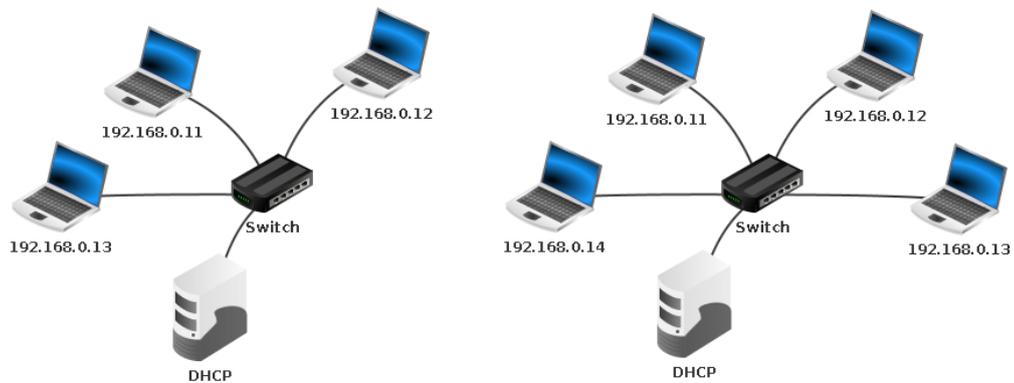
Exemple 4.2.1 Voici l'exemple de quatre portables connectés entre eux par un switch.



Exercice 4.1 Dans l'exemple précédent, regardez attentivement le temps mis par les paquets pour faire un aller-retour. Que constatez-vous ? Donnez une explication.

Définition 4.2.2 — DHCP. Le DHCP (Dynamic Host Configuration Protocol) est un protocole réseau chargé de la configuration automatique des adresses IP d'un réseau informatique. Il évite ainsi à l'utilisateur qui se connecte à un réseau, d'avoir à configurer l'IP de son équipement. Autrement dit un serveur DHCP permet à un équipement d'obtenir une IP automatiquement.

Exercice 4.2 Dans un réseau équipé d'un serveur DHCP on ajoute un nouvel ordinateur.



Que s'est-il passé ? Quel est le principal inconvénient du DHCP ?

■

Exercice 4.3 Voici la fenêtre de configuration d'un service DHCP.

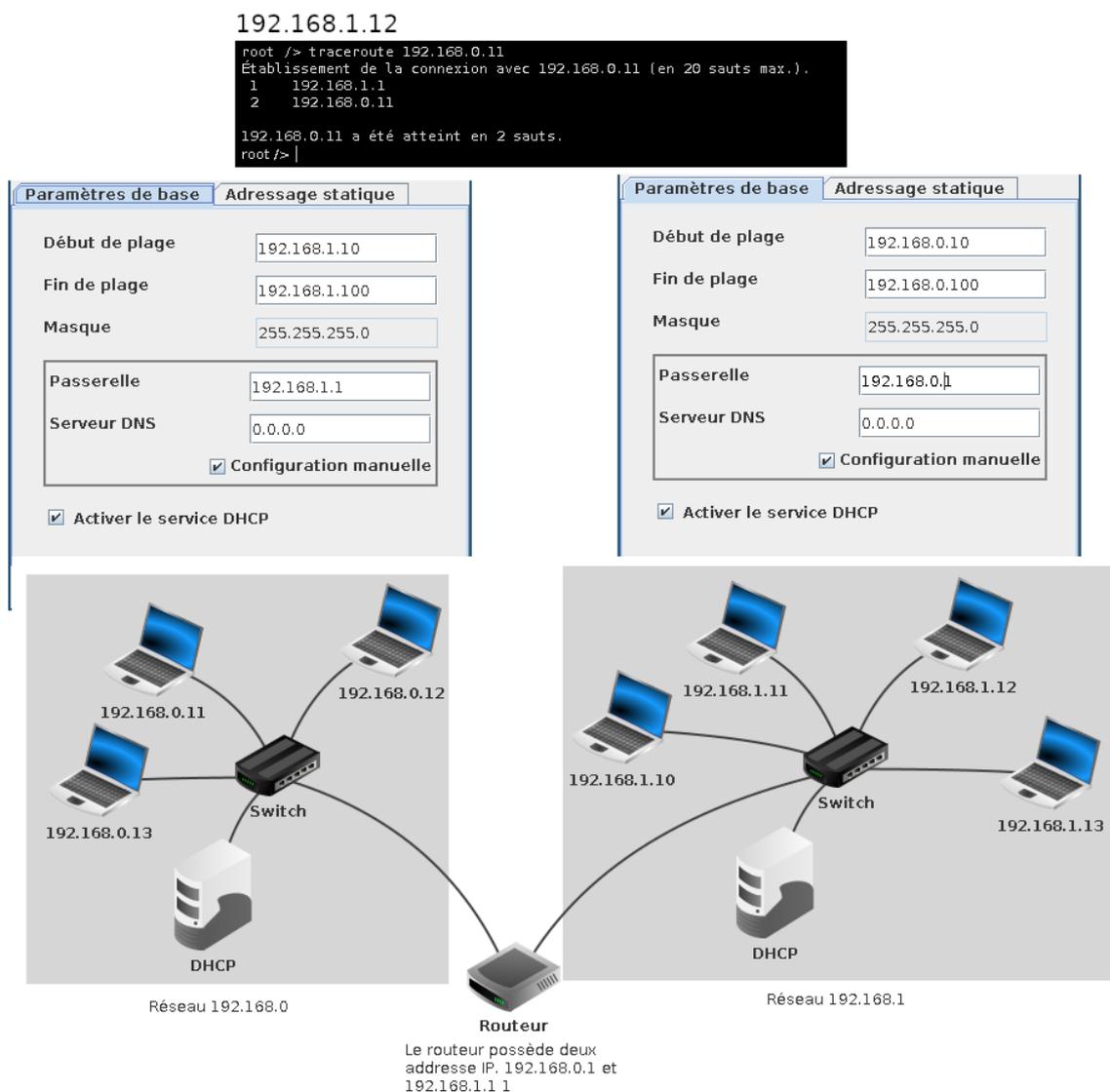
Paramètres de base	Adressage statique
Début de plage	192.168.0.10
Fin de plage	192.168.0.100
Masque	255.255.255.0
Passerelle	0.0.0.0
Serveur DNS	0.0.0.0
	<input type="checkbox"/> Configuration manuelle
<input checked="" type="checkbox"/> Activer le service DHCP	
OK	

A qui correspondent les paramètres "Début de la plage" et "Fin de la plage" ?

■

Définition 4.2.3 — Routeur. Un routeur est un équipement réseau informatique assurant le routage des paquets. Son rôle est de faire transiter des paquets d'une interface réseau vers une autre, au mieux, selon un ensemble de règles.

Exemple 4.2.2 Dans cet exemple, on dispose de deux réseaux "192.168.0" et "192.168.1", ces deux réseaux sont reliés par un routeur qui possède une IP dans chacun des réseaux. Le routeur sert donc de passerelle entre les deux réseaux. L'IP de ce routeur/passerelle doit être indiqué pour chaque machine si l'on désire pouvoir envoyer des données en dehors du réseau.



La fonction traceroute permet de connaître les sauts effectués par un paquet d'un point à un autre.

Exercice 4.4 En vous basant sur l'exemple précédent, décrivez le plus précisément possible le trajet du paquet émis par la fonction traceroute.

■

Travaux Pratiques 4.1 Une entreprise contient plusieurs départements :

- Administration : 8 ordinateurs
- Production : 8 ordinateurs
- Service qualité : 5 ordinateurs
- Service commercial : 5 ordinateurs
- Recherche et développement : 4 ordinateurs
- SAV : 3 ordinateurs

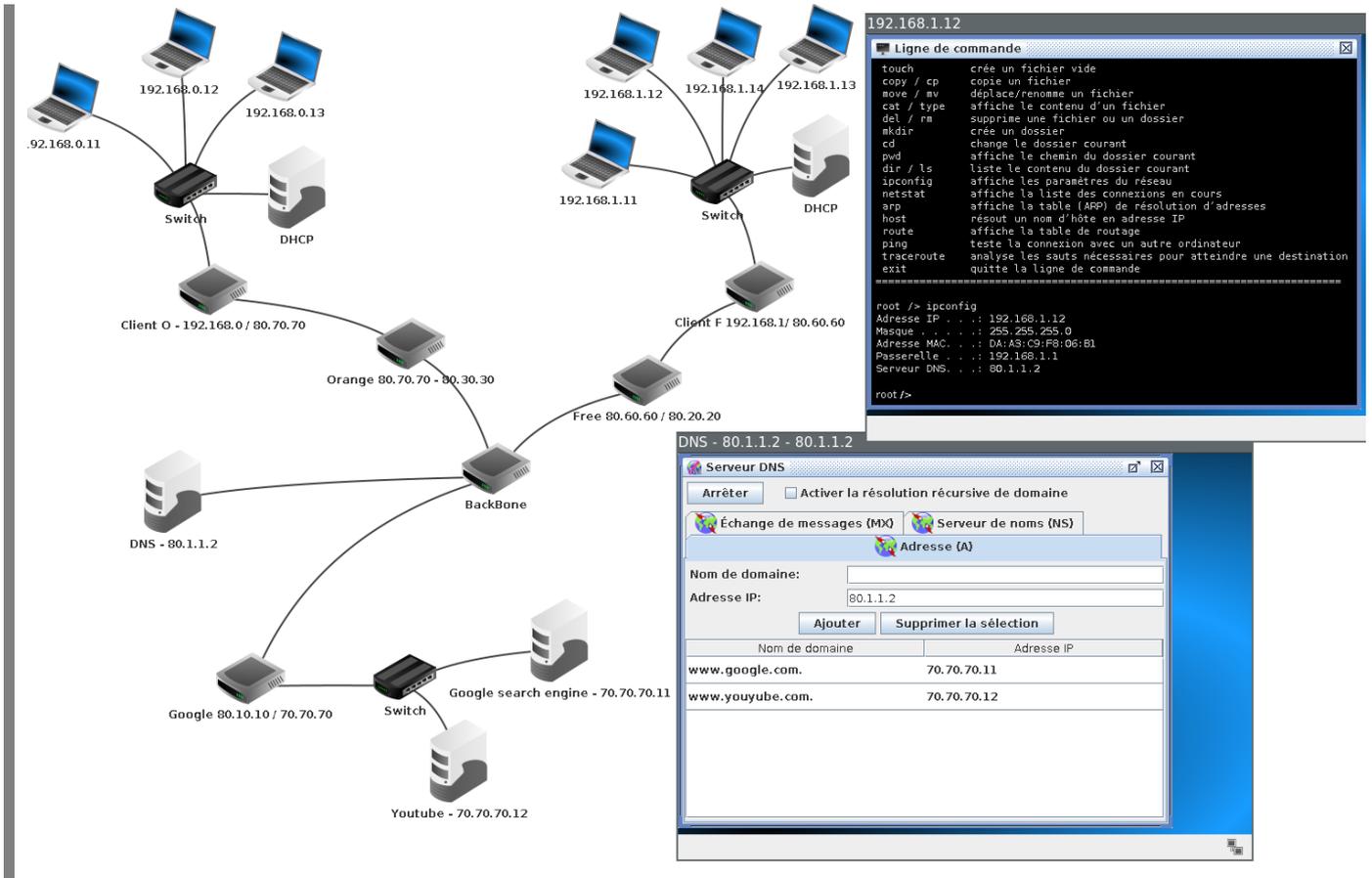
Organisez un réseau (pouvant être composé de plusieurs sous-réseaux) où l'ensemble des machines obtiennent leur IP par DHCP et sont toutes capables de communiquer entre elles. Vous simulerez votre réseau à l'aide du logiciel filius ou fournirez un document texte expliquant la structuration de votre réseau et les tests que vous avez effectués afin de vérifier que ce dernier soit fonctionnel. Un bonus sera accordé si vous optimisez (minimisez) le nombre de sauts nécessaire pour établir la communication entre deux machines.

4.3 Routage et DNS

Dans le cas de l'utilisation de plusieurs routeurs interconnectés, il est indispensable de définir un table de routage qui permettra de donner un paquet la marche à suivre afin d'atteindre sa destination.

Définition 4.3.1 — DNS. Le Domain Name System, généralement abrégé DNS, qu'on peut traduire en « système de noms de domaine », est le service informatique distribué utilisé pour traduire les noms de domaine Internet en adresse IP. Un serveur DNS permet par exemple à partir du nom de domaine "google.fr" de récupérer l'adresse IP du serveur de google et de pouvoir donc communiquer avec lui.

Exemple 4.3.1 Voici un exemple simulant le fonctionnement du WWW.



Exercice 4.5 Expliquez les étapes d'une requête "http ://www.google.com" effectuée par le PC dont l'IP est 192.168.1.12.

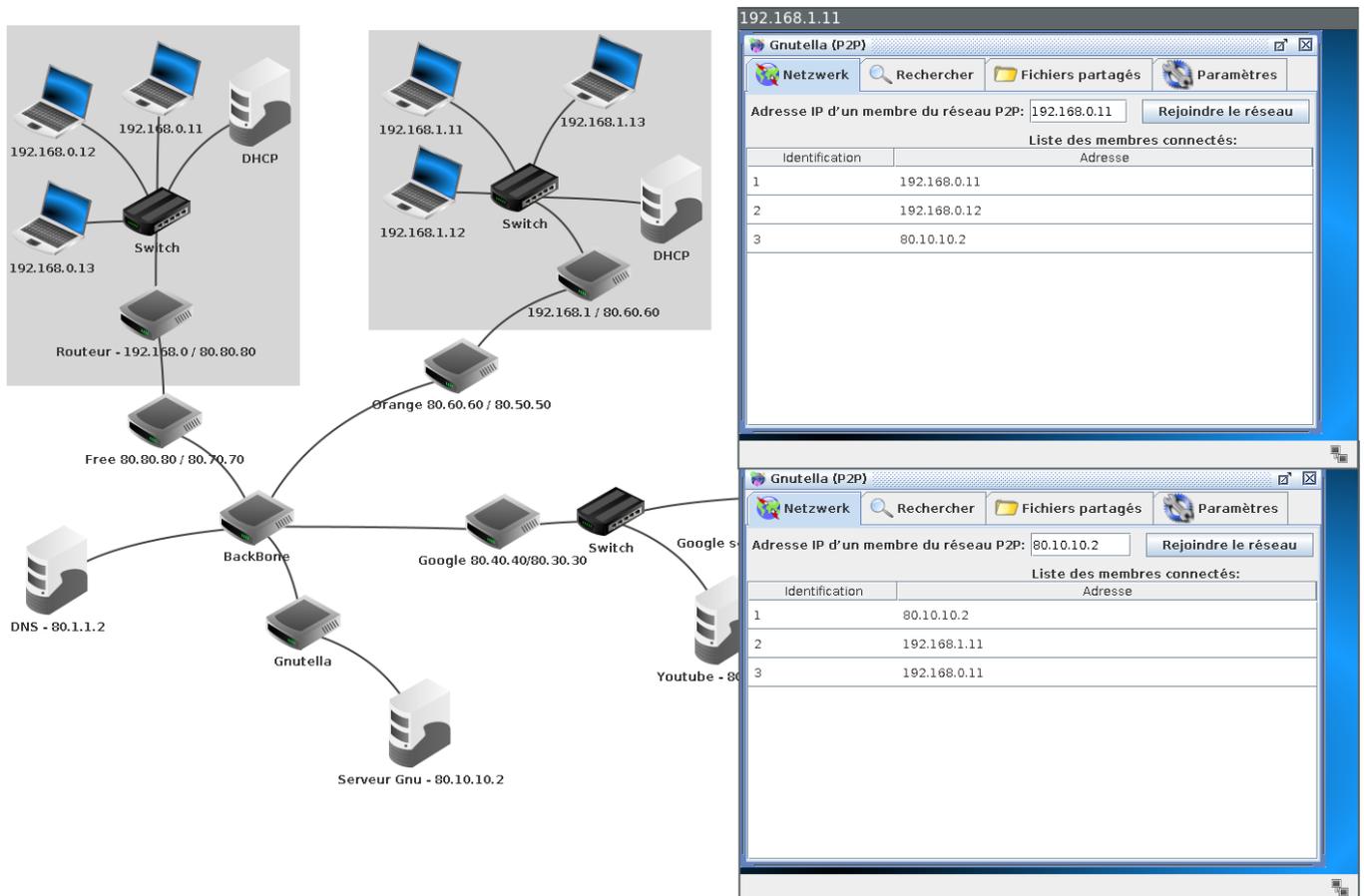
Travaux Pratiques 4.2 A partir du fichier de simulation de l'exemple précédent rajoutez les éléments suivants :

- Un autre client du réseau orange
- Un autre FAI ainsi qu'un client de ce dernier
- Un autre site web non affilié à google.

Définition 4.4.1 — P2P. Le pair-à-pair (en anglais peer-to-peer, souvent abrégé « P2P ») est un modèle d'échange où chaque entité du réseau est à la fois client et serveur, contrairement au modèle client-serveur.

Pour se connecter à un réseau P2P, il suffit de connaître l'IP d'un des membres du réseau. Ce membre vous transmettra alors l'annuaire contenant tous les membres actifs du réseau. Chaque membre (pair) du réseau peut donc émettre une requête (client) ou répondre à une requête (serveur). On crée donc un réseau dans le réseau.

Exemple 4.4.1 Voici un exemple d'un réseau P2P



- L'ordinateur 192.168.1.11 rejoint le réseau en se connectant à 192.168.0.11.
- L'ordinateur 192.168.0.12 rejoint le réseau en se connectant à 80.10.10.2.

A chaque fois qu'un nouvel ordinateur se connecte au réseau, l'annuaire du réseau se met à jour de pair en pair.

En rajoutant un couche d'anonymisation à un raison P2P, on obtient le Darknet.



La darknet est essentiellement utilisé par :

- Les dissidents politiques
- Pour des commerces illégaux
- Reporters sans frontières
- Les lanceurs d'alerte.

Travaux Pratiques 4.3 A partir du fichier associé à l'exemple précédent :

- Rajouter des membres au réseau P2P
- Echanger des fichiers grâce au réseau P2P
- Ecrire un descriptif des échanges lors des recherches et du téléchargement de fichier.

5. Structure de donnée

5.1 Base de données

Définition 5.1.1 — Base de données. Une base de données (que nous nommerons bdd par commodité) est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour. Au sein d'une database, les données sont organisées en lignes, colonnes et tableaux. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées.

Il existe de multiples modèles de base de données. Pour ce cours, nous nous limiterons à la découverte du modèle relationnel. Ce modèle regroupe, les données sous forme de tables, les informations sont indexées afin de lier les informations entre les tables. L'exemple suivant illustre ces notions :

Exemple 5.1.1 Voici la base de données IMDB (Internet Movie Database) qui est une base de données en ligne sur le cinéma mondial.



Voici deux tables de cette bdd. La table "base" regroupe les titres, longueur, genre, année d'exploitation etc.. de chaque film. La table "rate" regroupe la note moyenne de chaque film et le nombre de votes.

base_custom				rate_custom		
tconst	originalTitle	startYear	runtimeMinutes	tconst	averageRating	numVotes
tt0000009	Miss Jerry	1894	45	tt0000009	5.4	87
tt0000147	The Corbett-Fitzsimmons Fight	1897	20	tt0000147	5.2	331
tt0000574	The Story of the Kelly Gang	1906	70	tt0000574	6.1	559
tt0000679	The Fairylogue and Radio-Plays	1908	120	tt0000679	4.8	33
tt0001184	Don Juan de Serrallonga	1910	58	tt0001184	3.1	11
tt0001258	Den hvide slavehandel	1910	45	tt0001258	5.7	79
tt0001285	The Life of Moses	1909	50	tt0001285	5.1	30
tt0001498	The Battle of Trafalgar	1911	51	tt0001498	7.2	5
tt0001790	Les misérables - Époque 1: Jean Valjean	1913	60	tt0001790	5.8	21
tt0001812	Oedipus Rex	1911	56	tt0001812	5.8	6
tt0001892	Den sorte drøm	1911	53	tt0001892	5.9	180
tt0001911	Sweet Nell of Old Drury	1911	50	tt0001911	4.1	7
tt0001964	Die Verräterin	1911	48	tt0001964	6.1	52
tt0002026	Anny - en gatepiges roman	1912	68	tt0002026	4.0	7
tt0002101	Cleopatra	1912	100	tt0002101	5.2	431
tt0002130	L'Inferno	1911	68	tt0002130	7.0	2088
tt0002153	Dødsspring til hest fra cirkuskuplen	1912	45	tt0002153	5.7	43
tt0002186	Den flyvende cirkus	1912	46	tt0002186	6.3	79
tt0002199	From the Manger to the Cross; or, Jesus of Nazareth	1912	60	tt0002199	5.7	456
tt0002423	Madame DuBarry	1919	85	tt0002423	6.8	727
tt0002445	Quo Vadis?	1913	120	tt0002445	6.1	252
tt0002452	Independenta Romaniei	1912	120	tt0002452	6.6	194
tt0002461	Richard III	1912	55	tt0002461	5.5	217
tt0002514	De svartas maskerna	1912	60	tt0002514	5.9	7
tt0002588	Zigomar contre Nick Carter	1912	48	tt0002588	6.4	14
tt0002628	Der Andere	1913	48	tt0002628	5.8	64
tt0002646	Atlantis	1913	121	tt0002646	6.7	321
tt0002767	The Count of Monte Cristo	1913	69	tt0002767	5.6	37
tt0002822	What 80 Million Women Want	1913	56	tt0002822	4.2	26
tt0002832	Evangeline	1914	50	tt0002832	7.0	11
tt0002844	En stjerne på tolv måneder	1912	54	tt0002844	7.0	1079

Exercice 5.1 Le modèle relationnel est assez intuitif. Répondre aux questions suivantes à l'aide des deux tables ci-dessus :

1. Déterminer la note ainsi que le nombre de votant pour le film "L'inferno".

2. Quels sont les films ayant obtenu une note de 5,8 ?

3. Quels sont les films ayant reçus plus de 500 votes ?

Maintenant que vous avez expérimenté l'utilisation des tables, déterminer l'élément qui relie ces deux tables ?

■

Définition 5.1.2 — Clé primaire. Dans une base de données relationnelle, une clé primaire est la donnée qui permet d'identifier de manière unique un enregistrement dans une table.

Dans notre cas, la clé primaire est la colonne 'tconst'.

R La base de données de IMDB dépasse les 2Go, il est donc impossible de l'ouvrir avec des logiciels standards. Il faut donc un langage permettant d'interroger cette bdd. Le langage le plus répandu pour effectuer ce type d'action est le SQL (Server Query Language) mais compte tenu de votre amour inconditionnel du Python, nous allons utiliser Pandas qui est une bibliothèque de Python permettant la manipulation de bdd. Autre point, pour les besoins de portabilité du cours et compte tenu des performances des PC dont nous disposons les tables IMDB ont été très fortement allégées.

5.2 Manipulation d'une base de données

Même dans sa version réduite, les tables de la base de données IMDB se composent de plus de 3000 pages. De ce fait, un outil informatique est indispensable pour consulter, interroger et mettre à jour cette bdd. L'ensemble des opérations suivantes fait l'objet d'une vidéo d'illustration disponible sur lms.zone

5.2.1 Importation d'une table

Pour importer la table, nous allons utiliser un fichier csv (Comma Separated Values). Comme son nom l'indique, ce format sépare les données par des virgules (comma en anglais). La première ligne donne le nom des colonnes puis à chaque ligne correspond une nouvelle entrée dans la table. Voici à quoi ça ressemble :

```
tconst,originalTitle,startYear,runtimeMinutes,genres
tt0000009, Miss Jerry,1894,45,Romance
tt0000147, The Corbett-Fitzsimmons Fight,1897,20, "Documentary,News,Sport"
tt0000574, The Story of the Kelly Gang,1906,70, "Biography,Crime,Drama"
tt0000679, The Fairylogue and Radio-Plays,1908,120, "Adventure,Fantasy"
tt0001184, Don Juan de Serrallonga,1910,58, "Adventure,Drama"
tt0001258, Den hvide slavehandel,1910,45,Drama
tt0001285, The Life of Moses,1909,50, "Biography,Drama,Family"
tt0001498, The Battle of Trafalgar,1911,51,War
tt0001790, Les misérables - Époque 1: Jean Valjean,1913,60,Drama
tt0001812, Oedipus Rex,1911,56,Drama
tt0001892, Den sorte drøm,1911,53,Drama
tt0001911, Sweet Nell of Old Drury,1911,50, "Biography,Drama,History"
tt0001964, Die Verräterin,1911,48,Drama
tt0002026, Anny - en gatepiges roman,1912,68, "Drama,Romance"
tt0002101, Cleopatra,1912,100, "Drama,History"
```

Le fichier correspondant à la table comportant les noms des films est stockée dans le fichier `base_custom.csv` et la table contenant les évaluations des films est contenue dans la fichier `rate_custom.csv`

Exemple 5.2.1 Dans ce code nous allons importer les deux tables, puis vérifier le bon fonctionnement en affichant les 10 premières lignes de la table base.

```
#Importation de la bibliotheque Panda pour la manipulation de bdd
import pandas as pd

#On appelle la fonction read_csv dans panda en rajoutant le
#prefixe pd en stipulant le caractere de separation et la clef primaire
movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")

#La fonction head(n) renvoie les n premieres lignes d'une table
print(movies_base.head(10))
```

tconst	originalTitle	startYear	runtimeMinutes	genres
tt0000009	Miss Jerry	1894	45	Romance
tt0000147	The Corbett-Fitzsimmons Fight	1897	20	Documentary,News,Sport
tt0000574	The Story of the Kelly Gang	1906	70	Biography,Crime,Drama
tt0000679	The Fairylogue and Radio-Plays	1908	120	Adventure,Fantasy
tt0001184	Don Juan de Serrallonga	1910	58	Adventure,Drama
tt0001258	Den hvide slavehandel	1910	45	Drama
tt0001285	The Life of Moses	1909	50	Biography,Drama,Family
tt0001498	The Battle of Trafalgar	1911	51	War
tt0001790	Les misérables - Époque 1: Jean Valjean	1913	60	Drama
tt0001812	Oedipus Rex	1911	56	Drama

5.2.2 Sélection de données

Sélection simple

La sélection simple consiste en la sélection d'une partie d'une table sans condition.

Exemple 5.2.2 Dans cet exemple, nous allons simplement sélectionner les titres et les années d'exploitation des films dans la table `movies_bases`. La table sera donc amputée des autres colonnes à l'exception de la colonne `tconst` qui est la clef primaire.

```
import pandas as pd

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
```

```

movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")

movies_base_selected = movies_base[['startYear', 'originalTitle']]

print(movies_base_selected.head(10))

```

tconst	startYear	originalTitle
tt0000009	1894	Miss Jerry
tt0000147	1897	The Corbett-Fitzsimmons Fight
tt0000574	1906	The Story of the Kelly Gang
tt0000679	1908	The Fairylogue and Radio-Plays
tt0001184	1910	Don Juan de Serrallonga
tt0001258	1910	Den hvide slavehandel
tt0001285	1909	The Life of Moses
tt0001498	1911	The Battle of Trafalgar
tt0001790	1913	Les misérables - Époque 1: Jean Valjean
tt0001812	1911	Oedipus Rex

Exercice 5.2 En modifiant le programme précédent, sélectionner dans la base de données uniquement les colonnes de titres et des genres. ■

Sélection avancée

Cette sélection est de loin la plus intéressante et surtout la plus utile. Il s'agit de sélectionner une partie d'une table en fonction d'une condition.

Exemple 5.2.3 Dans cet exemple, nous allons sélectionner uniquement les films de l'année 2019 en utilisant le code suivant :

```

import pandas as pd

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")

movies_base_selected = movies_base[movies_base['startYear']==2019]

print(movies_base_selected.head(10))

```

Exercice 5.3 En modifiant le programme précédent, sélectionner tous les films dont le genre est Drama ■

Exemple 5.2.4 Maintenant on va faire une recherche avec deux paramètres. On veut les films sortis en 2019 et durant plus de 80 minutes.

```

import pandas as pd

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)

```

```
movies_base_selected = movies_base[ (movies_base['startYear']==2019) & (
                                     movies_base['runtimeMinutes']>80) ]

print(movies_base_selected.head(60))
```

Exercice 5.4 En modifiant le programme précédent, sélectionner les comédies de 2019. ■

Exemple 5.2.5 Dernière étape, on veut maintenant des films de 2009 ou de 2019 durant entre 65 et 100 minutes.

```
import pandas as pd

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)

# 2009 ou 2019 et entre 65 et 100
movies_base_selected = movies_base[
    ((movies_base['startYear'] == 2019) | (movies_base['startYear'] == 2009)) &
    ((movies_base['runtimeMinutes'] > 65) & (movies_base['runtimeMinutes'] <
                                             100)) ]

print(movies_base_selected.head(20))
```

Exercice 5.5 En modifiant le programme précédent, sélectionner tous les films qui ont les attributs suivants :

- Comédie ou drame
- année de parution 2018
- Durée supérieure à 80 minutes

5.3 La jointure

Définition 5.3.1 — Jointure. La jointure consiste à sélectionner les éléments en commun dans deux tables. Cette opération sélectionne les entrées ayant la même clef primaire et les fusionnent.

Exemple 5.3.1 On désire sélectionner les film de 2009 ou 2019 durant entre 65 et 140 minutes ayant obtenu une note supérieure à 9 et un nombre de votes supérieure à 100. Pour effectuer cette sélection nous allons suivre les étapes suivantes :

- Dans la base `movies_base`, nous allons sélectionner les films de 2009 ou 2019 durant entre 65 et 140 minutes.
- Dans la base `movies_rat`, nous allons sélectionner les films ayant obtenus une note supérieure à 9 et un nombre de vote supérieur à 100.
- Puis nous allons faire une jointure entre ces deux sélectionner pour obtenir un ensemble des résultats ayant toutes les propriétés.

```

import pandas as pd

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
movies_rate = pd.read_csv("rate_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)
print(movies_rate.dtypes)

# 2009 ou 2019 et entre 65 et 140
movies_base_selected = movies_base [
    ((movies_base['startYear'] == 2019) | (movies_base['startYear'] == 2009)) &
    ((movies_base['runtimeMinutes'] > 65) & (movies_base['runtimeMinutes'] <
        140)) ]

print(movies_base_selected.head(20))

# note > 9 et nbvote > 100
movies_rate_selected = movies_rate [ (movies_rate['averageRating'] > 9) & (
    movies_rate['numVotes'] > 100)]

print(movies_rate_selected.head(20))

# Jointure entre les deux bases
result = movies_base_selected.join(movies_rate_selected)

result = result.dropna()

print(result['originalTitle'].head(20))

```

Exercice 5.6 Votre travail consiste à donner les titres de plus mauvais film de 2019. Voici la liste des attributs :

- note inférieure à 3
- nombre de votes supérieur à 100
- durée supérieur à 70 minutes

Vous devrez faire une jointure pour obtenir la liste des titres des films. ■

5.4 Regroupement et graphiques

Pour des besoins d'analyse ou de représentation, il est utile de pouvoir regrouper les données suivant un critère. La fonction `groupby` regroupe les données. La fonction `size` renvoie la taille de chacun des groupes.

Exemple 5.4.1 Dans cet exemple nous sélectionnons les comédies dont l'année d'exploitation est supérieure à 2000. Sur cette sélection, nous appliquons la fonction `groupby('startYear')` qui va regrouper les résultats suivant l'année d'exploitation puis enfin la fonction `size` qui nous renverra la taille de chacun des groupes.

```

import pandas as pd

# Importation des donnees

```

```

movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)

#Compilation des donnees pour le graphique
group_data = movies_base[
    (movies_base['genres']=="Comedy") & (movies_base['startYear'] > 2000)
].groupby('startYear').size()

print(group_data)

```

startYear	size
2001	202
2002	199
2003	281
2004	310
2005	347
2006	385
2007	331
2008	436
2009	437
2010	439
2011	485
2012	480
2013	500
2014	494
2015	494
2016	559
2017	562
2018	631
2019	456
2020	62

On a donc 202 comédies sorties en 2001, 199 en 2002 etc...

A partir de ces données, on peut construire des graphiques.

Exemple 5.4.2 Dans cet exemple, nous allons construire un diagramme circulaire.

```

import pandas as pd
import matplotlib.pyplot as plt

#Configuration de la taille de la police
plt.rcParams.update({'font.size': 12, 'figure.figsize': (10, 8)})

#Importation des donnees
movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)

#Compilation des donnees pour le graphique
pie_data = movies_base[
    (movies_base['genres']=="Comedy") & (movies_base['startYear'] > 2000)
].groupby('startYear').size()

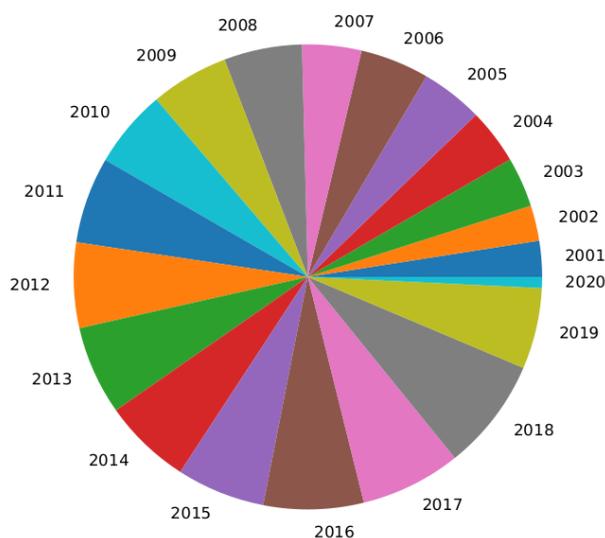
print(pie_data)

canvas = pie_data.plot(kind='pie')

#On enleve le label de l axe des ordonnees
canvas.set_ylabel('')

#Creation du graphique et sauvegarde dans un fichier.
fig = canvas.get_figure()
fig.savefig('graph.pdf')

```



Exemple 5.4.3 Dans cet exemple nous allons construire un coube.

```
import pandas as pd
import matplotlib.pyplot as plt

#Configuration de la taille de la police
plt.rcParams.update({'font.size': 12, 'figure.figsize': (10, 8)})

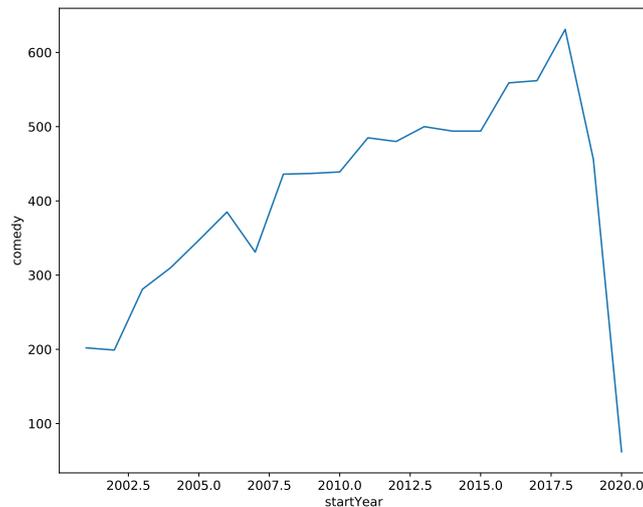
#Importation des donnees
movies_base = pd.read_csv("base_custom.csv", sep=',', index_col="tconst")
print(movies_base.dtypes)
#Compilation des donnees pour le graphique
movies_base_selected = movies_base[
    (movies_base['genres']=="Comedy") & (movies_base['startYear'] > 2000)
].groupby('startYear').size()

print(movies_base_selected)

canvas = movies_base_selected.plot()

#On enleve le label de l'axe des ordonnees
canvas.set_ylabel('comedy')

#Creation du graphique et sauvegarde dans un fichier.
fig = canvas.get_figure()
fig.savefig('graph.pdf')
```



5.5 Tâches finales

Ces deux tâches feront l'objet d'un oral où vous expliquerez vos programmes et les choix effectués.

Travaux Pratiques 5.1 Faire un programme permettant de construire une courbe permettant de visualiser l'évolution au cours du temps du nombre de comédies ayant les attribus suivants :

- Sortie après 1990
- Note moyenne supérieure à 8
- Nombre de votant supérieur à 150
- Durée supérieure à 60 minutes

La procédure de rendu est décrite sur lms.zone

Travaux Pratiques 5.2 A partir du travail précédent construire la fonction à arguments facultatifs :

```
get_graph(dureeMin, dureeMax, noteMin, noteMax, voteMin, voteMax, dateMin, dateMax, genre)
```

permettant de visualier l'évolution du nombre de film ayant les attributs suivants :

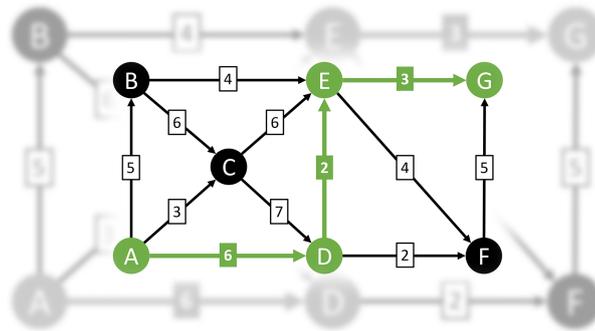
- Durée entre `dureeMin` et `dureeMax`
- Note entre `noteMin` et `noteMax`
- Nombre de votes entre `voteMin` et `voteMax`
- Date de sortie entre `dateMin` et `dateMax`
- film du genre : `genre`

Il est très conseillé de visualier la vidéo d'aide et de rappel sur les fonctions python.

6. Projet Final

6.1 Principe général du projet

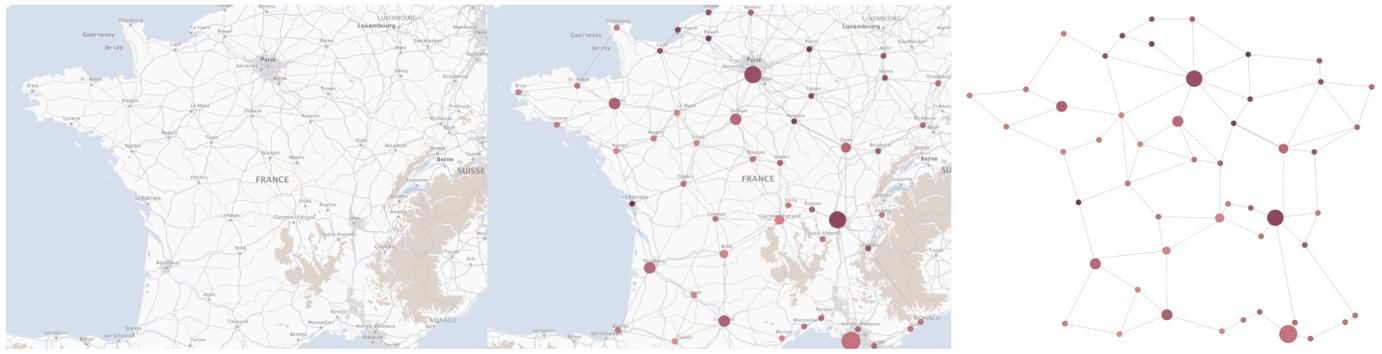
Le projet consiste en l'implémentation de l'algorithme de Dijkstra. Ce dernier permet de déterminer un plus court chemin entre deux noeuds d'un graphique comme l'illustre le schéma ci-dessous.



Ce projet s'appuie donc sur la théorie des graphes.

Définition 6.1.1 — Théorie des graphes. La théorie des graphes est la discipline mathématique et informatique qui étudie les graphes, lesquels sont des modèles abstraits de dessins de réseaux reliant des objets. Ces modèles sont constitués par la donnée de sommets (aussi appelés nœuds ou points, en référence aux polyèdres), et d'arêtes (aussi appelées liens ou lignes) entre ces sommets ; ces arêtes sont parfois non-symétriques (les graphes sont alors dits orientés) et sont appelés des flèches.

Exemple 6.1.1 Voici un exemple d'un graphe. On considère que chaque grande ville de France est un noeud. S'il existe une voie rapide entre deux villes, on trace une arête. Si en plus on pondère chaque arête avec le temps de parcours entre les deux villes qu'elle relie. Alors l'algorithme de Dijkstra donnera l'itinéraire le plus court entre deux villes. Il s'agit en effet de l'algorithme présent dans les GPS.



Ce projet ne se divise pas en tâche finale mais en étape. Ces étapes sont dépendantes, elles doivent

6.2.2 Etape 2 : Object Oriented Programming

Python comme la plupart des langages de programmation possède des structures de données comme par exemple :

- Les entiers (int)
- Les décimaux (float)
- Les chaînes de caractère (string)
- Les tableaux (array)

Mais pas de graphe. Or nous devons avoir une structure adaptée aux données que nous devons traiter. Il va donc falloir la créer. Nous allons donc implémenter les objets suivants :

- Node : Objet décrivant un noeud
- Nodes : Objet décrivant un ensemble de noeuds
- Edge : Objet décrivant une arête
- Edges : Objet décrivant un ensemble d'arêtes
- Graph : Objet décrivant un graphe (ensemble de noeuds et d'arêtes)

Cette étape est la plus guidée de toute. Votre travail consistera à compléter un code préexistant.

Exemple 6.2.1 Afin de démystifier ces objets voici un extrait du code implémentant l'objet Node et ses méthodes (actions d'un objet sur lui-même).

```
class Node :
    label = ""
    id = ""

    def __init__(self):
        self.label = ""
        self.id = ""

    def __init__(self, id, label):
        self.label = label
        self.id = id

    def print(self):
        print("node id: ", self.id, " label: ", self.label)
```

Et maintenant un peu de code utilisant la classe Node :

```
node1 = Node("n1", "A")
node1.print()
```

Que fait ce code ?

6.2.3 Etape 3 : L'algorithme de Dijkstra

Si vous en arrivez là, c'est que vous parsez votre fichier xml puis que vous avez injecté les données extraites dans vos objets (node, edge, ...). Les données sont donc structurées prêtes à être exploitées. Il vous

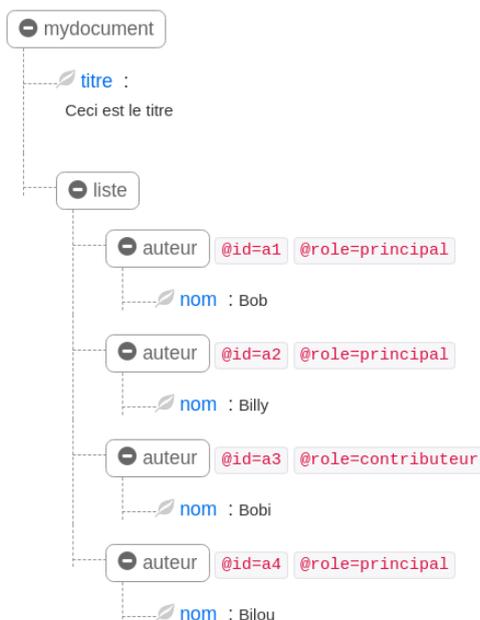
suffit d'appliquer l'algorithme de Dijkstra. Des modifications seront sans doute à effectuer sur vos classes (étape 2). L'algorithme en lui même est décrit dans la section idoine.

6.3 Etape 1 : Le parsing

Avant d'attaquer le parsing en lui même, regardons d'un peu plus pres comment est structuré un fichier xml sur un exemple simple.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <mydocument>
3   <titre>
4     Ceci est le titre
5   </titre>
6   <liste>
7     <auteur id="a1" role="principal">
8       <nom>Bob</nom>
9     </auteur>
10    <auteur id="a2" role="principal">
11      <nom>Billy</nom>
12    </auteur>
13    <auteur id="a3" role="contributeur">
14      <nom>Bobi</nom>
15    </auteur>
16    <auteur id="a4" role="principal">
17      <nom>Bilou</nom>
18    </auteur>
19  </liste>
20 </mydocument>
```

La première ligne est l'entête du fichier et précise son format et son encodage. Ceci doit vous rappeler le travail effectué sur les fichiers image. La suite du fichier est un enchainement de balises décrivant un arborescence (structure en forme d'arbre) en voici une illustration :



La librairie `xmltodict` permet d'aller chercher/trier/sélectionner les informations désirées dans cette arborescence.

```

1 import xmltodict
2
3 #Ouverture et lecture du fichier xml
4 #xmldict crée l'arborescence et l'enregistre dans la
   variable doc
5 with open('sample.xml') as fd:
6     doc = xmltodict.parse(fd.read())
7
8
9 #Selection du titre du document
10 #Titre étant une feuille unique la section est direct
11 titre = doc['mydocument']['titre']
12
13 print(titre)
14
15 #Impression d'un séparateur pour la présentation
16 print("\n\n ----- \n\n")
17
18 #Les balises de type auteur sont multiples. Il faut
19 donc parcourir chacun de ces feuilles pour extraire le
20 nom de chaque auteur (boucle for).
21 #Afin de capter l'attribut d'une balise, on l'appelle
22 en utilisant le @
23
24 for elt in doc['mydocument']['liste']['auteur'] :
25     print(elt['@id'])
26     print(elt['nom'])
27
28 #Impression d'un séparateur pour la présentation
29 print("\n\n ----- \n\n")
30
31 #On peut aussi faire des tests afin de faire une
32 sélection plus fin des données à exploiter. Si par
33 exemple on désire uniquement les auteurs principaux,
34 on fera
35
36 for elt in doc['mydocument']['liste']['auteur'] :
37     if(elt['@role']=="principal"):
38         print(elt['@id'])
39         print(elt['nom'])

```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <mydocument>
3   <titre>
4     Ceci est le titre
5   </titre>
6   <liste>
7     <auteur id="a1" role="principal">
8       <nom>Bob</nom>
9     </auteur>
10    <auteur id="a2" role="principal">
11      <nom>Billy</nom>
12    </auteur>
13    <auteur id="a3" role="contributeur">
14      <nom>Bobi</nom>
15    </auteur>
16    <auteur id="a4" role="principal">
17      <nom>Bilou</nom>
18    </auteur>
19  </liste>
20 </mydocument>

```

```

In [17]: runfile('/home/cid/Drive/Cours/
Parsing/main.py', wdir='/home/cid/Drive/
Cours/Parsing')
Ceci est le titre

-----

a1
Bob
a2
Billy
a3
Bobi
a4
Bilou

-----

a1
Bob
a2
Billy
a4
Bilou

In [18]: |

```

Une vidéo de présentation de l'exemple ci-dessus ainsi qu'une autre sur les spécificités du format graphml sont disponibles sur lms.zone.

Travaux Pratiques 6.1 Votre consiste consiste en l'élaboration d'un programme permettant de parser un fichier graphML en sélectionnant les éléments suivants :

- id des nodes
- label des nodes
- id des edges
- source de target de chaque edge
- label de chaque edge

Pour le moment, ces données n'étant pas exploitables, vous devez simplement les afficher à l'écran en attendant la suite.

```

node - id : node0 label : 0
node - id : node1 label : 1
node - id : node2 label : 2
node - id : node3 label : 3
node - id : node4 label : 4
edge - id edge0-1 source : node0 target : node1 label : 3
edge - id edge0-2 source : node0 target : node2 label : 15
edge - id edge0-4 source : node0 target : node4 label : 19
edge - id edge1-0 source : node1 target : node0 label : 1
edge - id edge1-3 source : node1 target : node3 label : 22
edge - id edge3-0 source : node3 target : node0 label : 3
edge - id edge3-1 source : node3 target : node1 label : 75
edge - id edge3-2 source : node3 target : node2 label : 70
edge - id edge4-0 source : node4 target : node0 label : 12
edge - id edge4-1 source : node4 target : node1 label : 10

```