

Fondamentaux du Web HTML, CSS, JavaScript

Plan de Cours

Projet fil rouge : Jàng Afrig

Licence 1 DUT1-Informatique — Semestre 2
2025–2026

Dr. El Hadji Bassirou TOURÉ
École Supérieure Polytechnique
Département Génie Informatique
Université Cheikh Anta Diop de Dakar

Informations pratiques

Volume horaire	15 séances de 2h = 30h
Public	Licence 1 DUT1-Informatique — DGI / ESP / UCAD
Semestre	Semestre 2 (2 ^e semestre de L1)
Prérequis	Algorithmique du S1. Savoir utiliser un ordinateur.
Outils	VS Code, Live Server, Git, GitHub, Chrome ou Firefox
Projet	Jàng Afrig — quiz géographique sur les 54 pays africains
Évaluation	Participation + avancement Git (voir détail p. 7)

Ce que vous saurez faire à la fin du cours

1. Écrire une page web complète en HTML5 (structure, sémantique, formulaires, tableaux)
2. Mettre en forme une page avec CSS3 (sélecteurs, box model, Flexbox, responsive mobile)
3. Programmer en JavaScript moderne (variables, fonctions, tableaux, objets, `forEach`, `map`, `filter`)
4. Modifier le contenu d'une page depuis JavaScript (DOM et événements)
5. Récupérer des données d'une API distante avec `fetch` et `async/await`
6. Sauvegarder des données dans le navigateur (`localStorage`)
7. Versionner votre code avec Git et le partager sur GitHub
8. Déployer une application web en ligne sur GitHub Pages

Table des matières

1	Présentation du cours	3
1.1	Deux types de séances : CM et Lab	3
1.2	Déroulement d'une séance	3

2	Le projet fil rouge : Jàng Afrig	4
2.1	Les modes de jeu	4
2.2	Comment on le construit	4
2.3	Les pages de l'application	4
3	Outils à installer	5
3.1	Git : la routine de fin de Lab	5
4	Plan des 15 séances	6
5	Évaluation	7
5.1	Philosophie	7
5.2	Critères d'évaluation	7
6	Ressources	8
6.1	Pendant le cours	8
6.2	Pour aller plus loin (optionnel)	8
6.3	Cours connexes du cursus	8

1 Présentation du cours

Ce cours est votre **premier contact avec un vrai langage de programmation qui s'exécute dans un outil réel**. Vous avez vu l'algorithmique en pseudo-code au S1 ; vous allez maintenant écrire du code qui *marche*, dans un navigateur, et qui produit des résultats visibles.

Au terme du semestre, vous aurez construit **Jàng Afrig**, un quiz géographique sur les 54 pays africains, accessible depuis n'importe quel téléphone via une URL publique que vous pourrez partager à votre famille.

Pourquoi 100% vanilla ?

Vous entendrez parler de frameworks (React, Vue, Next.js...). Ce sont des outils très puissants, mais ils **cachent** ce qui se passe vraiment dans le navigateur. Avant d'utiliser un framework, il faut comprendre ce qu'il fait à votre place. Ce cours se concentre donc sur les briques standard du Web — pas de framework, pas de **npm**. Les frameworks viendront en L2 (cours PRC) et L3 (cours IPDL1).

1.1 Deux types de séances : CM et Lab

Le cours alterne deux types de séances, qui se complètent :

- **CM (Cours Magistral)**. 4 séances au total. On introduit les concepts avec des analogies, des exemples visuels, une démo en direct, et un mini-exercice en classe. Aucun devoir à la maison après un CM.
- **Lab (TP guidé)**. 10 séances au total. On met les mains dans le code, on construit une partie de Jàng Afrig, on fait des exercices d'application, et on termine par la routine Git.

La dernière séance (S15) est un **hybride** : première heure sur le déploiement en ligne, deuxième heure démo collective.

1.2 Déroulement d'une séance

Moment	CM (2h)	Lab (2h)
5 min	Rappel séance précédente	Rappel séance précédente
10 min	Analogie + question centrale	Démo de l'objectif du jour
60 min	Contenu théorique en 3-4 blocs	Vous codez en suivant le pas-à-pas
30 min	Mini-exercice en classe	Exercices d'application + avancement Jàng Afrig
15 min	Résumé + teaser du Lab suivant	Routine Git + annonce S suivante

2 Le projet fil rouge : Jàng Afrig

Jàng signifie « *apprendre, étudier* » en wolof, et **Afrig** désigne l’Afrique. Le nom évoque donc littéralement « *apprendre l’Afrique* » — ce que fait l’étudiant en jouant au quiz tout en construisant l’application. Jàng Afrig est une application web de **quiz géographique** sur les 54 pays africains. L’utilisateur choisit un mode de jeu, répond à 10 questions à choix multiples, obtient un score, et garde un classement local de ses meilleures performances.

2.1 Les modes de jeu

Mode	Question type
Capitales	« Quelle est la capitale du Mali ? »
Drapeaux	« À quel pays appartient ce drapeau ? »
Langues	« Dans quel pays parle-t-on le swahili ? »
Voisins	« Combien de voisins a l’Algérie ? »
Chrono	Maximum de bonnes réponses en 60 secondes

2.2 Comment on le construit

Jàng Afrig n’est pas un projet à rendre. C’est l’application qu’on construit **ensemble**, séance après séance, en Lab. À chaque Lab, l’application gagne une fonctionnalité visible. Au bout de 15 séances, c’est une vraie application en ligne.

Les données

L’application utilisera l’API publique [REST Countries v3.1](#), qui fournit gratuitement des informations sur tous les pays du monde : nom, capitale, drapeau, langues, frontières, superficie. Aucune inscription, aucune clé API, aucune limite.

Un seul appel au démarrage, puis toutes les questions sont générées localement à partir du tableau des 54 pays africains.

2.3 Les pages de l’application

1. **Accueil** : choix du mode + saisie du pseudo + meilleur score
2. **Quiz** : énoncé + 4 propositions + barre de progression
3. **Résultat** : score final + bouton rejouer
4. **Mes Scores** : classement local par mode (top 5)
5. **À propos** : sources des données, technologies

3 Outils à installer

Avant la première séance, vous devez avoir installé tous les outils et fait votre premier cycle Git. Une fiche dédiée `Lab0_Setup.pdf` vous guide pas à pas, avec captures d'écran. **Faites-la à la maison la veille du cours.**

Outil	Usage	Lien
VS Code	Éditeur de code	code.visualstudio.com
Live Server	Extension VS Code (1 clic)	Marketplace VS Code
Git	Suivre les modifications du code	git-scm.com
Compte GitHub	Hébergement du code en ligne	github.com
Chrome ou Firefox	Navigateur + outils de développement	Préinstallé

Aucun npm

Le cours est volontairement vanilla. **Aucune installation Node.js**, aucun `package.json`, aucun `npm install`.

3.1 Git : la routine de fin de Lab

À la fin de chaque Lab, vous exécuterez la même séquence rituelle :

```
git add .
git commit -m "v0.X : titre du Lab"
git push
git tag v0.X
git push --tags
```

Cette routine devient automatique au bout de 3-4 Labs. Les commandes Git elles-mêmes sont toutes enseignées dans le Lab 0 — **aucun prérequis** n'est demandé.

4 Plan des 15 séances

N°	Type	Bloc	Titre	Tag Git
Lab 0	Lab	Setup	Mise en route (à la maison)	v0
S1	CM1	HTML	Le Web et HTML	—
S2	Lab 1	HTML	Ma première page HTML	v0.1
S3	Lab 2	HTML	Sémantique, formulaires, tableaux	v0.2
S4	CM2	CSS	Sélecteurs, cascade, box model, Flexbox	—
S5	Lab 3	CSS	CSS en pratique : styliser Jàng Afrig	v0.3
S6	Lab 4	CSS	Layouts + responsive mobile	v0.4
S7	CM3	JS	JavaScript : le langage qui anime les pages	—
S8	Lab 5	JS	JS pratique : variables, fonctions, conditions	v0.5
S9	Lab 6	JS	Tableaux et objets (<code>forEach</code> , <code>map</code> , <code>filter</code>)	v0.6
S10	Lab 7	JS	DOM et événements : premier quiz fonctionnel	v0.7
S11	CM4	API	Le Web moderne : APIs, JSON, asynchrone	—
S12	Lab 8	API	Fetch + <code>async/await</code> : brancher REST Countries	v0.8
S13	Lab 9	API	Tous les modes de jeu avec l'API	v0.9
S14	Lab 10	JS	<code>localStorage</code> , chrono, polish	v0.14
S15	Hybride	Final	Déploiement GitHub Pages + démo collective	v1.0

Le fil narratif

Chaque bloc est construit autour d'un CM qui pose les concepts, suivi de Labs qui les appliquent sur Jàng Afrig. À chaque fin de Lab, un tag Git v0.X marque la progression. À la fin, le tag v1.0 marque la version publiée en ligne.

À chaque CM

Pour chaque séance, un document PDF dédié vous est distribué :

- **CMX_[titre].pdf** pour les cours magistraux
- **LabX_[titre].pdf** pour les TP guidés

Le prof projette le PDF en direct, vous le suivez sur votre écran ou en papier. Tout est écrit, rien n'est à noter à la volée.

5 Évaluation

5.1 Philosophie

Ce cours **ne comporte pas de projet noté final** à rendre. Le projet Jàng Afrig n'est pas un livrable séparé : c'est l'application qu'on construit ensemble en Labs, séance après séance. L'évaluation porte sur la **participation continue** et la **progression visible**.

5.2 Critères d'évaluation

Critère	Comment c'est observé	Poids
Présence et participation active	Présence aux séances + interactions en Labs	20%
Avancement Git	Tags posés de <code>v0.1</code> à <code>v1.0</code>	30%
Qualité du code	Code lisible, indenté, commenté	20%
Fonctionnalité de l'app	L'app finale fonctionne sur le téléphone du jury	20%
Déploiement réussi	URL GitHub Pages accessible en S15	10%

Le seul vrai test

À la fin du cours, vous devez être capable de :

1. Ouvrir un terminal et faire un commit Git
2. Écrire une page HTML sémantique
3. Styliser une page avec Flexbox responsive
4. Manipuler le DOM en JavaScript
5. Manipuler un tableau avec `forEach`, `map`, `filter`
6. Récupérer des données d'une API avec `async/await`
7. Persister des données avec `localStorage`
8. Déployer un site statique sur GitHub Pages

Si oui, le cours a réussi.

6 Ressources

6.1 Pendant le cours

- **MDN Web Docs** — la documentation de référence.
<https://developer.mozilla.org/fr>
- **REST Countries v3.1** — l'API du projet.
<https://restcountries.com>

6.2 Pour aller plus loin (optionnel)

- Marijn Haverbeke, *Eloquent JavaScript* (3^e éd.), gratuit en ligne : <https://eloquentjavascript.net>
- Stanford CS193X — *Web Programming Fundamentals* (cours d'inspiration) : <https://web.stanford.edu/class/archive/cs/cs193x/cs193x.1176>

6.3 Cours connexes du cursus

- **L2 S4 — Programmation par Réutilisation de Composants (PRC)**. Suite naturelle : React, Flask, fil rouge SénTransport.
- **L2 S4 — Intégration de Modèles IA (IMA)**. APIs, Docker, fil rouge SénSanté.
- **L3 — IPDL1**. Ingénierie des processus, Next.js, équipes de 6.

Bon semestre à toutes et tous ✓

Le code, ça s'apprend en codant.