

De Turing à l'IA

D'où vient l'intelligence artificielle ? D'une longue histoire de machines : la carte perforée, la **machine universelle** de Turing, l'ordinateur, puis l'**apprentissage automatique** - avec ses promesses et ses pièges.



1 Trois parties, une vraie progression

Cours clair, repères à retenir, méthodes pas à pas, QCM de tête, exercices corrigés. On part des machines et de Turing, on découvre l'apprentissage, on termine sur les données et l'éthique.

2 « Les erreurs qui coûtent des points »

Dans chaque partie, les confusions à éviter dans le chapitre, en X / ✓, avec le réflexe qui les évite. Et le barème d'une question type, décrypté sur un exemple.

3 Rien que le programme officiel

Les savoir-faire exigibles du chapitre, et pas un de plus. **Un chapitre pour comprendre l'IA - ce qu'elle sait faire, et ce qu'il faut en questionner.**

4 L'essentiel pour réviser

Le formulaire du chapitre, écrit en phrases complètes, et la page des plans de réponse : pour chaque question courante, ce que ta copie doit contenir.

3

parties
progressives

12

exercices corrigés
pas à pas

12

repères
à retenir

7

méthodes
pas à pas

Conforme au programme d'**enseignement scientifique de terminale générale** - B.O. n° 25 du 22 juin 2023, thème 3 « Une histoire du vivant », partie **3.5 « De la machine de Turing à l'intelligence artificielle »**. Un test de positionnement en page 3 t'indique par quelle partie commencer.

« Une IA ne vaut pas mieux
que les données sur lesquelles elle a appris. »

Génies des Sciences

Méthodes & entraînement · Lycée

www.geniesdessciences.com

3 Par quelle partie commencer ?

CORRIGÉ FERMÉ

Seize questions sur les trois parties. Réponds **sur une feuille**, sans regarder les réponses. Ce test ne te note pas : il te dit seulement **par où commencer**.

01 Avant Turing, une machine à information faisait :

- a** une tâche prédéterminée **b** n'importe quel programme

02 Le concept de machine universelle est dû à :

- a** Turing **b** Jacquard

03 Un ordinateur comporte au minimum :

- a** un processeur et une mémoire vive **b** un écran et une souris

04 Un programme est aussi :

- a** un objet physique **b** une donnée

05 Numériser, c'est coder en :

- a** 0 et 1 **b** lettres

06 Un bogue, c'est :

- a** un composant **b** une erreur de programme

07 L'intelligence artificielle est née en :

- a** 1956 **b** 2007

08 L'IA d'aujourd'hui repose surtout sur :

- a** des règles écrites à la main **b** l'apprentissage automatique

09 Dans l'apprentissage automatique, la machine :

- a** s'entraîne sur des données **b** suit un programme figé

10 Une courbe de tendance sert à :

- a** trier des fichiers **b** estimer une valeur

11 L'apprentissage repère des :

- a** tendances (corrélations) **b** intentions

12 La qualité des résultats dépend surtout :

- a** des données **b** de la marque de l'ordinateur

13 Sur des données biaisées, une IA :

- a** amplifie les biais **b** les corrige seule

14 Deux grandeurs corrélées sont-elles forcément liées par une cause ?

- a** oui **b** non

15 Un test positif signifie-t-il toujours « malade » ?

- a** oui **b** non

16 Des usages de l'IA posent-ils des questions éthiques ?

- a** oui **b** non

QUESTIONS	PARTIE	SI TU EN RATES UNE...
1 à 6	1	les machines, Turing, la numérisation
7 à 11	2	l'apprentissage automatique
12 à 16	3	les données, les biais, l'esprit critique

✓ Où sont les réponses

Elles ne sont pas sur cette page, pour que tu répondes vraiment : tu les trouveras en tête des corrigés, **page 26**. Réponds d'abord aux seize questions sur ta feuille, puis va les comparer d'un seul coup.

★ Comment lire ton résultat

14 à 16 : enchaîne les trois parties dans l'ordre.

9 à 13 : commence par les parties où tu as raté une question, puis reprends les autres.

8 ou moins : fais les trois parties dans l'ordre. Le pack est construit pour ce parcours complet.

? D'où viennent ces questions

Chaque question porte sur une notion ou un savoir-faire du programme officiel, et sur rien d'autre.



1 octet = 8 bits ; en ASCII non compressé, 1 caractère = 1 octet ; 1 ko = 1000 octets

Un programme de plusieurs centaines de millions de lignes contient très probablement des erreurs, appelées bogues : elles rendent le comportement du programme inattendu, parfois avec des conséquences graves.

5 Les méthodes, pas à pas

MÉTHODE

1 Calculer la taille d'un texte en octets

- 1 **Je compte les caractères**
du texte, espaces et ponctuation compris.
- 2 **Je multiplie par 1 octet**
par caractère, puisque le codage est l'ASCII non compressé.
- 3 **Je divise par 1000 octets/ko**
quand l'énoncé demande la taille en kilo-octets.

L'EXEMPLE RÉDIGÉ

Une page de 2000 caractères pèse 2000 caractères ×
 $1 \text{ octet/caractère} = 2000 \text{ octets}$, soit $\frac{2000 \text{ octets}}{1000 \text{ octets/ko}} =$
2 ko.

2 Situer un ordre de grandeur

- 1 **Je repère la nature**
du fichier car texte, image, son et vidéo ne pèsent pas du tout la même chose.
- 2 **Je classe**
du plus léger (le texte) au plus lourd (la vidéo).
- 3 **Je raisonne en facteurs 1000**
pour passer des octets aux ko, puis aux Mo, puis aux Go.

L'EXEMPLE RÉDIGÉ

Une photo de 3 Mo pèse 3 000 000 octets, une page de texte 2000 octets. Le rapport vaut
 $\frac{3\,000\,000 \text{ octets}}{2000 \text{ octets}} = 1500$: la photo pèse **environ mille fois** plus, c'est-à-dire un ordre de grandeur de plus.

3 Corriger un algorithme bogué

- 1 **Je dis ce que le programme doit produire**
sur un petit cas dont je sais calculer le résultat à la main.
- 2 **Je déroule le programme ligne à ligne**
sur un petit cas, en notant la valeur de chaque variable.
- 3 **Je repère la première ligne**
où la valeur notée s'écarte de la valeur attendue, et je corrige cette ligne seule.

L'EXEMPLE RÉDIGÉ

Ce programme doit additionner les entiers de 1 à 5, donc afficher 15.

```
1 somme = 0
2 for i in range(1, 6):
3     somme = i
4 print(somme)
```

En déroulant, `somme` vaut 1, puis 2, 3, 4, 5 : la ligne 3 **remplace** la somme au lieu de l'augmenter. La correction est **`somme = somme + i`**, et le programme affiche alors 15.

1 Croire que Turing a construit le premier ordinateur

X « Turing a inventé l'ordinateur. » · ✓ Turing a proposé le **concept** de machine universelle (1936) ; les premiers ordinateurs sont venus une dizaine d'années plus tard.

Le réflexe : l'idée d'abord, la machine ensuite.

2 Croire qu'un programme n'est pas une donnée

X « Les données et les programmes, c'est tout différent. » · ✓ Un programme est lui aussi une **donnée** : on le stocke, on le copie, on le transmet.

Le réflexe : le programme est une donnée.

3 Confondre bit et octet

X « Un caractère fait 1 bit. » · ✓ Un **octet = 8 bits** ; un caractère ASCII occupe 1 octet, donc 8 bits.

Le réflexe : octet = 8 bits.

4 Confondre l'algorithme et le programme

X « Un algorithme, c'est du Python. » · ✓ L'algorithme est la **méthode**, qu'on peut écrire en français ; le programme est cette méthode traduite dans un **langage** que la machine exécute.

Le réflexe : la méthode d'abord, le langage ensuite.

7 Le QCM express

DE TÊTE, SANS CALCULATRICE

- | | |
|---|-------------------------------------|
| 1. Avant Turing, une machine faisait ? | 5. Numériser, c'est coder en ? |
| 2. Qui propose la machine universelle ? | 6. Un programme est-il une donnée ? |
| 3. Les premiers ordinateurs datent des années ? | 7. Un caractère ASCII pèse ? |
| 4. Un ordinateur comporte au minimum ? | 8. Un algorithme, c'est ? |

Cache cette ligne pendant que tu réponds. **Réponses** : 1. une ou quelques tâches prédéterminées - 2. Turing, en 1936 - 3. les années 1940 (vers 1946) - 4. un processeur et une mémoire vive - 5. 0 et 1 - 6. oui - 7. 1 octet, soit 8 bits - 8. une suite finie d'instructions qui résout un problème.

8 Exercices

DU PLUS SIMPLE AU PLUS COSTAUD

Exercice 1 Des machines à l'ordinateur★★★ ☐

On s'intéresse à l'histoire des machines à traiter l'information.

1. Qu'est-ce qui distingue une machine à tâche prédéterminée d'une machine universelle ?
2. Quelle idée Turing a-t-il proposée, et en quelle année ?
3. Turing a-t-il construit le premier ordinateur ? Justifier en une phrase.
4. Citer les deux éléments minimaux d'un ordinateur.
5. Pourquoi dit-on qu'un programme est aussi une donnée ?



La rédaction modèle est à gauche, les remarques du correcteur à droite. Un réflexe traverse tout le chapitre : derrière chaque résultat d'IA, demande-toi **d'où viennent les données**.

✓ **Les réponses du test de positionnement (page 4)**

1. a - 2. a - 3. a - 4. b - 5. a - 6. b - 7. a - 8. b - 9. a - 10. b - 11. a - 12. a - 13. a - 14. b - 15. b - 16. a

Questions 1 à 6 : partie 1 · questions 7 à 11 : partie 2 · questions 12 à 16 : partie 3. Une question ratée t'indique la partie à travailler en priorité.

Exercice 1 ★ **Des machines à l'ordinateur**

- Une machine à tâche prédéterminée n'exécute que le travail pour lequel elle a été construite ; une machine universelle exécute **n'importe quel programme** qu'on lui fournit.
- Le concept de **machine universelle**, en **1936**.
- Non : Turing a proposé un concept, c'est-à-dire une idée de logique mathématique. Les premiers ordinateurs, qui la matérialisent, arrivent une dizaine d'années plus tard.
- Un **processeur**, qui calcule, et une **mémoire vive**, qui retient.
- Parce qu'on peut le stocker, le copier et le transmettre exactement comme un texte ou une image : c'est une suite de 0 et de 1 comme une autre.

L'idée avant la machine

Turing a eu l'idée en 1936 ; les ordinateurs l'ont réalisée une dizaine d'années plus tard. Écrire « Turing a inventé l'ordinateur » coûte le point.

Exercice 2 ★ **Quelle taille ?**

- En ASCII non compressé, chaque caractère occupe 1 octet : $2500 \text{ caractères} \times 1 \text{ octet/caractère} = \mathbf{2500 \text{ octets}}$.
- $\frac{2500 \text{ octets}}{1000 \text{ octets/ko}} = \mathbf{2,5 \text{ ko}}$.
- $2500 \text{ octets} \times 8 \text{ bits/octet} = \mathbf{20\,000 \text{ bits}}$.
- La photo pèse 4 Mo = 4 000 000 octets, donc $\frac{4\,000\,000 \text{ octets}}{2500 \text{ octets}} = \mathbf{1600 \text{ fois}}$ plus que le texte.
- Une image code la couleur de chacun de ses très nombreux pixels, alors qu'un texte ne code qu'un octet par caractère : à surface égale, il y a beaucoup plus de pixels que de lettres.

1 caractère, 1 octet

En ASCII non compressé, la taille d'un texte se lit directement dans son nombre de caractères. Les conversions se font par facteurs 1000, et l'unité s'écrit à chaque ligne.

Exercice 3 ★★ **Ordres de grandeur**

- Du plus léger au plus lourd : la page de texte (5 ko), la photo (2 Mo), la chanson (5 Mo), le **film** (4 Go).
- $5 \text{ Mo} \times 1000 \text{ ko/Mo} = 5000 \text{ ko}$, soit $5000 \text{ ko} \times 1000 \text{ octets/ko} = 5\,000\,000 \text{ octets}$.
- La page pèse 5 ko = 5000 octets, donc $\frac{5\,000\,000 \text{ octets}}{5000 \text{ octets}} = \mathbf{1000 \text{ fois}}$ moins que la chanson.
- $\frac{16 \text{ Go}}{4 \text{ Go/film}} = \mathbf{4 \text{ films}}$.
- Le **film** : avec 4 Go par fichier, quatre films suffisent à remplir les 16 Go, alors qu'il faudrait des milliers de photos ou de chansons.

Des facteurs 1000

ko, Mo et Go se suivent de mille en mille : c'est ce qui creuse les écarts. Pour comparer deux tailles, ramène-les d'abord à la même unité.