

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

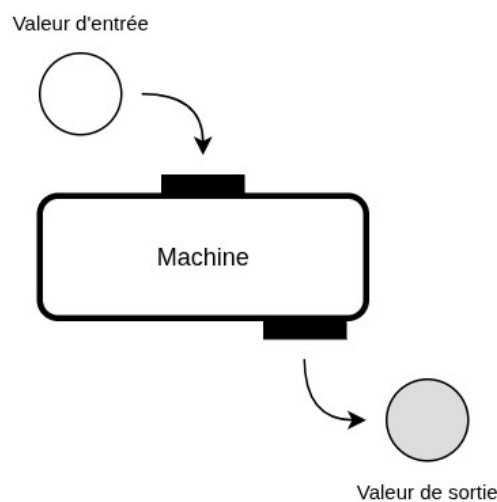
SNT — Fonction #2 : créer (définir) une nouvelle fonction

Objectifs :

- Savoir créer (définir) une nouvelle fonction prenant un paramètre d'entrée.

Introduction

Nous savons qu'une **fonction** est une sorte de « machine » :



Après avoir découvert comment les utiliser¹, voyons à présent comment les créer !

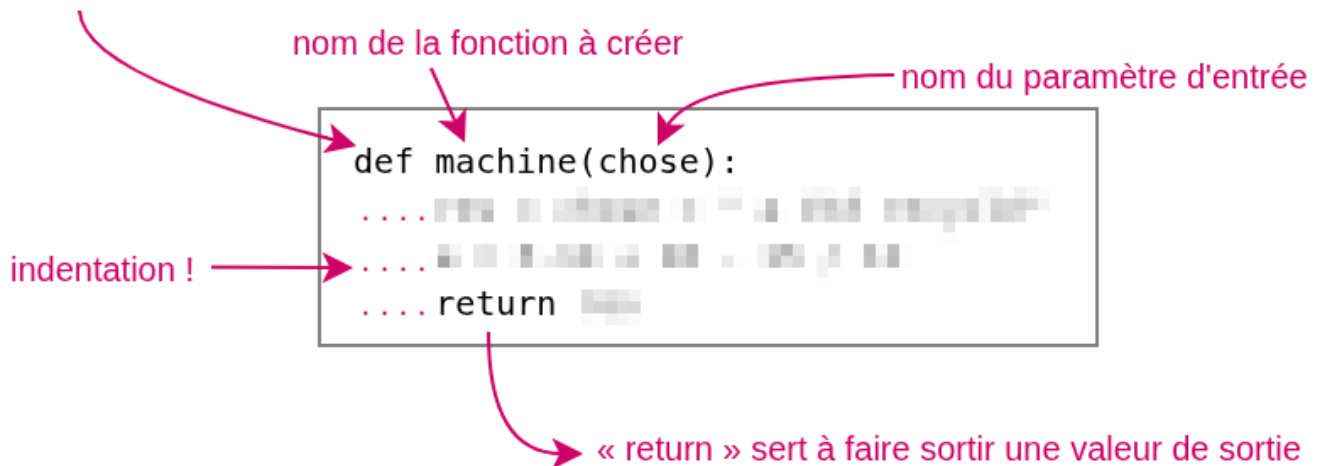
Vocabulaire

- Quand on dit « créer une fonction », cela signifie que nous devons la **définir**.
- En Python on définit une nouvelle fonction avec le mot-clé `def` suivi d'un espace puis du nom de la fonction à créer.
- La variable qui sert à récupérer la valeur d'entrée est baptisée **paramètre d'entrée**.
- Le code contenu dans la fonction (le *bloc de code*) est toujours **indenté**.
- Pour faire sortir une **valeur de sortie** de la fonction, on utilise le mot-clé `return` suivi d'un espace puis de la valeur à faire sortir.

¹ Voir le cours « Fonction : utiliser (appeler) une fonction existante ».

Structure de la création d'une nouvelle fonction, en Python

« def » sert à définir (à créer) une nouvelle fonction



La structure de la création d'une fonction en Python est toujours la même :

- Le mot-clé `def`, suivi du **nom de la fonction** à créer.
*On peut utiliser n'importe quel mot, du moment qu'il ne commence pas par un chiffre.
Dans cet exemple la fonction se nomme « machine ».*
- Sur la même ligne, on a toujours des parenthèses, puis le caractère deux points `:`
Les parenthèses contiennent le nom du **paramètre d'entrée**.
Dans cet exemple, il n'y a qu'un seul paramètre d'entrée, et il se nomme « chose ».
- Le code contenu dans la fonction (le *bloc de code*) est toujours **indenté** de quatre espaces.
- Le bloc de code se termine par le mot-clé `return`.
Il est toujours suivi d'un espace et de la valeur à faire sortir de la fonction (la **valeur de sortie**).

Exemple de création de nouvelle fonction, en Python

Voici le code servant à créer (définir) une nouvelle fonction baptisée `plus_trois`

```
1 def plus_trois(nbre):
2     resultat = nbre + 3
3     return resultat
```

Que peut-on dire de cette fonction, en analysant son code ?

- Cette fonction prend un paramètre d'entrée nommé `nbre`.
- Le bloc de code de cette fonction réalise un calcul.
- Le résultat de ce calcul est stocké dans une variable nommée `resultat`.
- La valeur de `resultat` est la valeur de sortie de cette fonction.

► Exercice 1 — Expliquer

Lire le code du programme ci-dessous puis répondre aux questions posées :

```
1 def puipui(a):
2     resultat = a ** 3
3     return resultat
```

N°	Questions	Réponses
1	Quel est le nom de la fonction ?	puipui
2	Quel est le nom du paramètre d'entrée ?	a
3	Quel est le nom de la variable servant à stocker la valeur de sortie ?	resultat

► Exercice 2 — Compléter

La fonction suivante est censée ajouter 10 à la valeur du paramètre d'entrée, mais elle est incomplète. Compléter les parties marquées par les symboles “.....”

```
1 def ajoute_dix(a):
2     r = a + 10
3     return r
```

► Exercice 3 — Créer

- Soit `fois_vingt` le nom d'une fonction qui prend un paramètre d'entrée nommé `k`.
- Cette fonction réalise un calcul dans lequel la valeur de `k` est multipliée par vingt.
- Le résultat de ce calcul est stocké dans une variable nommée `r`.
- La valeur de `r` est la valeur de sortie de cette fonction.

Écrire le code Python de cette fonction :

```
1 def fois_vingt(k):
2     r = k * 20
3     return r
```

Le bloc de code d'une fonction peut contenir tout type de code

Nous venons de découvrir des créations de fonctions (donc des *définitions de fonctions*) dont le bloc de code ne contenait que des calculs “basiques”. Penchons-nous à présent sur le fait qu'il est également possible de mettre dans ce bloc de code **toutes** les **structures** de code désirées : condition, boucle, etc. Exemples :

```
1 def est_positif(n):
2     if n >= 0:
3         return True
4     else:
5         return False
```

```
1 def tarif_cinema(age):
2     if age <= 16:
3         return 6.40
4     if age >= 65:
5         return 10.50
6     return 13.50
```

```
1 def loop(k):
2     r = 0
3     for _ in range(k):
4         r = r + 10
5     return r
```

► Exercice 4 — Expliquer

Lire le code des trois fonctions ci-dessus, puis répondre aux questions suivantes :

N°	Questions	Réponses
1	Quels sont les noms des trois fonctions ?	<code>est_positif</code> , <code>tarif_cinema</code> , <code>loop</code>
2	Quel est le nom du paramètre d'entrée de chacune de ces fonctions ?	<code>n</code> , <code>age</code> , <code>k</code>
3	Quelle est la valeur de sortie de la fonction <code>est_positif</code> si le paramètre d'entrée vaut <code>10</code> ?	<code>True</code>
4	Quelle est la valeur de sortie de la fonction <code>tarif_cinema</code> si le paramètre d'entrée vaut <code>18</code> ?	<code>13.50</code>
5	Quelle est la valeur de sortie de la fonction <code>loop</code> si le paramètre d'entrée vaut <code>3</code> ?	<code>30</code>