```
   Checking for foreign routines
   FRICAS="/usr/local/lib/fricas/target/x86_64-linux-gnu"
   spad-lib="/usr/local/lib/fricas/target/x86_64-linux-gnu//lib/libspad.so"
   foreign routines found
   openServer result -2
                          FriCAS Computer Algebra System
              Version: FriCAS 2024-04-15 built with sbcl 2.2.9.debian
                     Timestamp: Di 28 Mai 2024 21:49:04 CEST
   ------------------------------------------------------------------------------
      Issue )copyright to view copyright notices.
      Issue )summary for a summary of useful system commands.
      Issue )quit to leave FriCAS and return to shell.
   ------------------------------------------------------------------------------

      Function declaration sixel : TexFormat -> Void has been added to
         workspace.
   Value = #<INTERPRETED-FUNCTION NIL {10020A026B}>
(6) -> quickLoad "tmspt"

   Value = |$inclAssertions|
   Value = T
   To load "tmspt":
     Load 1 ASDF system:
       tmspt
   ; Loading "tmspt"
      The current FriCAS default directory is
         /home/kfp/quicklisp/local-projects/spadlib/tmspt/lib
      Compiling FriCAS source code from file
         /home/kfp/quicklisp/local-projects/spadlib/tmspt/lib/../src/tmspt.spad
         using old system compiler.
      TMSPT abbreviates package TexmacsSupport
   ------------------------------------------------------------------------
      initializing NRLIB TMSPT for TexmacsSupport
      compiling into NRLIB TMSPT
      compiling local format : (String,String) -> String
(11) -> setTypeFontColor "red"

   "red"
                                                              Type: SExpression
(9) -> ioHook()

   Value = #<INTERPRETED-FUNCTION NIL {1005351C9B}>
                                                                 Type: Void
(12) -> Hello

   Hello
                                                      Type: Variable(Hello)
(13) -> setFricasPrompt("",". ")

   MKPROMPT
                                                              Type: SExpression
14. setTypeFontColor "darkgreen"
```

"darkgreen"

Type: SExpression

```
15. X+2
```

$X+2$

Type: Polynomial(Integer)

```
16. setTypeFontSize "small"
```

"small"

Type: SExpression

```
17. EXPR INT
```

Expression(Integer)

Type: Type

```
18. %
```

Expression(Integer)

Type: Type

```
19. (x+2)
```

$x+2$

Type: Polynomial(Integer)

```
22. D(x*sin(x^n),x)
```

$\sin(x^n)+n\,x\,x^{(n-1)}\cos(x^n)$

Type: Expression(Integer)

```
23. latex  %
```

"{sin left({{{x} sp {n}}} right)}+{n x {{x} sp {{n-1}}} {cos left({{{x} sp {n}}} right)}}"

Type: String

```
24. verbatim %
```

```
 {\sin \left( {{{x} \sp {n}}} \right)}+{n \  x \  {{x} \sp {{n -1}}}
  \  {\cos \left( {{{x} \sp {n}}} \right)}}
```

Type: Void

```
25. )sh TMSPT
```

```
 TexmacsSupport is a package constructor
 Abbreviation for TexmacsSupport is TMSPT
 This constructor is exposed in this frame.
------------------------------- Operations --------------------------------

 addDefaultStyle : () -> Void          addStyle : String -> Void
 changeZoomFactor : Float -> Void      check_iohook : () -> Boolean
 clearAllFields : () -> Void           closeSession : () -> Void
 command : String -> Void              createSubsession : () -> Void
 evaluateAbove : () -> Void            evaluateAll : () -> Void
 evaluateBelow : () -> Void            evaluateFieldsInOrder : () -> Void
 firstField : () -> Void               fitAllToScreen : () -> Void
 fitToScreen : () -> Void              fitToScreenHeight : () -> Void
 fitToScreenWidth : () -> Void         foldAllFields : () -> Void
```

```
html : String -> Void                    inlineImage : String -> Void
input : String -> Void                   insertFieldAbove : () -> Void
insertFieldBelow : () -> Void            insertTextFieldAbove : () -> Void
insertTextFieldBelow : () -> Void        ioHook : () -> Void
lastField : () -> Void                   latex : String -> Void
linkImage : String -> Void               lispEval : String -> SExpression
mathInput : String -> String             message : (String, String) -> Void
nextField : () -> Void                   output : String -> Void
previousField : () -> Void               prompt : String -> Void
ps : String -> Void                      removeBanner : () -> Void
removeDefaultStyle : () -> Void          removeLastField : () -> Void
removeNextField : () -> Void             removePreviousField : () -> Void
removeStyle : String -> Void             scheme : String -> Void
sessionEval : () -> Void                 showTimings : () -> Void
splitSession : () -> Void                tmActive? : () -> Boolean
toggleFooter : () -> Void                toggleFullScreenMode : () -> Void
toggleHeader : () -> Void                toggleMathInput : () -> Void
toggleMathOutput : () -> Void            toggleMultilineInput : () -> Void
toggleTreeOutput : () -> Void            unfoldAllFields : () -> Void
verbatim : String -> Void                zoomIn : Float -> Void
zoomOut : Float -> Void
setFricasPrompt : (String, String) -> SExpression
setTypeFontColor : String -> SExpression
setTypeFontSize : String -> SExpression
toggleSchemeTreeOutput : () -> Void
```

25. `inlineImage "/home/kfp/Pictures/fricas_logo.png"`

Type: Void

Todo: size … link does not work properly :(

56. `linkImage "/home/kfp/Pictures/fricas_logo.png"`

Type: Void

28. 

28. `tmActive?()`

```
Value = TMACTIVE
```

true

Type: Boolean

```
29. check_iohook()
```

true

Type: Boolean

```
31. mathInput "x+2"
```

"$x+2$"

Type: String

```
32. html "<a href=_"https://www.fricas.org_">This is a link</a>"
```

This is a link

Type: Void

```
33. html "<img src=_"/home/kfp/Pictures/fricas_logo.png_" alt=_"fricas.org_"
    width=_"104_" height=_"142_">"
```

Type: Void

```
34. html "<table>
      <tr>
        <th>Company</th>
        <th>Contact</th>
        <th>Country</th>
      </tr>
      <tr>
        <td>Alfreds Futterkiste</td>
        <td>Maria Anders</td>
        <td>Germany</td>
      </tr>
      <tr>
        <td>Centro comercial Moctezuma</td>
        <td>Francisco Chang</td>
        <td>Mexico</td>
      </tr>
    </table>"
```

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexic o

Type: Void

```
37. html "<h1 style=_"background-color:Blue;_">Hello World</h1>"
```

Hello World

Type: Void

```
38. html "<p style=_"background-color:Red;_">Lorem ipsum...</p>"
```

Lorem ipsum...

Type: Void

```
39. html "<ul>
      <li>Coffee</li>
      <li>Tea</li>
      <li>Milk</li>
    </ul>"
```

- Coffee
- Tea
- Milk

```
   Type: Void
```

```
40. html "document.getElementById(_"demo_").innerHTML = _"Hello JavaScript!_";"
```

document.getElementById("demo").innerHTML = “Hello JavaScript!";

```
   Type: Void
```

```
41. html "<code>
    x = 5;
    y = 6;
    z = x + y;
    </code>"
```

```
x = 5;y = 6;z = x + y;
```

```
   Type: Void
```

```
42. html "<p>The area of a triangle is: 1/2 x <var>b</var> x <var>h</var>, where
    <var>b</var> is the base, and <var>h</var> is the vertical height.</p>"
```

The area of a triangle is: 1/2 x b x h, where b is the base, and h is the vertical height.

```
   Type: Void
```

```
44. html "<p>will display &part; &sum;</p>"
```

will display $\partial \sum$

```
   Type: Void
```

```
48. html "<form>
      <label for=_"fname_">First name:</label><br>
      <input type=_"text_" id=_"fname_" name=_"fname_"><br>
      <label for=_"lname_">Last name:</label><br>
      <input type=_"text_" id=_"lname_" name=_"lname_">
    </form>"
```

```
   Type: Void
```

```
49. html "<canvas id=_"myCanvas_" width=_"200_" height=_"100_" style=_"border:1px
    solid #000000;_">
    </canvas>"
```

```
   Type: Void
```

```
51. scheme "(times (plus 123 456) 67)"
```

```
38793
```

```
   Type: Void
```

```
52. verbatim "How are you!"
```

```
How are you!
```

```
   Type: Void
```

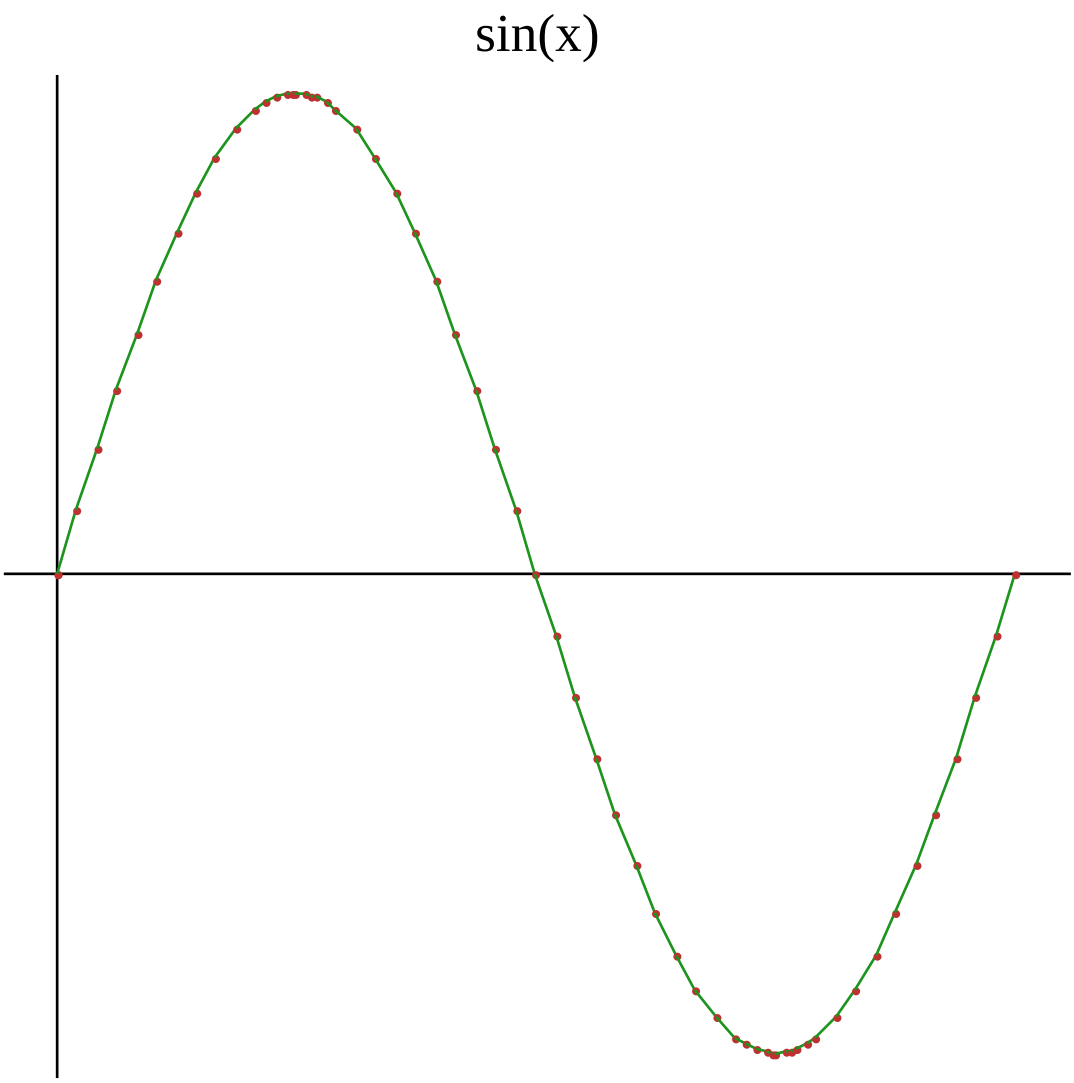
```
53. latex("This is a famous constant: \hbar")$TMSPT
```

```
This is a famous constant: ℏ
```

```
   Type: Void
```

```
58. inlineImage  "/home/kfp/Desktop/fricas2D.ps"
```

Type: Void



59. `draw(x*sin(x)^2,x=0..%pi)`

```
Compiling function     Graph data being transmitted to the viewport manager...
   FriCAS2D data being transmitted to the viewport manager...
```

TwoDimensionalViewport: "$x\sin(x)\hat{}2$"

Type: TwoDimensionalViewport

60. `)sh  TwoDimensionalViewport`

```
 TwoDimensionalViewport is a domain constructor.
 Abbreviation for TwoDimensionalViewport is VIEW2D
 This constructor is not exposed in this frame.
 31 Names for 37 Operations in this Domain.
------------------------------- Operations --------------------------------

 ?=? : (%, %) -> Boolean              close : % -> Void
 coerce : % -> OutputForm             key : % -> Integer
 latex : % -> String                  makeViewport2D : % -> %
 options : % -> List(DrawOption)      reset : % -> Void
 title : (%, String) -> Void          viewport2D : () -> %
```

```
 write : (%, String) -> String          ?~=? : (%, %) -> Boolean
 axes : (%, PositiveInteger, Palette) -> Void
 axes : (%, PositiveInteger, String) -> Void
 connect : (%, PositiveInteger, String) -> Void
 controlPanel : (%, String) -> Void
 dimensions : (%, NonNegativeInteger, NonNegativeInteger, PositiveInteger,
PositiveInteger) -> Void
 getGraph : (%, PositiveInteger) -> GraphImage
 getPickedPoints : % -> List(Point(DoubleFloat))
 graphState : (%, PositiveInteger, DoubleFloat, DoubleFloat, DoubleFloat,
DoubleFloat, Integer, Integer, Integer, Integer, Palette, Integer, Palette,
Integer) -> Void
 graphStates : % -> Vector(Record(scaleX: DoubleFloat,scaleY: DoubleFloat,
deltaX: DoubleFloat,deltaY: DoubleFloat,points: Integer,connect: Integer,
spline: Integer,axes: Integer,axesColor: Palette,units: Integer,unitsColor:
Palette,showing: Integer))
 graphs : % -> Vector(Union(GraphImage,undefined))
 makeViewport2D : (GraphImage, List(DrawOption)) -> %
 move : (%, NonNegativeInteger, NonNegativeInteger) -> Void
 options : (%, List(DrawOption)) -> %
 points : (%, PositiveInteger, String) -> Void
 putGraph : (%, GraphImage, PositiveInteger) -> Void
 region : (%, PositiveInteger, String) -> Void
 resize : (%, PositiveInteger, PositiveInteger) -> Void
 scale : (%, PositiveInteger, Float, Float) -> Void
 show : (%, PositiveInteger, String) -> Void
 translate : (%, PositiveInteger, Float, Float) -> Void
 units : (%, PositiveInteger, Palette) -> Void
 units : (%, PositiveInteger, String) -> Void
 update : (%, GraphImage, PositiveInteger) -> Void
 write : (%, String, List(String)) -> String
 write : (%, String, String) -> String
```

60. `%`

TwoDimensionalViewport: "$x \sin(x)\hat{}2$"
Type: TwoDimensionalViewport

61. `vp:=%`

TwoDimensionalViewport: "$x \sin(x)\hat{}2$"
Type: TwoDimensionalViewport

62. `getGraph(vp,1)`

Graph with 1 point list
Type: GraphImage

63. `)sh GraphImage`

```
 GraphImage is a domain constructor.
 Abbreviation for GraphImage is GRIMAGE
 This constructor is not exposed in this frame.
 15 Names for 22 Operations in this Domain.
------------------------------- Operations --------------------------------

 ?=? : (%, %) -> Boolean               coerce : % -> OutputForm
```

```
 graphImage : () -> %                    key : % -> Integer
 latex : % -> String                     sendGraphImage : % -> Void
 units : % -> List(Float)                ?~=? : (%, %) -> Boolean
 appendPoint : (%, Point(DoubleFloat)) -> Void
 coerce : List(List(Point(DoubleFloat))) -> %
 component : (%, List(Point(DoubleFloat)), Palette, Palette, PositiveInteger)
-> Void
 component : (%, Point(DoubleFloat)) -> Void
 component : (%, Point(DoubleFloat), Palette, Palette, PositiveInteger) ->
Void
 makeGraphImage : List(List(Point(DoubleFloat))) -> %
 makeGraphImage : (List(List(Point(DoubleFloat))), List(Palette),
List(Palette), List(PositiveInteger)) -> %
 makeGraphImage : (List(List(Point(DoubleFloat))), List(Palette),
List(Palette), List(PositiveInteger), List(DrawOption)) -> %
 point : (%, Point(DoubleFloat), Palette) -> Void
 pointLists : % -> List(List(Point(DoubleFloat)))
 putColorInfo : (List(List(Point(DoubleFloat))), List(Palette)) ->
List(List(Point(DoubleFloat)))
 ranges : % -> List(Segment(Float))
 ranges : (%, List(Segment(Float))) -> List(Segment(Float))
 units : (%, List(Float)) -> List(Float)
```

63. `g:=%`

Graph with 1 point list

Type: GraphImage

64. `sendGraphImage g`

```
>> Error detected within library code:
You are trying to draw over an existing graph
```

64. `show(vp,1,"Hello")`

```
>> Error detected within library code:
This viewport has already been closed!
```

64.