tVar

Init

```
require("tVar/init.lua")
```

EasyInput

```
tVar[[
--Define Var and print
a:=10:outRES()
--Do calc and print
b:=(a+10):print()
--Output LaTeX
# Variable %%b%% hast the value $$b$$
]]
tVar.intFile([string])
```

Global tVar.

```
numFormat = "%.3f"
mathEnviroment = "align"
debugMode = "off"
outputMode = "RES" --RES, RES_EQ, RES_EQ_N
numeration = true
decimalSeparator = "."
disableOutput = false
coloredOuput = false

(tMat|tVec).texStyle = "mathbf"|"vec"
tMat.eqTexAsMatrix = false
```

New

```
tVar:New(0.04,"r_{se}")
tVec:New({10,2,7},"v_{1}")
tMat:New({{10,2,5},{2,4,3},{7,4,3}},"a_{2}")
```

Output

```
:print()
:outRES_EQ_N(number[bool],enviroment[bool])
:outRES_EQ([bool],[bool])
:outRES([bool],[bool])
:out()
```

Set [tVar]

```
:setName([string])
:setUnit([string])
:clean(name[string])
```

Misc

```
[tVar]:bracR()
[tVar]:CRLF([string])
[tVar]:CRLFb([string])
[tVar]:copy()
tex.print([string])
```

Math

```
tVar.sqrt([tVar],[number])
tVar.PI
[tMat]:T()
[tMat]:Det()
[tMat]:Inv()
[tVec]:crossP()
```

```
Converted math functions:
[tVar].abs,[tVar].acos, [tVar].cos,
[tVar].cosh, [tVar].asin, [tVar].sin,
[tVar].sinh, [tVar].atan, [tVar].tan,
[tVar].tanh, [tVar].ceil, [tVar].floor,
[tVar].exp, [tVar].log, [tVar].log10,
[tVar].rad, [tVar].deg, [tVar].atan2
```

Plot

```
tPlot:New([tPlot]present)
[tPlot].add((fun|{{X1,Y1},{X2,Y2}}),title,st
yle) --gnuplot style
[tPlot].plot()
```