

Nom :	Prénom :	Classe :
-------	----------	----------

SNT — Programmation : une découverte

Objectifs :

- Découvrir l'intérêt de la programmation dans le monde actuel.
- Découvrir la syntaxe et le vocabulaire d'un langage de programmation.

Programmer ?

Savoir programmer permet de **comprendre** et de **créer** les **outils numériques** qui façonnent notre monde, plutôt que de simplement les utiliser. Cela développe aussi la **logique**, la **créativité** et la **capacité à résoudre des problèmes** de manière structurée.

Agriculture, technologies de l'information, santé et services médicaux, éducation et formation, commerce, construction, tourisme, finance, énergie ... Programmer est un savoir-faire transversal : on le retrouve désormais dans tous les domaines.

Instructions

Tout comme une **recette de cuisine** est constituée **d'étapes**, un **programme** est constitué **d'instructions**.

Imaginons par exemple un **robot** de cuisine à qui nous pourrions donner les **instructions** suivantes :

- Mélange 250g de farine avec 2 cuillères à soupe de sucre et 5 pincées de sel.
- Ajoute progressivement 50cl de lait et 10cl d'eau, tout en mélangeant doucement.
- Fais fondre 50g de beurre, mélange-le puis ajoute-le à la pâte.
- Ajoute 4 œufs sans la coquille et mélange.
- Laisse reposer la pâte pendant 1 heure puis prépare-moi 2 crêpes , stp.

→ Une **instruction** est une **directive** ou un **ordre donné** pour accomplir une **tâche** ou une **action spécifique**.

► Exercice 1 (à vous de jouer)

- Sur ta tablette, **ouvre la page Web** suivante :
- **Choisis**-toi un **robot** (celui que tu veux).
- Tu es dans le **niveau 1**.
Ton but est de traverser la salle dans laquelle se trouve ton robot.
Il y a une sortie, à droite de la salle, tu la vois ?
Comment faire pour atteindre cette sortie ?
- Ton robot ne comprend que les **instructions** suivantes :
 - haut()
 - bas()
 - droite()
 - gauche()
- À toi de jouer, il te faut **taper** les **bonnes instructions**, pour faire sortir ton robot !

Salle 1 — fonction

Fais sortir ton robot, en utilisant les instructions disponibles.

→ Chaque **instruction**, formée d'un mot suivi de parenthèses (ex. `droite()`) est une **fonction**.

Par exemple, les fonctions disponibles dans cette salle 1 sont : `haut()`, `bas()`, `droite()` et `gauche()`.

Salle 2 — appel de fonction

C'est la même chose que pour la salle précédente, sauf qu'il y a des obstacles (des murs) à éviter.

→ Lorsque nous exécutons une fonction, on dit que l'on **appelle** (exécute) une fonction.

Par exemple, dans cette salle 2 nous pouvons appeler les fonctions suivantes : `haut()`, `bas()`, `droite()` et `gauche()`.

Salle 3 — paramètre

Dans cette salle, nous disposons d'une nouvelle fonction : `deverrouiller()`

Cette instruction permet, comme son nom l'indique, de **déverrouiller** une **porte fermée**.

Pour l'utiliser, il faut que ton robot soit devant la porte fermée.

Et dans cette salle, il faut utiliser un **mot de passe** pour déverrouiller la porte.

Heureusement, nous connaissons la **valeur** (le contenu) de ce mot de passe, qui est : `"java"`.

→ Une **valeur** est une donnée qui peut être de différents types (nombre entier, nombre à virgule, texte, etc).

Pour déverrouiller la porte en **utilisant cette valeur**, nous pouvons taper l'instruction suivante :

```
deverrouiller("java")
```

→ Lorsque nous utilisons une valeur, lors de l'appel d'une fonction. Cette valeur se nomme un **paramètre**.

Par exemple, dans cette salle 3 nous allons utiliser la valeur `"java"` comme paramètre de la fonction `deverrouiller()`.

Salle 4 — variable

Nouvelle fonction disponible dans cette salle 4 : la fonction `lire()`.

Cette fonction permet de **recupérer** une valeur, disponible dans la salle.

Et justement, nous avons une valeur à récupérer en **case F9**.

Il nous faut donc guider notre robot jusqu'à cette case F9, puis récupérer la valeur avec la fonction `lire()`.

*Mais où **stocker** cette valeur ?*

Heureusement, dans le monde de la programmation, il existe des « boîtes » (des mots) dans lesquelles nous pouvons stocker une valeur. Ces boîtes sont nommées des **variables**.

La suite, page suivante ...

→ Une **variable** est une "boîte", nommée, dans laquelle on **stocke une valeur**.

Il est très facile de créer une variable, il suffit de mettre un **mot**, suivi du symbole `=`, suivi d'une **valeur**.

Par exemple lorsque je tape `age = 18` j'ai créé une variable dont le nom est `age`, et dont la valeur est `18` !

Tout ceci est fort intéressant, mais comment sortir notre robot de la salle 4 ?

Voici le plan :

1. Se rendre jusqu'à la case F9.
2. Stocker dans une variable la valeur qui est lue, avec l'instruction suivante : `mot_de_passe = lire()`

Ici nous créons une variable dont le nom est `mot_de_passe` et nous stockons dans cette variable la valeur retournée par la fonction `lire()`

3. Se rendre devant la porte verrouillée.
4. Déverrouiller la porte, avec l'instruction suivante : `deverrouiller(mot_de_passe)`

Ici nous appelons la fonction `deverrouiller()` en lui passant la valeur stockée dans la variable `mot_de_passe`

5. Avancer encore d'une case pour sortir de la salle !

Salle 5 — variables

Dans cette *dernière salle*, nous allons devoir créer (déclarer) non pas **une** mais **deux variables** différentes, car nous avons deux valeurs à stocker : une première située en C2, et une seconde située en H8.

Heureusement, dans le monde de la programmation, il est possible de **déclarer autant de variables que souhaitées**.

Par exemple, pour cette salle 5, nous pouvons stocker la première valeur (C2) dans une variable `code_1` :

```
code_1 = lire()
```

Puis nous pouvons stocker la deuxième valeur (H8) dans une variable `code_2` :

```
code_2 = lire()
```

Reste à savoir comment utiliser les valeurs stockées dans ces deux variables, et à quel moment ...

Ce qu'il faut retenir

Vous retrouverez ici l'essentiel du cours.

Astuce : transformez ces points en questions/réponses pour vos flashcards (Anki).

- **Savoir programmer** permet de **comprendre** et de **créer** les **outils numériques** qui façonnent notre monde, **plutôt que de simplement les utiliser**.
- Le savoir-faire en programmation est **transversal** car il est appliqué dans une **multitude de secteurs**, allant de l'informatique à la santé, en passant par l'éducation et l'industrie.
- La **programmation** consiste à écrire des **instructions** pour les ordinateurs
- Une **instruction** est une directive ou un ordre donné pour accomplir une tâche ou une action spécifique.
- Un ensemble complet d'instructions s'appelle un **programme**.
- Une instruction, lorsqu'elle est formée d'un mot suivi de parenthèses (ex. `droite()`), est une **fonction**.
- Lorsque nous exécutons une fonction, on dit que l'on **appelle (exécute) une fonction**.
- Une **valeur** est une information qui peut être de différents types : nombre entier, à virgule, du texte, etc.
- Lorsque nous utilisons une valeur, lors de l'appel d'une fonction. Cette valeur se nomme un **paramètre**.
- Une **variable** est un peu comme une **boîte étiquetée**, dans laquelle on stocke une valeur.
Par exemple, dans l'instruction `age = 18` le nom de la variable (étiquette de la boîte) est `age`, et la valeur (information dans la boîte) est `18`.
- Il est possible de déclarer **autant de variables** que souhaitées.

Pour aller plus loin

En ouvrant le lien <https://fmr.tf/bordy2> vous découvrirez d'autres niveaux à terminer.

Salle	Description
6	Faire appel aux trois fonctions <code>saisir_cable()</code> <code>inverser_branchement()</code> et <code>certifier()</code> pour que votre robot répare le câble réseau et quitte la salle.
7	Dans cette salle, ça se complique un peu ! - Créer (définir) une fonction nommée <code>generer_code</code> prenant <code>login</code> comme paramètre. - Cette fonction retourne (<code>return</code>) la valeur de <code>login</code> à l'envers → <code>login[::-1]</code> Une fois tous les câbles branchés, se rendre sur le terminal (case I2) puis récupérer dans une variable <code>mdp</code> ce que l'appel de la fonction <code>generer_code</code> retourne, lorsqu'on lui passe comme paramètre le texte <code>"bordy"</code> . Enfin, se rendre devant la sortie et <code>deverrouiller</code> la porte en utilisant la valeur dans <code>mdp</code> !
8	Cette salle est la même que la salle 7. Mais la mémoire disponible dans le robot a été diminuée (37 lignes maximum). Il faut donc ruser, en créant une fonction <code>reparer</code> , permettant de réparer un câble. Puis il ne reste plus qu'à appeler cette fonction <code>reparer</code> autant de fois que nécessaire !