

# Dév Python

Michael  
X  NATIS



Un dév, ça sert à quoi ?

MICHAEL



MICHAEL



# Un dév écrit des programmes

Oui, il fait de l'écriture !



Comment appelle-t-on un développeur à l'époque (avant les années 2000)?





# Analyste-Programmeur

**Analyste** : le/la développeur(se) doit savoir analyser la demande pour structurer une solution (c'est l'algorithmique)

**Programmeur** : à partir de sa solution, il/elle la traduit dans un langage (JS, PHP, Python etc.)



## OBJECTIFS

- Connaître les possibilités et limites de Python
- Maîtriser la syntaxe essentielle de Python
- Être capable de créer et maintenir des applications complètes et complexes en Python
- Découvrir les fonctions avancées du langage Python
- Découvrir les domaines d'applications variés
- Programmer pour le web avec Python

## POINTS CLES

- Découverte de la syntaxe Python
- Bibliothèques et outils StdLib, QA, TkInter
- Développement web

## PROGRAMME

Introduction : historique, succès et positionnement du langage Python

Syntaxe du langage Python

Les variables

Opérateurs et expressions

Les structures de contrôle

Les procédures et les fonctions

Rappels de l'approche Orientée Objet

Utilisation StdLib

Outils QA

Création IHM TkInter

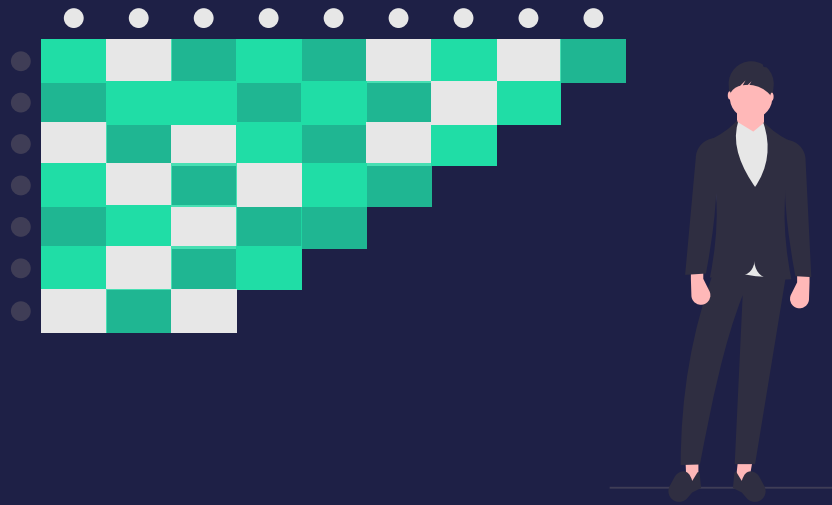
Interfaçage Python / Java

Interfaçage Python / Base de données

Programmation Web avec Python

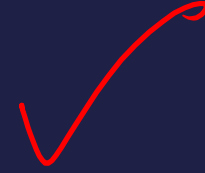
Maintenance, débogage et test des programmes

# PLAN



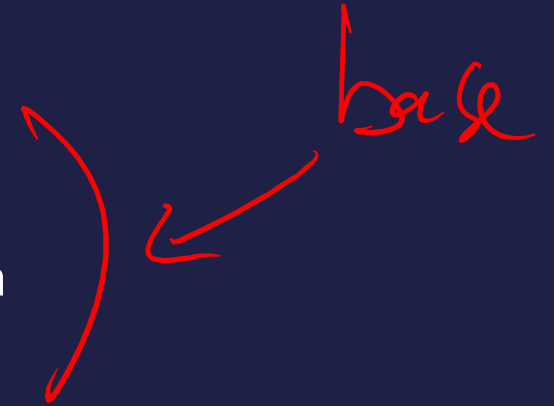
# 1. HISTOIRE

Historique et positionnement de Python



# 2. ALGORITHMIQUE

Algorithmique et structures de données en Python



# 3. POO

La programmation orientée objet

## 4. INTERFACAGE

Py4j, sqlite3, selenium

## 5. TK & WEB

TkInter & Flask

## 6. ENV & TESTS

Maintenance, environnement, doctest & unittest



# C'EST PARTI ?

