

---

# **WEB Dynamique**

*Version 0.0.1*

**Emery Assogba**

**oct. 11, 2021**



---

## Contents:

---

<b>1</b>	<b>HTML, XHTML et CSS</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Qu'est-ce que le HTML ? . . . . .	1
1.3	Qu'est-ce que XHTML ? . . . . .	2
1.4	Qu'est-ce que le CSS ? . . . . .	2
1.5	Qu'est-ce que WebFonts ? . . . . .	2
1.6	Exemples . . . . .	2
<b>2</b>	<b>JAVASCRIPT WEB APIS</b>	<b>3</b>
2.1	Qu'est-ce que le scripting ? . . . . .	3
2.2	Quelles interfaces de script sont disponibles ? . . . . .	3
2.3	Au-delà des scripts . . . . .	4
<b>3</b>	<b>GRAPHIQUES</b>	<b>5</b>
3.1	Qu'est-ce que les graphiques ? . . . . .	5
3.2	A quoi servent les graphiques ? . . . . .	6
3.3	Qu'est-ce que le PNG ? . . . . .	6
3.4	Qu'est-ce que le SVG ? . . . . .	6
3.5	Qu'est-ce que le CSS ? . . . . .	6
3.6	Qu'est-ce que l'API Canvas ? . . . . .	7
3.7	Qu'est-ce que WebCGM ? . . . . .	7
3.8	Exemples . . . . .	7
<b>4</b>	<b>Audio et Vidéo</b>	<b>9</b>
4.1	Qu'est-ce que l'audio et la vidéo ? . . . . .	9
4.2	A quoi servent l'audio et la vidéo ? . . . . .	9
4.3	Qu'est-ce que SMIL ? . . . . .	10
4.4	Qu'est-ce que le texte chronométré ? . . . . .	10
4.5	Que sont les fragments de médias ? . . . . .	10
4.6	Que sont les annotations multimédias ? . . . . .	10
4.7	Exemples . . . . .	10
<b>5</b>	<b>Accessibilité</b>	<b>13</b>
5.1	Pourquoi : les arguments en faveur de l'accessibilité du Web . . . . .	13
5.2	Quoi : exemples d'accessibilité Web . . . . .	14
5.3	Comment : rendre votre site Web et vos outils Web accessibles . . . . .	14

<b>6</b>	<b>Internationalisation</b>	<b>17</b>
6.1	Qu'est-ce que l'internationalisation ? . . . . .	17
6.2	Exemples . . . . .	17
<b>7</b>	<b>PHP</b>	<b>21</b>
7.1	Qu'est-ce que PHP ? . . . . .	21
7.2	Qu'est-ce qu'un fichier PHP ? . . . . .	21
7.3	Que peut faire PHP ? . . . . .	21
7.4	Pourquoi PHP ? . . . . .	22
7.5	Quoi de neuf en PHP 7 . . . . .	22
7.6	Installation de PHP . . . . .	22
<b>8</b>	<b>Syntaxe de PHP</b>	<b>23</b>
8.1	Syntaxe PHP de base . . . . .	23
8.2	Commentaires en PHP . . . . .	24
8.3	Les formulaires en PHP . . . . .	26
8.4	Validation de formulaire PHP . . . . .	28
8.5	PHP : se connecter à MySQL . . . . .	35
8.6	Dois-je utiliser MySQLi ou PDO ? . . . . .	35
8.7	Exemples MySQL dans la syntaxe MySQLi et PDO . . . . .	35
8.8	Créer une base de données MySQL en utilisant MySQLi et PDO . . . . .	37
8.9	Créer une table MySQL en utilisant MySQLi et PDO . . . . .	39
8.10	Insérer des données dans MySQL à l'aide de MySQLi et PDO . . . . .	41
8.11	Requêtes préparées et paramètres liés . . . . .	42
8.12	Sélectionnez les données d'une base de données MySQL . . . . .	44
<b>9</b>	<b>PHP - Présentation AJAX</b>	<b>47</b>
9.1	Qu'est-ce qu'AJAX ? . . . . .	47
9.2	AJAX est basé sur les normes Internet . . . . .	47
9.3	AJAX et PHP . . . . .	48
9.4	Exemple de base de données AJAX . . . . .	50
<b>10</b>	<b>Services Web</b>	<b>53</b>
10.1	Objectif de l'architecture de service Web . . . . .	53
<b>11</b>	<b>WebAssembly ou WASM</b>	<b>57</b>
11.1	Compilation d'un nouveau module C/C++ vers WebAssembly . . . . .	57
<b>12</b>	<b>Indices and tables</b>	<b>59</b>

### 1.1 Introduction

HTML (Hypertext Markup Language) et CSS (Cascading Style Sheets) sont deux des technologies de base pour la création de pages Web. HTML fournit la structure de la page, CSS la mise en page (visuelle et auditive), pour une variété d'appareils. Avec les graphiques et les scripts, HTML et CSS sont à la base de la création de pages Web et d'applications Web.

### 1.2 Qu'est-ce que le HTML ?

HTML est le langage utilisé pour décrire la structure des pages Web. HTML donne aux auteurs les moyens de :

- Publiez des documents en ligne avec des titres, du texte, des tableaux, des listes, des photos, etc.
- Récupérez des informations en ligne via des liens hypertextes, d'un simple clic.
- Concevoir des formulaires pour effectuer des transactions avec des services à distance, à utiliser pour rechercher des informations, effectuer des réservations, commander des produits, etc.
- Inclure des feuilles de calcul, des clips vidéo, des clips audio et d'autres applications directement dans leurs documents.

Avec HTML, les auteurs décrivent la structure des pages en utilisant le balisage (markup). Les éléments du langage étiquettent les éléments de contenu tels que «paragraphe», «liste», «tableau», etc.

## 1.3 Qu'est-ce que XHTML ?

XHTML est une variante de HTML qui utilise la syntaxe de XML, le langage de balisage extensible. XHTML a tous les mêmes éléments (pour les paragraphes, etc.) que la variante HTML, mais la syntaxe est légèrement différente. Étant donné que XHTML est une application XML, vous pouvez utiliser d'autres outils XML (comme XSLT, un langage pour transformer le contenu XML).

## 1.4 Qu'est-ce que le CSS ?

CSS est le langage utilisé pour décrire la présentation des pages Web, y compris les couleurs, la mise en page et les polices. Il permet d'adapter la présentation à différents types d'appareils, tels que les grands écrans, les petits écrans ou les imprimantes. CSS est indépendant du HTML et peut être utilisé avec n'importe quel langage de balisage basé sur XML. La séparation du HTML du CSS facilite la maintenance des sites, le partage des feuilles de style entre les pages et l'adaptation des pages à différents environnements. C'est ce qu'on appelle la séparation de la structure (ou : du contenu) de la présentation.

## 1.5 Qu'est-ce que WebFonts ?

WebFonts est une technologie qui permet aux utilisateurs d'utiliser des polices à la demande sur le Web sans nécessiter d'installation dans le système d'exploitation. Le W3C a de l'expérience dans les polices téléchargeables via HTML, CSS2 et SVG. Jusqu'à récemment, les polices téléchargeables n'étaient pas courantes sur le Web en raison de l'absence d'un format de police interopérable. L'effort de WebFonts prévoit de résoudre ce problème en créant un format de police ouvert et pris en charge par l'industrie pour le Web (appelé « WOFF »).

## 1.6 Exemples

L'exemple très simple suivant d'une partie d'un document HTML illustre comment créer un lien dans un paragraphe. Lorsqu'il est rendu à l'écran (ou par un synthétiseur vocal), le texte du lien sera le « rapport final » ; lorsque quelqu'un active le lien, le navigateur récupère la ressource identifiée par « <http://www.example.com/report> » :

```
<p class="moreinfo">Pour plus d'information consultez le  
<a href="http://www.example.com/report">rapport final</a>.</p>
```

L'attribut class sur la balise de début du paragraphe (« <p> ») peut être utilisé, entre autres, pour ajouter du style. Par exemple, pour mettre en italique le texte de tous les paragraphes avec une classe de « moreinfo », on pourrait écrire, en CSS :

```
p.moreinfo { font-style: italic }
```

En plaçant cette règle dans un fichier séparé, le style peut être partagé par plusieurs documents HTML.

Pour plus d'information visitez [http://slaout.linux62.org/html\\_css/](http://slaout.linux62.org/html_css/) et <https://www.w3.org/Style/Examples/011/firstcss>

### 2.1 Qu'est-ce que le scripting ?

Un script est un programme qui n'a pas besoin de pré-traitement (par exemple, compilation) avant d'être exécuté. Dans le contexte d'un navigateur Web, les scripts font généralement référence au code de programme écrit en JavaScript qui est exécuté par le navigateur lorsqu'une page est téléchargée ou en réponse à un événement déclenché par l'utilisateur.

Les scripts peuvent rendre les pages Web plus dynamiques. Par exemple, sans recharger une nouvelle version d'une page, il peut autoriser des modifications du contenu de cette page (DHTML (Dynamic HTML)), ou autoriser l'ajout ou l'envoi de contenu à partir de cette page (AJAX (Asynchronous JavaScript and XML)).

Au-delà, les scripts permettent de plus en plus aux développeurs de créer un pont entre le navigateur et la plateforme sur laquelle il s'exécute, permettant, par exemple, de créer des pages Web qui intègrent des informations de l'environnement de l'utilisateur, telles que l'emplacement actuel, les détails du carnet d'adresses, etc.

Cette interactivité supplémentaire fait que les pages Web se comportent comme une application logicielle traditionnelle. Ces pages Web sont souvent appelées applications Web et peuvent être mises à disposition soit directement dans le navigateur sous forme de page Web, soit empaquetées et distribuées sous forme de widgets.

### 2.2 Quelles interfaces de script sont disponibles ?

L'interface de script la plus basique développée au W3C est le DOM, le Document Object Model qui permet aux programmes et aux scripts d'accéder et de mettre à jour dynamiquement le contenu, la structure et le style des documents. Les spécifications DOM forment le cœur du DHTML.

Les modifications du contenu à l'aide du DOM par l'utilisateur et par des scripts déclenchent des événements que les développeurs peuvent utiliser pour créer des interfaces utilisateur riches.

Un certain nombre d'interfaces plus avancées sont en cours de standardisation, par exemple :

- XMLHttpRequest permet de charger du contenu supplémentaire à partir du Web sans charger un nouveau document, un composant central d'AJAX,
- l'API de géolocalisation rend l'emplacement actuel de l'utilisateur disponible pour les applications basées sur un navigateur,

- plusieurs API rendent transparente l'intégration des applications Web avec le système de fichiers local et le stockage.

WAI ARIA propose des mécanismes pour garantir que cette interactivité supplémentaire reste utilisable indépendamment des appareils et des handicaps. Des considérations supplémentaires s'appliquent au développement d'applications Web pour appareils mobiles.

## **2.3 Au-delà des scripts**

Alors que les scripts offrent une excellente opportunité de développer de nouvelles interfaces et d'expérimenter de nouvelles interactions avec les utilisateurs, au fil du temps, un certain nombre de ces ajouts bénéficient d'une approche plus déclarative ; par exemple, au lieu que chaque développeur réimplémente une interface de calendrier qui permet à un utilisateur de choisir une date, définir un type d'entrée (`<input type="date" />`) qui le fait automatiquement permet de gagner beaucoup de temps et les bugs, et crée un terrain pour de nouvelles innovations.

Au-delà de l'ensemble des interfaces déclaratives mises à disposition via HTML, plusieurs technologies ont été développées pour rendre possibles ces applications Web déclaratives.

Pour plus d'information visitez les liens suivants : [https://developer.mozilla.org/fr/docs/Learn/Getting\\_started\\_with\\_the\\_web/JavaScript\\_basics](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics) , <https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide/Introduction> , [https://developer.mozilla.org/fr/docs/Learn/Tools\\_and\\_testing/Client-side\\_JavaScript\\_frameworks](https://developer.mozilla.org/fr/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks) et [https://developer.mozilla.org/fr/docs/Web/JavaScript/Inheritance\\_and\\_the\\_prototype\\_chain](https://developer.mozilla.org/fr/docs/Web/JavaScript/Inheritance_and_the_prototype_chain)



Le Web, c'est plus que du texte et de l'information, c'est aussi un moyen d'expression de la créativité artistique, de la visualisation de données et d'optimisation de la présentation de l'information pour différents publics avec des besoins et des attentes différents. L'utilisation de graphiques sur les sites Web améliore l'expérience des utilisateurs, et le W3C dispose de plusieurs technologies différentes et complémentaires qui fonctionnent avec HTML et les scripts pour fournir aux créateurs de pages Web et d'applications Web les outils dont ils ont besoin pour fournir la meilleure représentation possible de leur contenu.

### 3.1 Qu'est-ce que les graphiques ?

Les graphiques Web sont des représentations visuelles utilisées sur un site Web pour améliorer ou permettre la représentation d'une idée ou d'un sentiment, afin d'atteindre l'utilisateur du site Web. Les graphiques peuvent divertir, éduquer ou avoir un impact émotionnel sur l'utilisateur et sont essentiels à la force de l'image de marque, à la clarté de l'illustration et à la facilité d'utilisation des interfaces.

Des exemples de graphiques incluent des cartes, des photographies, des conceptions et des modèles, des arbres généalogiques, des diagrammes, des plans architecturaux ou techniques, des graphiques à barres et des camemberts, une typographie, des schémas, des dessins au trait, des organigrammes et de nombreuses autres formes d'images.

Les graphistes disposent de nombreux outils et technologies pour tout, de l'impression au développement Web, et le W3C fournit de nombreux formats sous-jacents pouvant être utilisés pour la création de contenu sur la plate-forme Web ouverte.

## 3.2 A quoi servent les graphiques ?

Les graphiques sont utilisés pour tout, de l'amélioration de l'apparence des pages Web à la couche de présentation et d'interaction utilisateur pour les applications Web à part entière.

Différents cas d'utilisation des graphiques exigent des solutions différentes, il existe donc plusieurs technologies différentes. Les photographies sont mieux représentées avec PNG, tandis que les dessins au trait interactifs, la visualisation des données et même les interfaces utilisateur ont besoin de la puissance de SVG et de l'API Canvas. CSS existe pour améliorer d'autres formats comme HTML ou SVG. WebCGM répond aux besoins d'illustration et de documentation techniques de nombreuses industries.

## 3.3 Qu'est-ce que le PNG ?

Portable Network Graphics (PNG) est un format de fichier statique pour le stockage et l'échange sans perte, portable et bien compressé d'images raster (bitmaps). Il offre un contrôle des couleurs riche, avec une prise en charge des couleurs indexées, des niveaux de gris et des couleurs vraies et une transparence du canal alpha. PNG est conçu pour le Web, avec des capacités de streaming et de rendu progressif. Il est pris en charge de manière omniprésente dans les navigateurs Web, les outils de création graphique, les boîtes à outils d'images et d'autres parties de la chaîne d'outils de création. Les fichiers PNG ont l'extension de fichier « .PNG » ou « .png » et doivent être déployés en utilisant le type de média « image/png ». Les images PNG peuvent être utilisées avec HTML, CSS, SVG, l'API Canvas et WebCGM.

## 3.4 Qu'est-ce que le SVG ?

Les graphiques vectoriels évolutifs (SVG) sont comme le HTML pour les graphiques. Il s'agit d'un langage de balisage permettant de décrire tous les aspects d'une image ou d'une application Web, de la géométrie des formes au style du texte et des formes, en passant par l'animation et les présentations multimédias, y compris la vidéo et l'audio. Il est entièrement interactif et inclut un DOM scriptable ainsi qu'une animation déclarative (via la spécification SMIL). Il prend en charge un large éventail de fonctionnalités visuelles telles que les dégradés, l'opacité, les filtres, l'écartèlement et le masquage.

L'utilisation de SVG permet des graphiques entièrement évolutifs, fluides et réutilisables, des graphiques simples pour améliorer les pages HTML, aux graphiques entièrement interactifs et à la visualisation des données, aux jeux, aux images statiques autonomes de haute qualité. SVG est pris en charge nativement par la plupart des navigateurs modernes (avec des plugins pour permettre son utilisation sur tous les navigateurs) et est largement disponible sur les appareils mobiles et les décodeurs. Tous les principaux outils de dessin de graphiques vectoriels importent et exportent du SVG, et ils peuvent également être générés via des langages de script côté client ou côté serveur.

## 3.5 Qu'est-ce que le CSS ?

Les feuilles de style en cascade (CSS) sont le langage utilisé pour décrire la présentation des pages Web, y compris les couleurs, la mise en page et les informations sur les polices. Il peut être utilisé pour améliorer les aspects graphiques du HTML et du SVG.

## 3.6 Qu'est-ce que l'API Canvas ?

L'API Canvas est une technologie de script côté client permettant la création ou la modification d'images raster (bit-maps) . Il utilise des méthodes de programmation vectorielles pour créer des formes, des dégradés et d'autres effets graphiques, et comme il n'a pas de DOM, il peut fonctionner très rapidement. Les scripteurs dédiés peuvent développer des jeux ou même des applications complètes en utilisant l'API Canvas, seule ou intégrée dans HTML ou SVG. Il est pris en charge nativement dans la plupart des navigateurs modernes (avec des bibliothèques de scripts étendant la prise en charge à tous les principaux navigateurs), et même sur certains appareils mobiles.

## 3.7 Qu'est-ce que WebCGM ?

Web Computer Graphics Metafile (WebCGM) est le profil Web de CGM, la norme ISO pour la définition d'images vectorielles et composites vectorielles/raster. CGM a un nombre important d'adeptes dans l'illustration technique, la documentation électronique interactive, la visualisation de données géophysiques, entre autres domaines d'application, et est largement utilisé dans les domaines de l'ingénierie automobile, de l'aéronautique et de l'industrie de la défense. WebCGM possède bon nombre des mêmes fonctionnalités graphiques que SVG.

## 3.8 Exemples

Comme exemple simple de fichier image SVG, voici un cercle avec un dégradé pour lui donner un aspect 3D :

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%" viewBox="0 0 100
↪100">

  <defs>
    <radialGradient id="rg" cx="100" cy="100" fx="80" fy="80" gradientUnits=
↪"userSpaceOnUse">
      <stop offset="5%" stop-color="lightskyblue" />
      <stop offset="100%" stop-color="darkblue" />
    </radialGradient>
  </defs>
  <circle id="circle_1" cx="100" cy="100" r="95" fill="url(#rg)" />
</svg>
```

Pour plus d'informations visitez <https://www.w3.org/Consortium/Offices/Presentations/SVG/0.svg> ou <https://www.w3.org/TR/SVGPrintPrimer12/>



### 4.1 Qu'est-ce que l'audio et la vidéo ?

Les termes audio et vidéo font généralement référence au format de stockage multimédia basé sur le temps pour les informations sur le son/la musique et les images animées. L'enregistrement numérique audio et vidéo, également appelé codecs audio et vidéo, peut être non compressé, compressé sans perte ou compressé avec perte en fonction de la qualité souhaitée et des cas d'utilisation.

Les codecs audio peuvent généralement contenir un canal audio (mono), deux canaux audio (stéréo) ou plusieurs canaux (par exemple, surround « 5.1 »). Par exemple, la voix humaine est enregistrée en utilisant un canal tandis que la musique utilise en général deux canaux ou plus. La qualité variera en fonction du débit, c'est-à-dire du nombre de bits utilisés par unité de temps de lecture.

Les codecs vidéo contiendront une séquence d'images, c'est-à-dire des images fixes et, pour les formats compressés, des mouvements entre ces images. La qualité variera en fonction du nombre d'images par seconde, de l'espace colorimétrique, de la résolution, etc.

Les formats de stockage multimédia contiendront des flux de codecs audio, des flux de codecs vidéo, des sous-titres et des méta-informations. Il les combine pour fournir l'audio ou la vidéo, avec des matériaux alternatifs ou améliorés. En général, une vidéo aura un flux de codec vidéo, un ou plusieurs flux de codec audio alternatifs, et peut avoir des sous-titres et des méta-informations.

### 4.2 A quoi servent l'audio et la vidéo ?

L'audio et la vidéo sont utilisés pour améliorer l'expérience avec les pages Web (par exemple, l'arrière-plan audio) pour diffuser de la musique, des vidéos familiales, des présentations, etc. ou la langue des signes.

## 4.3 Qu'est-ce que SMIL ?

SMIL est le langage d'intégration multimédia synchronisé, un langage basé sur XML pour décrire des présentations multimédia interactives. Il combine audio, vidéo, hypertexte, images dans le temps et dans l'espace, permettant des transitions visuelles entre les deux.

## 4.4 Qu'est-ce que le texte chronométré ?

Timed Text est un langage basé sur XML pour les médias de texte chronométré dans le but d'échanger entre les systèmes de création. Il peut également être utilisé directement comme format de diffusion, donc adapté au sous-titrage ou à la description vidéo.

## 4.5 Que sont les fragments de médias ?

Les fragments de média fournissent un moyen standard indépendant du format de média d'adresser des fragments de média, dans le temps et dans l'espace, sur le Web à l'aide d'identifiants (URL, URI, IRI).

## 4.6 Que sont les annotations multimédias ?

Les annotations multimédias fournissent des moyens de décrire les ressources multimédias, à l'aide d'un ensemble commun de propriétés. Ces annotations aident à transmettre des informations qui peuvent ensuite être réutilisées dans les moteurs de recherche ou les systèmes de balisage.

## 4.7 Exemples

Voici quelques exemples de SMIL, SVG et HTML respectivement avec du contenu vidéo :

```
<smil xmlns="http://www.w3.org/ns/SMIL">
  <body>
    <par>
      <video src="http://www.example.org/MyVideo" fill="freeze"/>
      <text src="http://www.example.org/MyCaption" fill="freeze" />
    </par>
  </body>
</smil>
```

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">
  <g>
    <video xlink:href="http://www.example.org/MyVideo"
      x="0" y="0" width="360" height="240" />
  </g>
</svg>
```

```
<!DOCTYPE html>
<html>
  <head>
```

(suite sur la page suivante)

(suite de la page précédente)

```
<title>My Video</title>
</head>
<body>
  <video src="http://www.example.org/MyVideo"
    width="360" height="240">
</body>
</html>
```

Pour plus d'informations consultez [https://developer.mozilla.org/fr/docs/Web/SVG/SVG\\_animation\\_with\\_SMIL](https://developer.mozilla.org/fr/docs/Web/SVG/SVG_animation_with_SMIL) et <https://www.w3.org/TR/REC-smil/smil-animation.html>





Le Web est fondamentalement conçu pour fonctionner pour tous, quels que soient leur matériel, leurs logiciels, leur langue, leur emplacement ou leurs capacités. Lorsque le Web atteint cet objectif, il est accessible aux personnes ayant un large éventail de capacités auditives, motrices, visuelles et cognitives.

Ainsi, l'impact du handicap est radicalement modifié sur le Web car le Web supprime les barrières à la communication et à l'interaction auxquelles de nombreuses personnes sont confrontées dans le monde physique. Cependant, lorsque les sites Web, les applications, les technologies ou les outils sont mal conçus, ils peuvent créer des barrières qui empêchent les gens d'utiliser le Web.

L'accessibilité est essentielle pour les développeurs et les organisations qui souhaitent créer des sites Web et des outils Web de haute qualité, et ne pas empêcher les gens d'utiliser leurs produits et services.

La mission de la Web Accessibility Initiative (WAI) est de mener le Web à son plein potentiel pour être accessible, permettant aux personnes handicapées de participer de manière égale sur le Web.

## 5.1 Pourquoi : les arguments en faveur de l'accessibilité du Web

Le Web doit être accessible pour offrir un accès égal et des chances égales aux personnes ayant des capacités diverses. En effet, la Convention des Nations Unies relative aux droits des personnes handicapées reconnaît l'accès aux technologies de l'information et de la communication, y compris le Web, comme un droit humain fondamental.

L'accessibilité soutient l'inclusion sociale des personnes handicapées ainsi que d'autres personnes, telles que les personnes âgées, les habitants des zones rurales et les habitants des pays en développement.

L'accessibilité profite également aux personnes sans handicap. La vidéo Web Accessibility Perspectives montre des exemples de la façon dont l'accessibilité est essentielle pour les personnes handicapées et utile pour tout le monde dans diverses situations.

Il existe également une solide analyse de rentabilisation en faveur de l'accessibilité. L'accessibilité chevauche d'autres bonnes pratiques telles que la conception de sites Web mobiles, l'indépendance des appareils, l'interaction multimodale, la convivialité, la conception pour les utilisateurs plus âgés et l'optimisation des moteurs de recherche (SEO). Des études de cas montrent que les sites Web accessibles ont de meilleurs résultats de recherche, des coûts de maintenance réduits et une audience accrue, entre autres avantages. L'élaboration d'une analyse de rentabilisation de l'accessibilité du Web pour votre organisation détaille les avantages de l'accessibilité du Web.

## 5.2 Quoi : exemples d'accessibilité Web

Des sites Web et des outils bien conçus peuvent être utilisés par des personnes handicapées. Cependant, actuellement de nombreux sites et outils sont développés avec des barrières d'accessibilité qui rendent difficile voire impossible leur utilisation pour certaines personnes. Vous trouverez ci-dessous quelques exemples.

### 5.2.1 Texte alternatif pour les images

Les images doivent inclure un texte alternatif équivalent (texte alternatif) dans le balisage/le code.

Si le texte alternatif n'est pas fourni pour les images, les informations sur l'image sont inaccessibles, par exemple, aux personnes qui ne peuvent pas voir et utiliser un lecteur d'écran qui lit à haute voix les informations sur une page, y compris le texte alternatif de l'image visuelle.

Lorsqu'un texte alternatif équivalent est fourni, les informations sont disponibles pour les personnes aveugles, ainsi que pour les personnes qui désactivent les images (par exemple, dans les zones à bande passante chère ou faible). Il est également disponible pour les technologies qui ne peuvent pas voir les images, telles que les moteurs de recherche.

```
<img  
  alt= "Web accessibility initiative Logo"
```

### 5.2.2 La saisie au clavier

Certaines personnes ne peuvent pas utiliser de souris, y compris de nombreux utilisateurs plus âgés dont le contrôle de la motricité fine est limité. Un site Web accessible ne repose pas sur la souris ; il rend toutes les fonctionnalités disponibles à partir d'un clavier. Ensuite, les personnes handicapées peuvent utiliser des technologies d'assistance qui imitent le clavier, telles que la saisie vocale.

### 5.2.3 Transcriptions pour l'audio

Tout comme les images ne sont pas disponibles pour les personnes qui ne peuvent pas voir, les fichiers audio ne sont pas disponibles pour les personnes qui ne peuvent pas entendre. Fournir une transcription textuelle rend les informations audio accessibles aux personnes sourdes ou malentendantes, ainsi qu'aux moteurs de recherche et autres technologies qui ne peuvent pas entendre.

Il est facile et relativement peu coûteux pour les sites Web de fournir des transcriptions. Il existe également des services de transcription qui créent des transcriptions de texte au format HTML.

## 5.3 Comment : rendre votre site Web et vos outils Web accessibles

La plupart des bases de l'accessibilité sont assez faciles à mettre en œuvre. Cependant, si vous débutez dans le domaine de l'accessibilité, il faut du temps et des efforts pour apprendre les problèmes courants et les solutions. Voici les points de départ :

- [Principes d'accessibilité](#) — introduit les exigences d'accessibilité et les normes internationales.
- [Easy Checks - A First Review](#) — vous aide à commencer à tester l'accessibilité d'une page Web et à comprendre certains obstacles courants à l'accessibilité.
- [Conseils pour la mise en route](#) : fournit quelques considérations de base pour la conception, la rédaction et le développement.

Certains obstacles à l'accessibilité sont plus compliqués à éviter et les solutions demandent plus de temps et d'efforts de développement. W3C WAI fournit de nombreuses ressources pour vous aider, telles que des didacticiels et des supports liés à la présentation des WCAG 2.

L'utilisation d'outils de création prenant en charge l'accessibilité facilite la tâche des développeurs Web. Les navigateurs jouent également un rôle dans l'accessibilité. Ces rôles sont expliqués dans Composants essentiels de l'accessibilité Web.



L'accès au Web pour tous a été une préoccupation et un objectif fondamental du World Wide Web Consortium depuis le début. Malheureusement, il est facile de négliger les besoins des personnes de cultures différentes de la vôtre, ou qui utilisent des langues ou des systèmes d'écriture différents. Si vous le faites, vous créez des spécifications et du contenu qui présenteront des obstacles à l'utilisation de votre technologie ou de votre contenu pour de nombreuses personnes dans le monde.

### 6.1 Qu'est-ce que l'internationalisation ?

Si vous vous internationalisez, vous concevez ou développez votre contenu, votre application, vos spécifications, etc., d'une manière qui garantit qu'il fonctionnera bien ou qu'il pourra être facilement adapté pour des utilisateurs de toute culture, région ou langue.

Le mot « Internationalisation » est souvent abrégé en « i18n ». C'est une abréviation largement utilisée, dérivée du fait qu'il y a 18 lettres entre le “i” et le “n”.

### 6.2 Exemples

Un aspect fondamental de l'internationalisation est de s'assurer que la technologie prend en charge le texte dans n'importe quel système d'écriture du monde. C'est pourquoi les technologies du W3C reposent sur le jeu de caractères universel Unicode. Il peut être nécessaire de prendre également en charge d'autres jeux de caractères et encodages hérités.

e + ó ≡ é

Cependant, d'autres facteurs doivent être pris en compte lors de l'utilisation de caractères. Par exemple, les encodages basés sur Unicode permettent de stocker exactement le même texte en utilisant des combinaisons de caractères légèrement différentes. Pour plus d'efficacité et de précision dans la comparaison, le tri et l'analyse du texte, les différentes

séquences doivent être reconnues comme « canoniquement équivalentes ». Vous devez réfléchir à la manière de gérer cela lors du développement d'applications ou de spécifications qui exécutent ou reposent sur de telles tâches.

Parfois, différents systèmes d'écriture nécessitent un support spécial. Par exemple, le japonais, le chinois, le coréen et le mongol peuvent être écrits verticalement, de sorte que le W3C s'assure que CSS, SVG et XSL-FO permettront la prise en charge du texte vertical. Les méthodes d'alignement et de justification du texte sont également différentes pour de tels scripts, et différentes encore pour des scripts comme le thaï et le tibétain. D'autres conventions typographiques locales existent souvent pour des éléments tels que l'accentuation, les annotations, la numérotation des listes, etc. Ces approches typographiques doivent être supportées dans des feuilles de style.

L'arabe, l'hébreu, le persan, l'ourdou et les langues similaires mélangent du texte de droite à gauche et de gauche à droite sur la même ligne, et il est important de pouvoir contrôler la direction du contexte environnant pour que cela fonctionne correctement. Cela signifie que les développeurs de schémas et de formats doivent fournir aux auteurs des moyens de contrôler la direction de leur contenu. Les schémas, les langages de balisage et les formats doivent également prendre en charge un certain nombre d'autres constructions nécessaires pour une gestion efficace du contenu pendant la traduction et la localisation.

Image de texte bidirectionnel en arabe.

Si vous utilisez des formulaires HTML ou concevez des ontologies relatives aux noms et adresses de personnes, vous devrez réfléchir à la manière d'activer les nombreuses approches différentes de formatage des données qui sont possibles dans le monde. Vous devrez peut-être également prendre en charge des calendriers alternatifs, des fuseaux horaires et des heures d'été, des noms et des adresses sous forme native et translittérée, etc.

Les développeurs de contenu et les systèmes de gestion de contenu doivent également être préparés à faire face aux problèmes linguistiques et culturels. Par exemple, une phrase qui est construite en combinant plusieurs phrases ensemble dans une langue peut être impossible à traduire de manière sensée dans une langue avec une structure de phrase différente. Par exemple, dans la traduction japonaise de « Page 1 sur 34 », tous les éléments de la phrase seraient dans l'ordre inverse. Votre demande ne doit pas restreindre l'ordre dans lequel ces éléments peuvent être combinés. Les spécifications des technologies telles que les widgets et les navigateurs vocaux devraient également éviter d'enfermer les développeurs dans une syntaxe biaisée en anglais pour des choses telles que la composition de messages ou le déclenchement d'événements associés au texte.

Graphique illustrant l'ordre des mots entre l'anglais et l'hindi.

Graphique illustrant l'ordre des mots entre l'anglais et le finnois.

Un cercle peut représenter « oui » au Japon et une coche « non ».

Les problèmes culturels doivent également être pris en compte. Le symbolisme peut être spécifique à une culture. La coche signifie correct ou OK dans de nombreux pays. Dans certains pays, cependant, comme le Japon, cela peut être utilisé pour signifier que quelque chose est incorrect. Les localisateurs japonais peuvent avoir besoin de convertir les coches en cercles (leur symbole pour « correct ») dans le cadre du processus de localisation.

Ce ne sont que quelques exemples parmi tant d'autres. Le message clé est que la conception (qu'il s'agisse d'un langage de balisage, d'un protocole, d'un système de gestion de contenu, d'un widget ou d'une application, etc.) doit

être suffisamment flexible pour s'adapter aux besoins locaux.





### 7.1 Qu'est-ce que PHP ?

- PHP est l'acronyme de « PHP : Hypertext Preprocessor »
- PHP est un langage de script open source largement utilisé
- Les scripts PHP sont exécutés sur le serveur
- PHP est gratuit à télécharger et à utiliser

### 7.2 Qu'est-ce qu'un fichier PHP ?

Les fichiers PHP peuvent contenir du texte, du HTML, du CSS, du JavaScript et du code PHP. Le code PHP est exécuté sur le serveur et le résultat est renvoyé au navigateur en tant que code HTML brut. Les fichiers PHP ont l'extension *.php*

### 7.3 Que peut faire PHP ?

- PHP peut générer du contenu de page dynamique
- PHP peut créer, ouvrir, lire, écrire, supprimer et fermer des fichiers sur le serveur
- PHP peut collecter des données de formulaire
- PHP peut envoyer et recevoir des cookies
- PHP peut ajouter, supprimer, modifier des données dans votre base de données
- PHP peut être utilisé pour contrôler l'accès des utilisateurs
- PHP peut crypter les données

Avec PHP, vous n'êtes pas limité à la sortie HTML. Vous pouvez produire des images, des fichiers PDF et même des films Flash. Vous pouvez également sortir n'importe quel texte, tel que XHTML et XML.

## 7.4 Pourquoi PHP ?

- PHP fonctionne sur diverses plateformes (Windows, Linux, Unix, Mac OS X, etc.)
- PHP est compatible avec presque tous les serveurs utilisés aujourd'hui (Apache, IIS, etc.)
- PHP prend en charge un large éventail de bases de données
- PHP est gratuit. Téléchargez-le à partir de la ressource PHP officielle : [www.php.net](http://www.php.net)
- PHP est facile à apprendre et fonctionne efficacement côté serveur

## 7.5 Quoi de neuf en PHP 7

- PHP 7 est beaucoup plus rapide que la précédente version stable populaire (PHP 5.6)
- PHP 7 a amélioré la gestion des erreurs
- PHP 7 prend en charge des déclarations de type plus strictes pour les arguments de fonction
- PHP 7 prend en charge de nouveaux opérateurs (comme l'opérateur de vaisseau spatial : `<=>`)

## 7.6 Installation de PHP

### 7.6.1 De quoi ai-je besoin ?

Pour commencer à utiliser PHP, vous pouvez :

- Trouver un hébergeur avec support PHP et MySQL
- Installez un serveur Web sur votre propre PC, puis installez PHP et MySQL

### 7.6.2 Utiliser un hébergeur avec support PHP

Si votre serveur a activé la prise en charge de PHP, vous n'avez rien à faire. Créez simplement des fichiers .php, placez-les dans votre répertoire Web et le serveur les analysera automatiquement pour vous. Vous n'avez pas besoin de compiler quoi que ce soit ou d'installer des outils supplémentaires. Parce que PHP est gratuit, la plupart des hébergeurs proposent un support PHP.

### 7.6.3 Configurer PHP sur votre propre PC

Cependant, si votre serveur ne supporte pas PHP, vous devez :

- installer un serveur web
- installer PHP
- installer une base de données, telle que MySQL

Le site Web officiel de PHP (PHP.net) contient des instructions d'installation pour PHP : <http://php.net/manual/en/install.php> Compileur/éditeur PHP en ligne

---

## Syntaxe de PHP

---

Un script PHP est exécuté sur le serveur et le résultat HTML brut est renvoyé au navigateur.

### 8.1 Syntaxe PHP de base

Un script PHP peut être placé n'importe où dans le document.

Un script PHP commence par `< ?php` et se termine par `?>` :

```
<?php
    // Le code PHP va ici
?>
```

L'extension de fichier par défaut pour les fichiers PHP est `.php`.

Un fichier PHP contient normalement des balises HTML et du code de script PHP. Ci-dessous, nous avons un exemple de fichier PHP simple, avec un script PHP qui utilise une fonction PHP intégrée « `echo` » pour afficher le texte « Hello World ! » sur une page Web :

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Ma première page PHP</h1>
    <?php
      echo "Bonjour tout le monde !";
    ?>

  </body>
```

---

**Note :** les instructions PHP se terminent par un point-virgule (;).

---

### 8.1.1 Sensibilité à la casse PHP

En PHP, les mots-clés (par exemple, if, else, while, echo, etc.), les classes, les fonctions et les fonctions définies par l'utilisateur ne sont pas sensibles à la casse.

Dans l'exemple ci-dessous, les trois instructions echo ci-dessous sont égales et légales :

```
<!DOCTYPE html>
<html>
  <body>

    <?php
      ECHO "Bonjour tout le monde !<br>" ;
      echo "Bonjour tout le monde !<br>" ;
      EcHo "Bonjour tout le monde !<br>" ;

    ?>
  </body>
</html>
```

---

**Note :** Cependant ; tous les noms de variables sont sensibles à la casse !

---

Regardez l'exemple ci-dessous ; seule la première instruction affichera la valeur de la variable \$color ! En effet, \$color, \$COLOR et \$coLOR sont traités comme trois variables différentes :

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      $color = "rouge" ;
      echo "Ma voiture est". $color. "<br>" ;
      echo "Ma maison est " . $COLOR. "<br>" ;
      echo "Mon bateau est " . $coLOR. "<br>" ;

    ?>
  </body>
</html>
```

### 8.1.2 Exercices PHP

Insérez la partie manquante du code ci-dessous pour afficher « Bonjour le Monde ».

```
"Bonjour le monde";
```

## 8.2 Commentaires en PHP

Un commentaire dans le code PHP est une ligne qui n'est pas exécutée dans le cadre du programme. Son seul but est d'être lu par quelqu'un qui regarde le code.

Les commentaires peuvent être utilisés pour :

- Laisser les autres comprendre votre code
- Vous rappeler ce que vous avez fait - La plupart des programmeurs ont eu l'habitude de revenir à leur propre travail un an ou deux plus tard et d'avoir à repenser ce qu'ils ont fait. Les commentaires peuvent vous rappeler ce que vous pensiez lorsque vous avez écrit le code

PHP prend en charge plusieurs manières de commenter.

Syntaxe pour les commentaires sur une seule ligne :

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Ceci est un commentaire sur une seule ligne
      # Ceci est également un commentaire sur une seule ligne
    ?>
  </body>
</html>
```

Syntaxe pour les commentaires sur plusieurs lignes :

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      /*
        Ceci est un bloc de commentaires de plusieurs lignes
        qui s'étend sur plusieurs
        lignes
      */
    ?>
  </body>
</html>
```

Utiliser des commentaires pour omettre des parties du code :

```
<!DOCTYPE html>
<html>
  <body>
    <?php
      // Vous pouvez également utiliser des commentaires pour omettre des parties d
      ↪ 'une ligne de code
      $x = 5 /* + 15 */ + 5;
      echo $x;
    ?>
  </body>
</html>
```

Pour plus d'information visitez [https://fr.wikibooks.org/wiki/Programmation\\_PHP](https://fr.wikibooks.org/wiki/Programmation_PHP)

## 8.3 Les formulaires en PHP

Les superglobales PHP `$_GET` et `$_POST` sont utilisées pour collecter des données de formulaire.

### 8.3.1 PHP - Un formulaire HTML simple

L'exemple ci-dessous affiche un formulaire HTML simple avec deux champs de saisie et un bouton d'envoi :

```
<!DOCTYPE html>
<html>
<body>
  <form action="bienvenue.php" method="post">
    Nom : <input type="text" name="name"><br>
    Courriel : <input type="text" name="email"><br>
    <input type="submit">
  </form>
</body>
</html>
```

Lorsque l'utilisateur remplit le formulaire ci-dessus et clique sur le bouton *Soumettre*, les données du formulaire sont envoyées pour traitement dans un fichier PHP nommé « *bienvenue.php* ». Les données du formulaire sont envoyées avec la méthode *HTTP POST*.

Pour afficher les données soumises, vous pouvez simplement faire *echo* à toutes les variables. Le fichier « *bienvenue.php* » ressemble à ceci :

```
<!DOCTYPE html>
<html>
<body>
  Bienvenue <?php echo $_POST["name"]; ?><br>
  Votre adresse e-mail est : <?php echo $_POST["email"] ; ?>
</body>
</html>
```

La sortie pourrait être quelque chose comme ceci :

---

**Note :** Bienvenue Jean Votre adresse e-mail est [john.doe@example.com](mailto:john.doe@example.com)

---

Le même résultat pourrait également être obtenu en utilisant la méthode *HTTP GET* :

```
<html>
<body>
  <form action="welcome_get.php" method="get">
    Nom : <input type="text" name="name"><br>
    Courriel : <input type="text" name="email"><br>
    <input type="submit">
  </form>
</body>
</html>
```

et « *welcome\_get.php* » ressemble à ceci :

```
<html>
<body>
```

(suite sur la page suivante)

(suite de la page précédente)

```
Bienvenue <?php echo $_GET["name"]; ?><br>
Votre adresse e-mail est : <?php echo $_GET["email"] ; ?>
</body>
</html>
```

Le code ci-dessus est assez simple. Cependant, il manque le plus important. Vous devez valider les données du formulaire pour protéger votre script des codes malveillants.

Pensez SÉCURITÉ lors du traitement des formulaires PHP !

Cette page ne contient aucune validation de formulaire, elle montre simplement comment vous pouvez envoyer et récupérer des données de formulaire.

Cependant, les pages suivantes montreront comment traiter les formulaires PHP en gardant à l'esprit la sécurité ! Une validation correcte des données du formulaire est importante pour protéger votre formulaire des pirates et des spammeurs !

### 8.3.2 GET vs POST

GET et POST créent tous deux un tableau (par exemple, `array( key1 => value1, key2 => value2, key3 => value3, ... )`). Ce tableau contient des paires clé/valeur, où les clés sont les noms des contrôles de formulaire et les valeurs sont les données d'entrée de l'utilisateur.

GET et POST sont tous deux traités comme `$_GET` et `$_POST`. Ce sont des superglobales, ce qui signifie qu'elles sont toujours accessibles, quelle que soit la portée - et vous pouvez y accéder à partir de n'importe quelle fonction, classe ou fichier sans avoir à faire quoi que ce soit de spécial.

`$_GET` est un tableau de variables passé au script courant via les paramètres d'URL.

`$_POST` est un tableau de variables passé au script courant via la méthode HTTP POST.

### 8.3.3 Quand utiliser GET ?

Les informations envoyées depuis un formulaire avec la méthode GET sont visibles par tous (tous les noms et valeurs des variables sont affichés dans l'URL). GET a également des limites sur la quantité d'informations à envoyer. La limitation est d'environ 2000 caractères. Cependant, comme les variables sont affichées dans l'URL, il est possible de mettre la page en signet. Cela peut être utile dans certains cas.

GET peut être utilisé pour envoyer des données non sensibles.

---

**Note :** GET ne doit JAMAIS être utilisé pour envoyer des mots de passe ou d'autres informations sensibles !

---

### 8.3.4 Quand utiliser POST ?

Les informations envoyées à partir d'un formulaire avec la méthode POST sont invisibles pour les autres (tous les noms/valeurs sont intégrés dans le corps de la requête HTTP) et n'ont aucune limite sur la quantité d'informations à envoyer.

De plus, POST prend en charge des fonctionnalités avancées telles que la prise en charge de l'entrée binaire en plusieurs parties lors du téléchargement de fichiers sur le serveur.

Cependant, comme les variables ne sont pas affichées dans l'URL, il n'est pas possible de mettre la page en signet.

Les développeurs préfèrent POST pour envoyer des données de formulaire.

Ensuite, voyons comment nous pouvons traiter les formulaires PHP de manière sécurisée !

### 8.3.5 Exercices PHP

Si le formulaire suivant est soumis, comment pouvez-vous, dans welcome.php, afficher la valeur du champ « fname » ?

```
<form action="welcome.php" method="get">
    Prénom : <input type="text" name="fname">
</form>

<html>
    <body>
        Bienvenue <?php echo _____ ; ?>
    </body>
</html>
```

## 8.4 Validation de formulaire PHP

---

**Note :** Pensez SÉCURITÉ lors du traitement des formulaires PHP !

---

Cette section montre comment traiter les formulaires PHP en gardant à l'esprit la sécurité. Une validation correcte des données du formulaire est importante pour protéger votre formulaire des pirates et des spammeurs !

Le formulaire HTML sur lequel nous allons travailler dans ces chapitres contient divers champs de saisie :

- des champs de texte obligatoires et facultatifs,
- des boutons radio et un bouton d'envoi

Les règles de validation du formulaire sont les suivantes :

Règles de validation de champ \* Nom (obligatoire. + Ne doit contenir que des lettres et des espaces \* Email (requis. + Doit contenir une adresse email valide (avec @ et .) \* Site Internet (optionnel. S'il est présent, il doit contenir une URL valide \* Commentaire Facultatif. Champ de saisie multiligne (zone de texte) \* Sexe requis. Doit en sélectionner un

Tout d'abord, nous allons examiner le code HTML brut du formulaire :

Les champs nom, e-mail et site Web sont des éléments de saisie de texte et le champ de commentaire est une zone de texte. Le code HTML ressemble à ceci :

```
Nom : <input type="text" name="name">
Courriel : <input type="text" name="email">
Site Web : <input type="text" name="website">
Commentaire : <textarea name="comment" rows="5" cols="40"></textarea>
```

#### Boutons radio

Les champs du genre sont des boutons radio et le code HTML ressemble à ceci : Genre :

```
<input type="radio" name="gender" value="female">Femme
<input type="radio" name="gender" value="male">Homme
<input type="radio" name="gender" value="other">Autre
```

#### L'élément de formulaire

Le code HTML du formulaire ressemble à ceci :



```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
```

Lorsque le formulaire est soumis, les données du formulaire sont envoyées avec `method= »post »`.

#### 8.4.1 Qu'est-ce que la variable `$_SERVER[« PHP_SELF »]` ?

Le `$_SERVER[« PHP_SELF »]` est une variable super globale qui renvoie le nom de fichier du script en cours d'exécution.

Ainsi, le `$_SERVER[« PHP_SELF »]` envoie les données du formulaire soumis à la page elle-même, au lieu de passer à une autre page. De cette façon, l'utilisateur obtiendra des messages d'erreur sur la même page que le formulaire.

#### 8.4.2 Qu'est-ce que la fonction `htmlspecialchars()` ?

La fonction `htmlspecialchars()` convertit les caractères spéciaux en entités HTML. Cela signifie qu'il remplacera les caractères HTML tels que `<` et `>` par `&lt;` et `&gt;`. Cela empêche les attaquants d'exploiter le code en injectant du code HTML ou Javascript (attaques de type « Cross-site Scripting ») dans les formulaires.

**Note :** Grande note sur la sécurité des formulaires PHP

La variable `$_SERVER[« PHP_SELF »]` peut être utilisée par des hackers ! Si `PHP_SELF` est utilisé dans votre page, un utilisateur peut saisir une barre oblique (/) puis des commandes de Cross Site Scripting (XSS) à exécuter.

**Note :** Les scripts intersites (XSS) sont un type de vulnérabilité de sécurité informatique que l'on trouve généralement dans les applications Web. XSS permet aux attaquants d'injecter un script côté client dans des pages Web consultées par d'autres utilisateurs.

Supposons que nous ayons le formulaire suivant dans une page nommée « `test_form.php` » :

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"]; ?>">
```

Maintenant, si un utilisateur entre l'URL normale dans la barre d'adresse comme « `http://www.example.com/test_form.php` », le code ci-dessus sera traduit en :

```
<form method="post" action="test_form.php">
```

Jusqu'ici tout va bien.

Cependant, considérez qu'un utilisateur saisit l'URL suivante dans la barre d'adresse : `http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert("piraté")%3C/script%3E`

Dans ce cas, le code ci-dessus sera traduit en :

```
<form method="post" action="test_form.php/"><script>alert('piraté')</script>
```

Ce code ajoute une balise de script et une commande d'alerte. Et lorsque la page se charge, le code JavaScript sera exécuté (l'utilisateur verra une boîte d'alerte). Ceci est juste un exemple simple et inoffensif de la façon dont la variable `PHP_SELF` peut être exploitée.

Sachez que n'importe quel code JavaScript peut être ajouté à l'intérieur de la balise `<script>` ! Un pirate peut rediriger l'utilisateur vers un fichier sur un autre serveur, et ce fichier peut contenir un code malveillant qui peut modifier les variables globales ou soumettre le formulaire à une autre adresse pour enregistrer les données de l'utilisateur, par exemple.

### 8.4.3 Comment éviter les exploits de \$\_SERVER[« PHP\_SELF »] ?

Les exploits \$\_SERVER[« PHP\_SELF »] peuvent être évités en utilisant la fonction htmlspecialchars().

Le code du formulaire devrait ressembler à ceci :

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

La fonction htmlspecialchars() convertit les caractères spéciaux en entités HTML. Maintenant, si l'utilisateur essaie d'exploiter la variable PHP\_SELF, cela entraînera la sortie suivante :

```
<form method="post" action="test_form.php/&quot;&gt;&lt;script&gt;alert('piraté')&lt;/script&gt;">
```

La tentative d'exploit échoue, et aucun mal n'est fait !

### 8.4.4 Valider les données de formulaire avec PHP

La première chose que nous allons faire est de passer toutes les variables via la fonction htmlspecialchars() de PHP.

Lorsque nous utilisons la fonction htmlspecialchars(); puis si un utilisateur essaie de soumettre ce qui suit dans un champ de texte :

```
<script>location.href('http://www.hacked.com')</script>
```

— cela ne serait pas exécuté, car il serait enregistré en tant que code HTML échappé, comme ceci :

```
&lt;script&gt;location.href('http://www.hacked.com')&lt;/script&gt;
```

Le code peut désormais être affiché en toute sécurité sur une page ou dans un e-mail.

Nous ferons également deux autres choses lorsque l'utilisateur soumettra le formulaire :

- Supprimez les caractères inutiles (espace supplémentaire, tabulation, nouvelle ligne) des données d'entrée utilisateur (avec la fonction PHP trim())
- Supprimer les barres obliques inverses () des données d'entrée utilisateur (avec la fonction PHP stripslashes())

L'étape suivante consiste à créer une fonction qui effectuera toute la vérification pour nous (ce qui est beaucoup plus pratique que d'écrire le même code encore et encore).

Nous nommerons la fonction test\_input().

Maintenant, nous pouvons vérifier chaque variable \$\_POST avec la fonction test\_input(), et le script ressemble à ceci :

```
<?php
// définir les variables et définir des valeurs vides
$name = $email = $gender = $comment = $website = "" ;

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nom = test_input($_POST["nom"]);
    $email = test_input($_POST["email"]);
    $site_Web = test_input($_POST["site_Web"]);
    $commentaire = test_input($_POST["commentaire"]);
    $genre = test_input($_POST["genre"]);
}

fonction test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
```

(suite sur la page suivante)

(suite de la page précédente)

```

        $data = htmlspecialchars($data);
        renvoie $data ;
    }
?>

```

Notez qu'au début du script, nous vérifions si le formulaire a été soumis en utilisant `$_SERVER[« REQUEST_METHOD »]`. Si `REQUEST_METHOD` est `POST`, alors le formulaire a été soumis - et il doit être validé. S'il n'a pas été soumis, ignorez la validation et affichez un formulaire vierge.

Cependant, dans l'exemple ci-dessus, tous les champs de saisie sont facultatifs. Le script fonctionne bien même si l'utilisateur n'entre aucune donnée.

L'étape suivante consiste à rendre les champs de saisie obligatoires et à créer des messages d'erreur si nécessaire.

### 8.4.5 PHP - Champs obligatoires

Les champs « Nom », « E-mail » et « Sexe » sont obligatoires. Ces champs ne peuvent pas être vides et doivent être remplis dans le formulaire HTML.

Dans le code suivant, nous avons ajouté de nouvelles variables : `$nameErr`, `$emailErr`, `$genderErr` et `$websiteErr`. Ces variables d'erreur contiendront des messages d'erreur pour les champs requis. Nous avons également ajouté une instruction `if else` pour chaque variable `$_POST`. Ceci vérifie si la variable `$_POST` est vide (avec la fonction PHP `empty()`). S'il est vide, un message d'erreur est stocké dans les différentes variables d'erreur, et s'il n'est pas vide, il envoie les données d'entrée utilisateur via la fonction `test_input()` :

```

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }

    if (empty($_POST["website"])) {
        $website = "";
    } else {
        $website = test_input($_POST["website"]);
    }

    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
        $comment = test_input($_POST["comment"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    }
}

```

(suite sur la page suivante)

(suite de la page précédente)

```

} else {
    $gender = test_input($_POST["gender"]);
}
}
?>

```

### 8.4.6 PHP - Afficher les messages d'erreur

Ensuite dans le formulaire HTML, nous ajoutons un petit script après chaque champ obligatoire, qui génère le message d'erreur correct si besoin (c'est-à-dire si l'utilisateur essaie de soumettre le formulaire sans remplir les champs obligatoires) : Exemple

```

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
Name: <input type="text" name="name">
<span class="error">* <?php echo $nameErr; ?></span>
<br><br>
E-mail:
<input type="text" name="email">
<span class="error">* <?php echo $emailErr; ?></span>
<br><br>
Website:
<input type="text" name="website">
<span class="error"><?php echo $websiteErr; ?></span>
<br><br>
Comment: <textarea name="comment" rows="5" cols="40"></textarea>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="other">Other
<span class="error">* <?php echo $genderErr; ?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

```

L'étape suivante consiste à valider les données d'entrée, c'est-à-dire « Le champ Nom ne contient-il que des lettres et des espaces ? » et « Le champ E-mail contient-il une syntaxe d'adresse e-mail valide ? », et s'il est rempli, « Le champ Site Web contient-il une URL valide ? ».

```

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z-' ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
}

```

(suite sur la page suivante)

(suite de la page précédente)

```

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}

if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid (this regular expression also allows dashes
    ↪ in the URL)
    if (!preg_match("/\b(?:(:https?|ftp):\\\/|www\.)[-a-z0-9+&@#\/%?~_!:,.;]*[-a-z0-9+&@#\/%~_!|]/i", $website)) {
        ↪ $websiteErr = "Invalid URL";
    }
}

if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = test_input($_POST["comment"]);
}

if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = test_input($_POST["gender"]);
}
?>

```

L'étape suivante consiste à montrer comment empêcher le formulaire de vider tous les champs de saisie lorsque l'utilisateur soumet le formulaire.

```

<html>
<body>
    Name: <input type="text" name="name" value="<?php echo $name; ?>">
    E-mail: <input type="text" name="email" value="<?php echo $email; ?>">
    Website: <input type="text" name="website" value="<?php echo $website; ?>">
    Comment: <textarea name="comment" rows="5" cols="40"><?php echo $comment; ?></
    ↪ textarea>
    Gender:
    <input type="radio" name="gender" <?php if (isset($gender) && $gender=="female") ↪
    ↪ echo "checked"; ?> value="female">Female
    <input type="radio" name="gender" <?php if (isset($gender) && $gender=="male") ↪
    ↪ echo "checked"; ?> value="male">Male
    <input type="radio" name="gender" <?php if (isset($gender) && $gender=="other") ↪
    ↪ echo "checked"; ?> value="other">Other
</body>
</html>

```

### 8.4.7 Base de données MySQL

Avec PHP, vous pouvez vous connecter et manipuler des bases de données.

MySQL est le système de base de données le plus populaire utilisé avec PHP.

### 8.4.8 Qu'est-ce que MySQL ?

- MySQL est un système de base de données utilisé sur le Web
- MySQL est un système de base de données qui s'exécute sur un serveur
- MySQL est idéal pour les petites et les grandes applications
- MySQL est très rapide, fiable et facile à utiliser
- MySQL utilise le SQL standard
- MySQL compile sur un certain nombre de plates-formes
- MySQL est gratuit à télécharger et à utiliser
- MySQL est développé, distribué et pris en charge par Oracle Corporation
- MySQL porte le nom de la fille du co-fondateur Monty Widenius : My

Les données d'une base de données MySQL sont stockées dans des tables. Une table est une collection de données connexes et se compose de colonnes et de lignes.

Les bases de données sont utiles pour stocker des informations de manière catégorique. Une entreprise peut avoir une base de données avec les tables suivantes :

- Employés
- Produits
- Clients
- Commandes

### 8.4.9 Système de base de données PHP + MySQL

PHP combiné à MySQL est multiplateforme (vous pouvez développer sous Windows et servir sur une plate-forme Unix)

### 8.4.10 Requêtes de base de données

Une requête est une question ou une demande.

Nous pouvons interroger une base de données pour obtenir des informations spécifiques et obtenir un jeu d'enregistrements renvoyé.

Regardez la requête suivante (en utilisant le SQL standard) :

```
SELECT Nom de famille FROM Employés
```

La requête ci-dessus sélectionne toutes les données de la colonne « Nom » de la table « Employés ».

### 8.4.11 Télécharger la base de données MySQL

Si vous n'avez pas de serveur PHP avec une base de données MySQL, vous pouvez le télécharger gratuitement ici : <http://www.mysql.com>

### 8.4.12 Faits sur la base de données MySQL

MySQL est le système de base de données standard de facto pour les sites Web avec d'énormes volumes de données et d'utilisateurs finaux (comme Facebook, Twitter et Wikipedia).

Un autre avantage de MySQL est qu'il peut être réduit pour prendre en charge les applications de base de données intégrées.

Regardez <http://www.mysql.com/customers/> pour un aperçu des entreprises utilisant MySQL.

## 8.5 PHP : se connecter à MySQL

PHP 5 et les versions ultérieures peuvent fonctionner avec une base de données MySQL en utilisant :

- Extension MySQLi (le « i » signifie amélioré)
- PDO (objets de données PHP)

Les versions antérieures de PHP utilisaient l'extension MySQL. Cependant, cette extension a été dépréciée en 2012.

## 8.6 Dois-je utiliser MySQLi ou PDO ?

Si vous avez besoin d'une réponse courte, ce serait « Tout ce que vous voulez ».

MySQLi et PDO ont tous deux leurs avantages :

PDO fonctionnera sur 12 systèmes de bases de données différents, alors que MySQLi ne fonctionnera qu'avec les bases de données MySQL.

Ainsi, si vous devez changer de projet pour utiliser une autre base de données, PDO facilite le processus. Vous n'avez qu'à changer la chaîne de connexion et quelques requêtes. Avec MySQLi, vous devrez réécrire tout le code - requêtes incluses.

Les deux sont orientés objet, mais MySQLi propose également une API procédurale.

Les deux prennent en charge les déclarations préparées (requêtes préparées). Les instructions préparées protègent de l'injection SQL et sont très importantes pour la sécurité des applications Web.

## 8.7 Exemples MySQL dans la syntaxe MySQLi et PDO

Dans cette section et dans les section suivantes, nous montrons trois façons de travailler avec PHP et MySQL :

- MySQLi (orienté objet)
- MySQLi (procédural)
- PDO

### 8.7.1 Installation de MySQLi

Pour Linux et Windows : L'extension MySQLi est installée automatiquement dans la plupart des cas, lorsque le package php5 mysql est installé.

Pour plus de détails sur l'installation, allez sur : <http://php.net/manual/en/mysqli.installation.php>

### 8.7.2 Installation PDO

Pour plus de détails sur l'installation, allez sur : <http://php.net/manual/en/pdo.installation.php>

### 8.7.3 Ouvrir une connexion à MySQL

Avant de pouvoir accéder aux données de la base de données MySQL, nous devons pouvoir nous connecter au serveur :

Exemple (MySQLi Object-Oriented)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Create connection
    $conn = new mysqli($servername, $username, $password);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }
    echo "Connected successfully";
?>
```

Exemple (MySQLi Procedural)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";

    // Create connection
    $conn = mysqli_connect($servername, $username, $password);

    // Check connection
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Connected successfully";
?>
```

Exemple PDO

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
```

(suite sur la page suivante)



(suite de la page précédente)

```

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
?>

```

### 8.7.4 Fermer la connexion

La connexion sera fermée automatiquement à la fin du script. Pour fermer la connexion avant, utilisez ce qui suit :

MySQLi Object-Oriented :

```
$conn->close();
```

MySQLi Procedural :

```
mysqli_close($conn);
```

PDO :

```
$conn = null;
```

## 8.8 Créer une base de données MySQL en utilisant MySQLi et PDO

L'instruction CREATE DATABASE est utilisée pour créer une base de données dans MySQL.

Les exemples suivants créent une base de données nommée « myDB » :

Exemple (MySQLi Object-oriented)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

```

(suite sur la page suivante)

```
$conn->close();  
?>
```

### Exemple (MySQLi Procedural)

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
  
// Create connection  
$conn = mysqli_connect($servername, $username, $password);  
// Check connection  
if (!$conn) {  
    die("Connection failed: " . mysqli_connect_error());  
}  
  
// Create database  
$sql = "CREATE DATABASE myDB";  
if (mysqli_query($conn, $sql)) {  
    echo "Database created successfully";  
} else {  
    echo "Error creating database: " . mysqli_error($conn);  
}  
  
mysqli_close($conn);  
?>
```

### Exemple (PDO)

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
  
try {  
    $conn = new PDO("mysql:host=$servername", $username, $password);  
    // set the PDO error mode to exception  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    $sql = "CREATE DATABASE myDBPDO";  
    // use exec() because no results are returned  
    $conn->exec($sql);  
    echo "Database created successfully<br>";  
} catch(PDOException $e) {  
    echo $sql . "<br>" . $e->getMessage();  
}  
  
$conn = null;  
?>
```

## 8.9 Créer une table MySQL en utilisant MySQLi et PDO

L'instruction CREATE TABLE est utilisée pour créer une table dans MySQL.

```
CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)
```

Nous allons créer une table nommée « MyGuests », avec cinq colonnes : « id », « firstname », « lastname », « email » et « reg\_date » :

Exemple (MySQLi Object-oriented)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // sql to create table
    $sql = "CREATE TABLE MyGuests (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )";

    if ($conn->query($sql) === TRUE) {
        echo "Table MyGuests created successfully";
    } else {
        echo "Error creating table: " . $conn->error;
    }

    $conn->close();
?>
```

Exemple (MySQLi Procedural)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Create connection
```

(suite sur la page suivante)

(suite de la page précédente)

```

$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// sql to create table
$sql = "CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";

if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

## Exemple (PDO)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
↪$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // sql to create table
    $sql = "CREATE TABLE MyGuests (
        id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
        firstname VARCHAR(30) NOT NULL,
        lastname VARCHAR(30) NOT NULL,
        email VARCHAR(50),
        reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
    )";

    // use exec() because no results are returned
    $conn->exec($sql);
    echo "Table MyGuests created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>

```

## 8.10 Insérer des données dans MySQL à l'aide de MySQLi et PDO

Une fois qu'une base de données et une table ont été créées, nous pouvons commencer à y ajouter des données.

Voici quelques règles de syntaxe à suivre :

- La requête SQL doit être entre guillemets en PHP
- Les valeurs de chaîne à l'intérieur de la requête SQL doivent être entre guillemets
- Les valeurs numériques ne doivent pas être entre guillemets
- Le mot NULL ne doit pas être entre guillemets

L'instruction INSERT INTO est utilisée pour ajouter de nouveaux enregistrements à une table MySQL :

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Exemple (MySQLi Object-oriented)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);
    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
    VALUES ('John', 'Doe', 'john@example.com')";

    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();
?>
```

Exemple (MySQLi Procedural)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Create connection
    $conn = mysqli_connect($servername, $username, $password, $dbname);
    // Check connection
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }

    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
```

(suite sur la page suivante)

(suite de la page précédente)

```
VALUES ('John', 'Doe', 'john@example.com');"

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

## Exemple (PDO)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDBPDO";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
↪$password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $sql = "INSERT INTO MyGuests (firstname, lastname, email)
        VALUES ('John', 'Doe', 'john@example.com')";
    // use exec() because no results are returned
    $conn->exec($sql);
    echo "New record created successfully";
} catch(PDOException $e) {
    echo $sql . "<br>" . $e->getMessage();
}

$conn = null;
?>
```

## 8.11 Requêtes préparées et paramètres liés

Une requête préparée est une fonctionnalité utilisée pour exécuter les mêmes instructions SQL (ou similaires) à plusieurs reprises avec une grande efficacité.

Les requêtes préparées fonctionnent essentiellement comme ceci :

- Préparer : un modèle d'instruction SQL est créé et envoyé à la base de données. Certaines valeurs sont laissées non spécifiées, appelées paramètres (étiquetées » ? «). Exemple : INSERT INTO MyGuests VALUES(?, ?, ?)
- La base de données analyse, compile et exécute l'optimisation des requêtes sur le modèle d'instruction SQL et stocke le résultat sans l'exécuter
- Exécuter : ultérieurement, l'application lie les valeurs aux paramètres et la base de données exécute l'instruction.

L'application peut exécuter l'instruction autant de fois qu'elle le souhaite avec des valeurs différentes

Par rapport à l'exécution directe d'instructions SQL, les instructions préparées présentent trois avantages principaux :

- Les requêtes préparées réduisent le temps d'analyse car la préparation de la requête n'est effectuée qu'une seule fois (bien que l'instruction soit exécutée plusieurs fois)

- Les paramètres liés minimisent la bande passante vers le serveur car vous devez envoyer uniquement les paramètres à chaque fois, et non la requête entière
- Les requêtes préparées sont très utiles contre les injections SQL, car les valeurs des paramètres, qui sont transmises ultérieurement à l'aide d'un protocole différent, n'ont pas besoin d'être correctement échappées. Si le modèle d'instruction d'origine n'est pas dérivé d'une entrée externe, l'injection SQL ne peut pas se produire.

Exemple (MySQLi with Prepared Statements)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDB";

    // Create connection
    $conn = new mysqli($servername, $username, $password, $dbname);

    // Check connection
    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // prepare and bind
    $stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email) VALUES_
    ↪ (?, ?, ?)");
    $stmt->bind_param("sss", $firstname, $lastname, $email);

    // set parameters and execute
    $firstname = "John";
    $lastname = "Doe";
    $email = "john@example.com";
    $stmt->execute();

    $firstname = "Mary";
    $lastname = "Moe";
    $email = "mary@example.com";
    $stmt->execute();

    $firstname = "Julie";
    $lastname = "Dooley";
    $email = "julie@example.com";
    $stmt->execute();

    echo "New records created successfully";

    $stmt->close();
    $conn->close();
?>
```

Exemple (PDO with Prepared Statements)

```
<?php
    $servername = "localhost";
    $username = "username";
    $password = "password";
    $dbname = "myDBPDO";

    try {
        $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
```

(suite sur la page suivante)

(suite de la page précédente)

```

// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// prepare sql and bind parameters
$stmt = $conn->prepare("INSERT INTO MyGuests (firstname, lastname, email)
VALUES (:firstname, :lastname, :email)");
$stmt->bindParam(':firstname', $firstname);
$stmt->bindParam(':lastname', $lastname);
$stmt->bindParam(':email', $email);

// insert a row
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

// insert another row
$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

// insert another row
$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();

echo "New records created successfully";
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null;
?>

```

## 8.12 Sélectionnez les données d'une base de données MySQL

L'instruction SELECT permet de sélectionner des données dans une ou plusieurs tables :

SELECT column\_name(s) FROM table\_name

ou nous pouvons utiliser le caractère \* pour sélectionner TOUTES les colonnes d'une table :

SELECT \* FROM nom\_table

Exemple (MySQLi Object-oriented)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

```

(suite sur la page suivante)



(suite de la page précédente)

```

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row[
↪ "lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

## Exemple (MySQLi Procedural)

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row[
↪ "lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>

```



### 9.1 Qu'est-ce qu'AJAX ?

AJAX = JavaScript et XML asynchrones.

AJAX est une technique de création de pages Web rapides et dynamiques.

AJAX permet aux pages Web d'être mises à jour de manière asynchrone en échangeant de petites quantités de données avec le serveur dans les coulisses. Cela signifie qu'il est possible de mettre à jour des parties d'une page Web, sans recharger la page entière.

Les pages Web classiques (qui n'utilisent pas AJAX) doivent recharger la page entière si le contenu devait changer.

Exemples d'applications utilisant AJAX : onglets Google Maps, Gmail, Youtube et Facebook.

### 9.2 AJAX est basé sur les normes Internet

AJAX est basé sur les normes Internet et utilise une combinaison de :

- Objet XMLHttpRequest (pour échanger des données de manière asynchrone avec un serveur)
- JavaScript/DOM (pour afficher/interagir avec les informations)
- CSS (pour styliser les données)
- XML (souvent utilisé comme format de transfert de données)

## 9.3 AJAX et PHP

Exemple

```
<html>
<head>
  <script>
    function showHint(str) {
      if (str.length == 0) {
        document.getElementById("txtHint").innerHTML = "";
        return;
      } else {
        var xmlhttp = new XMLHttpRequest();
        xmlhttp.onreadystatechange = function() {
          if (this.readyState == 4 && this.status == 200) {
            document.getElementById("txtHint").innerHTML = this.responseText;
          }
        };
        xmlhttp.open("GET", "gethint.php?q=" + str, true);
        xmlhttp.send();
      }
    }
  </script>
</head>
<body>

  <p><b>Start typing a name in the input field below:</b></p>
  <form action="">
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname" onkeyup="showHint(this.value)">
  </form>
  <p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

### 9.3.1 Explication du code :

Tout d'abord, vérifiez si le champ de saisie est vide (`str.length == 0`). Si c'est le cas, effacez le contenu de l'espace réservé `txtHint` et quittez la fonction.

Cependant, si le champ de saisie n'est pas vide, procédez comme suit :

- Créer un objet `XMLHttpRequest`
- Créer la fonction à exécuter lorsque la réponse du serveur est prête
- Envoyez la demande à un fichier PHP (`gethint.php`) sur le serveur
- Notez que le paramètre `q` est ajouté à l'url (`gethint.php?q= »+str`)
- Et la variable `str` contient le contenu du champ de saisie

### 9.3.2 Le fichier PHP - « gethint.php »

Le fichier PHP vérifie un tableau de noms et renvoie le(s) nom(s) correspondant(s) au navigateur :

```
<?php
    // Array with names
    $a[] = "Anna";
    $a[] = "Brittany";
    $a[] = "Cinderella";
    $a[] = "Diana";
    $a[] = "Eva";
    $a[] = "Fiona";
    $a[] = "Gunda";
    $a[] = "Hege";
    $a[] = "Inga";
    $a[] = "Johanna";
    $a[] = "Kitty";
    $a[] = "Linda";
    $a[] = "Nina";
    $a[] = "Ophelia";
    $a[] = "Petunia";
    $a[] = "Amanda";
    $a[] = "Raquel";
    $a[] = "Cindy";
    $a[] = "Doris";
    $a[] = "Eve";
    $a[] = "Evita";
    $a[] = "Sunniva";
    $a[] = "Tove";
    $a[] = "Unni";
    $a[] = "Violet";
    $a[] = "Liza";
    $a[] = "Elizabeth";
    $a[] = "Ellen";
    $a[] = "Wenche";
    $a[] = "Vicky";

    // get the q parameter from URL
    $q = $_REQUEST["q"];

    $hint = "";

    // lookup all hints from array if $q is different from ""
    if ($q != "") {
        $q = strtolower($q);
        $len=strlen($q);
        foreach($a as $name) {
            if (stristr($q, substr($name, 0, $len))) {
                if ($hint == "") {
                    $hint = $name;
                } else {
                    $hint .= ", $name";
                }
            }
        }
    }

    // Output "no suggestion" if no hint was found or output correct values
```

(suite sur la page suivante)

```
echo $hint === "" ? "no suggestion" : $hint;
?>
```

## 9.4 Exemple de base de données AJAX

AJAX peut être utilisé pour une communication interactive avec une base de données. Exemple de base de données AJAX

L'exemple suivant montre comment une page Web peut récupérer des informations à partir d'une base de données avec AJAX :

Exemple expliqué

Dans l'exemple ci-dessus, lorsqu'un utilisateur sélectionne une personne dans la liste déroulante ci-dessus, une fonction appelée « showUser() » est exécutée.

La fonction est déclenchée par l'événement onchange.

Voici le code HTML :

Exemple

```
<html>
<head>
<script>
    function showUser(str) {
        if (str == "") {
            document.getElementById("txtHint").innerHTML = "";
            return;
        } else {
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.onreadystatechange = function() {
                if (this.readyState == 4 && this.status == 200) {
                    document.getElementById("txtHint").innerHTML = this.responseText;
                }
            };
            xmlhttp.open("GET", "getuser.php?q="+str, true);
            xmlhttp.send();
        }
    }
</script>
</head>
<body>

<form>
    <select name="users" onchange="showUser(this.value)">
        <option value="">Select a person:</option>
        <option value="1">Peter Griffin</option>
        <option value="2">Lois Griffin</option>
        <option value="3">Joseph Swanson</option>
        <option value="4">Glenn Quagmire</option>
    </select>
</form>
<br>
<div id="txtHint"><b>Person info will be listed here...</b></div>
```

(suite sur la page suivante)

(suite de la page précédente)

```
</body>
</html>
```

Explication du code :

Tout d'abord, vérifiez si la personne est sélectionnée. Si aucune personne n'est sélectionnée (str == « »), effacez le contenu de txtHint et quittez la fonction. Si une personne est sélectionnée, procédez comme suit :

Créer un objet XMLHttpRequest  
Créer la fonction à exécuter lorsque la réponse du serveur est prête  
Envoyer la demande à un fichier sur le serveur  
Notez qu'un paramètre (q) est ajouté à l'URL (avec le contenu de la liste déroulante)

### 9.4.1 Le fichier PHP

La page sur le serveur appelée par le JavaScript ci-dessus est un fichier PHP appelé « getuser.php ».

Le code source dans « getuser.php » exécute une requête sur une base de données MySQL et renvoie le résultat dans une table HTML :

```
<!DOCTYPE html>
<html>
<head>
  <style>
    table {
      width: 100%;
      border-collapse: collapse;
    }
    table, td, th {
      border: 1px solid black;
      padding: 5px;
    }
    th {text-align: left;}
  </style>
</head>
<body>

<?php
  $q = intval($_GET['q']);
  $con = mysqli_connect('localhost','peter','abc123','my_db');
  if (!$con) {
    die('Could not connect: ' . mysqli_error($con));
  }
  mysqli_select_db($con,"ajax_demo");
  $sql="SELECT * FROM user WHERE id = '". $q ."'";
  $result = mysqli_query($con,$sql);
  echo "<table>
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
    <th>Hometown</th>
    <th>Job</th>
  </tr>";
  while($row = mysqli_fetch_array($result)) {
    echo "<tr>";
```

(suite sur la page suivante)

(suite de la page précédente)

```
        echo "<td>" . $row['FirstName'] . "</td>";
        echo "<td>" . $row['LastName'] . "</td>";
        echo "<td>" . $row['Age'] . "</td>";
        echo "<td>" . $row['Hometown'] . "</td>";
        echo "<td>" . $row['Job'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
    mysqli_close($con);
?>
</body>
</html>
```

Explication : Lorsque la requête est envoyée du JavaScript vers le fichier PHP, les événements suivants se produisent :

- PHP ouvre une connexion à un serveur MySQL
- La bonne personne est trouvée
- Un tableau HTML est créé, rempli de données et renvoyé à l'espace réservé « txtHint »



### 10.1 Objectif de l'architecture de service Web

Les services Web fournissent un moyen standard d'interopérabilité entre différentes applications logicielles, s'exécutant sur une variété de plates-formes et/ou de frameworks. Le WSA fournit un modèle conceptuel et un contexte pour comprendre les services Web et les relations entre les composants de ce modèle.

Le WSA décrit à la fois les caractéristiques minimales communes à tous les services Web et un certain nombre de caractéristiques nécessaires à de nombreux services Web, mais pas à tous.

L'architecture de services Web est une architecture d'interopérabilité : elle identifie les éléments globaux du réseau mondial de services Web qui sont nécessaires pour assurer l'interopérabilité entre les services Web.

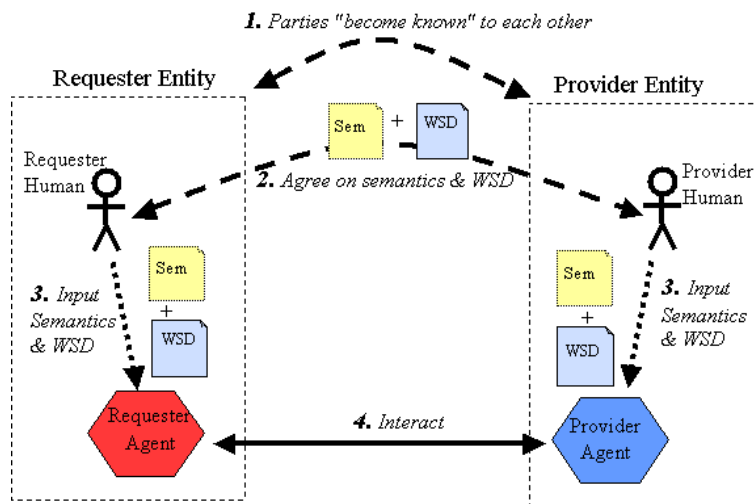
#### 10.1.1 Agents et services

Un service Web est une notion abstraite qui doit être implémentée par un agent concret. (Voir Figure 1-1) L'agent est la partie concrète du logiciel ou du matériel qui envoie et reçoit des messages, tandis que le service est la ressource caractérisée par l'ensemble abstrait de fonctionnalités qui est fourni. Pour illustrer cette distinction, vous pouvez implémenter un service Web particulier en utilisant un jour un agent (peut-être écrit dans un langage de programmation) et un autre agent le lendemain (peut-être écrit dans un langage de programmation différent) avec la même fonctionnalité. Bien que l'agent puisse avoir changé, le service Web reste le même.

### 10.1.2 Demandeurs et fournisseurs

Le but d'un service Web est de fournir certaines fonctionnalités au nom de son propriétaire - une personne ou une organisation, telle qu'une entreprise ou un individu. L'entité fournisseur est la personne ou l'organisation qui fournit un agent approprié pour mettre en œuvre un service particulier. (Voir Figure 1-1 : Rôles architecturaux de base.)

Une entité demandeuse est une personne ou une organisation qui souhaite utiliser le service Web d'une entité fournisseur. Il utilisera un agent demandeur pour échanger des messages avec l'agent fournisseur de l'entité fournisseur.



### 10.1.3 Description du service

Les mécanismes de l'échange de messages sont documentés dans une description de service Web (WSD). (Voir Figure 1-1) Le WSD est une spécification exploitable par machine de l'interface du service Web, écrite en WSDL. Il définit les formats de message, les types de données, les protocoles de transport et les formats de sérialisation de transport qui doivent être utilisés entre l'agent demandeur et l'agent fournisseur. Il spécifie également un ou plusieurs emplacements de réseau auxquels un agent fournisseur peut être invoqué, et peut fournir des informations sur le modèle d'échange de messages qui est attendu. Essentiellement, la description du service représente un accord régissant les mécanismes d'interaction avec ce service.

### 10.1.4 Sémantique

La sémantique d'un service Web est l'attente partagée sur le comportement du service, notamment en réponse aux messages qui lui sont envoyés. En effet, il s'agit du « contrat » entre l'entité demandeuse et l'entité fournisseur concernant le but et les conséquences de l'interaction. Bien que ce contrat représente l'accord global entre l'entité demandeuse et l'entité fournisseur sur comment et pourquoi leurs agents respectifs interagissent, il n'est pas nécessairement écrit ou explicitement négocié. Il peut être explicite ou implicite, oral ou écrit, traitable par machine ou orienté vers l'homme, et il peut s'agir d'un accord juridique ou d'un accord informel (non juridique). (Encore une fois, il s'agit d'une légère simplification qui sera expliquée plus en détail dans 3.3 Utilisation des services Web.)

Alors que la description du service représente un contrat régissant les mécanismes d'interaction avec un service particulier, la sémantique représente un contrat régissant le sens et le but de cette interaction. La ligne de démarcation entre ces deux n'est pas nécessairement rigide. Comme des langages plus riches sur le plan sémantique sont utilisés pour décrire les mécanismes de l'interaction, davantage d'informations essentielles peuvent migrer de la sémantique informelle vers la description du service. Au fur et à mesure que cette migration se produit, une plus grande partie du travail requis pour obtenir une interaction réussie peut être automatisée.

### 10.1.5 Définitions

Les services Web sont des composants d'application Web. Les services Web peuvent être publiés, trouvés et utilisés sur le Web.

### 10.1.6 WSDL

- WSDL signifie Web Services Description Language
- WSDL est un langage basé sur XML pour décrire des services Web.
- WSDL est une recommandation du W3C

### 10.1.7 SOAP

- SOAP signifie Simple Object Access Protocol
- SOAP est un protocole basé sur XML pour accéder aux services Web.
- SOAP est basé sur XML
- SOAP est une recommandation du W3C

### 10.1.8 RDF

- RDF signifie Resource Description Framework
- RDF est un framework pour décrire des ressources sur le web
- RDF est écrit en XML
- RDF est une recommandation du W3C

### 10.1.9 RSS

- RSS signifie Really Simple Syndication
- RSS vous permet de syndiquer le contenu de votre site
- RSS définit un moyen facile de partager et d'afficher les titres et le contenu
- Les fichiers RSS peuvent être automatiquement mis à jour
- RSS permet des vues personnalisées pour différents sites
- RSS est écrit en XML



---

## WebAssembly ou WASM

---

WebAssembly est un nouveau type de code qui peut être exécuté dans les navigateurs Web modernes. Il s'agit d'un langage de bas niveau de type assembleur avec un format binaire compact qui s'exécute avec des performances quasi natives et fournit des langages tels que C/C++, C# et Rust avec une cible de compilation afin qu'ils puissent s'exécuter sur le Web. Il est également conçu pour fonctionner avec JavaScript, permettant aux deux de fonctionner ensemble.

WebAssembly a d'énormes implications pour les plate-formes Web. Il fournit un moyen d'exécuter du code écrit dans plusieurs langues sur le Web à une vitesse quasi native, avec des applications clientes s'exécutant sur le Web qui n'auraient pas pu le faire auparavant.

WebAssembly est conçu pour compléter et s'exécuter avec JavaScript — en utilisant les API JavaScript WebAssembly, vous pouvez charger des modules WebAssembly dans une application JavaScript et partager des fonctionnalités entre les deux. Cela vous permet de profiter des performances et de la puissance de WebAssembly ainsi que de l'expressivité et de la flexibilité de JavaScript dans les mêmes applications, même si vous ne savez pas comment écrire du code WebAssembly.

### 11.1 Compilation d'un nouveau module C/C++ vers WebAssembly

Lorsque vous avez écrit un nouveau module de code dans un langage comme C/C++, vous pouvez le compiler dans WebAssembly à l'aide d'un outil comme Emscripten.

#### 11.1.1 Configuration de l'environnement Emscripten

Tout d'abord, configurons l'environnement de développement requis.

Obtenez le SDK Emscripten en suivant ces instructions : [https://emscripten.org/docs/getting\\_started/downloads.html](https://emscripten.org/docs/getting_started/downloads.html)

### 11.1.2 Compiler un exemple

Une fois l'environnement configuré, voyons comment l'utiliser pour compiler un exemple C vers Emscripten. Un certain nombre d'options sont disponibles lors de la compilation avec Emscripten, mais les deux principaux scénarios que nous couvrirons sont :

- Compilation vers wasm et création du code HTML pour exécuter notre code, ainsi que tout le code JavaScript nécessaire pour exécuter le wasm dans l'environnement Web.
- Compilation en wasm et création du JavaScript.

Nous examinerons les deux ci-dessous.

### 11.1.3 Création HTML et JavaScript

C'est le cas le plus simple que nous examinerons, dans lequel vous obtenez emscripten pour générer tout ce dont vous avez besoin pour exécuter votre code, en tant que WebAssembly, dans le navigateur.

Nous avons d'abord besoin d'un exemple à compiler. Prenez une copie de l'exemple C simple suivant et enregistrez-le dans un fichier appelé `hello.c` dans un nouveau répertoire sur votre disque local :

```
#include <stdio.h>

int main() {
    printf("Hello World\n");
}
```

Maintenant, à l'aide de la fenêtre de terminal que vous avez utilisée pour entrer dans l'environnement du compilateur Emscripten, accédez au même répertoire que votre fichier `hello.c` et exécutez la commande suivante :

```
emcc hello.c -s WASM=1 -o hello.html
```

Les options que nous avons transmises avec la commande sont les suivantes :

- `-s WASM=1` — Spécifie que nous voulons une sortie wasm. Si nous ne le spécifions pas, Emscripten affichera simplement `asm.js`, comme il le fait par défaut.
- `-o hello.html` — Spécifie que nous voulons qu'Emscripten génère une page HTML pour exécuter notre code (et un nom de fichier à utiliser), ainsi que le module wasm et le code JavaScript « glue » pour compiler et instancier le wasm afin qu'il peut être utilisé dans l'environnement Web.

À ce stade dans votre répertoire source, vous devriez avoir :

- Le code binaire du module wasm (`hello.wasm`)
- Un fichier JavaScript contenant du code de collage à traduire entre les fonctions C natives et JavaScript/wasm (`hello.js`)
- Un fichier HTML pour charger, compiler et instancier votre code wasm et afficher sa sortie dans le navigateur (`hello.html`)

Exécuter votre exemple

Il ne vous reste plus qu'à charger le `hello.html` résultant dans un navigateur prenant en charge WebAssembly.

## CHAPITRE 12

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`