


# Développement WEB PHP - Cours complet

**BUT Informatique parcours DACS**



# Table of contents:

- Programme
  - Déroulement
    - Séance 1 - Introduction
    - Séance 2 - Front-end
    - Séance 3 - Back-end
    - Séance 4 - Interaction base-de-données
    - Séance 5 - Introduction à la sécurité
    - Séance 6 - Framework
    - Séance 7 - Laravel
    - Séance 8-15 (7 séances) - Projet noté
  - Objectifs pédagogiques
    - Développement WEB
    - Architecture logicielle
- index
- Présentation de la matière et des objectifs
- 📖 | Introduction à PHP
  - Coder en PHP
  - Différences de PHP
    - Echo
    - Debug
    - For each
    - Match
    - Tableaux associatifs
    - Opérateur "." pour ajouter des strings
- 📚 | TD 1 : Premiers scripts
  - Cloner le projet
  - Faire les exercices
- ⚙️ | Fonctions utilitaires & autres outils
  - Utilitaires pour les string (chaînes de caractères)
    - String en majuscule ou minuscule
    - Séparer avec explode et regrouper avec implode
    - Supprimer les espaces en trop avec trim
    - Remplacer une partie d'un string avec replace
  - Utilitaires pour les int et float (nombres)
    - Transformer une variable en int ou float
  - Autres utilitaires

- Checker le type d'une variable
- Checker si une variable existe (et si elle est vide)
-  | TD 2 : Utilitaires & outils
  - Cloner le projet
  - Faire les exercices

# Programme

Ce cours vise à enseigner le langage de programme PHP et le framework Laravel aux étudiants en 2ème année de BUT Informatique en parcours DACS de l'IUT Lyon1.

## Déroulement

### ATTENTION

Les versions de PHP et Laravel utilisées sont respectivement : **PHP 8.2** et **Laravel 11**

## Séance 1 - Introduction

- Présentation de la matière et des objectifs
- Introduction à PHP
- TD 1 : Premiers scripts
- Fonctions utilitaires & autres outils
- TD 2 : XXXXXX

## Séance 2 - Front-end

- Introduction aux concepts de stack WEB (front-end, back-end, base-de-données), rendu côté serveur (SSR) et au protocole HTTP(S)
- TP : Gérer un formulaire
- TP : Page de profil utilisateur

## Séance 3 - Back-end

- Introduction aux concepts d'API
- TP : Serveur HTTP basique
  - Routing

## Séance 4 - Interaction base-de-données

- Introduction à PDO et au concept d'ORM
- TP : Requêtes basiques et mini-ORM

## Séance 5 - Introduction à la sécurité

- Introduction à la cyber-sécurité (injection SQL, XSS, man in the middle, DDoS)
- TP : Challenges "root-me" like
  - Tag (simple) -> mdp dans une balise
  - Guess the password (simple) -> le mdp c'est "admin"
  - All Routes Lead To Rome (simple) -> trouver mdp dans le fichier "Rome" grâce à l'URL  
`http://monserv/get_file?filename=../../Rome`
  - Botox injection (medium) -> Injection SQL
  - XSS (medium) -> Attaque XSS
  - Decrypt me (medium) -> Décoder du base64
  - MitM (hard) -> Truc style voler un token via wireshark

## Séance 6 - Framework

- Introduction au concept de framework
- TP : Créer son propre framework minimaliste en assemblant les parties précédentes
  - Features attendues :
    - Routing
    - Moteur de templating
    - Mini-ORM
    - Configuration utilisateur (host, port, bdd)
  - Le moins de failles de sécurité possible

## Séance 7 - Laravel

- Introduction à Laravel (routing, controllers, requests/responses, validation, migrations, seeding, commands, jobs), présentation de la documentation et de l'écosystème
- Mini exposé : Chaque élève présente un élément de l'écosystème Laravel (description, exemples d'usage, complexité, popularité, prix, fiabilité)
- Introduction au concept de package manager et composer
- TP : Création d'un projet Laravel vide

## Séance 8-15 (7 séances) - Projet noté

- Introduction du sujet du Projet
  - "Création d'une application de gestion d'applications et de tickting"
  - Seul ou en groupe de 2
  - Rendu sur GitHub
  - Description de cas d'usages :
    - Le développeur et l'admin peuvent créer une application. Seul l'admin peut supprimer une application
      - L'application est liée à plusieurs utilisateurs, possède un nom, une description et un état (en développement, en test, déployée, en maintenance)
    - Les utilisateurs peuvent créer des tickets pour une application, seulement l'admin peut supprimer un ticket
      - Un ticket possède un titre, une description et un état (non résolu, résolu)
    - L'utilisateur et le développeur ne voient que les applications auxquelles ils sont affectés et leurs tickets. L'admin voit tout
  - Features attendues :
    - Authentification (Breeze)
    - 3 rôles, admin, développeur, utilisateur (laravel-permissions)
    - Recherche full-text pour les projets/tickets (Scout)
    - Reporting journalier des tickets au format CSV (task scheduling + job + file storage)

## Objectifs pédagogiques

### ! INFO

Extrait du programme national de BUT Informatique (2022) - [PDF](#)

## Développement WEB

1.3.1 - page 182

### Compétences ciblées :

- Développer – c'est-à-dire concevoir, coder, tester et intégrer – une solution informatique pour un client.

- Proposer des applications informatiques optimisées en fonction de critères spécifiques : temps d'exécution, précision, consommation de ressources..
- Installer, configurer, mettre à disposition, maintenir en conditions opérationnelles des infrastructures, des services et des réseaux et optimiser le système informatique d'une organisation
- Concevoir, gérer, administrer et exploiter les données de l'entreprise et mettre à disposition toutes les informations pour un bon pilotage de l'entreprise SAÉ au sein de laquelle la ressource peut être mobilisée et combinée :
- SAÉ 3.Deploi.01 | Création et déploiement de services applicatifs

### **Descriptif :**

L'objectif de cette ressource est de poursuivre l'apprentissage de la programmation autour de technologies web. Cette ressource met en situation de développement à partir de spécification, ce qui est la suite logique de l'apprentissage du développement. Savoirs de référence étudiés

- Programmation web (par ex. : côté client ou côté serveur, gestion des contextes, authentications, services web...)
- Sensibilisation à la sécurité web (par ex. : injection, filtrage...)
- Sensibilisation à la sécurité des applications (par ex. : encodage des mots de passe, typage des saisies...)

### **Prolongements suggérés**

- Initiation aux patrons d'architectures (par ex. : MVC...)

### **Apprentissages critiques ciblés :**

- AC21.01 | Élaborer et implémenter les spécifications fonctionnelles et non fonctionnelles à partir des exigences
- AC21.02 | Appliquer des principes d'accessibilité et d'ergonomie
- AC21.03 | Adopter de bonnes pratiques de conception et de programmation
- AC22.03 | Comprendre les enjeux et moyens de sécurisation des données et du code
- AC23.03 | Sécuriser les services et données d'un système
- AC24.03 | Organiser la restitution de données à travers la programmation et la visualisation
- AC24.04 | Manipuler des données hétérogènes

**Mots clés :** Programmation web – Spécifications – Sécurité

**Volume horaire :** Volume horaire défini nationalement : 26 heures dont 20 heures de TP

## Architecture logicielle

2.3.1 - Page 204

Compétences ciblées :

- Développer — c'est-à-dire concevoir, coder, tester et intégrer — une solution informatique pour un client.
- Installer, configurer, mettre à disposition, maintenir en conditions opérationnelles des infrastructures, des services et des réseaux et optimiser le système informatique d'une organisation
- Acquérir, développer et exploiter les aptitudes nécessaires pour travailler efficacement dans une équipe informatique SAÉ au sein de laquelle la ressource peut être mobilisée et combinée :
- SAÉ 4.Deploi.01 | Déployer et sécuriser des services dans un réseau

**Descriptif :** L'objectif de cette ressource est de présenter des composants de la programmation qui peuvent être utilisés dans plusieurs domaines.

### Savoirs de référence étudiés

- Patrons d'architecture (par ex. : MVC, MVVM...)
- Utilisation de briques logicielles, d'interfaces de programmation, de bibliothèques tierces
- Développement de services web

### Prolongements suggérés

- Utilisation de services web (par ex. : requêtes asynchrones, formats d'échange de données...)
- Organisation de l'accès aux données : base de données, annuaires, services Web...

### Apprentissages critiques ciblés :

- AC21.01 | Élaborer et implémenter les spécifications fonctionnelles et non fonctionnelles à partir des exigences
- AC21.02 | Appliquer des principes d'accessibilité et d'ergonomie



- AC21.03 | Adopter de bonnes pratiques de conception et de programmation
- AC23.01 | Concevoir et développer des applications communicantes
- AC26.02 | Appliquer une démarche pour intégrer une équipe informatique au sein d'une organisation

**Mots clés :** Services web – Bibliothèques – Patrons d'architecture – Accès aux données

**Volume horaire :** Volume horaire défini nationalement : 32 heures dont 16 heures de TP

# index

Cours de PHP - PDF [Lien](#)



# Présentation de la matière et des objectifs

La matière s'oriente et s'organise autour du développement, de la conception et du déploiement d'applications WEB sécurisées. Elle s'imprègne entièrement du parcours DACS (Déploiement d'Applications Communicantes et Sécurisées). À travers ce cours, nous apprendrons PHP, célèbre langage du WEB, ainsi qu'un de ses frameworks "Laravel". Vous serez également sensibilisés à la sécurité WEB ainsi qu'au déploiement autonome d'application.

Plusieurs types de titres indiquant le type de contenu :

- 📖 Cours théorique
- ⚙️ Cours pratique
- 📚 TD/TP
- 📝 Contrôle/QCM

## ! INFO

Des commentaires avec les instructions seront présents tout au long des TD/TP pour vous guider sur les réponses à fournir.

Par exemple :

```
<?php

$ma_var = "toto";

// Ici, il faut afficher $ma_var
...
```

## ! INFO

Quand le commentaire suivant est présent dans un fichier, **vous devez fournir un lien vers l'endroit où vous avez trouvé la réponse.**

```
/**  
 * Lien : https://mon_lien.com  
 **/
```

Généralement, vous pouvez trouver la réponse sur 3 sites différents :

- Ce site, en cliquant sur le titre de la partie, vous pourrez récupérer son lien. Par exemple : [https://phd.julien-cpsn.com/courses/PHP/seance\\_1#remplacer-une-partie-dun-string-avec-replace](https://phd.julien-cpsn.com/courses/PHP/seance_1#remplacer-une-partie-dun-string-avec-replace)
- Sur la documentation officielle de PHP. Par exemple : <https://www.php.net/manual/en/function.str-replace.php>
- Un blog, un forum, ou n'importe quel autre site en ligne (donc pas ChatGPT :)

### DANGER

**Tout commentaire non rempli sera considéré non rendu.**

- Si le travail à rendre est **non-noté**, je viendrai vous poser la question.
- Si le travail à rendre est **noté**, la réponse sera comptée comme fausse.

### ASTUCE

**Vous recevrez la liste des éléments à réviser un peu de temps avant l'examen. Si j'oublie, n'hésitez pas à me le rappeler.**



# | Introduction à PHP

PHP, acronyme de "PHP: Hypertext Preprocessor", est un langage de script principalement conçu pour le développement web. PHP a vu le jour en 1995 avec sa première version créée par Rasmus Lerdorf, et depuis, il a considérablement évolué pour devenir l'un des langages de script les plus utilisés pour le développement web. Écrit en C, PHP tire parti de la performance et de la flexibilité offertes par ce langage de bas niveau, tout en fournissant une syntaxe plus simple et accessible aux développeurs web. La version actuelle, 8.3.9, intègre de nombreuses améliorations et nouvelles fonctionnalités par rapport à ses prédécesseurs, telles que des optimisations de performance, des améliorations de sécurité, et des fonctionnalités de programmation modernes. Grâce à ces mises à jour régulières, PHP continue de s'adapter aux besoins évolutifs du développement web moderne et reste un choix pertinent pour le développement côté serveur.

## Scripting

Contrairement aux langages de programmation traditionnels qui nécessitent la création de programmes complets, un langage de script comme PHP est utilisé pour écrire de petits scripts ou fichiers qui exécutent des actions spécifiques sur un serveur web.

## Interprété

PHP est **interprété**, c'est-à-dire que le code est **exécuté ligne par ligne par un interpréteur**, contrairement aux langages compilés (comme C, C++ ou Java) où le code source est transformé en code machine/binaire avant d'être exécuté.

## Multi-plateforme

PHP est **multi-plateforme**, ce qui signifie qu'il **peut fonctionner sur différents systèmes d'exploitation**, tout comme Java.

## Programmation Orientée Object

Il prend également en charge la **Programmation Orientée Objet (POO)**, permettant la création de classes, d'objets, et l'utilisation de concepts comme l'héritage et le polymorphisme.

### Faiblement typé

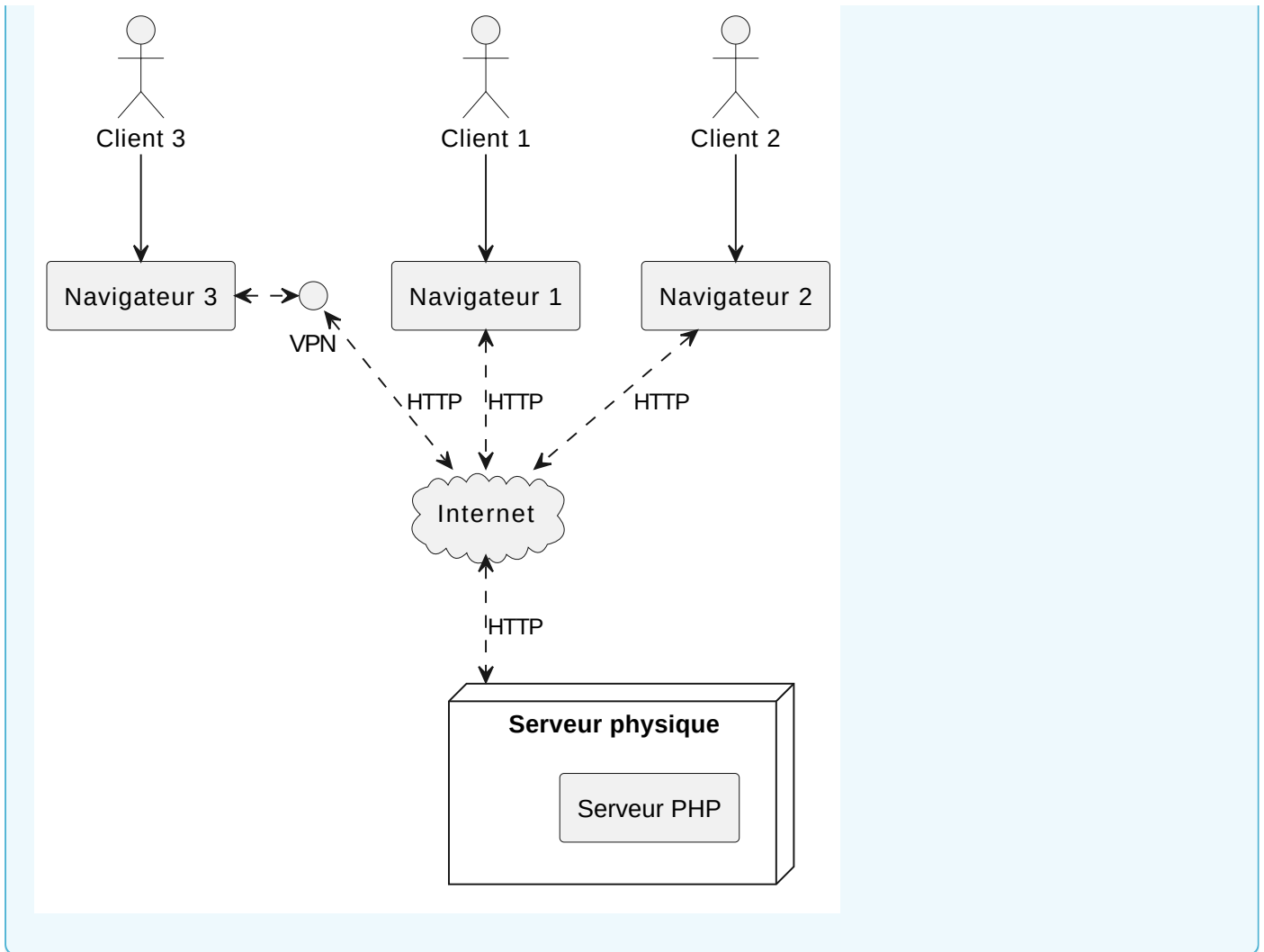
---

PHP est faiblement typé (ou typé de manière souple), ce qui signifie que les types de données des variables ne sont pas strictement définis et peuvent être changés dynamiquement (ex: un int peut devenir un string), facilitant ainsi l'écriture du code mais pouvant parfois conduire à des erreurs subtiles.

### Execution côté serveur

---

À noter que PHP s'exécute uniquement côté serveur : **le code est interprété sur le serveur**, puis le **résultat est renvoyé au client** (navigateur web), ce qui en fait un langage idéal pour la création de pages web dynamiques.



## Coder en PHP

### 💡 ASTUCE

Toujours garder la [documentation de PHP](#) près de soi

- Extension ".php"
- Les variables ont toutes un \$, ex: `$ma_var`
- Point virgule à la fin des lignes
- Convention de nommage snake\_case, ex `ma_super_fonction`
- Tous les fichiers commencent avec `<?php`, et peuvent finir par `?>`
- Il est possible de mettre des balises `<?php . . . ?>` dans du HTML (toujours dans un fichier `.php`)



```
<div>
  <p>Mon paragraphe</p>
  <div><?php echo $ma_var; ?></div>
</div>
```

# Différences de PHP

## Echo

Print avec `echo` (ne pas oublier d'ajouter un `\n`, car il ne retourne pas à la ligne)

```
$ma_var = "toto";
echo $ma_var;
echo "Salut c'est $ma_var\n";
```

totoSalut c'est toto

## Debug

Debugguer une variable avec `var_dump()`. Cela permet d'afficher le contenu complet de la variable au moment où est appelée la fonction pour régler d'éventuels problèmes.


```
$ma_var = [1, 2, 5];
var_dump($ma_var);
```

array(3) { [0]=> int(1) [1]=> int(2) [2]=> int(5) }

## For each

Le `for each` ("pour chaque")


```
$mes_valeurs = [1, 2, 3, 4];  
  
foreach ($mes_valeurs as $var) {  
    echo "$var\n";  
}
```

 Sortie

```
1  
2  
3  
4
```

Il est aussi possible d'utiliser une clé

```
$mes_valeurs = [10, 11, 12, 13];  
  
foreach ($mes_valeurs as $index => $var) {  
    echo "$index: $var\n";  
}
```


 Sortie

## Match

Plus récent, le `match`

```
$ma_var = 2;  
  
$mon_autre_var = match ($ma_var) {  
    1, 2 => "toto",  
    $ma_var > 10 => "tata",  
    default => "cas par défaut"  
};
```


```
echo $mon_autre_var;
```

 Sortie

## Tableaux associatifs

Pour créer des tableaux avec des index personnalisés (comme un dictionnaire)


```
$ma_var = [  
    0 => "toto",  
    "mon index 1" => "tata",  
    "mon autre index" => "tonton"  
];  
  
echo $ma_var["mon autre index"];
```

 Sortie

## Opérateur "." pour ajouter des strings

Pour ajouter/concatener des strings en PHP, on utilise l'opérateur `.`

```
echo "mon string 1" . "mon string 2";
```

 Sortie



# | TD 1 : Premiers scripts

## Cloner le projet

Cloner le projet <https://github.com/PHP-BUT2-DACS/TP-TD>

```
git clone https://github.com/PHP-BUT2-DACS/TP-TD.git
```

### ⚠ ATTENTION

**Ce projet vous servira tout le long du module.** Gardez le précieusement !

## Faire les exercices

1. Dans un terminal, exécutez la commande `php -v` (ou `php --version`) pour être informé de la version de votre interpréteur PHP.
2. Ensuite, naviguez jusqu'au dossier du projet `/Séance 1/TD1`. Puis démarrez le serveur PHP avec la commande suivante :

```
php -S localhost:8080
```

Comprendre la commande

3. Ouvrir le navigateur à l'URL <http://localhost:8080/exo1.php>
4. Une fois la commande lancée et l'URL ouverte, votre terminal devrait afficher quelque-chose de similaire :

### 💻 SORTIE

```
PHP DACS/Séance 1/TD1 on main via
$ php -S localhost:8080
[Tue Sep 3 21:24:35 2024] PHP 8.2.20 Development Server (http://localhost:8080) started
[Tue Sep 3 21:24:44 2024] [::1]:46752 Accepted
[Tue Sep 3 21:24:44 2024] [::1]:46752 [200]: GET /exo1.php
[Tue Sep 3 21:24:44 2024] [::1]:46752 Closing
[Tue Sep 3 21:24:44 2024] [::1]:46758 Accepted
[Tue Sep 3 21:24:44 2024] [::1]:46758 [404]: GET /favicon.ico - No such file or directory
[Tue Sep 3 21:24:44 2024] [::1]:46758 Closing
```

5. Répondez aux exercices du TD1. Vous pouvez demander de l'aide si nécessaire.

#### ASTUCE

La commande `php -h` (ou `php --help`) pourra vous aider si vous êtes perdus avec d'autres commandes de PHP.

# | Fonctions utilitaires & autres outils

## Utilitaires pour les string (chaînes de caractères)

[Lien vers la documentation de PHP](#)


### String en majuscule ou minuscule

```
$ma_var = "toto";  
echo strtoupper($ma_var) . "\n";  
  
$ma_var = "TATA";  
echo strtolower($ma_var);
```

 Sortie

### Séparer avec explode et regrouper avec implode

```
$ma_var = "toto;tata;tonton";  
  
// Séparer un string vers un tableau  
$mon_tableau = explode(";", $ma_var);  
var_dump($mon_tableau);  
  
// Regrouper des éléments d'un tableau vers un string  
$mon_autre_var = implode(";", $mon_tableau);  
echo $mon_autre_var;
```

 Sortie

## Supprimer les espaces en trop avec trim

```
$ma_var = "  toto  ";  
$mon_autre_var = trim($ma_var);  
echo $ma_var;
```



Sortie

Il existe aussi ltrim (left trim) pour supprimer les espaces seulement au début, et rtrim (right trim) pour supprimer les espaces seulement à la fin.

## Remplacer une partie d'un string avec replace

```
$ma_var = "toto";  
$mon_autre_var = str_replace("o", "a", $ma_var);  
echo $ma_var;
```



Sortie

## Utilitaires pour les int et float (nombres)

### Transformer une variable en int ou float

```
$ma_var = "10";  
$mon_int = intval($ma_var);  
var_dump($ma_var);
```



Sortie

Même principe avec `floatval()`

## Autres utilitaires

## Checker le type d'une variable

- `is_array()` — Détermine si une variable est un tableau
- `is_bool()` — Détermine si une variable est un booléen
- `is_callable()` — Détermine si une valeur peut être appelé comme une fonction dans la portée courante.
- `is_countable()` — Vérifie si le contenu de la variable est une valeur dénombrable
- `is_float()` — Détermine si une variable est de type nombre décimal
- `is_int()` — Détermine si une variable est de type nombre entier
- `is_numeric()` — Détermine si une variable est un nombre ou une chaîne numérique

```
var_dump(is_int(23));  
var_dump(is_int("23"));
```

 Sortie

## Checker si une variable existe (et si elle est vide)

`isset()` détermine si une variable est considérée définie, ceci signifie qu'elle est déclarée et est différente de `null`.

```
$est_ce_que_ma_var_existe = isset($ma_var)  
var_dump($est_ce_que_ma_var_existe)
```

 Sortie

`empty()` détermine si une variable est considérée définie et si sa valeur équivaut à `false`.

```
$mon_tableau = ["toto", "tata", false]  
  
$est_ce_que_toto_existe = empty($mon_tableau[0])  
$est_ce_que_tonton_existe = empty($mon_tableau[2])  
var_dump($est_ce_que_toto_existe)  
var_dump($est_ce_que_tonton_existe)
```





Sortie



# | TD 2 : Utilitaires & outils

## Cloner le projet

Si ce n'est pas déjà fait, cloner le projet <https://github.com/PHP-BUT2-DACS/TP-TD>

```
git clone https://github.com/PHP-BUT2-DACS/TP-TD.git
```

## Faire les exercices

1. Lancer le serveur PHP dans le dossier /Séance 1/TD2

```
php -S localhost:8080
```

2. Répondez aux exercices du TD2. Vous pouvez demander de l'aide si nécessaire.