



LE RÉFÉRENTIEL DES COMPÉTENCES NUMÉRIQUES EXEMPLE D'ACTIVITÉS

NOM DE L'ACTIVITÉ	<i>Apprendre à programmer</i>
DISCIPLINE(S)	<i>Mathématiques</i>
CLASSE OU NIVEAU	<i>3ème</i>
DOMAINE(S)	<i>3. Création de contenu</i>
COMPÉTENCE(S)	<i>3.4 Programmer</i>
NIVEAU DU CADRE DE RÉFÉRENCE DES COMPÉTENCES NUMÉRIQUE	<i>Niveau 1 et 2</i>
OUTILS UTILISÉS DOMAINE(S)	<i>Document web + Logiciel libre sous Windows « Scratch »</i>
DESCRIPTION DE L'ACTIVITÉ	<p><i>Tout le monde compte en base 10, mais comment fonctionne notre mode de comptage ?</i></p> <p><i>Nous comptons en base 10 (système décimal), système issu de nos 10 doigts. Il en découle que :</i></p> <ul style="list-style-type: none"><i>• Il existe 10 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9</i><i>• Avec ces chiffres on peut compter jusqu'à 9 (la plus haute valeur des chiffres)</i> <p><i>Pour aller au-delà de 9, il faut changer de rang. Cela signifie que si le rang des unités est plein, on commence le rang des dizaines et on remet les unités à zéro, et ainsi de suite.</i></p> <p><i>Par exemple, arrivé à 39, le rang des unités est « plein ». On ajoute donc une dizaine et on remet à zéro le rang des unités : on obtient donc à 40.</i></p> <p><i>On voit que une centaine vaut 10 dizaines et que une dizaines vaut 10 unités.</i></p> <p><i>Plus mathématiquement, un rang est égale au précédent multiplié 10.</i></p> <p><i>De cette manière, le nombre $73 = 70 + 3$ peut aussi écrire : $73 = 7 \times 10^1 + 3 \times 10^0$</i></p> <p><i>C'est la décomposition de 73 en puissances de 10 (unités, dizaines, etc.).</i></p>

On peut décomposer chaque nombre en puissances de 10 successives.

Par exemple : $43207 = 4 \times 10^4 + 3 \times 10^3 + 2 \times 10^2 + 0 \times 10^1 + 7 \times 10^0$

LES ÉTAPES DE L'ACTIVITÉ

Le binaire.

Présentation.

Le binaire est le mode de comptage non plus en base 10 mais **en base 2**. Il est utilisé par les ordinateurs, car les machines ne peuvent comparer que deux valeurs : des 1 et des 0.

En binaire on emploie le mot « bit » (contraction de « binary-digit », signifiant simplement « rang binaire »). Par exemple, le nombre en base 2 « 10111 » s'étale sur 5 bit.

En binaire, **chaque rang ne peut prendre que deux valeurs** (il pouvait en prendre dix en décimal). Donc, dès que le rang atteint sa deuxième soit sa plus haute valeur, on change de rang. **En binaire, un rang commence à 0 et se termine à 1.**

Ainsi :

0 en binaire s'écrit 0

1 s'écrit 1

2 s'écrit 1 0 (puisque le « chiffre » 2 n'existe pas..., on passe au rang suivant)

3 est égal 1 1

4 s'écrit 1 0 0

5 s'écrit 1 0 1

6 s'écrit 1 1 0

7 s'écrit 1 1 1

8 s'écrit 1 0 0 0

9 s'écrit 1 0 0 1

10 s'écrit 1 0 1 0

etc.

Il suffit d'appliquer une règle : entamer le rang suivant quand celui en cours est complet.

Conversion du décimal en binaire.

Pour le moment, on a compté jusqu'à dix, maintenant **convertissons**.

Méthode 1 : les puissances de 2

Pour y arriver, on doit décomposer notre nombre en puissances de 2. C'est le même principe que la décomposition en puissances de dix, sauf que l'on ne décompose pas en milliers, centaines et dizaines, mais en puissances de deux ; qui sont : 1, 2, 4, 8, 16, 32, 64 ..., 512, 1024, etc.

(une valeur est égale à la précédente multipliée par 2).

Ainsi, si l'on prend l'exemple du nombre 26, on obtient la décomposition suivante :

$$26 = 16 + 8 + 2$$

Il suffit ensuite de remplacer ces nombres par les puissances de 2 correspondantes.

$$26 = 16 + 8 + 2$$

$$26 = 1 \times 16 + 1 \times 8 + 1 \times 2$$

$$26 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1$$

(on écrit les coefficients sous forme de puissances de 2)

$$26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

(on ajoute les puissances de 2 qui manquent)

$$26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

(voyez les puissances de 2 qui sont toutes là)

$$26 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

(en orange : notre nombre en binaire !)

Il est important de ne pas oublier les puissances dont les coefficients sont zéro.

Finalement, pour obtenir le nombre 26 en binaire, il suffit de mettre les coefficients qui sont devant les puissances de 2 à la suite. On obtient : 11010.

On écrit : $(26)_{dec} = (11010)_{bin}$

Méthode 2 : les divisions euclidiennes par 2

Cette méthode est préférable pour les grands nombres et est plus facile à utiliser en programmation (il est facile d'en faire un algorithme).

- On a notre nombre en décimal.
- On le divise par 2 et on note le reste de la division (c'est soit un 1 soit un 0).
- On refait la même chose avec le quotient précédent, et on met de nouveau le reste de côté.
- On réitère la division, et ce jusqu'à ce que le quotient soit 0.
- Le nombre en binaire apparaît : le premier à placer est le dernier reste obtenu. Ensuite, on remonte en plaçant les restes que l'on avait. On les place à droite du premier 1.

Exemple : (avec le nombre 164 en décimal)

$$\bullet 164 \div 2 = 82 + 0$$

$$\bullet 82 \div 2 = 41 + 0$$

$$\bullet 41 \div 2 = 20 + 1$$

$$\bullet 20 \div 2 = 10 + 0$$

$$\bullet 10 \div 2 = 5 + 0$$

- $5 \div 2 = 2 + 1$
- $2 \div 2 = 1 + 0$
- $1 \div 2 = 0 + 1$

On voit apparaître notre nombre binaire en rouge : (il faut le lire de bas en haut)

1010 0100

Voir ici pour des compléments : <https://lehollandaisvolant.net/tuto/bin/>

1) S'exercer en convertissant en binaire les nombres suivants :

a) à l'aide de la méthode 2.2.1 :

14 ; 25 ; 120

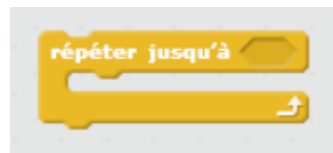
b) à l'aide de la méthode 2.2.2 : (pour la comprendre et la mémoriser)

7 ; 22 ; 120 ; 1000

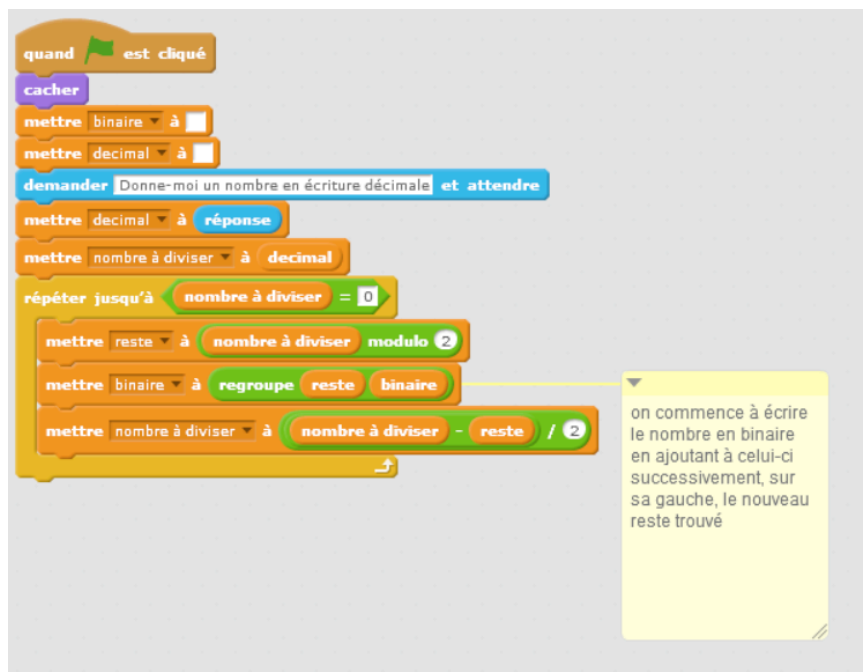
c) Interrogez-vous entre élèves !

2) Programmer avec Scratch un convertisseur « décimal VERS binaire » en vous inspirant de la méthode 2.2.2.

Vous aurez sans doute besoin de l'instruction suivante :



Une programmation possible : (méthode des divisions euclidiennes par 2)



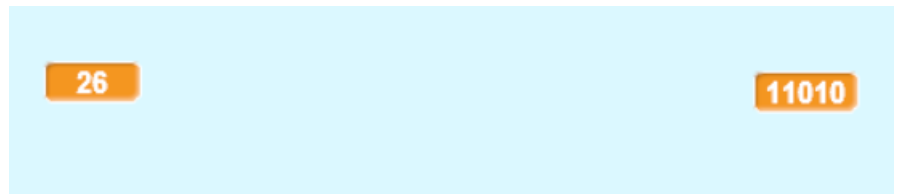
Avec 26 :

A screenshot of a Scratch input field. The field has a light blue border and contains the text "Donne-moi un nombre en écriture décimale". Below the text, the number "26" is entered. A blue checkmark icon is visible on the right side of the field, indicating that the input is valid.

Donne-moi un nombre en écriture décimale

26

On obtient :

A screenshot of a Scratch output area. It features two orange rectangular boxes. The left box contains the number "26", and the right box contains the binary number "11010".

26

11010

**CONSIGNES DONNÉES
AUX ÉLÈVES**

Programmer avec Scratch un convertisseur « décimal VERS binaire