Pensieve header: This is the main notebook for the nb2tex project, containing both the documentation and the implementation.

To do

The phrase "∧ FreeQ[*c*, _Cell, {1, ∞}]" within the program seems unnecessary, and is presently commented out. Remove or justify!

Insert definitions for \nbpdfPrint and for \nbpdfText.

Experimentation

```
In[•]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\nb2tex"]
Out[•]=
C:\drorbn\AcademicPensieve\Projects\nb2tex

In[•]:= nb = NotebookGet[NotebookOpen@FileNameJoin[{Directory[], "nb2tex.nb"}]];
nb[[1, 1]]
Out[•]=
Cell[
 Pensieve header: This is the main notebook for the nb2tex project, containing both the
   documentation and the implementation., Text, CellChangeTimes →
  {{3.79032×10^9, 3.79032×10^9}, {3.79035×10^9, 3.79035×10^9}, {3.79035×10^9, 3.79035×10^9}}]

In[•]:= Cases[nb, c_Cell /; FreeQ[c, _Cell, {1, ∞}], ∞][[11]]
Out[•]=
Cell[
 BoxData[FormBox[StyleBox[RowBox[{L, StyleBox[AdjustmentBox[A, BoxBaselineShift → -0.4,
         BoxMargins → {{-0.5, -0.3}, {0, 0}}], FontSize → Smaller], T,
       AdjustmentBox[E, BoxBaselineShift → 0.5, BoxMargins → {{-0.3, 0}, {0, 0}}], X}],
    SingleLetterItalics → False], TraditionalForm]]]
```

In[•]:=
```
Table[
 type = cell[[2]];
 tag = CellTags /. Cases[cell, _Rule] /. CellTags → "";
 {type, tag},
 {cell, Cases[nb, c_Cell /; Length[c] ≥ 2 ∧ FreeQ[c, _Cell, {1, ∞}], ∞]}
]
```

Out[•]=
```
{{Text, }, {Subsection, }, {Input, }, {Output, }, {Input, }, {Output, }, {Input, },
 {Output, }, {Input, }, {Output, }, {Text, tex}, {Text, tex}, {Text, tex},
 {Subsection, pdf}, {Text, tex}, {Text, tex}, {Text, tex}, {Text, }, {Text, tex},
 {Input, pdf}, {Echo, pdf}, {Output, pdf}, {Input, pdf}, {Output, pdf}, {Text, tex},
 {Input, pdfgraph}, {Output, pdfgraph}, {Text, tex}, {Input, pdf}, {Text, tex},
 {Text, tex}, {Text, exec}, {Input, pdf}, {Output, pdf}, {Text, exec}, {Subsection, pdf},
 {Input, pdf}, {Subsection, pdf}, {Input, pdf}, {Message, pdf}, {Message, pdf},
 {Message, pdf}, {Message, pdf}, {Message, pdf}, {Message, pdf}, {Message, pdf},
 {Message, pdf}, {Message, pdf}, {Message, pdf}, {Output, pdf}, {Text, tex}}
```

In[•]:=
```
PDFWidth = 7.2;
PDFFolder = "Sample";
If[FileType[PDFFolder] === None, CreateDirectory[PDFFolder]];
PDFCounter = 0;
lines = Table[
   type = cell[[2]];
   tag = CellTags /. List @@ cell[[3 ;;]] /. CellTags → "";
   Which[
    type == "Text" ∧ tag == "tex", cell[[1]],
    StringMatchQ[tag, "pdf" ~~ ___], (
     pdfname = PDFFolder <> "/" <> ToString[++PDFCounter] <> ".pdf";
     Export[pdfname, Append[cell, PageWidth → 108 PDFWidth]];
     StringReplace[
      "\\noindent\\nbpdfXXXType{pdfname}",
      {"XXX" → StringDrop[tag, 3], "Type" → type, "pdfname" → pdfname}
     ]
    ),
    type == "Text" ∧ tag == "exec", ToExpression[cell[[1]]]; "",
    True, ""
   ],
   {cell, Cases[nb, c_Cell /; FreeQ[c, _Cell, {1, ∞}], ∞]}
  ];
StringJoin @@ Riffle[DeleteCases[lines, ""], "\n\n"]
```

Out[•]=
```
\documentclass[12pt,reqno]{amsart}
\usepackage{graphicx}
\usepackage[textwidth=6.5in,textheight=9in,headsep=0.15in,centering]{geometry}

\def\nbpdfInput#1{\vskip 1mm\noindent\includegraphics{#1}}
\def\nbpdfEcho#1{\vskip 1mm\noindent\includegraphics{#1}}
```

```
\def\nbpdfOutput#1{\vskip 1mm\noindent\includegraphics{#1}}
\def\nbpdfgraphInput#1{\vskip 1mm\noindent\includegraphics{#1}}
\def\nbpdfgraphOutput#1{\vskip 1mm\noindent\includegraphics{#1}}

\begin{document}

The nb2tex project aims to write a converter that
  takes Mathematica notebooks and converts them into latex files.

Text cells with tag ``tex'' becaome verbatim tex output. They may
  include anything texish --- like formulas $1+1=2$, or anything else.

Untagged cells are ignored; for example,
  the following one should not appear in the latex result:

Cells with tag ``pdf'' are converted into numbered pdf files in a pdf-subfolder
  (default name: same as the notebook's name; in this case, ``Sample''). The
  latex produced is \verb$\nbpdfType{Sample/nnn.pdf}$, where \verb$Type$ is
  the type of the current cell (\verb$Text$, \verb$Input$, \verb$Output$,
  \verb$Echo$, etc.), and \verb$nnn$ is the number of the current pdf file.

For example:

\noindent\nbpdfInput{Sample/1.pdf}

\noindent\nbpdfEcho{Sample/2.pdf}

\noindent\nbpdfOutput{Sample/3.pdf}

\noindent\nbpdfInput{Sample/4.pdf}

\noindent\nbpdfOutput{Sample/5.pdf}

If a tag is of the form ``pdfXXX'', the string XXX gets added to the
  \verb$\nbpdf$ command, so it becomes \verb$\nbpdfXXXType$. This is useful for
  graphics inclusions, for example, where a useful tag would be ``pdfgraph'':

\noindent\nbpdfgraphInput{Sample/6.pdf}

\noindent\nbpdfgraphOutput{Sample/7.pdf}

Invoking nb2tex (suffixes are automatically added and should not be included):

\noindent\nbpdfInput{Sample/8.pdf}

Valid options include:
\begin{itemize}
```

```
\item \verb$"PDFFolder" -> foldername$ (a string).
\item \verb$"PDFWidth" -> width$ (in inches).
\end{itemize}

Text cells with tag ``exec'' get executed using \verb$ToExpression$ at the time
  of their processing, with no output produced. This is useful for setting /
  re-setting options within the notebook itself. For example, a text cell with
  tag exec and content \verb"nb2tex$PDFWidth=10" will allow very wide outputs:

\noindent\nbpdfgraphInput{Sample/9.pdf}

\noindent\nbpdfgraphOutput{Sample/10.pdf}

\end{document}

\noindent\nbpdfInput{Sample/11.pdf}
```

LaTeX Prologue

tex

```
\documentclass[12pt,reqno]{amsart}
\usepackage[export]{adjustbox} % This also includes graphicx.
\usepackage{needspace}
\usepackage[textwidth=6.5in,textheight=9in,headsep=0.15in,centering]{geometry}
```

tex

```
\def\nbpdfInput#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfEcho#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfPrint#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfText#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfMessage#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfOutput#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfSection#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfSubsection#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfSubsubsection#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfgraphInput#1{\vskip 1mm\par\noindent\includegraphics{#1}}
\def\nbpdfgraphOutput#1{\vskip 1mm\par\noindent\includegraphics[width=1.5in]{#1}}
```

tex

```
\begin{document}
```

pdf

Documentation

tex

The nb2tex project aims to write a converter that takes Mathematica notebooks and converts them into latex files.

Text cells with tag ``tex'' become verbatim tex output. They may include anything texish --- like formulas $1+1=2$, or anything else.

Untagged cells are ignored; for example, the following one should not appear in the latex result:

This cell must not appear in the latex output.

tex

Cells with tag ``pdf'' are converted into numbered pdf files in a pdf-subfolder (default name: same as the notebook's name; in this case, ``Sample''). The latex produced is \verb$\nbpdfType{Sample/nnn.pdf}$, where \verb$Type$ is the type of the current cell (\verb$Text$, \verb$Input$, \verb$Output$, \verb$Echo$, etc.), and \verb$nnn$ is the number of the current pdf file.

For example:

pdf

In[•]:=
```
binom = Echo[(a + b)^32] // Expand
```

pdf

» $(a+b)^{32}$

Out[•]=
pdf

$$a^{32}+32\,a^{31}\,b+496\,a^{30}\,b^2+4960\,a^{29}\,b^3+35\,960\,a^{28}\,b^4+201\,376\,a^{27}\,b^5+906\,192\,a^{26}\,b^6+3\,365\,856\,a^{25}\,b^7+10\,518\,300\,a^{24}\,b^8+28\,048\,800\,a^{23}\,b^9+64\,512\,240\,a^{22}\,b^{10}+129\,024\,480\,a^{21}\,b^{11}+225\,792\,840\,a^{20}\,b^{12}+347\,373\,600\,a^{19}\,b^{13}+471\,435\,600\,a^{18}\,b^{14}+565\,722\,720\,a^{17}\,b^{15}+601\,080\,390\,a^{16}\,b^{16}+565\,722\,720\,a^{15}\,b^{17}+471\,435\,600\,a^{14}\,b^{18}+347\,373\,600\,a^{13}\,b^{19}+225\,792\,840\,a^{12}\,b^{20}+129\,024\,480\,a^{11}\,b^{21}+64\,512\,240\,a^{10}\,b^{22}+28\,048\,800\,a^{9}\,b^{23}+10\,518\,300\,a^{8}\,b^{24}+3\,365\,856\,a^{7}\,b^{25}+906\,192\,a^{6}\,b^{26}+201\,376\,a^{5}\,b^{27}+35\,960\,a^{4}\,b^{28}+4960\,a^{3}\,b^{29}+496\,a^{2}\,b^{30}+32\,a\,b^{31}+b^{32}$$

\verb$Series[1/(1+x),{x,0,128}]$

```
Series[1 / (1 + x), {x, 0, 128}]
```

Out[•]=

$$1-x+x^2-x^3+x^4-x^5+x^6-x^7+x^8-x^9+x^{10}-x^{11}+x^{12}-x^{13}+x^{14}-x^{15}+x^{16}-x^{17}+x^{18}-x^{19}+x^{20}-x^{21}+x^{22}-x^{23}+x^{24}-x^{25}+x^{26}-x^{27}+x^{28}-x^{29}+x^{30}-x^{31}+x^{32}-x^{33}+x^{34}-x^{35}+x^{36}-x^{37}+x^{38}-x^{39}+x^{40}-x^{41}+x^{42}-x^{43}+x^{44}-x^{45}+x^{46}-x^{47}+x^{48}-x^{49}+x^{50}-x^{51}+x^{52}-x^{53}+x^{54}-x^{55}+x^{56}-x^{57}+x^{58}-x^{59}+x^{60}-x^{61}+x^{62}-x^{63}+x^{64}-x^{65}+x^{66}-x^{67}+x^{68}-x^{69}+x^{70}-x^{71}+x^{72}-x^{73}+x^{74}-x^{75}+x^{76}-x^{77}+x^{78}-x^{79}+x^{80}-x^{81}+x^{82}-x^{83}+x^{84}-x^{85}+x^{86}-x^{87}+x^{88}-x^{89}+x^{90}-x^{91}+x^{92}-x^{93}+x^{94}-x^{95}+x^{96}-x^{97}+x^{98}-x^{99}+x^{100}-x^{101}+x^{102}-x^{103}+x^{104}-x^{105}+x^{106}-x^{107}+x^{108}-x^{109}+x^{110}-x^{111}+x^{112}-x^{113}+x^{114}-x^{115}+x^{116}-x^{117}+x^{118}-x^{119}+x^{120}-x^{121}+x^{122}-x^{123}+x^{124}-x^{125}+x^{126}-x^{127}+x^{128}+O[x]^{129}$$

tex

\par If a tag is of the form ``pdf:Name'', a pdf file by the name ``NameType.pdf'' is created in the pdf folder (where Type is the type of the cell), yet nothing is added to the tex file. This allows to treat certain cells is a special manner. For example, using latex commands that will inline them or use them as side figure. As an example, if the Mathematica input is \includegraphics[valign=t]{Snips/FactorialInput}, the output is \includegraphics[valign=t]{Snips/FactorialOutput}.

(Alt) In[•]:=
pdf:Factorial

```
20!
```

(Alt) Out[•]=
pdf:Factorial

```
2 432 902 008 176 640 000
```

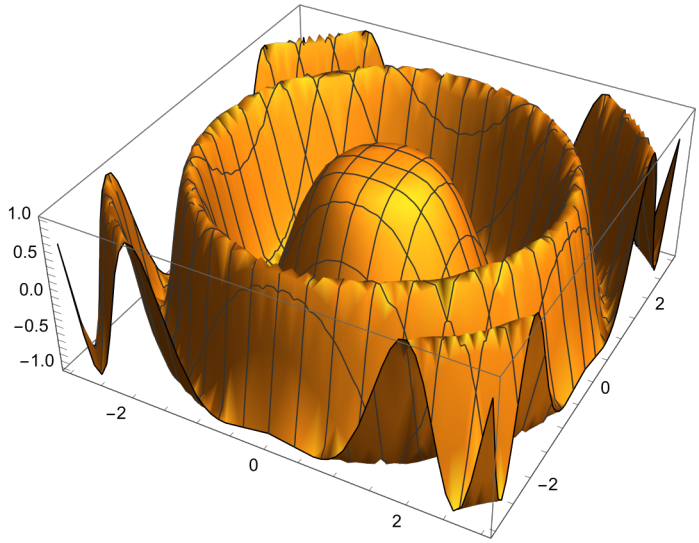
tex

\par If a tag is of the form ``pdfXXX'', with XXX a string not beginning with the character ``:'', the string XXX gets added to the \verb$\nbpdf$ command, so it becomes \verb$\nbpdfXXXType$. This is useful for graphics inclusions, for example, where a useful tag would be ``pdfgraph'':

pdfgraph
In[•]:=

```
plot = Plot3D[Cos[x^2 + y^2], {x, -3, 3}, {y, -3, 3}]
```

Out[•]=
pdfgraph



tex

\par Invoking nb2tex (suffixes are automatically added and should not be included):

pdf

```
nb2tex[nb_String, opts___Rule];
nb2tex[nb_String, tex_String, opts___Rule];
```

tex

\par Valid options include:
\begin{itemize}
\item \verb$"PDFFolder" -> foldername$ (a string).
\item \verb$"PDFWidth" -> width$ (in inches).
\end{itemize}

Input cells with tag ``exec'' get executed using \verb$ToExpression$ at the time of their processing (even if they do not have the property ``Evaluatable), with no output produced. This is useful for setting / re-setting options within the notebook itself. For example, an input cell with tag exec and content \verb"nb2tex$PDFWidth=10" will allow very wide outputs:

exec

In[•]:=
```
nb2tex$PDFWidth = 10
```

tex

\needspace{25mm}

pdf

In[•]:=
```
binom
```

Out[•]=

pdf

$$a^{32} + 32\,a^{31}\,b + 496\,a^{30}\,b^2 + 4960\,a^{29}\,b^3 + 35\,960\,a^{28}\,b^4 + 201\,376\,a^{27}\,b^5 + 906\,192\,a^{26}\,b^6 + 3\,365\,856\,a^{25}\,b^7 + 10\,518\,300\,a^{24}\,b^8 + 28\,048\,800\,a^{23}\,b^9 + 64\,512\,240\,a^{22}\,b^{10} + 129\,024\,480\,a^{21}\,b^{11} + 225\,792\,840\,a^{20}\,b^{12} + 347\,373\,600\,a^{19}\,b^{13} + 471\,435\,600\,a^{18}\,b^{14} + 565\,722\,720\,a^{17}\,b^{15} + 601\,080\,390\,a^{16}\,b^{16} + 565\,722\,720\,a^{15}\,b^{17} + 471\,435\,600\,a^{14}\,b^{18} + 347\,373\,600\,a^{13}\,b^{19} + 225\,792\,840\,a^{12}\,b^{20} + 129\,024\,480\,a^{11}\,b^{21} + 64\,512\,240\,a^{10}\,b^{22} + 28\,048\,800\,a^{9}\,b^{23} + 10\,518\,300\,a^{8}\,b^{24} + 3\,365\,856\,a^{7}\,b^{25} + 906\,192\,a^{6}\,b^{26} + 201\,376\,a^{5}\,b^{27} + 35\,960\,a^{4}\,b^{28} + 4960\,a^{3}\,b^{29} + 496\,a^{2}\,b^{30} + 32\,a\,b^{31} + b^{32}$$

exec
```
nb2tex$PDFWidth = 6.5
```

tex

\par Similarly, changing \verb”nb2tex$TeXFileName” will change the file where the latex output is written.

pdf

Implementation

As in http://drorbn.net/AcademicPensieve/Projects/nb2tex/.

(Alt) In[•]:=

pdf
```
SetOptions[$FrontEndSession, PrintingStyleEnvironment → "Working"];
nb2tex[nb_String, opts___Rule] := nb2tex[nb, nb, opts];
```

(Alt) In[ ]:=
pdf

```
nb2tex[nb_String, tex_String, opts___Rule] := Module[
  {notebook, PDFCounter = 0, type, tag, pdfname, cells, cell, c, cl, texfiles = {}, TeXOut,
   PDFFolder = PDFFolder /. {opts} /. PDFFolder → nb
  },
  nb2tex$TeXFileName = tex <> ".tex";
  nb2tex$PDFWidth = PDFWidth /. {opts} /. PDFWidth → 6.5;
  TeXOut[s_String] := (texfiles = texfiles ⋃ {nb2tex$TeXFileName};
    WriteString[nb2tex$TeXFileName, s]);
  notebook = NotebookGet[NotebookOpen@FileNameJoin[{Directory[], nb <> ".nb"}]];
  If[FileType[PDFFolder] === None, CreateDirectory[PDFFolder]];
  DeleteFile /@ FileNames["*.pdf", PDFFolder];
  cells = Cases[notebook, c_Cell /; Length[c] ≥ 2, ∞];
  Do[
   type = cell[[2]];
   tag = CellTags /. Cases[cell, _Rule] /. CellTags → "";
   Which[
    type == "Text" ∧ tag == "tex", TeXOut[
     StringReplace[cell[[1]], {"’" → "'", "”" → "\""}] <> "\n\n"],
    StringMatchQ[tag, "pdf:" ~~ ___], (
     pdfname = PDFFolder <> "/" <> StringDrop[tag, 4] <> type <> ".pdf";
     Export[pdfname, Join[cell, Cell[PageWidth → 80 nb2tex$PDFWidth / 0.75]]];
     cl = "c:\\drorbn\\bin\\cpdf.exe -scale-page \"0.75 0.75\" " <>
       pdfname <> " -o " <> pdfname;
     Close@OpenRead["!" <> cl];
    ),
    StringMatchQ[tag, "pdf" ~~ ___], (
     pdfname = PDFFolder <> "/" <> ToString[++PDFCounter] <> ".pdf";
     Export[pdfname, Join[cell, Cell[PageWidth → 80 nb2tex$PDFWidth / 0.75]]];
     cl = "c:\\drorbn\\bin\\cpdf.exe -scale-page \"0.75 0.75\" " <>
       pdfname <> " -o " <> pdfname;
     Close@OpenRead["!" <> cl];
     TeXOut[StringReplace[
       "\\nbpdfXXXType{pdfname}\n",
       {"XXX" → StringDrop[tag, 3], "Type" → type, "pdfname" → pdfname}
      ]]
    ),
    type == "Input" ∧ tag == "exec", ToExpression[cell[[1]]],
    True, Null
   ],
   {cell, cells}
  ];
  Close /@ texfiles;
 ]
```

pdf

Run

(Alt) In[ ]:=
pdf

```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Projects\\nb2tex"];
nb2tex["nb2tex", PDFFolder → "Snips"];
Run@
 "\"C:\\Users\\drorb\\AppData\\Local\\Programs\\MiKTeX\\miktex\\bin\\x64\\pdflatex.exe\"
   nb2tex.tex"
```

(Alt) Out[ ]=
pdf

```
0
```

LaTeX Epilogue

tex

```
\end{document}
```