



**Initiation**  
**Dorian.H Mekni**

29|01|2020

# Historique

- ★ 1970
- ★ Language de bas niveau
- ★ Language de programmation imperative
- ★ Denis Richie & Ken tomson
- ★ Language
  - ★ syntaxe simplifie par rapport aux languages d'assembleurs
  - ★ Gérer la memoire, reserve un espace dans la memoire d l'ordi pour y placer des infos.
  - ★ Stockage de données, les gérer et les libérer.
  - ★ Sans garbage collector (ramasse miettes)

# Pourquoi apprendre le C ?

- ★ Longue longévité.
- ★ Ce n'est pas un langage propriétaire
- ★ Apple vendu en C sans reverser de droits
- ★ Standard ouvert (normes C89, C99, C11, Anzi)
- ★ Feuille de route + routes ( règles de codage, syntaxe, méthode de programmation)

**C89 écrit en 1989**

**C99 écrit en 1999**

**C11 écrit en 2011**

# Faisabilité

- ★ Même syntaxe windows, Linux ou Mac.
- ★ Sans cross-compilation
- ★ executable en fonction de l'OS sur lequel on compile: compilé sur Mac executable sur Mac, idem pour les autres.
- ★ Peu de concepts
- ★ Syntaxe syntaxique
- ★ Bien allouer la mémoire
- ★ Focus

# Les + du C

- ★ Inclus dans tous les appareils électroniques
- ★ Systeme embarqué ex: Drones, Frigo, Tele
- ★ 95% ces petits systèmes programmés en C
- ★ Programmation système sur serveur | en C
- ★ Des jeux + applis en temps reels | en C
- ★ Des bibliothèques facilitatrices pour les Devs
- ★

-> C <-

- Le noyau Linux - Kernel | en C
- SDL code pour manipuler les images | en C
-

# EDI

- Code Binaire = executable
- Environment de dev: éditeur de code - colorer votre syntaxe, avoir des raccourcis
- Debbuger



# Preferences

- ★ Compilation en ligne de commande
- ★ Toujours apte à développer
- ★ Rester indépendant
- ★ Polyvalent

# MAC OS

Xcode

# Extension de fichier

**.c**