

TCH056 - Programmation Web

1. Description du cours (annuaire)

Ce cours vise l'initiation aux rudiments de la programmation Web. Au terme de ce cours, l'étudiant ou l'étudiante sera en mesure:

- de développer des applications Web dynamiques;
- d'expliquer les particularités d'une application Web et ses modèles de communication ;
- d'exploiter les langages côté client : balisage HTML, feuilles de styles et manipulation dynamique du modèle-objet (DOM) de façon programmatique;
- de mettre en place une architecture typique n-tiers comprenant un serveur Web et un serveur de base de données;
- d'implémenter la logique métier de l'application en exploitant des langages côté serveur;
- de créer des applications interagissant de façon synchrone et asynchrone avec le serveur;
- d'utiliser un cadriciel (*framework*) pour une conception rapide et structurée d'une application Web.

Principales technologies nécessaires à la conception et au déploiement d'applications Web. Architecture générique, protocoles et langages d'une application Web. Balisage HTML, feuilles de styles et manipulation dynamique du modèle-objet (DOM). Serveur Web et serveur de base de données. Cadriciel (*framework*).

2. Plan du cours

Le cours comprend 13 séances de cours théorique et 12 séances de laboratoire. Il est donné en format accéléré à raison de deux séances de cours et deux séances de laboratoire par semaine.

Le cours adopte une approche d'enseignement orientée par la pratique: les étudiant-e-s devront mettre en pratique immédiatement les concepts enseignés. Ainsi, tout au long de la session, une application sera développée durant les séances de cours. Une application différente mais similaire sera développée par les étudiantes et les étudiants durant les séances de laboratoire.

La matière du cours, décrite ci-dessous est répartie sur à travers le cinq thèmes principaux du cours:

- Développement Web et langage HTML
- Mise en forme d'un document à travers des règles de style CSS
- Modification dynamique de documents Web à l'aide de Javascript et la manipulation du DOM
- Générer des pages web dynamiquement côté serveur
- Développement d'une application client/serveur à l'aide d'une API REST.

3. Descriptif des séances de cours

Thème 1: Introduction au développement Web et le langage HTML

Séances 1 et 2: Développement Web et les bases du HTML

- Qu'est-ce que le Web et la programmation Web?
- Les (multiples) langages du Web
- Le modèle client/serveur et le protocole HTTP
- Structurer l'information - Le langage HTML :
 - Notion de balisage
 - Structure d'une page HTML – Introduction à l'arborescence d'une page Web
- Les principales balises HTML5 (réf. : [MDN-HTML](#)) :
 - Organisation sémantique d'un document HTML / Moteurs de recherche
 - En-tête du document et balises de méta-données
 - La racine du document (body) et les balises de sectionnement
 - Balises d'organisation du contenu textuel (paragraphe, citation, ...)
 - Balises sémantiques du contenu textuel (liens, ...)
 - Contenu embarqué (médias, iframe, scripts, svg ...)
 - Contenu tabulaire
 - Les formulaires
- Validation d'un document HTML
- Notion de séparation des préoccupations - Séparation entre le contenu du document et sa forme, introduction au règles de style CSS.

Séances de laboratoire:

- Prise en main des outils de développement
- Concevoir un document HTML structuré sémantiquement à partir d'un texte.
- Validation d'un document HTML

Thème 2 : Mise en forme d'un document HTML – Les feuilles de style en cascade (CSS)

Séance 3:

- HTML vs CSS
- Approches de définition de style : en-ligne, en entête du document, fichier de style.
- Concept de règle et syntaxe de base d'une règle de style CSS
- Définition d'une règle:
 - Les Sélecteurs
 - Les éléments de style
 - Dimensions et unités
 - Parcours du site de référence ([MDN-CSS](#))
- Le positionnement (*Flexbox, Grid*)

Séance 4:

- Conception réactive (*responsive design*)
- Outils contemporains de développement CSS
 - Les limites de CSS : code verbeux, enchevêtrement des règles et maintenabilité, pas d'héritage, ...
 - La librairie de développement CSS Bootstrap
 - Concept de pré-processeur CSS et introduction à SASS/LESS

Séances de laboratoire:

- Mise en forme de l'interface client du projet de manière itérative:
 - CSS de base: Mise en forme et positionnement;
 - Interface réactive;
 - Utilisation de la librairie Bootstrap

Thème 3 : Modification dynamique du document Web: Le langage Javascript et le *Document Object Model* (DOM)

Séance 5: Le Document Object Model (DOM) et les Bases de JavaScript

Le Document Object Model (DOM)

- Introduction au DOM
 - Définition et rôle du DOM dans les applications web.
 - Structure arborescente et relation avec HTML/CSS.
- Accès et Manipulation du DOM

- Utilisation de `document.getElementById`, `document.querySelector`, etc.
- Modification des propriétés des éléments (texte, style, attributs).

Les Bases du Langage JavaScript

- Syntaxe et Fondamentaux
 - Variables, types de données, opérateurs.
 - Structures de contrôle (boucles, conditions).
- Fonctions en JavaScript
 - Déclaration et utilisation.
 - Portée des variables et fonctions fléchées.

Sélection et Modification de Contenu des Noeuds du DOM

- Sélection des Éléments
 - Méthodes pour sélectionner des éléments (par ID, classe, sélecteur CSS).
 - Parcours des éléments enfants, parents, frères.
- Modification des Éléments
 - Modification du contenu (`innerText`, `innerHTML`).
 - Manipulation des styles CSS via JavaScript.

Séance 6: Programmation Événementielle et jQuery

Les Événements et la Programmation Événementielle

- Déclenchement d'Événements
 - Types d'événements (click, load, mouseover, etc.).
 - Ajout d'écouteurs d'événements.
- Intercepter des Événements
 - Fonctionnement de la propagation des événements.
 - Prévention du comportement par défaut.
- Notion de Fonction de Rappel
 - Explication et exemples de fonctions de rappel.
 - Utilisation dans les écouteurs d'événements.

Utilisation de la Librairie jQuery pour Cibler et Modifier des Nœuds du DOM

- Introduction à jQuery
 - Avantages de l'utilisation de jQuery.
 - Syntaxe de base et inclusion dans un projet.
- Manipulation du DOM avec jQuery
 - Sélection d'éléments avec jQuery (`$`).
 - Modification du contenu et des attributs.
- Gestion des Événements avec jQuery
 - Simplification de l'ajout d'écouteurs d'événements.
 - Exemples d'événements courants avec jQuery.

Séances de laboratoire:

- Mise en forme de l'interface client du projet de manière itérative:
 - CSS de base: Mise en forme et positionnement;
 - Interface réactive;
 - Utilisation de pré-processeurs.

Séance 7: Examen Intra (3h)

Thème 4: Générer des documents Web Dynamiquement - Programmation côté serveur

Séance 8: Introduction au PHP et Modèle Client/Serveur

Introduction au Langage PHP

- **Fonctionnement du Modèle Client/Serveur**
 - Rôle du client (navigateur) et du serveur.
 - Communication via HTTP.
- **Cycle de Vie d'une Requête HTTP**
 - Envoi de la requête par le client.
 - Traitement par le serveur (PHP).
 - Réponse du serveur et affichage sur le client.
- **Syntaxe de Base**
 - Structure d'un script PHP.
 - Variables, types de données, opérateurs.

- Structures de contrôle (if, for, while, etc.).

- **Fonctions en PHP**

- Déclaration et utilisation des fonctions.
- Portée des variables.

Séance 9: Générer des pages dynamiquement avec une base de données

- **Interrogation d'une base de données**

- Utilisation de MySQLi et PDO.
- Gestion des erreurs de connexion.
- Rappels de la syntaxe de base SQL (SELECT, INSERT, UPDATE, DELETE).

- **Sessions et Cookies**

- Création et utilisation des sessions.
- Stockage et récupération des données de session.
- Création et lecture des cookies.
- Sécurité et confidentialité des cookies.

- **Traitement des Formulaires en PHP**

- Récupération des données du formulaire.
- Validation et nettoyage des données.
- Insertion/mise à jour des données dans la base de données.

Séances de laboratoire:

- Création d'une page générée dynamiquement côté serveur
- Construction du contenu initial d'une page à partir de données stockées dans une base de données
- Ajout de l'authentification à l'application

Thème 5: Conception d'une application Client-Serveur en concevant et en utilisant une API REST

Séance 10: Introduction aux API REST et Principes de Base

1. Introduction aux API REST

- Définition et importance des API dans le développement web.
- Différence entre API REST et autres types d'API (SOAP, GraphQL, etc.).

2. Principes de base des API REST

- Architecture sans état (Stateless).
- Utilisation des méthodes HTTP (GET, POST, PUT, DELETE).
- Structure des URL et des ressources.
- Formats de réponse (JSON, XML).

3. Conception d'une API REST simple

- Planification des ressources (ex: utilisateurs, articles, commentaires).
- Définition des routes et méthodes.
- Bonnes pratiques (naming conventions, versioning).

4. Introduction à PHP pour les API REST

- Rappel rapide sur PHP (syntaxe de base, connexion à une base de données).
- Création d'un petit serveur local en PHP.
- Exemple de code pour une route GET simple.

Séance de laboratoire

- Création d'un schéma de base pour une API (choix des ressources, définition des routes).
- Développement d'une route GET
- Test de l'API avec un logiciel d'interaction avec une API comme *Postman*.

Séance 11: Développement d'une API REST avec PHP

1. Routage

- Concept de routage
- Gestionnaires de code (*handlers*)
- Méthodes HTTP et actions correspondantes
- Paramètres statiques (`/users`) et dynamiques (`/users/:id`)

2. Développement d'une API en PHP

- Gestion des requêtes HTTP en PHP.
- Introduction au format JSON
- Récupération et envoi de données en format JSON.

3. Communication avec l'API REST

- l'objet Javascript *XMLHttpRequest*
- Requêtes asynchrones avec `Axios`

4. Sécurité de base pour une API REST

- Authentification (API keys).
- Validation et nettoyage des données entrantes (sanitization, validation).

Séance de laboratoire:

- Développement de plusieurs routes pour l'API (GET, POST, PUT, DELETE).
- Test de l'API avec un logiciel d'interaction avec une API comme *Postman*.

Séance 12: Utilisation de l'API avec JavaScript Vanilla

1. Communication avec l'API REST

- Envoi de requêtes à l'API (GET, POST, etc.).
- Gestion des réponses et des erreurs.
- Mise à jour dynamique du contenu HTML avec les données de l'API.

2. Construction d'une petite application web

- Création d'une interface utilisateur simple (HTML/CSS).
- Interaction avec l'API pour afficher, ajouter, modifier et supprimer des données.

3. Travaux pratiques

- Développement d'une application web simple qui utilise l'API créée.
- Test et débogage de l'application.

4. Conclusion et meilleures pratiques

- Revue des points clés.
- Discussion sur les meilleures pratiques en développement d'API et en JavaScript.

Séances de laboratoire:

- Réalisation de l'application

Séance 13: Compléments de cours et révision

3. Évaluations

Évaluation	Pondération	Séance
Examen Intra	20%	7
Laboratoires notés	40%	Toutes les séances de laboratoire.
Examen final	40%	Après la séance 13

Ce cours adopte le double seuil: L'obtention d'une note pondérée de 50% à l'ensemble des évaluations à caractère individuel est une condition nécessaire (mais pas suffisante) pour la réussite du cours.

Notes concernant les évaluations:

- L'examen intra et l'examen final sont réalisés sur machine dans l'environnement ENAQUIZ (MOODIE) en utilisant SafeExamBrowser. Il est de la responsabilité des personnes étudiantes de s'assurer du bon fonctionnement de leur machine avant de se présenter à l'examen.
- Les laboratoires notés sont réalisés par paires, selon la méthode du "*pair programming*". Les laboratoires visent à mettre en application la matière enseignée en cours à travers la réalisation d'un projet de session qui évolue semaine par semaine selon la matière présentée en cours. Il y aura cinq évaluations d'avancement, à la fin de chacun des thèmes du cours.
- L'examen final est un examen cumulatif où toute la matière du cours sera évaluée. Cependant, une plus grande prépondérance sera accordée à la matière vue à partir de la séance 8 du cours (après l'Intra).

Références

Références obligatoires:

Aucune

Références optionnelles (en construction):

- [MDN Web Docs](#)
- [DevDocs](#)
- [CSS Reference](#)
- [You don't know JS Yet](#)