

Proposition de Référentiel en Sciences Informatiques de SI²



Dans le cadre de Digital Wallonia d'une part et du Pacte pour un Enseignement d'Excellence d'autre part, il est prévu d'étoffer les compétences numériques des jeunes en y incluant des compétences en algorithmique, représentation de données,... Par conséquent, il est nécessaire de former en amont les enseignants aux sciences informatiques, tant dans le cadre de la refonte de leur formation initiale que via la formation continue pour les enseignants déjà en poste. Les universités et hautes écoles sections informatiques, regroupées dans le groupe SI² - Sciences Informatiques dans le Secondaire Inférieur, ont été mandatées par la Région Wallonne, en accord avec la Fédération Wallonie Bruxelles, pour développer des offres de formation dans ce domaine.

Ce groupe de travail vise à terme à introduire l'enseignement des sciences informatiques dans le secondaire inférieur, les arguments en ce sens sont repris dans un document disponible sur le site <http://sicarre.be>. Dans cette perspective et pour donner un cadre aux formations proposées actuellement aux enseignants, un référentiel de compétences a été élaboré.

Retrouvez plus d'informations sur <http://sicarre.be> ou en envoyant un mail à info@sicarre.be

Ce référentiel de compétences s'articule autour de **5 axes** nécessaires pour comprendre le fonctionnement de cette machine particulière qu'est l'ordinateur et sa « capacité » à réaliser des tâches variées via des logiciels.

Le premier axe « *Représentation des données* » consiste à percevoir comment les nombres, les caractères et donc les textes mais également les images sont représentés en machine. L'objectif est que l'élève perçoive que toutes ces données sont stockées dans un format binaire où l'ordinateur ne fait finalement que manipuler des 0 et des 1, ces manipulations étant facilement réalisables par une machine électrique.

Le deuxième axe « *Algorithmique* » vise à faire percevoir à l'élève que toute solution à un problème donné doit être traduite en étapes et ensuite en instructions simples que l'ordinateur peut effectuer. L'élève doit ainsi au terme du cursus être capable de lire et appliquer un algorithme sur différents jeux de données et évaluer la qualité du résultat obtenu. Il doit également être capable de concevoir un algorithme pour résoudre un problème simple¹ et de discuter de l'efficacité comparée de deux algorithmes résolvant le même problème.

Le troisième axe « *Programmation* » consiste à traduire ces algorithmes dans un langage de programmation² pour pouvoir les faire exécuter par un ordinateur. Cette étape importante permet d'aborder l'apprentissage des sciences informatiques via des projets qui mènent à des réalisations contraintes. Les domaines d'application des sciences informatiques s'ouvrent alors à une large gamme d'apprentissages : mathématiques, sciences, langues étrangères, ... (voir le document décrivant quelques activités possibles sur le site sicarre.be).

Le quatrième axe « *Matériel* » vise à acquérir les clés qui permettent de comprendre comment une machine peut exécuter un programme en percevant le rôle des composants de base d'un ordinateur. Il ne s'agit évidemment pas d'expliquer en détail le fonctionnement d'un processeur ou de technologies permettant de fabriquer un disque dur ; l'objectif est d'avoir un modèle mental organisant les composants d'un ordinateur et explicitant leur rôle.³

Le cinquième axe « *Réseaux et sécurité* » aborde le fonctionnement des réseaux pour que cet éclairage scientifique permette de mieux percevoir les problématiques liées à la sécurité des données⁴.

Chaque axe est composé de compétences de base que, selon nous, tout citoyen, et donc *a fortiori* tout élève, devrait avoir acquises et de pistes pour aller plus loin, sans pour autant

¹ Par « simple », on imagine des algorithmes qui ne nécessitent pas plus de quelques dizaines d'instructions.

² L'approche choisie ici consiste à utiliser au début des apprentissages un langage de programmation visuel tel que Blockly ou Scratch. Ces langages ont l'avantage qu'ils évitent les erreurs de syntaxe, l'oubli d'un « ; » par exemple. Ils permettent donc que l'apprenant se concentre sur la structure de son programme et l'algorithme qu'il souhaite implémenter.

³ Comme les notions d'atome et de molécule fournissent un modèle permettant de « penser » la chimie.

⁴ Lorsqu'il a acquis quelques éléments permettant de comprendre le fonctionnement d'un réseau informatique tel qu'Internet, l'élève sait où peuvent circuler les données qu'il y place et les traces qu'il y laisse. Il est alors plus enclin à suivre les « bons conseils » qu'on peut lui donner en matière de sécurité informatique.

vouloir faire de chaque élève un informaticien. Quelques exemples d'activités pouvant être réalisées avec des élèves sont listées. Elles ne sont bien évidemment pas exhaustives. Des exemples interdisciplinaires sont suggérés également sur le site sicarre.be.⁵

⁵ https://sicarre.be/pmwiki/uploads/Documents/informatique_et_autres_disciplines.pdf

Représentation des données

Au terme de la formation en Sciences informatiques des deux premières années du secondaire, l'élève sera capable de :

Comprendre la représentation interne des données par une machine

- Expliquer pourquoi une machine représente les données sous forme binaire (0 et 1)

Comprendre la représentation binaire des nombres

- Passer de la représentation décimale d'un nombre naturel vers sa représentation binaire
- Passer de la représentation binaire d'un nombre naturel vers sa représentation décimale

Comprendre la représentation binaire des caractères

- Utiliser un code pour représenter un caractère sous forme binaire (coder)
- Utiliser un code pour retrouver un caractère à partir de son encodage binaire (décoder)
- Discuter des avantages et inconvénients de l'utilisation d'un standard d'encodage des caractères (p.e. ASCII)

Comprendre la représentation binaire des images

- Numériser une image sous forme d'une grille de pixels
- Reconstituer une image à partir d'une grille de pixels
- Discuter du lien entre la qualité de l'image et sa résolution
- Discuter des avantages et inconvénients d'un algorithme de compression d'image

Pour aller plus loin :

- Expliquer l'intérêt et utiliser un algorithme de chiffrement basique (chiffre de César)
- Expliquer l'intérêt et utiliser un code correcteur et détecteur d'erreur basique (bit de parité)
- Utiliser un code pour la représentation des notes de musique, de vidéo, du braille

Exemples d'activités

- <http://visatice.ulg.ac.be/> section « Comprendre le traitement de l'information numérique » et « Produire et retravailler une image »
- <https://www.csunplugged.org/en/topics/binary-numbers/>
- <https://sciencesinfuses.info.ucl.ac.be> section « Données »

Algorithmique

Au terme de la formation en Sciences informatiques des deux premières années du secondaire, l'élève sera capable de :

Lire et comprendre un algorithme simple permettant de résoudre un problème donné

- Expliquer un algorithme simple
- Simuler l'exécution d'un algorithme simple
- Adapter un algorithme simple existant

Concevoir un algorithme simple permettant de résoudre un problème donné

- Écrire un algorithme comme une suite d'instructions
- Identifier les instructions de base à disposition
- Définir des instructions non ambiguës
- Définir et utiliser des expressions de manière appropriée dans un algorithme
- Définir et utiliser des variables de manière appropriée dans un algorithme
- Utiliser des structures de contrôles (instructions conditionnelles ou boucles) de manière appropriée dans un algorithme
- Combiner des instructions, des structures de contrôle, des variables pour écrire un algorithme

Utiliser un algorithme simple pour résoudre un problème donné

- Identifier les entrées/sorties d'un algorithme simple
- Discuter des conditions d'utilisation d'un algorithme simple
- Comparer l'efficacité de deux algorithmes
- Identifier une suite d'instructions qui constituent un tout réutilisable
- Réutiliser une suite d'instructions en définissant une fonction

Exemples d'activités

- <https://sciencesinfuses.info.ucl.ac.be> section « Algo »
- <https://pixees.fr/dis-maman-ou-papa-cest-quoi-un-algorithme-dans-ce-monde-numerique-%E2%80%A8/>
- <https://www.csunplugged.org/en/topics/searching-algorithms/>

Programmation

Au terme de la formation en Sciences informatiques des deux premières années du secondaire, l'élève sera capable de :

Comprendre qu'un langage de programmation est une façon de décrire un algorithme de sorte qu'il soit compréhensible par une machine

- Identifier la correspondance entre les instructions de base d'un algorithme et celles du langage de programmation à utiliser
- Identifier et utiliser les types de données adéquats
- Traduire un algorithme simple dans un langage de programmation
- Rédiger un programme simple dans un langage de programmation visuel
- Exécuter sur machine un programme simple
- Établir les cas de base à tester
- Vérifier que l'exécution du programme produit les résultats attendus sur base d'un jeu de données de test
- Argumenter de l'intérêt d'utiliser plusieurs jeux de données de test

Pour aller plus loin :

- Expliquer qu'à un programme simple écrit dans un langage de programmation visuel correspond un programme dans un langage de programmation textuel.
- Traduire un algorithme simple dans un langage de programmation textuel

Exemples d'activités

- <https://studio.code.org/s/20-hour>
- <https://scratch.mit.edu/>

Matériel

Au terme de la formation en Sciences informatiques des deux premières années du secondaire, l'élève sera capable de :

Comprendre le rôle des différents composants d'un ordinateur dans l'exécution d'un programme

- Lister les composants de base d'un ordinateur (processeur, mémoire de travail, mémoire à long terme, périphériques d'entrée-sortie)
- Expliquer la fonction de chaque composant de base d'un ordinateur
- Discuter de la différence entre la mémoire à long terme et la mémoire de travail
- Simuler l'exécution d'une séquence d'instructions pour illustrer le rôle des composants de base d'un ordinateur (par exemple, somme de 3 nombres)
- Discuter de la manière dont une opération de haut niveau utilise les composants de base (par exemple, la visualisation d'une vidéo)

Exemples d'activités :

- Découverte des composants d'un ordinateur en ouvrant une machine (activité testée mais pas encore en ligne)

Réseaux et sécurité

Au terme de la formation en Sciences informatiques des deux premières années du secondaire, l'élève sera capable de :

Expliquer comment accéder à des données sur une machine distante

- Décrire le rôle du navigateur comme programme qui affiche une page web distante (aller la chercher et interpréter le code)
- Expliquer que le contenu et la mise en forme d'une page web reposent sur un langage de balises (HTML)
- Identifier les balises qui permettent d'insérer des liens hypertextes et des images dans une page web
- Identifier les composants d'un lien internet (protocole, nom de domaine, nom du document)
- Discuter de la relation entre le nom de domaine et l'adresse IP
- Décrire le découpage des données en paquets et leur acheminement via des relais
- Distinguer Web, Internet et réseau afin d'utiliser ces termes à bon escient

Pour aller plus loin :

- Décrire l'impact de l'architecture du web sur la sécurité
 - sur le pistage des utilisateurs (tracking)
 - sur la confidentialité des données et leur intégrité
- Identifier les balises qui permettent d'insérer du code dans une page web
- Expliciter le rôle des cookies

Exemples d'activités :

- Simulation de réseaux de façon déconnectée (activité testée mais pas encore en ligne)