

|  |   |
|--|---|
| <b>Quel est le rôle d'une fonction dans un programme informatique ?</b>                  | Regrouper des instructions que l'on pourra utiliser plusieurs fois dans le programme principal sans avoir à les réécrire à chaque fois.<br><i>Découper son code en fonctions permet également un débogage et une maintenance beaucoup plus facile de ce code...</i> |
| <b>Qu'est-ce que la définition d'une fonction ?</b>                                      | L'ensemble des instructions qui décrivent le traitement que doit faire la fonction.   |
| <b>A quoi correspondent les paramètres d'une fonction ?</b>                              | Les variables qui « récupéreront » les données envoyées par le programme principal, et que la fonction utilisera pour réaliser son traitement :<br><br><pre>def nom_de_la_fonction(parametres) :</pre>  |
| <b>Comment une fonction peut-elle renvoyer un résultat au programme principal ?</b>      | Avec le mot clé <code>return</code> :<br><br><pre>return resultat</pre>   |
| <b>A quoi correspond l'appel d'une fonction ?</b>  | L'instruction dans le programme principal qui déclenche l'exécution de la fonction.   |
| <b>Que sont les arguments ( ou paramètres formels ) lors de l'appel d'une fonction ?</b> | Les données envoyées par le programme principal à la fonction. Du même type et dans le même ordre que les paramètres :<br><br><pre>nom_de_la_fonction(arguments)</pre>  |
| <b>L'appel d'une fonction affiche-t-elle toujours le résultat que celle-ci renvoie ?</b> | Non ! Il faut soit stocker ce résultat dans une variable, soit l'afficher avec la fonction <code>print()</code> , sinon ce résultat est « perdu » :<br><br><pre>print(nom_de_la_fonction(arguments))</pre>  |

Indiquer dans l'exemple ci-dessous :

- la définition de la fonction
- l'appel de la fonction
- les paramètres de la fonction
- le résultat renvoyé par la fonction
- les arguments passés à la fonction

