Plan de cours

Instructions de base

Structures conditionnelles

Structures répétitives

Fonctions et modules

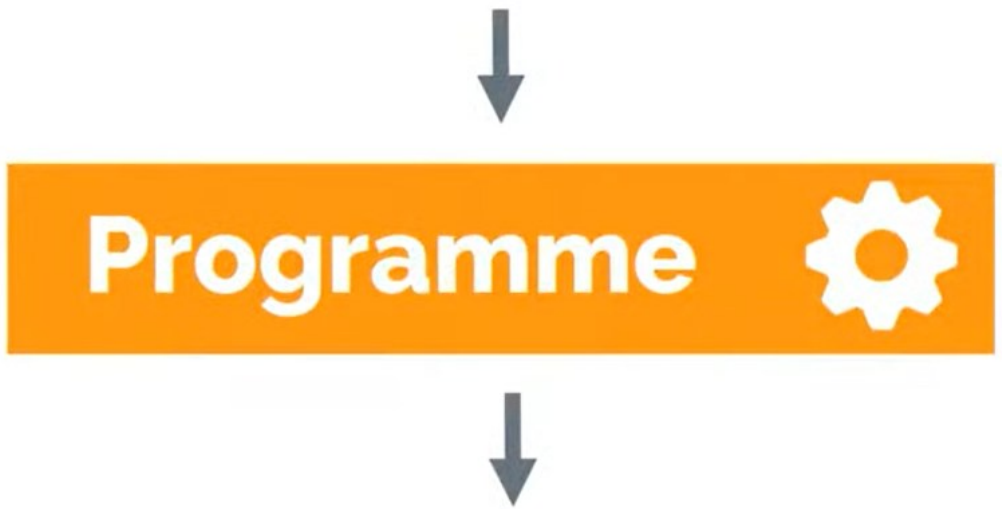
Structures de données

Chaine de caractères

Gestion des fichiers

Concepts avancés

Programme



Programme

Entrées

Programme

Entrées

↓

Programme ⚙

↓

Résultats

Sous-programmes

Entrées

Résultats

Sous-programmes

Entrées

Entrées

Sous_Programme_1

Résultats

Entrées

Sous_Programme_2

Résultats

...

...

...

Entrées

Sous_Programme_n

Résultats

Résultats

Sous-programme

Fonction

Chapeau de magicien

Chapeau de magicien

Chapeau de magicien

Chapeau de magicien

Chapeau de magicien

Fonction

Fonction

Fonction

Fonction

Y

F(...)

Fonction

3

Puissance

Fonction

Fonction

Puissance

Fonction

Puissance

Fonction

9

Puissance

YouTube

Python #8 : Structures répétitives : break et continue

978 vues • 7 mars 2021

59 0 PARTAGER ENREGISTRER ...

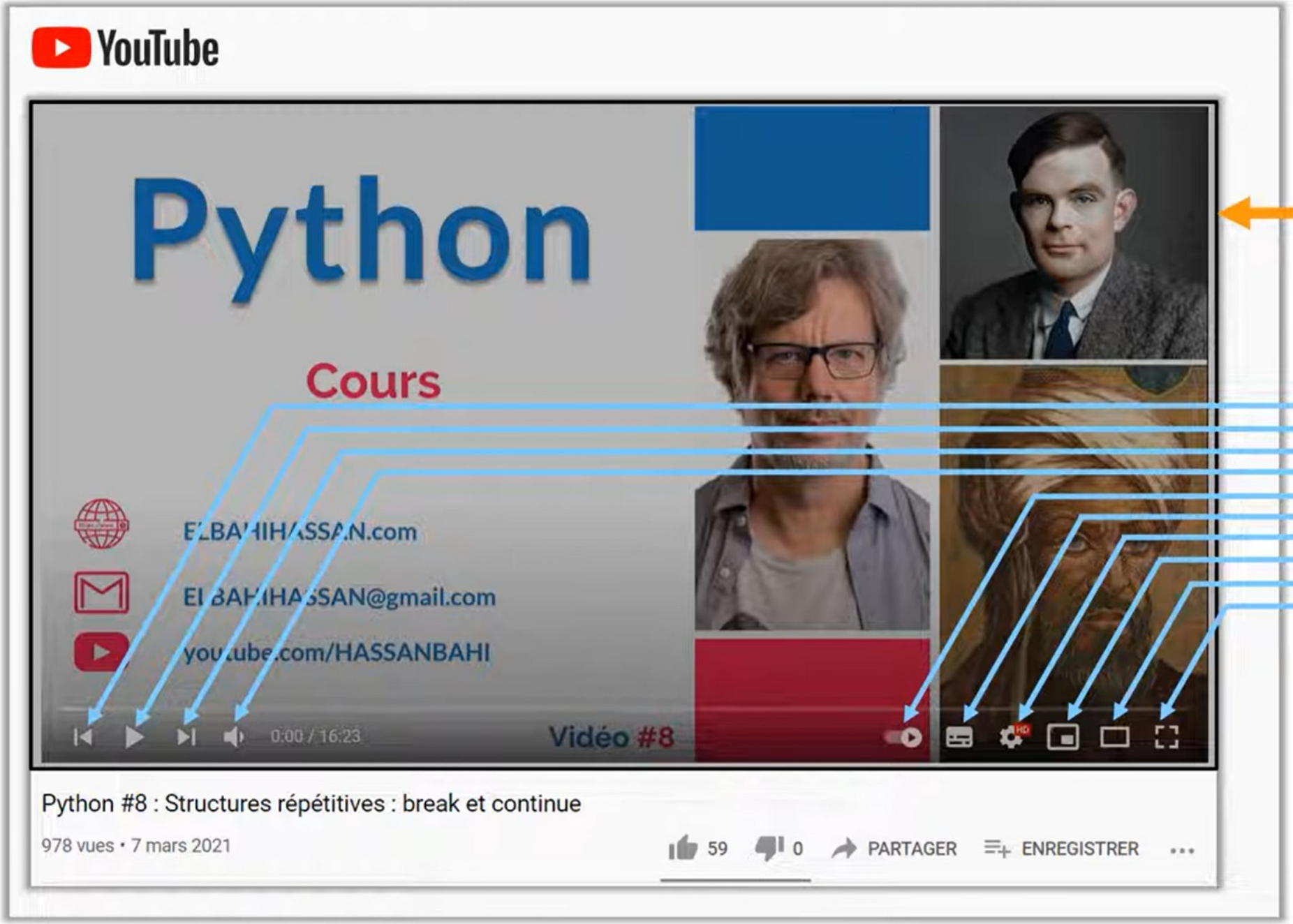
YouTube

Python #8 : Structures répétitives : break et continue

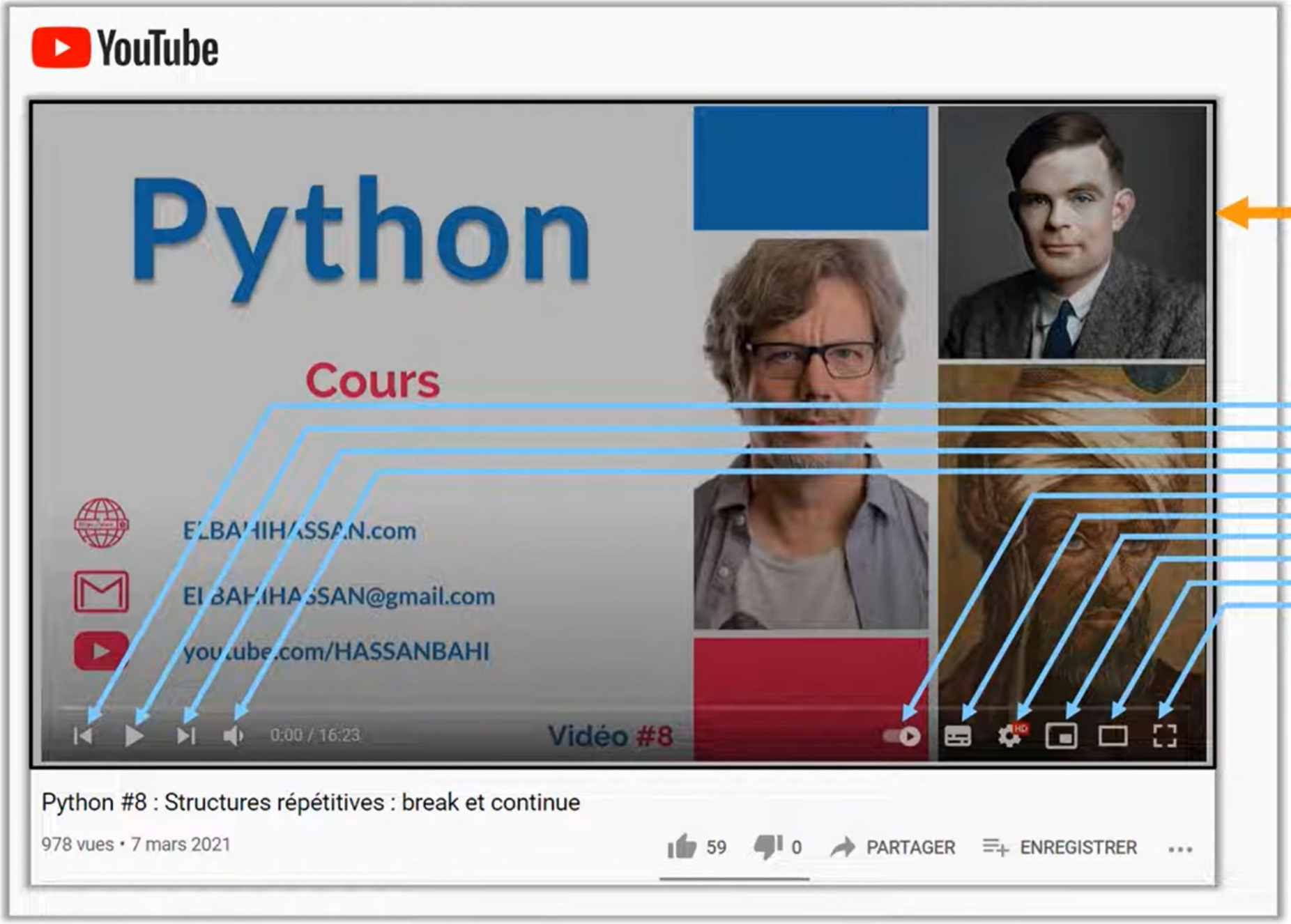
978 vues • 7 mars 2021

59 0 PARTAGER ENREGISTRER ...

← Programme


YouTube
Python
Cours
ELBAHIHASSAN.com
ELBAHIHASSAN@gmail.com
youtube.com/HASSANBAHI
0:00 / 16:23
Vidéo #8
Python #8 : Structures répétitives : break et continue
978 vues • 7 mars 2021
59
0
PARTAGER
ENREGISTRER

Programme



Programme

Fonctions

Fonctions

Qu'est ce qu'une fonction ?

Fonctions

Qu'est ce qu'une fonction ?

Une fonction est une suite d'instructions regroupées sous un nom; elle prend en entrée des paramètres (arguments) et retourne un résultat.

Fonctions

Exemples :

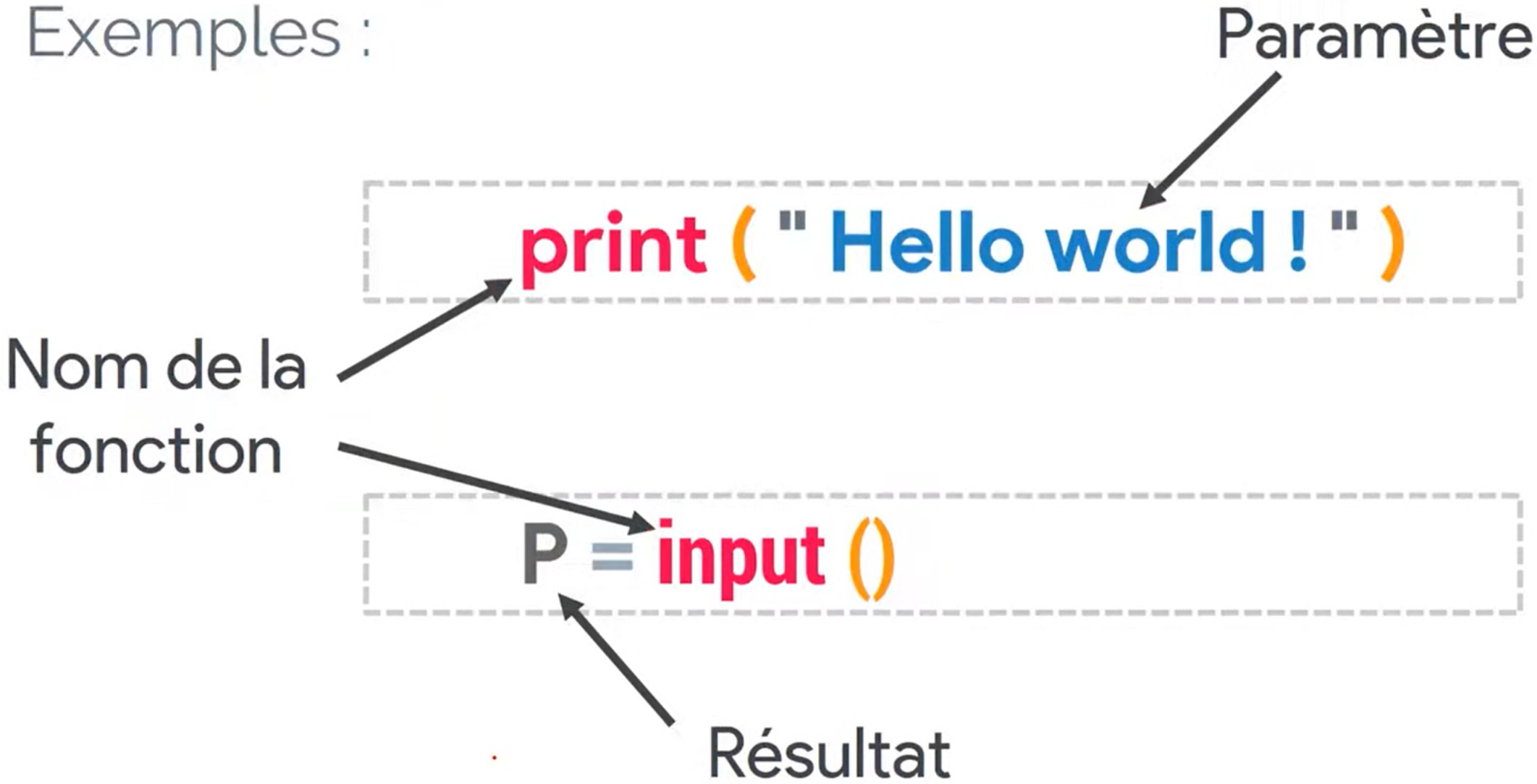
Fonctions

Exemples :

```
print ( " Hello world ! " )
```

```
P = input ()
```

Exemples :


Paramètre
print (" Hello world ! ")
Nom de la fonction
P = input ()
Résultat

Fonction dans langage Python

Fonction dans langage Python

1

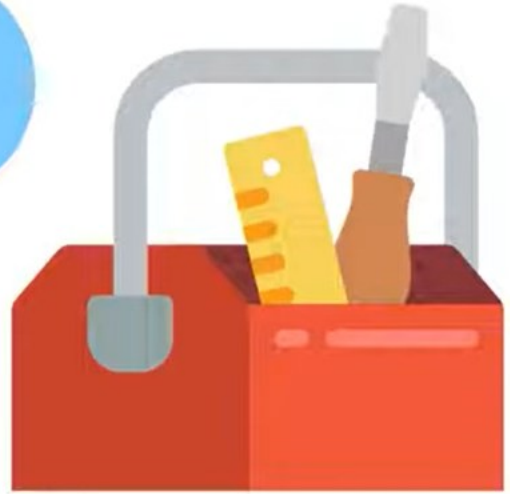
Définir la fonction

2

Appeler la fonction

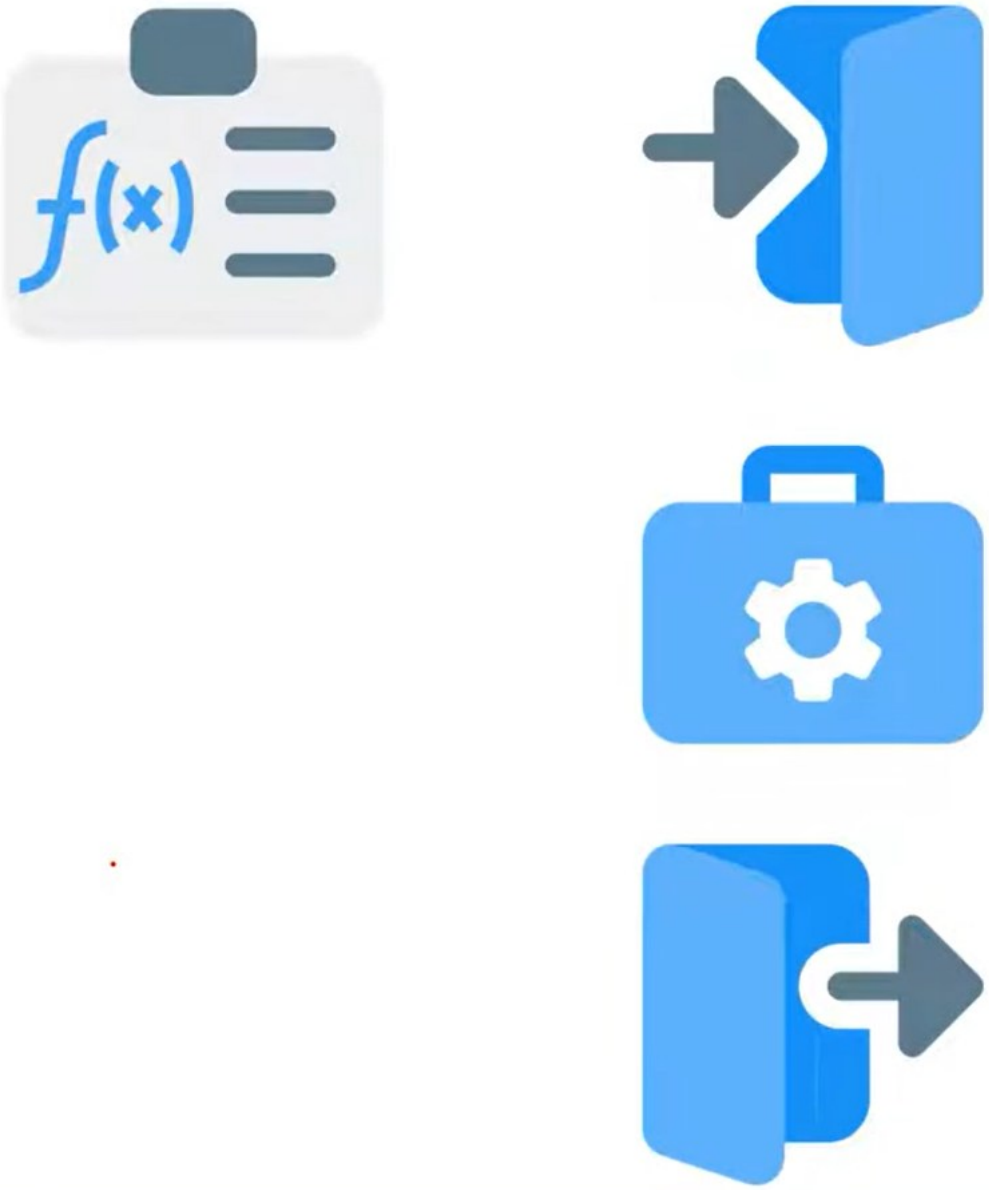
Fonction dans langage Python

1



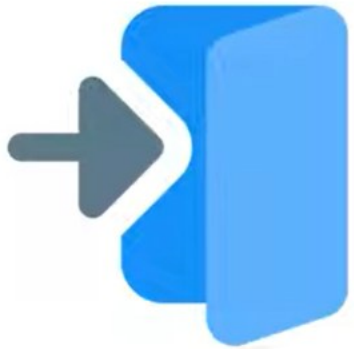
Définir la fonction

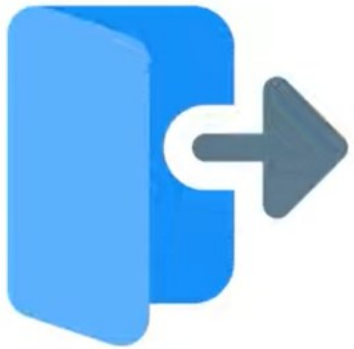
Définition de la fonction



Définition de la fonction

Nom de fonction

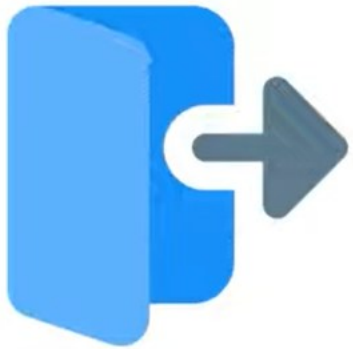




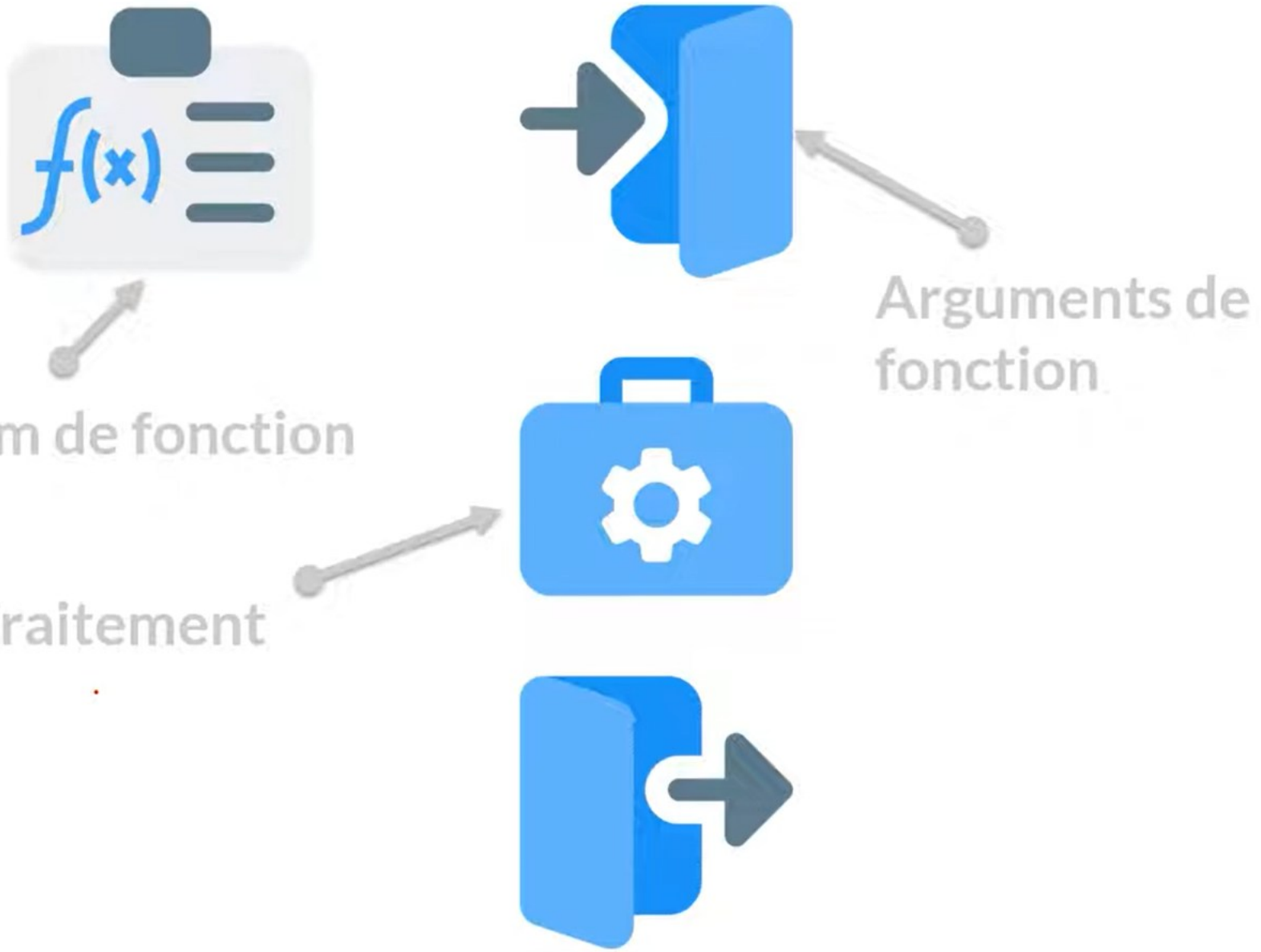
Définition de la fonction

Nom de fonction

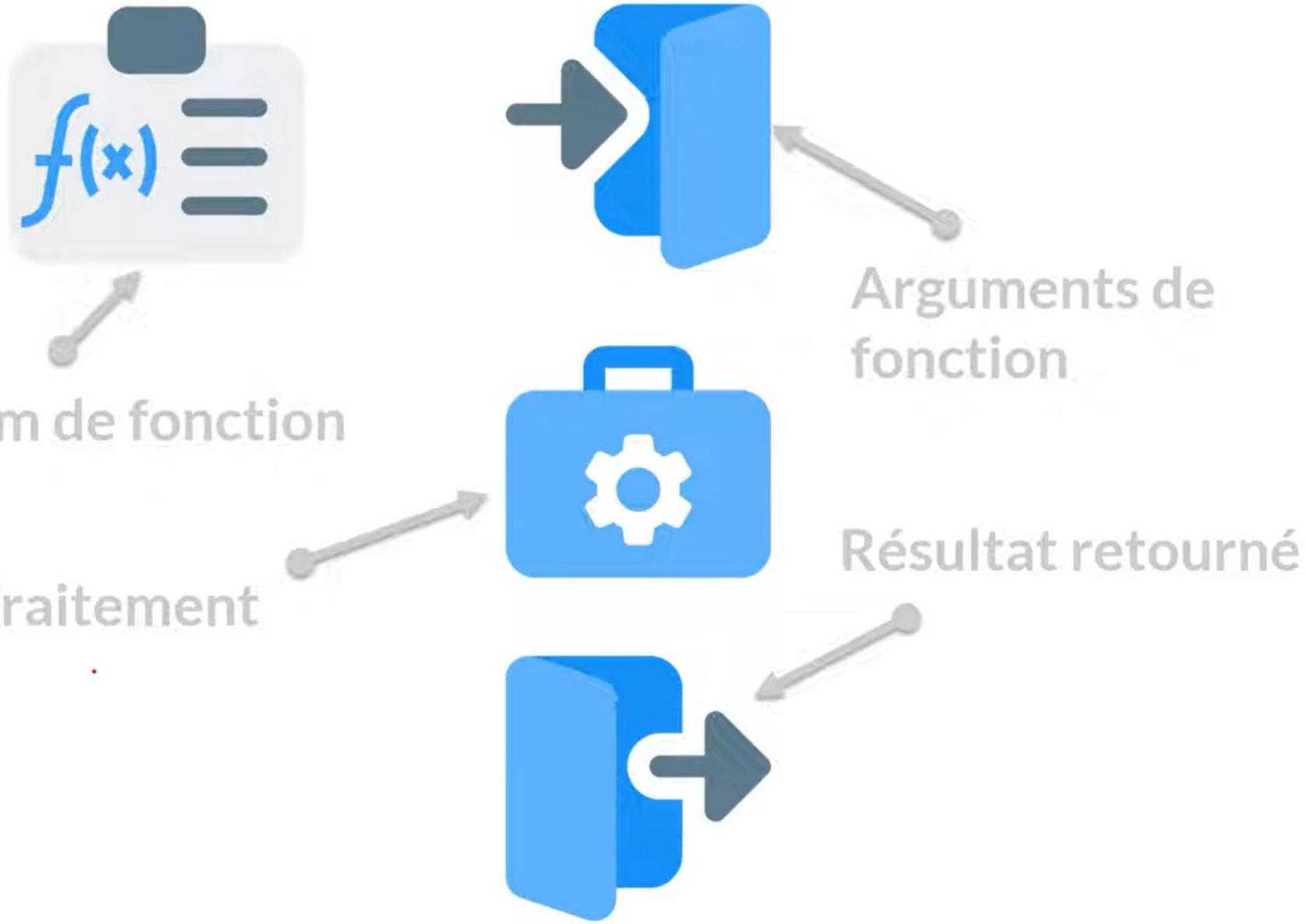
Arguments de fonction



Définition de la fonction

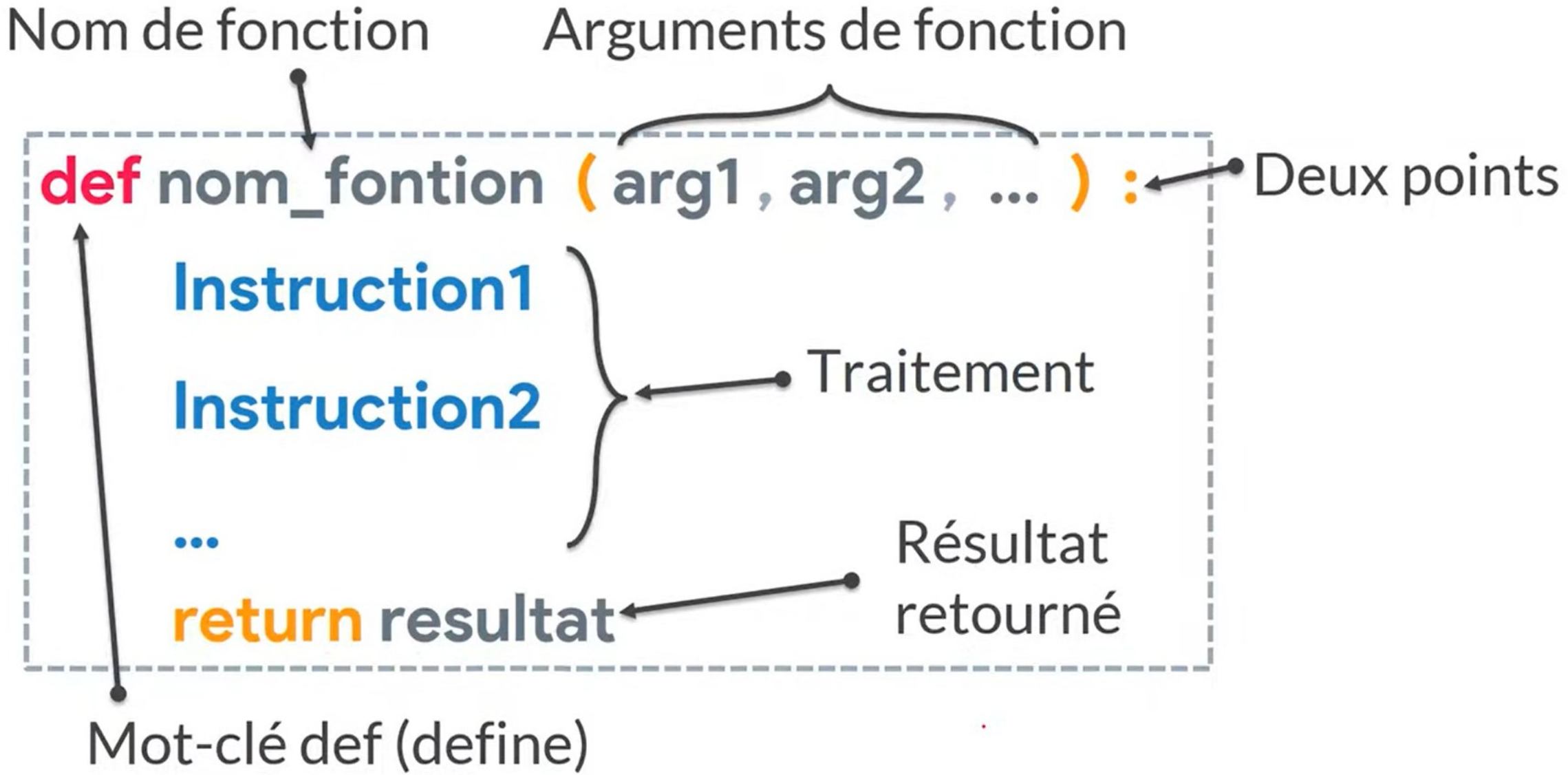


Définition de la fonction



Définition de la fonction

```
def nom_fontion ( arg1 , arg2 , ... ) :
        Instruction1
        Instruction2
        ...
        return resultat
```


Nom de fonction
Arguments de fonction
def nom_fontion (arg1 , arg2 , ...) :
Deux points
Instruction1
Instruction2
...
Traitement
return resultat
Résultat retourné
Mot-clé def (define)

Définition de la fonction

Pas de retour et pas d'arguments

Pas de retour et avec arguments

Avec retour et pas d'arguments

Avec retour et avec arguments

Définition de la fonction

```
def nom_fonction ():
    Instruction1
    Instruction2
    ...
```

Pas de retour et pas d'arguments

```
def nom_fonction (arg1 , arg2 , ... ):
    Instruction1
    Instruction2
    ...
```

Pas de retour et avec arguments

```
def nom_fonction ( ):
    Instruction1
    Instruction2
    ...
    return resultat
```

Avec retour et pas d'arguments

```
def nom_fonction (arg1 , arg2 , ... ):
    Instructions
    Instruction2
    ...
    return resultat
```

Avec retour et avec arguments

Définition de la fonction

```
def bonjour ( ) :
    print ( " Bonjour Ali ! " )
```

Pas de retour et pas d'arguments

```
def bonjour ( nom ) :
    print ( " Bonjour " , nom , " ! " )
```

Pas de retour et avec arguments

```
def somme ( ) :
    A = 10
    B = 5
    C = A + B
    return C
```

Avec retour et pas d'arguments

```
def somme ( A , B ) :
    C = A + B
    return C
```

Avec retour et avec arguments

Fonction dans langage Python

2

Appeler la fonction

Appel de la fonction

Appel de la fonction

Pas de retour et pas d'arguments

Pas de retour et avec arguments

Avec retour et pas d'arguments

Avec retour et avec arguments

Appel de la fonction

```
...
nom_fonction ()
...
```

Pas de retour et pas d'arguments

```
...
nom_fonction (arg1 , arg2 , ... )
...
```

Pas de retour et avec arguments

```
...
x = nom_fonction ()
x =  y / nom_fonction ()
print (nom_fonction () )
...
```

Avec retour et pas d'arguments

```
...
x = nom_fonction (arg1 , arg2 , ... )
x =  y / nom_fonction (arg1 , arg2 , ... )
print (nom_fonction (arg1 , arg2 , ... ))
...
```

Avec retour et avec arguments

Exemple : fonction somme

```
def somme ( a , b ):
        c = a + b
        print ( " a + b = ", c )
a = int ( input ( " Veuillez entrer la valeur de a : " ) )
b = int ( input ( " Veuillez entrer la valeur de b : " ) )
somme ( a , b )
```

```
def somme ( a , b ):
        c = a + b
        return c
a = int ( input ( " Veuillez entrer la valeur de a : " ) )
b = int ( input ( " Veuillez entrer la valeur de b : " ) )
print ( " a + b = " , somme ( a , b ) )
```

```
def somme ( a , b ):
        c = a + b
        print ( " a + b = ", c )
a = int ( input ( " Veuillez entrer la valeur de a : " ) )
b = int ( input ( " Veuillez entrer la valeur de b : " ) )
somme ( a , b )
```

Définir la fonction

Appeler la fonction

```
def somme ( a , b ):
        c = a + b
        return c
a = int ( input ( " Veuillez entrer la valeur de a : " ) )
b = int ( input ( " Veuillez entrer la valeur de b : " ) )
print ( " a + b = " , somme ( a , b ) )
```

Définir la fonction

Appeler la fonction

Exercice

Ecrire un programme qui demande à l'utilisateur de saisir les valeurs de deux variables A et B. Ensuite, il permet de définir et d'appeler les fonctions suivantes :

- Une fonction qui retourne si les valeurs de A et B sont de même signe ou non. (*Une fonction sans valeur de retour et avec arguments*)
- Une fonction qui renvoie le minimum de A et B. (*Une fonction avec une valeur de retour et avec arguments*)
- Une fonction qui renvoie le maximum de A et B. (*Une fonction avec une valeur de retour et avec arguments*)

```python
#Sous programmes
##Definition de la fonction signe
def signe(A, B):
    if A * B < 0:
        print(f"{A} et {B} sont de signe contraire!")
    elif A * B > 0 :
        print(f"{A} et {B} sont de meme signe!")
    else :
        print("L'un ou les deux sont null")
#Definition de la fonction maximum
def maximum(A, B):
    max = A
    if A < B :
        max = B
    return max
#Definition de la fonction minimum
def minimum(A, B):
    min = A
    if A > B :
        min = B
    return min

#Programme
print("Entrez deux nombres pour determiner leurs natures!")
X = int(input("Entrez 1er nombre : "))
Y = int(input("Entrez 2em nombre : "))

##Appel de la fonction signe
signe(X, Y)

##Appel de la fonction minimum
print(f"Le minimum entre {X} et {Y} est : {minimum(X, Y)}")

##Appel de la fonction maximum
print(f"Le maximum entre {X} et {Y} est : {maximum(X, Y)}")
```

```python
#Sous programmes
##Definition de la fonction signe
def signe(A, B):
    if A * B < 0:
        print(f"{A} et {B} sont de signe contraire!")
    elif A * B > 0 :
        print(f"{A} et {B} sont de meme signe!")
    else :
        print("L'un ou les deux sont null")
#Definition de la fonction maximum
def maximum(A, B):
    max = A
    if A < B :
        max = B
    return max
#Definition de la fonction minimum
```

Run scratch

```
Entrez 1er nombre : -3
Entrez 2em nombre : 2
-3 et 2 sont de signe contraire!
Le minimum entre -3 et 2 est : -3
Le maximum entre -3 et 2 est : 2
```