

> C <

Variables

# Spécificités

- Volatile
- + Stockage plein + l'ordinateur fonctionnera de manier lente ->

# Commentaire

- `//` Commentaire d'une ligne
- `/* */` Commentaire de plusieurs lignes ->

# Type

- Type dynamique
- Chaque type = \*octet
- nombre algébrique soit positif | négatif
- signed int OU unsigned int == nombre entier
- signed char OU unsigned char == lettre simple
- char == chaine de caractère | texte
- float == nombre à virgule ->

# C embarqué

- Drone
- Pratique écologique des types pour économiser de la mémoire
- Robotique ->

# Des mots réservés

auto, break, case, char, const, default, do, double, else, enum, extern, float, for, goto, if, int, long, register, return, short, signed, sizeof, static, struct, switch, typedef, switch, typedef, union, unsigned, void, volatile, while ->

# Code

- Camel case
- Sensitive language
- pop\_style | popStyle | \_popStyle
- `int popSyle // Déclaration d'une variable de type entier (int)`
- Les variable doivent être déclarées au debut de votre fonction ->

# var ID

- Déclaration et affectation
- `int popStyle = 0; // popStyle = 0xF650C`
- `0xF650C` = adresse de stockage pour la variable
- `0xF650C` = adresse
- `int` = affectation de variable `int` ->



# Variable

- La modification de la valeur est variable et peut donc changer

# CAST

```
float number= 123.56;
```

Cast | Conversion

```
int integer = (int)number
```

->

# Aussi...



- Dans un code source, il est possible de voir:

```
int integer = (int)123
```

OU

```
int integer = (int)123.46
```

->

# Affichage

/\*

%d : nombre entier

%f : nombre flottant (float)

%c : caractère (char)

%s : chaîne de caractère (texte)

\*/

->

# Mémoire: register

- `register int integer = 5;`
- Registre | au niveau du processeur, c'est la mémoire la + rapide sur l'ordi avec la mémoire cache qui y est associé
- on pas stocké dans la mémoire vive du Mac donc la rame mais dans le registre
- S'assurer qu'elle soit manipulable plus rapidement
- On gère ainsi la priorité des variable
- Ce qui rend le programme + rapide + performant
- `register` c'est la mémoire la plus rapide. On la rend prioritaire

# Mémoire cache

- Mémoire très petite | quelque mega octets
- Voir sur le processeur

->

# Mémoire: volatile

- **volatile** int number = 67
- **volatile** est accessible seulement au niveau de la rame
- Certaines données accessible au niveau du clavier et non de la souris. Elles seront stockées dans une variable **register**
- A l'inverse, pour des informations accessibles via la souris ou clavier, les variables seront encapsulées dans un **volatile** ->

# Const

- `const` encapsule une valeur qui ne peut changer
- les constantes sont toujours nommés en majuscule
- `const float PI = 3.14`

->



**END**