Camp de jour Folie Technique   
Été 2019

**INFORMATIQUE**Thématique 9-17 ans

POLYTECHNIQUE MONTRÉAL  
Par  Charly Jeffrey et Bénédicte Samuel-Lafleur

**Table des matières**

[**Introduction**](#_30j0zll) **2**

[**Horaire de la semaine**](#_1fob9te) **2**

[**Conseils**](#_3znysh7) **2**

[**“Nom de l’activité”**](#_eh9ubcvm6ayx) **3**

…..

# **Introduction**

# **Horaire de la semaine**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lundi | Mardi | Mercredi | Jeudi | Vendredi |
| 9 :00 | Spectacle du lundi | Cryptographie | Intelligence artificielle | Activité santé et sécurité | Projet individuel ou défis python |
| 9 :30 |
| 10 :00 | Introduction à l’informatique |
| 10 :30 |
| 11 :00 | Diner | | | | |
| 11 :30 |
| 12 :00 | Activité à l’extérieur | | | | |
| 12 :30 |
| 13 :00 | Introduction à python | Tic-tac-toe | Langage de programmation + défis python et/ou tic-tac-toe | Master-Mind | Kermesse |
| 13 :30 |
| 14 :00 |
| 14 :30 |
| 15 :00 |
| 15 :30 |
| 16 :00 |  |  |  |  |  |

# **Conseils**

Les jeunes auront chacun une clé usb. Il pourrait être pertinent de renommer la clé en début de semaine ou d’écrire leur nom dessus puisque la clé doit rester à polytechnique toute la semaine, mais ils peuvent partir avec à la fin.

# Introduction de la semaine

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  1 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  **1** |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  **-** |  | **Notions abordées**  **Introduction** |

**Théorie**

Pour l’histoire, les ordinateurs ,dans leur utilité de machine à calculer, existent depuis le début du siècle dernier. Cependant, les premiers algorithmes et programmes sont apparus bien avant vers la moitié du 19e siècle. On considère souvent Turing comme étant l’inventeur du premier ordinateur, ce qui est partiellement vrai. L’utilisation des transistors dans les ordinateurs ont permis de réduire drastiquement la taille de ceux-ci et de les rendre plus accessibles.

Pour le fonctionnement de l’ordinateur, en gros, il reçoit des informations, les traite grâce à son processeur puis retourne la sortie ou change son comportement de façon interne. Les entrées peuvent être données par l’utilisateur sous forme d’un clic de souris, de touche sur le clavier ou encore d’action sur une manette. De même, l’ordinateur peut aussi se donner ses propres entrées. Les données sont transférées dans le processeur qui les traitera et qui utilisera dans certain cas de la mémoire. Le disque contient toute l’information du système, mais c’est très lent, on préféra donc écrire la mémoire nécessaire sur de la cache qui sera lu plus rapidement par le processeur. À noter que ce processeur peut aussi être sur une carte graphique, ce qui maximise les performances de l’ordinateur. Finalement, les sorties peuvent prendre différentes formes : l’apparition de quelque chose à l’écran, un son produit ou une écriture de mémoire par exemple.

Le ventilateur sert à refroidir le système, le CPU est le processeur du système, c’est lui qui fait la majorité de calcul. La carte mère sert à faire les ponts entre toutes les composantes. La mémoire vive permet d’accéder rapidement à certains éléments de mémoires, car l’accès au disque dur se fait lentement, donc le processeur tend à mettre les informations dont il a besoin dans peu de temps ou fréquemment dans la mémoire vive. Le disque dur autrefois était soir forme de disque, maintenant on voit apparaitre de plus en plus les SSD, qui sont sous forme de puces électroniques. Elles sont donc plus rapides, mais durent en général moins longtemps. La carte graphique est très utile pour les jeux vidéo. Elle contient aussi des processeurs afin que les images restent fluides.

La communication avec un ordinateur se fait sous forme de code. Dans la semaine, nous allons utiliser un langage de haut niveau, python. À la base, l’ordinateur se parle à lui-même en 0 et en 1. Ensuite, si on doit communiquer à la structure interne, on utilise le langage assembleur. Puis on peut monter de niveau pour faciliter la compréhension.

La sécurité informatique est très importante. Plus tard dans la semaine il sera question de comment bien protéger ses données. Il sera aussi question de la cryptographie qui est la science de cacher des codes.

Finalement, l’intelligence artificielle étant le sujet de l’heure, cette notion sera aussi abordé plus tard dans la semaine.

**Matériel**

aucun

**Manipulation**

Ouvrir la présentation sur l’ordinateur

**Présentation suggérée**

Poser des questions aux campeurs pour savoir ce qu’ils connaissent déjà

**Sources d’erreur**

-

# Introduction à python

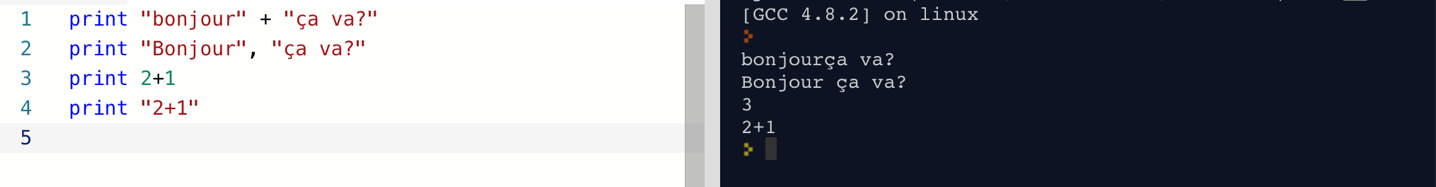
**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  3 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  **3** |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  La programmation python |

**Théorie**

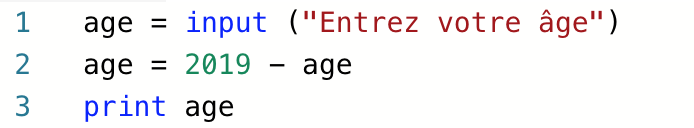
Python est un langage de haut niveau, très utile pour programmer efficacement. En effet, il est facile à comprendre et à déverminer.

Il faudra commencer avec la fonction print. Cette fonction sert à imprimer des choses dans la console. Pour bien comprendre comment fonctionne print, faites des tests avec les campeurs, comme l’image suivante



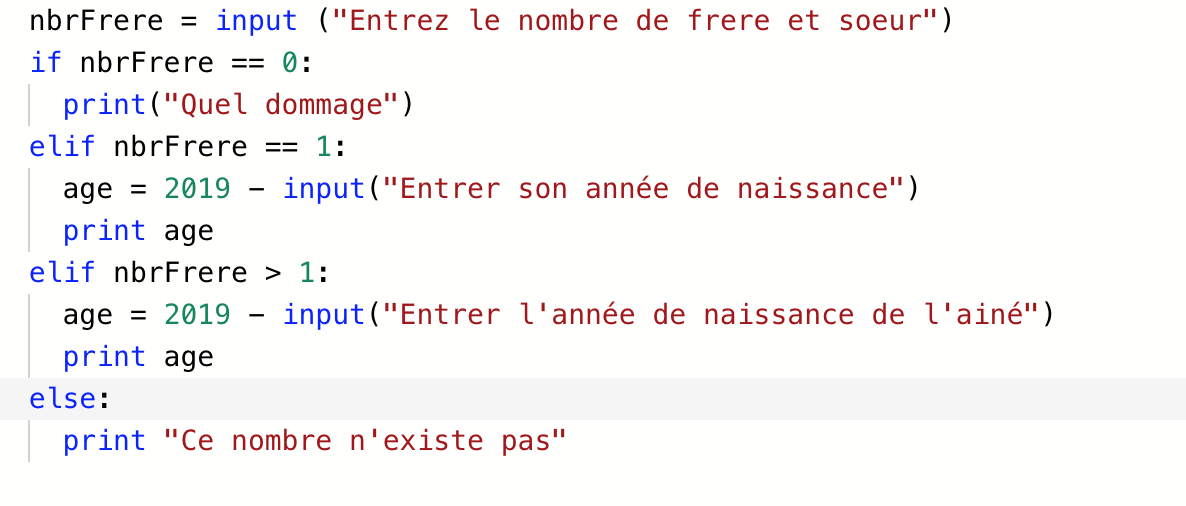
Ensuite, il faut introduire la notion de variable avec leur type, leur utilité et leur manipulation. La fonction input peut être présenté à ce moment afin de faire comprendre que cette fonction donne une variable qu’il faut sauvegarder pour qu’elle soit réutiliser, contrairement au print.

À ce point, demander aux campeurs de faire en sorte que la console demande leur année de naissance et que l’ordinateur retourne leur âge.

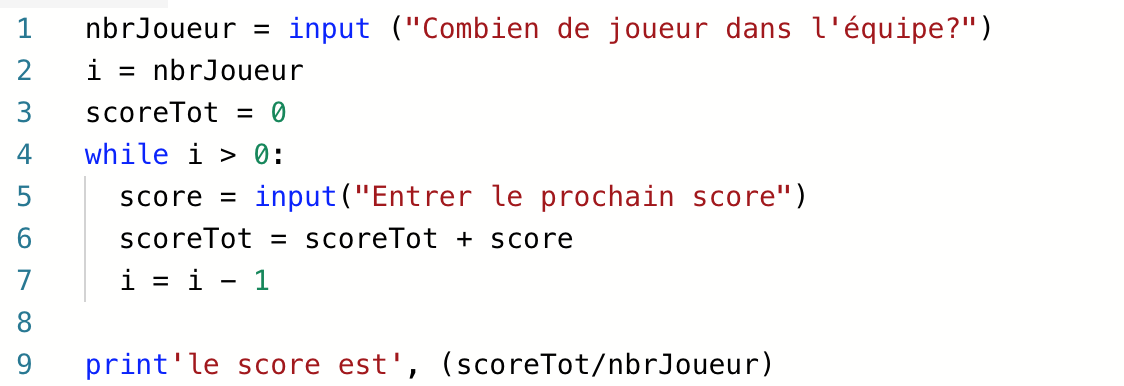


La présentation des conditions suivra. TRÈS IMPORTANT : les campeurs doivent comprendre que les conditions fonctionnent avec l’indentation en python et qu’ils doivent respecter cela.

Demander aux campeurs de faire un programme qui demande le nombre de frères et sœurs. Si l’utilisateur répond 0, il ne se passe rien. Si l’utilisateur répond une autre réponse valide, l’ordinateur lui demande d’entrer l’année de naissance d’un d’eux pour calculer son âge.



Si le temps et la maturité du groupe le permet, aborder la notion des boucles et de leur différence (while et for). Demander aux campeurs de faire un programme qui demande le nombre de joueur dans une équipe



Regarder avec eux les défis python proposés qui seront à faire dans leur temps libre

**Matériel**

-

**Manipulation**

Écrire au tableau l’arbre des conditions

Suivre le pwp

**Présentation suggérée**

Pour les 9-10, faire en même temps qu’eux les exercices. Pour les autres, les laisser expérimenter un peu

**Sources d’erreur**

Multiple, mais c’est ça programmer!

# Cryptographie

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  2 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  3 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  Cryptographie |

**Théorie**

Lire le powerPoint et le comprendre. Pour la situation typique, on a bob qui veut poser une question à alice, mais il sait que ça question est un peu niaiseuse donc il ne veut pas que eve comprenne ce qu’il demande. Que peut-il faire alors? Il va envoyer une lettre verrouiller dont seule alice a la clé. Comment eve envoit-elle sa réponse? Par le même principe. Dans la vraie vie, ce message est encrypté sous un format qui est connue de tous, quel pourrait alors être le problème? En fait, sans la clé, il est pratiquement impossible de connaitre le contenu du message. Sa force est donc dans la clé, pas dans son encryption.

Aussi, le temps pour décrypter un algorithme sans la clé n’est pas proportionnel à la puissance de l’ordinateur. Ce sera plutôt sous un format quadratique, donc plus un ordinateur est puissant, plus c’est lent de deviner son code.

Pour le masque jettable, on change la clé à chaque fois. On ne l’utilise pas sur internet car le message voulue verrait sa taille doublé, car il faudrait une clé pour chaque bit transmis.

**Matériel**

**Manipulation**

**Présentation suggérée**

Laisser les jeunes expliquer leur théories sur l’encryptions pendant le pwp.

Pour la partie du masque jettable, entre moniteur vous pouvez vous poser des questions ayant des réponses binaires. À chaque fois, si c’est pile on dit la vérité et face le contraire. On laisse les jeunes deviner comment la cryptographie se fait.

Commencer par le power point puis poursuivre en expliquant comment écrire un code caché. Laisser les jeunes écrire leur code d’encryption par la suite en utilisant python.

**Sources d’erreur**

…………………………….

# Tic-tac-toe

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  3 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  4 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  **python** |

**Théorie**

La théorie est la même que lors de l’introduction à Python

L'activité devrait débutée en demandant aux jeunes s'ils connaissent bien le jeu *Tic-Tac-Toe* et de démenteler les mécanismes du jeu avec eux. Comment le jeu se joue? Comment peut-on gagner? Quelles sont les règles? etc. Cette partie est cruciale -surtout pour les plus jeunes- car ce qui sera dit servira de base pour la programmation du jeu.

Lorsque les jeunes ont bien décortiqué les principes du jeu, il faut maintenant les aider à penser comment implémenter ces mécanismes sous forme de code. Poser des questions peut les aider à réfléchir: Comment représenter la grille? Comment obtenir le choix des joueurs? Comment déterminer quel joueur joue? etc. Adaptez-vous aux jeunes! Les 9-10 auront énormement de plus de difficulté que les 14-17; aider les si ils pédalent ou vont dans une direction trop compliquée!

Lorsque le *brainstorming* est terminé, c'est le temps de faire la programmation. Les plus jeunes peuvent commencer avec une version trouée, ou non. Les plus vieux feront le programme au complet par eux-même. L'animation devrait être adaptée selon le groupe: une animation magistrale devrait bien fonctionner avec les plus jeunes, mais les 14-17 préfèreront peut-être plus être laissé à eux-même. Sentez la vibe de votre groupe.

**Matériel**

-

**Manipulation**

1. Ouvrir le fichier
2. Faire le schéma de la logique du code
3. Aider les campeurs

**Présentation suggérée**

Commencer par montrer aux campeurs comment ouvrir un fichier sauvegardé sur une clé USB sur le site de repl.it. Faire un schéma de ce qui doit être fait par la suite au tableau.

Pour le schéma, montrer qu’on veut d’abord demander aux joueurs leur noms. Par la suite, on veut demander au joueur où il veut jouer. Si l’entrée n’est pas valide, on demande à l’utilisateur d’entrer de nouveau une réponse jusqu’à ce qu’elle le soit. Si elle est valide, on pose l’endroit où le joueur à joué et on passe à l’autre round. On détermine s’il y a un gagnant en regardant quelle combinaison peuvent gagner et voilà.

**Sources d’erreur**

Multiple, mais c’est ça programmer!

…………………………….

# Intelligence artificielle

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  2 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  1 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  Intelligence artificielle |

**Théorie**

Voir la présentation pwp pour comprendre les notes suivantes :

À l’origine, les tests pour trouver si les machines étaient des machines ou des humains, on demandait d’effectuer un calcul complexe. Puisque les puissances de calculs des machines sont bien plus élevées que celles des humains, les machines répondaient rapidement tandis que les humains lentement. On a donc ajouté un délai dans la réponse des machines. Par la suite, les testeurs ont compris qu’utiliser l’humour ou des expressions plus familière était une bonne technique pour vérifier s’il s’agit d’une machine, puisqu’évidemment à la base les machines ne comprennent pas cela. Les programmeurs ont donc ajouté ces particularités dans la conception des bots.

Le deep learning est une nouvelle façon d’apprivoiser l’intelligence artificielle avec un réseau de neurone.

Plusieurs experts ne s’entendent pas sur ce que les robots pourront faire à la place des humains ou pas. Par contre, ce qui fait quand même un consensus c’est les professions humaines qui vont être le plus difficiles à remplacer par exemple les psychologues, les travailleuses sociales, les infirmières, contrairement aux emplois demandant un travail plus physique, comme dans la construction ou les usines.

L’avantage d’avoir des robots dans les usines sont nombreux : ils sont généralement plus efficaces, n’ont pas besoin de pause ou de congé, ne tombe pas enceinte, n’ont pas besoin de dormir et peuvent se « blesser » sans que l’impact soit réellement grand.

L’intelligence artificielle est aussi de mieux en mieux outillé pour détecter des anomalies lors de scan. Présentemment, les IA sont déjà plus efficace que les radiologistes, et ceux-ci ont vu leurs tâches grandement diminuer au fil des dernières années. Elle sont cependant encore peu utilisé car on doit encore avoir la contre vérification du médecin(même si on sait que celui-ci fait plus d’erreurs que la machine) et on doit s’assurer de la protection des données, car l’IA partage ses données pour apprendre. Au niveau des robots chirurgiens, beaucoup de travail reste encore à faire, les machines doivent se pratiquer, mais comment?

Les voitures autonomes commencent à faire leur entrées. On spécule qu’elles pourraient réduire le traffic et le temps de transport tout en étant plus sécuritaire et efficace. Dans certaines villes, les métros se conduisent déjà seuls. Dans les avions, beaucoup de composantes sont implémantés (et très très très sécuritaire) pour éviter des erreurs du pilote.

D’autres métiers tels que les assistants personnels, les cuisiniers, les compositeurs, dans les calculs en finances, les comptables, les concierge, même certains programmeurs seront remplacés par des intelligences artificielles un jour!

Contrairement à ce qu’hollywood essaie de nous faire croire, il est impossible que les intelligences artificielles contrôlent un jour le monde en surpassant les humains. Les ordinateurs n’ont pas de conscience et sont à la base programmés pour réaliser une tâche qu’elles font en s’adaptant et en s’améliorant. Elle ne sont pas prévues pour faire une action dépassant son cadre d’activité.

**Matériel**

**Manipulation**

**Présentation suggérée**

**Sources d’erreur**

…………………………….

# Activité Santé et Sécurité

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  2 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  1 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  Santé et sécurité |

**Théorie**

La posture est très importante quand on passe plusieurs heures à faire une même chose. Comme les travailleurs d’usines et les chefs cuistots qui ont des postes de travail adapté, celui de l’informaticien devrait l’être aussi. Il faut essayer d’avoir le dos et le cou droit de façon naturelle lorsqu’assis au poste. Il est aussi important de prendre des pauses régulières afin de se dégourdir les jambes. En effet, certaines études suggèrent que le fait de rester assis pendant longtemps annule les bienfaits que peuvent avoir la pratique d’une activité physique intense dans la journée. De plus en plus dans les entreprises informatiques ont voit apparaitre des stations de travail alternative où l’informaticien peut faire du vélo, être assis sur un ballon, marcher ou simplement être debout pendant qu’il effectue ses tâches. Il faut aussi faire attention à la lumière bleue qui, selon certaines études, perturbent le cycle du sommeil quand elle sont utilisé avant d’aller dormir.

Il faut aussi faire attention à avoir d’autres activités que les jeux vidéo dans la journée. À petite dose, les jeux vidéo peuvent avoir de nombreux bienfaits : amélioration de la coordination, des réflexes, la vision périphérique, etc. Certaines compagnies de jeux vidéo travaillent également à faire des jeux pour les gens ayant besoin de réadaptation ou encore ayant certains problèmes de santé mentale. Cependant, certaines compagnies de jeux vidéo, tel que Epic Game (Fortnite) sont reconnus pour engager des psychologues et des experts en neuroscience. Le but de ces professionnels : rendre le jeu le plus addictif possible. En fait, dans un jeu comme Fortnite, lors d’évènements positifs, le cerveau se met à produire de la dopamine. Le but des professionnels engagés par la compagnie est de s’assurer que cela se produit juste assez souvent pour que le joueur reste accroché. Au bout d’un certain temps, le joueur ne jouera plus tellement par plaisir, mais plutôt parce qu’il en sent le besoin. De plus, l’aspect social autour d’un jeu comme Fortnite est très important. Tout le monde y joue ou presque dans les écoles primaires, il est donc facile de se sentir exclue quand on n’y joue pas. Ce sentiment de nécessité de jouer au jeu mène à la cyberdépendance, qui est maintenant reconnu comme un problème médical à part entière, car il possède les mêmes caractéristiques qu’une dépendance à la drogue ou au jeu d’argent par exemple. La meilleure prévention est de prendre des pauses régulières et de se fixer une limite de temps.

Aussi, bien qu’on en parle moins, on voit apparaitre dans les jeux vidéos des « Loot Box », des boites dont on fournit la clé au joueur mais dont on ne donne pas la boite, l’utilisateur doit payer pour l’avoir. On ne sait pas ce qui se trouve dans la boite, cela peut être une arme cool, un équipement puissant, ou d’autres choses. Le principe de ne pas savoir ce qu’on remporte, malgré avoir payé pour du contenu, ressemble étrangement au principe derrière les loteries. Cela peut donc amener des problèmes de jeux. Dans certains pays, on considère même cela carrément comme une loterie illégale et elles sont interdites.

**Matériel**

-

**Manipulation**

Démarrer un chronomètre qui sonne à chaque demi-heure

**Présentation suggérée**

Poser des questions aux jeunes sur ce qu’ils connaissent de la santé et de la sécurité par rapport à l’informatique

À chaque fois que l’alarme sonne, les campeurs devront prendre conscience de leur posture et se replacer

**Sources d’erreur**

# MasterMind

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  3 h |  | **Coût approximatif**  **par jeune**  0$ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  4 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  Python |

**Théorie**

La même que celle vue dans l’introduction à Python

**Matériel**

-

**Manipulation**

Ouvrir le fichier avec les jeunes, puis faire le schéma logique avec eux

**Présentation suggérée**

Pour les plus jeunes, faire le code avec eux

**Sources d’erreur**

Multiple, Mais c’est ça programmer!

# Défis Pythons

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  ? h |  | **Coût approximatif**  **par jeune**  0 $ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  2 à 5 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  python |

**Théorie**

La même que l’introduction à python

**Matériel**

-

**Manipulation**

Montrer aux campeurs l’emplacement des fichiers et comment l’ouvrir

**Présentation suggérée**

Pour les plus vieux, leur montrer qu’il est aussi possible de trouver des réponses sur internet en cherchant

**Sources d’erreur**

Multiple, mais c’est ça programmer!

# Évolution des langages de programmation

**Sommaire**

|  |  |  |
| --- | --- | --- |
| **Durée approximative**  1 h |  | **Coût approximatif**  **par jeune**  0 $ |
|  |  |  |
| **Niveau de difficulté**  **(1 à 5)**  1 |  | **Groupe d’âge**  9-17 ans |
|  |  |  |
| **Matériel**  - |  | **Notions abordées**  Langages de programmation |

**Théorie**

Un langage de programmation permet d’avoir un certain niveau d’abstraction entre ce qui se passe physiquement sur le processeur et ce que je veux qu’il fasse. Cela permet à moi-même et aux autres de mieux comprendre ce que j’écris. De même, il devient alors plus facile de voir et de corriger les erreurs.

Certains langages sont de plus bas niveau, ce qui signifie qu’ils seront plus difficiles à lire et à analyser, mais seront traités plus rapidement par la machine. Dans certains cas comme l’assembleur, on doit mentionner quelle partie de l’ordinateur utiliser, ce qui n’est pas évident à déchiffrer. À l’inverse, des codes de haut niveau tel python permettent de lire et de comprendre facilement, mais le temps d’exécution sera plus long. Certains programmes sont intermédiaires et offrent un bon compromis. Cependant, les ordinateurs étant de plus en plus rapides, dans bien des cas la vitesse d’exécution ne sera plus vraiment un critère de sélection.

**Matériel**

-

**Manipulation**

-Suivre la présentation PWP

**Présentation suggérée**

Poser des questions aux jeunes

**Sources d’erreur**

-