ACTIVITÉ (DÉCONNECTÉE) – DÉBAT MOUVANT : LANGAGES DE PROGRAMMATION

CRÉATION DE CONTENU > 3.4 PROGRAMMATION

CONVIENT POUR	AGE	NIVEAU DE COMPÉTENCE	FORMAT	DROITS D'AUTEUR	LANGUE(S)
Tous publics	Adolescents, Adultes, Seniors	Niveau 3	Fiche d'activité	Creative Commons (BY-SA)	Français

Cette activité s’organise autour d’un débat émouvant pour comprendre et surtout pour dédramatiser la programmation et ses langages. En effet, la programmation n’a pas à être effrayante et peut être accessible à tous !

Objectif général	Connaissances
Temps de préparation pour l'animateur	moins d'une 1 heure
Domaine de compétence	3 – Création de contenu
Temps requis pour compléter l'activité (pour l'apprenant)	0 – 1 heure
Nom de l'auteur	Aurelie Corvot
Ressource originellement créée	Français

DÉROULÉ

1 Introduction

Le débat mouvant, qu'est-ce que c'est ?

Il s'agit d'un débat dans lequel les participant.e.s prendront part physiquement. Ielles se déplaceront dans l'espace en fonction de leur réponse (d'accord/neutre/pas d'accord).
Par exemple, ielles se placeront en ligne : pour répondre « d'accord » ielles feront un pas en avant, pour « neutre » ielles resteront sur place et pour « pas d'accord », un pas en arrière.
Pour cette première phase, les participant.e.s ne parlent pas pour ne pas s'influencer les uns les autres.
Dans un second temps les participant.e.s s'expriment sur la question posée en expliquant leurs choix.

Conseil médiation

Les affirmations proposées ci-dessous sont des exemples testés par nos médiateur.rice.s, nous savons donc qu'elles suscitent bien le débat. Cela ne vous empêche pas de créer vos propres affirmations ! En revanche, nous vous conseillons de choisir des affirmations qui n'ont pas de réelle réponse (pour créer un débat) et surtout de les tester avant (avec vos collègues, vos amis, votre famille…) pour vous assurer qu'elles fonctionnent bien pour ce type d'animation.

Vous pouvez utiliser ce débat mouvant en guise d'introduction d'une activité autour de la programmation. Ou l'adapter pour en faire un bilan de connaissance à la suite d'une activité.

2 Les questions - Débat sur les langages de programmation

1. Le choix d'un langage, c'est un peu comme le choix d'une arme sur un champ de bataille.

On peut aimer les grosses massues ou alors, la fine épée. Aucune des deux n'est meilleure que l'autre, tout est dans l'art et la manière, mais aussi et surtout du besoin initial.

2. Vous devriez choisir le bon langage de codage dès le début

Il y a un débat centré sur le langage le plus approprié pour débuter en programmation et sur le langage le plus adapté aux enfants. Si nous considérons les langages utilisés dans la vraie vie, il est toujours mieux de commencer avec un langage dont la syntaxe est facile à apprendre. Python et Javascript en sont de bons exemples et peuvent être facilement utilisés sur tous les appareils et tous les navigateurs web. Mais il est impossible de choisir le bon langage, car chaque enfant est différent et, alors que certains enfants peuvent facilement saisir la syntaxe d'un langage, d'autres pourraient avoir du mal à l'apprendre. Chaque enfant doit choisir son langage de programmation préféré, ou abandonnera la programmation, ce qui n'est pas grave. De plus, il est très compliqué de prévoir quel langage sera le plus demandé quand cette nouvelle génération grandira.

3. Il est nécessaire d'apprendre plusieurs langages de programmation

L'apprentissage d'un langage de programmation doit se faire par rapport à votre but, d'une personne à une autre, l'apprentissage ne se fera pas de la même façon. Il est donc important de se poser les bonnes questions. Il est clair que si vous programmez juste pour le plaisir vous n'allez certainement pas vous embêter à apprendre plusieurs langages de programmation. Posez-vous ces questions : quel est votre but dans la programmation ? Quel est votre projet ? Quelle est votre ambition ? Suis-je un programmeur du dimanche ou un programmeur expert ?

4. Tous les langages informatiques sont des langages de programmation.

Il existe une multitude de langages informatiques. Beaucoup sont des langages de programmation, mais pas tous. Cette confusion est fréquente avec les langages utilisés dans la création de sites WEB. (x)Html n'est en aucun cas un langage de programmation. Il permet simplement de présenter un document de façon statique. Pour les CSS, c'est pareil.

5. Un bon développeur c'est celui qui sait programmer dans tous les langages

Être un bon développeur ne signifie pas simplement bien programmer. Il faut avoir de bonnes facultés d'écoute, de synthèse et d'anticipation. L'écoute évidemment pour satisfaire au mieux les demandes. Un esprit synthétique pour être le plus efficace que possible. Et l'anticipation, pour appréhender les futurs besoins, c'est-à-dire permettre la modularité de son travail afin de faciliter le travail pour les personnes qui seront amenées à faire évoluer ou à poursuivre notre travail.

6. La programmation est réservée aux surdoués ou aux matheux

Préjugé produit par des personnes n'ayant jamais pratiqué la programmation ou qui ne comprennent pas ce que représente ce domaine. Il est complètement faux bien entendu. D'autant que si vous utilisez les maths en programmation ce sera bien souvent des maths de base à la portée de tous. Il est évident qu'il

faut avoir de bonnes bases en informatique de manière générale pour bien assimiler ce domaine.

7. Pour programmer, il faut être bon en anglais

C’est très recommandé de connaître un minimum l’anglais quand on commence la programmation sans pour autant être bilingue. Il faut juste savoir lire et comprendre l’anglais technique orienté informatique. C’est indispensable pour pouvoir programmer. Bien entendu certains mots techniques vous seront inconnus mais en faisant une petite traduction des mots en question sur la toile cela fera l’affaire. Mais pourquoi faut-il avoir quelques notions en anglais me direz-vous ? C’est simple : la plupart des ressources sont fournies en anglais et souvent uniquement dans cette langue. C’est souvent le cas de la documentation par exemple. Des traductions existent bien entendu mais très souvent elles sont de mauvaise qualité et changent le sens de ce qui est initialement écrit en anglais. Lorsque vous tomberez sur un message d’erreur provoqué par votre code il sera en anglais. Il vous sera plus facile de trouver réponse à celui-ci sur des sites anglo-saxons plutôt que sur un site francophone.

8. Il faut du matériel onéreux pour commencer la programmation

Contrairement à ce que l’on pourrait penser, il n’est pas nécessaire d’investir dans du matériel très haut de gamme pour commencer à programmer. Un simple ordinateur de base vous suffira pour commencer la programmation. Bien entendu en investissant dans du matériel plus onéreux vous aurez plus de puissance et donc plus d’aisance dans votre programmation. Mais il existe des exceptions à la règle. Si vous souhaitez faire des applications mobiles pour iOS (iPhone ou iPad) vous aurez besoin d’un Mac et ce sont des appareils très onéreux. Vous pouvez aussi avoir besoin de matériel supplémentaire dans certains cas. C’est le cas si vous souhaitez faire de la programmation sur une carte Arduino où vous devrez posséder la carte pour tester ce que vous programmez. Le coût n’est pas énorme (il faut compter une vingtaine d’euros) mais il faut prendre en compte ces achats.

9. Un programmeur est une personne solitaire et introvertie

Un programmeur est une **personne normale** comme tout le monde. Il sort de chez lui, s’amuse, partage, rencontre d’autres personnes, explore le monde qui l’entoure et surtout analyse le côté conceptuel de ce qu’il rencontre. Vous vivez votre vie comme n’importe quel autre être humain sur Terre ? J’ai une bonne nouvelle pour vous : vous avez le comportement qu’il faut pour pouvoir programmer.

10. La programmation est une discipline qui fait appel à des compétences masculines

Le pionnier de la programmation est une pionnière du nom de Ada Lovelace. Pourtant, l’informatique de manière générale est perçue comme un métier d’homme. Beaucoup de femmes ont une mauvaise image de l’informatique comme la plupart des débutant.e.s notamment à cause du préjugé selon lequel le programmeur est quelqu’un de solitaire et d’introverti. Le métier commence à se démocratiser chez

les femmes et de plus en plus de femmes apprennent la programmation. Au passage, il faut savoir qu'une association existe pour valoriser et promouvoir les développeuses. Elle se nomme « Duchess France ».

3 Pour aller plus loin : petit point sur l'histoire des langages de programmation

Année : 1842-1843

Langage : Premier langage de programmation

Ada Lovelace est connue pour avoir réalisé le premier programme informatique, lors de son travail sur un ancêtre de l'ordinateur : la machine analytique de Charles Babbage

Année : 1957

Langage : Fortran

FORmula TRANslation, est le plus vieux langage encore utilisé. Créé par John Backus, le langage a été développé pour effectuer des calculs scientifiques, mathématiques et statistiques de haut niveau.

Le langage est encore utilisé en aérospatial, industries automobiles, par le gouvernement, et les institutions de recherche.

Année : 1959

Langage : Cobol

Common Business Oriented Langage est derrière la majorité des systèmes de transactions commerciales exécutant le traitement de cartes de crédit, les guichets automatiques, les appels téléphoniques et cellulaires, les systèmes hospitaliers, les systèmes gouvernementaux, les systèmes automobiles et les systèmes de signalisation routière. L'équipe de développement de cobol, dirigée par le Dr. Grace Murray Hopper, a entrepris de créer un langage uniforme et convivial pour les transactions commerciales.

Année : 1964

Langage : Basic

Développé par des étudiants de l'université de Dartmouth, le code d'instruction symbolique débutants tout usage a été conçu pour être un langage simplifié pour ceux qui n'ont pas de solides bases techniques ou mathématiques. Une version modifiée, écrite par Bill Gates et Paul Allen est devenu le premier produit de Microsoft. Il a été vendu à M.I.T.S. pour l'Altair.

Année : 1969

Langage : C

« C » a été développé entre 1969 et 1973 par Dennis Ritchie dans les laboratoires de téléphonie Bell pour une utilisation avec le système d'exploitation Unix. Il a été nommé « C » parce que ses caractéristiques ont été dérivées d'une langue antérieure appelée « B ». « C » était devenu assez puissant pour que la plus grande partie du noyau Unix soit réécrite dans « C ». L'un des premiers noyaux du système d'exploitation implémentés dans une langue autre que l'assemblage.

Année : 1970

Langage : Pascal

Le langage a été nommé pour Blaise Pascal, crédité pour avoir inventé la première machine à additionner en 1641. Niklaus Wirth a créé Pascal comme un outil d'enseignement et il a évolué pour servir à un usage commercial.

Année : 1983

Langage : C++

À partir des laboratoires Bell, Bjarne Stroustrup a modifié le langage C en C ++ et a créé ce que beaucoup considèrent comme le langage de programmation le plus populaire de tous les temps. Il figure dans le top dix des langages de programmation depuis 1986 et a obtenu le statut de Temple de la renommée en 2003.

Année : 1987

Langage : Perl

Larry Wall, un programmeur Unix, a créé Perl après avoir tenté d'extraire des données pour un rapport

et trouvé que Unix ne pouvait pas effectuer les opérations dont il avait besoin. Le **Practical Extraction Report Langage** a été décrit par son inventeur comme un langage pour « faire son travail ».

Année : 1991

Langage : Python

Monty Python a servi d'inspiration pour le nom de ce langage. Guido Van Rossum a développé Python pour résoudre les problèmes dans le langage ABC et continue de servir de concepteur principal.

Année : 1993

Langage : Ruby

Yukihiro « matz » Matsumoto nommé le langage Ruby, pour la pierre de naissance de Juillet. Il a développé le langage en mélangeant des parties de ses langages préférés, Perl, Smalltalk, Eiffel, Ada et Lisp.

Année : 1995

Langage : PHP

Rasmus Lerdorf a développé PHP pour remplacer un ensemble de scripts Perl utilisés pour maintenir sa page d'accueil personnelle. Aujourd'hui, PHP est devenu une partie intégrante de l'architecture Web fonctionnant sur plus de 20 millions de sites Web.

Année : 1995

Langage : Java

Une équipe de développeurs du Sun Microsystems dirigée par James Gosling a créé java pour gérer des décodeurs pour la télévision interactive. Java fonctionne maintenant sur plus de 1,1 milliard de PC dans le monde entier et de nombreux sites Web ne peuvent pas fonctionner sans le langage.

Année : 1995

Langage : Javascript

Java et javascript ne sont pas liés et ont une sémantique très différente.

Javascript a été à l'origine développé par Brendan Eich de Netscape sous le nom Mocha.

Javascript utilise une syntaxe influencée par celle de « C ». Bien que destiné à fonctionner sur le client (navigateur), il trouve maintenant l'utilisation sur le serveur en tant que node.js. En outre, Ajax dépend de javascript.

Année : 1995

Langage : Ruby on rail

Ruby on rails a été extrait par David Heinemeier Hansson de son travail sur Basecamp, un outil de gestion de projet de 37signals. Hansson a publié Ruby on rails en open source en juillet 2004, mais n'a pas partagé les droits d'engagement sur le projet avant février 2005. Il est maintenant sur la version 3.0.7 et compte plus de 1800 contributeurs.

4 Pour aller plus loin

Conseil médiation

Pour aller plus plus loin sur le sujet, nous vous conseillons de vous référer à la fiche outil « Algorithmes et langages de programmation«