

Développement web côté client

Cours 3 : XML et Bootstrap

Hai Nam TRAN

Université de Bretagne Occidentale – L2 INFO

Cours 3

- **Objective**

- **Présenter les aspects essentiels de XML de manière concise et illustrée par de nombreux exemples**
- **Illustrer la gestion de base de données par une interface web**
 - Affichage de données
 - Add/Edit/Delete/Update
- **Présenter le framework Bootstrap de manière introductif**

Références

[1] L'essentiel de XML – Oliver Carton

[2] Introduction au langage XML – Nhan LE THANH

[3] XML | Short Course | Step by Step for Beginners – Raghav Pal

[4] Learning XML, 2nd Edition - Erik T. Ray

[5] <https://getbootstrap.com/docs/5.2/getting-started/introduction/>

[6] <https://www.pierre-giraud.com/bootstrap-apprendre-cours/>

XML

Cours 3

Jalon 4

Hai Nam TRAN

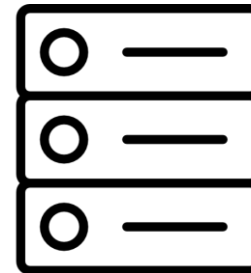
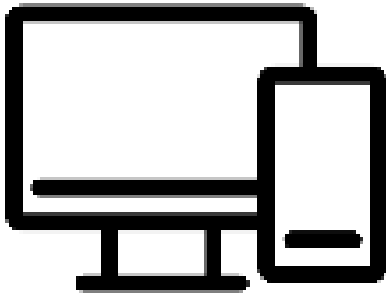
Université de Bretagne Occidentale – L2 INFO

Introduction

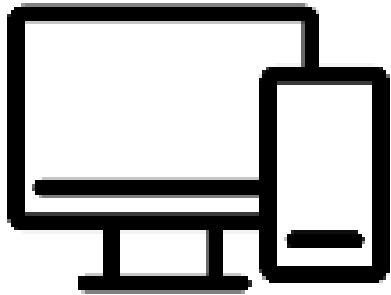
Il y a 3 étudiants
dans mon groupe :
Rémi, Léo et Clément



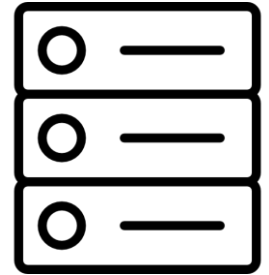
Il y a 3 étudiants
dans mon groupe :
Rémi, Léo et Clément



Introduction



```
<group>
  <student>
    <name>Rémi</name>
  </student>
  <student>
    <name>Léo</name>
  </student>
  <student>
    <name>Clément</name>
  </student>
</group>
```



- **Messages de communication**
 - Structuré
 - Formaté
 - Syntaxe et règles générales

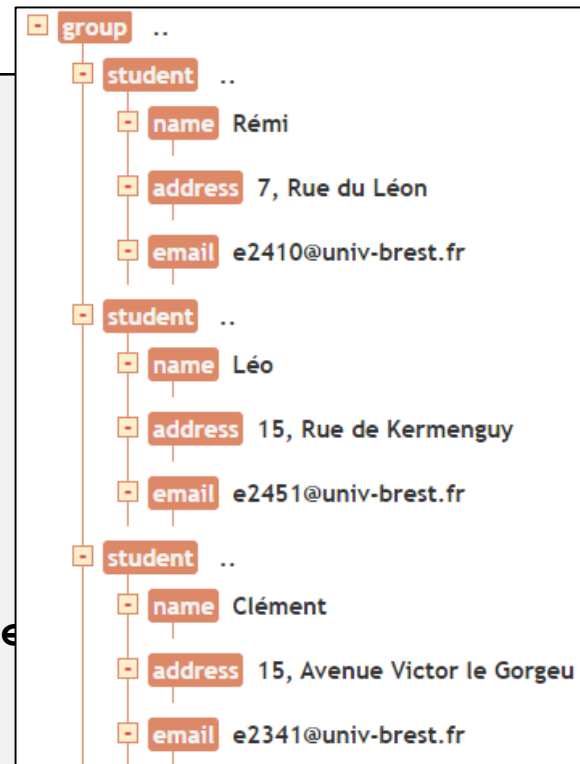
Introduction

- **XML – eXtensible Markup Language**
 - **Markup** : balises (<>) sont utilisées dans XML
 - **eXtensible** : on peut créer des balises
 - L'ensemble des balises autorisées n'est pas figé comme **HTML**
 - **Recommandation W3C (World Wide Web Consortium) pour**
 - L'échange, la transformation, l'intégration et l'interrogation des données sur le Web
 - **Caractéristiques**
 - **Principe clé** : séparation stricte entre contenu et présentation
 - Cette idée existe aussi dans le langage HTML
 - La séparation est plus marquée en XML
 - Simplicité, universalité et extensibilité
 - Structuration forte
 - Modèles de documents
 - Format libre

Introduction

group.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<group>
  <student>
    <name>Rémi</name>
    <address>7, Rue du Léon</address>
    <email>e2410@univ-brest.fr</email>
  </student>
  <student>
    <name>Léo</name>
    <address>15, Rue de Kermenguy</address>
    <email>e2451@univ-brest.fr</email>
  </student>
  <student>
    <name>Clément</name>
    <address>15, Avenue Victor le Gorgeu</address>
    <email>e2341@univ-brest.fr</email>
  </student>
</group>
```



Topics à traiter

DWCC

1. Syntaxe de XML

2. XML Namespace

DWCC

3. DOM

4. DTD

5. XPATH

6. XSLT

7. XML Schéma

DWCC

8. XML Parseur

Notre besoin : lire et afficher le contenu d'un document XML dans une page HTML

Syntaxe de XML

group.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<group>
  <student>
    <name>Rémi</name>
    <address>7, Rue du Léon</address>
    <email>e2410@univ-brest.fr</email>
  </student>
  <student>
    <name>Léo</name>
    <address>15, Rue de Kermenguy</address>
    <email>e2451@univ-brest.fr</email>
  </student>
  <student>
    <name>Clément</name>
    <address>15, Avenue Victor le Gorgeu</address>
    <email>e2341@univ-brest.fr</email>
  </student>
</group>
```

Prologue : déclaration XML, la DTD, des commentaires, des instructions de traitements

Contenu du document :
Élément racine + contenu

Syntaxe de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is a comment -->
<group>
  <student gender="male">
    <name>Rémi</name>
    <address>7, Rue du Léon</address>
    <email>e2410@univ-brest.fr</email>
  </student>
  ...
</group>
```

Well-formed XML

- XML Prologue : `<?xml version="1.0" encoding="UTF-8"?>`
- Élément racine - le parent de tous les éléments : `<group>`
- Balises, valeurs, attributs
 - `<balise_ouvrante attribut="attr">valeur</balise_fermante>`
- Imbrication doit être faite correctement
- Commentaires : `<!-- This is a comment -->`

DOM

- **Document a une forme arborescente**

- **DOM – Document Object Model: Un arbre, constitué de noeuds typés (éléments, commentaires, valeurs, etc)**
 - Une représentation orientée objet du document XML dans la mémoire
 - Faciliter la lecture et l'écriture du contenu de document XML
 - Application Programming Interface (API)
 - On peut utiliser JavaScript pour manipuler un document XML (et HTML)
 - Live exemple : https://www.w3schools.com/xml/tryit.asp?filename=try_dom_loadxmltext



```
xmlDoc.getElementsByTagName("address")[0].childNodes[0].nodeValue;
```



```
7, Rue du Léon
```

XSLT : Transformations XML Style Language

- XSL : une feuille de style d'un document XML

XML

```
▼ <CATALOG>
  ▼ <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  ▼ <CD>
    <TITLE>Hide your heart</TITLE>
    <ARTIST>Bonnie Tyler</ARTIST>
    <COUNTRY>UK</COUNTRY>
    <COMPANY>CBS Records</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1988</YEAR>
  </CD>
  ▼ <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
</CATALOG>
```

XSL

HTML

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook

[https://www.w3schools.com/xml/tryit.asp?file
name=try_cdcatalog](https://www.w3schools.com/xml/tryit.asp?file=name=try_cdcatalog)

XSLT : Transformations XML Style Language

Même si XSLT est une solution très intéressante pour formater le contenu d'un fichier XML, nous n'avons pas assez de temps de l'aborder dans le cadre de cette UE. Au lieu de cela, nous utiliserons JavaScript pour faire ce travail.

XML Parseur

- **Avant qu'un document XML soit accessible, il doit être chargé dans un objet DOM**
 - **Tous les navigateurs modernes ont un XML parseur intégré qui peut convertir du texte en un objet DOM.**

XML Parseur

https://www.w3schools.com/xml/tryit.asp?filename=try_dom_loadxmltext

```
<p id="demo"></p>
```

```
<script>
```

```
var parser, xmlDoc;
```

```
var text = "<bookstore><book>" +
```

Contenu XML

```
"<title>Everyday Italian</title>" +
```

```
"<author>Giada De Laurentiis</author>" +
```

```
"<year>2005</year>" +
```

```
"</book></bookstore>";
```

XML DOM Parseur

```
parser = new DOMParser();
```

```
xmlDoc = parser.parseFromString(text, "text/xml");
```

XML DOM Objet

```
let demo = document.getElementById("demo");
```

```
let val =
```

```
xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue;
```

```
demo.innerHTML = val
```

```
</script>
```

XMLHttpRequest

- **Les objets XMLHttpRequest (XHR) permettent d'interagir avec des serveurs**
 - Récupérer des données à partir d'une URL
 - XML Parseur intégré

```
XMLHttpRequest.open(method, url, async, user, password)
```

- **Instancie une nouvelle requête**
- **method** : La méthode de requête HTTP à utiliser telles que GET, POST, PUT, DELETE
- **url** : Une DOMString représentant l'URL à laquelle envoyer la requête
- **async** (facultatif) : true - traitement de la requête en **asynchrone**
- **user** (facultatif) : à utiliser dans un but d'authentification
- **password** (facultatif) : à utiliser dans un but d'authentification

XMLHttpRequest

- **Synchrone**

- Une communication en temps réel pendant laquelle chaque partie reçoit les messages (et, si nécessaire, les traite et y répond) **dès que possible après qu'ils aient été envoyés**
 - Exemple : le téléphone - au cours d'un appel téléphonique vous avez tendance à répondre à la personne immédiatement
 - Temps d'attente après envoyer un message

- **Asynchrone**

- Un environnement de communication où chaque partie reçoit et traite les messages **lorsque c'est possible ou plus pratique**, au lieu de le faire au même moment
 - Exemple : le courriel - l'expéditeur envoie un courriel et le destinataire se contente d'y répondre quand cela lui convient
 - Pas de temps d'attente pour la réponse

<https://developer.mozilla.org/fr/docs/Glossary/Synchronous>
<https://developer.mozilla.org/fr/docs/Glossary/Asynchronous>

XMLHttpRequest

```
XMLHttpRequest.send()
```

- Envoie la requête au serveur
- Si la requête est **asynchrone** (par défaut)
 - La méthode envoie un retour dès que la requête est partie et le **résultat est intégré en utilisant les évènements**
 - Synchrones, elle ne renvoie rien tant que la réponse n'est pas retournée.

```
XMLHttpRequest.onreadystatechange = callback
```

- Un gestionnaire d'évènement invoqué à chaque fois que l'attribut `readyState` change
- **callback** : la fonction exécutée lorsque **readyState** change

XMLHttpRequest

`XMLHttpRequest.readyState`

- La **propriété** `XMLHttpRequest.readyState` renvoie l'état dans lequel se trouve un client `XMLHttpRequest`

Value	State	Description
0	UNSENT	Le client a été créé. <code>open()</code> n'a pas encore été appelé.
1	OPENED	<code>open()</code> a été appelé.
2	HEADERS_RECEIVED	<code>send()</code> a été appelé, et les en-têtes et le statut sont disponibles.
3	LOADING	Téléchargement; <code>responseText</code> contient des données partielles.
4	DONE	L'opération est terminée.

XMLHttpRequest

`XMLHttpRequest.status`

- La **propriété** en lecture seule renvoie le code d'état HTTP numérique de la réponse de XMLHttpRequest.
 - 200 OK : indique la réussite d'une requête
 - Ref : Codes de réponse HTTP - <https://developer.mozilla.org/fr/docs/Web/HTTP/Status>

`XMLHttpRequest.responseXML`

- La propriété en lecture seule renvoie un document contenant le code XML récupéré par la requête
- Permettre d'obtenir les données de la réponse sous forme de données XML
 - XMLHttpRequest a un XML parseur intégré

XMLHttpRequest

```
var xmlhttp = new XMLHttpRequest();
function loadXMLDoc() {
    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
            myFunction();
        }
    };
    xmlhttp.open("GET", "https://phobos.univ-
    brest.fr/media/edu/kx07kar7/book_catalog.xml", true);
    xmlhttp.send();
}
```

- Envoyer une requête XMLHttpRequest asynchrone pour récupérer **book_catalog.xml**
- Lorsque la requête est réussie, appeler la fonction **myFunction()**

XMLHttpRequest

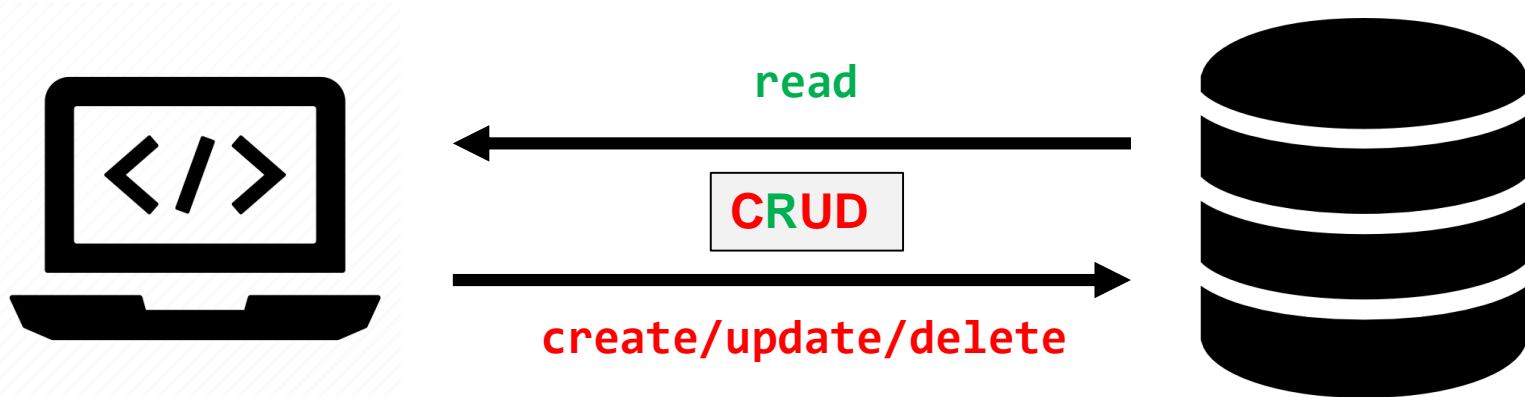
myFunction()

```
var i;
var xmlDoc = xmlhttp.responseXML;
var table = "<tr><th>Book</th><th>Authors</th></tr>";
var x = xmlDoc.getElementsByTagName("BOOK");
for (i = 0; i < x.length; i++) {
    table += "<tr><td>"
    + x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue
    + "</td><td>"
    + x[i].getElementsByTagName("AUTHOR")[0].childNodes[0].nodeValue
    + "</td>"
    + "<td><button type=\"button\" onclick=\"editBook(\""
    + x[i].getElementsByTagName("ID")[0].childNodes[0].nodeValue
    + ")\">"
    + "Edit</button></td></tr>";
}
document.getElementById("demo").innerHTML = table;
```

XMLHttpRequest

Book	Authors	
Really Friendly Git Intro	Tracy Osborn	<input type="button" value="Edit"/>
The Object-Oriented Thought Process	Matt Weisfeild	<input type="button" value="Edit"/>
The Art of Unit Testing	Roy Osherove	<input type="button" value="Edit"/>
Head First Design Patterns	Eric Freeman, Elisabeth Robson	<input type="button" value="Edit"/>

Interaction BDD – Application Web



CM

- 1-Introduction: HTML, CSS, et JS
- Rappel : Technologies Informatiques (L1 ISI) - Partie Web
- Devoir 1 - à rendre avant le 23/10
- Serveur Discord
- 2-JavaScript : animations & validation de données (MAJ : 04/11/2022)
- Devoir 2 - à rendre avant le 06/11
- 3-XML et Bootstrap

Modifier

U Modifier


D Modifier

C Modifier

+ Ajouter une activité ou ressource

Interaction BDD – Application Web

×➔



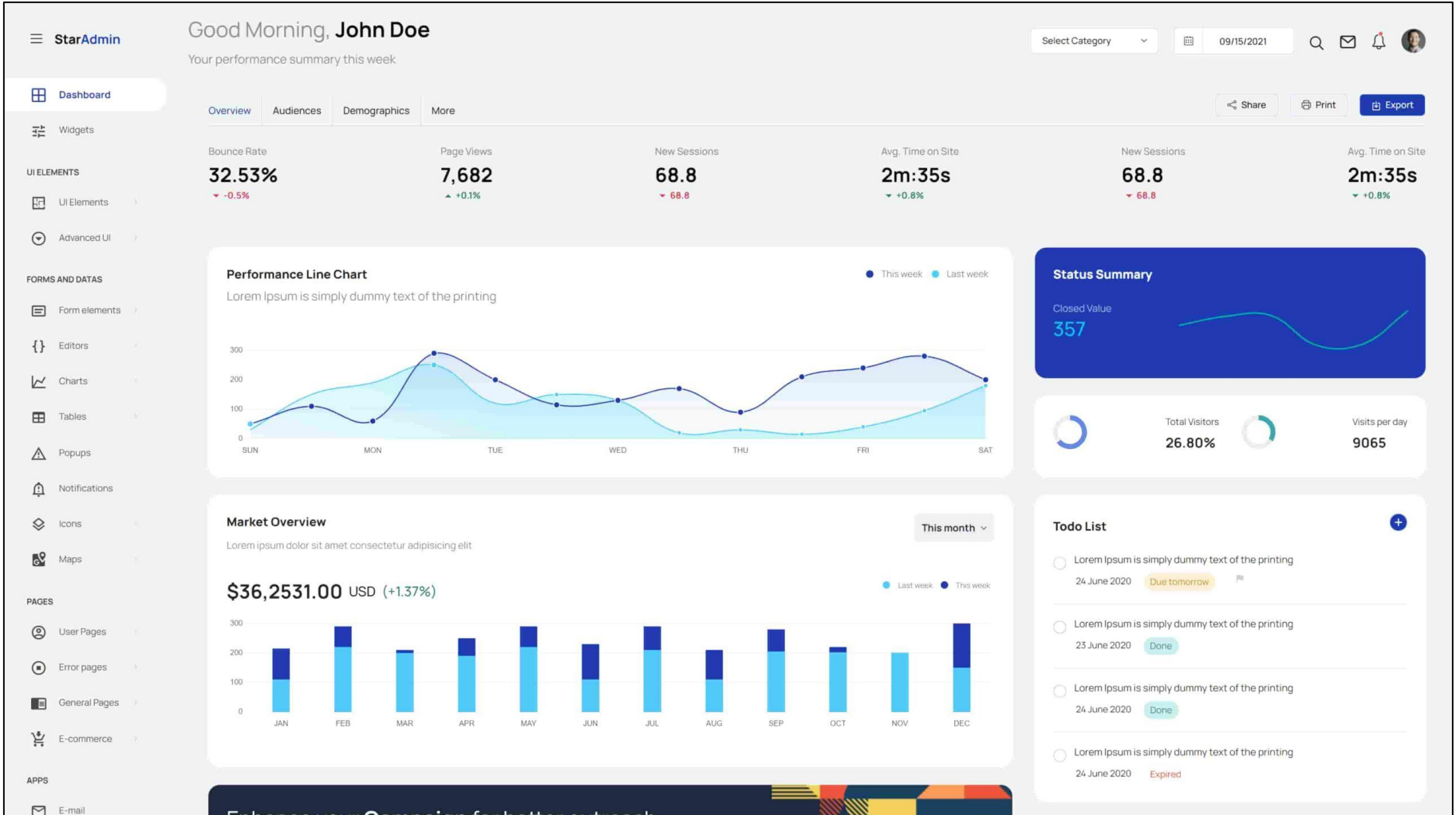
Hello,

- Dashboard
- Customers
- Category
- Sizes
- Product Sizes
- Products
- Orders

Product Sizes Item

S.N.	Product Name	Size	Stock Quantity	Action	
1	<input type="text"/>	L	10	Edit	Delete
2	<input type="text"/>	L	8	Edit	Delete
3	<input type="text"/>	L	10	Edit	Delete
4	<input type="text"/>	L	10	Edit	Delete
5	<input type="text"/>	M	9	Edit	Delete
6	<input type="text"/>	M	6	Edit	Delete
7	<input type="text"/>	Free	5	Edit	Delete

Interaction BDD – Application Web



Interaction BDD – Application Web

R

C

My book collection

Add

Book	Authors	U	D
Really Friendly Git Intro	Tracy Osborn	Edit	Delete
The Object-Oriented Thought Process	Matt Weisfeld	Edit	Delete
The Art of Unit Testing	Roy Osherove	Edit	Delete
Head First Design Patterns	Eric Freeman, Elisabeth Robson	Edit	Delete

Book Title	<input type="text" value="Really Friendly Git Intro"/>
Authors	<input type="text" value="Tracy Osborn"/>
	<input type="button" value="Update"/> <input type="button" value="Save"/>

Jalon 4

Jalon 4

- XML - Base de données

- Créer une base de données et exporter les données en format XML
- Afficher les données sur une page HTML
- **CRUD** : implémenter une interface web pour les quatre opérations de base d'une base de données

Tutoriel XML + BDD

Bootstrap

Cours 3

Jalon 5

Hai Nam TRAN

Université de Bretagne Occidentale – L2 INFO

Qu'est-ce que Bootstrap ?

Framework : un ensemble de bibliothèques regroupées dans un but précis et possédant des règles internes que doivent suivre les utilisateurs.

- **Un front-end toolkit**

- **Un framework CSS/JS**

- Un ensemble de fichiers CSS et JavaScript qui contiennent des règles prédéfinies et qui définissent des composants



New in v5.2 CSS variables, responsive offcanvas, new utilities, and more!

Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

Bootstrap : Avantages et limites

Utiliser Bootstrap VS tout coder "à la main"

- **Avantages**

- Un gain de temps de développement significatif
- Améliorer la cohérence de votre site web.
- Une certaine robustesse dans l'architecture globale du code

- **Limites**

- **Coût d'apprentissage : est-ce que c'est facile d'apprendre et d'utiliser Bootstrap**
 - **Oui** - si vous avez une bonne connaissance de HTML, CSS, JS
- **Un framework relativement lourd à charger**
- **Limite nos possibilités de design en dehors de ce qui est prédéfini dans Bootstrap** (vrai si vous ne maîtrisez pas bien HTML, CSS, JS, sinon "the sky is the limit")

Comment utiliser Bootstrap ?

- **Lier Bootstrap à nos fichiers HTML**
 - Comme les fichier CSS, JavaScript
 - Deux possibilités :
 - Télécharger les fichiers Bootstrap (CSS et JavaScript) sur le site <https://getbootstrap.com/>
 - Utiliser un CDN (Content Delivery Network ou réseau de distribution de contenu) et inclure le lien vers les fichiers dans nos fichiers HTML

<https://getbootstrap.com/docs/5.2/getting-started/introduction/>

Comment utiliser Bootstrap ?

- **Systeme de disposition (Layout)**

- Bootstrap utilise un système de « grid » à 12 colonnes de base reposant sur le modèle des boites flexibles (flexbox) comme système de disposition principal

- **Composants Bootstrap**

- Liste complète : <https://getbootstrap.com/docs/5.2/getting-started/introduction/>
- C'est possible d'utiliser ces composants sans le système de disposition Bootstrap

Comment utiliser Bootstrap ?

Démonstration de composants Bootstrap


Jalon 5 (Bonus)

- **Bootstrap**

- Avant de commencer le jalon 5, finaliser et sauvegarder votre projet
- Créer un dossier projet_vbootstrap
- Copier le contenu de votre projet dans ce nouveau dossier
- Appliquer le framework Bootstrap à cette version

Dépôt finale

- Un fichier zip nommé **depot_finale.zip**, ce fichier contient :
 - dossier **devoir** : devoir 1 + 2
 - dossier **tp** : tp 1 2 3 4
 - dossier **td** : td 1 2
 - dossier **projet**
 - html/css/js
 - la conception du gabarit de vos pages web
 - dossier **projet_vbootstrap**



Veuillez nommer le dossier comme indiqué ci-dessus !