

Table des matières

1	Information	1
1.1	Personnel enseignant	1
1.2	Disponibilités	1
1.3	Contenu des démonstrations	1
1.4	Travaux de session et devoirs	2
1.5	Remise des travaux	2
1.6	Intégrité	3
	Plagiat et fraude	3
	Intelligence artificielle générative	3
2	Apprendre à vous connaître	5
	Profil du groupe	5
	Motivations et carrière	5
3	Création de votre compte DIRO	8
3.1	Procédure d’activation du compte	8
4	Connexion au DIRO	9
4.1	Introduction : la ligne de commande	9
4.2	Windows	9
4.3	macOS et Linux	9
5	Introduction aux commandes UNIX	12
	Navigation et inspection	12
	Fichiers et répertoires	12
	Lecture et édition rapide	13
	Recherche	14
	Permissions	14
	Archives et compression	14
	Réseau et transferts	15
	Système et aide	15
6	Introduction à L^AT_EX	19
	Principe de balisage	19
7	Ressources utiles pour comprendre Linux et L^AT_EX	21
	Ressources Linux	21
	Ressources L ^A T _E X	21

1 Information

Département d'informatique et de recherche opérationnelle (DIRO)
Session d'automne 2025

IFT1005 (3 crédits)
DESIGN ET DÉVELOPPEMENT WEB

Dernière mise à jour le 17 septembre 2025

1.1 Personnel enseignant

Abdelhakim Hafid

Chargé de cours

abdelhakim.hafid@umontreal.ca

Franz Girardin

Auxiliaire démonstrateur

franz.girardin@umontreal.ca

Martin Medina Mejia

Auxiliaire correcteur

martin.medina@umontreal.ca

1.2 Disponibilités



MARTIN

| Jeudi 13h30 - 14h30

Bibliothèque de Lettres et Sciences Humaines, local 5131



FRANZ

| Mercredi 17h30 - 18h20

Bibliothèque de Mathématiques et Informatique, à droite de l'entrée

1.3 Contenu des démonstrations

1. Outils de développement.

Compte DIRO, éditeur de texte ([Neovim](#), VS Code), Linux, \LaTeX , etc.

2. Architecture de l'internet.

Protocoles, réseau, [modèle TCP/IP](#).

3. Syntaxe du balisage HTML.

Structure de base, éléments textuels, entité et caractères, *sémantique*.

4. Syntaxe CSS.

[Sélecteurs CSS](#), hiérarchie, organisation du code, etc.

5. Manipulation des éléments graphiques.

Positionnement et présentation des images, utilisation de fichiers auxiliaires [SVG](#), etc.

6. Mise en page.

Organisation des éléments textuels et visuels, [typographie](#), *responsive design*.

7. Introduction au scriptage.

Concept de *Document Object Model* ([DOM](#)), JavaScript, JQuery.

1.4 Travaux de session et devoirs

DEVOIRS

- ◆ Environ 4 - 6 devoirs qui vaudront pour 10% de la note finale.
- ◆ Les devoirs sont à effectuer en équipes de deux personnes *obligatoirement*.
- ◆ La présentation de l'énoncé est faite par Franz, la correction par Martin.

TRAVAUX DE SESSION

- ◆ Il y aura 2 travaux de session qui vaudront pour 15 % de la note *chacun*
- ◆ Les devoirs sont à effectuer en équipes de deux personnes *obligatoirement*.

†La quantité de devoirs (4 à 6) pourrait être affectée par la progression des cours théoriques tout au long de la session. Pour chaque devoirs Martin et moi communiquons pour nous assurer que l'énoncé qui vous sera présenté est congruant vis-à-vis les critères de correction. Si vous avez des questions quant à la correction, n'hésitez pas à [me les communiquer](#). Si non, vous pouvez toujours entrer en contact directement avec [Martin](#)

1.5 Remise des travaux

- ◆ Les travaux doivent être remis sur [Studium](#) *obligatoirement*. Les travaux remis par courriel ne seront pas corrigés.
- ◆ Un seul membre de l'équipe remet le devoir, donc il faut indiquer le nom et le matricule des deux étudiants.
- ◆ Il y aura une **pénalité de 18% par jour de retard**. Un jour de retard correspond à une tranche de 24 h après la date de remise officielle.

1.6 Intégrité

PLAGIAT ET FRAUDE

Les manquements à l'intégrité peuvent prendre plusieurs formes. La tricherie durant une épreuve notée, la soumission d'un travail réalisé dans un semestre antérieure ou remis pour plusieurs cours de la session courante, l'invention de faits et sources, les manquements au référencement et à l'emprunt sont des pratiques prohibées. Les sanctions peuvent aller jusqu'à l'exclusion de l'université. Consulter le [site Intrégrité](#) pour plus d'information

INTELLIGENCE ARTIFICIELLE GÉNÉRATIVE

Dans ce cours, les outils d'IA générative (IAG) peuvent être employés en soutien aux activités d'apprentissage des personnes participantes, par exemple :

- ◆ pour aider à comprendre des concepts du cours ;
- ◆ pour générer des questions afin d'étudier pour les examens ;
- ◆ pour analyser et reformuler les énoncés des devoirs ;
- ◆ pour effectuer une révision linguistique des textes que vous écrivez.

Il est important de réaliser que l'arrivée de l'IA générative a profondément bousculé les habitudes du monde académique et professionnel. Étudiants, professeurs, facultés et employeurs se trouvent tous confrontés à des situations où l'usage de l'IA peut sembler être une solution rapide : pour gagner du temps, pour compenser un manque de ressources ou pour éviter un effort jugé inutile. Mais céder à cette tentation sans recul peut aussi révéler des failles : est-ce un simple raccourci acceptable, une réaction de déni (*cope*), un manque de compétence (*skill issue*) ou un comportement inapproprié ?

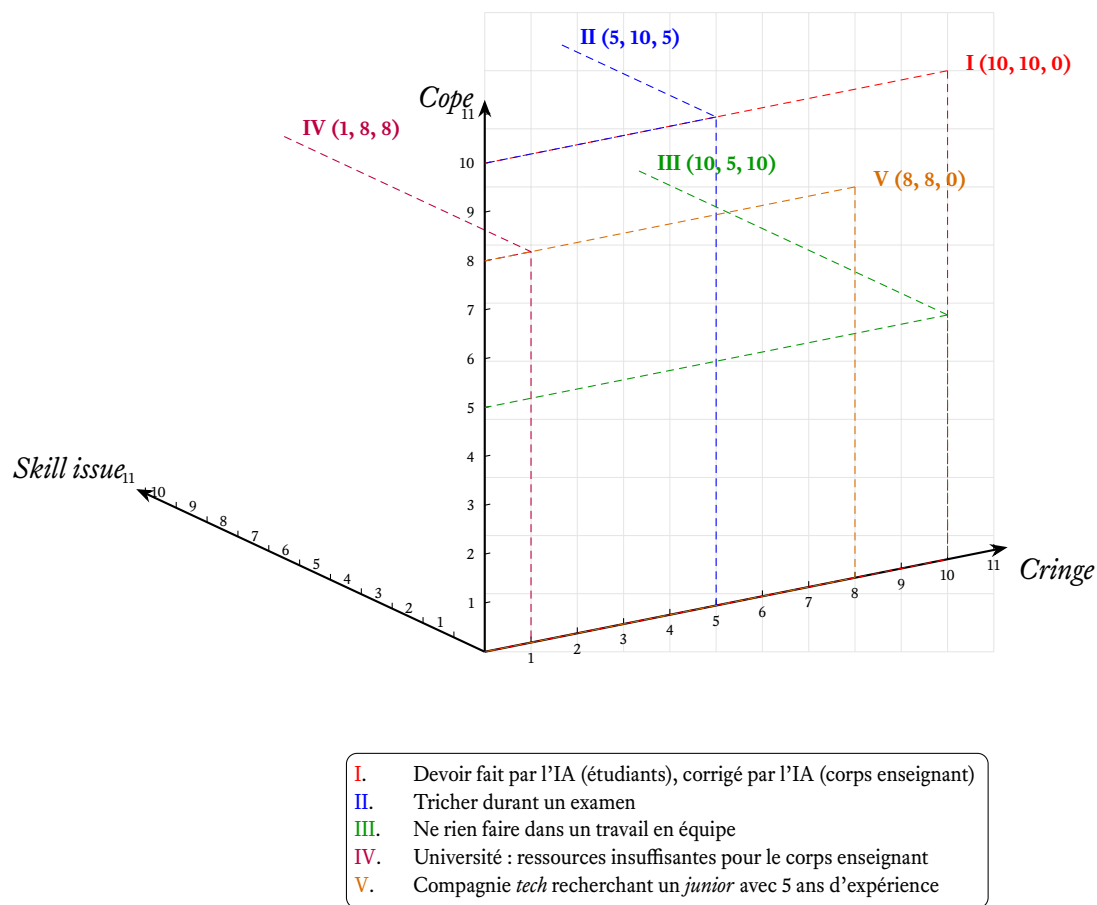
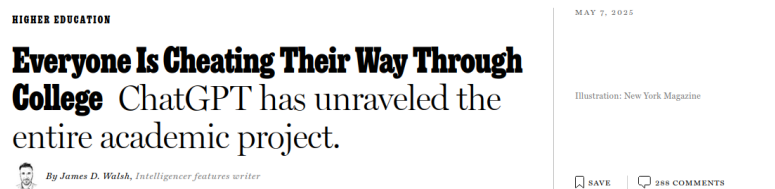


Figure 1.1 – Repère « *Cringe / Cope / Skill issue* » illustrant certains défis académiques et professionnels à l'aire de l'IA générative

Tous ne trichent pas par paresse. La pression de performance, les attentes familiales et l'effet « tout le monde le fait » pèsent. Côté institution, le manque de ressources et de support offert par l'établissement invitent également certaines personnes du corps enseignant à être tenté par l'idée de prendre des raccourcis. Sur le marché du travail, l'omniprésence de l'IA générative contribue à ériger les critères de sélection et mène à l'avenue d'un marché hyper compétitif où les candidats sont eux-mêmes tenté par la voie de la simplicité. Malgré tout, l'IA générative ne devrait pas supplanter l'esprit : qu'elle demeure un outil, un allié de la réflexion, et non son substitut.



Référence : [New York Magazine](#)

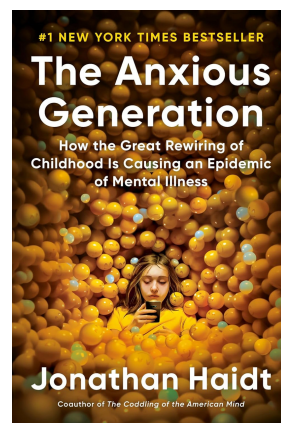
2 Apprendre à vous connaître

PROFIL DU GROUPE

Pour que nous ayons une meilleure idée du niveau d'expérience et du profil de chaque individu dans le groupe, nous aimerions vous poser quelques questions :

- ◆ Qui sont les étudiants internationaux ou étudiants d'un autre système que le système québécois ?
- ◆ Qui sont les étudiants qui ont fait un programme d'informatique au Cégep ?
- ◆ Pourquoi avez-vous choisi l'informatique (👨‍💻, 🧑‍💻, 🚀, 📁, 🏆, 😊) ?
- ◆ Quel âge avez-vous ?
- ◆ Avez-vous déjà utilisé Linux ?

MOTIVATIONS ET CHOIX DE CARRIÈRE



Percent U.S. Anxiety Prevalence

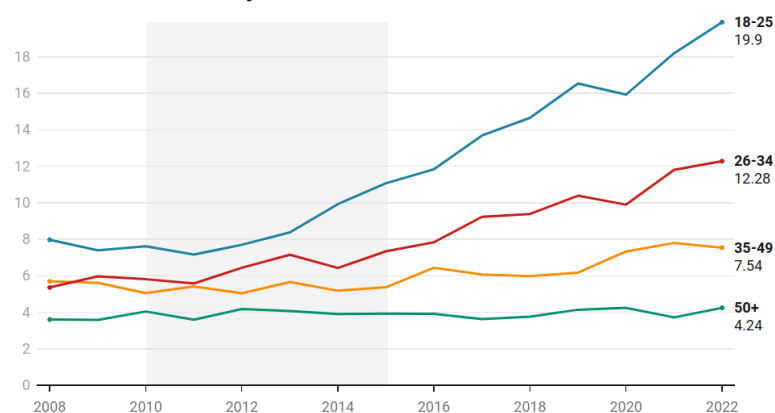


Figure 2.1 – Prévalence de l'anxiété aux États-Unis selon la tranche d'âge (2008–2022). Source : [Google Sheet \(anxiété\)](#).

Les données montrent une explosion de l'anxiété chez les jeunes, avec une augmentation de 139 % chez les 18-25 ans et de 103 % chez les 26-34 ans depuis 2010.

U.S. Emergency Department Visits for Self-Harm (Ages 10-14)

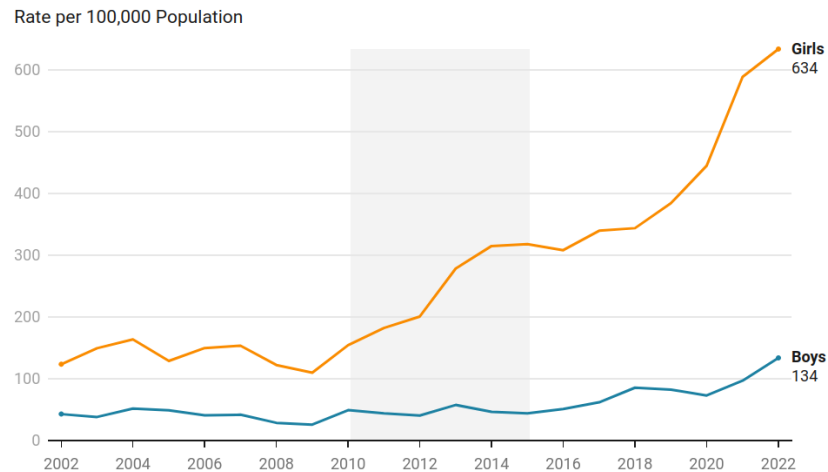


Figure 2.2 – Visites aux urgences pour automutilation (10–14), filles et garçons. Source : [CDC WISQARS](#).

Pour les visites aux urgences liées à l'automutilation, Haidt cite une augmentation de 188 % chez les filles contre 48 % chez les garçons sur la même période. Bien que ce ne soit pas présenté en graphe ici, ce contraste illustre une divergence aiguë dans l'impact psychologique selon le genre.

U.S. Teen Suicide Rate (Ages 15-19)

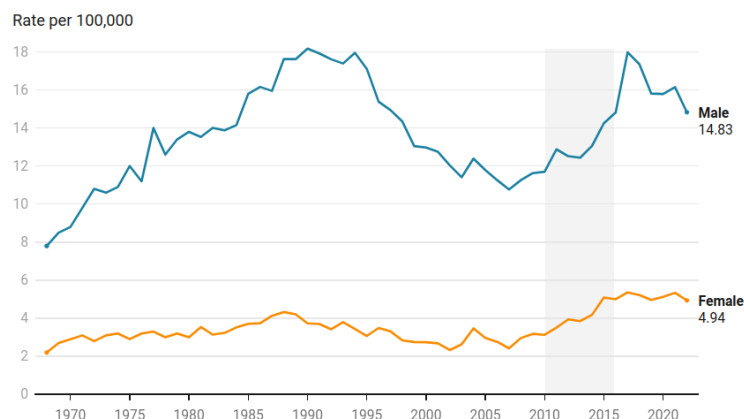


Figure 2.3 – Taux de suicide chez les adolescents (15–19), par sexe. Source : [Google Sheet \(données CDC\)](#).

En définitive, quelle que soit la motivation initiale—innovation, passion, pragmatisme—la plupart des diplômés en informatique contribueront à des applications, des plateformes ou des appareils liés au divertissement ou dont la consommation sera massive. Dans notre ère technocritique (celle où la

technologie et la culture ne peuvent plus être dissociées), cela implique que leur travail alimentera bien qu'*indirectement* une crise néanmoins réelle de santé mentale chez des centaines de milliers de jeunes. **Si ce constat les préoccupe**, il devrait éclairer chaque choix de carrière.

3 Création de votre compte DIRO

Le DIRO offre à ceux qui en sont membre un accès aux ordinateurs du département. Vous pouvez en bénéficier *en personne*, en utilisant, entre autres les laboratoires informatiques ou les ordinateurs de la [bibliothèque de mathématiques et informatique](#) du pavillon André-Aisenstadt. Vous pouvez également établir une *connexion à distance* sur une machine du département en utilisant votre compte du DIRO. Posséder un compte présente de nombreux avantages :

- ◆ Accès aux serveurs DIRO et aux laboratoires par connexion SSH.
- ◆ Espace de stockage personnel avec sauvegardes.
- ◆ Logiciels spécialisés et environnements Linux configurés.
- ◆ Impression et ressources matérielles du département.

3.1 Procédure d'activation du compte

- 1 Accédez à l'adresse : <https://www.iro.umontreal.ca/cgi-bin/motdepasse/motdepasse.cgi>
- 2 Entrez le *code d'accès* pour la vérification.
- 3 Confirmez votre identité en effectuant une authentification auprès du portail de l'UdeM.
- 4 Gardez note de votre *username* :

nom de famille 1^{ère} lettre du prénom

Remarque 3.1 (Mot de passe). Une page vous indiquant votre *mot de passe temporaire* associé au *username* DIRO apparaîtra. Il est important noter vos identifiants et de vous souvenir de l'endroit où vous les aurez enregistré. ┘

- 6 Retranscrivez votre nom d'utilisateur et *mot de passe temporaire* qui vous est présenté ok.

Remarque 3.2 (Erreur d'entrée). Si la page s'actualise, c'est que vous avez fait une erreur de retranscription, réessayez. ┘

- 7 Un message indiquant que le mot de passe a été changé s'affichera à l'écran lorsque vous aurez réussi. Vous êtes maintenant prêt à utiliser les ordinateurs Linux du DIRO.

4 Connexion au DIRO

4.1 Introduction : la ligne de commande

Historiquement, les premiers ordinateurs n’avaient pas d’écran graphique. L’interaction se faisait uniquement en ligne de commande. Plus tard, même avec l’apparition d’écrans, les interfaces graphiques n’existaient pas encore : tout passait par du texte. Aujourd’hui, quand vous cliquez sur une icône ou un bouton, une commande est exécutée en arrière-plan sans que vous le sachiez.

C’est pourquoi les commandes essentielles sont souvent très courtes : elles étaient conçues pour être tapées rapidement et fréquemment.

Se connecter au DIRO

Chaque étudiant possède un identifiant DIRO. La structure générale est :

┌────────┐ ┌────────┐
nom de famille 1^{ere} lettre du prénom

Ainsi, un étudiant nommé Durand Pierre aura l’identifiant durandp. Ce dernier est associé à un mot de passe fourni par le DIRO.

Pour vous connecter à la machine arcade, tapez :

```
ssh durandp@arcade.iro.umontreal.ca
```

4.2 Windows

Sous Windows, l’option la plus simple est [Windows Terminal](#). Une fois installé, ouvrez un terminal et utilisez la commande ssh comme ci-dessus.

4.3 macOS et Linux

macOS étant basé sur Unix, il inclut déjà une application [Terminal](#). Ouvrez-la et utilisez directement les commandes Linux présentées dans ce tutoriel. Vous pouvez utiliser **Command + Espace** et taper **Terminal**.

Explorer d’autres machines

Après vous être connecté à arcade, vous pouvez « sauter » vers une autre machine, par exemple :

```
ssh blm15
```

Voir qui est connecté

Pour voir qui d'autre est présent sur la machine :

```
w
```

Accéder à votre espace personnel

Votre répertoire personnel se trouve dans `/home/www/users/username`. Il est aussi possible d'y accéder via `cd /home/www-ens/users/username` Exemple :

```
cd /home/www/users/durandp
```

- Vérifiez qu'il existe un dossier `public_html` dans ce répertoire avec la commande `ls`.
- Si vous n'y voyez pas le dossier `public_html`, suivez les [instructions fournies par le DIRO](#) et tentez de vous déplacer vers le dossier suivant :

```
cd /www/ens/<hiver|ete|aut>/<nomUsager>/public_html
```

Transférer des fichiers

Pour transférer un fichier de votre ordinateur vers le DIRO :

```
scp monfichier.txt durandp@arcade.iro.umontreal.ca:~/monfichier.txt
```

Pour rapatrier un fichier du DIRO vers votre machine :

```
scp durandp@arcade.iro.umontreal.ca:~/monfichier.txt monfichier.txt
```

Accéder à vos pages Web et les consulter

Votre répertoire de page Web se trouve dans `/home/www/users/username/public_html`. Ce répertoire peut contenir des pages Web auxquelles vous pourrez avoir accès depuis votre navigateur, puisqu'elles sont hébergées par les serveurs du DIRO. Dans cet espace, vous trouverez un fichier HTML créé pour vous par défaut portant le nom `index.html`. Vous pouvez le modifier avec un éditeur de texte terminal (`vi`, `vim`) :

```
vi index.html
```

Vous pouvez consulter cette même page dans votre navigateur :

```
www-ens.iro.umontreal.ca/hiver/~girardif/index.html
```

ou simplement :

```
www-ens.iro.umontreal.ca/hiver/~girardif/
```

Conclusion

Comprendre la ligne de commande est fondamental : c'est un héritage historique de l'informatique, et c'est un outil puissant. Au DIRO, la maîtrise de ces commandes simples (`ssh`, `w`, `scp`) vous permettra de travailler efficacement et d'accéder à l'ensemble des ressources offertes.

5 Introduction aux commandes UNIX



NAVIGATION ET INSPECTION

pwd — Où suis-je ?

Affiche le répertoire courant.

```
pwd
# /home/etudiant
```

ls — Lister des fichiers

Liste le contenu d'un répertoire.

```
ls
ls -lah          # détails + tailles lisibles
ls -lah /var/log # lister un autre dossier
```

cd — Changer de dossier

Se déplacer dans l'arborescence.

```
cd cours/ift2035
cd ..           # remonter d'un niveau
cd ~            # aller dans le $HOME
```

tree — Arborescence (si installé)

Affiche l'arborescence (utile pour visualiser la structure).

```
tree -L 2      # profondeur 2
```

FICHIERS ET RÉPERTOIRES

mkdir — Créer un dossier

Crée un nouveau répertoire (avec parents si besoin).

```
mkdir data
mkdir -p projets/ift2035/tp1
```

touch — Créer fichier vide / mettre à jour date

Crée un fichier vide ou met à jour l'horodatage.

```
touch notes.txt
```

cp — Copier

Copie des fichiers ou dossiers.

```
cp a.txt b.txt
cp -r src/ backup/src/  # copie récursive d'un dossier
```

mv — Déplacer / renommer

Déplace ou renomme un fichier/dossier.

```
mv rapport.md doc/rapport.md
mv oldname.txt newname.txt
```

rm — Supprimer

Supprime des fichiers (prudence!). -r pour dossiers, -i pour demander confirmation.

```
rm old.tmp
rm -ri build/  # récursif + interactif
```

LECTURE ET ÉDITION RAPIDE

cat, less — Lire un fichier

Afficher le contenu (less permet de scroller / rechercher).

```
cat README.md
less long-rapport.txt  # q pour quitter, /mot pour chercher
```

head, tail — Début / fin de fichier

Voir les premières/dernières lignes (suivi live avec tail -f).

```
head -n 20 data.log
tail -n 50 data.log
tail -f server.log  # suivre en temps réel
```

nano, vim — Éditer vite

Éditeurs en ligne de commande (nano = simple, vim = puissant).

```
nano notes.txt
vim main.py
```

RECHERCHE

grep — Rechercher du texte

Filtre des lignes par motif (regex), utile avec des pipes.

```
grep -n "TODO" src/*.py
dmesg | grep -i error
```

find — Trouver des fichiers

Recherche par nom, type, taille, date... dans l'arborescence.

```
find . -name "*.txt"
find ~/projets -type f -size +10M
```

PERMISSIONS

chmod — Droits d'accès

Change les permissions (lecture/écriture/exécution).

```
chmod u+x script.sh      # ajoute exécution pour l'utilisateur
chmod 640 secrets.txt    # rw- r-- --- (u g o)
```

ARCHIVES ET COMPRESSION

tar, gzip — Archiver / compresser

Créer et extraire des archives.

```
tar -czf projet.tar.gz projet/  # créer archive compressée
tar -xzf projet.tar.gz          # extraire
```

> — Redirection de sortie

Envoie la sortie standard d'une commande vers un fichier. Un seul chevron > écrase le fichier cible ; deux chevrons >> ajoutent à la fin.

```
echo "Bonjour" > salut.txt      # crée/écrase salut.txt
head -n 5 notes.log >> recap.txt # ajoute 5 lignes à recap.txt
```

| — Pipeline (pipe)

Transfère la sortie d'une commande vers l'entrée d'une autre, permettant de chaîner des traitements.

```
ls -lah | grep ".txt"           # filtre uniquement les .txt
dmesg | less                    # naviguer dans la sortie du noyau
cat data.csv | wc -l            # compter le nombre de lignes
```

RÉSEAU ET TRANSFERTS

ssh — Connexion distante

Ouvrir une session sur un hôte distant (DIRO, par ex.).

```
ssh utilisateur@arcade.iro.umontreal.ca
```

scp — Copier via SSH

Transférer des fichiers entre ta machine et un serveur (DIRO).

```
# vers le serveur
```

```
scp rapport.pdf user@arcade.iro.umontreal.ca:~/rapport.pdf
```

```
# depuis le serveur
```

```
scp user@arcade.iro.umontreal.ca:~/data.csv ./data.csv
```

wget, curl — Télécharger

Récupérer des fichiers en HTTP(S).

```
wget https://exemple.org/donnees.csv
```

```
curl -L -o donnees.csv https://exemple.org/donnees.csv
```

SYSTÈME ET AIDE

whoami, w — Identité et présence

Qui suis-je? Qui est connecté?

```
whoami
```

```
w
```

which, echo, env

Localiser un binaire, afficher du texte, variables d'environnement.

```
which python
```

```
echo "Bonjour"
```

```
env | grep HOME
```

man — Manuel d'une commande

Consulter l'aide officielle (niveaux man 1, man 5, etc.).

```
man grep
```

```
man 5 crontab # section 5 (fichiers de config)
```

Astuce : en cas de doute, lance commande `--help`.

Exercice — Redirection et pipeline simples

Objectif : enchaîner **10 commandes de base** et rediriger toutes leurs sorties dans un seul fichier `resultat.txt`. Utilise uniquement les commandes vues dans ce document (`pwd`, `ls`, `touch`, `echo`, `cat`, `head`, `tail`, `tree`, `wc`, `whoami`). Chaque nouvelle sortie doit être **ajoutée** (`>>`) sauf la toute première (`>`).

```
# Préparer un mini environnement de test
mkdir demo
cd demo
touch a.txt b.txt c.txt
echo "Ligne1" > a.txt
echo "Ligne2" >> a.txt
echo -e "un\ndeux\ntrois\nquatre" > b.txt
echo "fichier C" > c.txt

# 1) Répertoire courant
pwd > resultat.txt

# 2) Lister les fichiers
ls >> resultat.txt

# 3) Voir l'arborescence (niveau 1)
tree -L 1 >> resultat.txt

# 4) Afficher le contenu de a.txt
cat a.txt >> resultat.txt

# 5) Afficher seulement les 2 premières lignes de b.txt
head -n 2 b.txt >> resultat.txt

# 6) Afficher seulement la dernière ligne de b.txt
tail -n 1 b.txt >> resultat.txt

# 7) Compter les lignes de b.txt
wc -l b.txt >> resultat.txt

# 8) Ajouter le nom de l'utilisateur
whoami >> resultat.txt

# 9) Ajouter une ligne personnalisée
echo "Fin de l'exercice" >> resultat.txt

# 10) Relire le fichier final pour vérifier
cat resultat.txt >> resultat.txt
```

Contenu attendu de resultat.txt (exemple)

```
/chemin/vers/demo
```

```
a.txt
```

```
b.txt
```

```
c.txt
```

```
.
```

```
├─ a.txt
```

```
├─ b.txt
```

```
└─ c.txt
```

```
Ligne1
```

```
Ligne2
```

```
un
```

```
deux
```

```
quatre
```

```
4 b.txt
```

```
etudiant
```

```
Fin de l'exercice
```

```
/chemin/vers/demo
```

```
a.txt
```

```
b.txt
```

```
c.txt
```

```
.
```

```
├─ a.txt
```

```
├─ b.txt
```

```
└─ c.txt
```

```
Ligne1
```

```
Ligne2
```

```
un
```

```
deux
```

```
quatre
```

```
4 b.txt
```

```
etudiant
```

```
Fin de l'exercice
```

Introduction à L^AT_EX

L^AT_EX est un système de composition de documents conçu pour produire des textes de haute qualité typographique. Très utilisé dans les milieux universitaires et scientifiques, il permet de séparer clairement le contenu de la forme et facilite l'intégration des mathématiques, des références et des structures complexes. Son développement s'inscrit dans une longue histoire issue du travail de Donald Knuth (T_EX) et de Leslie Lamport (L^AT_EX). Pour en comprendre le contexte historique et l'évolution, veuillez consulter le document complet disponible sur [Studium](#).

PRINCIPE DE BALISAGE

L^AT_EX et HTML (le langage que nous étudierons plus tard dans la session) ont pour point commun d'être des **langages de balisage**. Bien que leur syntaxe diffère, on y retrouve des similitudes de principes et de paradigmes : on structure le contenu avec des balises, on sépare la forme du fond et on compose des documents lisibles, réutilisables et cohérents.

Voici un exemple minimal de document L^AT_EX :

```
\documentclass{article}
\begin{document}
\title{Exemple simple}
\author{Étudiant·e}
\date{}
\maketitle

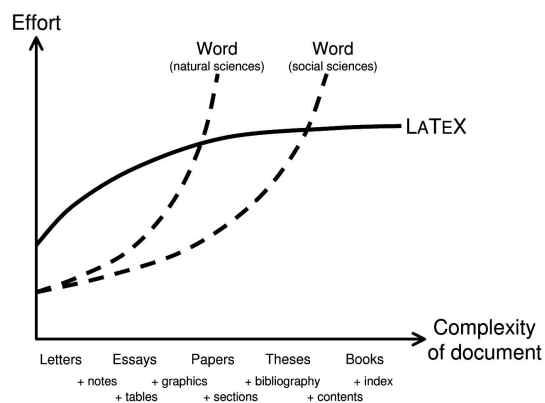
\section{Introduction}
Ceci est un document en \LaTeX{}.
On peut mettre un mot en \emph{italique}.
\end{document}
```

Voici son équivalent en HTML :

```

<!DOCTYPE html>
<html lang="fr">
  <head><meta charset="utf-8">
  <title>Exemple simple</title></head>
  <body>
    <h1>Exemple simple</h1>
    <h2>Introduction</h2>
    <p>Ceci est un document en HTML.
      On peut mettre un mot en <em>italique</em>.
    </p>
  </body>
</html>

```



7 Ressources utiles pour comprendre Linux et L^AT_EX

RESSOURCES LINUX

- I. Linux en moins de 3 minutes**
Vidéo YouTube concise de *Fireship IO*
- Explication par *Mental Outlaw*
- II. GNU vs Linux / GNU+Linux**
- III. Documentaire GNU et Linux**
Documentaire complet (Adam N.)
- Courte parodie humoristique sur RMS
- IV. Caricature de Stallman**

RESSOURCES L^AT_EX

- I. Histoire de L^AT_EX**
Vidéo d'Eitan Lees (histoire et contexte)
- Intro L^AT_EX (documents mathématiques, série Bazett)
- II. Introduction à L^AT_EX**
- III. Vim + L^AT_EX en IDE**
Configurer Vim + L^AT_EX pour une expérience complète
- Caricature des utilisateurs de L^AT_EX
- IV. Caricature L^AT_EX**