

# Initiation au développement Web

---

## Ressource R209

David Gatel - [david.gatel@univ-rennes1.fr](mailto:david.gatel@univ-rennes1.fr)

Michel Guilleméau - [michel.guilleméau@univ-rennes1.fr](mailto:michel.guilleméau@univ-rennes1.fr)



# #3

# Les fonctions

---

**Ressource R209**

# Les fonctions PHP

---

Le PHP est d'un très grand ensemble de fonctions qui permettent, par exemple,

- le traitement des variables
- le traitement des tableaux
- les fonctions mathématiques
- les fonctions de traitement des caractères
- les fonctions date et heures
- ...

En complément, il est possible de créer des fonctions entièrement définies par l'utilisateur.

# Documentation sur les fonctions PHP

---

**<https://www.php.net/manual/fr/funcref.php>**

# Lecture des informations

Nom de la fonction

- version PHP
- description sommaire

Description

- prototype de la fonction

Liste des paramètres

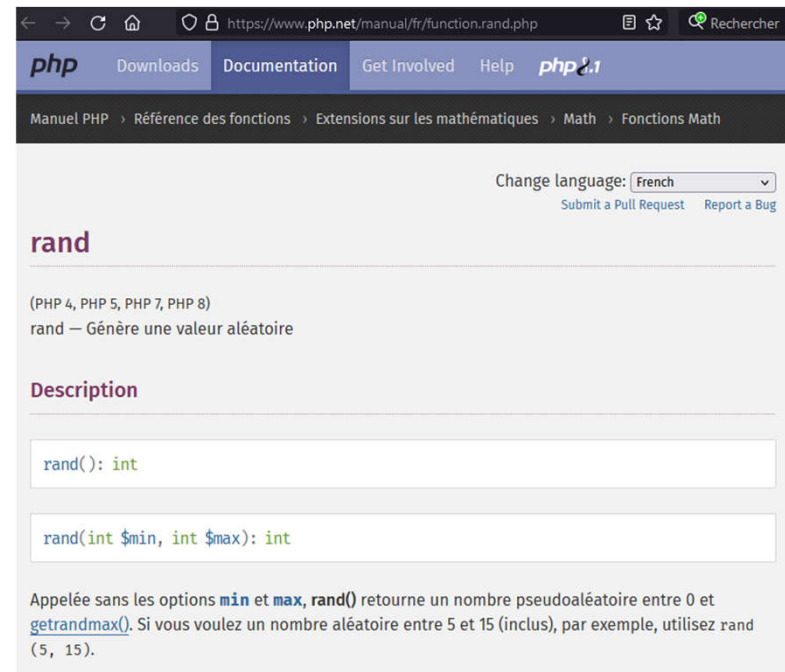
Valeurs de retour

Historique

Exemples

Notes

Voir aussi



# Description, liste des paramètres, valeur de retour

---

Exemple de la fonction rand

- Description

rand(): **int**

rand(**int** \$min, **int** \$max): **int**

- Liste de paramètres

**min** : la plus petite valeur à retourner (par défaut, 0)

**max** : la plus grande valeur à retourner (par défaut, mt\_getrandmax())

- Valeurs de retour

Une valeur pseudoaléatoire, comprise entre min (ou 0) et max (ou mt\_getrandmax(), inclusif).

# Exemple

---

- Exemple

Exemple #1 Exemple avec rand()

```
<?php  
echo rand() . "\n";  
echo rand() . "\n";  
echo rand(5, 15);  
?>
```

Résultat de l'exemple ci-dessus est similaire à :

```
7771  
22264  
11
```

# Notes, Voir aussi

---

## ■ Notes

Avertissement : La plage min max doit se situer dans la plage `getrandmax()`. i.e.  $(\text{max} - \text{min}) \leq \text{getrandmax}()$  sinon, `rand()` peut retourner des nombres aléatoires de mauvaise qualité.

## ■ Voir aussi ¶

- `srand()` - Initialise le générateur de nombres aléatoires
- `getrandmax()` - Plus grande valeur aléatoire possible
- `mt_rand()` - Génère une valeur aléatoire via le générateur de nombre aléatoire Mersenne Twister
- `random_int()` - Génère des nombres entiers pseudo-aléatoire cryptographiquement sécurisé
- `random_bytes()` - Génère des octets pseudo-aléatoire cryptographiquement sécurisé
- `openssl_random_pseudo_bytes()` - Génère une chaîne pseudo-aléatoire d'octets



# Fonctions

## Traitement des variables (1/2)

---

<code>boolval</code>	Récupère la valeur booléenne d'une variable
<code>debug_zval_dump</code>	Extrait une représentation sous forme de chaîne d'une valeur interne à Zend pour affichage
<code>doubleval</code>	Alias de <code>floatval</code>
<code>empty</code>	Détermine si une variable est vide
<code>floatval</code>	Convertit une chaîne en nombre à virgule flottante
<code>get_defined_vars</code>	Liste toutes les variables définies
<code>get_resource_type</code>	Retourne le type de ressource
<code>gettype</code>	Retourne le type de la variable
<code>import_request_variables</code>	Importe les variables de GET/POST/Cookie dans l'environnement global
<code>intval</code>	Retourne la valeur numérique entière équivalente d'une variable
<code>is_array</code>	Détermine si une variable est un tableau
<code>is_bool</code>	Détermine si une variable est un booléen
<code>is_callable</code>	Détermine si l'argument peut être appelé comme fonction
<code>is_countable</code>	Vérifie si le contenu de la variable est une valeur dénombrable
<code>is_double</code>	Alias de <code>is_float</code>
<code>is_float</code>	Détermine si une variable est de type nombre décimal

# Fonctions

## Traitement des variables (2/2)

---

<code>is_int</code>	Détermine si une variable est de type nombre entier
<code>is_integer</code>	Alias de <code>is_int</code>
<code>is_iterable</code>	Détermine si le contenu de la variable est itérable.
<code>is_long</code>	Alias de <code>is_int</code>
<code>is_null</code>	Indique si une variable vaut NULL
<code>is_numeric</code>	Détermine si une variable est un type numérique
<code>is_object</code>	Détermine si une variable est de type objet
<code>is_real</code>	Alias de <code>is_float</code>
<code>is_resource</code>	Détermine si une variable est une ressource
<code>is_scalar</code>	Indique si une variable est un scalaire
<code>is_string</code>	Détermine si une variable est de type chaîne de caractères
<code>isset</code>	Détermine si une variable est déclarée et est différente de NULL
<code>print_r</code>	Affiche des informations lisibles pour une variable
<code>serialize</code>	Génère une représentation stockable d'une valeur
<code>settype</code>	Affecte un type à une variable
<code>strval</code>	Récupère la valeur d'une variable, au format chaîne
<code>unserialize</code>	Crée une variable PHP à partir d'une valeur linéarisée
<code>unset</code>	Détruit une variable
<code>var_dump</code>	Affiche les informations d'une variable
<code>var_export</code>	Retourne le code PHP utilisé pour générer une variable

# Fonctions

## Traitement des tableaux (1/4)

---

<code>array_change_key_case</code>	Change la casse de toutes les clés d'un tableau
<code>array_chunk</code>	Sépare un tableau en tableaux de taille inférieure
<code>array_column</code>	Retourne les valeurs d'une colonne d'un tableau d'entrée
<code>array_combine</code>	Crée un tableau à partir de deux autres tableaux
<code>array_count_values</code>	Compte le nombre de valeurs d'un tableau
<code>array_diff_assoc</code>	Calcule la différence de deux tableaux, en prenant aussi en compte les clés
<code>array_diff_key</code>	Calcule la différence de deux tableaux en utilisant les clés pour comparaison
<code>array_diff_uassoc</code>	Calcule la différence entre deux tableaux associatifs, à l'aide d'une fonction de rappel
<code>array_diff_ukey</code>	Calcule la différence entre deux tableaux en utilisant une fonction de rappel sur les clés pour comparaison
<code>array_diff</code>	Calcule la différence entre des tableaux
<code>array_fill_keys</code>	Remplit un tableau avec des valeurs, en spécifiant les clés
<code>array_fill</code>	Remplit un tableau avec une même valeur
<code>array_filter</code>	Filtre les éléments d'un tableau grâce à une fonction de rappel
<code>array_flip</code>	Remplace les clés par les valeurs, et les valeurs par les clés
<code>array_intersect_assoc</code>	Calcule l'intersection de deux tableaux avec des tests sur les index
<code>array_intersect_key</code>	Calcule l'intersection de deux tableaux en utilisant les clés pour comparaison
<code>array_intersect_uassoc</code>	Calcule l'intersection de deux tableaux avec des tests sur les index, compare les index en utilisant une fonction de rappel
<code>array_intersect_ukey</code>	Calcule l'intersection de deux tableaux en utilisant une fonction de rappel sur les clés pour comparaison
<code>array_intersect</code>	Calcule l'intersection de tableaux
<code>array_key_exists</code>	Vérifie si une clé existe dans un tableau
<code>array_key_first</code>	Récupère la première clé d'un tableau
<code>array_key_last</code>	Récupère la dernière clé d'un tableau

# Fonctions

## Traitement des tableaux (2/4)

---

<code>array_keys</code>	Retourne toutes les clés ou un ensemble des clés d'un tableau
<code>array_map</code>	Applique une fonction sur les éléments d'un tableau
<code>array_merge_recursive</code>	Combine un ou plusieurs tableaux ensemble, récursivement
<code>array_merge</code>	Fusionne plusieurs tableaux en un seul
<code>array_multisort</code>	Trie les tableaux multidimensionnels
<code>array_pad</code>	Complète un tableau avec une valeur jusqu'à la longueur spécifiée
<code>array_pop</code>	Dépile un élément de la fin d'un tableau
<code>array_product</code>	Calcule le produit des valeurs du tableau
<code>array_push</code>	Empile un ou plusieurs éléments à la fin d'un tableau
<code>array_rand</code>	Prend une ou plusieurs clés, au hasard dans un tableau
<code>array_reduce</code>	Réduit itérativement un tableau
<code>array_replace_recursive</code>	Remplace récursivement dans le premier tableau les éléments des autres tableaux fournis
<code>array_replace</code>	Remplace les éléments d'un tableau par ceux d'autres tableaux
<code>array_reverse</code>	Inverse l'ordre des éléments d'un tableau
<code>array_search</code>	Recherche dans un tableau la clé associée à la première valeur
<code>array_shift</code>	Dépile un élément au début d'un tableau
<code>array_slice</code>	Extrait une portion de tableau
<code>array_splice</code>	Efface et remplace une portion de tableau
<code>array_sum</code>	Calcule la somme des valeurs du tableau
<code>array_udiff_assoc</code>	Calcule la différence entre des tableaux avec vérification des index
<code>array_udiff_uassoc</code>	Calcule la différence de deux tableaux associatifs, compare les données et les index avec une fonction de rappel
<code>array_udiff</code>	Calcule la différence entre deux tableaux en utilisant une fonction rappel

# Fonctions

## Traitement des tableaux (3/4)

---

<code>array_intersect_assoc</code>	Calcule l'intersection de deux tableaux avec des tests sur l'index, compare les données en utilisant une fonction de rappel
<code>array_intersect_uassoc</code>	Calcule l'intersection de deux tableaux avec des tests sur l'index, compare les données et les index des deux tableaux en utilisant une fonction de rappel séparée
<code>array_intersect</code>	Calcule l'intersection de deux tableaux, compare les données en utilisant une fonction de rappel
<code>array_unique</code>	Dédoublonne un tableau
<code>array_unshift</code>	Empile un ou plusieurs éléments au début d'un tableau
<code>array_values</code>	Retourne toutes les valeurs d'un tableau
<code>array_walk_recursive</code>	Applique une fonction de rappel récursivement à chaque membre d'un tableau
<code>array_walk</code>	Exécute une fonction fournie par l'utilisateur sur chacun des éléments d'un tableau
<code>array</code>	Crée un tableau
<code>arsort</code>	Trie un tableau en ordre inverse et conserve l'association des index
<code>asort</code>	Trie un tableau et conserve l'association des index
<code>compact</code>	Crée un tableau à partir de variables et de leur valeur
<code>count</code>	Compte tous les éléments d'un tableau ou quelque chose d'un objet
<code>current</code>	Retourne l'élément courant du tableau
<code>each</code>	Retourne chaque paire clé/valeur d'un tableau
<code>end</code>	Positionne le pointeur de tableau en fin de tableau

# Fonctions

## Traitement des tableaux (4/4)

---

extract	Importe les variables dans la table des symboles
in_array	Indique si une valeur appartient à un tableau
key_exists	Alias de array_key_exists
key	Retourne une clé d'un tableau associatif
krsort	Trie un tableau en sens inverse et suivant les clés
ksort	Trie un tableau suivant les clés
list	Assigne des variables comme si elles étaient un tableau
natcasesort	Trie un tableau avec l'algorithme à "ordre naturel" insensible à la casse
natsort	Trie un tableau avec l'algorithme à "ordre naturel"
next	Avance le pointeur interne d'un tableau
pos	Alias de current
prev	Reculé le pointeur courant de tableau
range	Crée un tableau contenant un intervalle d'éléments
reset	Remet le pointeur interne de tableau au début
rsort	Trie un tableau en ordre inverse
shuffle	Mélange les éléments d'un tableau
sizeof	Alias de count
sort	Trie un tableau
uasort	Trie un tableau en utilisant une fonction de rappel
uksort	Trie un tableau par ses clés en utilisant une fonction de rappel
usort	Trie un tableau en utilisant une fonction de comparaison

# Tri des tableaux (1/2)

---

- Il est possible de trier un tableau sur ses valeurs, mais également sur ses clés.
- Trier un tableau permet de maintenir la corrélation entre la clé et la valeur, mais ce n'est pas systématique.
- L'ordre du tri s'appuie sur un ordre qui est peut-être alphabétique, croissant, décroissant, numérique, naturel, aléatoire ou personnalisé.
- Ressources : <https://www.php.net/manual/fr/array.sorting.php>

# Tri des tableaux (2/2)

Nom de la fonction	Tri par	Association clé-valeur	Ordre de tri	Fonctions associée
array_multisort()	valeur	associatif oui, numérique non	premier tableau, ou bien options de tri	array_walk()
asort()	valeur	oui	croissant	arsort()
arsort()	valeur	oui	décroissant	asort()
krsort()	clé	oui	décroissant	ksort()
ksort()	clé	oui	croissant	asort()
natcasesort()	valeur	oui	naturel, insensible à la casse	natsort()
natsort()	valeur	oui	naturel	natcasesort()
rsort()	valeur	non	décroissant	sort()
shuffle()	valeur	non	aléatoire	array_rand()
sort()	valeur	non	croissant	rsort()
uasort()	valeur	oui	Défini par une fonction utilisateur	uksort()
uksort()	clé	oui	Défini par une fonction utilisateur	uasort()
usort()	valeur	non	Défini par une fonction utilisateur	uasort()



# Fonctions

## Fonctions mathématiques (1/2)

---

abs	Valeur absolue
acos	Arc cosinus
acosh	Arc cosinus hyperbolique
asin	Arc sinus
asinh	Arc sinus hyperbolique
atan2	Arc tangente de deux variables
atan	Arc tangente
atanh	Arc tangente hyperbolique
base_convert	Convertit un nombre entre des bases arbitraires
bindec	Convertit de binaire en décimal
ceil	Arrondit au nombre supérieur
cos	Cosinus
cosh	Cosinus hyperbolique
decbin	Convertit de décimal en binaire
dechex	Convertit de décimal en hexadécimal
decoct	Convertit de décimal en octal
deg2rad	Convertit un nombre de degrés en radians
exp	Calcul l'exponentielle de e
expm1	Calcule précisément exponentiel moins un
floor	Arrondit à l'entier inférieur
fmod	Retourne le reste de la division
getrandmax	Plus grande valeur aléatoire possible
hexdec	Convertit de hexadécimal en décimal

# Fonctions

## Fonctions mathématiques (2/2)

---

hypot	Calcul la longueur de l'hypoténuse d'un triangle à angle droit
intdiv	Division d'Entier
is_finite	Indique si un nombre est fini
is_infinite	Indique si un nombre est infini
is_nan	Indique si une valeur n'est pas un nombre
lcg_value	Générateur de congruence combinée linéaire
log10	Logarithme en base 10
log1p	Calcule précisément $\log(1 + \text{nombre})$
log	Logarithme naturel (népérien)
max	La plus grande valeur
min	La plus petite valeur
mt_getrandmax	La plus grande valeur aléatoire possible
mt_rand	Génère une valeur aléatoire via le générateur de nombre aléatoire Mersenne Twister
mt_srand	Initialise le générateur de nombres aléatoires Mersenne Twister
octdec	Conversion d'octal en décimal
pi	Retourne la valeur de pi
pow	Expression exponentielle
rad2deg	Conversion de radians en degrés
rand	Génère une valeur aléatoire
round	Arrondit un nombre à virgule flottante
sin	Sinus
sinh	Sinus hyperbolique
sqrt	Racine carrée
srand	Initialise le générateur de nombres aléatoires
tan	Tangente
tanh	Tangente hyperbolique

# Fonctions

## Traitement des caractères (1/2)

---

<code>addslashes</code>	Ajoute des slash dans une chaîne, à la mode du langage C
<code>addslashes</code>	Ajoute des antislashes dans une chaîne
<code>bin2hex</code>	Convertit des données binaires en représentation hexadécimale
<code>chop</code>	Alias de <code>rtrim</code>
<code>chr</code>	Générer une chaîne d'un octet à partir d'un nombre
<code>chunk_split</code>	Scinde une chaîne
<code>convert_cyr_string</code>	Convertit une chaîne d'un jeu de caractères cyrillique à l'autre
<code>convert_uuencode</code>	Décode une chaîne au format uuencode
<code>convert_uuencode</code>	Encode une chaîne de caractères en utilisant l'algorithme uuencode
<code>count_chars</code>	Retourne des statistiques sur les caractères utilisés dans une chaîne
<code>crc32</code>	Calcule la somme de contrôle CRC32
<code>crypt</code>	Hachage à sens unique (indéchiffrable)
<code>echo</code>	Affiche une chaîne de caractères
<code>explode</code>	Scinde une chaîne de caractères en segments
<code>fprintf</code>	Écrit une chaîne formatée dans un flux
<code>get_html_translation_table</code>	Retourne la table de traduction des entités utilisée par <code>htmlspecialchars</code> et <code>htmlentities</code>
<code>hebreu</code>	Convertit un texte logique hébreux en texte visuel
<code>hebreuc</code>	Convertit un texte logique hébreux en texte visuel, avec retours à la ligne
<code>hex2bin</code>	Convertit une chaîne binaire encodée en hexadécimal
<code>html_entity_decode</code>	Convertit les entités HTML à leurs caractères correspondant
<code>htmlentities</code>	Convertit tous les caractères éligibles en entités HTML
<code>htmlspecialchars_decode</code>	Convertit les entités HTML spéciales en caractères
<code>htmlspecialchars</code>	Convertit les caractères spéciaux en entités HTML

# Fonctions

## Traitement des caractères (2/2)

---

<code>implode</code>	Rassemble les éléments d'un tableau en une chaîne
<code>join</code>	Alias de <code>implode</code>
<code>lcfirst</code>	Met le premier caractère en minuscule
<code>levenshtein</code>	Calcule la distance Levenshtein entre deux chaînes
<code>localeconv</code>	Lit la configuration locale
<code>ltrim</code>	Supprime les espaces (ou d'autres caractères) de début de chaîne
<code>md5_file</code>	Calcule le md5 d'un fichier
<code>md5</code>	Calcule le md5 d'une chaîne
<code>metaphone</code>	Calcule la clé metaphone
<code>money_format</code>	Met un nombre au format monétaire
<code>nl_langinfo</code>	Rassemble des informations sur la langue et la configuration locale
<code>nl2br</code>	Insère un retour à la ligne HTML à chaque nouvelle ligne
<code>number_format</code>	Formate un nombre pour l'affichage
<code>ord</code>	Convertit le premier octet d'une chaîne en une valeur entre 0 et 255
<code>parse_str</code>	Analyse une chaîne de caractères d'une variable
<code>print</code>	Affiche une chaîne de caractères
<code>printf</code>	Affiche une chaîne de caractères formatée
<code>quoted_printable_decode</code>	Convertit une chaîne quoted-printable en chaîne 8 bits
<code>quoted_printable_encode</code>	Convertit une chaîne 8 bits en une chaîne quoted-printable
<code>quotemeta</code>	Protège les métacaractères
<code>rtrim</code>	Supprime les espaces (ou d'autres caractères) de fin de chaîne
<code>setlocale</code>	Modifie les informations de localisation
<code>sha1_file</code>	Calcule le sha1 d'un fichier
<code>sha1</code>	Calcule le sha1 d'une chaîne de caractères
<code>similar_text</code>	Calcule la similarité de deux chaînes

# Fonctions

## Dates et heures (1/2)

---

checkdate	Valide une date grégorienne
date_add	Alias de DateTime::add
date_create_from_format	Alias de DateTime::createFromFormat
date_create_immutable_from_format	Alias de DateTimeImmutable::createFromFormat
date_create_immutable	Alias de DateTimeImmutable::__construct
date_create	Alias de DateTime::__construct
date_date_set	Alias de DateTime::setDate
date_default_timezone_get	Récupère le décalage horaire par défaut utilisé par toutes les fonctions date/heure d'un script
date_default_timezone_set	Définit le décalage horaire par défaut de toutes les fonctions date/heure
date_diff	Alias de DateTime::diff
date_format	Alias de DateTime::format
date_get_last_errors	Alias de DateTime::getLastErrors
date_interval_create_from_date_string	Alias de DateInterval::createFromDateString
date_interval_format	Alias de DateInterval::format
date_isodate_set	Alias de DateTime::setISODate
date_modify	Alias de DateTime::modify
date_offset_get	Alias de DateTime::getOffset
date_parse_from_format	Récupère les informations d'une date donnée suivant un format spécifique
date_parse	Retourne un tableau associatif avec des informations détaillées sur une date donnée

# Fonctions

## Dates et heures (2/2)

---

<code>date_sub</code>	Alias de <code>DateTime::sub</code>
<code>date_sun_info</code> de l'aube	Retourne un tableau avec les informations sur le lever/coucher du soleil ainsi que le début et la fin
<code>date_sunrise</code>	Retourne l'heure de lever du soleil pour un jour et un endroit donnés
<code>date_sunset</code>	Retourne l'heure de coucher du soleil pour un jour et un endroit donnés
<code>date_time_set</code>	Alias de <code>DateTime::setTime</code>
<code>date_timestamp_get</code>	Alias de <code>DateTime::getTimestamp</code>
<code>date_timestamp_set</code>	Alias de <code>DateTime::setTimestamp</code>
<code>date_timezone_get</code>	Alias de <code>DateTime::getTimezone</code>
<code>date_timezone_set</code>	Alias de <code>DateTime::setTimezone</code>
<code>date</code>	Formate une date/heure locale

# Fonctions définies par l'utilisateur

## Intérêts

---

- Les intérêts de construire des fonctions utilisateurs sont multiples
  - Organisation du code
  - Maintenance du code
  - Partage du code
  - Simplification de l'écriture de pages en évitant les répétitions
- Par exemple, les fonctions utilisateurs servent à
  - Créer une connexion vers une base de données
  - Exploiter le contenu d'un fichier
  - Appliquer un traitement particulier à des données
  - Créer du code html pour la structure d'une page (entête, barre de navigation, pied de page, ...)

# Fonctions définies par l'utilisateur

## Utilisation

---

- Pour appeler une fonction, on utilise son nom avec entre parenthèses, les arguments utiles à son fonctionnement.

```
<?php
// définition de la fonction
    function generate($p){
        echo "<p>Copyright $p</p>";
    }

// appel de la fonction
    generate(2022);

?>
```



# Fonctions définies par l'utilisateur

## Définition d'une fonction

---

- La définition d'une fonction utilise le mot clé `function`

```
function nomfonction($param1 , $param2 , . . . )  
{  
    . . .  
    return ... ;  
}
```

- Pas de \$ devant le nom de la fonction
- Plusieurs arguments possibles
- `return` permet de renvoyer un unique résultat.
- Les fonctions n'ont pas besoin d'être définies avant d'être utilisées, sauf lorsqu'une fonction est définie conditionnellement.
- Il est possible d'appeler des fonctions récursives.

# Fonctions définies par l'utilisateur

## Les arguments

---

- La liste d'argument est séparée par une virgule
- Passage d'arguments par valeur :
  - Par défaut c'est la valeur qui est passée.
  - La valeur d'un argument dans la fonction ne change pas sa valeur à l'extérieur de la fonction.
- Passage d'arguments par référence
  - Pour changer la valeur à l'extérieur de la fonction, il faut utiliser une référence en notant & devant le nom de l'argument.
- Valeur par défaut des argument
  - Il est possible de définir une valeur par défaut de argument :

```
function servir_cafe ($type = "cappuccino"){ return "Servir un $type.\n";}
```

- Déclaration de type
  - Par défaut, le type de l'argument n'est pas précisé.
  - En précisant le type de l'argument, cela permet d'être certain que l'argument passé correspond au type attendu dans la fonction.
  - array, bool, float, int, string, iterable, object

# Fonctions définies par l'utilisateur

## Les valeurs de retour

---

- La valeur de retour d'une fonction est optionnelle.
- Lorsqu'une fonction renvoie une valeur, c'est avec le mot clé return.
- Il est éventuellement possible de déclarer le type de retour de la fonction.