

# Le grand livre de eForth web

version 1.0 - 20 octobre 2023



Auteur(s)

- Marc PETREMANN

Collaborateur(s)

- xxx

## Table des matières

<b>Premiers pas avec eFORTH web.....</b>	<b>3</b>
Premiers mots et premières définitions.....	3
Voir les autres vocabulaires.....	3
Première compilation de code FORTH.....	4
Sélection mode texte ou graphique.....	4
<b>Import de fichiers vers eFORTH web.....</b>	<b>6</b>
Edition des fichiers sources pour eFORTH.....	6
Chargement d'un fichier source vers eForth.....	6

# Premiers pas avec eFORTH web

## Premiers mots et premières définitions

Si vous n'avez pas installé eFORTH web, vous pouvez tester notre version en ligne. Lien :

<https://eforth.arduino-forth.com/web-eforth/index/>

Pour constater le bon fonctionnement de eFORTH, tapez **words** dans l'espace de travail eFORTH:

### Essayez eFORTH avec la version en ligne

```
uEforth v7.0.7.15 - rev 564a8fc68b545eb3ab
Forth dictionary: 4065196 free + 79456 used = 4144652 total (98% free)
3 x Forth stacks: 16384 bytes each
ok
--> words
FORTH ok colors ms start-task task pause tasks END OF ENDCASE CASE +to
to exit ; { (local) asm words vlist order see .s startswith? str= :noname
forget dump spaces assert set-title page at-xy normal bg fg ansi ms-ticks
web structures f.s f. #fs set-precision precision fvariable fconstant fliteral
afliteral sf, internals sealed previous also only transfer{ }transfer transfer
definitions vocabulary [IF] [ELSE] [THEN] DEFINED? quit evaluate prompt
refill tib accept echo abort" abort z>s s>z r~ r| r" z" ." s" n. ? . u.
binary decimal octal hex str #> sign #s # hold <# extract pad hld cr space
emit bye terminate key? key type fill32 >name is defer +to to value throw
catch handler K J I loop +loop leave UNLOOP ?do do next for nest-depth
postpone recurse aft repeat while else if then ahead until again begin
[char] char ['] ' used remaining fdepth depth fp0 rp0 sp0 >link >link&
>size >params >name-length >flags >flags& align aligned #! \ ( CALL FP@
FP! SF@ SF! FDUP FNIP FDROP FOVER FSWAP FROT FNEGATE F0< F0= F= F< F> F<>
```

L'exécution de **words** affiche le contenu du vocabulaire FORTH. Tous ces mots sont l'équivalent de fonctions dans d'autres langages. FORTH se différencie de tous les autres langages de programmation, car en FORTH toute nouvelle définition de mot FORTH agrandit le langage FORTH. Il n'y a pas de différence entre une application en FORTH et l'extension du dictionnaire.

Pour effacer le contenu de l'espace de travail, exécutez le mot **page**.

## Voir les autres vocabulaires

**words** affiche le contenu du vocabulaire FORTH. Mais eFORTH web embarque d'autres vocabulaires. Pour voir la liste de ces vocabulaires, tapez **internals voclist**. Affiche:

```
--> internals voclist
tasks
ansi
web
structures
internalized
internals
FORTH
```

```
ok
-->
```

Parmi ces vocabulaires, celui qui va nous intéresser est le vocabulaire **web**. Pour voir les définitions de ce vocabulaire, taper **web vlist**. Affiche:

```
--> web vlist
yielding-task yielding import rm ls download cat include-file upload upload-file
session? web-key? web-key web-type scripts scripts# random button mouse
textWidth fillText font text-size! log raw-http-upload http-download raw-download
upload-success? upload-done? upload-start ms-ticks silence tone importScripts
release keyCount getKey clearItems removeItem getItem setItem smooth gpop
gpush rotate scale translate show-text keys-height mobile textRatios viewport@
window line fill stroke lineTo moveTo beginPath box lineWidth color! text
gr grmode shouldEcho? web-terminate web-key?-raw web-key-raw web-type-raw
jseval JSWORD: jsslot jseval!
ok
```

Un vocabulaire permet d'intégrer des mots utilisables dans un certain contexte. Ici, le contenu du vocabulaire web n'est défini que pour la version eFORTH web. Le vocabulaire web n'existe pas sur les autres versions de eFORTH (Windows, Linux, ESP32...).

## Première compilation de code FORTH

Si une définition FORTH est courte, vous pouvez la compiler immédiatement depuis l'invite de commande de eFORTH web:

```
: myLoop 10 0 do i . loop ;
```

Tapez ensuite **myLoop**, ce qui exécutera le code qui vient d'être compilé par eFORTH web:

```
uEforth v7.0.7.9 - rev fb3db70da6d11b1fdf0
Forth dictionary: 4068200 free + 76708 used = 4144908 total (98%
free)
3 x Forth stacks: 16384 bytes each
ok
--> : myLoop 10 0 do i . loop ;
ok
--> myloop
0 1 2 3 4 5 6 7 8 9 ok
-->
```

## Sélection mode texte ou graphique

Le mot **gr** sélectionne le mode graphique. Ce mot rend visible un espace graphique de type canvas:

```
web gr
```

Pour revenir au mode text, taper **text**.

Voici un exemple de graphisme:

```
web
```

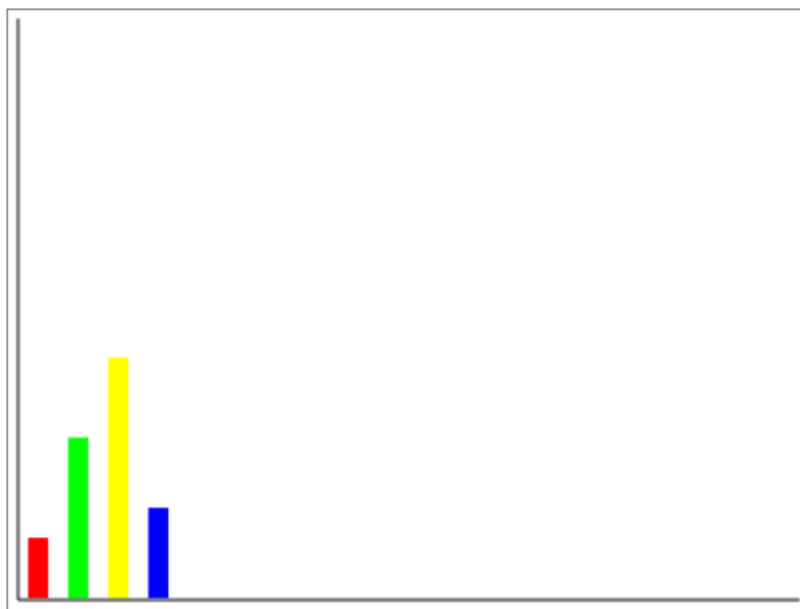
```

: grTest ( -- )
  gr 400 300 window
  $000000 color!
  5 295 395 295 line
  5 295 5 5 line
  $ff0000 color! 10 294 10 -30 box
  $00ff00 color! 30 294 10 -80 box
  $ffff00 color! 50 294 10 -120 box
  $0000ff color! 70 294 10 -45 box
  key drop
  text
;

```

L'exécution de **grTest** affiche ce graphisme:

## Try eFORTH



uForth v7.0.7.9 - rev fh3dh70da6d111b1fdf0

## Import de fichiers vers eFORTH web

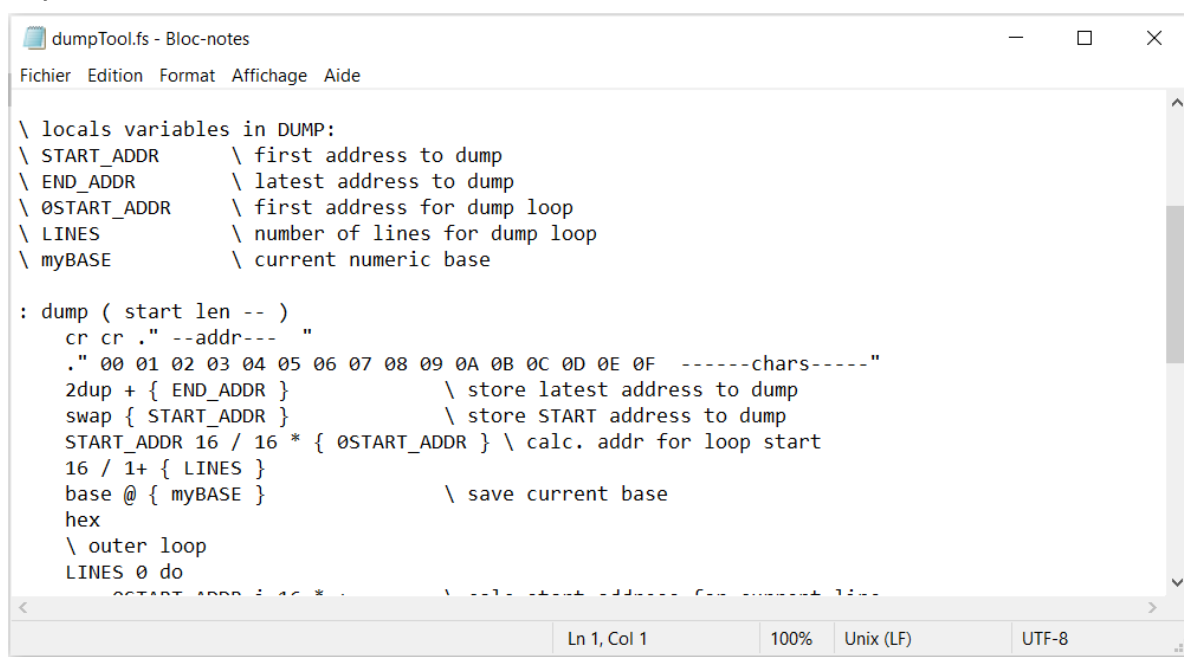
Vous avez installé eFORTH sur votre site web? Ou en local au pire des cas?

OK.

Voyons donc les choses sérieuses. FORTH est intéressant si on développe de vrais programmes.

## Edition des fichiers sources pour eFORTH

Les fichiers sources pour eFORTH peuvent être édités avec n'importe quel éditeur texte: Wordpad, bloc-notes...



```
\ locals variables in DUMP:
\ START_ADDR      \ first address to dump
\ END_ADDR        \ latest address to dump
\ ØSTART_ADDR     \ first address for dump loop
\ LINES           \ number of lines for dump loop
\ myBASE          \ current numeric base

: dump ( start len -- )
  cr cr ." --addr--- "
  ." 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  -----chars-----"
  2dup + { END_ADDR }          \ store latest address to dump
  swap { START_ADDR }         \ store START address to dump
  START_ADDR 16 / 16 * { ØSTART_ADDR } \ calc. addr for loop start
  16 / 1+ { LINES }
  base @ { myBASE }           \ save current base
  hex
  \ outer loop
  LINES 0 do
    ØSTART_ADDR + 16 * + \ calc. start address for current line
```

Ici, édition du fichier **dumpTool.fs** avec le bloc-notes.

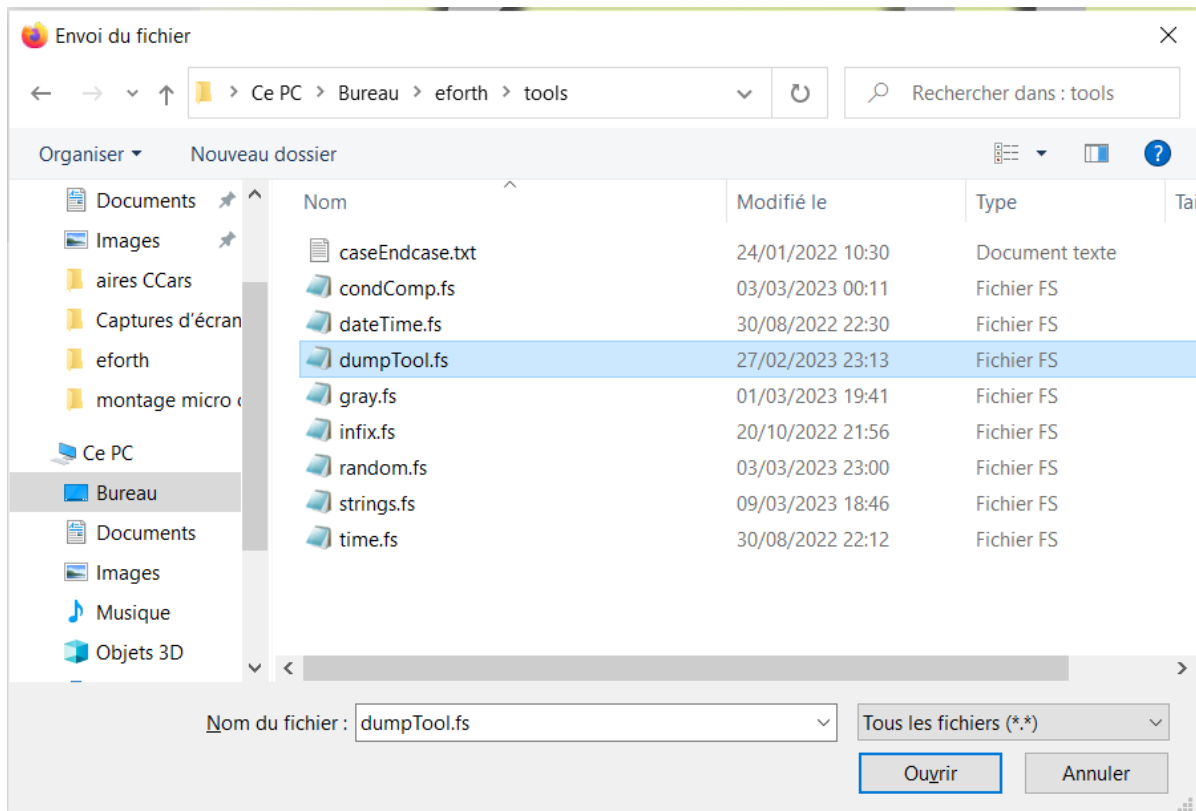
Sauvegardez vos fichiers source dans le répertoire de votre choix.

## Chargement d'un fichier source vers eForth

Pour charger le contenu de notre fichier **dumpTool.fs**, il suffit d'exécuter ces deux mots depuis l'invite de commande de eFORTH:

```
web import
```

La fenêtre de sélection de fichier s'ouvre sur votre ordinateur:



Ici, on sélectionne le fichier **dumpTool.fs** précédemment édité et sauvegardé.

L'envoi du fichier provoque immédiatement le traitement de son contenu. Ici, on a compilé le mot **dump** qu'il suffit ensuite de tester:

```
ok
--> here 100 dump
--addr--- 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F -----chars-----
0001-F450 D4 90 01 00 00 00 00 00 00 00 00 00 00 00 00 .....
0001-F460 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0001-F470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0001-F480 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0001-F490 00 00 00 00 00 00 00 00 00 00 00 00 30 30 31 2D .....0001-
0001-F4A0 46 34 33 33 00 00 00 00 00 00 00 00 00 00 00 F400.....
0001-F4B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
ok
-->
```

Un fichier source pour eFORTH peut contenir des mots à compiler, mais aussi des commandes, dont l'exécution des mots précédemment compilés. Cette facilité permet de gagner un temps considérable en temps de développement.

Seule différence, avec eFORTH pour windows, les fichiers source ne peuvent pas contenir un appel de fichier par include.