

# Licence 3

Sciences, Technologies, Santé

2023-2024

*Informatique*

# informatique



L3 info.



DIPLÔME  
NATIONAL DE  
LICENCE  
CONTRÔLÉ  
PAR L'ÉTAT

# SOMMAIRE

Contacts de la formation	03
Calendrier	04
Présentation de la formation	05
Volumes horaires et évaluations	06
<b>Contenu des enseignements</b>	08

*Sommaire interactif  
pour revenir  
au sommaire  
cliquer sur ►►*



## CONTACTS DE LA FORMATION

— Sandrine TRAVIER : **Directrice Adjointe à la Pédagogie**  
*sandrine.travier@univ-angers.fr*

— David GENEST : **Directeur des études portail MPCIE**  
*david.genest@univ-angers.fr*

— Igor STEPHAN : **Responsable pédagogique et Président du Jury**  
Bureau H106  
Tél. : 02 41 73 54 66  
*igor.stephan@univ-angers.fr*

— **Gestion de la scolarité et des examens**  
Tél. : 02 41 73 53 99

### SCOLARITÉ – EXAMENS

Bâtiment A, Rez-de-chaussée

Horaires d'ouverture

8h30 – 12h00

13h30 – 16h30




Du lundi au vendredi

Fermé le mercredi après-midi



# CALENDRIER

## Semaine

36	Début des cours	04 septembre 2023
36-41	Période 11	Du 04 septembre 2023 au 13 octobre 2023
41	Période 11 - Contrôle continu	
42-43	Période 12	Du 16 octobre 2023 au 27 octobre 2023
44	<i>Vacances d'automne</i> 	<i>Du 28 octobre 2023 au 05 novembre 2023</i>
45-48	Période 12	Du 06 novembre 2023 au 01 décembre 2023
48	Période 12 – Contrôle continu	
49-51	Période 13	Du 04 décembre 2023 au 22 décembre 2023
52-1	<i>Vacances de fin d'année</i> 	<i>Du 23 décembre 2023 au 07 janvier 2024</i>
2-4	Période 13	Du 08 janvier 2024 au 19 janvier 2024
3	Inscription E1	Du 17 janvier 2024 au 26 janvier 2024
4	Période 13 – Contrôle continu	
5	E1	Du 29 janvier 2024 au 02 février 2024
6-9	Période 14	Du 05 février 2024 au 01 mars 2024
10	<i>Vacances d'hiver</i> 	<i>Du 02 mars 2024 au 10 mars 2024</i>
11-12	Période 14	Du 11 mars 2024 au 22 mars 2024
12	Période 14 – Contrôle continu	
13-22	Période 15 - Stage	Du 25 mars 2024 au 30 mai 2024
21-22	Période 15 - Soutenances	
23	Inscription E2	Du 05 juin 2024 au 07 juin 2024
25	E2	Du 17 juin 2024 au 21 juin 2024
28	Jury	10 juillet 2024

**Sous réserve de modifications**



## PRÉSENTATION DE LA FORMATION

### OBJECTIF DE LA FORMATION

Les enseignements de la 3<sup>ème</sup> année de la licence informatique permettent d'acquérir les concepts fondamentaux dans les principaux domaines de l'informatique : architecture, algorithmique, modélisation, bases de données. Tous ces enseignements comportent un volet théorique et un volet pratique. Les étudiants doivent accomplir un important travail de réalisation lors des TP et des projets tuteurés afin d'acquérir de bonnes compétences pour des développements informatiques professionnels.

La licence prépare principalement les étudiants à une poursuite d'étude en MASTER, afin de continuer vers la recherche (doctorat) ou d'occuper un poste d'ingénieur dans une entreprise ou une organisation.

Elle n'exclut pas pour certains étudiants une sortie vers le monde professionnel dès Bac + 3. La licence d'informatique peut permettre d'intégrer une entreprise en tant qu'ingénieur d'études, analyste programmeur, développeur web, ... ou peut permettre d'occuper des emplois pour lesquels les compétences informatiques représentent un atout : administration, enseignement...

### OPTIONS ET STAGES

La période 14, propose 2 enseignements à choisir parmi une liste d'options, qui permettent la découverte de domaines ou d'applications plus spécifiques (synthèse d'images, introduction au machine learning avec Python, développement d'interfaces graphiques, initiation aux systèmes intelligents).

La période 15 comporte une unité de stage obligatoire (d'une durée de deux mois). Elle se déroule au département d'informatique ou en entreprise. Chaque étudiant ou binôme est encadré par un enseignant du département : un rapport est rédigé et une soutenance publique devant un jury est organisée.

# VOLUMES HORAIRES – ÉVALUATIONS

## Licence 3 Informatique

Période	Intitulés	Volumes horaires				ECTS	Coef	Chance	
		CM	TD	TP	Total			Chance 1 <i>Assidus</i>	Chance 2 <i>Assidus</i>
<b>ANGLAIS</b>									
<b>Anglais 1</b>									
P11	Anglais			8,0	8,0	2	2	CC 100% -1h20	CT 100% - 1h20
P12	Anglais			8,0	8,0				
<b>Anglais 2</b>									
P13	Anglais			8,0	8,0	2	2	CC 100% -1h20	CT 100% - 1h20
P14	Anglais			8,0	8,0				
<b>3PE</b>									
	3PE			20,0	20,0	0	0	Suivi individuel. Pas d'évaluation.	
<b>STAGE choix 1 parmi 2</b>									
P15	Entreprise		1,0			6	6	note	Pas de 2 <sup>nd</sup> chance
P15	Laboratoire		2,0					note	
<b>CHOIX DE 2 UE parmi les 4 suivantes</b>									
<b>Note plancher 1</b>									
P14	Synthèse d'images	6,4		17,6	24,0	2	2	TP 50% - 1h Projet 50%	TP 50% - 1h20 Projet 50% ●
P14	Initiation aux systèmes intelligents	6,9	3,4	13,7	24,0	2	2	CT 33% - 2h Projet 67%	CT 33% - 2h Projet 67% ●
P14	Développement d'interface graphique	4,0		20,0	24,0	2	2	TP 100% - 2h	TP 100% - 2h
P14	Introduction au machine learning avec Python	7,0	4,6	12,4	24,0	2	2	100% Projet	CT 67% - 1h30 Projet 33% ●
<b>Fondements</b>									
<b>Note plancher 5</b>									
P11	Fondements de l'informatique 4	10,0	10,0		20,0	3	3	CT 100% - 2h	CT 100% - 2h
<b>Théorie des langages 2 et compilation</b>									
P12	Théorie des langages 2 et compilation	10,0	8,7	6,0	24,0	6	6	CC 33% - 2h	CT 100% - 2h
P13	Théorie des langages 2 et compilation	10,0	8,7	6,0	24,0			CC 67% - 2h	
<b>Architecture et ordinateurs</b>									
P13	Architecture des ordinateurs	10,0	5,8		15,8	6	6	CC 33% - 2h	CT 100% - 1h
P14	Architecture des ordinateurs	10,0	5,8	4,2	20,0			CC 67% - 2h	
<b>Algorithmique et programmation</b>									
<b>Note plancher 5</b>									
<b>Programmation logique</b>									
P13	Programmation logique	12,0		12,0	24,0	3	3	CT 67% - 1h Projet 33%	CT 67% - 1h Projet 33% ●
<b>Programmation fonctionnelle</b>									
P11	Programmation fonctionnelle	10,0	6,0	8,0	24,0	3	3	CT 67% - 1h Projet 33%	CT 100% - 1h ●
<b>Algorithmique des graphes</b>									
P14	Algorithmique des graphes	12,0	4,0	8,0	24,0	3	3	CT 67% - 2h Projet 33%	CT 67% - 2h Projet 33% ●
<b>Programmation orientée objet 2</b>									
P12	Programmation orientée objet 2	10,0		22,0	32,0	7	7	TP 40% - 2h	CT 100% - 2h
P13	Programmation orientée objet 2	10,0		22,0	32,0			TP 60% - 2h	
<b>Technologie de l'informatique</b>									
<b>Note plancher 5</b>									
<b>Traitements de données</b>									
P11	Traitement de données	6,7		13,3	20,0	3	3	CT 67% - 2h Projet 33%	CT 67% - 2h Projet 33% ●
<b>Bases de données 3</b>									
P11	Bases de données 3	4,0		12,0	16,0	6	6	TP 40% - 1h30	CT 100% - 1h30
P12	Bases de données 3	8,0	12,0	4,0	24,0			CC 60% - 1h30	
<b>Développement web 3</b>									
P11	Développement web 3	5,3		10,7	16,0	6	6	TP 40% - 1h30	TP 100% - 1h30
P12	Développement web 3	8,0		16,0	24,0			TP 60% - 1h30	
<b>499,9</b>						<b>60</b>	<b>59,9</b>		

- La note de projet est conservée
- La note de projet n'est pas conservée



 Conditions de validation du bloc

➤ Moyenne du bloc supérieure ou égale à 10

 Conditions de validation de l'année

➤ Moyenne générale supérieure ou égale à 10 ET Moyenne de bloc supérieure ou égale à la note plancher pour les blocs ayant une note plancher ET Note d'UE supérieure ou égale à la note plancher pour les UE ayant une note plancher (options). Acquisition des ECTS par UE (note à l'UE  $\geq 10$ ) ou par bloc (si note de bloc  $\geq 10$ , les unités composant le bloc sont acquises) si 60 ECTS validés (moyenne générale  $\geq 10$  et pas de note de UE  $<$  note plancher)



## CONTENUS DES ENSEIGNEMENTS

PI1

PI2

PI3

PI4

### ANGLAIS

Responsable [Romain Laudier](#)

#### • PRÉ-REQUIS

##### **Notions et contenus**

Les bases de la langue anglaise

##### **Compétences**

Dans l'idéal, maîtriser le niveau B1 du CE-CRL (dit « d'utilisateur indépendant »)

#### • CONTENUS DE L'UE

##### **Objectifs pédagogiques**

— Permettre aux étudiants de continuer à travailler les cinq compétences en langue (Compréhension écrite et orale, expression écrite et orale, et interaction orale) à travers des supports authentiques (articles, documentaires, documents audio et vidéo d'internet, graphiques...) et des activités variées (exercices de compréhension, d'expression écrite, jeux de rôle, débats, présentations orales...).

— Étoffer les connaissances lexicales

— Améliorer la prononciation (bases de phonologie)

—Revoir et comprendre des points de langue (les temps, les modaux par exemple)

##### **Compétences**

En fin de licence, on vise le niveau de compétence B2 du Cadre Européen Commun de Référence pour les Langues (CE-CRL) qui est résumé comme suit : « Peut comprendre le contenu essentiel de sujets concrets ou abstraits dans un texte complexe, y compris une discussion technique dans sa spécialité. Peut communiquer avec un degré de spontanéité et d'aisance tel qu'une conversation avec un locuteur natif ne comporte de tension ni pour l'un ni pour l'autre. Peut s'exprimer

de façon claire et détaillée sur une grande gamme de sujets, émettre un avis sur un sujet d'actualité et exposer les avantages et les inconvénients de différentes possibilités.»





## Choix de 2 UE parmi les 4

01

P14

### SYNTHÈSE D'IMAGES

Responsable Igor Stephan

#### • PRÉ-REQUIS

#### **Notions et contenus**

Algorithmique 1 à 5

#### **Compétences**

Maîtrise des concepts basiques de la programmation en C.

#### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

— Éléments de mathématique propre au fonctionnement d'une carte graphique.

— Présentation des transformations en OpenGL qui sous tendent la création d'objets composés et leur mise en mouvement.

— Mise en œuvre de l'ensemble des notions liées à la caméra, aux lumières, aux couleurs, au texturage, au blending et à l'animation 2D.

#### **Compétences**

Maîtrise des outils mathématiques élémentaires régissant le fonctionnement d'une carte graphique.

Maîtrise de la librairie logicielle OpenGL en C permettant de réaliser des images de synthèse 3D en temps réel.

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 24032

02

P14

### INITIATION AUX SYSTÈMES INTELLIGENTS

Responsable Caroline Devred

#### • PRÉ-REQUIS

#### **Notions et contenus**

Programmation objet en C++  
Programmation prolog.

#### **Compétences**

Savoir programmer.

Savoir enrichir un programme objet existant.

#### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

L'objectif de ce cours est, au travers d'un challenge de programmation de systèmes intelligents livrant des parties d'un jeu donné, de découvrir les algorithmes Min-Max, Alpha-Beta et Monte-Carlo.

#### **Compétences**

Cet enseignement permettra de consolider les compétences en programmation. D'apprendre et de comprendre les algorithmes Min-Max, Alpha-Beta, Monte-Carlo. ET de découvrir GGP et GDL.

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 19317

<http://gpp.stanford.edu/gamemaster/homepage/gametester.php>



## DÉVELOPPEMENT D'INTERFACES GRAPHIQUES

Responsable [Marc Legeay](#)

### - PRÉ-REQUIS

#### **Notions et contenus**

Programmation Orientée Objet en C++

#### **Compétences**

- Maîtriser les concepts de la programmation orientée objet (notamment l'héritage et le polymorphisme)
- Maîtriser les outils de compilation

### - CONTENUS DE L'UE

#### **Objectifs pédagogiques**

Présentation du fonctionnement des interfaces graphiques en générale, et plus particulièrement du fonctionnement de la bibliothèque Qt. Vous serez capable à la fin de cette unité de produire des interfaces graphiques simples et évoluées pour vos applications. L'UE se focalise sur le développement d'interfaces graphiques pour application bureau, mais une courte introduction au développement d'interfaces mobiles via Qt sera également donnée en fin d'UE.

#### **Compétences**

- Connaître les éléments de bases d'une interface graphique
- Maîtrise du système Qt de signaux et slots
- Savoir gérer les événements (clavier, souris, ...)
- Pouvoir définir un élément graphique personnalisé
- Maîtrise du système Qt des layouts

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 19315

## INTRODUCTION AU MACHINE LEARNING AVEC PYTHON

Responsable [Olivier Goudet](#)

### - CONTENUS DE L'UE

#### **Objectifs pédagogiques**

Ce cours a pour but d'introduire des outils de traitement de données en python.

- Une première partie portera sur les éléments de base de la syntaxe du langage python ainsi que la manipulation de grand jeux de données avec les bibliothèques Numpy et Pandas.
- Une deuxième partie portera sur la visualisation des données avec Matplotlib et Seaborn.
- Enfin une dernière partie sera une introduction au Machine Learning avec la présentation d'algorithmes d'apprentissage supervisé (k plus proches voisins, perceptron).

# FONDEMENTS

P11

## FONDEMENTS DE L'INFORMATIQUE 4

Responsables [Frédéric Saubion](#), [Eric Monfroy](#)

### • PRÉ-REQUIS

#### **Notions et contenus**

Fondements 1,2,3  
Mathématiques discrètes  
Ordres, ensembles, fonctions, logique propositionnelle, logique du premier ordre

#### **Compétences**

— Comprendre, Analyser, Modéliser, Formaliser,  
— Raisonner

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

Mettre en avant deux aspects complémentaires et fondamentaux de la modélisation, de la mise en œuvre du calcul au premier ordre et de la caractérisation du résultat attendu selon deux points de vue :  
(i) fonctionnel (fonctions récursives primitives, point fixe...)  
(ii) systèmes à base de règles (terminaison, convergence, forme normale...)  
Ces deux points de vue permettent de mieux appréhender les concepts fondamentaux du calcul et d'en aborder les propriétés essentielles (correction, complétude et terminaison).

#### **Compétences**

Modéliser et formaliser un processus de calcul

#### **Bibliographie**

— Term Rewriting and All That, Franz Baader, Tobias Nipkow, Cambridge University Press, 1999  
— Logique, réduction, résolution, R. Lalement, Masson 1990  
— Résolution d'équations dans les lan-

gages d'ordre 1, 2, ..., omega, G. Huet, 1976

<http://gallium.inria.fr/~huet/PUBLIC/Huet1976.pdf>

— Term Rewriting Systems and Algebra, P. Lescanne, 7th International Conference on Automated Deduction, Lecture Notes in Computer Science 170, pp 166—174, Springer, 1984.

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 10529

P12

P13

## THÉORIE DES LANGAGES 2 ET COMPILATION

Responsable [Claire Lefèvre](#)

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

Ce module est une présentation des concepts et principaux résultats de la théorie des langages permettant de caractériser formellement la notion de langages, leur classification et les propriétés qui en découlent. Le cours permet, en particulier, d'introduire les concepts de base nécessaires à la caractérisation des classes de langages utilisés en compilation pour l'analyse efficace des langages de programmation.

P13

P14

## ARCHITECTURE DES ORDINATEURS

Responsable [Jean-Michel Richer](#)

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

— Comprendre le fonctionnement d'un ordinateur au travers de l'étude de ses différents composants (principalement



microprocesseur, mémoire, mémoire cache).

— Mise en pratique par utilisation du langage assembleur commun aux architectures x86 (Intel, AMD).

— Représentation de l'information (nombres binaires, binaires signés, à virgule flottante).

— Algèbre de Boole : simplification de fonctions booléennes.

### **Bibliographie**

Programmation Assembleur x86 32 et 64 bits sous Linux Ubuntu, Jean-Michel Richer, accessible gratuitement depuis l'URL :  
<https://leria-info.univ-angers.fr/~jean-michel.richer/assembleur.php>

### **Ressource en ligne disponibles**

Les TD et TPs sont accessibles à partir de l'URL :  
<http://localhost/pub/richer/ensl3i.php>

# ALGORITHMIQUE ET PROGRAMMATION

P13

## PROGRAMMATION LOGIQUE

Responsable [Igor Stephan](#)

### • PRÉ-REQUIS

#### **Notions et contenus**

Fondements de l'informatique

#### **Compétences**

Comprendre les notions de langage logique, d'interprétation et de preuve de la logique du premier ordre.

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

— Présentation du cadre théorique de la programmation logique et comparaison avec les autres grands paradigmes de programmation.

— Présentation de la programmation en clauses de Horn (et de sa sémantique via la règle de SLD).

— Présentation du langage Prolog (et de ses composantes extra logiques).

#### **Compétences**

— Comprendre les concepts élémentaires de la programmation logique (mécanismes de la résolution SLD).

— Maîtriser la manipulation des structures de données élémentaires (liste, arbre et graphe) en programmation logique.

— Maîtriser le parcours d'un espace de recherche décrit en extension.

— Comprendre la notion de négation par l'échec.

— Comprendre les points communs et les différences entre les différents paradigmes de programmation.

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 24032

P11

## PROGRAMMATION FONCTIONNELLE

Responsable [Claire Lefèvre](#)

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

— Comprendre les concepts fondamentaux de la programmation fonctionnelle (fonctions, récursivité et filtres).

— Maîtriser la manipulation des structures de données élémentaires (liste, arbre et graphe) en programmation fonctionnelle.

P14

## ALGORITHMIQUE DES GRAPHS

Responsable [Jin-Kao Hao](#)

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

Les graphes sont un objet mathématique qui dispose de très nombreuses applications dans presque tous les domaines. Dans ce cours, nous présentons les concepts généraux (graphe orienté, graphe non-orienté, graphe valué, degré, chemin, cycle, fermeture, connexité, isomorphisme, clique, stable, coloration...), quelques classes de graphe spécifiques (planaire, biparti, complet) et des algorithmes pour trois problèmes classiques d'optimisation (plus court chemin, arbre couvrant minimum et flot-maximum).



## PROGRAMMATION ORIENTEE OBJET 2

Responsable [David Genest](#)

### • PRÉ-REQUIS

#### **Notions et contenus**

Algorithmique 3

Programmation Orientée Objet 1

#### **Compétences**

- Maîtriser les bases de la programmation impérative en C++.
- Maîtriser les bases de la programmation orientée objet.
- Connaître le cycle de développement d'une application dans un langage compilé : compilation, édition de liens, débogage, tests.

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

- Maîtriser les concepts de la programmation orientée objet (notamment encapsulation, héritage, classes abstraites, polymorphisme) et être capable de concevoir et développer efficacement une application complexe.
- Être capable de développer en « C++ Moderne », en comprenant et utilisant correctement les spécificités du C++ (notamment méthodes virtuelles et virtuelles pures, surcharge d'opérateurs, modèles de classes, bibliothèque standard) y compris les évolutions récentes du langage (notamment les pointeurs intelligents ou les fonctions anonymes).
- Être capable de choisir de bonnes structures de données et algorithmes pour résoudre un problème, aussi bien en termes d'occupation mémoire que d'efficacité des traitements.
- Savoir utiliser des outils facilitant le développement et la mise au point (environnement de développement intégré, débogueur, analyseur statique et dynamique du code, mise en place de tests).
- Posséder les bases de développement

d'une interface graphique à l'aide de la bibliothèque Qt (slots et signaux ; quelques widgets simples).

#### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 11942

<https://en.cppreference.com/w/>

# TECHNOLOGIE DE L'INFORMATION

P11

## TRAITEMENTS DE DONNÉES

Responsable [Marc Lgeay](#)

### • PRÉ-REQUIS

#### **Notions et contenus**

Algo 1, 2, 3, 4, 5 (L1-MI, L2-I)  
Système et UNIX (L2-I)  
Python (L1-I)

#### **Compétences**

Savoir concevoir des algorithmes et des structures de données afin de répondre à un problème donné.

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

L'unité « Traitements de données » enseigne tout le déroulé d'un traitement de données scientifiques : mettre en forme les données bruts pour qu'elles puissent être utilisables par la suite, analyser les données pour créer de la connaissance. Utilisation du paradigme « vectoriel » pour le traitement des données volumineuses.

Présentation des notebooks pour faciliter la documentation du code.

#### **Compétences**

— Savoir se servir aisément des plusieurs styles/paradigmes algorithmiques et de programmation ainsi que des langages de script.

— Savoir concevoir des traitements informatisés de données scientifiques : formatage de données brutes et traitement des données formatées.

— Être capable de choisir, sur des critères objectifs, les structures de données et de construire les algorithmes les mieux adaptés à un problème donné.

— Savoir utiliser un environnement logiciel de calculs formels ou scientifiques.

P11 P12

## BASES DE DONNÉES

Responsable [Touria Ait El Mekki](#)

### • CONTENUS DE L'UE

#### **Objectifs pédagogiques**

— Savoir créer et gérer une base de données non normalisée et à valeur complexe.

— Maîtriser le langage procédural PL/pgSQL (ajouter des structures de contrôle ; créer des fonctions standards, des exceptions, des règles et des triggers ; effectuer des traitements complexes ; pouvoir utiliser tous les types, les fonctions et les opérateurs définis par les utilisateurs).

— Connaître les concepts de base pour optimiser les requêtes relationnelles (structures physiques, structures logiques, modèles de coût et choix du meilleur plan).

P11 P12

## DÉVELOPPEMENT WEB 3

Responsable [David Lesaint](#)

### • PRÉ-REQUIS

#### **Notions et contenus**

UE Développement Web 2

UE Bases de données 2

UE Programmation orientée objets 2

#### **Compétences**

— Maîtriser les bases du langage de programmation PHP (génération de page HTML, manipulation de structures de données et de contrôle, validation de formulaires, mise en place de sessions et cookies, accès à fichiers et bases de données).

— Maîtriser les bases du langage de programmation Javascript (couches fonctionnelle et objet, manipulation du DOM



d'une page HTML).

## • CONTENUS DE L'UE

### **Objectifs pédagogiques**

— Présentation de concepts avancés en PHP : couche objet et réflexion, principales API et librairies.

— Présentation de concepts avancés en Javascript : programmation événementielle, DOM et formulaires, API fetch et promesses, cookies et persistance locale.

— Initiation aux méthodes et outils du développement web : style d'architecture Modèle-Vue-Contrôleur, tests unitaires, moteurs de gabarits.

### **Compétences**

— Savoir développer une application web utilisant PHP et Javascript.

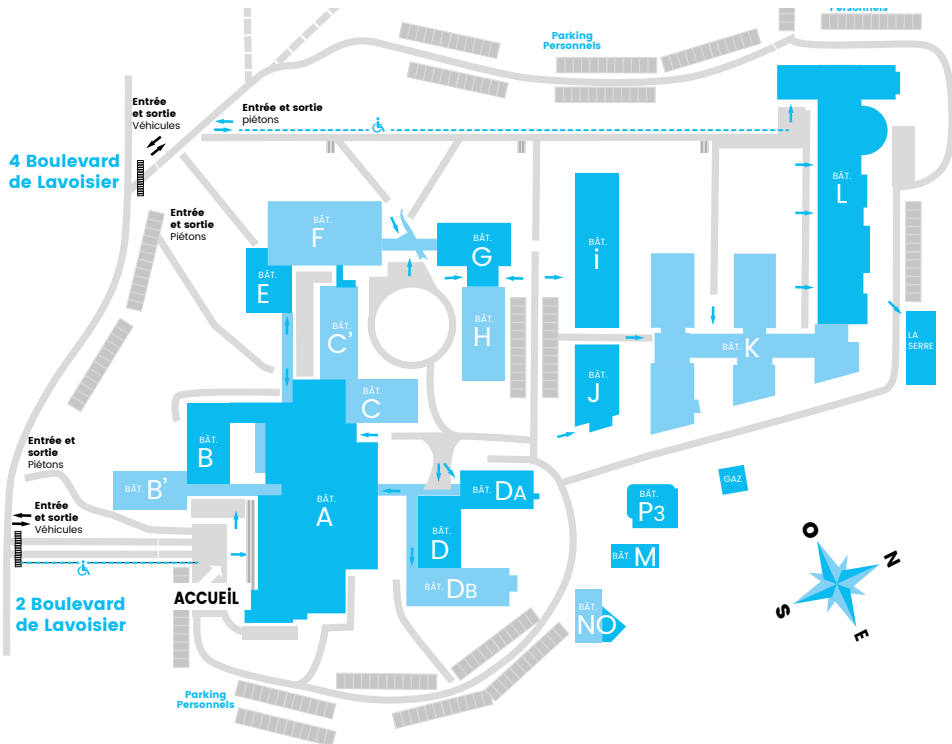
— Maîtriser les principes des architectures MVC dans le cadre des applications web.

### **Ressource en ligne disponibles**

Numéro de cours sur Moodle - 6924







- A** Administration | Scolarité | Enseignement (Amphi A à E)
- B** Biologie végétale | Physiologie végétale | Travaux pratiques biologie
- B'** Travaux pratiques biologie
- C** Travaux pratiques chimie
- C'** Département de Géologie | Recherche environnement (LETG -LEESA) | Recherche géologie (LPGN-BIAF)
- D** Travaux pratiques physique
- Da** Enseignement | Travaux pratiques physique
- Db** Département de Physique | Recherche physique (LPHIA)
- E** Travaux pratiques biologie
- F** Département de Biologie | Recherche neurophysiologie (SIFCIR) | Travaux pratiques biologie, géologie
- GH** Département informatique | Recherche informatique (LERIA) | Travaux pratiques géologie
- I** Département Mathématiques | Recherche Mathématiques (LAREMA)
- J** Chimie enseignement | Travaux pratiques
- K** Département de Chimie | Recherche Chimie (MOLTECH Anjou)
- L** Espace multimédia | Enseignement (Amphi L001 à L006) | Salle d'examen rez-de-jardin

**Ua**  
**FACULTÉ  
 DES SCIENCES**  
 UNIVERSITÉ D'ANGERS

2, Boulevard Lavoisier  
 49045 ANGERS CEDEX 01  
 T.0241735353  
[www.univ-angers.fr](http://www.univ-angers.fr)