

Dénombrement : le cours

Dénombrer, c'est compter sans énumérer.

- Combien de codes possibles ?
- Combien de mains de cartes ?
- Combien de comités ?
- Combien de chemins ?

Ce chapitre est la base des probabilités en situation d'équiprobabilité : $P = \frac{\text{cas favorables}}{\text{cas possibles}}$. Il n'y a pas de formule magique : il faut **choisir le bon modèle**.

Dans tout le cours, tu verras des astuces , des pièges et un peu d'humour .

Comment utiliser ce cours

- Lis une partie, puis fais tout de suite la fiche correspondante.
- Devant un problème, pose-toi toujours **deux questions** : « L'ordre compte-t-il ? » « Peut-on répéter un élément ? » Le tableau de la partie 7 en découle.
- Entraîne-toi à **rédigier le raisonnement en français** : « on choisit d'abord... puis... ». C'est ce que le correcteur veut lire, pas seulement le résultat.

1. Ensembles et cardinal

Le **cardinal** d'un ensemble fini E , noté $|E|$ ou $\text{Card}(E)$, est son nombre d'éléments. Soient A et B deux parties d'un ensemble E de cardinal n .

Réunion :

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Si A et B sont **disjoints** ($A \cap B = \emptyset$), alors $|A \cup B| = |A| + |B|$.

Complémentaire : $|\overline{A}| = n - |A|$

Produit cartésien : $|A \times B| = |A| \times |B|$

Exemple. Dans une classe de 30 élèves :

- 18 font de l'anglais ;
- 15 font de l'espagnol ;
- 5 font les deux.

Élèves qui font au moins une des deux langues : $18 + 15 - 5 = 28$.

Élèves qui n'en font aucune : $30 - 28 = 2$.

2. Les principes additif et multiplicatif

Principe additif. Si un ensemble se partage en cas **disjoints**, on **additionne** les nombres de possibilités de chaque cas.

Principe multiplicatif. On effectue **successivement** des choix :

- le premier choix offre n_1 possibilités ;
- le deuxième en offre n_2 , quel que soit le premier ;
- et ainsi de suite jusqu'au dernier.

Alors le nombre total de possibilités est :

$$n_1 \times n_2 \times \cdots \times n_p$$

Un **arbre de dénombrement** illustre le principe multiplicatif : chaque branche est un choix ; chaque chemin est une possibilité.

Exemple. Un menu se compose de :

- une entrée (3 choix) ;
- un plat (4 choix) ;
- un dessert (2 choix).

Nombre de menus : $3 \times 4 \times 2 = 24$.

Astuce : « et » se traduit par $\times$ (choix successifs). « ou » se traduit par $+$ (cas disjoints).

3. Les k -uplets (avec répétition possible)

Un **k -uplet** d'éléments d'un ensemble E de cardinal n est une suite ordonnée $(x_1, x_2, \dots, x_k)$ où les éléments peuvent se répéter. Il y a n^k k -uplets.

Cela modélise les **tirages successifs avec remise** : codes, mots, lancers de dés ou de pièces.

Exemples.

Un digicode à 4 chiffres : $10^4 = 10\,000$ codes.

Un mot de 5 lettres (sans souci du sens) : $26^5 = 11\,881\,376$ mots.

Applications d'un ensemble à p éléments vers un ensemble à n éléments : n^p (on choisit une image pour chaque élément).

4. Permutations et k -uplets d'éléments distincts

Factorielle : $n! = n \times (n-1) \times \cdots \times 2 \times 1$, avec $0! = 1$.

Permutations : il y a $n!$ façons d'ordonner n éléments distincts.

k -uplets d'éléments distincts (arrangements) : le nombre de k -uplets d'éléments **deux à deux distincts** d'un ensemble à n éléments ($k \leq n$) est :

$$n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!}$$

Cela modélise les **tirages successifs sans remise** : podiums, bureaux (président, secrétaire, trésorier), mots à lettres distinctes.

Exemples.

Un podium (1er, 2e, 3e) parmi 8 coureurs : $8 \times 7 \times 6 = 336$.

Six personnes dans une file : $6! = 720$.

Piège : $n!$ grandit très vite ($10! = 3\,628\,800$). Pour $\frac{n!}{(n-k)!}$, on ne garde que les k premiers facteurs, sans développer les factorielles.

5. Parties d'un ensemble et combinaisons

Un ensemble E à n éléments a 2^n **parties**. Pour chaque élément, on choisit « dedans » ou « dehors » : 2 choix, n fois.

Une **combinaison** de k éléments de E est une **partie** de E à k éléments :

- l'ordre ne compte pas ;
- il n'y a pas de répétition.

Leur nombre se note $\binom{n}{k}$ et se lit « k parmi n ». Sur la calculatrice, c'est la touche **nCr**. En Python, c'est `comb(n, k)`.

Formule :

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n \times (n-1) \times \cdots \times (n-k+1)}{k!}$$

Pourquoi ? Il y a $\frac{n!}{(n-k)!}$ k -uplets distincts. Chaque partie à k éléments correspond à $k!$ de ces k -uplets (ses $k!$ ordres possibles). On divise donc par $k!$.

Valeurs et propriétés :

- $\binom{n}{0} = 1$
- $\binom{n}{1} = n$
- $\binom{n}{2} = \frac{n(n-1)}{2}$
- $\binom{n}{n} = 1$

Symétrie : $\binom{n}{k} = \binom{n}{n-k}$. Choisir les k éléments retenus revient à choisir les $n-k$ éléments écartés.

Relation de Pascal :

$$\binom{n}{k} + \binom{n}{k+1} = \binom{n+1}{k+1}$$

Le triangle de Pascal. La ligne n contient $\binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n}$. Chaque terme est la somme des deux termes situés au-dessus de lui.

- ligne 0 : 1
- ligne 1 : 1 1
- ligne 2 : 1 2 1
- ligne 3 : 1 3 3 1
- ligne 4 : 1 4 6 4 1
- ligne 5 : 1 5 10 10 5 1
- ligne 6 : 1 6 15 20 15 6 1

Astuces :

- Pour calculer $\binom{n}{k}$ à la main, utilise la symétrie pour prendre le **plus petit** des deux nombres.
 $\binom{12}{10} = \binom{12}{2} = 66$.
- Tirage **simultané** (sans ordre, sans remise) : pense **combinaison**.
- Comité, main de cartes, échantillon : **combinaison**.

6. Somme des coefficients et formule du binôme

Somme des $\binom{n}{k}$:

$$\binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{n} = 2^n$$

Justification : chaque partie de E a un cardinal k compris entre 0 et n . Le total des parties est donc la somme des $\binom{n}{k}$, et il vaut 2^n .

Formule du binôme de Newton : pour tous réels a, b et tout entier $n \geq 0$:

$$(a+b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

Par exemple :

$$(a+b)^4 = a^4 + 4a^3b + 6a^2b^2 + 4ab^3 + b^4$$

Les coefficients sont ceux de la ligne 4 du triangle.

Deux conséquences :

- Avec $a = b = 1$, on retrouve la somme 2^n .

- Avec $a = 1$ et $b = -1$, pour $n \geq 1$: $\sum_{k=0}^n (-1)^k \binom{n}{k} = 0$. Les parties de cardinal pair et celles de cardinal impair sont aussi nombreuses : 2^{n-1} chacune.

7. Choisir le bon modèle

- **L'ordre compte et on peut répéter** : k -uplets, n^k . Tirages successifs avec remise, codes.
- **L'ordre compte et on ne répète pas** : $\frac{n!}{(n-k)!}$. Tirages successifs sans remise, podiums, bureaux. Si on prend tous les éléments : $n!$ (permutations).
- **L'ordre ne compte pas et on ne répète pas** : $\binom{n}{k}$. Tirage simultané, comités, mains de cartes.

Les grandes stratégies :

Choisir successivement (principe multiplicatif).

Décide dans quel ordre choisir, en commençant par la contrainte la plus forte.

Par exemple, le premier chiffre d'un nombre, qui ne peut pas être 0.

Séparer en cas disjoints (principe additif). Par exemple : « exactement 2 rouges », puis « exactement 3 rouges ».

Passer au complémentaire pour « au moins un ». Nombre total – nombre de cas où il n'y en a aucun.

Choisir des positions pour les anagrammes.

ANANAS a 3 A, 2 N et 1 S.

Places des A : $\binom{6}{3} = 20$.

Places des N parmi les 3 restantes : $\binom{3}{2} = 3$.

Le S prend la dernière place.

Total : $20 \times 3 = 60$.

Compter des chemins sur un quadrillage. De $(0,0)$ à (a,b) , avec des pas vers la droite et vers le haut. On choisit lesquels des $a+b$ pas sont vers le haut : $\binom{a+b}{b}$.

Pièges :

- **Confondre ordre et non-ordre** : compter les comités $\{A, B\}$ et $\{B, A\}$ comme deux comités différents.
- **Additionner au lieu de multiplier** (ou l'inverse).
- **Compter deux fois** le même cas quand les cas ne sont pas disjoints.
- Oublier les contraintes (un nombre ne commence pas par 0).

Si ton résultat dépasse le nombre d'atomes de l'univers pour un digicode, c'est peut-être que tu as multiplié quand il fallait additionner.

8. Dénombrement et probabilités

Quand toutes les issues sont **équiprobables** :

$$P(A) = \frac{|A|}{|\Omega|}$$

c'est-à-dire $\frac{\text{cas favorables}}{\text{cas possibles}}$. Pour un tirage au hasard :

- on dénombre Ω au dénominateur ;
- on dénombre A au numérateur ;
- avec le **même modèle** (avec ou sans ordre).

9. Python

```
from math import comb, factorial, perm

factorial(6)    # 720
perm(8, 3)     # 336 (k-uplets d'éléments distincts)
comb(10, 3)    # 120

from itertools import product, permutations, combinations

len(list(product(range(10), repeat=3))) # 1000 codes
len(list(permutations(range(5), 3)))    # 60
len(list(combinations(range(5), 3)))    # 10
```

Énumérer par Python permet de **vérifier** un dénombrement pour de petites valeurs.

10. Les 10 pépites du chapitre

1. **Deux questions** : l'ordre compte-t-il ? peut-on répéter ?
2. « **Et** » = $\times$, « **ou** » = $+$.
3. « Au moins un » : **passé au complémentaire**.
4. Commence par la **contrainte la plus forte** quand tu choisis successivement.
5. Tirage simultané, comité, main de cartes : **combinaison**.
6. Podium, bureau, mot à lettres distinctes : **k -uplet d'éléments distincts**.
7. Utilise la **symétrie** $\binom{n}{k} = \binom{n}{n-k}$ pour simplifier.
8. Un ensemble à n éléments a **2^n parties**.
9. Pour une probabilité, garde le **même modèle** au numérateur et au dénominateur.
10. **Vérifie avec de petites valeurs** ($n = 3$ ou 4) ou avec Python.

Complément du programme : parties, mots binaires et n -uplets

Un ensemble E à n éléments a **2^n parties**. On peut le voir de trois façons équivalentes.

- Une **partie** de E ;
- un **mot binaire** de longueur n (le bit i vaut 1 si le i -ème élément est dans la partie, 0 sinon) ;
- un **n -uplet de $\{0; 1\}$** , c'est-à-dire un élément de $\{0; 1\}^n$.

Exemple. Pour $E = \{a; b; c\}$, la partie $\{a; c\}$ correspond au mot 101.

Mot	000	001	010	011	100	101	110	111
Partie	$\emptyset$	$\{c\}$	$\{b\}$	$\{b; c\}$	$\{a\}$	$\{a; c\}$	$\{a; b\}$	E

Il y a bien $2^3 = 8$ parties. C'est aussi le nombre d'**issues** d'un schéma de Bernoulli à 3 épreuves (chaque épreuve : succès 1 ou échec 0).

```
n = 3
for k in range(2 ** n):
    print(format(k, '03b'))    # affiche les 8 mots 000, 001, ..., 111
```

Complément du programme : mélanger une liste avec Python

Pour tirer une **permutation au hasard**, on utilise ``shuffle`` :

```
from random import shuffle

liste = [1, 2, 3, 4, 5]
shuffle(liste)
print(liste)    # par exemple [3, 1, 5, 2, 4]
```

11. Ce qui peut tomber au bac

- Des **principes de dénombrement**, k -uplets, permutations, arrangements (fiche 1).
- Des **combinaisons**, tirages simultanés, mains, comités, chemins, anagrammes (fiche 2).
- Le **triangle de Pascal**, le **binôme de Newton** et des démonstrations par dénombrement (fiche 3).
- Des **problèmes de type bac** mêlant dénombrement et probabilités, avec Python (fiche 4 et exercices type bac).

Tu as maintenant le cours. Passe aux quatre fiches, aux exercices type bac, puis au Guide de rédaction du chapitre.