

M 107 : Développer des sites web dynamiques

1 ère année

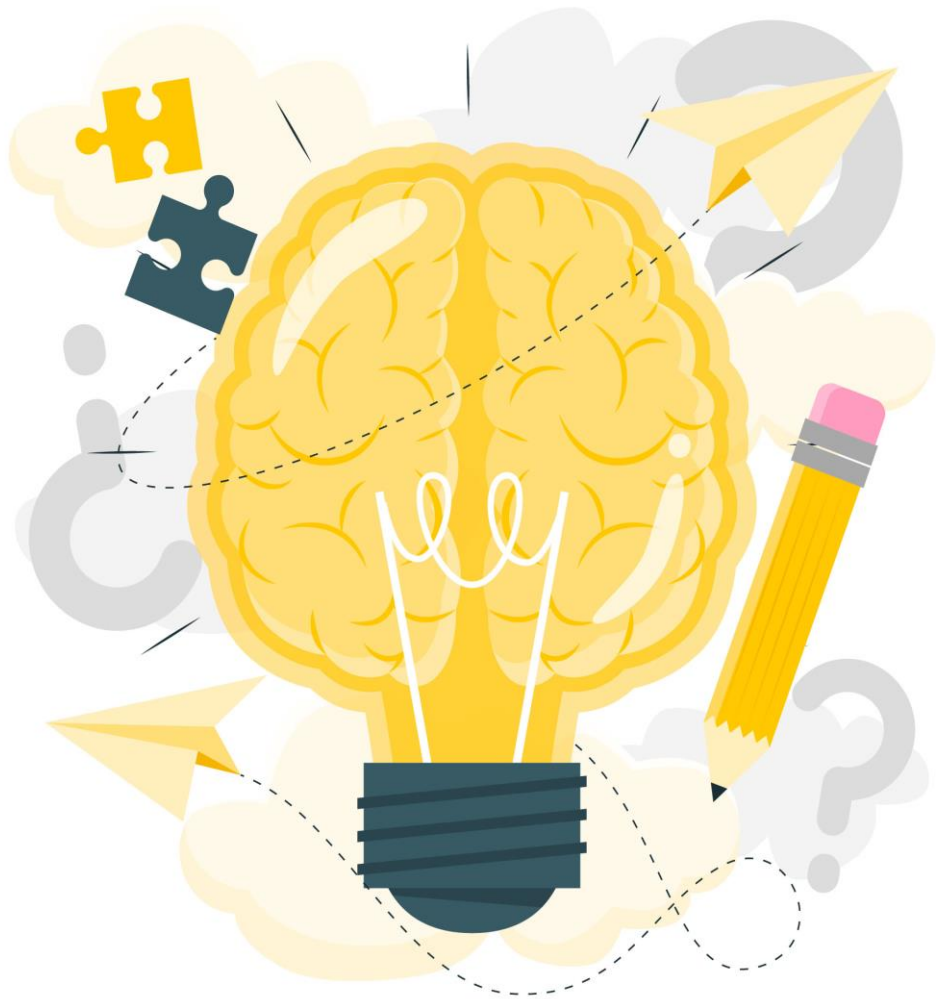
Filière : Développement Digital (Tronc commun)



...

Brainstroming

Web dynamique ?





Réponse simple :

👉 Un site web dynamique est un site dont le contenu peut changer automatiquement en fonction des interactions des utilisateurs ou des mises à jour des données.

Réponse technique :

👉 Un site web dynamique utilise un langage de programmation côté serveur (comme PHP, Python ou Node.js) pour générer du contenu en temps réel. Il peut aussi s'appuyer sur une base de données (comme MySQL) pour stocker et récupérer des informations.

Réponse comparative :

👉 Contrairement à un site statique qui affiche toujours le même contenu, un site dynamique adapte ses pages en fonction des actions de l'utilisateur (ex : connexion, recherche, personnalisation).

Réponse orientée utilisateur :

👉 Un site dynamique permet aux visiteurs d'interagir avec le contenu, comme s'inscrire, publier des commentaires, ajouter des articles à un panier, etc.

Plan

➤ **Approfondir la notion client/serveur**

01 Principe Client/serveur

02 Architectures 2-tiers

03 Types de serveurs web

➤ **Introduire le langage PHP**

01 Définition du PHP

02 Généralités sur les fonctionnalités du PHP

...

Principe client/serveur



Introduction

- ✓ Dès lors que les architectures matérielles peuvent être interconnectées, il est possible de mettre en place une architecture **client-serveur**.
- ✓ Cette architecture repose sur un poste central, appelé **serveur**, qui fournit des services et des données aux machines clientes. Les programmes qui se connectent au serveur sont appelés **clients** (ex : client FTP, client mail, navigateur web)
- ✓ L'architecture client-serveur est basée sur l'interaction et la communication entre **deux types de logiciels** :

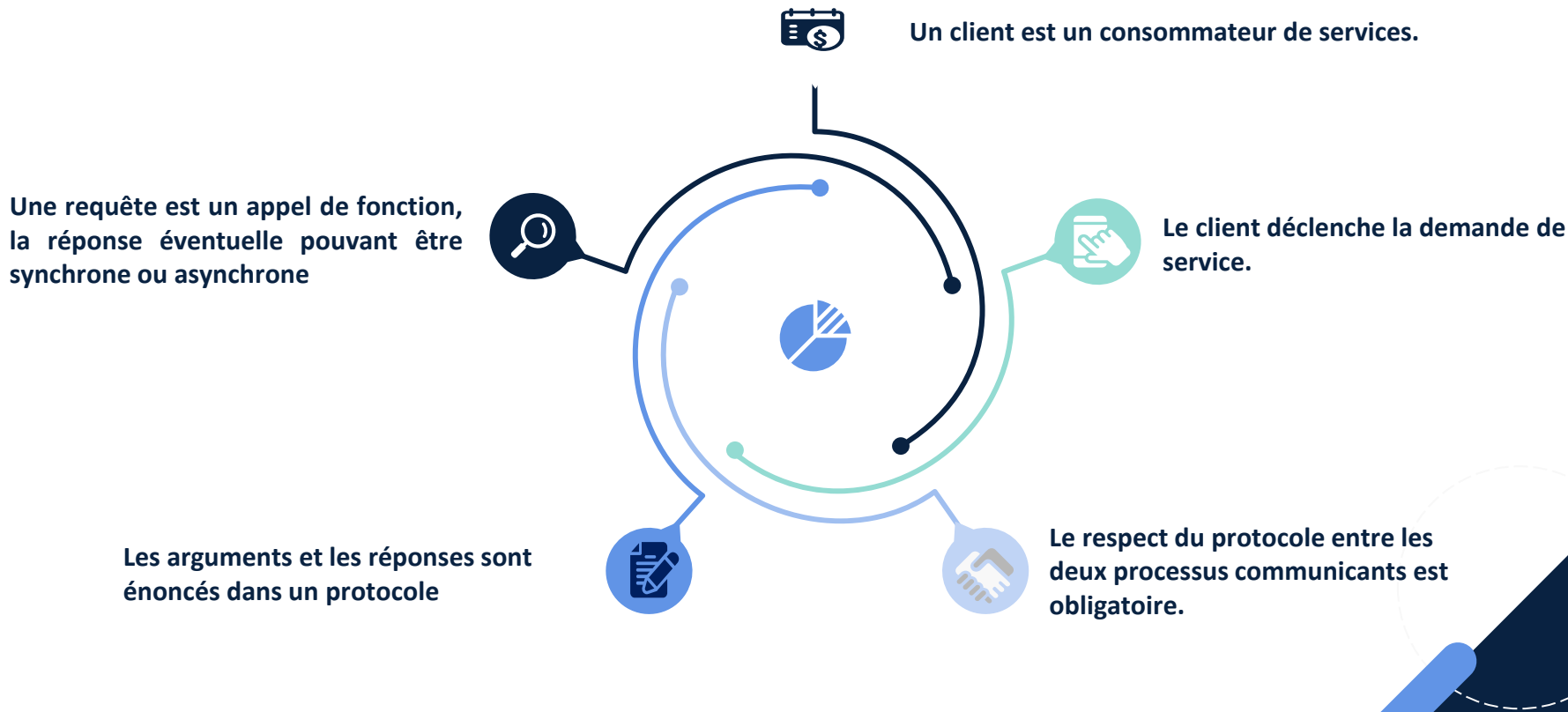
Un **logiciel serveur**, qui gère les **requêtes** et fournit les **ressources**,

Un **logiciel client**, qui envoie des **demandes** au serveur et affiche les résultats.

- ✓ Un des avantages majeurs du modèle client-serveur est son **indépendance** vis-à-vis des plateformes matérielles et logicielles, ce qui permet une interconnexion entre systèmes **hétérogènes**. De plus, le serveur est transparent pour les clients, qui n'ont pas besoin de connaître sa localisation exacte



Un client c'est quoi



C'est quoi un serveur ?



Caractéristiques

- Plus puissant que les ordinateurs personnels en **calcul, entrées-sorties et connexion réseau** pour gérer plusieurs clients.
- Doit être constamment connecté pour assurer un service ininterrompu (ex. via Internet).
- Un même logiciel serveur peut être utilisé sur plusieurs environnements (Unix, Mac, PC...).



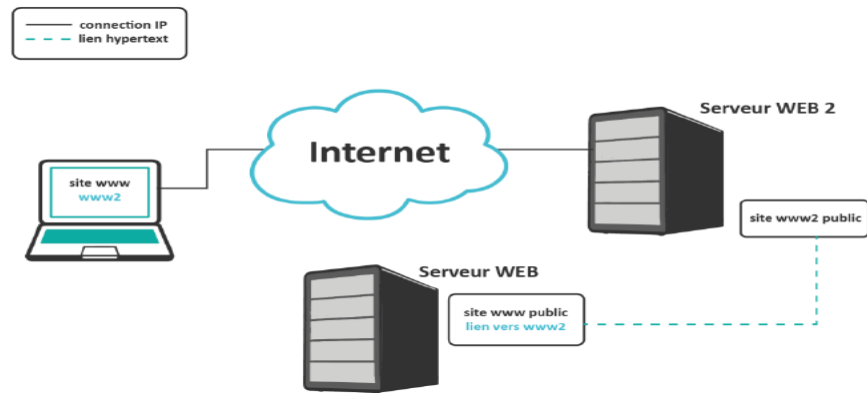
Types

- **Itératif** : traite une requête à la fois
- **Parallèle** : gère plusieurs requêtes simultanément



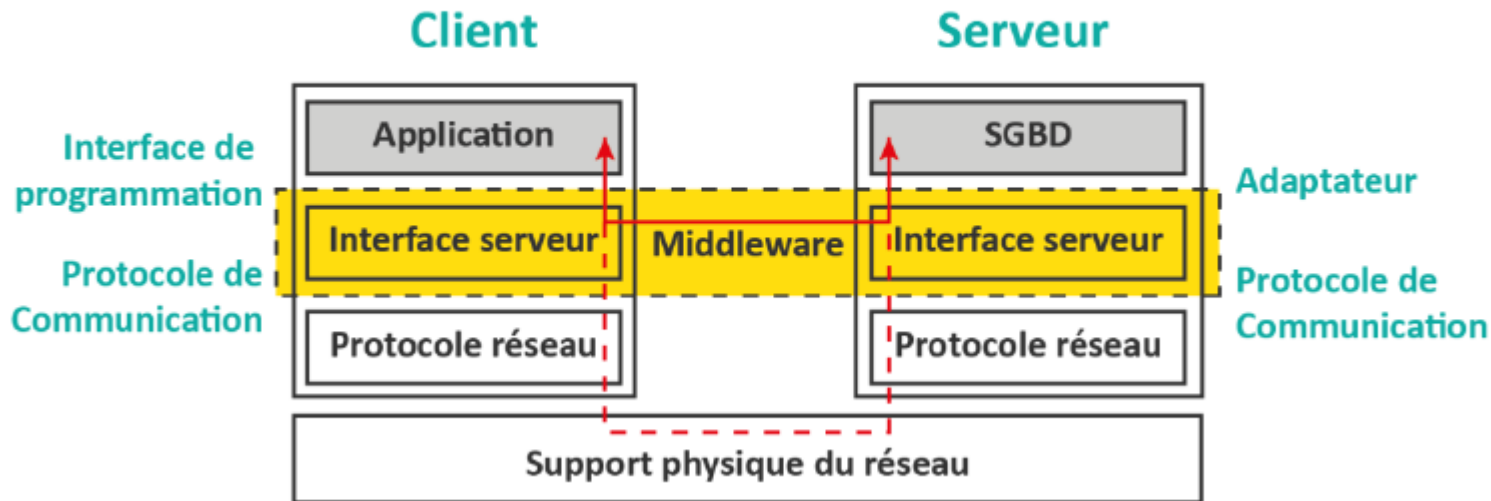
Types de services

- **Avec état** : conserve des informations sur chaque client connecté.
- **Sans état** : ne garde aucune trace des requêtes précédentes.



C'est quoi un Middleware ?

Un **middleware** est un logiciel intermédiaire qui facilite la communication et l'échange de données entre un **client** et un **serveur**. Il agit comme un **pont** qui gère des fonctionnalités supplémentaires sans que le client et le serveur aient besoin de les connaître directement.



C'est quoi un Middleware ?

Rôle du Middleware

- Filtrage : Vérifier les permissions (ex : utilisateur connecté ?).
- Transformation : Convertir des données (ex : JSON → XML).
- Journalisation (Logging) : Enregistrer les requêtes pour le débogage.
- Gestion des erreurs : Renvoyer un message clair si une API échoue.

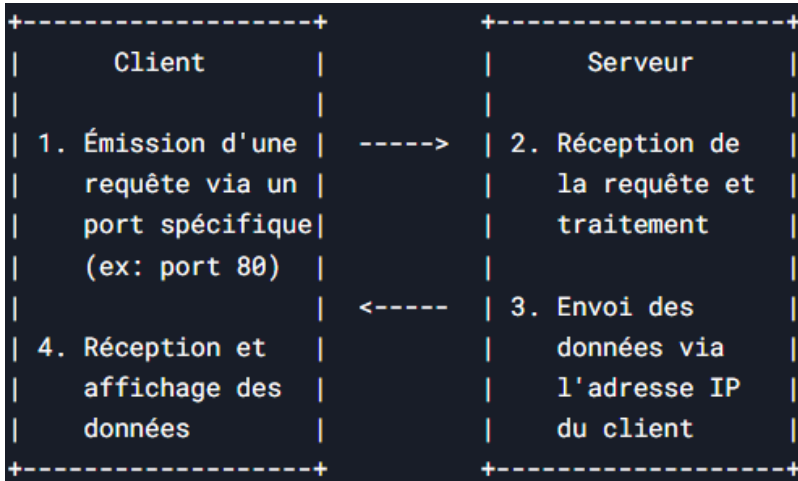
Pourquoi Utiliser un Middleware ?

- Éviter la répétition (ex : vérification d'authentification sur chaque route).
- Séparation des responsabilités (chaque middleware a un rôle précis).
- Flexibilité : On peut ajouter/enlever des fonctionnalités facilement.

Les services d'un Middleware

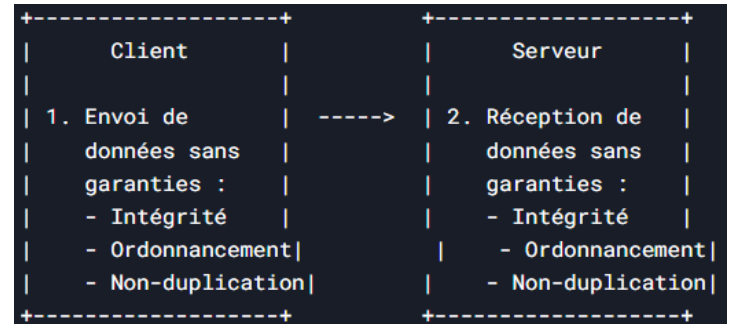
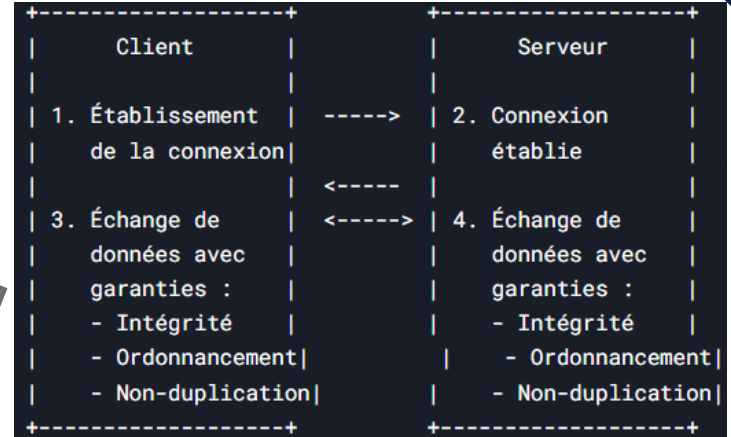
- Conversion
- Routage
- Sécurité
- Communication

Fonctionnement client/serveur



Mode
connecté

Mode non
connecté



Types d'architectures

01

Architecture 1-tiers (Centralisée)

Les **trois couches** (présentation, logique métier et données) sont exécutées sur une **seule et même machine**.

02

Architecture 2-tiers (Client-Serveur)

Le **client** s'occupe de l'interface utilisateur et délègue la gestion des **données** à un **serveur**. repose sur une communication directe entre les **deux**

03

Architecture 3-tiers (Client-Serveur Distribué)

Séparation des responsabilités en **trois couches distinctes** : **Interface utilisateur** (Client) / **Logique métier** (Serveur d'application) / **Données** (Serveur de base de données).

04

Architecture n-tiers (Distribuée et évolutive)

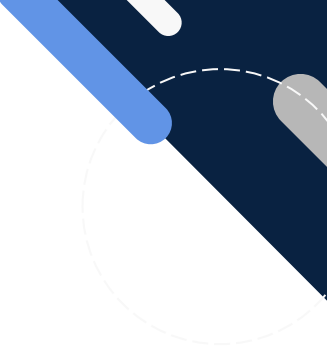
Extension du modèle **3-tiers** avec un nombre de couches supplémentaires ($n \geq 3$). Utilise des **composants interconnectés** via un **bus logiciel** (exemple : **microservices**). Très utilisée pour les **grandes applications web et cloud**.

Exercice :

Réalisez un tableau comparatif de l'architecture 2-tiers et 3 et n-tiers



	2-tiers	3 et n tiers
Administration du système	Complexe (la couche application est physiquement répartie sur plusieurs postes clients)	Moins complexe (les applications peuvent être gérées centralement sur le serveur)
Sécurité	Faible (sécurité au niveau des données)	Elevée (raffinée au niveau des services ou des méthodes)
Encapsulation des données	Faible (les tables de données sont directement accessibles)	Elevée (le client fait appel a des services ou méthodes)
Performance	Faible (plusieurs requêtes SQL sont transmise sur le réseaux, les données sélectionnées doivent êtres acheminées vers le client pour analyse)	Bonne (seulement les appels de service et les réponse sont mise sur le réseau)
Extensibilité	Faible (gestion limitée des liens réseaux avec le clients)	Excellente (possibilité de répartir dynamiquement la charge sur plusieurs serveurs)
Réutilisation	Faible (application monolithique sur le client)	Excellente (réutilisation des services et des objets)



Lien Serveur-serveur	Non	Oui (via le middleware Serveur/Serveur)
Intégration des systèmes déjà en place	Non	Oui (via des passerelles encapsulées par les services ou objets)
Sources de données hétérogènes	Non	Oui (les applications 3-tier peuvent utiliser plusieurs bases de données dans la même transaction)
Flexibilité d'architecture matériel	Limitée	Excellente (possibilité de faire résider les couches 2 et 3 sur une ou plusieurs machines)

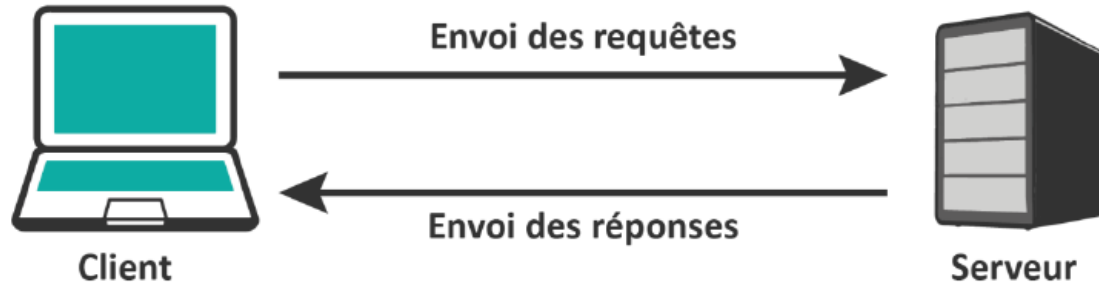
.....

Architecture 2-tiers



Introduction

- L'architecture 2-tiers (aussi appelée client-serveur de première génération, ou encore client-serveur de données) **caractérise les systèmes clients/serveurs pour lesquels le client à un besoin et le serveur lui fournit directement**, en utilisant ses propres ressources.
- Le serveur traite la demande du client **sans recours à des applications tierces**.
- Ce type d'architecture permet de bénéficier pleinement de la puissance des ordinateurs déployés en réseau pour **fournir à l'utilisateur une interface riche**, tout en garantissant la cohérence des données, qui reste gérée de façon centralisée.
- La gestion des données est **prise en charge par un SGBD** (Système de gestion de base de données) centralisé, s'exécutant le plus souvent sur un serveur dédié.
- **Le langage de requête SQL** (Structured Query Language) est la méthode la plus utilisée pour consulter la base de données.
- Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes.



C'est quoi un client

Le client demande un service au serveur.

Un client lourd est une application où les traitements sont principalement effectués sur la machine locale dite cliente.

Caractéristiques d'un client :

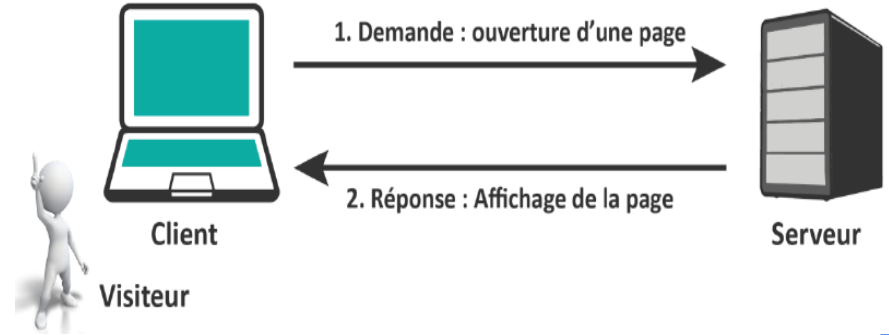
- Il est actif.
- Il envoie des requêtes au serveur.
- il attend et reçoit les réponses du serveur.

C'est quoi un serveur

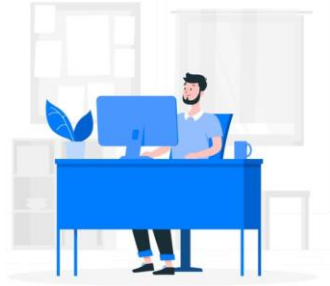
Le serveur exécute la requête et renvoie le résultat au client.

Caractéristiques d'un serveur :

- Il est passif. Il est à l'écoute, prêt à répondre aux requêtes envoyées par les clients.
- Il traite la requête du client dès qu'il la reçoit.
- Il envoie une réponse au client après le traitement de la requête.



Récapitulatif



01

Le client envoie une
requête au serveur

02

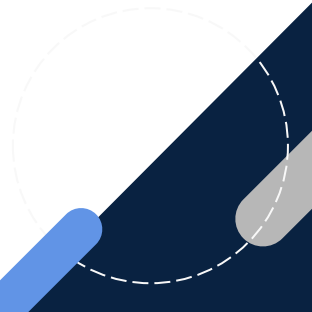
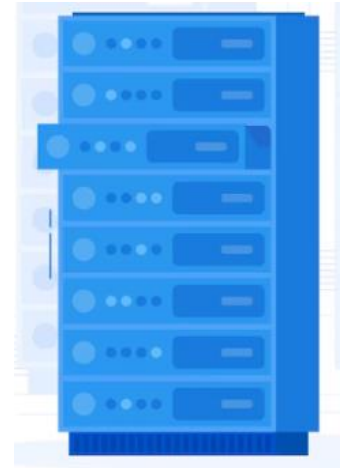
Le serveur reçoit et
traite la requête

03

Le serveur renvoie la
réponse au client

04

Le client affiche la
réponse



Niveau d'abstraction



Couches	Description
Couche Présentation (Couche IHM : Interface Homme-Machine)	<ul style="list-style-type: none">• Elle permet l'interaction d'application avec l'utilisateur.• Elle contrôle en effet les saisies au clavier et à la souris ainsi que la présentation à l'écran.• Elle doit être conviviale et ergonomique
Couche Traitement (Logique Applicative ou couche métier)	<ul style="list-style-type: none">• Elle décrit les traitements à exécuter par l'application afin de répondre aux requêtes clients.• Les traitements locaux : tiennent compte les contrôles effectués au niveau du dialogue avec l'IHM (formulaires, champs, boutons,...).• Les traitements globaux : représentent les règles de l'application appelées aussi logique métier (Business Logique).
Couche Données (couche Persistance)	<ul style="list-style-type: none">• Elle prend en charge les actions liées aux accès aux données.• Regroupe l'ensemble des mécanismes permettant la gestion des informations stockées par l'application.• Garantie souvent les fonctions classiques d'un SGBD (Définition de données, Manipulation de données, Sécurité de données, gestion des transactions, ...).

- Gartner Group a proposé un découpage en six vues distinctes montrant les différentes possibilités de répartition entre clients et serveur des trois couches logicielles.

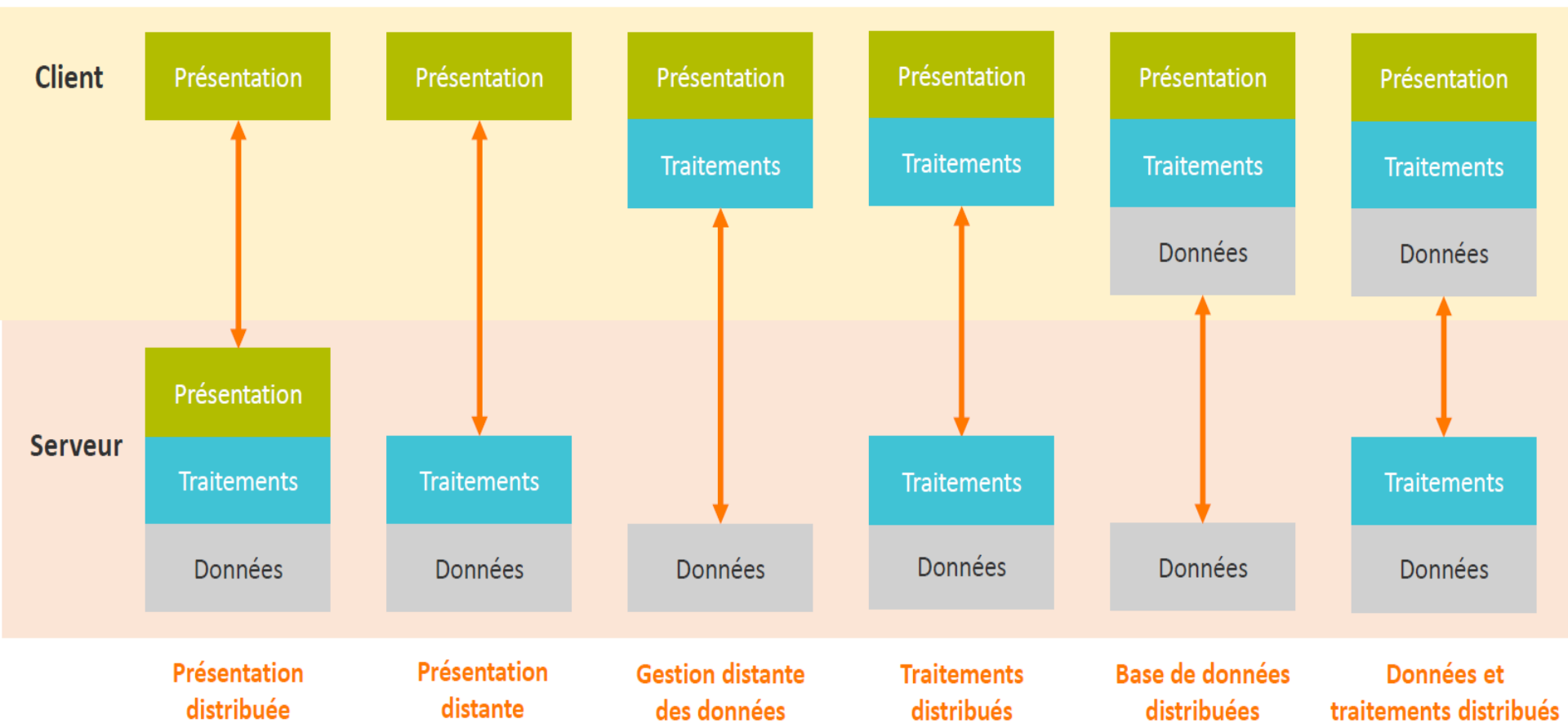


Fig : Modèle de Gartner pour les systèmes à deux niveaux (2-tiers)

.....

Types de serveurs web

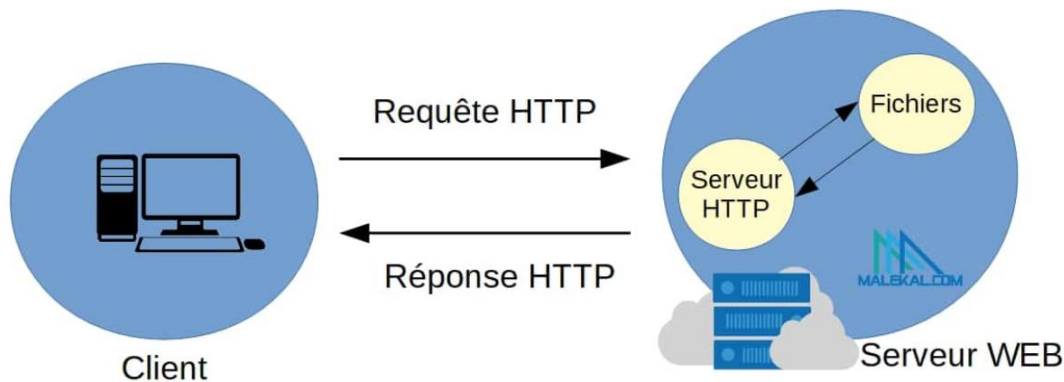


Définition



Serveur Web

- Un serveur Web (aussi appelé serveur http), est tout type de serveur qui permet de diffuser des contenus Web sur Internet ou Intranet. C'est un service logiciel utilisé pour communiquer entre deux appareils sur un réseau.
- Un serveur web sert à rendre accessible des pages web sur internet via le protocole HTTP.
- Un serveur web répond par défaut sur le port 80.
- Pour qu'un site Web soit accessible à tout moment, le serveur Web sur lequel il est hébergé doit être connecté à Internet en permanence
- Un serveur Web en architecture 3 tiers est composé d'un système d'exploitation, un serveur HTTP, un langage serveur et un système de gestion de base de données (SGBD), cela constituant une plate-forme.



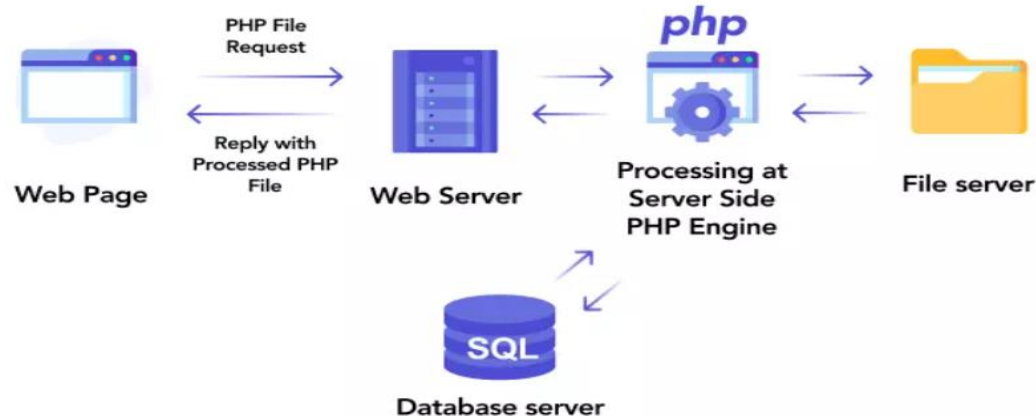
Définition

Serveur Web Statique

- (aussi appelé une pile) est composé d'un ordinateur (matériel) et d'un serveur HTTP (logiciel).
- Il est appelé « statique » car le serveur envoie les fichiers hébergés « tels quels » vers le navigateur..

Serveur Web dynamique

- Possède d'autres composants logiciels, certains qu'on retrouve fréquemment dont un serveur d'applications et une base de données.
- Il est appelé « dynamique » car le serveur d'applications met à jour les fichiers hébergés avant de les envoyer au navigateur via HTTP.



Exercice :

1- Faites une recherche sur les navigateurs web populaires sur le marché

2- Recherchez les serveurs web les plus connus dans le domaine informatique



Types de clients

Les navigateurs web jouent le rôle de clients :



Navigateur d'Apple, Safari est optimisé pour les appareils macOS et iOS, offrant une intégration fluide avec l'écosystème Apple

Développé par Google, Chrome est reconnu pour sa rapidité, sa compatibilité avec les normes web et sa vaste bibliothèque d'extensions.



Connu pour ses fonctionnalités intégrées telles que le VPN gratuit et le bloqueur de publicités, Opera offre une expérience de navigation unique.



brave

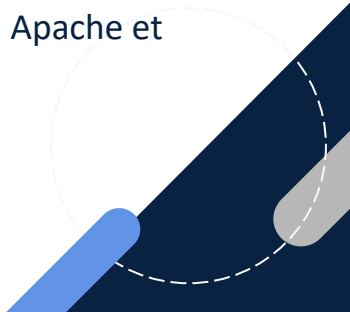


Types de serveurs web



Les serveurs web sont des logiciels qui servent le contenu web aux utilisateurs finaux via les navigateurs. Voici quelques-uns des serveurs web les plus répandus :

- **Apache HTTP Server** : Lancé en 1995, Apache est l'un des serveurs web les plus anciens et les plus utilisés, alimentant une grande partie des sites web mondiaux.
- **NGINX** : Connu pour sa performance et sa capacité à gérer un grand nombre de connexions simultanées, NGINX est largement adopté pour servir des sites à fort trafic.
- **Microsoft Internet Information Services (IIS)** : Serveur web de Microsoft intégré aux systèmes Windows Server, IIS est utilisé principalement dans les environnements Windows.
- **LiteSpeed** : Connu pour sa rapidité et son efficacité, LiteSpeed est une alternative performante à Apache et NGINX, souvent utilisé pour améliorer les performances des sites web.



Types de serveurs web



Le protocole spécifie comment le client et le serveur communiquent.

- **Protocole utilisé** : HyperText Transfer Protocol (**HTTP**)

Les demandes contiennent l'Uniform Ressource Locator (URL) de la page à afficher. L'URL est l'extension de la notion de nom de fichier sur un réseau

- **Syntaxe d'une URL** : protocole://adresse/fichier

<https://www.ofppt.ma/fr/nos-formations>

- **Protocole** : HTTP, FTP, SMTP,...
- **Adresse** : Spécifique au protocole et en général @ IP du serveur HTTP (ou le nom, qui sera résolu par appel au serveur DNS)
- **Fichier** : nom du fichier à récupérer, facultatif (si omis, page par défaut : index.html)

.....

Introduire le langage PHP

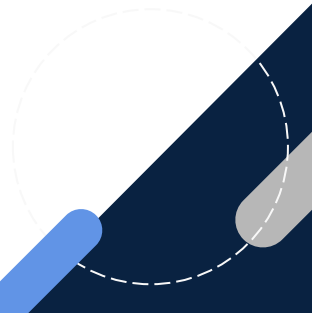
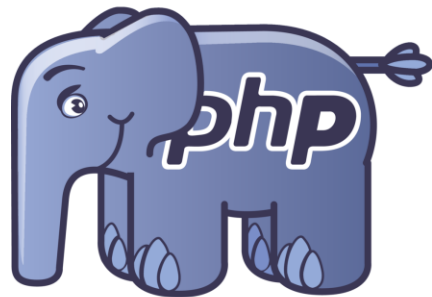


Définition du PHP

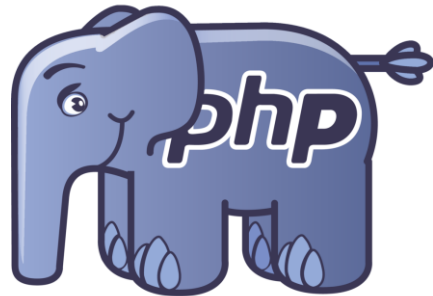


PHP (Hypertext Preprocessor) est un langage de programmation côté **serveur** conçu pour le développement web. Il permet de générer des pages web **dynamiques** et interagir avec des **bases de données** et réaliser une grande variété de tâches côté serveur. PHP est un langage **open-source**, largement utilisé en combinaison avec des technologies comme **HTML**, **CSS**, **JavaScript**, et des bases de données comme **MySQL**.

Il permet d'écrire des **scripts** qui sont exécutés sur le serveur avant que la page ne soit envoyée au navigateur, ce qui permet de créer des pages web **interactives** et **personnalisées**



Définition du PHP



Historique de PHP

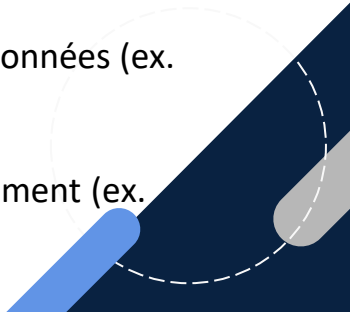
- **1994** : Créé par **Rasmus Lerdorf** pour suivre les visiteurs sur son site
- **1995** : Publication de la première version **PHP 1.0**.
- **2000** : Version **PHP 4** avec un moteur plus performant.
- **2004** : **PHP 5** introduit la **programmation orientée objet (POO)**.
- **2015** : **PHP 7** améliore les performances.
- **2020** : **PHP 8** introduit de nouvelles fonctionnalités.

Définition du PHP

Caractéristiques de PHP



- ✅ **Langage interprété orienté Web** : PHP est exécuté au moment de l'exécution, conçu principalement pour générer des pages web dynamiques.
- ✅ **Syntaxe inspirée de C, Java et Perl** : La syntaxe de PHP est simple et structurée, facilitant l'apprentissage pour ceux qui connaissent déjà ces langages.
- ✅ **Création dynamique de pages HTML** : PHP permet de générer du contenu HTML dynamique, en fonction des données ou des interactions de l'utilisateur.
- ✅ **Open Source et Licence GNU GPL** : PHP est un logiciel libre, permettant à chacun de l'utiliser, le modifier et le redistribuer gratuitement sous la licence GNU GPL.
- ✅ **Accès facile aux bases de données** : PHP facilite la connexion et l'interaction avec des bases de données (ex. MySQL, PostgreSQL) pour gérer des données dans des applications web.
- ✅ **Manipulation d'images** : PHP dispose de fonctions pour créer et modifier des images dynamiquement (ex. redimensionnement, ajout de texte).

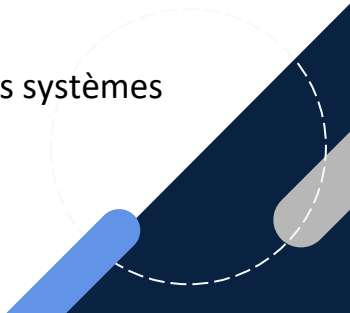


Définition du PHP



Caractéristiques de PHP

- ✅ **Support des Sockets et Protocoles Internet** : PHP permet de gérer les connexions réseau via des sockets et prend en charge divers protocoles Internet comme HTTP, FTP, SMTP.
- ✅ **Gratuit et disponible en Open Source** : PHP est gratuit et son code source est ouvert, permettant à la communauté de l'adapter à ses besoins.
- ✅ **Simplicité de l'écriture de scripts** : La syntaxe claire et la facilité d'écriture font de PHP un langage accessible, même pour les débutants.
- ✅ **Intégration facile avec HTML** : PHP peut être inséré directement dans des pages HTML, permettant de mélanger contenu statique et dynamique.
- ✅ **Compatibilité avec de nombreux serveurs et systèmes d'exploitation** : PHP fonctionne sur divers systèmes d'exploitation (Windows, Linux, macOS) et avec différents serveurs web (Apache, IIS).



Définition du PHP

Langage de script



PHP a considérablement évolué, ces dernières années, d'un langage de **script** non organisé à un langage de **programmation backend orienté objet**.

- Un script est un ensemble d'instructions de programmation qui est interprété à l'exécution.
- Les scripts sont généralement intégrés dans d'autres environnements logiciels.

Les langages de script sont différents des langages de programmation :

- Les langages de script sont spécifiques à une plateforme, tandis que les langages de programmation sont multiplateformes car ils ont la capacité de s'exécuter eux-mêmes.
- Les langages de script sont souvent interprétés, les langages de programmation sont compilés.
- Le langage de programmation est compilé grâce à son propre compilateur qui traduit la syntaxe en code machine **avant** l'exécution.
- Les langages de script sont interprétés ligne par ligne **pendant** l'exécution par l'interpréteur de la plateforme sur laquelle ils fonctionnent.
- Les compilateurs effectuent un traitement collectif des erreurs avant l'exécution, tandis que les interpréteurs évaluent le code ligne par ligne.

.....

Généralités sur les fonctionnalités de PHP



3 types de domaines différents

Langage de script côté serveur :

L'utilisation principale de PHP nécessite **trois composants** : un **analyseur PHP** (CGI ou module serveur), un **serveur web**, et un **navigateur web** pour accéder aux pages générées dynamiquement.



Langage de programmation en ligne de commande :

PHP peut s'exécuter en ligne de commande sans serveur ni navigateur, en utilisant uniquement son exécutable. Il est idéal pour les tâches automatisées (cron sous Linux, planificateur de tâches sous Windows) et le traitement de fichiers texte.



Écrire des applications clientes graphiques :

PHP-GTK permet de créer des applications graphiques de bureau en PHP. C'est une extension non incluse par défaut, offrant une portabilité élevée pour les programmes indépendants du web.



Fonctionnalités du PHP



- PHP est un langage côté serveur, ce qui signifie que son exécution se fait sur le serveur web et non sur le navigateur client.
- Indépendant de la plate-forme utilisée puisqu'il est exécuté côté serveur et non côté client.
- C'est un langage interprété
- PHP supporte de nombreux protocoles comme LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM encore d'autres.
- Vous pouvez ouvrir des sockets réseau, et interagir avec n'importe quel autre protocole.
- PHP est devenu le langage le plus utilisé dans le développement Web, en raison de son interaction avec la base de données MySQL et de son intégration facile avec le code HTML.
- PHP permet de développer des scripts suivant le modèle procédural et/ou le modèle objet.
- La dernière version 8.4.3, publiée le 17 janvier 2025.

Syntaxe du PHP



PHP est **compatible** avec la plupart des **systèmes d'exploitation** (Linux, Windows, macOS, Unix, etc.).

Une page PHP est une page **HTML** intégrant du **code PHP**, identifié par les balises `<?php ... ?>`.

L'extension du fichier doit être **.php** pour être reconnue par le serveur.

Il existe plusieurs façons d'encadrer le code PHP, mais la syntaxe recommandée pour des raisons de compatibilité est :

```
<?php  
.....  
?>
```

Configuration du PHP



Pour vos développements des pages web dynamiques avec PHP, vous aurez besoin :

- **Un OS** : Windows, Linux
- **Un serveur web** avec un interpréteur PHP
- **Un éditeur de texte** pour écrire des scripts
- Éventuellement une **base de données**



Exemples d'application :

- Wiki (MediaWiki, WikiNi, DokuWiki...)
- Forum (phpBB, IPB, punBB...)
- Systèmes de gestion de blog (Dotclear, WordPress...)
- Systèmes de gestion de contenu (appelés aussi CMS) (Spip, ExpressionEngine, Drupal, Xoops...)
- Administration de bases de données (phpMyAdmin)
- Frameworks (Zend Framework, CodeIgniter, Symfony, Laravel etc.)

