tVar

Init
```
require("../tVar.lua")
matrix = require("../matrix")
```

Global tVar.
```
numFormat = "%.3f"
mathEnviroment = "align"
debugMode = "off"
outputMode = "RES" --RES, RES_EQ, RES_EQ_N
numeration = true
```

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() --abh. v OutputMode
:outFull(number[bool],enviroment[bool])
:outHalf([bool],[bool])
:outVar([bool],[bool])
:out() --nur Wert
```

Set [tVar]
```
:setName([string])
:setUnit([string])
:clean(name[string]) --berechn. Schritte
entf.
Tipp: können verkettet werden
```

Misc
```
[tVar]:bracR() --Runde Klammern
[tVar]:CRLF([string]) --neuwline, [string]
wird vor und nach Umbruch eingefügt
```

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

Example
```
\begin{luacode*}
require("../tVar.lua")
matrix = require("../matrix")

numFormat = "%.2f"
outputMode = "RES_EQ_N"
numeration = false

sigma_x = tVar:New(11.4,"\\sigma_{x}")
tau_xy = tVar:New(2.43,"\\tau_{xy}")

sigma_v = tVar.sqrt(sigma_x^2+3*tau_xy^2,2)
sigma_v:setName("\\sigma_{v}")
sigma_v:setUnit("N/m^2")

sigma_v:print()
\end{luacode*}
```