

---

## La machine à voter

---

L'objectif de ce TD est de concevoir, puis de programmer à l'aide de `Scilab` une machine à voter :

On fait voter des électeurs tant qu'il y en a entre un candidat A et un candidat B. À la fin, on affiche le nom du vainqueur (s'il y en a un). L'algorithme doit être fiable à 100% et ne doit pas permettre d'erreur de saisie.

### 1 Algorithme

Voici des questions intermédiaires permettant une résolution progressive du problème :

1. On suppose qu'il n'y a qu'un seul électeur et qu'il ne fait pas d'erreur de saisie.
2. On suppose qu'il n'y a qu'un seul électeur et qu'il peut lui arriver de faire une erreur de saisie, mais on ne lui demande pas (encore) de revoter.
3. On suppose qu'il n'y a qu'un seul électeur et qu'il peut lui arriver de faire une erreur de saisie, et on ne lui demande de revoter si nécessaire.
4. On suppose toujours qu'il n'y a qu'un seul électeur, mais on donne le résultat **après** les opérations de vote.
5. On suppose qu'il y a trois électeurs.
6. On choisit le nombre d'électeurs avant de commencer les opérations de vote. (Attention, dans ce cas, il peut y avoir égalité entre les deux candidats.)
7. Avant de commencer les opérations de vote, on demande s'il y a un électeur. (Réponse `oui/non` et erreur de saisie traitée)
8. Enfin, on fait voter des électeurs tant qu'il y en a.

Voici quelques améliorations possibles :

1. Afficher le score.
2. Voter entre trois candidats.
3. Choisir le nombre de candidats (et leurs noms) avant de commencer les opérations de vote. (Attention, c'est plus difficile.)
4. Etc.

### 2 Programme en Scilab

`Scilab` est un logiciel de calcul numérique. Il comporte une console dans laquelle les calculs sont effectués et un éditeur qui permet d'écrire et de sauvegarder des programmes.

Quelques instructions utiles :

- Il n'y a pas besoin de `;` en fin de ligne.
- Il n'y a pas besoin de déclarer les variables.
- Demander une valeur et la lire :

```
x=input("Quel est votre âge? ")
```

 pour lire un nombre

```
x=input("Quel est votre nom? ", "string")
```

 pour lire un texte

- Assigner une valeur à une variable :

```
x=3
```

 pour assigner un nombre

```
x="Texte"
```

 pour assigner un texte

- Faire des calculs :

```
x+y  
x-y  
x*y  
x/y  
x^y
```

- Afficher un message :

```
afficher("Bonjour!")
```

 avec le module lycée

*Attention : On peut demander d'afficher une suite de messages séparés par des virgules, mais l'affichage s'effectue alors dans l'ordre inverse. Par exemple, la commande `afficher("Un", "Deux", "Trois")` donne l'affichage `Trois Deux Un`.*

- Structure conditionnelle :

```
if ... then  
    ...  
else  
    ...  
end
```

avec if et then obligatoirement sur la même ligne

Pour écrire les conditions, on peut utiliser :

- Les valeurs booléennes : %T pour vrai et %F pour faux
- Les tests : `x=y` pour l'égalité et `x<>y` pour la différence
- Les comparatifs numériques : `<`, `>`, `>=`, `<=`
- Les connecteurs booléens : `~` pour la négation, `&` pour la conjonction, `|` pour la disjonction

- Répéter une suite d'instructions un nombre fixé de fois :

```
for i=1:2:10  
    ...  
end
```

de 1 à 10 par pas de 2

- Répéter une suite d'instructions tant qu'une condition est vérifiée :

```
while ...  
    ...  
end
```

### 3 Un exercice à faire à la maison

Dans le même ordre d'idée, concevoir et programmer un jeu de « Trop grand / Trop petit ».

Une commande utile en Scilab :

- Tirer un entier au hasard entre 0 et 99 :

```
nombre=tirage_entier(1,0,99)
```

 avec le module lycée