

# P00

Mr Palermo David  
david.palermo@ynov.com

# David Palermo

- ▶ Société Yantra Technologies : Formations, Expertise, Conseil, Développement  
[www.yantra-technologies.com](http://www.yantra-technologies.com)
  
- ▶ Formateur Chez Ynov 2018-2019
  - ▶ C++
  - ▶ POO
  - ▶ C
  - ▶ Python

# Règles du jeu - Retard en cours

- ▶ Début des cours à l'horaire prévu et non arrivée à l'horaire prévu
- ▶ Soyez présents et opérationnels avant
- ▶ En cas de retard, passer par l'administration pour demander un mot de retard
- ▶ Ceci est valable aussi pour les inter cours

# Règles du jeu - Absences

- ▶ Informer l'administration
- ▶ Me prévenir par mail
- ▶ Se renseigner auprès de la promo (délégués) sur ce qui a été fait ou donné à faire
- ▶ **Toute absence non justifiée à une évaluation sanctionnée d'un 0**

# Les modalités d'évaluation

- ▶ Les notes seront sur 20 points avec un coefficient associé.
- ▶ Les notes d'exercices et de projet seront notées de telle sorte que :
  - ▶ Une note inférieure à 10 implique que le niveau n'est pas acquis
  - ▶ Une note entre 10 et 15 implique que le niveau est acquis
  - ▶ Une note entre 16 et 20 implique que le niveau est acquis et que le travail demandé est fait avec des fonctionnalités supplémentaires soit non demandées, soit optionnelles, soit des exercices en plus ....
- ▶ Pour les projets les groupes doivent être normalement de 3 personnes :
  - ▶ Si le groupe est de 4 personnes il faut en faire plus qu'un groupe de 3 et la note en tiendra compte
  - ▶ Si le groupe est de 2 personnes il faut en faire autant qu'un groupe de 3

# Règles du jeu - Retard de livraison, absence à une évaluation

- ▶ En cas de retard :
  - ▶ Me prévenir par mail
  - ▶ Tout non rendu sera sanctionné par un 0
  - ▶ Pénalités de retard :
    - ▶ Moins 12h de retard : -2 points sur la note
    - ▶ De 12h à 24h de retard : -4 points sur la note et signalé à l'administration
    - ▶ Au-delà de 24 h la note sera de 0 : Vous serez convoqué(e) un samedi pour réaliser un travail qui sera noté pour rattraper la note

# Règles du jeu - Comportement

- ▶ Téléphone portable : strictement interdit, rangé dans le sac ou la poche sauf autorisation expresse de ma part
- ▶ Ordinateur :
  - ▶ Opérationnel (batterie chargée, alimentation disponible, connexion WiFi active, pas de mise à jour OS dans la journée) ; en cas de panne, anticipez : au moins un ordinateur pour 2
  - ▶ Utilisation en lien direct avec le travail à réaliser (prise de note, rédaction de livrables, recherches) ; tout autre usage (réseaux sociaux, jeux etc...) est strictement interdit
- ▶ **Niveau sonore : pas de bavardages pendant les parties théoriques des sessions, maintenir un niveau sonore modéré pendant les travaux en groupe**
- ▶ Déplacements dans la classe ou sorties de la salle : exceptionnels et uniquement sur autorisation expresse de ma part
- ▶ A part de l'eau, aucun aliment ou boisson ne doivent être visible dans la salle
- ▶ En cas de manquements :
  - ▶ Un rappel verbal
  - ▶ Exclusion immédiate si un deuxième rappel s'avérait nécessaire

# Pédagogie inversée

## Le principe

« Ce que nous devons apprendre à faire, nous l'apprenons en le faisant. »

Aristote

## La pratique

- ▶ Exercices pratiques
  - ▶ Le premier dès aujourd'hui
- ▶ Apports théoriques en cours
  - ▶ Je vous donne les clés et les outils pour avancer, je réponds à vos questions, je suis là pour vous aider
- ▶ Recherches personnelles
  - ▶ L'initiative vient de vous
  - ▶ Soyez curieux, « Le remède à l'ennui, c'est la **curiosité**. ... »



# Participation

« Il n'y a pas de questions idiotes, seulement des réponses idiotes »

Albert Einstein

# Planning prévisionnel et contenu du cours

1. Génie Logiciel : Introduction ( 1 à 2 séances )
2. UML : Introduction ( 3 à 4 séances )
3. Programmation Orientée Objet : Les Concepts( 6 à 4 séances)
  - Présentation
    - Les styles de programmation
    - L'approche Objet et Langage UML
    - Qu'est-ce qu'un Objet ?
    - La modélisation avec UML
    - L'approche orientée objet
    - Langage Objet
    - Les avantages de la POO
  - Concepts de bases
    - Présentation de la notion d 'Objet
    - La modélisation avec UML
    - Le principe de l'encapsulation
    - La Classe
    - L'Héritage
    - Agrégation
    - Polymorphisme
    - Les classes abstraites
    - Relations d'association
    - Le Langage Objet
    - Développement d 'un projet orienté Objet

Contenu donné à titre indicatif et susceptible d'être modifié

# Objectif du module : compétences acquises en fin de module

- L'objectif est de pouvoir maîtriser les différentes notions fondatrices de la programmation orientée objet.

# Ressources

- ▶ Découvrir la modélisation UML

- ▶ <https://www.lynda.com/fr/Astah-tutoriels/Decouverte-modelisation-UML/193468-2.html>

- ▶ Les fondements de la programmation : La conception orientée objet

- ▶ <https://www.lynda.com/fr/Software-Development-tutorials/fondamentaux-programmation-conception-orientee-objet/193430-2.html>

- ▶ Comprendre la programmation orientée objet

- ▶ <https://www.lynda.com/fr/Python-tutoriels/Comprendre-programmation-orientee-objet/193145/344657-4.html>

# Evaluations

- ▶ QCM (20 questions coefficient 1 , 10 questions coefficient 0,5)
  - ▶ QCM les 15/03/19, 22/03/19, 25/03/19
- ▶ Application de la POO dans le cours Python

Ceci est le planning « minimum », d'autres évaluations peuvent avoir lieu à tout moment

# Vos attentes

Merci de remplir le formulaire en ligne  
Le lien est dans vos emails

« Vis comme si tu devais mourir demain. Apprends comme si tu devais vivre toujours. »

Gandhi

« Quand un homme a faim, mieux vaut lui apprendre à pêcher que de lui donner un poisson. »

Confucius

« J'ai appris que la voie du progrès n'était ni rapide ni facile. »

Marie Curie