

# elisp pour Emacs



Par Vincent Côté  
pour le cours  
E14-Maintenance d'applications

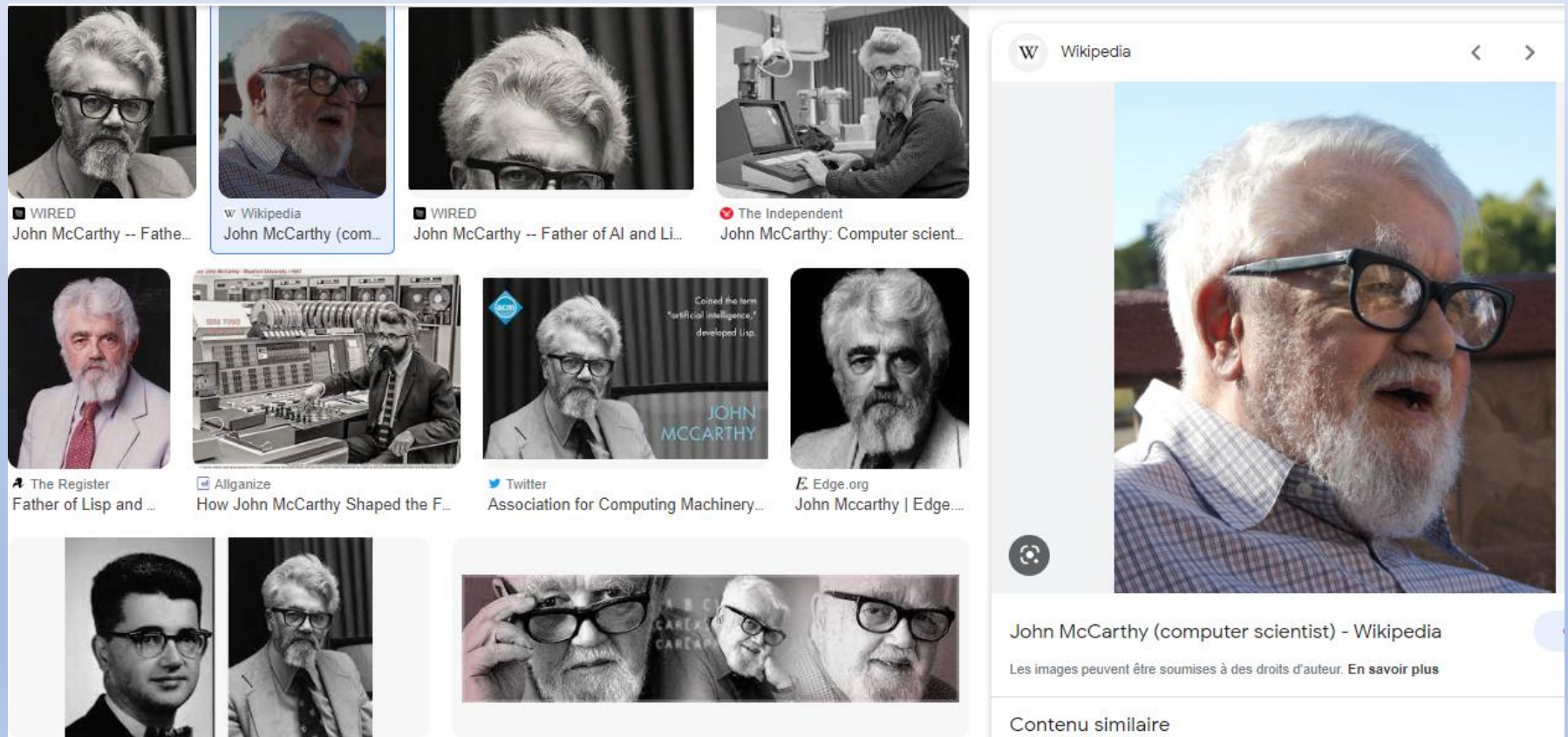
Présenté à  
Jocelyn Cloutier  
le 26 janvier 2023

# Bienvenue

- Acronyme : elisp = Emacs Lisp = Editing MACroS LISt Processor
- Apprentissages pour la vie
- Objectif : Présenter l'existence du langage elisp et de l'éditeur Emacs
- Déroulement de l'exposé : d'où ça vient, à quoi ça sert et exemples.

# Lisp 1958 par John McCarthy

- Objectif : Créer un langage simple & extrêmement puissant dans le cadre de ses Recherches en IA
- Langage Fonctionnel
- Gestion Mathématique plutôt que d'objets
- Plus efficace parce qu'il ne s'occupe pas des états globaux et parce qu'il utilise des concepts simples par rapport à ceux plus complexes de l'Orienté Objet comme l'héritage et le polymorphisme.



# Langage à parenthèses ouvertes VS accolés

*;;;Opérations mathématiques*

(+ 2 2)

(\* (+ 2 3) (- 5 3))

```
static void Main(string[] args)
{
    {
        Titre();
        InitierJeux();
        PreparerTableaux();
        Jouer();
    }
    while (Encore());
}
```

# On parle de langage, mais en réalité C'est une "Famille de langages"

- Elisp = dialect qui découle du dialect CommonLisp



- elisp est intégré à Emacs afin d'étendre ses fonctionnalités



Clojure



# Théorie

- Syntaxe (symbole ou parenthèses)

`(make-list 3 voter)` → → →

```
1 Debugger entered--Lisp error: (void-variable voter)
2 (make-list 3 voter)
3 elisp--eval-last-sexp(nil)
4 eval-last-sexp(nil)
5 funcall-interactively(eval-last-sexp nil)
6 command-execute(eval-last-sexp)
```

- + \*L'apostrophe (') permet de retourner un argument tel qu'il est écrit sans l'évaluer.

↓

↓

↓

↓

`(make-list 3 'voter)` → `(voter voter voter)`

# Théorie

- Structure(listes,cellules,tableaux)
- Fonctionnement de base (faiblement typé)

```
(defvar surprise 4011)
```

4011

```
(setq surprise "coucou")
```

"coucou"

- + \*Homoiconicité

```
(+ 2 2)
```

```
(* (+ 2 3) (- 5 3))
```

[https://fr.wikipedia.org > wiki > Homoiconicité](https://fr.wikipedia.org/wiki/Homoiconicit%C3%A9)

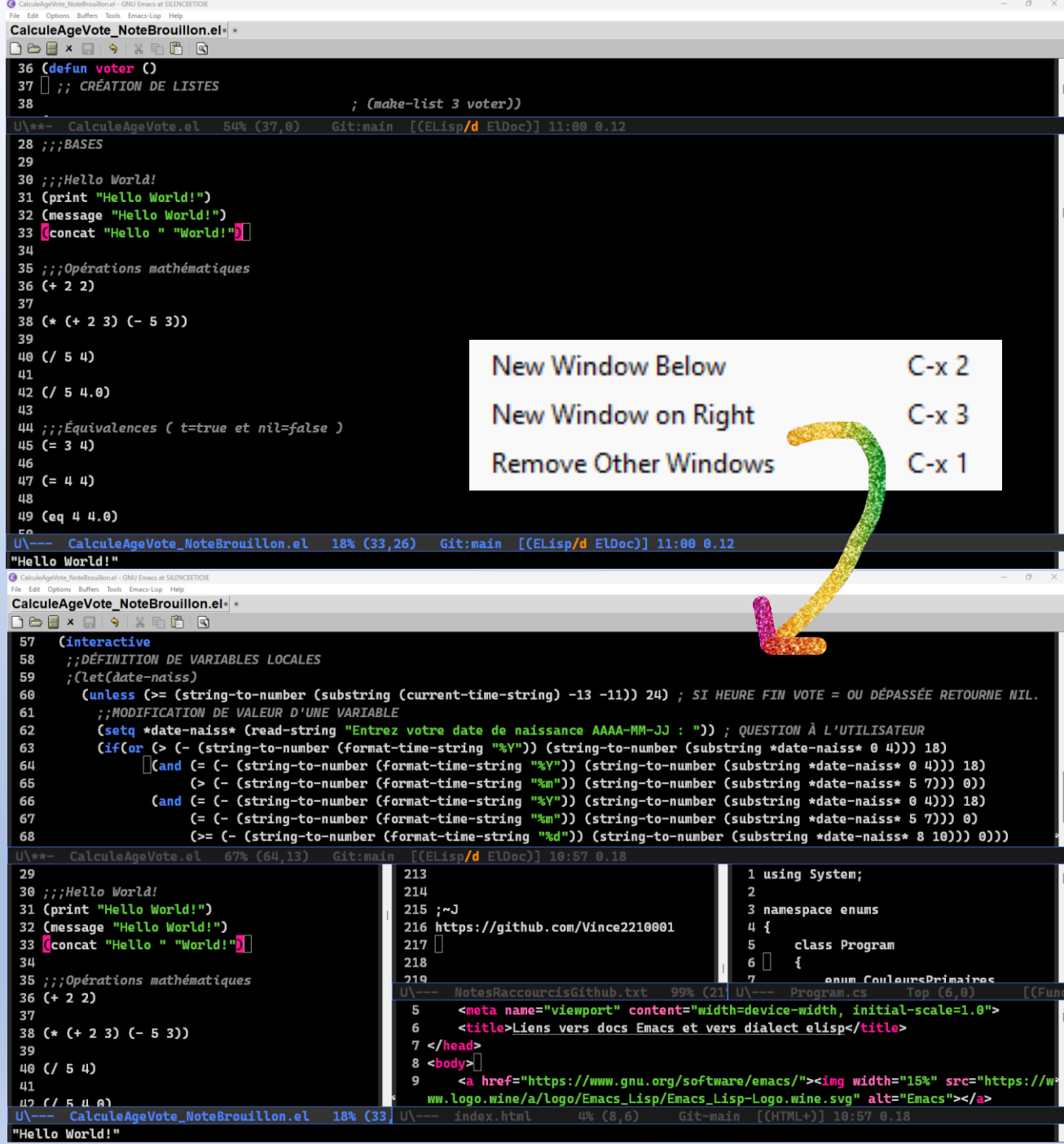
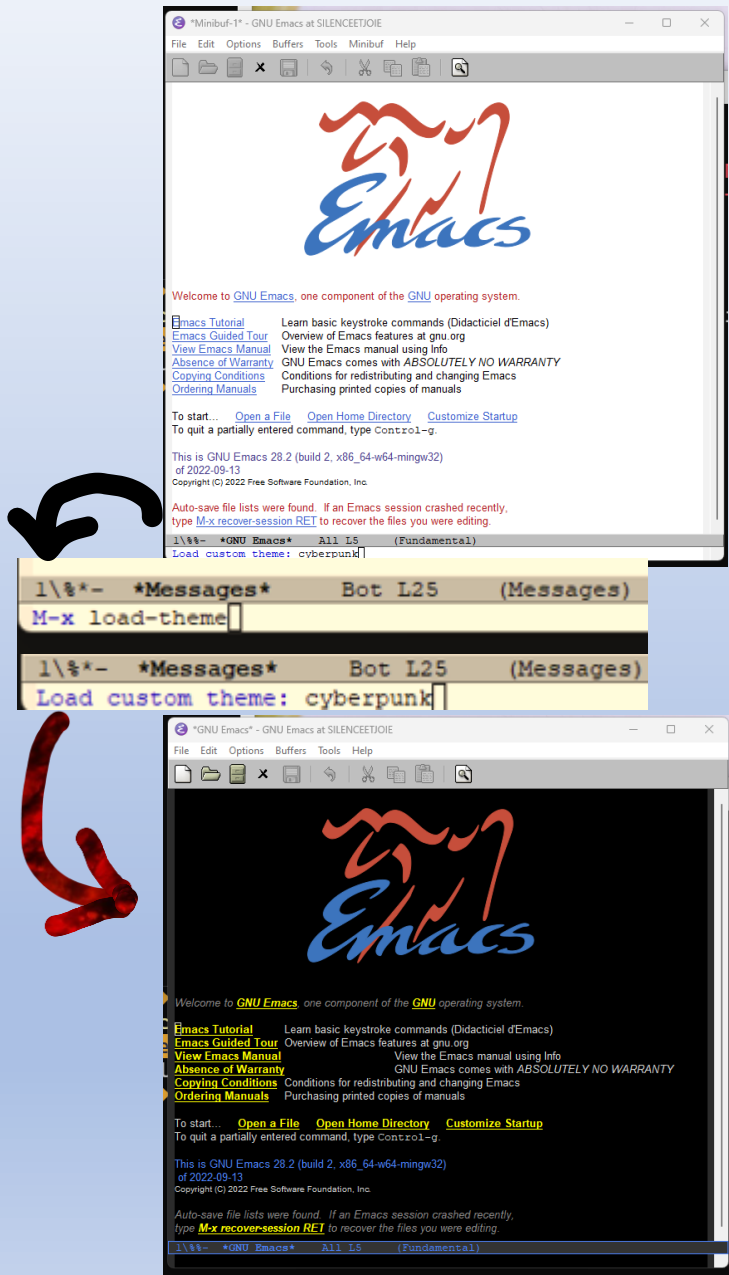
## Homoiconicité — Wikipédia

En programmation informatique, l'**homoiconicité** (de homo- : semblable, et icône : image) est une propriété de certains langages dans lesquels la principale représentation des programmes est aussi une structure de données d'un type primitif du langage.

```
(defun msg-trop-jeune ()  
  (concat "Vous pourrez voter dans "  
          (number-to-string  
            (- 18 (- (string-to-number (format-time-string "%Y"))  
                    (string-to-number (substring *date-naiss* 0 4))))  
          " an(s).")  
  )
```



# Personnalisation Emacs





# Exemple code et autres liens utiles!!!

- <https://github.com/Vince2210001/emacs-elisp>

```
2  ;;;;'/*****
3  ;;;;'*****
4  ;;;;'**
5  ;;;;'** TITRE : E14_TP2_2210001 - PRÉSENTATION ELISP - AVANT MACHINE POUR VOTER
6  ;;;;'** DESCRIPTION : PROGRAMME QUI ÉVALUE LA POSSIBILITÉ DE VOTER EN FONCTION DE L'ÂGE
7  ;;;;'** ET QUI CALCULE L'ÂGE DE L'UTILISATEURICE.
8  ;;;;'** ( CODÉ EN ELISP AVEC L'ÉDITEUR EMACS)
9  ;;;;'** FAIT PAR : VINCENT CÔTÉ ;~J
10 ;;;;'** LE : 26 JANVIER 2023
11 ;;;;'** RÉVISÉ PAR :
12 ;;;;'** LE :
13 ;;;;'**
14 ;;;;'*****
15 ;;;;'*****/
```

# Merci

;~J

- On s'appelle qu'elisp st un langage Fonctionnel à ( ) ouvertes plus simple et aussi plus efficace.
- À la base, je voulais simplement découvrir quelque chose de différent et j'ai eu beaucoup de plaisir. J'ai eu un coup de foudre pour Emacs et pour elisp pour sa polyvalence et sa facilité à lire et à écrire.