



JUMPSPEAK.COM

Learn a New Language By Actually Speaking It

Sign up

SCI6373 Programmation documentaire

Cours 1

Été 2025

Apprendre JS en 135 heures

- Ça ne se fait pas !
- C'est complètement inutile !
- L'IA, est LA solution ultime...
 - Comme partout, mais surtout en programmation !
- *Pas d'accord...!* → *Bibliographie*
 - Travis, Popper, Bergstrom, Broussard, Endler

A close-up, low-angle shot of the ocean surface from underwater. The water is a deep teal color. Numerous small, white, spherical bubbles are scattered across the frame, some rising towards the surface and others falling. The lighting creates a soft glow on the bubbles and the water's surface.

45 heures en classe
+ 90 heures de travail personnel !

IMMERSION

Immersion, mais...

Avec parcimonie !
Principe de *PYM*

Au programme aujourd’hui (1/2)

- Logistique
- Présentations
- Objectifs / intérêts
- Qu'est-ce que la programmation ?
- Pourquoi apprendre à programmer ?
 - Programmation et société
- *End-user programming*

Au programme aujourd’hui (2/2)

- Programmation et écriture, esthétique
- Ce cours et ses choix
 - Approche et langage
 - Les trois environnements touchés
 - Prérequis
 - Pourquoi JavaScript ?
- Les ressources du cours
- Votre premier script !

Logistique !

- Tout se passe pas mal sur StudiUM
- Pause
- Disponibilité presque continue
- Les quiz se feront (probablement) sur StudiUM
- Utilisation de l'IA interdite pour les travaux évalués
 - Nie l'objectif d'apprentissage

Présentations

Vos objectifs / intérêts ?

- Pour le cours lui-même
- Intérêts pour travail d'exploration
 - API (quelques idées [ici](#))
 - Autres langages
 - Matériel programmable des labos de l'EBSI

Vous avez dit programmation ?

- Paramétrage ↔ Programmation
 - Continuum
- Existe pour de nombreux types d'outils
 - Paramétrage ↔ Fabrication
- Avec les outils informatiques
 - Point de bascule : syndrome de la page blanche
 - *Configuration as Code*

Pourquoi apprendre à programmer ?

Programmation et société

Parce que... (1/3)

- Programmation: excellent pour développer une compréhension profonde des technologies
 - Vous serez un interlocuteur plus efficace avec les équipes de développement ou les TI
 - Excellent pour développer une approche logique à la résolution de problèmes

Parce que... (2/3)

- Compétences transférables à d'autres activités de nature "technologique"
 - Personnalisation / paramétrage avancés
 - Création / paramétrage de bases de données
 - Fonctions avancées dans différents systèmes
 - Stylage avancé de documents
 - Tableurs
 - Validation de formulaires
 - etc.

Parce que... (3/3)

- Autonomie dans sa vie numérique
 - Professionnelle et personnelle
- Vous pourriez même vous retrouver à programmer vous-même dans votre travail
 - Rare, mais ça arrive !
- Mais surtout... *C'est très satisfaisant !*
- Et... *Très amusant !*

Aussi parce que...

- Paradoxalement, c'est dans l'air du temps!
 - Se démocratise avec le Web
 - Nombre impressionnant de ressources pour apprendre la programmation aux *enfants* ! → Bibliographie du cours
 - *Coding education should begin early and become a standard part of school curriculum, despite advances in AI that make software development more accessible than ever before*
- GitHub CEO T. Dohmke [The Times of India](#) 2025-04-29

Aussi parce que...

- "(...) emerging chasm between people who code and people who do not"
– Tim Berners-Lee, Washington Post 2013-03-10

End-user programming

- *End-user programming, development ou computing*
 - Concept identifié dans les années 1980
 - "Démocratisation" du développement informatique
 - Exemples: feuilles de calcul, formulaires en traitement de texte, formulaires en PDF, etc.
 - Encouragé par le web et les langages de scriptage, car on peut **voir** et **copier** le code
 - *Comme quand on apprend une langue !*

Points de vue sur la programmation

- Programmation et écriture
 - Edsger W. Dijkstra (1975)

“Besides a mathematical inclination, an exceptionally good mastery of one’s native tongue is the most vital asset of a competent programmer.”
 - Literate Programming (Donald Knuth, 1984)
 - Writing in software development (Stavely, 2011)
- Programmation et esthétique
 - Vikram Chandra (2014)
 - The Beauty of Code, the Code of Beauty

Le cours

- Programmation procédurale (ou *impérative*)...
- Orientée-objet...
- Avec un langage de scriptage...
- *Programmation documentaire* ...

Programmation procédurale

- Le traitement à effectuer est exprimé comme une suite d'énoncés *impératifs* ou instructions indiquant à la machine quoi faire et dans quel ordre (recette)
- Les énoncés sont groupés en *procédures*
- Les procédures peuvent s'appeler l'une l'autre (recette vs "sous-recette")
- Les procédures forment un *programme*

Orientation-objets (OO)

- Les "instructions" travaillent sur (créent, modifient, etc.) des "objets" qui existent dans un *environnement d'exécution*
- Un objet *encapsule* données **et** instructions
- Certains objets sont **prédéfinis** dans le *langage de programmation*
- D'autres sont créés dynamiquement par le programme lui-même

Langage de programmation OO

- Un langage de programmation OO comprend:
 - Règles syntaxiques
 - Sémantique
 - Objets **prédéfinis**
- Langage + environnement spécifiques déterminent une *plateforme* de programmation (ou développement)

Langages de scriptage

- Exemples :
 - JavaScript, PHP, Perl, Python, Ruby
- Scriptage versus quoi?
 - Langages « statiques », traditionnels, compilés, etc.
- Voisins des langages de « macros »
- Dynamiques (moins contraignants)
- **Code visible jusqu'à l'utilisateur final**

Programmation documentaire

- Il s'agit de documents *dynamiques*
- Qui peuvent se transformer et / ou interagir avec leur "lectrice" (utilisatrice)
- Parce qu'ils contiennent du *code exécutable*, c.-à-d. des programmes (*scripts* ou *macros*)
- Ce sont des documents *programmables*
 - D'où "programmation documentaire"

Dans le cours

- Langage: JavaScript
 - Sauf peut-être dans le travail d'exploration
- Trois environnements vus (ou entrevus) :
 - Navigateur Web (JavaScript côté client)
 - HTML + JavaScript côté serveur
 - Système d'exploitation (OS) : Windows
- Accent sur l'environnement navigateur

Prérequis

- *SCI6005 Information numérique et informatique documentaire*
 - compréhension et maniement de base : fichiers, fichiers texte, Web
 - Jeux de caractères (Unicode)
 - HTML, CSS
- Le texte *Introduction au cours* pointe à des ressources utiles comme rappels

Pourquoi JavaScript? (1/3)

- “*JavaScript is the world's most popular programming language. JavaScript is the programming language of the Web. JavaScript is easy to learn.*”
[Tutoriel JavaScript W3Schools](#)
- Le JavaScript dans les navigateurs modernes est extrêmement performant
→ Diverses lectures en bibliographie

Pourquoi JavaScript? (2/3)

- Compétences transférables:
 - Bureautique (Excel, Word, PDF, etc.)
 - Bases de données
 - Autres langages (style de syntaxe répandu)

Pourquoi JavaScript? (3/3)

- Le *End-user programming* est applicable !
 - En particulier, personnaliser des scripts déjà faits est envisageable
- JavaScript possède une multitude d'API (Application Program Interface), de "frameworks" (JQuery, AngularJS), de bibliothèques de fonctions, etc.
 - Qui découpent ses domaines d'utilisation

Mais...

- JS est un des rares langages plus faciles à écrire qu'à lire !
 - Plusieurs façons de tout faire
- Je préconise et applique le principe de Pierre Yves McSween (PYM)...
 - En as-tu vraiment besoin ?
 - Faites-moi confiance !

Les ressources du cours

[Site du cours](#)

Votre premier script !

- Exercice en classe
 - Console JavaScript des navigateurs
 - Maniement de base
 - Autres formes de JS en navigateur
 - Pseudo-protocole javascript :
 - En ligne adresse / En "bookmarklet"
 - Votre premier script intégré
 - Point de départ :
 - [Infrastructure HTML vide \(exemple 005\)](#)

Où on s'en va ?

- À lire :
 - Intro au cours
 - Intro aux expressions JS
 - Jusqu'à Section 8 inclusivement
- Explorer à la console
- Essayer d'aller plus loin avec votre premier script
- "Exercices après C1" (lien dans StudiUM)