Cours 01 - Modalités de l'unité d'enseignement et introduction à PHP

https://github.com/heig-vd-progserv1-course

Support de cours · Présentation (web) · Présentation (PDF)

L. Delafontaine, avec l'aide de GitHub Copilot.

Bienvenue à l'unité d'enseignement Programmation serveur 1 (ProgServ1) !

Qui suis-je

Ludovic Delafontaine

E-mail · GitHub

Mes objectifs et souhaits pour ProgServ1

PHP va vous accompagner tout au long de vos études à la HEIG-VD (ProgServ1, ProgServ2, DevProdMed, etc.) et aussi plus tard dans votre vie professionnelle.

Mon objectif est de vous donner des bases solides et une bonne compréhension de ce language pour vos études et pour la suite.

Si quelque chose ne convient pas dans mon cours, n'hésitez pas à me le dire. Je suis ouvert à toutes critiques pour améliorer mon enseignement.

Comment me contacter

Selon vos préférences, vous pouvez utiliser l'un des canaux suivants pour toutes questions relatives à l'unité d'enseignement :

- En personne, durant les sessions de cours ou en dehors
- Par e-mail (ludovic.delafontaine@heig-vd.ch)
- Microsoft Teams
  - Dans le canal Teams de l'unité d'enseignement (de préférence - n'hésitez pas à vous entraider si je ne suis pas disponible)
  - Message privé sur Teams (à éviter si possible)

Retrouvez plus de détails dans le support de cours

Cette présentation est un résumé du support de cours. Pour plus de détails, consultez le support de cours.

Objectifs (1/3)

- Lister les objectifs de l'unité d'enseignement
- Lister les modalités d'organisation de l'unité d'enseignement
- Lister les modalités d'évaluation
- Expliquer le concept d'architecture client-serveur
- Lister les outils nécessaires pour écrire et exécuter du code PHP

Objectifs (2/3)

- Expliquer comment PHP fonctionne dans un environnement web
- Expliquer la syntaxe de base de PHP
- Expliquer les variables en PHP
- Expliquer les constantes en PHP
- Expliquer la nature dynamique des variables et constantes en PHP
- Expliquer les opérateurs en PHP

Objectifs (3/3)

- Expliquer les structures de contrôle conditionnelles en PHP
- Rédiger du code PHP simple

Modalités de l'unité d'enseignement

Objectifs de l'unité d'enseignement (1/2)

Selon la fiche d'unité, à la fin de cette unité d'enseignement, vous devriez être capable de :

“
- *Connaître les principes de base de la programmation serveur*
- *Manipuler des tableaux associatifs complexes*
- *Générer des documents Web dynamiquement*
- *Gérer la persistance des données applicatives dans un Système de Gestion de Base de Données (SGBD)*

”

Objectifs de l'unité d'enseignement (2/2)

En résumé, vous devriez être capable de :

- Comprendre les bases de PHP et son rôle dans le monde web.
- Écrire un code PHP propre et organisé.
- Gérer les formulaires HTML et les données qu'ils contiennent de manière sûre.
- Persister des données dans une base de données SQLite.
- Implémenter des concepts de programmation orientée objet.
- Gérer les cookies et les sessions utilisateurs.

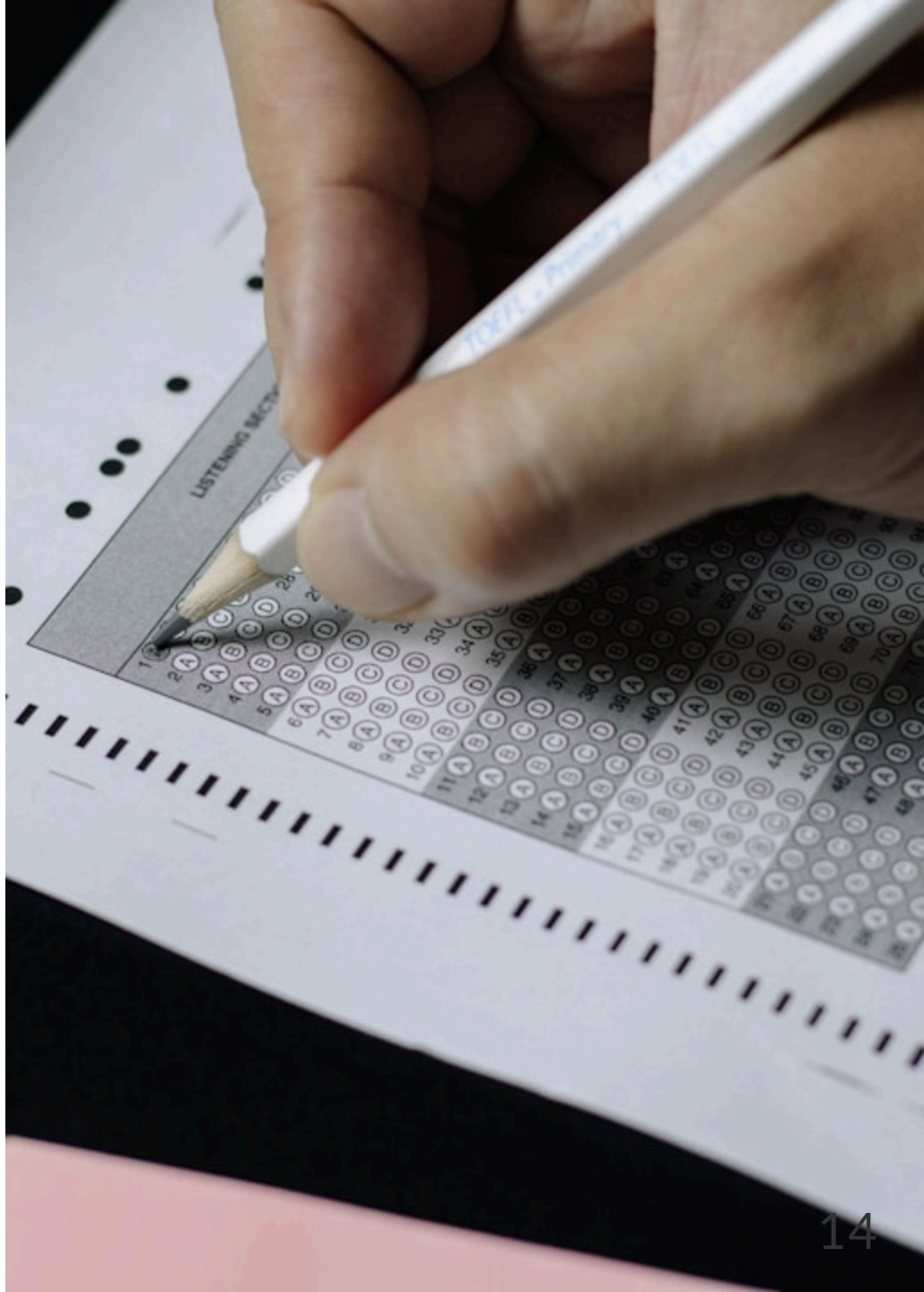
Modalités d'organisation de l'unité d'enseignement

- En présentiel chaque semaine dans cette même salle.
- Mélange de théorie et de pratique pour un meilleur apprentissage :
  - Moment de théorie court pour expliquer les concepts.
  - Mini-projet à réaliser tout au long de l'unité d'enseignement.
  - Exercices à faire en classe ou à la maison.
- Espace de discussion pour poser des questions et obtenir de l'aide (**il n'y a pas de questions bêtes !**, je suis payé pour ça).

Modalités d'évaluation

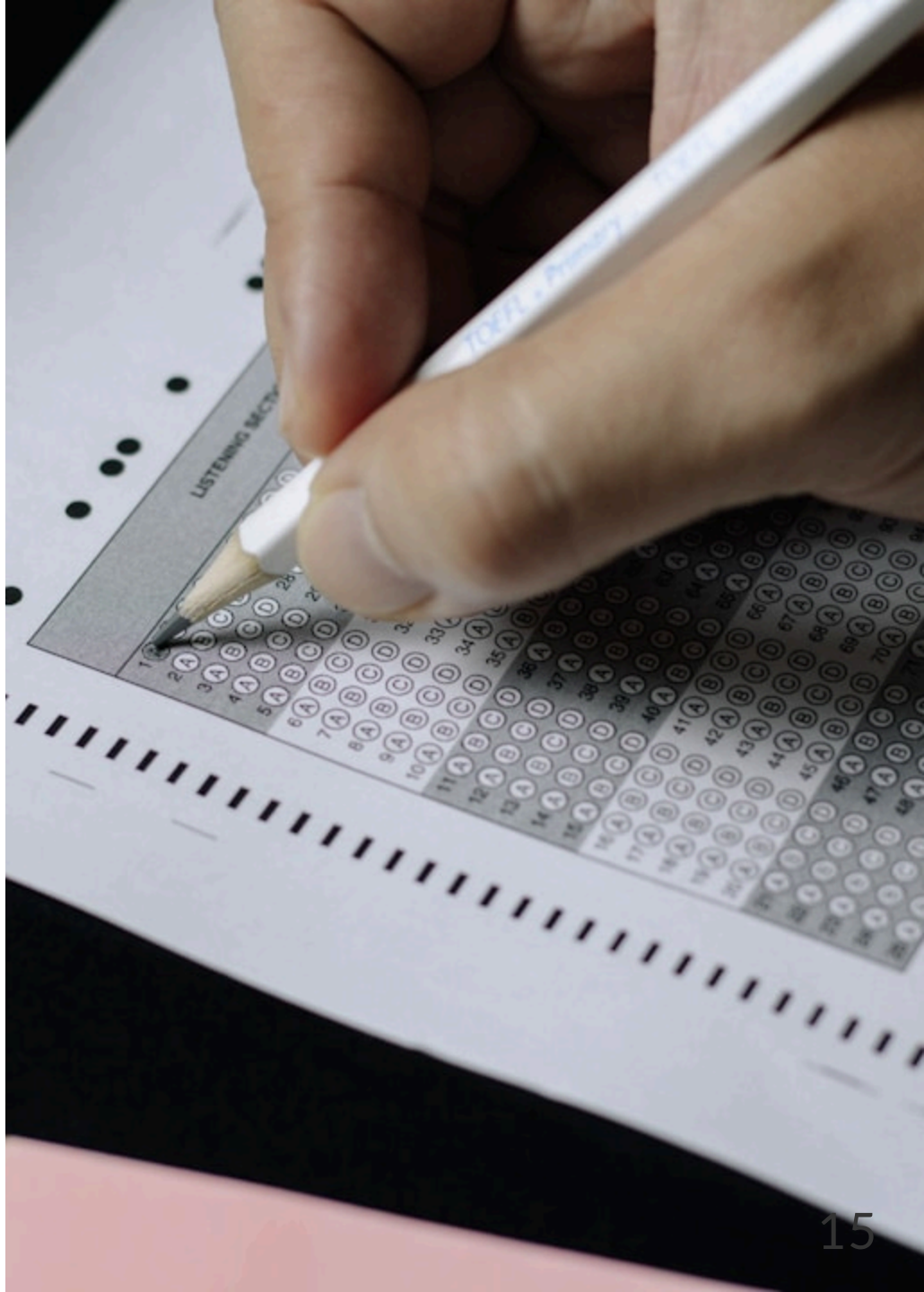
Le cours sera évalué à l'aide d'un seul examen final composé de deux parties, à effectuer sur ordinateur :

- Partie théorique
  - 40% de la note finale
- Partie pratique
  - 60% de la note finale



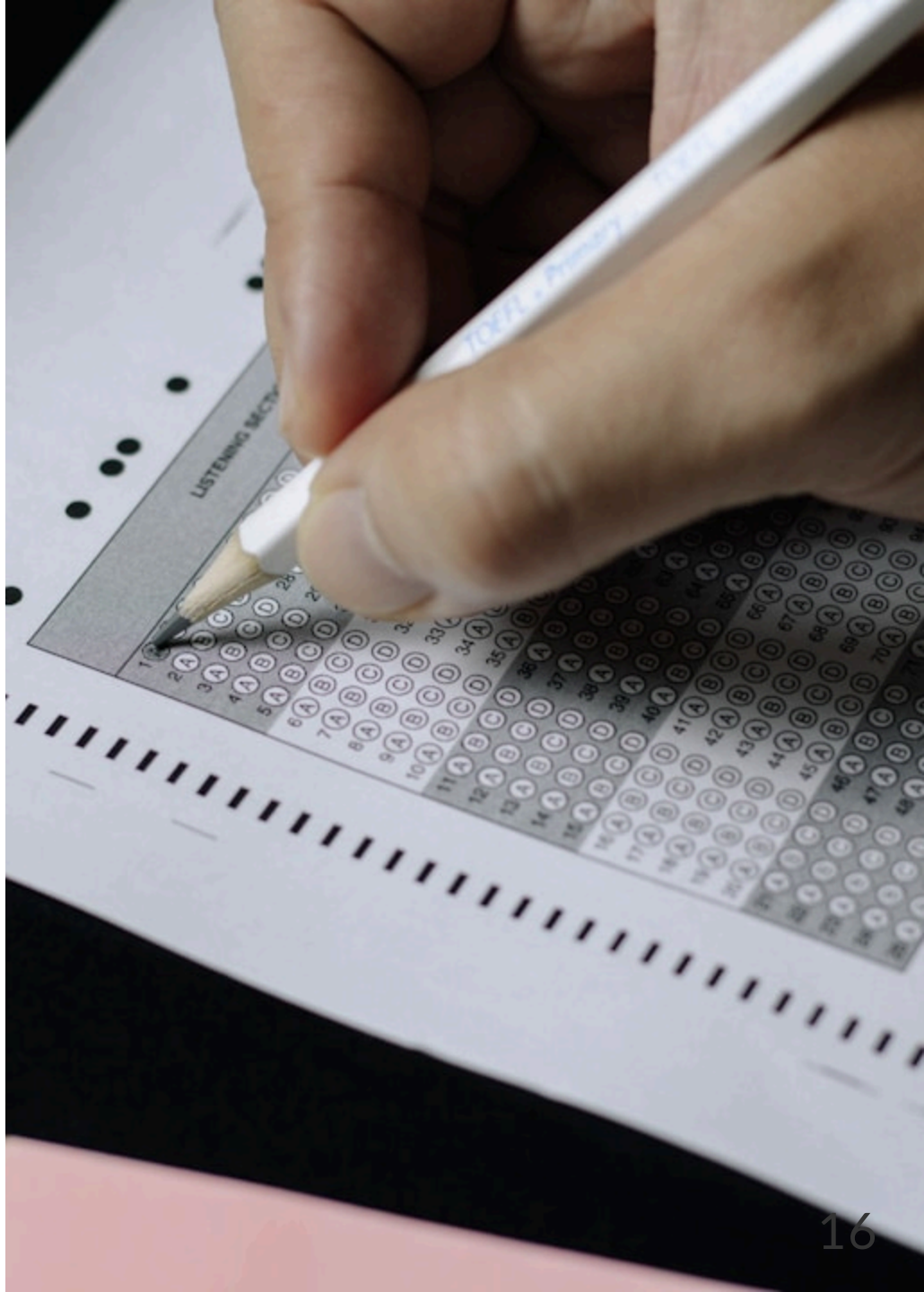
Partie théorique

- Évaluation sur :
  - Les connaissances théoriques acquises tout au long de l'unité d'enseignement
  - Les exercices
- Durée d'environ 45 minutes
- Devrait utiliser la plateforme d'évaluation en ligne de la HEIG-VD
- **Aucune aide autorisée**



Partie pratique

- Évaluation sur :
  - Les exercices
  - Le mini-projet
- Durée d'environ 2h15
- Petit projet à réaliser
- Contenus du cours, notes personnelles, php.net et developer.mozilla.org autorisés
- **Toute autre aide interdite**



La programmation et l'anglais

Le domaine de la programmation est très largement anglophone. La majorité des ressources que vous trouverez dans votre carrière sont en anglais.

Dans le but de vous préparer à cette réalité, les exemples de code que nous utiliserons dans les cours seront en anglais (commentaires en français par contre).

Le reste du cours restera néanmoins en français. Si l'anglais est une barrière pour vous, n'hésitez pas à me le faire savoir.

Bibliographie et ressources utilisées

- https://www.php.net/manual/index.php
- https://developer.mozilla.org
- https://phptherightway.com/
- https://www.w3schools.com/php/
- https://github.com/ziadoz/awesome-php

Introduction à PHP

Qu'est-ce que PHP (1/2)

- Jusqu'ici, vous avez développé des applications Java qui s'exécutent sur une seule machine.
- Avec PHP, plusieurs personnes vont pouvoir accéder à une application depuis leur navigateur web.

Qu'est-ce que PHP (2/2)

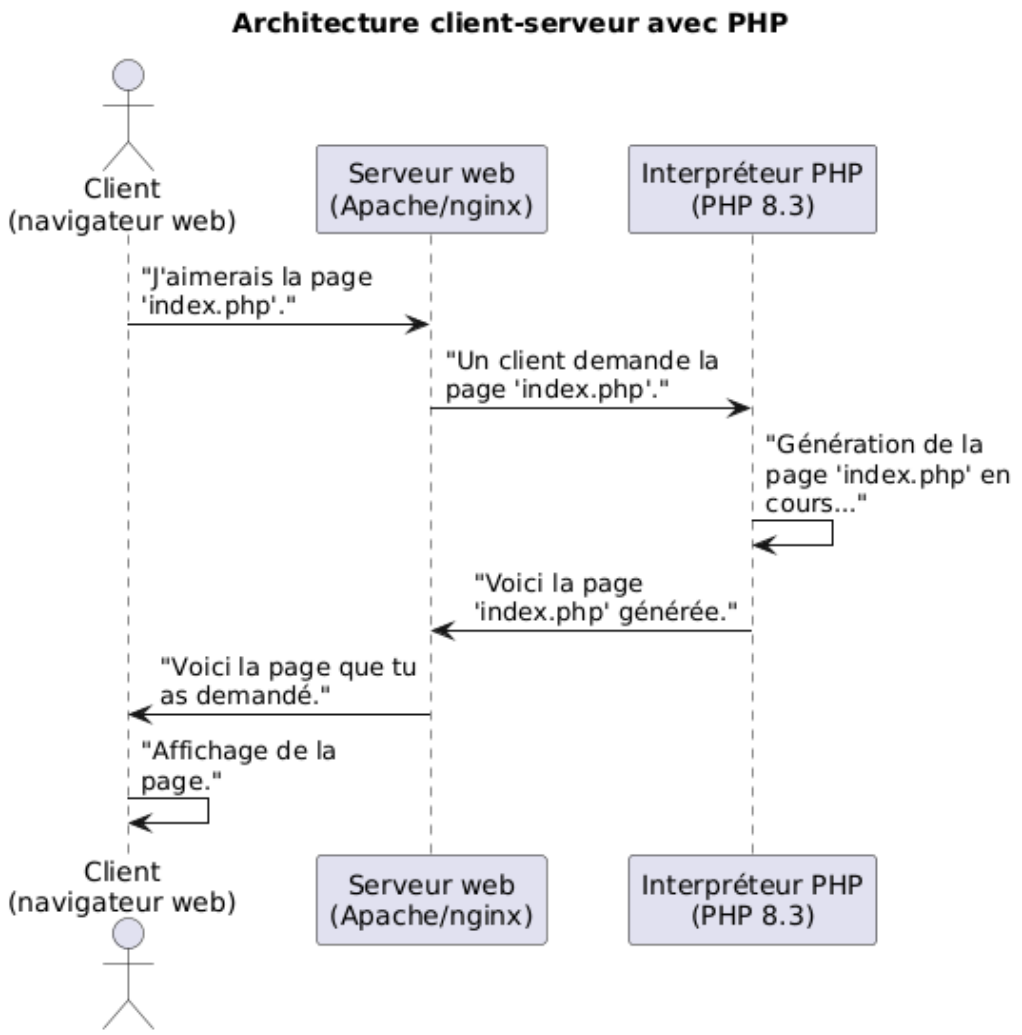
- PHP est un language de programmation datant de 1994.
- Très utilisé pour le développement web.
- Basé sur une architecture client-serveur.
- Actuellement à la version 8.4.
- Un des languages les plus utilisés pour le développement web.

Applications web et architecture client-serveur

- La plupart des applications web modernes reposent sur une architecture dite *"client-serveur"* :
  1. Un client (navigateur web) envoie une requête à un serveur.
  2. Le serveur répond aux requêtes des différents clients.
  3. Le client affiche le résultat de la requête.
- PHP repose sur cette même architecture.

Comment fonctionne PHP

- PHP fonctionne grâce aux outils suivants :
  - Un serveur web.
  - PHP installé sur le serveur web.
  - Un navigateur web.
  - Un éditeur de code (pour le développement).



Comment écrire du code PHP (1/3)

- Code PHP dans un fichier `.php`.
- Le code PHP est écrit entre des balises `<?php` et `?>`.
- Le code PHP peut être mélangé avec du HTML.
- Uniquement `<?php` s'il n'y a que du code PHP.

Comment écrire du code PHP (2/3)

```php
<?php
// Code PHP, dans un fichier `.php`
// Ici, il n'y a pas de balise de fermeture PHP
echo "Hello, World!";
```

```java
// Équivalent en Java
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Comment écrire du code PHP (3/3)

```
<!-- Code PHP, dans un fichier `.php` avec de l'HTML -->
<!-- Il y a la balise de fermeture PHP -->
<!DOCTYPE html>
<html>
<head>
    <title>PHP Test</title>
</head>
<body>
    <h1><?php echo "Hello, World!"; ?></h1>
</body>
</html>
```

Comment exécuter du code PHP

- Il faut avoir un serveur web avec PHP installé sur votre machine
- Heureusement, il existe des solutions toutes faites pour cela :
  - WampServer
  - MAMP
  - XAMPP

Syntaxe de base de PHP

- Similaire à Java, JavaScript et d'autres languages de programmation.
- Comme n'importe quelle langue ou language de programmation, PHP a ses propres règles de syntaxe.
- Il s'agit de les apprendre et de les comprendre pour lire et écrire du code PHP de manière efficace.

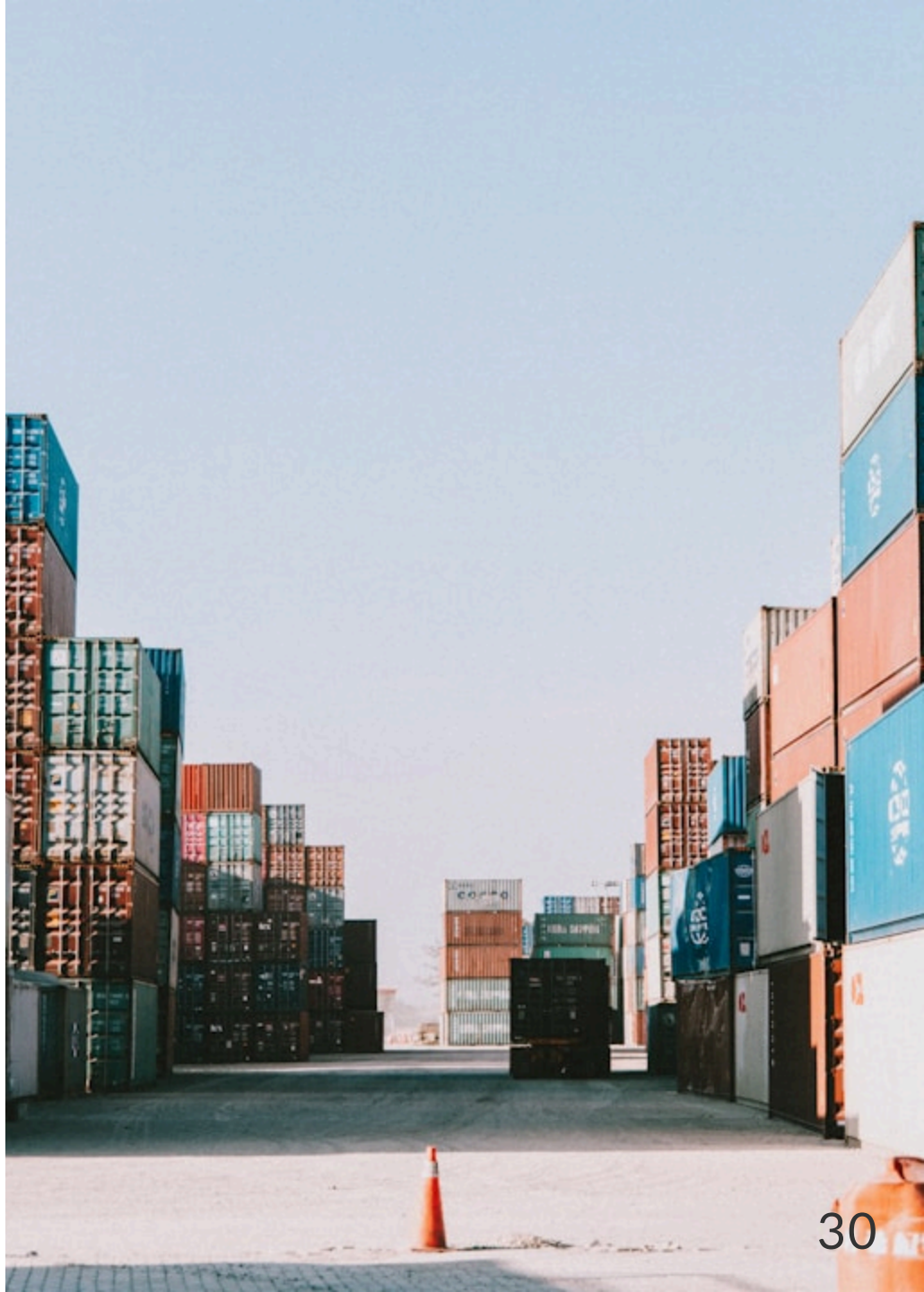
Les commentaires

- Comme dans Java, `//` pour un commentaire sur une seule ligne
- Comme dans Java, `/* ... */` pour un commentaire sur plusieurs lignes

```php
<?php
// Ceci est un commentaire sur une seule ligne

/*
Ceci est un commentaire
sur plusieurs lignes
*/
```

Les variables (1/3)

- Les variables sont des conteneurs pour stocker des données.
- Les variables sont déclarées avec le signe `$`.
- Les variables peuvent contenir des chaînes de caractères, des nombres, des booléens, des tableaux, etc.



Les variables (2/3)

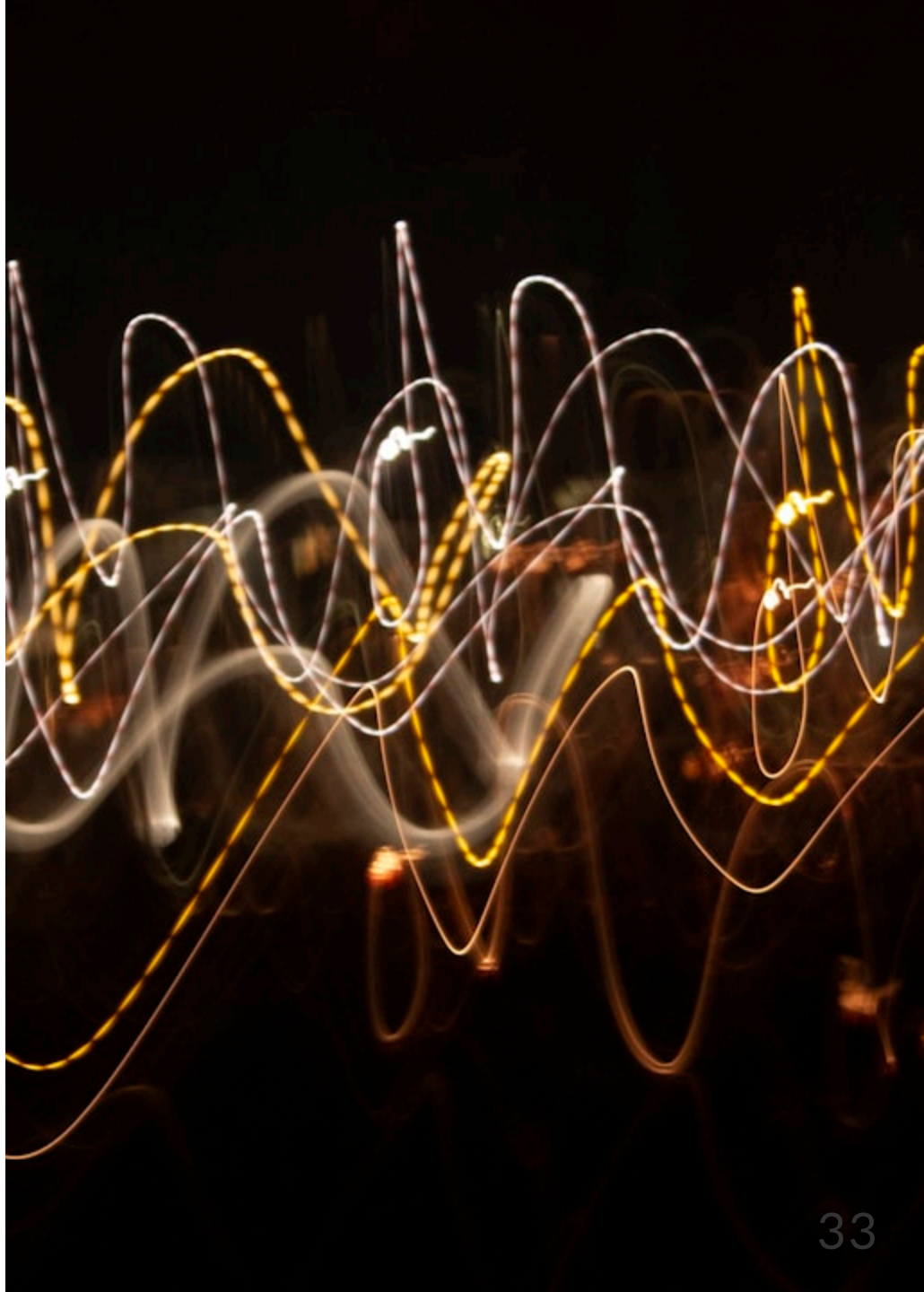
```php
<?php
// Déclaration d'une variable - une variable commence par le signe `$`
$variable = "Hello, World!";

// Affichage de la variable
echo $variable;

// Modification de la variable
$variable = "Goodbye, World!";

// Affichage de la variable modifiée
echo $variable;
```

Les variables (3/3)

```java
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'une variable
    String variable = "Hello, World!";

    // Affichage de la variable
    System.out.print(variable);

    // Modification de la variable
    variable = "Goodbye, World!";

    // Affichage de la variable modifiée
    System.out.print(variable);
}
```

Type de données et typage dynamique (1/3)

- PHP est un language de programmation à typage dynamique.
- Il n'y a pas besoin de déclarer le type de données d'une variable.
- Le type de données d'une variable est déterminé par la valeur qui lui est assignée.



Type de données et typage dynamique (2/3)

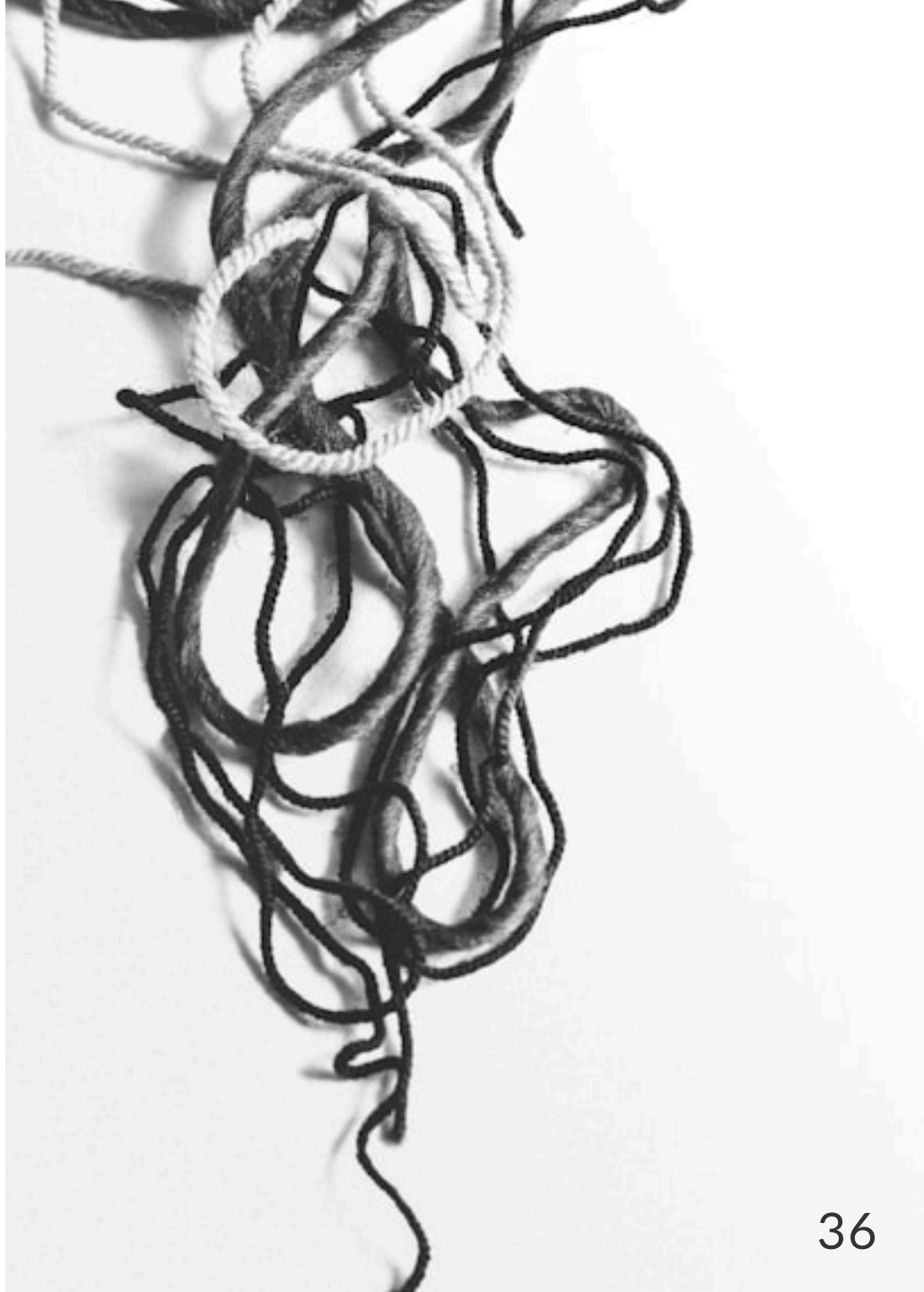
```php
<?php
// Variable de type chaîne de caractères
$variable = "Hello, World!";

// Variable de type nombre
$variable = 42;

// Variable de type nombre flottant
$variable = 3.14;

// Variable de type booléen
$variable = true;
```

Type de données et typage dynamique (3/3)

```java
// Équivalent en Java
public static void main(String[] args) {
    // Variable de type chaîne de caractères
    String variable1 = "Hello, World!";

    // Variable de type nombre
    int variable2 = 42;

    // Variable de type nombre flottant
    double variable3 = 3.14;

    // Variable de type booléen
    boolean variable4 = true;
}
```

Les chaînes de caractères (1/3)

- Les chaînes de caractères sont déclarées entre des guillemets simples (`'`) ou doubles (`"`).
- Permettent de créer des mots ou des phrases.
- Pour concaténer des chaînes de caractères, on utilise le point (`.`).



Les chaînes de caractères (2/3)

```php
<?php
$string = "Hello, World!";

echo $string;
```

```java
// Équivalent en Java
public static void main(String[] args) {
    String string = "Hello, World!";

    System.out.print(string);
}
```

Les chaînes de caractères (3/3)

```php
<?php
$first = "Hello, ";
$second = "World!";

echo $first . $second;
```

```java
// Équivalent en Java
public static void main(String[] args) {
    String first = "Hello, ";
    String second = "World!";

    System.out.print(first + second);
}
```

Les nombres (1/3)

- Les nombres sont déclarés sans guillemets.
- Il existe deux types de nombres en PHP : les entiers et les flottants.
- Les entiers sont des nombres entiers positifs ou négatifs.
- Les flottants sont des nombres à virgule, eux aussi, positifs ou négatifs.

Les nombres (2/3)

```php
<?php
// Entier
$integer = 42;

// Flottant
$float = 3.14;

// Affichage des nombres
echo "\$integer contains $integer<br>";
echo "\$float contains $float";
```

Les nombres (3/3)

```java
// Équivalent en Java
public static void main(String[] args) {
    // Entier
    int myInteger = 42;

    // Flottant
    double myFloat = 3.14;

    // Affichage des nombres
    System.out.println("myInteger contains " + myInteger);
    System.out.print("myFloat contains " + myFloat);
}
```

Les booléens (1/3)

- Les booléens sont des valeurs qui peuvent être soit vraies (`true`) soit fausses (`false`).
- Les booléens sont déclarés en PHP avec les mots-clés `true` et `false` .
- Les booléens sont souvent utilisés pour des conditions dans les structures de contrôle conditionnelles.

Les booléens (2/3)

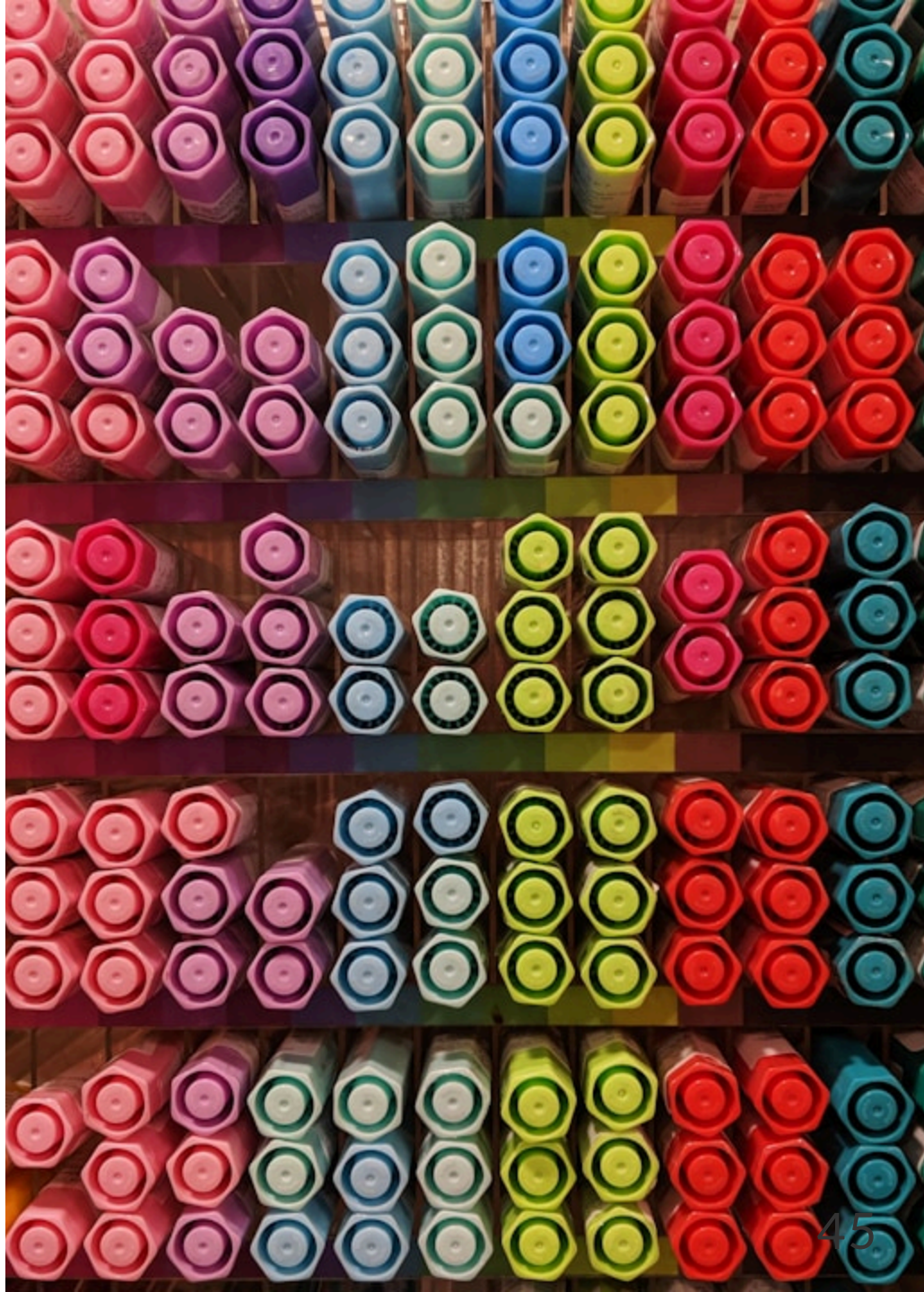
```php
<?php
// Vrai
$doILikeDogs = true;

// Faux
$doILikeHomework = false;

// Affichage des booléens - `false` est affiché comme une chaîne vide
echo "\$doILikeDogs contains $doILikeDogs<br>";
echo "\$doILikeHomework contains $doILikeHomework";
```

Les booléens (3/3)

```java
// Équivalent en Java
public static void main(String[] args) {
    // Vrai
    boolean doILikeDogs = true;

    // Faux
    boolean doILikeHomework = false;

    // Affichage des booléens
    System.out.println("doILikeDogs contains " + doILikeDogs);
    System.out.print("doILikeHomework contains " + doILikeHomework);
}
```

Les tableaux (1/3)

- Les tableaux sont des collections de valeurs.
- Les tableaux sont déclarés entre des crochets (`[]`) ou avec la fonction `array()` .
- Les valeurs d'un tableau sont indexées à partir de 0.
- Nous étudierons les tableaux plus en détails dans un prochain cours.



Les tableaux (2/3)

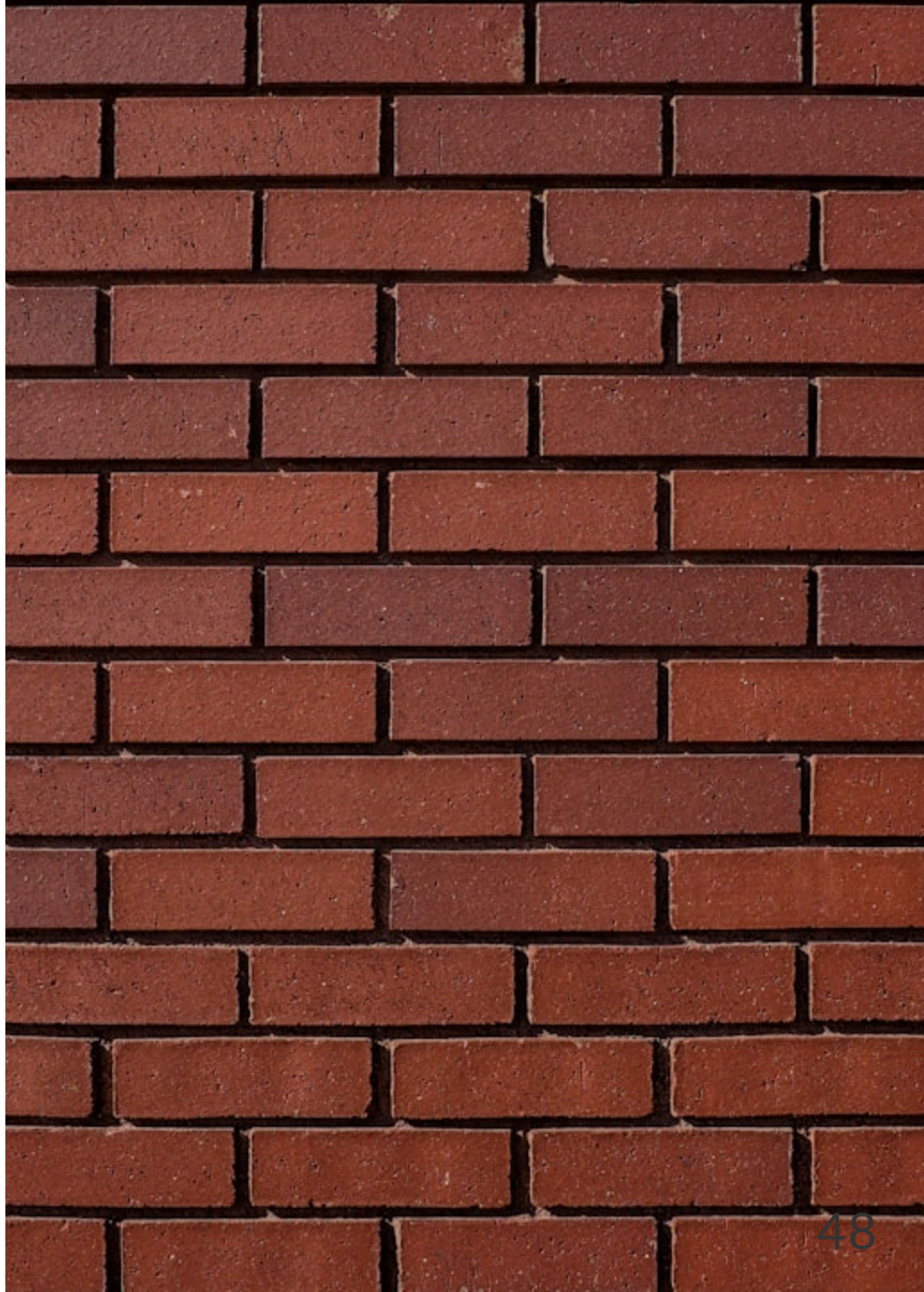
```php
<?php
// Déclaration d'un tableau
$array = ["apple", "banana", "cherry"];

// Affichage de la première valeur du tableau
// Les tableaux sont indexés à partir de 0
echo "$array[0]<br>";

// Déclaration d'un tableau (alternative)
$array = array("apple", "banana", "cherry");

// Affichage de la troisième valeur du tableau
echo "$array[2]";
```

Les tableaux (3/3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'un tableau
    String[] array = {"apple", "banana", "cherry"};

    // Affichage de la première valeur du tableau
    System.out.println(array[0]);

    // Affichage de la troisième valeur du tableau
    System.out.println(array[2]);
}
```

Les constantes (1/3)

- Les constantes sont des valeurs qui ne peuvent pas être modifiées.
- Les constantes sont déclarées avec le mot-clé `const` ou avec la fonction `define()`.
- La convention veut que les constantes soient écrites en majuscules.



Les constantes (2/3)

```php
<?php
// Définition d'une constante
const CONSTANT = "Hello, World!";

// Affichage de la constante
echo CONSTANT;

// Tentative de modification de la constante (erreur)
CONSTANT = "Goodbye, World!";

// Définition d'une constante (alternative)
define("CONSTANT", "Hello, World!");
```

Les constantes (3/3)

```
// Équivalent en Java
public static void main(String[] args) {
    // Définition d'une constante
    final String CONSTANT = "Hello, World!";

    // Affichage de la constante
    System.out.println(CONSTANT);

    // Tentative de modification de la constante (erreur)
    CONSTANT = "Goodbye, World!";
}
```

Les opérateurs (1/5)

- Permet d'effectuer des opérations sur des variables et des valeurs.
- Opérateurs arithmétiques : `+`, `-`, `*`, `/`, `%` (modulo)
- Opérateurs de comparaison : `==` (égal), `!=` (différent), `>` (supérieur), `<` (inférieur)
- Opérateurs logiques : `&&` (et), `||` (ou), `!` (non/inversion)

Les opérateurs (2/5)

```php
<?php
$sum = 1 + 1; // `$sum` contiendra 2
$difference = $sum - 1; // `$difference` contiendra 1
$product = 2 * 2; // `$product` contiendra 4
$quotient = $product / 2; // `$quotient` contiendra 2

echo "Sum: $sum<br>";
echo "Difference: $difference<br>";
echo "Product: $product<br>";
echo "Quotient: $quotient";
```

Les opérateurs (3/5)

```java
// Équivalent en Java
public static void main(String[] args) {
    int sum = 1 + 1; // `sum` contiendra 2
    int difference = 2 - 1; // `difference` contiendra 1
    int product = 2 * 2; // `product` contiendra 4
    int quotient = 4 / 2; // `quotient` contiendra 2

    System.out.println("Sum: " + sum);
    System.out.println("Difference: " + difference);
    System.out.println("Product: " + product);
    System.out.print("Quotient: " + quotient);
}
```

Les opérateurs (4/5)

```php
<?php
// `$modulo` contiendra 1
// Il reste 1 après la division de 5 par 2, 5 n'est donc pas pair.
$modulo = 5 % 2;

echo "Modulo: $modulo<br>";

// `$modulo` contiendra 0
// Il ne reste rien après la division de 6 par 2, 6 est donc pair.
$modulo = 6 % 2;

echo "Modulo: $modulo";
```

Les opérateurs (5/5)

```java
// Équivalent en Java
public static void main(String[] args) {
    // `modulo` contiendra 1
    // Il reste 1 après la division de 5 par 2, 5 n'est donc pas pair.
    int modulo = 5 % 2;

    System.out.println("Modulo: " + modulo);

    // `modulo` contiendra 0
    // Il ne reste rien après la division de 6 par 2, 6 est donc pair.
    modulo = 6 % 2;

    System.out.print("Modulo: " + modulo);
}
```

Les structures de contrôle conditionnelles (1/7)

- Permettent de contrôler le flux d'exécution d'un programme.
- Utilisent les opérateurs de comparaison et logiques.
- Elles se composent de `if`, `else`, `elseif` et `switch`.

Les structures de contrôle conditionnelles (2/7)

```php
<?php
$a = 1;
$b = 2;

if ($a < $b) {
    echo "a is less than b";
} elseif ($a == $b) {
    echo "a is equal to b";
} else {
    echo "a is greater than b";
}
```

Les structures de contrôle conditionnelles (3/7)

```java
// Équivalent en Java
public static void main(String[] args) {
    int a = 1;
    int b = 2;

    if (a < b) {
        System.out.print("a is less than b");
    } else if (a == b) {
        System.out.print("a is equal to b");
    } else {
        System.out.print("a is greater than b");
    }
}
```

Les structures de contrôle conditionnelles (4/7)

```php
<?php
// Déclaration de deux variables
$age = 18;
$country = "Switzerland";

// Vérification si `$age` est supérieur ou égal à 18
// et si `$country` est égal à "Switzerland"
if ($age >= 18 && $country == "Switzerland") {
    echo "You are allowed to vote in Switzerland.";
}
```

Les structures de contrôle conditionnelles (5/7)

```java
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration de deux variables
    int age = 18;
    String country = "Switzerland";

    // Vérification si `$age` est supérieur ou égal à 18
    // et si `$country` est égal à "Switzerland"
    if (age >= 18 && country.equals("Switzerland")) {
        System.out.print("You are allowed to vote in Switzerland.");
    }
}
```

Les structures de contrôle conditionnelles (6/7)

```php
<?php
// Déclaration d'une variable
$color = "red";

// Vérification de la variable `$color`
switch ($color) {
    // Si la variable `$color` est égale à "red"
    case "red":
        echo "The color is red.";
        break;
```

```
    // Si la variable `$color` est égale à "blue"
    case "blue":
        echo "The color is blue.";
        break;
    // Par défaut
    default:
        echo "The color is neither red nor blue.";
}
```

Les structures de contrôle conditionnelles (7/7)

```
// Équivalent en Java
public static void main(String[] args) {
    // Déclaration d'une variable
    String color = "red";

    // Vérification de la variable `$color`
    switch (color) {
        // Si la variable `$color` est égale à "red"
        case "red":
            System.out.println("The color is red.");
            break;
```

```
        // Si la variable `$color` est égale à "blue"
        case "blue":
            System.out.println("The color is blue.");
            break;
        // Par défaut
        default:
            System.out.println("The color is neither red nor blue.");
    }
}
```

Conclusion

- PHP est un language de programmation très utilisé pour le développement web
- PHP repose sur une architecture client-serveur
- PHP est un language de programmation simple à prendre en main (il suffit de modifier le code, rafraîchir la page et le tour est joué)

Questions

Est-ce que vous avez des questions ?

Mini-projet

Mini-projet

- Application web en PHP pour gérer des animaux de compagnie (ajout, visualisation, modification, suppression)
- Permet de mettre en pratique le contenu théorique du cours.
- À réaliser tout au long de l'unité d'enseignement de façon guidée.
- Je suis là pour vous aider si besoin.

Exercices

Exercices

- Permet d'exercer les concepts vus en cours, autant théoriques que pratiques.

À vous de jouer !

- (Re)lire le support de cours.
- Réaliser le mini-projet.
- Faire les exercices.
- Poser des questions si nécessaire.

Pour le mini-projet ou les exercices, n'hésitez pas à vous entraidez si vous avez des difficultés !

Sources (1/3)

- Illustration principale par Richard Jacobs sur Unsplash
- Illustration par Aline de Nadai sur Unsplash
- Illustration par Nguyen Dang Hoang Nhu sur Unsplash
- Illustration par Tim van Cleef sur Unsplash
- Illustration par Michiel Leunens sur Unsplash
- Illustration par Taylor Vick sur Unsplash
- Illustration par Aaron Burden sur Unsplash
- Illustration par Fejuz sur Unsplash

Sources (2/3)

- Illustration par Jan Huber sur Unsplash
- Illustration par Kier in Sight Archives sur Unsplash
- Illustration par Nick Hillier sur Unsplash
- Illustration par Brooke Lark sur Unsplash
- Illustration par Faris Mohammed sur Unsplash
- Illustration par Kenny Eliason sur Unsplash
- Illustration par charlesdeluvio sur Unsplash
- Illustration par Arham Jain sur Unsplash

Sources (3/3)

- Illustration par Alec Favale sur Unsplash
- Illustration par Samuel Girven sur Unsplash
- Illustration par Nikita Kachanovsky sur Unsplash