#wallcode | digital wallonia .be

Avec le soutien de la Wallonie

INFORMATIQUE, QUELLES COMPÉTENCES ?

De 8 à 15 ans

COORDONNÉ PAR : ASBL HYPOTHESE

AVEC LA PARTICIPATION ET L'INTELLIGENCE COLLECTIVE DE :

technobel

{SI}² Sciences Informatiques pour le Secondaire Intéressant

interface3 namur — Se connecter pour plus d'égalité

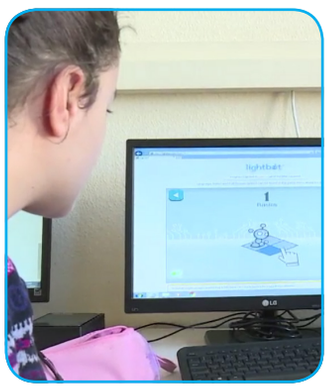
pass

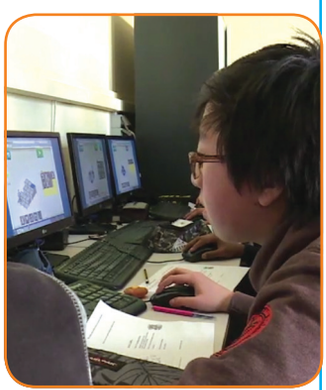
KODO

CoderDojo

RÉFÉRENTIEL DE COMPÉTENCES POUR LES ACTEURS DE WALLCODE

CAHIER ANIMATEUR





Introduction

Le projet #WallCode Digital Wallonia fédère les acteurs et les initiatives visant à développer les compétences numériques de la nouvelle génération de talents et à stimuler les vocations pour les métiers informatiques, particulièrement dans le domaine de la programmation informatique, de la logique algorithmique et de la robotique.

Les partenaires du projet #WallCode ont réfléchi ensemble aux objectifs poursuivis lors des ateliers éducatifs qu'ils mettaient en place. Ce document est le résultat de leur cogitation.

Les compétences à développer ont été organisées en 5 domaines : Algorithmique – Programmation – Hardware (l'ordinateur et ses composants) - Représentation des données - Réseaux et sécurité.

Sans avoir l'ambition de l'exhaustivité, cette liste des apprentissages constitue pour nous un premier référentiel de compétences pour que chaque acteur du domaine puisse situer son travail. Cette base commune est amenée à évoluer, nous vous invitons à transmettre vos suggestions.

SOMMAIRE

ALGORITHMIQUE

Lexique :

- Un algorithme est une suite finie et non ambigüe d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat. Par exemple, suivre une recette de cuisine, suivre les instructions d'un gps,... sont des exemples d'algorithmes dans la vie quotidienne.
- Un algorithme simple est un algorithme composé d'une dizaine d'instructions et qui ne comprend pas de fonction.
- Une expression dans un algorithme est une partie du code qui peut être évaluée. Par exemple «5+3» est une expression numérique qui vaut 8.

Les compétences que nous souhaitons travailler :

1) Lire et comprendre un algorithme ou appréhender la notion d'algorithme :

- 1.1. Décortiquer une tâche en une suite d'actions ordonnées;

Identifier chaque instruction présente dans un algorithme. Mettre en évidence que ces instructions exécutées l'une à la suite de l'autre permettent de résoudre un problème ou d'obtenir un résultat.

- 1.2 .Expliciter oralement un algorithme;

Rendre claires et précises les instructions qui composent un algorithme.

- 1.3. Expliciter via un schéma naturel et normalisé (logigramme) un algorithme;

Écrire un algorithme dans un formalisme type schéma.

- 1.4. Exécuter un algorithme;

Suivre les instructions données par un algorithme. Attention à ne pas anticiper/deviner/...

- 1.5. Généraliser;

À partir de cas particuliers, extraire une solution générale.

- 1.6. Identifier dans un algorithme ce que sont les instructions, les structures de contrôle (boucles, conditions, ...) et les variables;

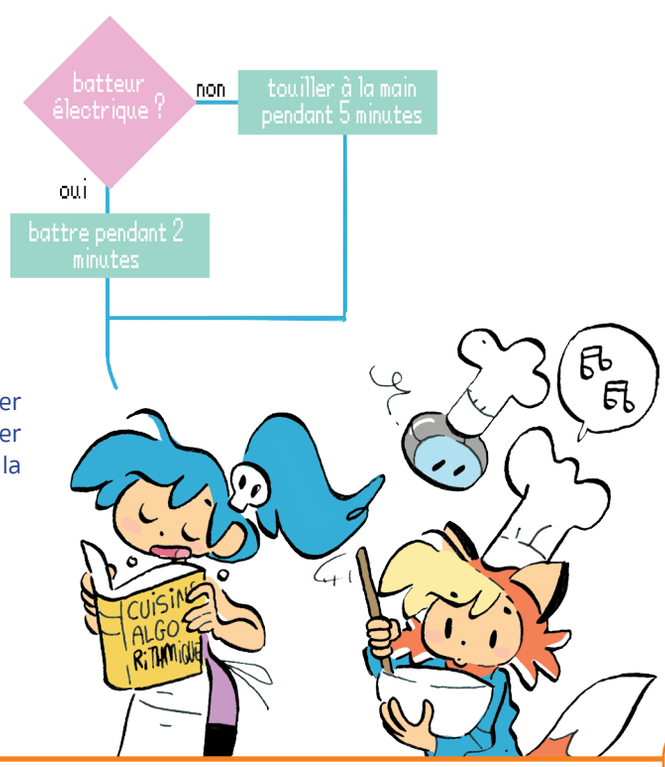
Une **instruction** est une opération de base;
Une **boucle** est une **structure de contrôle** destinée à exécuter une suite d'instructions plusieurs fois de suite;
Les **instructions conditionnelles** permettent d'effectuer des tests vérifiant certaines conditions puis d'exécuter des instructions en fonction des résultats de ces tests;
Les **variables** en algorithmique permettent d'assigner un nom à une valeur. La valeur de la variable peut évoluer durant l'exécution de l'algorithme.

- 1.7. Identifier les entrées/sorties d'un algorithme simple;

Il est important de pouvoir identifier quelles données sont nécessaires pour la bonne exécution d'un algorithme. Les entrées et sorties permettent de valider l'exécution après coup.

- 1.8. Identifier une suite d'instructions qui constituent un tout réutilisable.

Identifier une suite d'instruction qu'on peut réutiliser en l'insérant dans une fonction permet de simplifier la lecture, l'écriture, de ne pas dupliquer et facilite la maintenance lorsqu'il s'agit d'un programme.



2) Concevoir et écrire un algorithme :

• 2.1. Écrire un algorithme comme une suite d'instructions ;
- Identifier les instructions de base à disposition ;
- Formuler des instructions non ambiguës ;
- Identifier ou donner les instructions les plus simples qui permettent de construire un algorithme. Attention à la précision du langage naturel.

• 2.2. Utiliser des expressions de manière appropriée dans un algorithme ;
L'objectif est de pouvoir faire la différence entre ce qui peut être évalué et ce qui ne peut pas l'être, en vue d'utiliser, composer et construire les expressions sur base des données présentes.

• 2.3. Utiliser des variables de manière appropriée dans un algorithme ;
Être capable d'identifier les cas qui demandent de stocker une valeur et bien comprendre la différence entre l'identifiant (le nom de la variable) et sa valeur (quand on l'évalue).

• 2.4. Utiliser des structures de contrôles (instructions conditionnelles ou boucles) de manière appropriée dans un algorithme ;
Une instruction conditionnelle permet de créer un branchement dans un algorithme. La branche sélectionnée dépend de la valeur exprimée dans la condition (Si <condition> Alors <instructions> Sinon <instructions>). L'utilisation d'une boucle est adéquate quand il faut répéter une ou plusieurs instructions.

• 2.5. Combiner des instructions, des structures de contrôle, des variables pour écrire un algorithme ;
Pouvoir utiliser les différentes sortes d'action de manière appropriée.

• 2.6. Réutiliser une suite d'instructions en définissant une fonction.
Identifier une suite d'instructions à utiliser plusieurs fois dans un algorithme. Nommer cette suite et la réutiliser à bon escient.

3) Utiliser un algorithme :

• 3.1. Discuter des conditions d'utilisation d'un algorithme simple (expliciter les conditions d'utilisation);
Documenter les cas d'utilisation de l'algorithme.

• 3.2. Comparer l'efficacité de deux algorithmes.
Pour un problème donné ou une tâche à accomplir, on peut imaginer différentes méthodes de résolution.
On désire une méthode qui nécessite un temps d'exécution minimum ou qui minimise la quantité de mémoire utilisée. On compare l'efficacité de deux algorithmes par rapport au temps d'exécution ou à la quantité de mémoire utilisée.

Les activités pour travailler...

L'ALGORITHMIQUE

1) Lire et comprendre un algorithme ou appréhender la notion d'algorithme :

- 1.1. Décortiquer une tâche en une suite d'actions ordonnées ;
- 1.2 .Expliciter oralement un algorithme ;
- 1.7. Identifier les entrées/sorties d'un algorithme simple.

2) Concevoir et écrire un algorithme :

- 2.1. Écrire un algorithme comme une suite d'instructions.

Identifier les instructions de base à disposition ;
Formuler des instructions non ambiguës.

Activité 1 : Algorithme de la tartine (SI² - Kodo Wallonie)

Description :

L'activité se déroule de manière individuelle.
L'animateur se transforme pour une durée déterminée en robot.
Les élèves lui donnent des instructions afin qu'il réalise la tâche suivante :
«Faire une tartine beurrée à la confiture».

Les activités pour travailler...

L'ALGORITHMIQUE

1) Lire et comprendre un algorithme ou appréhender la notion d'algorithme :

- 1.4. Exécuter un algorithme ;
- 1.5. Généraliser ;
- 1.6. Identifier dans un algorithme ce que sont les instructions, les structures de contrôle (boucles, conditions, ...) et les variables.

2) Concevoir et écrire un algorithme :

- 2.4. Utiliser des structures de contrôle (instructions conditionnelles ou boucles) de manière appropriée dans un algorithme ;
- 2.5. Combiner des instructions, des structures de contrôle, des variables pour écrire un algorithme.

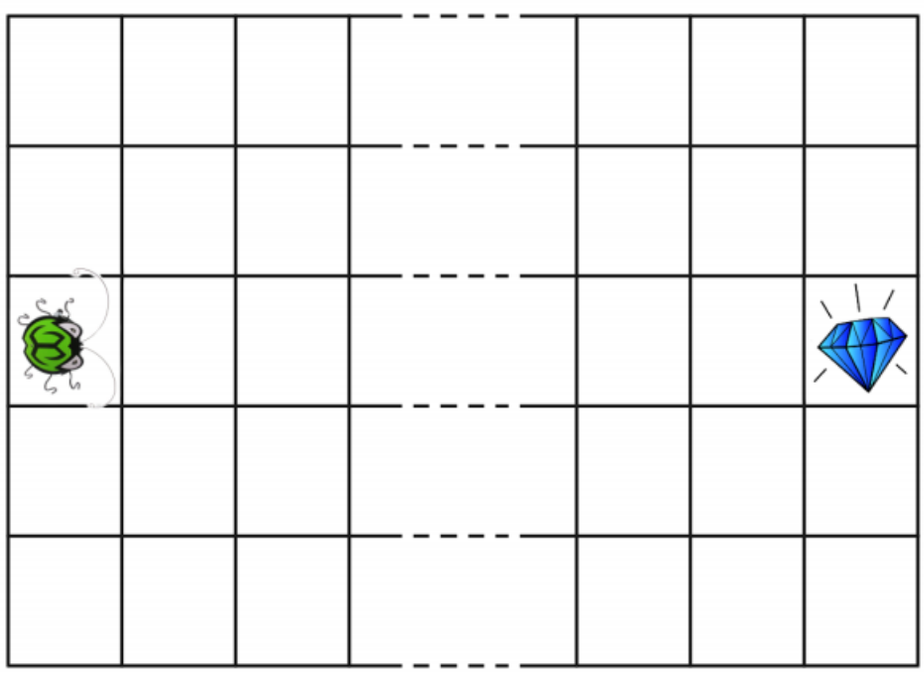
3) Utiliser un algorithme :

- 3.2. Comparer l'efficacité de deux algorithmes.

Activité 2 : Cherchons le trésor (SI²)

Description : (from SI²)

À chaque grille donnée, les apprenants rédigent des instructions claires au travers d'un algorithme pour que l'insecte robot atteigne le diamant. Ils doivent se mettre d'accord entre eux sur les instructions à utiliser. L'animateur donne des grilles de plus en plus complexes. Les apprenants réalisent des algorithmes de plus en plus généraux.



Exemple de grille pour «Cherchons le trésor» - SI² (ci-dessus)

Les activités pour travailler...

L'ALGORITHMIQUE

1) Lire et comprendre un algorithme ou appréhender la notion d'algorithme :

- 1.3. Expliciter via un schéma (naturel et normalisé - logigramme) un algorithme;
- 1.5. Généraliser;
- 1.8. Identifier une suite d'instructions qui constituent un tout réutilisable.

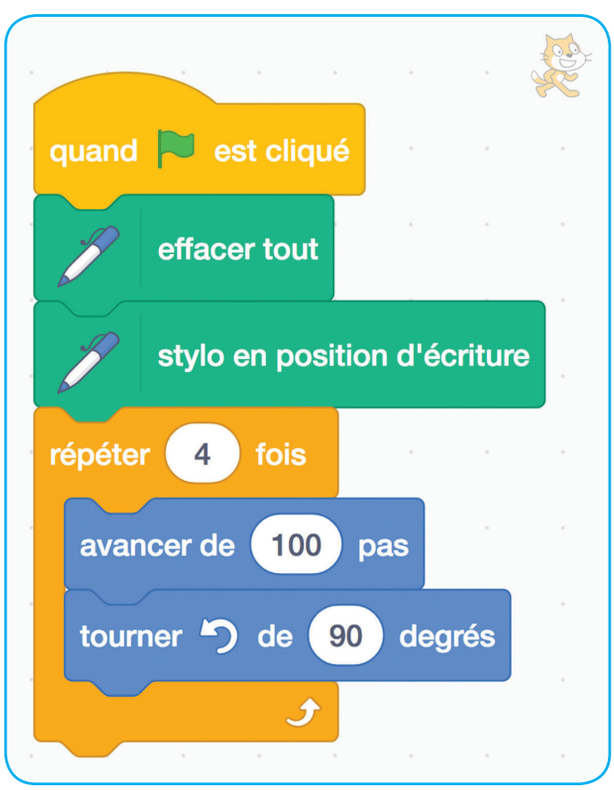
2) Concevoir et écrire un algorithme :

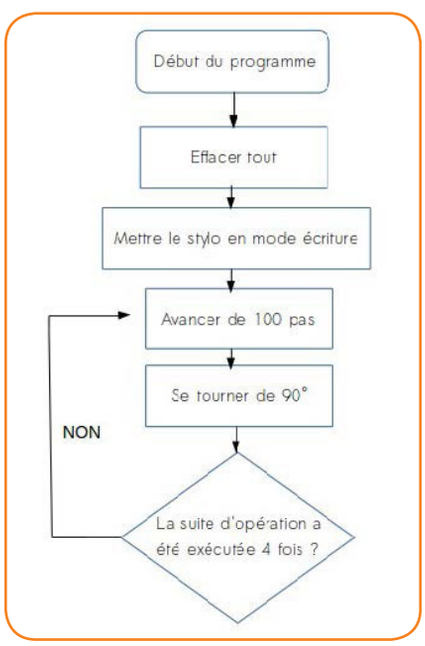
- 2.4. Utiliser des structures de contrôle (instructions conditionnelles ou boucles) de manière appropriée dans un algorithme.

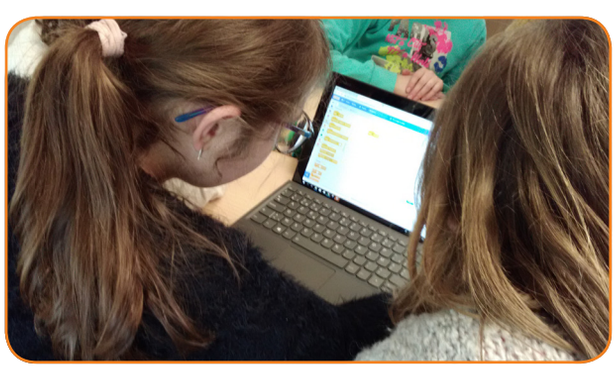
Activité 3 : Programme Scratch – Tracer des polygônes

Description :

- À l'aide de Scratch, commencer par tracer un carré, ensuite un triangle,... ;
- Créer une fonction qui permet de tracer n'importe quel polygône ;
- Dessiner un logigramme pour représenter l'algorithme.







PROGRAMMATION

Lexique :

• La programmation consiste à traduire des algorithmes dans un langage de programmation pour pouvoir les faire exécuter par un ordinateur.

• Un programme est un texte qui décrit un algorithme que l'on souhaite faire exécuter par une machine. Ce texte est écrit dans un langage particulier, appelé langage de programmation.

Les compétences que nous souhaitons travailler :

1) Traduire un algorithme simple dans un langage de programmation visuel :

- 1.1. Identifier la correspondance entre les instructions de base d'un algorithme et celles du langage de programmation à utiliser ;

Un langage de programmation permet à une machine de comprendre un algorithme. Toutes les étapes de l'algorithme que nous voulons traduire doivent être exprimées dans le langage de programmation.

- 1.2. Identifier et utiliser le type de données adéquates.

Certaines expressions représentent des nombres, des chaines de caractères, des valeurs booléennes,... Utiliser ces expressions à bon escient.

2) Vérifier/justifier de l'efficacité d'un programme :

- 2.1. Vérifier que l'exécution du programme produit les résultats attendus sur base d'un jeu de données de test ;

Vérifier que l'implémentation d'un algorithme est correcte en le testant avec un jeu de données et en comparant le résultat obtenu avec le résultat attendu.

- 2.2. Argumenter de l'intérêt d'utiliser plusieurs données de test ;

Choisir plusieurs jeux de données de test afin que toutes les possibilités du programme (notamment pour les instructions conditionnelles) soient parcourues au moins une fois. Tester des cas particuliers.

Les activités pour travailler...

L'ALGORITHMIQUE

2) Concevoir et écrire un algorithme :

- 2.5. Combiner des instructions, des structures de contrôle, des variables pour écrire un algorithme ;
- 2.6. Réutiliser une suite d'instructions en définissant une fonction.

Les activités pour travailler...

LA PROGRAMMATION

1) Traduire un algorithme simple dans un langage de programmation visuel :

- 1.1. Identifier la correspondance entre les instructions de base d'un algorithme et celles du langage de programmation à utiliser ;

Activité 1 : LightBot

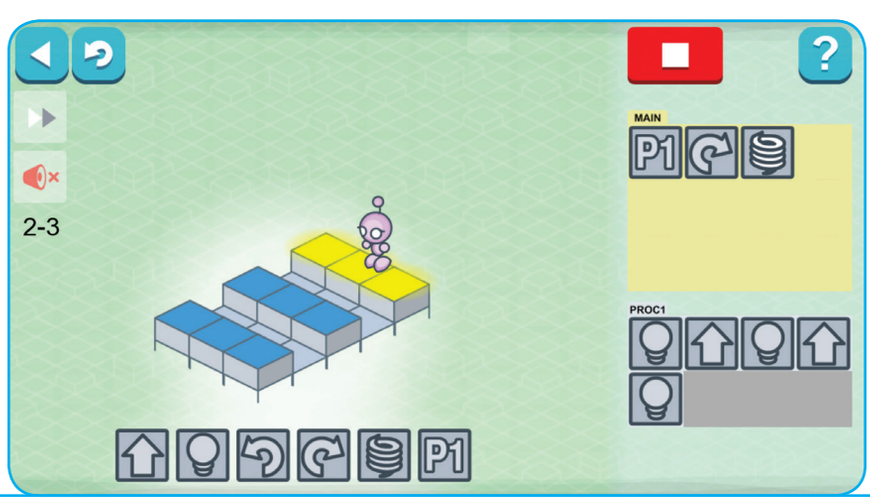
Description :

Rendez-vous sur *http://lightbot.com/flash.html*

L'animateur présente LightBot aux élèves et explique la mission du petit robot : allumer toutes les cases bleues par lesquelles il passe.
LightBot est composé de plusieurs niveaux de difficulté. Au fur et à mesure des défis réussis, les apprenants accèdent à des niveaux plus difficiles et à de nouvelles instructions.

Les élèves identifient les instructions de base qui sont à leur disposition. Ils combinent ces instructions ainsi que des structures de contrôle pour écrire l'algorithme permettant de réaliser le défi proposé. Ils doivent aussi pouvoir écrire et réutiliser une suite d'instructions en définissant une fonction (procédure).

Exemple de construction d'une procédure avec LightBot :



Les activités pour travailler...

LA PROGRAMMATION

1) Traduire un algorithme simple dans un langage de programmation visuel :

1.2. Identifier et utiliser le type de données adéquates.

2) Vérifier/justifier de l'efficacité d'un programme :

2.1. Vérifier que l'exécution du programme produit les résultats attendus sur base d'un jeu de données de test ;
2.2. Argumenter de l'intérêt d'utiliser plusieurs données de test ;

Activité 2 : Scratch

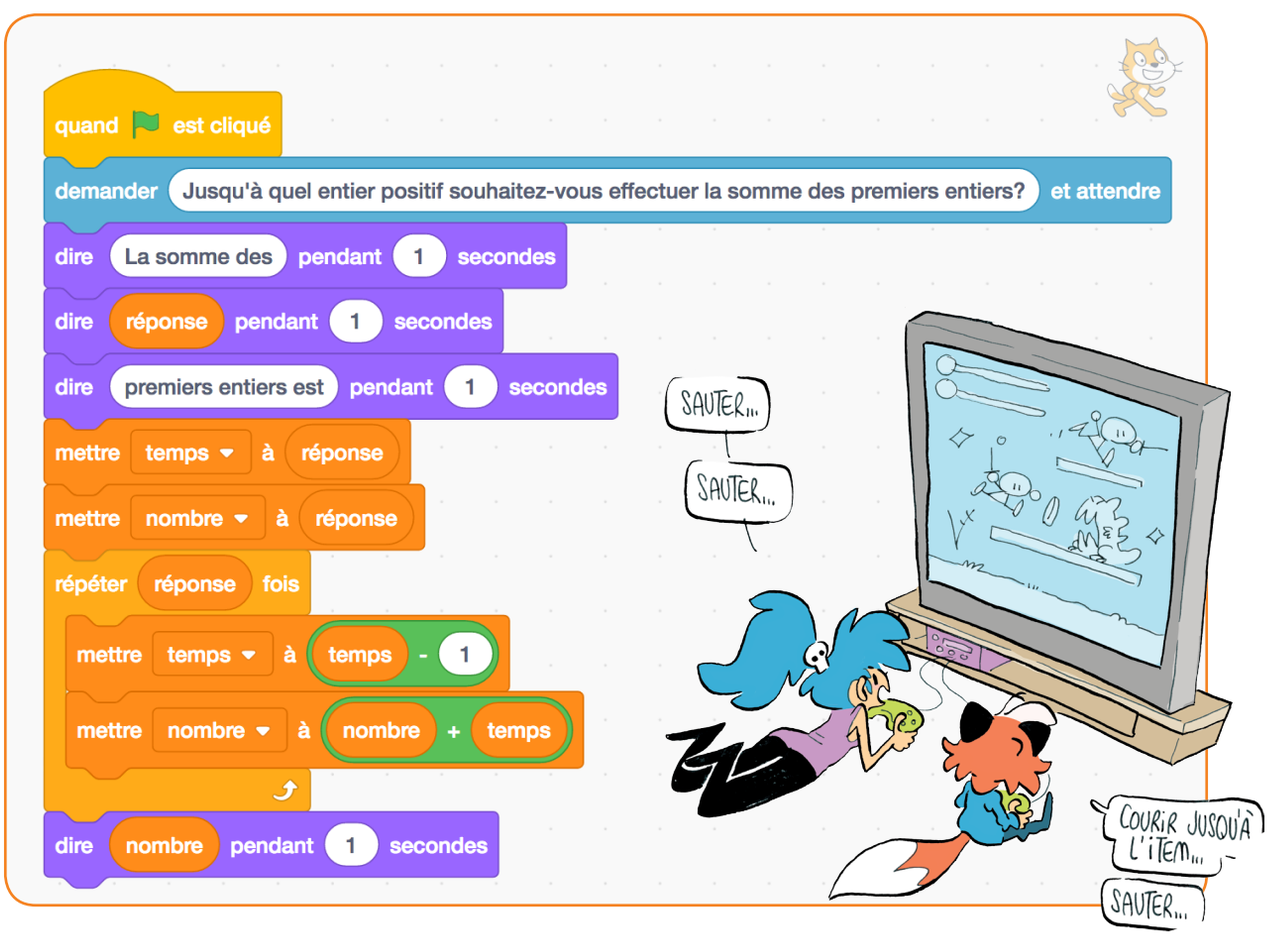
Description :

Lors d'une activité de programmation, on donne la consigne suivante :

"Construisez un programme pour calculer la somme de n premiers entiers."

- Que se passe-t-il si l'utilisateur ne donne pas un nombre entier comme réponse ?
- Que se passe-t-il si l'utilisateur donne un nombre entier négatif ?
- ...

Exemple de programme qu'un élève pourrait écrire :




JE SUIS
UN ORDINATEUR !

HARDWARE OU "L'ORDINATEUR ET SES COMPOSANTS"

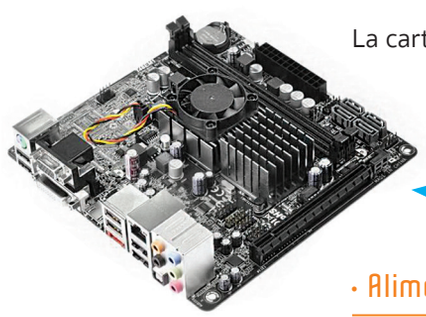
Lexique :

- Hardware : ensemble de l'équipement matériel, mécanique, magnétique, électrique et électronique, qui entre dans la constitution d'un ordinateur ou des machines de traitement de l'information en général.

Cette section a sa raison d'être pour…

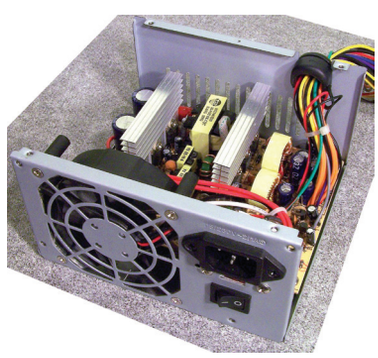
- Connaitre et comprendre les outils informatiques (devices) que nous utilisons / qui nous entourent au quotidien (smartphone, tablette, ordinateur fixe ou portable, console de jeu, assistant virtuel, dispositif domotique,...) ;
- Décoder la publicité, acheter un appareil qui convient à ce qu'on souhaite en fonction de son usage, acheter de manière écologique ;
- Permettre une plus grande autonomie (pour personnaliser ou réparer ses appareils).

• Carte mère :



La carte mère est le centre nerveux d'un ordinateur, lieu d'échanges et de calculs.

• Alimentation :



L'alimentation fournit du courant électrique à l'ensemble des composants d'un ordinateur. Elle est chargée de convertir la tension électrique du secteur en différentes tensions continues utilisées par la carte mère et les périphériques.

• Processeur ou CPU (« Central Processing Unit ») :

C'est le processeur qui organise les échanges de données entre les différents composants et qui fait les calculs qui permettent à l'ordinateur d'interagir avec l'extérieur.

• RAM :

La mémoire vive ou RAM («Random Access Memory») accueille, de manière temporaire, toutes les informations (données ou programmes) qui viennent progressivement enrichir les capacités du système.

• Disque dur :

Mémoire de l'ordinateur. On y enregistre les données et les programmes que l'ordinateur est susceptible de traiter.

• Périphériques entrées-sorties et leurs connectiques :

Un périphérique est un dispositif connecté à un système de traitement de l'information central et qui ajoute à ce dernier des fonctionnalités. Un périphérique peut fournir des informations au système informatique (périphérique d'entrée) ou il peut en sortir (périphérique de sortie). On entend par connectique les connexions physiques permettant les liaisons électriques et la transmission des données.

Les compétences que nous souhaitons travailler :

1) Identifier et expliquer les différents composants d'un ordinateur :

- 1.1. Pouvoir distinguer de quoi est composé un ordinateur (carte mère, alimentation, processeur, RAM, disque dur, périphériques entrées-sorties et leurs connectiques) ;
- 1.2. Citer et expliquer la fonction des composants de base d'un ordinateur ;

► Distinguer les caractéristiques de la mémoire à long terme et de la mémoire de travail ;

La mémoire à long terme sert à stocker à long terme de grandes quantités d'informations.

La mémoire de travail sert à stocker des informations qui peuvent à tout moment être consultées ou modifiées. Le contenu de cette mémoire disparaît lorsque l'appareil utilisé n'est plus sous tension.

► Différencier les différents supports de mémoire ;

Il en existe trois :

La ROM = mémoire morte. Elle est fixée une fois pour toute lors de la programmation de l'appareil. Elle peut être lue plusieurs fois mais ne peut pas être modifiée.

La RAM = mémoire vive, mémoire en "accès direct". Elle accueille de manière temporaire toutes les informations qui viennent progressivement enrichir les capacités du système. Elle perd les informations lorsque l'appareil est mis hors tension.

La mémoire Flash est identique à la RAM mais elle ne perd pas les informations lorsque l'appareil est mis hors tension.

► Associer à un périphérique sa connectique en fonction de son rôle ; par exemple, pouvoir utiliser un projecteur pour projeter l'écran de son ordinateur, pouvoir charger des photos depuis son smartphone,...

- 1.3. Identifier les différents ordinateurs parmi les objets du quotidien.

Un ordinateur est une machine qui permet (avec des données en entrée) de traiter, d'effectuer des opérations et de donner des résultats en sortie. Un ordinateur contient une mémoire.

2) Expliquer le rôle du système d'exploitation :

- 2.1. Définir et décrire le rôle du système d'exploitation comme interface avec les composants physiques de l'ordinateur ;

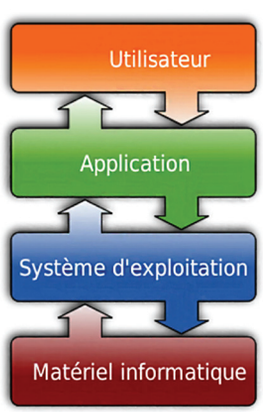
Pour comprendre que la couche matérielle en elle-même offre beaucoup d'instructions de base à la couche applicative au travers du système d'exploitation.

- 2.2. Définir et décrire le rôle du système d'exploitation comme interface avec les autres logiciels ;

Expliquer comment une opération de haut niveau utilise les composants de base.

- 2.3. Savoir qu'il existe plusieurs systèmes d'exploitation (mac OS, gnu Linux, Android, Windows,...) et ce qui les différencie (compatibilité, prix, licence, ergonomie,...) ;

Voici quelques caractéristiques qui différencient les différents systèmes d'exploitation :



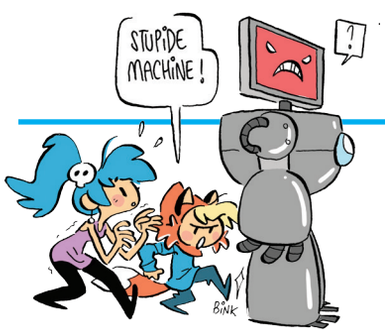
- MacOS : Closed source, bonne gestion de l'installation/désinstallation de logiciels, moins de logiciels disponibles qu'avec Windows, très stable globalement (moins de « plantage »), pour les débutants permet de moins encrasser la machine (moins de virus), design soigné, plus convivial et performant. Surcoût lié à la marque.
- GNU/Linux : Open source, installation du matériel moins facile, taux de « plantage » très réduit, pour les débutants permet de moins encrasser la machine, gratuit.
- Android : Système d'exploitation proposé par Google pour les appareils mobiles, gratuit, open source, problèmes de compatibilité et de sécurité, décliné pour plusieurs objets connectés (TV, montre, voiture,...).
- Windows : Code source fermé, payant, plus difficile pour installation/désinstallation de logiciel, plus de compatibilité au niveau des périphériques.

Les activités pour travailler...

LE HARDWARE

1. Identifier et expliquer les différents composants d'un ordinateur :

- 1.1. Pouvoir distinguer de quoi est composé un ordinateur (carte mère, alimentation, processeur, RAM, disque dur, périphériques entrées-sorties et leurs connectiques);
- 1.2. Citer et expliquer la fonction des composants de base d'un ordinateur ;
- 1.3. Identifier les différents ordinateurs parmi les objets du quotidien.



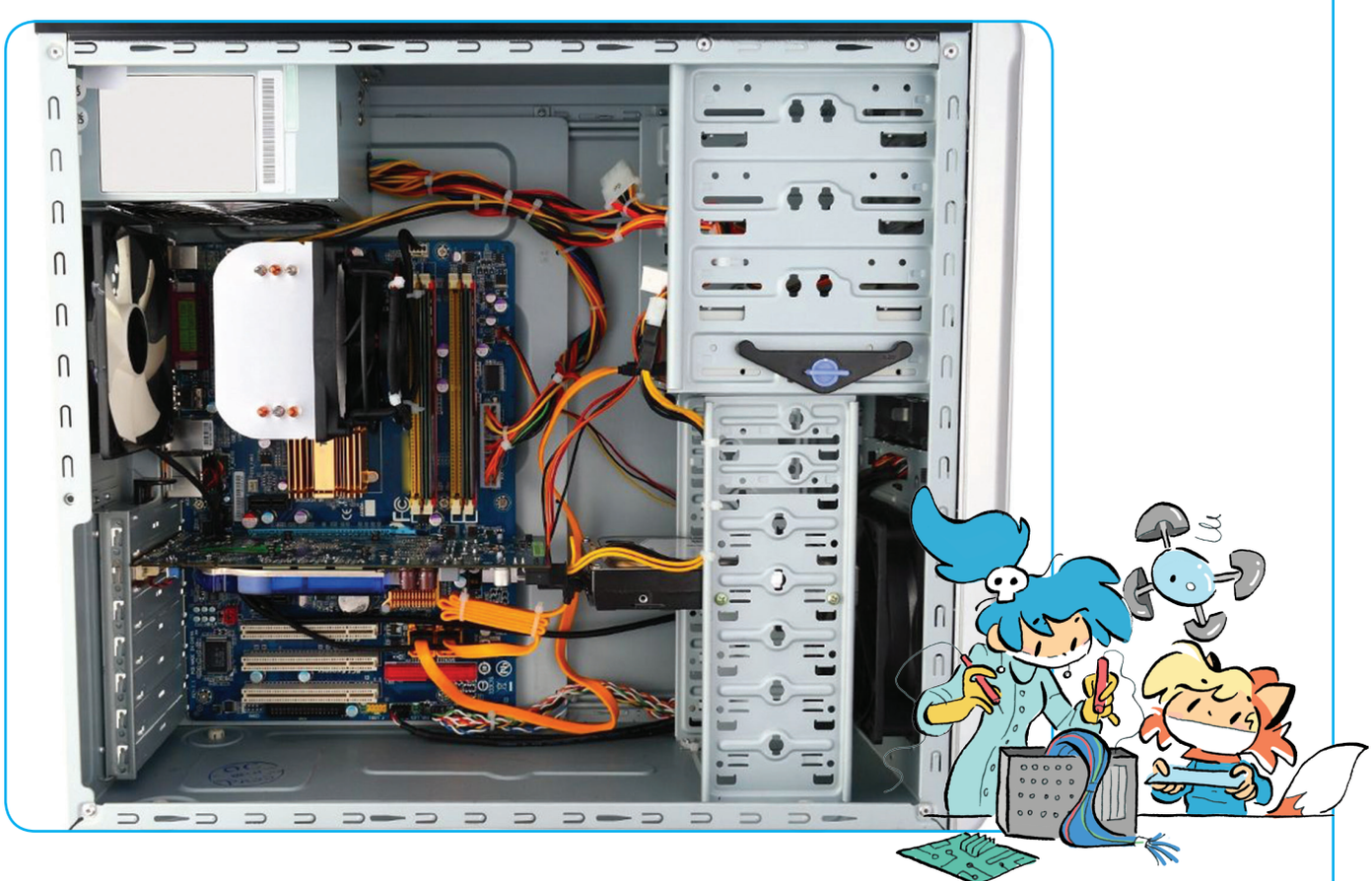
Activité 1 : Un ordinateur, cette machine stupide (SI²)

Description :

Au début de l'activité, l'animateur demande aux apprenants de communiquer leur(s) représentation(s) de ce qu'est un ordinateur.

Ensuite, il présente aux élèves un ensemble d'images parmi lesquelles sont représentés des ordinateurs.
Il demande aux élèves d'entourer ceux qu'ils pensent être des ordinateurs. Lors de la mise en commun, on établit les caractéristiques qu'un objet doit présenter pour être un ordinateur.

L'animateur ouvre une tour d'ordinateur et présente les éléments de base qui la composent.



À la fin de l'activité, les élèves sont capables de dire sous quelles caractéristiques une machine est un ordinateur et ils sont capables d'en reconnaître un parmi les objets du quotidien.
Ils peuvent citer les composants de base d'un ordinateur et en expliquer la fonction.


01001000
10011101
BLA!
BLA!
BLA!
LE LANGAGE
HUMAIN-MACHINE
EN 10 LEÇONS

REPRÉSENTATION DES DONNÉES

1) Binaire :

• 1.1. Expliquer pourquoi une machine exprime les données sous forme binaire.

Un ordinateur ne traite que des données sous forme binaire (base deux). À l'origine, seule la présence (représentée par un « 1 ») ou l'absence (représentée par un « 0 ») de courant dans un transistor permet de représenter une unité d'information.

2) Nombres :

• 2.1. Passer d'une représentation décimale (base dix) d'un nombre à sa représentation binaire (base deux) et vice-versa.

Comprendre la décomposition d'un nombre en base 2. Pouvoir donner la représentation décimale d'un nombre lorsque celui-ci est exprimé en base 2.

3) Caractères :

• 3.1. Utiliser un code pour représenter un caractère sous forme binaire et vice-versa ;

Par exemple avec la norme ASCII, qui est une correspondance entre une représentation binaire des caractères de l'alphabet, avec les signes et symboles qui constituent cet alphabet.

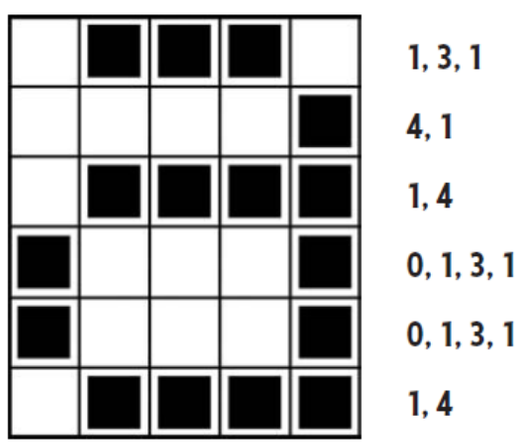
• 3.2. Formuler les avantages et inconvénients de l'utilisation d'un standard d'encodage.

Par exemple, la norme ASCII propose une écriture plus compacte, mais elle est adaptée à une langue particulière, d'où toutes les extensions qui existent.

4) Images :

• 4.1. Numériser une image (noir et blanc ou couleurs) sous forme d'une grille de pixels ;

Les écrans (d'ordinateur) sont divisés en une grille de petits points appelés « pixels ». Dans une image en noir et banc, chaque pixel est soit noir, soit blanc. Lorsque l'ordinateur enregistre une image, il enregistre l'emplacement de chaque pixel. Ci-contre, voici comment nous pourrions représenter une image de la lettre « a ». Le premier nombre représente toujours le nombre de pixels blancs, le second représente le nombre de pixels noirs,.



• 4.2. Utiliser un code pour représenter la couleur d'un pixel sous forme binaire et vice-versa ;

Pour représenter une image colorée en binaire, on utilise un nombre pour représenter la longueur de la séquence et un nombre pour préciser la couleur.

• 4.3. Reconstituer une image à partir d'une grille de pixels ;

Voir 4.1.

• 4.4. Faire le lien entre la qualité et la résolution d'une image ;

Plus le nombre de pixels est élevé sur la grille, plus la résolution de l'image sera élevée.

• 4.5. Créer un pixel art/ un gif ;

http://pix-attack.com/editeur-pixel-art

5) Organisation des données :

- 5.1. Savoir qu'un ordinateur a besoin de formats pour interpréter des données et les stocker sous forme de fichier;

Les données (ou informations) en informatique sont constituées de séquence d'octets, i.e. d'une séquence de nombres, permettant des usages divers. Le format d'un fichier désigne la nature des informations contenues dans celui-ci et donc du ou des logiciel(s) nécessaire(s) pour le lire.

- 5.2. Illustrer avec quelques exemples les formats de fichier permettant de stocker différents types de données ;

Par exemple :

- pdf : portable document format. Le format pdf pour un fichier est universel. Que ce soit une image ou un document. Ce format permet de conserver les polices, les images, la mise en page,...
- png : portable network graphics. Format pour stocker les images. Le plus souvent utilisé pour les sites nternet.

- 5.3. Expliquer l'intérêt de stocker des fichiers ;
- 5.4. Expliquer l'intérêt d'organiser des fichiers ;
- 5.5. Organiser le stockage des données en arborescence (répertoires, fichiers).

Les activités pour travailler...

LA PROGRAMMATION

Initiation à la programmation

Les activités pour travailler...

LE HARDWARE

1) Identifier et expliquer les différents composants d'un ordinateur :

- 1.1. Pouvoir distinguer de quoi est composé un ordinateur (carte mère, alimentation, processeur, RAM, disque dur, périphériques entrées-sorties et leurs connectiques);
- 1.2. Citer et expliquer la fonction des composants de base d'un ordinateur.

Les activités pour travailler...

LA REPRÉSENTATION DES DONNÉES

1) Binaire :

- 1.1. Expliquer pourquoi une machine exprime les données sous forme binaire.

2) Nombres :

- 2.1. Passer d'une représentation décimale (base dix) d'un nombre à sa représentation binaire (base deux) et vice-versa.

3) Caractères :

- 3.1. Utiliser un code pour représenter un caractère sous forme binaire et vice-versa.

Activité 1 : Ordi, étrange animal numérique (Pass)

Description :

Le but de l'animation est de familiariser les élèves à l'outil informatique et d'apprendre à comprendre ce qu'est un ordinateur et comment il fonctionne.
Il y a 3 étapes distinctes : l'anatomie de l'ordinateur (les composants et comment ils fonctionnent), la façon de penser de l'ordinateur (introduction au binaire), et comment le « dresser » (programmation).

Déroulement succinct, gestion du public, rôle de l'animateur :

L'animation est suivie par les élèves en suivant les indications de l'animateur.

1. La première étape se fait autour de l'animateur. Il s'agit ici de reconnaître et comprendre les différents composants d'un ordinateur (carte mère, processeur, RAM, disque dur, ventilateur, alimentation, écran).
2. La seconde se fait d'abord autour de l'animateur puis en groupe de deux.
   Elle a pour but d'expliquer le concept du binaire en résolvant quelques énigmes de traduction. La fin de cette étape permet d'allumer les ordinateurs.
3. La troisième partie se fait alors sur les ordinateurs. C'est une initiation à la programmation.

Les activités pour travailler...

LA REPRÉSENTATION DES DONNÉES

4) Images :

- 4.1. Numériser une image (noir et blanc ou couleurs) sous forme d'une grille de pixels ;
- 4.2. Utiliser un code pour représenter la couleur d'un pixel sous forme binaire et vice-versa ;
- 4.3. Reconstituer une image à partir d'une grille de pixels ;
- 4.4. Faire le lien entre la qualité et la résolution d'une image.

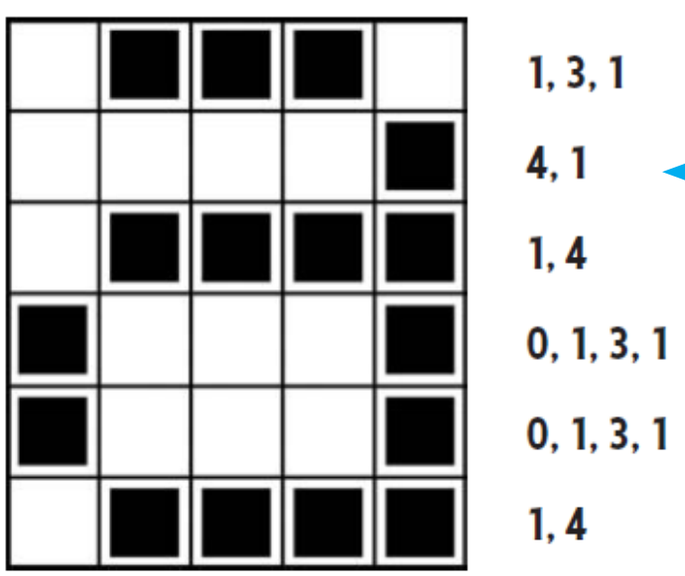
Activité 2 : La couleur par les nombres – Représentation d'une image (CS Unplugged)

Description :

Pour introduire le sujet, l'animateur pose quelques questions :

- « Dans quels cas un ordinateur a-t-il besoin de stocker des images ? » ;
- « Comment un ordinateur peut-il stocker des images alors qu'il ne sait que traiter des nombres ? ».

L'animateur explique comment un ordinateur peut stocker et interpréter les images en noir et blanc (voir image ci-dessous).



Exemple avec une image de la lettre « a ».

Les élèves codent et décodent plusieurs images en noir et blanc.
L'animateur introduit un code pour la couleur, les élèves inventent une image en couleurs et donnent le code qui correspond.

RÉSEAUX ET SÉCURITÉ

1) Web :

- 1.1. Distinguer web, Internet et réseau pour les utiliser à bon escient ;

Web = principale application d'internet permettant de partager des contenus ;
Internet = Réseau informatique qui relie des machines entre elles ;
Réseau = ensemble de machines connectées entre elles pour échanger des informations.

- 1.2. Décrire le rôle du navigateur comme logiciel qui affiche une page web distante ;

Les navigateurs Internet tels que Google Chrome, Mozilla Firefox, Internet Explorer,... sont conçus pour consulter et afficher des pages web distantes.

- 1.3. Reconnaitre une page web comme un format particulier et qui repose sur un langage de balises ;

Le Web est codifié par un langage de balises (langage HTML = *HyperText Markup Language*). Ces balises sont utilisées pour mettre en forme le texte de la page Web. Ces balises permettent aussi d'inclure dans le texte des éléments interactifs tels que des liens, des images, des animations,...

- 1.4. Identifier quelques balises, notamment celles qui permettent d'insérer des liens hypertextes et des images dans une page web ;

Par exemple, pour les liens hypertextes (pour une page pointée vers l'extérieur), on utilise :
 Site de l'ASBL Hypothèse

- 1.5. Décrire le découpage des données en paquets et leur acheminement via des relais.

Les activités pour travailler...

LE WEB

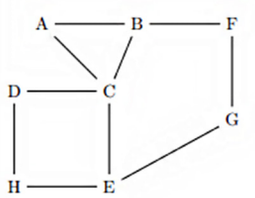
- 1.1. Distinguer web, Internet et réseau pour les utiliser à bon escient ;
- 1.5. Décrire le découpage des données en paquets et leur acheminement via des relais.

Activité 1 : « Le routage élastique » (Marie Duflot-Kremer)

Description :

Dans les réseaux informatiques, pour gérer la quantité impressionnante de données qui circulent, il y a des routeurs. Ces machines peuvent recevoir et envoyer des messages pour leur permettre, après plusieurs routeurs d'atteindre leur destination.

Pour ce faire, ils ont des tables qui leur disent vers quels voisins ils doivent envoyer un message ou pour quel destinataire est destiné le message. Pour cette activité, les élèves forment un réseau grâce à des liens accrochés à une ceinture qu'ils portent. Tous les élèves ne sont pas reliés entre eux. Ils représentent physiquement un réseau (voir figure ci-jointe pour le réseau et la liste des tables de routage).



A	B	C
B → B	A → A	A → A
C → C	C → C	B → B
D → C	D → C	D → D
E → C	E → C	E → E
F → B	F → F	F → B
G → C	G → F	G → E
H → C	H → C	H → E
D	**E**	**F**
A → C	A → C	A → B
B → C	B → C	B → B
C → C	C → H	C → G
E → H	D → H	D → B
F → C	F → G	E → G
G → H	G → G	G → G
H → H	H → H	H → G
G	**H**	
A → F	A → D	
B → F	B → D	
C → E	C → D	
D → E	D → D	
E → E	E → E	
F → F	F → E	
H → E	G → E	

La lettre en haut de chaque table dit à quel routeur elle appartient, et une ligne F ▶ B signifie que si j'ai un message pour F, je dois l'envoyer à B.

Pour l'activité, chaque élève reçoit la table qui correspond à son routeur.

Ensuite, chaque élève reçoit des messages (2 ou 3) avec une destination et on laisse les apprenants les acheminer.

Sur l'image ci-jointe, les apprenants sont reliés entre eux avec des élastiques.

RÉFÉRENCES :

Site internet :

- *https://csunplugged.org*, consulté le 8 avril 2019 ;
- *https://www.cnrtl.fr/definition/hardware*, consulté le 14 mai 2019 ;
- *https://www.futura-sciences.com/tech/definitions/informatique-cartemere-1870/*, consulté le 14 mai 2019 ;
- *https://fr.wikipedia.org/wiki/Bloc_d%27alimentation*, consulté le 14 mai 2019 ;
- *https://cours-informatique-gratuit.fr/dictionnaire/processeur/*, consulté le 14 mai 2019 ;
- *https://www.digital-dynamics.fr/FR/materiel-informatique/hardware-andperipherals/types-de-connectique-informatique.html*, consulté le 14 mai 2019 ;
- *https://www.webmarketing-com.com/2012/11/06/16580-quels-sont-les-9-formats-differents-pour-une-image*, consulté le 15 mai 2019 ;
- *http://glossaire.infowebmaster.fr/html/*, consulté le 15 mai 2019 ;
- *https://members.loria.fr/Mduflot/files/med/routage.html*, consulté le 16 mai 2019.

Séquence de cours :

- Initiation à la programmation, SI², D. Boels, H. de Groot, D. Lemoine, O. Goletti.

Fiches activités :

- Fiche « ordi, étrange animal numérique » le Pass ;
- Fiche « le routage élastique », *https://members.loria.fr/MDuflot/files/med/doc/Routage/routageelastique.pdf*, Marie Duflot-Kremer.

Ont collaboré à ce document :

BALANCIER Pascal - **Digital Wallonia**

CLAUSSE Nathalie - **PASS**

COLAS Céline - **Kodo Wallonie**

DARO Sabine - **ASBL Hypothèse**

GILLON Valérie - **CoderDojo**

GOLETTI Olivier - **SI²**

HANOTTE Pierre - **Interface3Namur**

JACQUES Anthony - **Technobel**

LEROY Marie - **ASBL Hypothèse**

MOREAU Simon - **Interface3Namur**

NOEL Cécile - **ASBL Hypothèse**

PONCIN Chantal - **SI²**

Septembre 2019