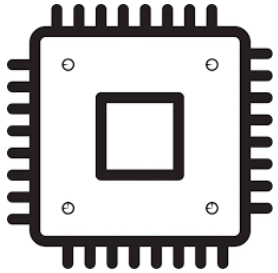
Activités en première



CPU

Programmation
Assembleur

Modèle de
von Neumann

Architectures …

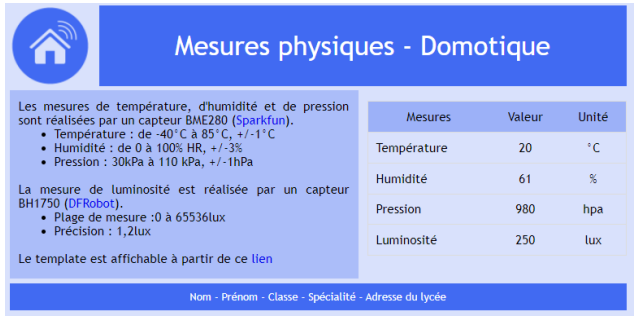
La base 10	15	14	13
La base 2	1111	1110	1101
La base 16	F	E	D

Decimal	Hex	Char
64	40	@
65	41	A

Représentation des données : types et valeurs de base

P. Mariano

IHM

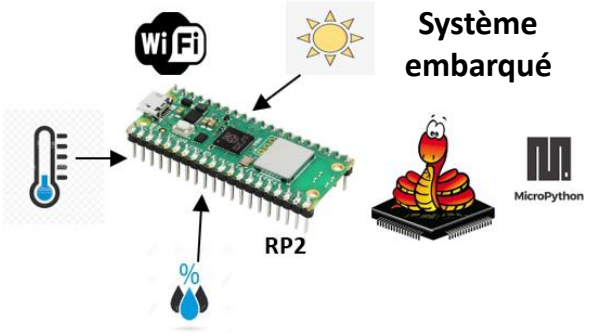


TCP/IP

Réseaux

Architectures …

Système
embarqué

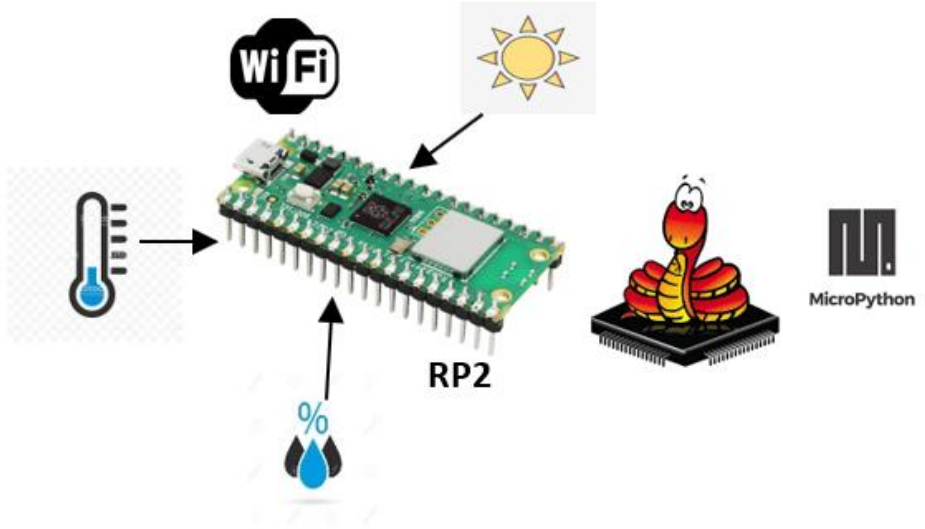


Architectures …

Linux

Systèmes d'exploitation

Exemple de mini projet (guidé)



Domotique

Raspberry Pi 4

Activités en première

Y. Leclerc

```
import csv
with open('scientifiques.csv') as fichier:
    scientifiques = list(csv.DictReader(fichier,
delimiter=";"))
```

Code partiel

	A	B	C	D	E	F
1	nom	prenom	jour	mois	annee	projet
2	Kernighan	Brian	1	1	1942	programmer avec style
3	Hopper	Grace	9	12	1906	code machine
4	Torvald	Linus	28	12	1969	systeme d'exploitation
5	Knuth	Donald	10	1	1938	tout sur les algorithmes

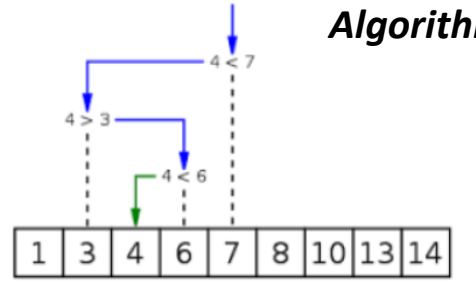
Traitement de données en tables

Tri par sélection

```
def TriSelection(T,n) :
  for i in range(n-1) :
    Posmin=i
    for j in range(i+1,n) :
      if T[j]<T[Posmin] :
        Posmin=j
    T[i],T[Posmin]=T[Posmin],T[i]
```

3	7	2	6	5	1	4
1	7	2	6	5	3	4
1	2	7	6	5	3	4
1	2	3	6	5	7	4
1	2	3	4	5	7	6
1	2	3	4	5	7	6
1	2	3	4	5	6	7

Recherche dichotomique



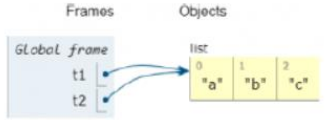
Algorithmique

```
//Boucle de recherche partielle
  Tant que trouvé != vrai et début <= fin:
      mil ← partie_entière((début + fin)/2)
      si t[mil] == val:
         trouvé ← vrai
      sinon:
         si val > t[mil]:
            début ← mil+1
         sinon:
            fin ← mil-1
```

Représentation des données : types construits

Tableau

```
t1 = ['a', 'b', 'c']
t2 = t1
```



Dictionnaire

```
dict = {} # dictionnaire vide
dict = {'nom':'Martin', 'prenom':'Pierre-Emile'}
```

Exemples de mini projet (guidé)

```
A vous de jouer!
   0 1 2 3 4 5 6
  ---------------
5 |.|.|.|.|.|.|.|
4 |.|.|.|.|.|.|.|
3 |.|.|.|x|.|.|.|
2 |.|.|0|x|0|.|.|
1 |.|.|x|x|x|.|.|
0 |0|0|0|x|x|.|0|
  ---------------

###########################################
Bravo ! Humain (1-x) Aléa vous avez gagné !
La partie est terminée !

###########################################
```

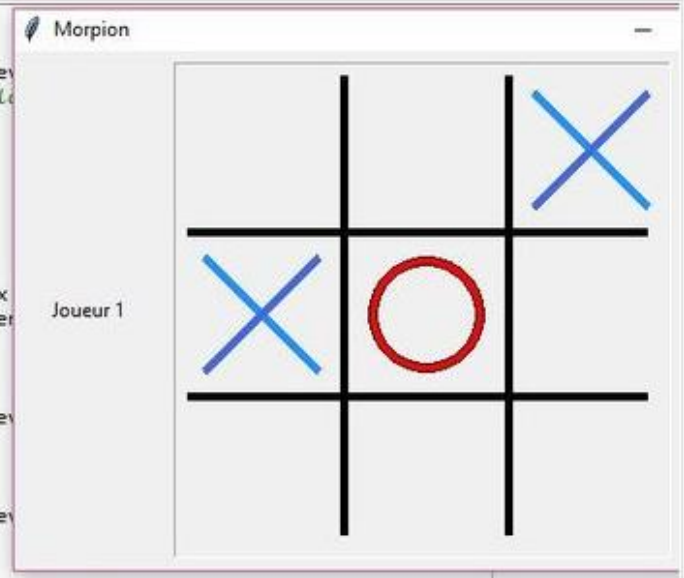
Puissance 4

Tic-Tac-Toe

```
        self.joueur = 1

def analyser_position_clic(self, ev
    #Detection de la position de la
    # 0 | 1 | 2
    # - + - + -
    # 3 | 4 | 5
    # - + - + -
    # 6 | 7 | 8

    if self.grille[0] == 0:
        if event.x > 0 and event.x
            if event.y > 0 and ever
                i = 1
                self.x = 55
                self.y = 55
            if event.y > 100 and ev
                i = 4
                self.x = 55
                self.y = 155
            if event.y > 200 and ev
                i = 7
                self.x = 55
                self.y = 255
        elif event.x > 100 and event.x < 200:
            if event.y > 0 and event.y < 100:
                i = 2
                self.x = 155
                self.y = 55
```



Morpion