

Raccourcis personnalisés

VINCENT DOUCE

pour un T_EX_{MACS} très efficace en cours privés distanciels

TABLE DES MATIÈRES

(0) méthode	1
1. mode texte : lettres répétées	1
triples	1
quadruples	1
2. mode maths : lettres répétées	1
doubles marchent pas	2
doubles	2
triples	2
3. mode maths : symboles divers	2
4. mode maths AZ : codes un peu plus longs	2
5. Exports automatisé	2
6. Mode texte AZ : gros codes	2
7. Explications du code SCHEME	?

(0) méthode

- sélectionner l'objet voulu dans un document .tm et faire :
Éditer→copier vers TeXmacs Scheme
- ensuite, dans my-init-texmacs, faire Éditer→coller depuis TeXmacs Scheme
- ou parfois : Éditer→coller depuis Verbatim
- tout raccourci clavier doit être dans un :

```
(delayed (lazy-keyboard-force)
  (kbd-map
    (:mode in-math?)
    (("a b c")....ici....)
  ))
```

→ mais la difficulté est de savoir dans le «ici» s'il faut mettre (insert '()) ou (make) etc.

1. mode texte : lettres répétées

triples

e e e	p p p	r r r	R R R	s s s					
a)	NOM	code R	bloc R	ssi					

quadruples

= = = =									
<i>ligne fine</i>									

2. mode maths : lettres répétées

doubles marchent pas d d puis g g

doubles

entre [crochets] : explication de pourquoi il faut laisser libre le raccourci

a a	b b	b b →	c c	e e	f f	i i	l l	m m	n n
dessus au	en bas	$\binom{n}{k}$	[arccos]	e xpo	frac- -tion	i	$\lim_{x \rightarrow +\infty} f(x)$	$()$	f
p p	p p →	r r	r r →	s s	t t	v v	x x		
	Π	√	$\sqrt[3]{x}$	Σ	√texte	\vec{u}	×		

doubles encore libres : h, j, k, o, u, y, z

triples

entre [crochets] : explication de pourquoi il faut laisser libre le raccourci

						i i i	i i i →		
						f	∫		
						v v v			
						[v]			

+ toutes les majuscules, en double : A...Z ou en triple : A...X.

3. mode maths : symboles divers

Un certain nombre de dx avec les combinaisons d x puis :

= →	= =	= → →	= → → →	< -	< - →	&	& →	& → →	
↔	↔	⇒	≡	< -	←	∩	∧	∩ truc	
						%	% →	% → →	% → → →
						∪	∨	∪ truc	%
A →	A → →	C →	E →			%	% →	% → →	% → → →
∇	ℵ	C_n^p	∃			∪	∨	∪ truc	%
I →	R → →	T →							
ℑ	℔	⊥							
@1	@6	@k	@c	@p	@t	@@1 →	@@9 →		
□	⊠	♥	♦	♠	♣	②	⑨		

(le dernier tableau ne marche qu'en thème Styx.)

4. mode maths A Z : codes un peu plus longs

- AZS : displaystyle ;
- AZR : systèmes : {
- AZE : petit tableau, voir paragraphe 6 ;
- divers :

AZD : division euclidienne posée :

$$\begin{array}{r|l} 9 & 81 \\ 8 & \\ 1 & 7 \end{array}$$

AZX : systèmes paramétriques de droites dans \mathbb{R}^3

$$\begin{cases} x = \\ y = \\ z = \end{cases}$$

5. Exports automatisé

trois raccourcis :

- ⌘E+P : exporter en pdf avec le même nom dans le même dossier ;

- **E+H** : exporter en html avec le même nom dans le même dossier ;
- **E+A** : les deux

6. Mode texte AZ : gros codes

- **AZE** : petit tableau (marche aussi en mode maths) :

1	2	3
---	---	---
- **AZA** : cadre bordure fine largeur 1par ;

--

- **AZD** : dessin libre **AZ** : cercle trigo éditable **AZF** : petite figure

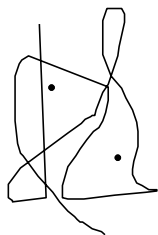


Figure 1.

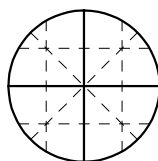


Figure 2.

Figure 3.

- **AZ0** : tableau de variations aucun extremum ;

x	
$f'(x)$	
f	

- **AZ1** : un extremum ;

x	
$f'(x)$	
f	

- **AZ2** : deux extremums ;

x	
$f'(x)$	
f	

- **AZC** : tableau de convexité ;

x	$-\infty$	$+\infty$
$f''(x)$		
	concave	infl convexe

- **AZP** : arbre de probabilités ;

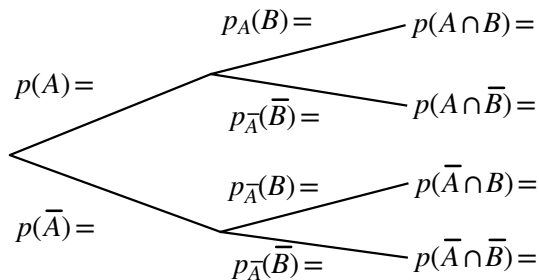


Figure 4.

- `AZ@` : lien URL cliquable en taille petit : www.liencliquable_en_petit.com
- `AZR` : «plié» :
 - reponses (cliquer sur ◦)
- `AZT` : tableau invisible largeur 1par et 2 colonnes ;
- `AZI` : no-indent ;
- `AZM` : coller depuis le presse-papier une image à la taille 5cm ;
→ très très utile car le raccourci habituel `⌘P` colle par défaut à la taille 1par.
- `AZB` : `BINGO` pour donner un sourire après un long calcul...
- `AZQ` : cercle trigo

7. Explications du code SCHEME

(lazy-keyboard-force)

`TEXMACS` loads the definitions in its own files when it needs them, not at startup---

you can look it up in Section 1.4 of the Scheme developer manual : *The module system and lazy definitions*.

→ where to find it ?

If you define in your files something that also `TEXMACS` defines, `TEXMACS` definition might be executed *after* your definition and overwrite it. Because of this, you force the loading of the `TEXMACS` definitions before your own, then you overwrite it with your own, and in this way `TEXMACS` will not execute its own anymore and will not overwrite yours.

(delayed) I do not know :-)

(let)

`let` is a construct which defines a *scope*, that is a portion of the program where some variables take values that are valid in that portion only.

In `(let ((img (tree-search-upwards (cursor-tree) 'image)))` one defines `img` as the value of `(tree-search-upwards (cursor-tree) 'image)`, which I think (I did not check) is the closest tree, searching upwards from the cursor, whose tag is `image`—since you just copied an image, `TeXmacs` will have created one (this is again guessing, but the function works, so it may be good guessing) and you will find the one `TeXmacs` just created. Now that you hold the tree, you have to change its leaves, and you do that with the `tree-set!` instructions.

The exclamation mark in the name of the function is a hint that the function will change already-existing variables—naming functions in this way is a custom of Scheme; the ones without exclamation mark according to the custom do not change any already-defined variable. One can violate the custom and if they do so may confuse someone else.

In `(let ((q "0.866")))` : the structure is a *quoted* expression, which remains equal to itself after evaluation. So `TEXMACS` never sees the value of `q`, sees the symbol `q` and does not know what to do with it.

You have to let Scheme know that you want to evaluate `q`. You do it in the following way:

- you substitute the quote with a `quasiquote`; in this way you let Scheme know that inside the `quasiquoted` expression there might be parts which Scheme has to evaluate. The `quasiquote` symbol is the backtick

`quasiquote ``

- you prepend an `unquote` operator to the expressions that you want Scheme to evaluate; the `unquote` symbol is the comma

`unquote ,`