

# ProfLycee (*simple doc*)

## User's Guide for L<sup>A</sup>T<sub>E</sub>X (v4.00a)

Cédric Pierquet  
cpierquet@outlook.fr

12/12/2025

The purpose of this simplified documentation is to present the main macros of ProfLycee, including:

- available names (base / French alias / English alias) if available;
- available keys (French / English);
- arguments (optional or not).

The general presentation of this *short* documentation is:

```
Short explanation | Courte explication [Syntax]
\macro(*) [opt arg]{arg}
% --- [keys/clés] -----
% key1[en] = key1[fr] = default
% key2[en] = key2[fr] = default
% ...
% -----
```

```
Short explanation | Courte explication [Sample(s)]
\LaTeX
-----
LATEX
```

For further details (especially regarding keys), the French documentation provides additional explanations and examples.

As the commands and keys are currently being anglicized, some may not be included in this documentation. Please note that the vast majority of ProfLycee commands are already compatible with both English and French.

# Contents

<b>1</b>	<b>Loading, global usage</b>	<b>3</b>
1.1	Loading . . . . .	3
1.2	Global usage . . . . .	3
<b>2</b>	<b>Core mathematical utilities</b>	<b>4</b>
2.1	Main macros . . . . .	4
2.2	Simple conversion between basis . . . . .	4
<b>3</b>	<b>Analysis</b>	<b>5</b>
3.1	Functions . . . . .	5
3.2	Sequences . . . . .	7
3.3	Integrals . . . . .	8
3.4	Signs, convexity . . . . .	9
<b>4</b>	<b>Code presentation</b>	<b>11</b>
4.1	Python Code . . . . .	11
4.2	Pseudo-Code . . . . .	14
<b>5</b>	<b>Statistics, probabilities</b>	<b>15</b>
5.1	Statistics . . . . .	15
5.2	Probabilities . . . . .	15
5.3	Random . . . . .	16
<b>6</b>	<b>Arithmetic</b>	<b>19</b>
6.1	Conversions . . . . .	19
6.2	Prime numbers, PGCD . . . . .	19
6.3	Divisors . . . . .	20
6.4	Miscellaneous . . . . .	21
<b>7</b>	<b>Recreations</b>	<b>23</b>
7.1	Fractals . . . . .	23
7.2	House of cards . . . . .	23
7.3	Stack of balls . . . . .	24

# 1 Loading, global usage

## 1.1 Loading

Proflycee can be loaded with:

```
🇺🇸 Loading | 🇫🇷 Chargement [Syntax]
\usepackage[options]{ProfLycee}
\usepackage[options]{ProfLycee-Light}

% --- {options} -----
% xcolor := load xcolor with [table,svgnames]
% nosiunitxfr = nonsiunitxfr := do not load sinuitx with french parameters
% noamssymb = nonamssymb := do not load amssymb
% warningsoff = nonwarnings := suppress warnings for unicode-math
% notikzbabel = nontikzbabel := do not load tikzbabel library
% nocancel = noncancel := do not load cancel
% nofa = nonfa := do not load fontawesome (fa5 default loaded)
% fa6 := load fa6
% fa7 := load fa7
% french := load french version (activated by default)
% english := load english version
% german := load german version
% spanish := load spanish version
% -----
```

## 1.2 Global usage

Modules (libraries) can be loaded in addition to main package:

```
🇺🇸 Libraries | 🇫🇷 Librairies [Syntax]
\useproflyclib{lib}

% --- {lib for full} -----
% piton
% space = espace
% mathwriting = ecritures
% -----

% --- {lib for light} -----
% analysis = analyse
% listings
% probas
% stats
% arithm
% random = aleatoire
% seq = suites
% space = espace
% mathwriting = ecritures
% piton
% -----
```

Language can be loaded with:

```
🇺🇸 Language | 🇫🇷 Langue [Syntax]
\setproflyclng{lng}

% --- {lng} -----
% fr / en / de / es
% -----
```

## 2 Core mathematical utilities

### 2.1 Main macros

Core calculus | Calculs de base

[Syntax]

```
\pflnum(*) [prec]{expr}
\pflnumfrac[format]{expr}

% --- (starred) -----
% * for real expr
%
% --- [format] -----
% for math output (d/n/t)
% -----
```

Core calculus | Calculs de base

[Sample(s)]

```
\pflnum{2^12+1} |
\pflnum*{4096/(12+1)} |
\pflnum*[4]{4096/(12+1)} |
\pflnum*[5]{sqrt(1+exp(9.75))} |
\pflnumfrac{74+1/5} \pflnumfrac[d]{74+1/5}
```

4097 | 315.076923076923 1 | 315.0769 | 130.97797 |  $\frac{371}{5}$   $\frac{371}{5}$

Round | Arrondi

[Syntax]

```
\pflround(*) [prec]{expr}
\pflarrondi(*) [prec]{expr}

% --- (starred) -----
% * for display sign
% -----
```

Round | Arrondi

[Sample(s)]

```
$1+\dfrac{7}{11} \approx \pflround{1+7/11} \approx \pflround[5]{1+7/11} \approx
\pflround*[2]{1+7/11}$

$\ln\big(1+e^4\big) \approx \pflround[6]{\log(1+exp(4))}$
```

$1 + \frac{7}{11} \approx 1.636 \approx 1.63636 \approx +1.64$   
 $\ln(1 + e^4) \approx 4.018150$

### 2.2 Simple conversion between basis

Conversion between basis | Conversion entre bases

[Syntax]

```
\pflbasetobase{init basis}{final basis}{number}
\pflbasetobase{base départ}{base arrivée}{nombre}
```

Conversion between basis | Conversion entre bases

[Sample(s)]

```
\pflbasetobase{10}{2}{27} | % 10 -> 2
\pflbasetobase{10}{16}{456} | % 10 -> 16
\pflbasetobase{2}{16}{10011111} | % 2 -> 16
\pflbasetobase{16}{7}{AB5F} | % 16 -> 7
```

11011 | 1C8 | 9F | 241622

### 3 Analysis

#### 3.1 Functions

🇺🇸 Determine Maximum | 🇫🇷 Détermine Max [Syntax]

```
\pfldetmax[round]{fct}{init}{step}{final}[\tmpmax][\tmpmaxvalx]
\DetermineMax[arrondi]{fct}{init}{pas}{final}[\tmpmax][\tmpmaxvalx]
```

🇺🇸 Determine Maximum | 🇫🇷 Détermine Max [Sample(s)]

```
\pfldetmax{80*x*exp(-0.2*x)}{3}{10}\tmpmax in \tmpmaxvalx
```

---

147.1517764685769 in 5

🇺🇸 Determine Minimum | 🇫🇷 Détermine Min [Syntax]

```
\pfldetmin[round]{fct}{init}{step}{final}[\tmpmax][\tmpmaxvalx]
\DetermineMin[arrondi]{fct}{init}{pas}{final}[\tmpmax][\tmpmaxvalx]
```

🇺🇸 Determine Minimum | 🇫🇷 Détermine Min [Sample(s)]

```
\pfldetmin[0.001]%
{log(exp(-x)+1)+0.25*x}{1}{1.2}%
[\mymin]%
[\myminx]%
\mymin in \myminx
```

---

0.5623351587104050 in 1.099

🇺🇸 Approx sol f(x)=k | 🇫🇷 Résol approach f(x)=k [Syntax]

```
\pflapproxsol[<keys>]{f(x)=k}[macro]
\ResolutionApprochee[<clés>]{f(x)=k}[macro]
\pflresolapproch[<clés>]{f(x)=k}[macro]
```

```
% --- [keys/clés] -----
% prec      = Precision    = 2
% var       = Variable     = x
% interval  = Intervalle   = {0:10}
%
% --- [macro] -----
% \macrod := round by default
% \macroe := round by excess
% \macroa := round
% -----
```

🇺🇸 Approx sol f(x)=k | 🇫🇷 Résol approach f(x)=k [Sample(s)]

```
\pflapproxsol[var=t,interval=-1:2,prec=4]{3*t*exp(-0.5*t+1)=5.365}[SolA]%
$t_1 \approx \num[minimum-decimal-digits=4]{\SolAd}$\\
$t_1 \approx \num[minimum-decimal-digits=4]{\SolAe}$\\
$t_1 \approx \num[minimum-decimal-digits=4]{\SolAa}$
```

---

$t_1 \approx 1.1968$   
 $t_1 \approx 1.1969$   
 $t_1 \approx 1.1969$

```
\pflivtsol[<clés>]{f(x)}{k}
\SolutionTVI[<clés>]{f(x)}{k}
\pflsoltvi[<clés>]{f(x)}{k}
```

```
% --- [keys/clés] -----
% fct name   = NomFct      = f
% prec      = Precision   = 2
% sol name   = NomSol      = \alpha
% stretch   = Stretch     = 1.15
% sweep      = Balayage    = false
% calculator = Calculatrice = false
% capital    = Majuscule   = true
% -----
```

```
\pflivtsol[va=1.414,vb=1.415,prec=5]{x**2-2}{0}
```

$$\begin{cases} f(1.414) \approx -0.00060 < 0 \\ f(1.415) \approx 0.00223 > 0 \end{cases} \Rightarrow 1.414 < \alpha < 1.415$$

```
\pflcanonicform(*)[format]{a}{b}{c}
\pflformcanoniq(*)[format]{a}{b}{c}
\FormeCanonique(*)[format]{a}{b}{c}
```

```
% --- (starred) -----
% * for minus sign
%
% --- [format] -----
% for math output (d/n/t)
% -----
```

```
$2x^2+4x-6 = \pflcanonicform{2}{4}{-6} = \pflcanonicform*{2}{4}{-6}$
```

$$2x^2 + 4x - 6 = 2(x + 1)^2 - 8 = 2(x - (-1))^2 - 8$$

```
\pflhomogfct(*)[format]{a}{b}{c}
\pflfcthomogr(*)[format]{a}{b}{c}
\FonctionHomographique(*)[format]{a}{b}{c}
```

```
% --- (starred) -----
% * for minus sign
%
% --- [format] -----
% for math output (d/n/t)
% -----
```

```
$\dfrac{11}{3x+6} = \pflhomogfct{0}{11}{3}{6} = \pflhomogfct*{0}{11}{3}{6}$
```

$$\frac{11}{3x+6} = \frac{\frac{11}{3}}{x+2} = \frac{\frac{11}{3}}{x-(-2)}$$

## 3.2 Sequences

🇫🇷 Calc recurrence | 🇫🇷 Calcul terme récurrence

[Syntax]

```
\pflcalcrecurr[<keys>]{fct}
\CalculTermeRecurrence[<clés>]{fct}

% --- [keys/clés] -----
% first n = No
% first un = UNo
% prec = Precision = 3
% final n = N
% -----
```

🇫🇷 Calc recurrence | 🇫🇷 Calcul terme récurrence

[Sample(s)]

```
\pflcalcrecurr[prec=6,first n=0,first un=50,final n=20]{1/(x+2)}
```

0.414214

🇫🇷 Threshold | 🇫🇷 Seuil

[Syntax]

```
\pflsolthreshold[<keys>]{fct}{val}
\SolutionSeuil[<clés>]{fct}{val}
\pflsolseuil[<clés>]{fct}{val}

% --- [keys/clés] -----
% prec = Precision = 3
% stretch = Stretch = 1.15
% sweeping = Balayage = false
% calc = Calculatrice = false
% capital = Majuscule = true
% sign = Sens = {>}
% exact = Exact = false
% exactB = ExactB = false
% conclusion = Conclusion = true
% simple = Simple = false
% -----
```

🇫🇷 Threshold | 🇫🇷 Seuil

[Sample(s)]

```
\pflsolthreshold[prec=4,first n=1,first un=2]{1+(1+x**2)/(1+x)}{10}
```

$$\begin{cases} u_{28} \approx 9.9408 \leq 10 \\ u_{29} \approx 10.1236 > 10 \end{cases} \Rightarrow n \geq 29$$

### 3.3 Integrals

🇺🇸 Integral approx | 🇫🇷 Approch intégrale

[Syntax]

```
\pflapproxintegral[<keys>]{fct}{a}{b}
\IntegraleApprochee[<clés>]{fct}{a}{b}
\pflintegrapproch[<clés>]{fct}{a}{b}

% --- [keys/clés] -----
% prec          = Precision      = 3
% nb subdiv     = NbSubDiv       = 10
% method        = Methode        = Simpson
% raw result    = ResultatBrut   = false
% disp formula  = AffFormule     = false
% sign          = Signe          = {\approx}
% expr          = Expr           = f(x)
% var           = Variable       = x
% -----
```

🇺🇸 Integral approx | 🇫🇷 Approch intégrale

[Sample(s)]

```
\pflapproxintegral[raw result]{sqrt(x)}{4}{10}
```

15.74851726347158

🇺🇸 Raw value of integral | 🇫🇷 Valeur 'brute' d'une intégrale

[Syntax]

```
\pflcalintegral(*)[prec]<num of subdiv>{fct}{a}{b}
\CalcIntegrale(*)[prec]<num of subdiv>{fct}{a}{b}
\pflvalintegr(*)[prec]<nb de subdiv>{fct}{a}{b}

% --- (starred) -----
% * for siunitx output
% -----
```

🇺🇸 Raw value of integral | 🇫🇷 Valeur 'brute' d'une intégrale

[Sample(s)]

```
\CalcIntegrale{80*x*exp(-0.2*x)}{1}{20}
```

1781.797364979251

🇺🇸 Average value | 🇫🇷 Valeur moyenne

[Syntax]

```
\pflaveragevalintegral(*)[prec]<num of subdiv>{fct}{a}{b}
\pflvalmoyintegr(*)[prec]<nb de subdiv>{fct}{a}{b}

% --- (starred) -----
% * for siunitx output
% -----
```



### 3.4 Signs, convexity

sign diagram | schémas signes

[Syntax]

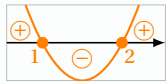
```
\pflminisignplot* [<keys>] <tikz options>
\pflschemasignes* [<clés>] <options tikz>
\MiniSchemaSignes* [<clés>] <options tikz>

% --- [keys/clés] -----
% code = Code = da+
% color = Couleur = red
% width = Largeur = 2
% height = Hauteur = 1
% border = Cadre = true
% roots = Racines = 2
% -----
```

sign diagram | schémas signes

[Sample(s)]

```
\pflminisignplot* [code=pa+d+, roots={1/2}, color=orange]
```



sign diagram with tkztab | schémas signes avec tkztab

[Syntax]

```
\pflminisignplottkztab [<keys>] {line} [scale] [hoffset]
\pflschemasignestkztab [<clés>] {line} [scale] [hoffset]
\MiniSchemaSignesTkzTab [<clés>] {ligne} [echelle] [décal H]

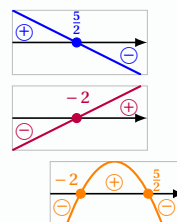
% --- [keys/clés] -----
% code = Code = da+
% color = Couleur = red
% width = Largeur = 2
% height = Hauteur = 1
% border = Cadre = true
% roots = Racines = 2
% -----
```

sign diagram with tkztab | schémas signes avec tkztab

[Sample(s)]

```
\begin{tikzpicture}
\tkzTabInit[espc1=2]%
  {$x$/1,$-2x+5$/1,$2x+4$/1,$p(x)$/1}{-$\infty$, $-2$, $2.5$, $+\infty$}
\tkzTabLine{+,t,+,z,-,}
\tkzTabLine{-,z,+,t,+,}
\tkzTabLine{-,z,+,z,-,}
\pflminisignplottkztab[code=da-, roots={\tfrac{5}{2}}, color=blue]{1}
\pflminisignplottkztab[code=da+, roots=-2, color=purple]{2}
\pflminisignplottkztab[code=pa-d+, roots=-2/{\tfrac{5}{2}}, color=orange]%
  {3}[0.85][2]
\end{tikzpicture}
```

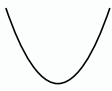



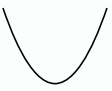
$x$	$-\infty$	$-2$	$2.5$	$+\infty$	
$-2x + 5$	+	+	0	-	
$2x + 4$	-	0	+	+	
$p(x)$	-	0	+	0	-



```
\pfltkztabcvx(*){line}{'cells'}
\tkzTabLineConvex(*){ligne}{liste 'cases'}

% --- (starred) -----
% * for text version
% -----
```

```
\begin{tikzpicture}
\tkzTabInit[lgt=3,espcl=2.5]%
  {$x/1,$f''(x)/1,Convexity\of $f$/2}{$0$,$1$,$3$,$5$}
\tkzTabLine{+,z,-,z,+}
\pfltkztabcvx{2}{,cvx,i*,ccv,i*,cvx,}
\end{tikzpicture}
```

$x$	0	1	3	5	
$f''(x)$	+	0	-	0	+
Convexity of $f$					

## 4 Code presentation

### 4.1 Python Code

environment python code listings | environnement code python listings [Syntax]

```
\begin{pflpythoncode}[<keys>]{tcbbox options} ... \end{pflpythoncode}
\begin{CodePythonLst}[<clés>]{options tcbbox} ... \end{CodePythonLst}

% --- (starred) -----
% * for removing line numbers
%
% --- [keys/clés] -----
% width      = Largeur      = \linewidth
% first line = PremLigne   = 1
% hsep       = EspaceNum   = 14pt
% -----
```

environment python code listings | environnement code python listings [Sample(s)]

```
\begin{pflpythoncode}[width=0.75\linewidth]{flush right}
nb = int(input("n = "))
if (nb %7 == 0) :
    print(f"{nb} is ok")
#endif

def f(x) :
    return x**2
\end{pflpythoncode}
```

Code Python

```
nb = int(input("n = "))
if (nb %7 == 0) :
    print(f"{nb} is ok")
#endif

def f(x) :
    return x**2
```

environment python code piton | environnement code python piton [Syntax]

```
\begin{pflpitoncode}[<keys>]{tcbbox options} ... \end{pflpitoncode}
\begin{CodePiton}[<clés>]{options tcbbox} ... \end{CodePiton}
\pflpitoncodefile[<keys>]{tcbbox options}
\CodePitonFichier[<clés>]{options tcbbox}

% --- [keys/clés] -----
% width      = Largeur      = \linewidth
% align      = Alignement   = center
% lines      = Lignes       = true
% gobble     = Gobble       = {}
% font size  = TaillePolice = \footnotesize
% watermark  = Filigrane    = false
% style      = Style        = classic/modern/(light)carbon/tab
% border     = Cadre        = true
% title bar  = BarreTitre   = true
% title      = Titre        = ...
% usep      = EspacementV   = 0.5\baselineskip
% num color  = CouleurNombres = {orange!75!black}
% color      = Couleur      = red
% carbon icons = IconesCarbon = ...
% -----
```

environment python code piton | environnement code python piton [Sample(s)]

```
\begin{pflpitoncode}%  
  [gobble=tabs,width=12cm,style=lightcarbon]{}  
def f(x) :  
  return x**2  
\end{pflpitoncode}
```



</> Python code

```
1 def f(x) :  
2   return x**2
```

console python piton | console python piton [Syntax]

```
\begin{pflpitonconsole}[<keys>]{tcbbox options} ... \end{pflpitonconsole}  
\begin{ConsolePiton}[<clés>]{options tcbbox} ... \end{ConsolePiton}
```

```
% --- [keys/clés] -----  
% logo           = Logo           = true  
% width          = Largeur          = \linewidth  
% align          = Alignement        = {flush left}  
% num color      = CouleurNombres    = {orange!75!black}  
% beforeafter sep = CoeffAvantApres = 0.75  
% -----
```

console python piton | console python piton [Sample(s)]

```
\begin{pflpitonconsole}{}  
1+1  
2**10  
\end{pflpitonconsole}
```

Start of Python console

```
>>> 1+1  
2  
>>> 2**10  
1024
```

End of Python console

thonny piton | thonny piton [Syntax]

```
\begin{pflpitonthonnyeditor}[<keys>]{...}\end{pflpitonthonnyeditor}  
\begin{PitonThonnyEditor}[<clés>]{...}\end{PitonThonnyEditor}  
\pflpitonthonnyeditorfile[<keys>]{...}  
\PitonThonnyEditorFichier[<clés>]{...}
```

```
% --- [keys/clés] -----  
% filename       = NomFichier        = {script.py}  
% consolename    = NomConsole         = console  
% gobble         = Gobble              = {}  
% num color      = CouleurNombres     = {orange!75!black}  
% width          = Largeur             = \linewidth  
% -----
```

```
\begin{pflpitonthonyeditor}[NomFichier=tpcapytale.py,Largeur=12cm]{}
nb = int(input("n = "))
if (nb %7 == 0) :
    print(f"{nb} is ok")
#endif

def f(x) :
    return x**2
\end{pflpitonthonyeditor}
```

tpcapytale.py ×

```
1 nb = int(input("n = "))
2 if (nb %7 == 0) :
3     print(f"{nb} is ok")
4 #endif
5
6 def f(x) :
7     return x**2
```

```
\begin{pflpitonthonyconsole}<keys>{tcbbox} ... \end{pflpitonthonyconsole}
\begin{PitonThonyConsole}<clés>{tcbbox} ... \end{PitonThonyConsole}

% --- [keys/clés] -----
% filename      = NomFichier      = {script.py}
% consolename   = NomConsole      = console
% console intro = IntroConsole    = {Python 3.11.6 /usr/bin/python}
% gobble        = Gobble          = {}
% num color     = CouleurNombres  = {orange!75!black}
% width         = Largeur         = \linewidth
% -----
```

```
\begin{pflpitonthonyconsole}[consolename={python 3.8.10},width=12cm]{}
#Run script.py
from random import randint
nb1 = randint(1,100000)
nb2= randint(1,100000)
print(nb1, nb2)
\end{pflpitonthonyconsole}
```

python 3.8.10 ×

```
Python 3.11.6 /usr/bin/python
>>> #Run script.py
>>> from random import randint
>>> nb1 = randint(1,100000)
>>> nb2= randint(1,100000)
>>> print(nb1, nb2)
76141 86108
```

## 4.2 Pseudo-Code

environment pseudocode piton | environnement pseudocode piton [Syntax]

```
\begin{pflpseudocodepiton}[<keys>]{tcbbox} ... \end{pflpseudocodepiton}
\begin{PseudoCodePiton}[<clés>]{tcbbox} ... \end{PseudoCodePiton}
\pflpitonpseudocodefile[<keys>]{tcbbox}
\PseudoCodePitonFichier[<clés>]{tcbbox}
```

```
% --- [keys/clés] -----
% width      = Largeur      = \linewidth
% align      = Alignement   = center
% lines      = Lignes       = true
% gobble     = Gobble       = {}
% font size  = TaillePolice = \footnotesize
% watermark  = Filigrane    = false
% style      = Style        = Classique
% border     = Cadre        = true
% title bar  = BarreTitre   = true
% title      = Titre        = ...
% usep       = EspacementV  = 0.5\baselineskip
% color      = Couleurs     = true
% -----
```

environment pseudocode piton | environnement pseudocode piton [Sample(s)]

```
%in french...
\begin{PseudoCodePiton}[Lignes=false,Filigrane=false,Couleurs=true]{f}
Algorithme : Périmètre de rectangles
Variables  : Long, Larg, Perim (réels)
            Choix (chaîne) # pour refaire ou non l'exécution
Début
# initialisation de l'indicateur pour entrer dans la boucle
Choix ← "o"
# boucle TantQue permettant de refaire le traitement selon le choix
TantQue Choix = "o" Faire
# traitement
Afficher("Donner les dimensions du rectangle") et Saisir(Long,Larg)
Perim ← 2 × (Long + Larg)
Afficher("Le périmètre du rectangle est", Perim)
#saisie du choix de recommencer ou non
Afficher("Voulez-vous continuer (o/n) ?") et Saisir(Choix)
FinTantQue
Fin
\end{PseudoCodePiton}
```

### Pseudocode

```
Algorithme : Périmètre de rectangles
Variables  : Long, Larg, Perim (réels)
            Choix (chaîne) # pour refaire ou non l'exécution
Début
# initialisation de l'indicateur pour entrer dans la boucle
Choix ← "o"
# boucle TantQue permettant de refaire le traitement selon le choix
TantQue Choix = "o" Faire
# traitement
Afficher("Donner les dimensions du rectangle") et \
↳ Saisir(Long,Larg)
Perim ← 2 × (Long + Larg)
Afficher("Le périmètre du rectangle est", Perim)
#saisie du choix de recommencer ou non
Afficher("Voulez-vous continuer (o/n) ?") et Saisir(Choix)
FinTantQue
Fin
```

## 5 Statistics, probabilities

### 5.1 Statistics

🇺🇸 Stats parameters | 🇫🇷 Paramètres stats [Syntax]

```
\DeterminerParamStats(*){liste}[\macromin]...[\macromax]
\pflparamstats(*){liste}[\macromin]...[\macromax]

% --- (starred) -----
% * for data with regroupment
% -----
```

🇺🇸 Stats parameters | 🇫🇷 Paramètres stats [Sample(s)]

```
\def\listedonnees%
{0.7426,0.7429,0.7435,0.7448,0.7456,0.7478,%
 0.7478,0.748,0.7487,0.7494,0.7494,0.7496}
\pflparamstats{\listedonnees}
\monmin/\monquartileun/\mamediane/\monquartiletrois/\monmax

0.7426/0.7435/0.7478/0.7487/0.7496
```

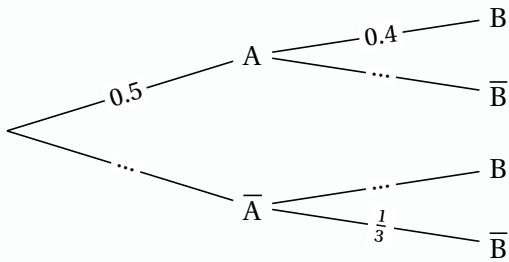
### 5.2 Probabilities

🇺🇸 Proba tree | 🇫🇷 Arbre de probas [Syntax]

```
\pflprobtrees[<keys>]{data}
\ArbreProbasTikz[<clés>]{donnees}
\pflarbreprobas[<clés>]{donnees}

% --- [keys/clés] -----
% unit = Unite = 1cm
% level sep = EspaceNiveau = 3.25
% child sep = EspaceFeuille = 1
% type = Type = 2x2
% font = Police = ...
% prob font = PoliceProbas = ...
% sloped = InclineProbas = true
% arrows = Fleche = false
% arc style = StyleTrait = {}
% arc thickness = EpaisseurTrait = semithick
% prob pos = PositionProbas = {}
% bg color = CouleurFond = white
% arc color = CouleurTraits = black
% prob color = CouleurProbas = black
% node color = CouleurNoeuds = black
% root = Racine = {}
% -----
```

```
\def\ArbreDeuxDeux{
  $$/\num{0.5}/,
  $$/\num{0.4}/,
  $$\overline{B}$$/.../,
  $$\overline{A}$$/.../,
  $$B$$/.../,
  $$\overline{B}$$/$\frac{1}{3}$$/
}
\pflprobtree{\ArbreDeuxDeux}
```



### 5.3 Random

```
\pflrandnum[n]{a}{b}{macro}
\NbAlea[n]{a}{b}{macro}
\pflnbalea[n]{a}{b}{macro}
```

```
\pflvarrandnum{macro}{calculs}
\VarNbAlea{macro}{calculs}
\pflvarnbalea{macro}{calculs}
```

```
Integer between 1 and 50: \pflrandnum{1}{50}{\FirstNbAlea}
\FirstNbAlea
```

```
\pflvarrandnum{\SecondNbAlea}{\FirstNbAlea+randint(0,10)}
\SecondNbAlea
```

```
Float between 0 and 10.999... : \pflrandnum[3]{0}{10}{\FirstFloatAlea}
\FirstFloatAlea
```

Integer between 1 and 50: 6

11

Float between 0 and 10.999... : 8.638



```
\pflrandintnumbers[<keys>]{macro}
\TirageAleatoireEntiers[<clés>]{macro}
\pfltiragealeaent[<clés>]{macro}
```

```
% --- [keys/clés] -----
% min value = ValMin      = 1
% max value = ValMax      = 50
% nb values = NbVal       = 6
% sep       = Sep         = {,}
% sort      = Tri         = non
% repeat    = Repetition = false
% -----
```

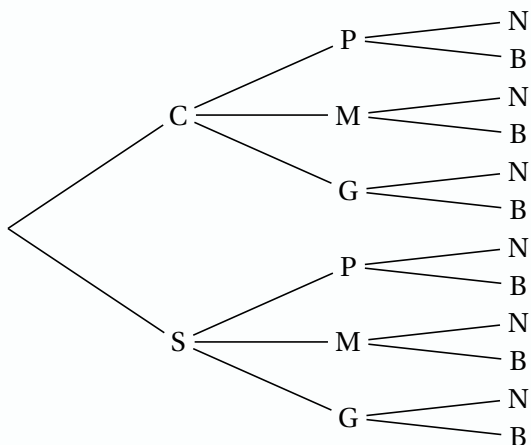
```
\pflrandintnumbers{\FirstListAlea}\FirstListAlea
```

13,49,6,9,48,20

```
\pflchoicetree[<keys>]<tikz options>{list}
\ArbreChoix[clés]<options tikz>{liste}
\pflarbrechoix[clés]<options tikz>{liste}
```

```
% --- [keys/clés] -----
% level sep = EspaceNiveaux = 2.25
% child sep = EspaceFeuilles = 0.5
% scale     = Echelle       = 1
% repeat    = Repet         = {}
% help      = Notice        = {}
% help rules = TraitsNotice = false
% level color = CouleursNiveaux = black
% show results = AffResultats = false
% result sep = SepResultats = {}
% -----
```

```
\pflchoicetree{C/S,P/M/G,N/B}
```



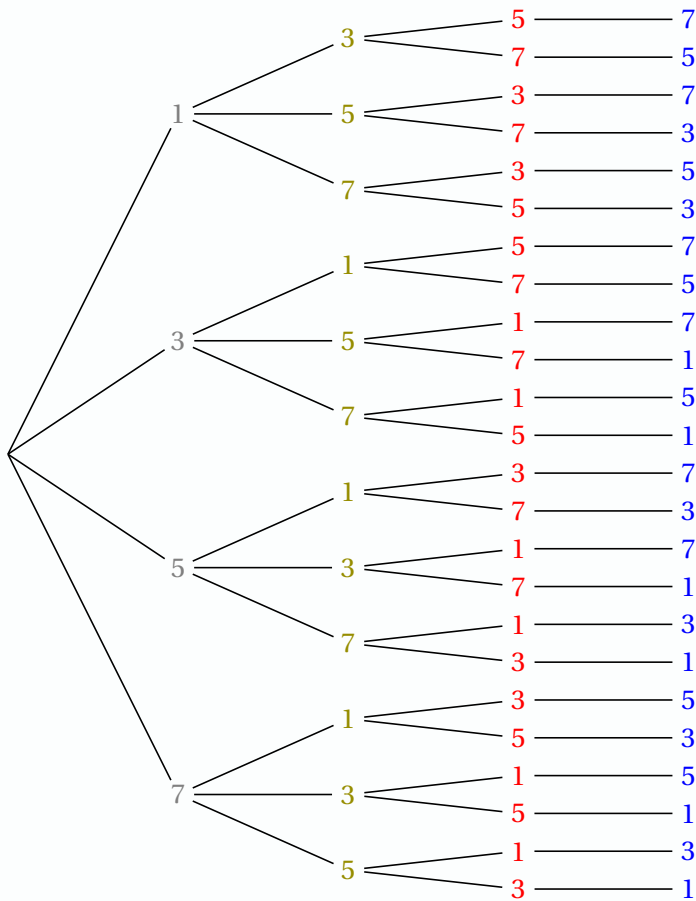
```

\pflchoicetreorepl<tikz options>{list}
\ArbreChoixSansRemise[clés]<options tikz>{liste}
\pflarbrechoixssremise[<clés>]<options tikz>{liste}

% --- [keys/clés] -----
% level sep      = EspaceNiveaux   = 2.25
% child sep      = EspaceFeuilles  = 0.5
% scale          = Echelle          = 1
% repeat         = Repet            = {}
% help           = Notice           = {}
% help rules     = TraitsNotice     = false
% level color    = CouleursNiveaux  = black
% show results   = AffResultats     = false
% result sep     = SepResultats     = {}
% -----
    
```

```

\pflchoicetreorepl[level color={gray,olive,red,blue}]{1,3,5,7}
    
```



## 6 Arithmetic

### 6.1 Conversions

🇺🇸 Conversion to dec | 🇫🇷 Conversion en b10

[Syntax]

```
\pflconvtodec[keys]{number}
\pflconvversdec[clés]{nb}
\ConversionVersDec[clés]{nb}
```

```
% --- [keys/clés] -----
% base from = BaseDep = 2
% show base = AffBase = true
% details   = Details = true
% zeros     = Zeros   = true
% -----
```

🇺🇸 Conversion to dec | 🇫🇷 Conversion en b10

[Sample(s)]

```
$\ConversionVersDec[zeros=false,base from=16]{AC0DC}$
```

$$AC0DC_{16} = 10 \times 16^4 + 12 \times 16^3 + 13 \times 16^1 + 12 \times 16^0 = 704732_{10}$$

🇺🇸 Conversion from dec | 🇫🇷 Conversion depuis b10

[Syntax]

```
\pflconvdepuisdix[clés]{nombre}{base arrivée}
\pflconvfromten[keys]{number}{final basis}
\ConversionDepuisBaseDix[clés]{nombre}{base arrivée}
```

```
% --- [keys/clés] -----
% color     = Couleur   = red
% hsep      = DecalH    = 2pt
% usep      = DecalV    = 3pt
% rect      = Rect      = true
% node      = Noeud     = EEE
% res color = CouleurRes = false
% -----
```

🇺🇸 Conversion from dec | 🇫🇷 Conversion depuis b10

[Sample(s)]

```
\ConversionDepuisBaseDix{78}{2}
```

$$\left\{ \begin{array}{l} 78 = 2 \times 39 + 0 \\ 39 = 2 \times 19 + 1 \\ 19 = 2 \times 9 + 1 \\ 9 = 2 \times 4 + 1 \\ 4 = 2 \times 2 + 0 \\ 2 = 2 \times 1 + 0 \\ 1 = 2 \times 0 + 1 \end{array} \right. \Rightarrow 78_{10} = 1001110_2$$

### 6.2 Prime numbers, PGCD

🇺🇸 Isprime ? | 🇫🇷 Est premier ?

[Syntax]

```
\pflisprimebool{expr}
\pflboolestpremier{expr}
```

🇺🇸 Isprime ? | 🇫🇷 Est premier ?

[Sample(s)]

```
\pflisprimebool{25457} and \pflisprimebool{25458}
```

1 and 0

🇺🇸 Number factor | 🇫🇷 Décompo facteurs premiers

[Syntax]

```
\pflnumberfact(*){expr}  
\pfldecompnb(*){expr}
```

```
% --- (starred) -----  
% * for 1 power  
% -----
```

🇺🇸 Number factor | 🇫🇷 Décompo facteurs premiers

[Sample(s)]

```
$24458=\pflnumberfact{24458}=\pflnumberfact*{24458}$
```

24458 = 2 × 7 × 1747 = 2<sup>1</sup> × 7<sup>1</sup> × 1747<sup>1</sup>

🇺🇸 Gcd presentation | 🇫🇷 présentation PGCD

[Syntax]

```
\pflprespgcd[keys]{a}{b}  
\PresentationPGCD[clés]{a}{b}
```

```
% --- [keys/clés] -----  
% color      = Couleur          = red  
% rect sep   = DecalRect        = 2pt  
% rect       = Rectangle        = true  
% node       = Noeud            = FFF  
% res color  = CouleurResultat  = false  
% disp ccl   = AfficheConclusion = true  
% disp delim = AfficheDelimiteurs = true  
% -----
```

🇺🇸 Gcd presentation | 🇫🇷 présentation PGCD

[Sample(s)]

```
\pflprespgcd[res color,color=CouleurVertForet]{1250}{450}.
```

$$\left\{ \begin{array}{l} 1250 = 450 \times 2 + 350 \\ 450 = 350 \times 1 + 100 \\ 350 = 100 \times 3 + \boxed{50} \\ 100 = 50 \times 2 + 0 \end{array} \right. \Rightarrow \text{PGCD}(1250; 450) = 50.$$

## 6.3 Divisors

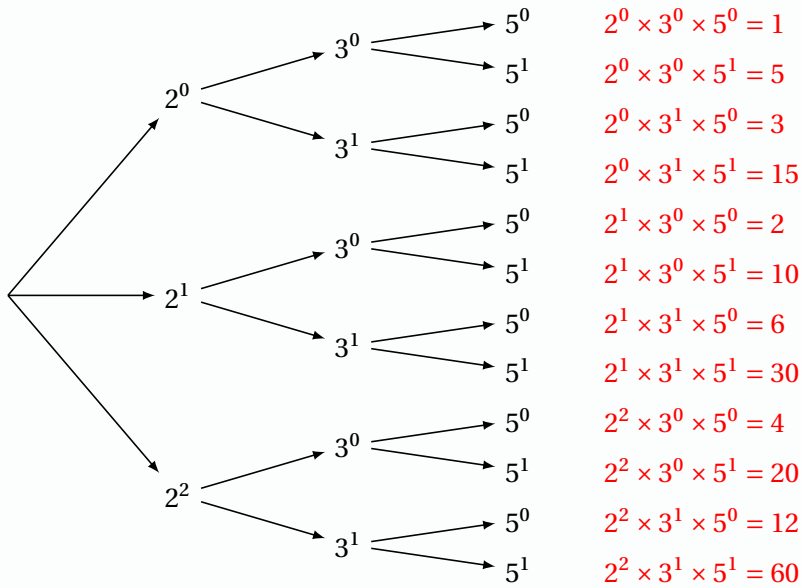
🇺🇸 Tree divisors | 🇫🇷 Arbre des diviseurs

[Syntax]

```
\pfltreediv[keys]<tikz options>{expr}  
\pflarbrediv[clés]<options tikz>{expr}  
\ArbreDiviseurs[clés]<options tikz>{expr}
```

```
% --- [keys/clés] -----  
% level sep  = EspaceNiveau    = 2.25  
% child sep  = EspaceFeuille   = 0.66  
% details    = Details         = true  
% details color = CouleurDetails = red  
% scale      = Echelle         = 1  
% arrows     = Fleches         = true  
% -----
```

`\pfltreediv{60}`



## 6.4 Miscellaneous

```
\pflperiodfraction[keys]{before period}{period}
\pflfracperiod[clés]{avant période}{période}
\FractionPeriode[clés]{avant période}{période}
```

```
% --- [keys/clés] -----
% statement = Enonce = true
% d         =         = true
% unknown   = Inconnue = x
% sol       = Solution = false
% raw       = Brut     = true
% simple    = Simple   = false
% -----
```

`\pflperiodfraction[sol]{45.1}{23}`

We set  $x = 45.1\overline{23}$ .

We *shift* the period close to the decimal point by multiplying by  $10^1$ :

$$\boxed{10^1 \times x = 451.\overline{23}} \quad (1)$$

We *shift* the period before the decimal point by multiplying equality (1) by  $10^2$ :

$$10^2 \times 10^1 \times x = 10^2 \times 451.\overline{23} \Rightarrow \boxed{10^3 \times x = 45123.\overline{23}} \quad (2)$$

We subtract the two equalities, (2) – (1), which allows us to *remove* the decimal part:

$$\begin{aligned} 10^3 \times x - 10^1 \times x &= 45123.\overline{23} - 451.\overline{23} \Rightarrow (10^3 - 10^1) \times x = 45123 - 451 \\ &\Rightarrow 990 \times x = 44672 \\ &\Rightarrow x = \frac{44672}{990} \end{aligned}$$

Thus we have  $45.1\overline{23} = \frac{44672}{990} = \frac{22336}{495}$ .

## 7 Recreations

### 7.1 Fractals

Fractals, with TikZ | Fractales, avec TikZ

[Syntax]

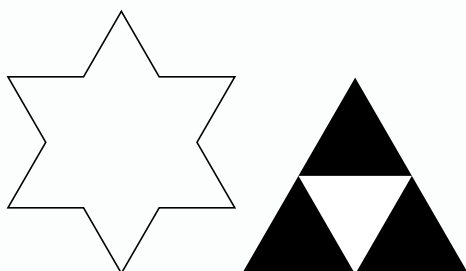
```
\pfltkzfractal[type=...,<keys>]<tikZ options>
\FractaleTikz[Type=...,<clés>]<options tikz>
\pflfractaltikz[Type=.,<clés>]<options tikz>

% --- [type] -----
% type      = Type          = Koch or Sierp
%
% --- [keys/clés] -----
% thickness = Epaisseur    = 0.6pt
% color     = Couleur      = black
% side width = LongueurCote = 3
% step      = Etape        = 1
% fill      = Remplir      = false
% fill style = Remplissage = lightgray
% start     = Depart       = {(0,0)}
% v align   = AlignV       = false
% offset    = Offset       = 2pt
% -----
```

Fractals, with TikZ | Fractales, avec TikZ

[Sample(s)]

```
\pfltkzfractal[type=Koch]
\pfltkzfractal[type=Sierp]
```



### 7.2 House of cards

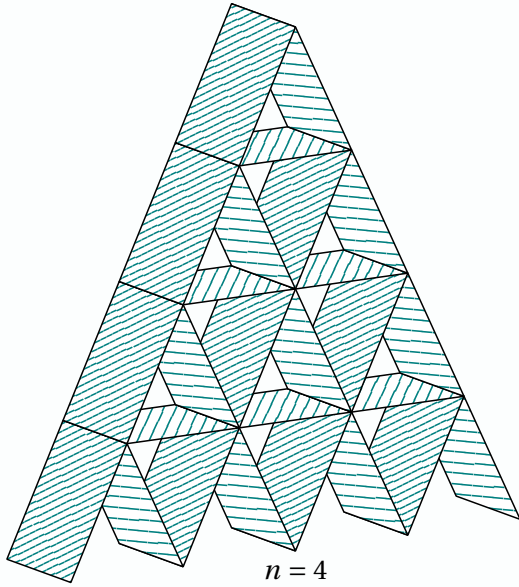
House of cards | Château de cartes

[Syntax]

```
\pflhouseofcards[keys]{levels}<tikz options>
\pflchateaucartes[clés]{étages}<options tikz>
\ChateauCartes[clés]{étages}<options tikz>

% --- [keys/clés] -----
% scale     = Echelle      = 1
% deco color = CouleurDeco = black
% rounded   = Arrondi      = true
% y angle   = AngleY       = 20
% x angle   = AngleX       = 8
% bottom    = Bas          = false
% legend    = Legende      = false
% legend font = PoliceLegende = \normalsize\normalfont
% deco      = Deco         = remplir
% -----
```

```
\pflhouseofcards[deco color=teal,deco=hatch,scale=2,legend]{4}
```



### 7.3 Stack of balls

```
\pflballstacking[keys]{levels}<tikz options>
\pflempilballes[clés]{niveaux}<options tikz>
\EmpilementBalles[clés]{niveaux}<options tikz>
```

```
% --- [keys/clés] -----
% color = Couleur = gray
% rotation = Rotation = -5
% scale = Echelle = 1
% -----
```

```
\pflballstacking[scale=0.33,color=red]{3-8}
```

