

Penser en binaire



Il n'y a que 10 types de personnes dans le monde. Ceux qui comprennent le binaire et ceux qui ne le comprennent pas!

Pour représenter toutes les données, les ordinateurs utilisent des **séquences binaires** - c'est-à-dire des séquences de 0 et de 1 -. Ca veut dire que nos photographies, vidéos, messages SMS, applications mobiles, documents Word et tous les autres objets numériques sont représentés par des séquences de **bits**.

Dans ce cours , nous allons nous intéresser au système de nombres binaires, c'est-à-dire à la représentation des nombres par des séquences de 0 et de 1. Le système de numération binaire est très similaire à notre système décimal. Mais alors que le système décimal utilise 10 chiffres - 0,1,2,3,4,5,6,7,8,9 pour représenter des nombres, le système binaire n'utilise que deux chiffres : 0 et 1

Nombres binaires (Slide Presentation)

La raison pour laquelle les ordinateurs utilisent le système binaire est que ses deux chiffres, 0 et 1, sont faciles à représenter en électronique, le 0 pouvant être représenté par "off ou éteint" et le 1 par "on ou allumé". Ou alors le 0 par une basse tension et le 1 par une tension élevée.

Comme vous l'avez vu, un bit est la plus petite unité d'information. Un bit peut être 0 ou 1.

Le système binaire utilise seulement 2 chiffres, alors que le système décimal utilise 10 chiffres, donc le système de nombres binaires est plus simple que le système décimal.

Ce qui est étonnant, c'est qu'en combinant des 0 et des 1, nous pouvons quand même représenter tous les nombres et toutes les données ou informations que les ordinateurs utilisent et traitent.

Video: Compter en binaire

Avez-vous compris la blague du début? Si non, vous la comprendrez certainement après cette vidéo de [Computer Science Unplugged](#), qui illustre le fonctionnement du système binaire.

Conversion du binaire au décimal

Comme vous l'avez vu dans la vidéo, le système de numération binaire est un système de numération de position. La valeur d'un chiffre particulier dépend de sa place. Selon sa position à partir de la droite de la droite le 1 vaudra (en décimal) 1, 2, 4, 8, 16, etc. Comparez cela avec les valeurs aux mêmes positions pour le système décimal : 1, 10, 100, etc.

Pour déterminer la valeur décimale d'un nombre binaire, nous additionnons les valeurs correspondant à toutes les positions contenant un 1.

Voici un algorithme pour convertir un nombre binaire en décimal:

1. Ecrivez le nombre binaire en laissant un espace entre les chiffres. Par exemple:

1 0 1 0 1

2. En commençant par le côté droit du nombre, écrivez la valeur de position de chaque chiffre binaire dans le nombre situé au-dessus des chiffres, en commençant par 1, 2, 4, 8 et ainsi de suite, en doublant à chaque fois la valeur de position. Par exemple :

16 8 4 2 1 (valeurs de position)
1 0 1 0 1

3. Additionnez les valeurs de position de chaque chiffre qui est 1. Pour cet exemple, cela donnerait $16 + 4 + 1$, qui est 21.

16 8 4 2 1 (Place values)
1 0 1 0 1 (Decimal value: $16 + 4 + 1 = 21$)

Essayez : utilisez [cette application interactive](#) pour convertir du binaire au décimal.

Convertir un nombre Décimal en nombre Binaire

Vous pouvez utiliser un algorithme similaire pour convertir des nombres décimaux en nombres binaires. Par exemple, voilà l'algorithme appliqué au nombre décimal 25:

1. En commençant par la droite et en allant vers la gauche, nous écrivons la valeur décimale des digits binaires 1, puis 2, puis 4, puis 8, etc. Ce sont les puissances de 2

32 16 8 4 2 1

2. On cherche la plus grande valeur qui soit inférieure ou égale au nombre décimal à convertir en binaire et on écrit 1 en dessous. Par exemple, pour 25, la valeur la plus grande est 16 :

32	16	8	4	2	1
					1

3. On soustrait cette valeur au nombre décimal à convertir. Donc en retirant 16 à 25 on obtient 9. On retient ce reste comme nouveau nombre décimal à convertir en binaire.

4. On répète ensuite les étapes 2 et 3 jusqu'à obtenir 0.
 Donc pour cet exemple avec 25 au départ, on va écrire un 1 sous le 8 (après celui du 16). Il va rester 1, on va donc écrire un 1 sous le 1. Le reste est 0, donc on a fini.

32	16	8	4	2	1
		1			1

5. Sur la 2^o ligne, on écrit 0 aux endroits qui sont vides et ça nous donne le nombre binaire. La conversion en binaire du nombre décimal 25 est donc 11001.

32	16	8	4	2	1	
		1	1	0	0	1

Essayez: Vous pouvez utiliser l'appli Web [suivante](#) pour passer d'un nombre décimal à un nombre binaire.

Bases et systèmes de numération Positionnels

Notre système décimal est à **base 10**. Ça veut dire qu'il utilise 10 chiffres, de 0 à 9, et la valeur de chaque chiffre dans un nombre dépend de sa position, en puissances de 10 depuis la droite. La valeur de ce chiffre est obtenue en le multipliant par 1, une dizaine, une centaine, un millier, etc. On commence par un et à chaque décalage vers la gauche la valeur est multipliée par 10.

Dans le système à **base 2**, il n'y a que 2 chiffres ou digits, 0 and 1. Leur est multipliée par 2 à chaque déplacement vers la gauche. Ce sont les puissances de 2 : 1, 2, 4, 8, etc. Pour chaque décalage vers la gauche on multiplie par 2 au lieu de 10.

Un des problèmes avec les nombres binaires, c'est qu'ils prennent beaucoup de place quand on les écrit. Par exemple, pour écrire 1 million en binaire, il faut 20 chiffres ou digits binaires:

11110100001001000000.

Nombres en octal

Because binary numbers like 11110100001001000000 are hard to read, computer scientists often represent them in the **octal (base 8)** number system. The octal system has 8 digits, 0 through 7, and its place values are powers of 8. So it has a 1s place, an 8s place, a 64s place, a 512s place, and so on. The reason that octal is so convenient for representing binary numbers is that all 7 octal digits can be represented as 3-digit binary numbers. You already know how to count to 7 in binary:

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

To represent a binary number in octal is easy, just break it up into groups of 3, and represent each group of 3 with an octal digit. For example, in octal we would represent 1 million as follows:

```
11 110 100 001 001 000 000
 3  6  4  1  1  0  0
```

So, in octal, 1 million is represented as 3641100.

Nombres Hexadécimaux

Octal is a more economical way to represent large binary numbers. Another common number system used by computer scientists is the **hexadecimal (base 16)** system. The hexadecimal (or **hex**) system has 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, where A through F represent decimal 10 through 15. And its places are powers of 16. So it has a 1s place, a 16s place, a 256s place, 4096s place, and so on.

The hexadecimal system is even more economical than the octal. But to use it we have to learn the hex equivalents of the first 16 binary numbers. You already know how to represent the first 16 binary numbers, each of which take 4 bits, which, by the way, is called a **nibble**:

Hex	Binary	Hex	Binary	Hex	Binary	Hex	Binary
0	0000	4	0100	8	1000	C	1100
1	0001	5	0101	9	1001	D	1101
2	0010	6	0110	A	1010	E	1110
3	0011	7	1110	B	1011	F	1111

Then, to represent 1 million in hex, we can break it up into groups of 4 bits and represent each nibble by a hex digit. For example, in hex we would represent 1 million as follows:

```
1111 0100 0010 0100 0000
  F   4   2   4   0
```

So 1 million in hex is F4240.

Try it: Use [this interactive web app](#) to convert some simple binary numbers to hexadecimal.

Abstraction

Perhaps you can see that there is a *pattern* that all of these positional number systems follow, which can be summarized as follows:

- The **base** determines the number of digits.
- The **place values**, from right to left, start with the 1s place.
- To find the next place value to the left, we multiply the current place value by the base.
- To calculate the numbers value, we multiply the value of the digit by the value of its place.

Once you see the abstract pattern, you should be able to figure out numbers in just about any base. For example, what are the digits in a base-5 system? And what value would 1432 be in base-5? Try it, you can do it! (Answers at the bottom of the page.)

Pour en savoir plus?

If you'd like to learn more about the relationship between binary and decimal numbers, take a look at this [Khan Academy](#) video:

Pour votre Portfolio

In your portfolio, create a new page named **Binary Numbers** under the *Reflections* category of your portfolio and answer the following questions:

1. Figure out what decimal value is represented by the following binary number 0011 1010 0011
2. Represent the decimal value 517 as a binary number.
3. The binary number system is *base 2* and has 2 digits. The decimal number system is *base 10* and has 10 digits. The *octal system* is *base 8*. How many digits does it have? What are they, starting at 0.
4. **Challenging:** Suppose the number 523 is a base-8 octal number. What would its value be in decimal? In binary?

(The digits in the base 5 system are 0 through 4 and the base-5 number 1432 would be $2 \times 1 + 3 \times 5 + 4 \times 25 + 1 \times 125 = 2 + 15 + 100 + 125 = 242$ in decimal.)