

## Activité de découverte – Qu'est-ce qu'une « machine » ?

Activité débranchée (sans ordinateur)

NSI – Première – Séquence 3 : Les fonctions

Avant de découvrir comment on écrit des **fonctions** en Python, cette activité vous propose de comprendre ce qu'est une fonction **sans ordinateur**, à l'aide d'un mot plus parlant : une **machine**. Une machine prend une ou plusieurs choses en entrée, applique une règle fixe, et produit une sortie.

Travaillez au brouillon, seul(e) ou en binôme. Aucune connaissance de Python n'est nécessaire pour cette activité : tout se joue avec du papier, un crayon, et du raisonnement.

| Partie                        | Objectif                              | Durée indicative |
|-------------------------------|---------------------------------------|------------------|
| A – Machines mystères         | Deviner une règle à partir d'exemples | 10 min           |
| B – Annoncer ou transmettre ? | Distinguer afficher et transmettre    | 8 min            |
| C – Repérer les répétitions   | Regrouper des actions répétées        | 10 min           |

### 1 Machines mystères

#### Activité : Deviner la règle d'une machine

Une **machine** prend une entrée, lui applique une règle fixe et secrète, et produit une sortie. On ne voit jamais ce qui se passe « à l'intérieur » de la machine : seulement les entrées qu'on lui donne, et les sorties qu'elle produit. Pour chacune des deux machines ci-dessous, on vous donne quelques exemples déjà réalisés : à vous de deviner la règle, puis de l'appliquer à de nouvelles entrées.

#### Exercice 1: Deux machines mystères

**Machine 1** (une seule entrée) :

| Entrée | Sortie |
|--------|--------|
| 2      | 7      |
| 6      | 11     |
| 9      | 14     |

**Machine 2** (deux entrées à la fois, pour une seule sortie) :

| Entrée 1 | Entrée 2 | Sortie |
|----------|----------|--------|
| 2        | 3        | 6      |
| 4        | 2        | 8      |
| 5        | 1        | 5      |

1. Quelle règle applique la Machine 1 ? Quelle sortie produirait-elle pour l'entrée 0 ? Pour l'entrée 15 ?
2. Quelle règle applique la Machine 2 ? Quelle sortie produirait-elle pour les entrées 3 et 3 ? Pour les entrées 6 et 0 ?

## Correction de l'exercice 1

1. La Machine 1 **ajoute 5** à l'entrée :  $2 + 5 = 7$ ,  $6 + 5 = 11$ ,  $9 + 5 = 14$ . Pour l'entrée **0**, elle produirait **5** ; pour l'entrée **15**, elle produirait **20**.
2. La Machine 2 **multiplie** ses deux entrées :  $2 \times 3 = 6$ ,  $4 \times 2 = 8$ ,  $5 \times 1 = 5$ . Pour **3** et **3**, elle produirait **9** ; pour **6** et **0**, elle produirait **0**.

## Remarque : Le vocabulaire que vous retrouverez dans le cours

Ce que vous venez de manipuler porte un nom précis, que vous découvrirez formellement dans le cours qui suit cette activité : une machine s'appelle une **fonction**, ses entrées sont ses **paramètres**, et sa sortie est la valeur **renvoyée**. Deviner la règle à partir d'exemples, comme vous venez de le faire, est exactement ce que fait un programmeur qui lit la **spécification** d'une fonction sans connaître son code. Une machine n'effectue d'ailleurs pas forcément un calcul : elle peut tout aussi bien transformer un nom en phrase de salutation, par exemple.

## 2 Annoncer ou transmettre ?

## Activité : Un piège classique

Une machine peut produire un résultat de deux façons très différentes : en se contentant de l'**annoncer**, ou en le **transmettant** réellement à la suite du processus. La situation suivante illustre pourquoi cette différence compte.

## Exercice 2: Sam et Alex

Dans un magasin, deux employés ont pour tâche de calculer le prix total d'un panier de courses.

- **Sam** calcule le total, puis le **crie à voix haute** dans le magasin.
- **Alex** calcule le total, l'**écrit sur un ticket**, et donne ce ticket à la personne suivante.

Le gérant du magasin décide d'appliquer une réduction de 2 € sur le total, **avant** de l'annoncer au client. Pour cela, il a besoin de récupérer le total déjà calculé, afin de lui soustraire 2.

1. Le gérant peut-il directement réutiliser le travail de **Sam** pour calculer « total - 2 » ? Pourquoi ?
2. Le gérant peut-il directement réutiliser le travail d'**Alex** pour calculer « total - 2 » ? Pourquoi ?
3. D'après vous, laquelle de ces deux façons de faire correspond à ce qu'on appellera **afficher** un résultat, et laquelle correspond à ce qu'on appellera le **renvoyer** ?

## Correction de l'exercice 2

1. Non : Sam s'est contenté d'**annoncer** le total à voix haute. Cette information n'a été transmise à personne en particulier ; le gérant ne dispose d'aucune valeur qu'il pourrait réellement réutiliser dans un calcul, à moins de tout recalculer lui-même.
2. Oui : Alex a **transmis** le total sous une forme réutilisable (le ticket). Le gérant peut prendre ce ticket et calculer directement « total - 2 ».
3. Le travail de Sam correspond à **afficher** un résultat (on le montre, mais on ne le transmet à personne pour un usage ultérieur) ; le travail d'Alex correspond à le **renvoyer** (on le transmet réellement, pour qu'il puisse être réutilisé). C'est exactement la différence entre **print** et **return** que vous formaliserez dans le cours.

### 3 Repérer les répétitions

#### Activité : Une même tâche, répétée plusieurs fois

Voici les instructions suivies un matin par M. Martin pour préparer le goûter de ses trois enfants. Lisez-les attentivement.

#### Exercice 3: Le goûter de M. Martin

*Pour Emma : il sort une tartine, il tartine du beurre, il ajoute de la confiture, il coupe la tartine en deux, il la met dans la boîte à goûter d'Emma.*

*Pour Lucas : il sort une tartine, il tartine du beurre, il ajoute du miel, il coupe la tartine en deux, il la met dans la boîte à goûter de Lucas.*

1. Quelle suite d'actions est répétée, presque à l'identique, pour les deux enfants ? Qu'est-ce qui change à chaque fois ?
2. Proposer une **machine** (donnez-lui un nom, et indiquez clairement ses **entrées**) qui regrouperait cette suite d'actions répétée, puis réécrire les deux lignes de M. Martin sous la forme de deux appels à votre machine.

#### Correction de l'exercice 3

1. La suite répétée est : « sortir une tartine, tartiner du beurre, ajouter [une garniture], couper la tartine en deux, la mettre dans la boîte à goûter de [un enfant] ». Ce qui change à chaque fois : la **garniture** (confiture, miel) et l'**enfant** concerné (Emma, Lucas).
2. Par exemple, une machine `PreparerTartine` avec deux entrées, l'enfant et la garniture :

— `PreparerTartine(Emma, confiture)`

— `PreparerTartine(Lucas, miel)`

La suite d'actions répétée n'est plus décrite qu'une seule fois (dans la définition de la machine) : c'est exactement ce que vous ferez en Python en **décomposant un programme en fonctions**, et c'est la démarche que vous appliquerez au jeu du pendu lors du premier TP de cette séquence.

#### Remarque : Ce qu'il faut retenir de cette activité

Une **machine** prend une ou plusieurs entrées, applique une règle fixe, et produit une sortie – sans qu'on ait besoin de connaître ce qui se passe à l'intérieur pour l'utiliser. Un résultat **annoncé** n'est pas la même chose qu'un résultat **transmis**. Une suite d'actions répétée peut être regroupée dans une machine nommée, avec des entrées pour ce qui change. Toutes ces idées seront formalisées, avec du vrai code Python, dans le cours qui suit – qui vous montrera aussi comment des machines déjà écrites peuvent se **combiner** entre elles pour obtenir de nouveaux résultats sans rien réinventer.