

# Pensieve Assembly File

Pensieve Header: This notebook assembles the web version of my academic pensieve.

---

## Actions

### ■ Assemble All

```
Print [Date []];
AcademicPensieveAssemble ["all"];
{2010, 6, 26, 7, 31, 17.0480000}
Exporting C:\drorbn\AcademicPensieve\AssemblePensieve.nb
  into C:\drorbn\AcademicPensieve\nb\AssemblePensieve.pdf ...
Exporting C:\drorbn\AcademicPensieve\PublicNotebook\2010-06.one
  into C:\drorbn\AcademicPensieve\2010-06\one\ ...
Writing C:\drorbn\AcademicPensieve\2010-06\one\Report_on_a_visit_to_Ghana,_June_2010.pdf ...
C:\drorbn\AcademicPensieve\2010-06\index.html
C:\drorbn\AcademicPensieve\2010-06\one\index.html
C:\drorbn\AcademicPensieve\index.html
```

### ■ Convert all .nb Files

```
AcademicPensieveAssemble ["nb"]
```

### ■ Convert all .one Files

```
AcademicPensieveAssemble ["one"];
```

### ■ Make thumbnails for all images

```
AcademicPensieveAssemble ["thumbs"]
```

### ■ Create Index Pages

```
AcademicPensieveAssemble ["indexes"]
```

- AcademicPensieveAssembleRandom.php

```
AcademicPensieveAssemble ["random.php"]
AcademicPensieveAssemble [random.php]
```

- Delete all TOC files

```
(* DeleteFile /@ FileNames["TOC.m", {AcademicPensieveDirectory}, Infinity]; *)
```

- Delete all index files

```
DeleteFile /@ FileNames["index.html", {AcademicPensieveDirectory}, Infinity];
```

## Programs

- Assemble All

```
BeginPackage ["AcademicPensieve`"];
AcademicPensieveDirectory = "C:\\drorbn\\AcademicPensieve";
AcademicPensieveAssemble ["all"] :=
  AcademicPensieveAssemble /@ {"nb", "one", "thumbs", "indexes", "random"};
EndPackage []
```

- Utilities / General

```
BeginPackage ["AcademicPensieve`"]; Begin["`Private`"];
RootDir = AcademicPensieveDirectory;
StripRootDir[s_] := StripDir[s, RootDir];
StripDir[s_, sdir_] := Module[{dir},
  dir = StringReplace[s, sdir -> ""];
  While[dir != "" && (StringTake[dir, 1] == "\\ " || StringTake[dir, 1] == "/"),
    dir = StringDrop[dir, 1]
  ];
  dir
];
KosherFilename[s_String] := StringReplace[s, {
  " " -> "_", ":" -> "-", "/" -> "-", "?" -> "Q", "\" -> "'"
}];
End[]; EndPackage []
```

## ■ For .nb files

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
OpenNotebooks = Notebooks[];
NB2PDF[NotebookFilename_String] := Module[
  {SplitName, PDFFilename, PDFDir, nb, TOCFilename, toc, summary},
  SplitName = StringSplit[NotebookFilename, {"\\", "/" }];
  PDFFilename = ToFileName[
    PDFDir = ToFileName[Append[Drop[SplitName, -1], "nb"],
    StringDrop[Last@SplitName, -2] <> "pdf"
  ];
  ];
  If[
    Or[
      FileType[PDFFilename] === None,
      AbsoluteTime[FileDate[PDFFilename]] < AbsoluteTime[FileDate[NotebookFilename]]
    ],
    Print["Exporting ", NotebookFilename, " into ", PDFFilename, " ..."];
    If[FileType[PDFDir] === None, CreateDirectory[PDFDir]];
    NotebookPrint[
      nb = NotebookOpen[NotebookFilename], PDFFilename
    ];
    If[AbsoluteTime[FileDate[PDFFilename]] < AbsoluteTime[FileDate[NotebookFilename]],
      Print["PDF creation failed for ", NotebookFilename]
    ];
    TOCFilename = ToFileName[PDFDir, "TOC.m"];
    If[FileType[TOCFilename] != File, toc = {},
      toc = Get[TOCFilename]
    ];
    summary = Cases[
      NotebookGet[nb],
      cc_String?(StringMatchQ[#, ("Pensieve Header: ") ~~ ___] &),
      Infinity, 1
    ];
    toc = DeleteCases[toc, StringDrop[Last@SplitName, -3] → _];
    If[summary != {},
      AppendTo[toc, StringDrop[Last@SplitName, -3] → StringDrop[First[summary], 17]]
    ];
    Put[toc, TOCFilename];
    If[!MemberQ[OpenNotebooks, nb], NotebookClose[nb]];
  ];
  PDFFilename

```

```

];
AcademicPensieveAssemble["nb"] := Module[
  {legits, orphans},
  legit = NB2PDF /@ FileNames["*.nb", {RootDir}, Infinity];
  orphans = Complement[
    Flatten[
      If[FileType[#] === Directory,
        FileNames["*", {#}],
        {}
      ] & /@ FileNames["nb", {RootDir}, Infinity]
    ],
    legit,
    FileNames["index.html" | "TOC.m", {RootDir}, Infinity]
  ];
  DeleteFile[orphans]
];
End[]; EndPackage[]

```

## ■ For .one files

More on the Mathematica / .NET interface is at <http://reference.wolfram.com/mathematica/NETLink/tutorial/CallingNETFrom-Mathematica.html>

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
Needs["NETLink`"];
If[! NETObjectQ[OneNoteLink],
  InstallNET[];
  OneNoteLink = CreateCOMObject["OneNote.Application"]
];
One2PDF[OneNoteFilename_String] := Module[
  {
    SplitName, PDFDirectory, OneNoteDocument,
    XMLString, XML, PageDescriptors, legit = {}, tocfile
  },
  SplitName = StringSplit[OneNoteFilename, {"\\", "/" }];
  PDFDirectory = ToFileName[Flatten[{
    DeleteCases[Drop[SplitName, -1], "PublicNotebook" | "old"],
    {StringDrop[Last@SplitName, -4], "one"}
  }]];
  If[
    ! Or[
      FileType[PDFDirectory] === None,

```

```

FileType[ToFileName[PDFDirectory, "TOC.m"]] === None,
AbsoluteTime[FileDate[PDFDirectory]] < AbsoluteTime[FileDate[OneNoteFilename]]
],
False && Print["Skipping ", OneNoteFilename, "."],
Print["Exporting ", OneNoteFilename, " into ", PDFDirectory, " ..."];
If[FileType[PDFDirectory] === None, CreateDirectory[PDFDirectory]];
OneNoteLink@OpenHierarchy[OneNoteFilename, "", OneNoteDocument];
OneNoteLink@GetHierarchy[OneNoteDocument, 4, XMLString];
XML = ImportString[XMLString, "XML"];
PageDescriptors = Cases[XML, XMLElement[[_, "Page"], page_, {}] -> page, Infinity];
legits = OnePage2PDF[#, PDFDirectory] & /@ PageDescriptors;
Put[XML, tocfile = ToFileName[PDFDirectory, "TOC.m"]];
AppendTo[legits, tocfile];
DeleteFile[Complement[FileNames["*", PDFDirectory], legit]]
];
];
OnePage2PDF[desc_List, dir_String] := Module[
{ID, name, dateTime, lastModifiedTime, pdffilename},
{ID, name, dateTime, lastModifiedTime} =
{"ID", "name", "dateTime", "lastModifiedTime"} /. desc;
pdffilename = ToFileName[dir,
KosherFilename[name] <> ".pdf"
];
If[
Or[
FileType[pdffilename] === None,
AbsoluteTime[DatePlus[FileDate[pdffilename], {- $TimeZone, "Hour"}]] <
AbsoluteTime[lastModifiedTime]
],
Print["Writing ", pdffilename, " ..."];
If[FileType[pdffilename] != None, DeleteFile[pdffilename]];
OneNoteLink@Publish[ID, pdffilename, 3, ""]
];
pdffilename
];
AcademicPensieveAssemble["one"] := If[! NETObjectQ[OneNoteLink],
$Failed, One2PDF /@ FileNames["*.one", {RootDir}, Infinity]];
AcademicPensieveAssemble["one"] := If[! NETObjectQ[OneNoteLink], $Failed,
One2PDF /@ FileNames["*.one", {RootDir}, Infinity]
];
End[]; EndPackage[]

```

## ■ Make Thumbnails

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
ImageTypes = (*.jpg | *.gif | *.png);
MakeThumb[ImageFilename_String] := Module[
  {SplitName, ThumbFilename, ThumbDir},
  SplitName = StringSplit[ImageFilename, {"\\", "/" }];
  ThumbFilename = ToFileName[
    ThumbDir = ToFileName[Append[Drop[SplitName, -1], "thumbs"],
    Last@SplitName
  ];
  If[
    And[
      SplitName[[-2]] != "thumbs",
      Or[
        FileType[ThumbFilename] === None,
        AbsoluteTime[FileDate[ThumbFilename]] < AbsoluteTime[FileDate[ImageFilename]]
      ]
    ],
    Print["Thumbnailing ", ImageFilename, " into ", ThumbFilename, " ..."];
    If[FileType[ThumbDir] === None, CreateDirectory[ThumbDir]];
    Export[ThumbFilename, ImageResize[Import[ImageFilename], {120}]];
  ];
  ThumbFilename
];
AcademicPensieveAssemble["thumbs"] := Module[
  {legits, orphans},
  legit = MakeThumb /@ FileNames[ImageTypes, {RootDir}, Infinity];
  orphans = Complement[
    Flatten[
      If[FileType[#] === Directory,
        FileNames["*", {#}],
        {}
      ] & /@ FileNames["thumbs", {RootDir}, Infinity]
    ],
    legit,
    FileNames["index.html", {RootDir}, Infinity]
  ];
  DeleteFile[orphans]
];
End[]; EndPackage[]

```

## ■ Cell Extraction

```
BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
Clear[ExtractDataCell];
ExtractDataCell[cn_String] := ExtractDataCell[cn] = First[Cases[
  NotebookGet[EvaluationNotebook[]],
  Cell[
    cc_String?(StringMatchQ[#, ("Data Cell " <> cn <> ":\n") ~~ ___] &),
    "Text", ___
  ] => Last[StringSplit[cc, "\n", 2]],
  Infinity, 1
]];
(* ExtractDataCell /@ {"Index Template", "random.php"}; *)
End[]; EndPackage[]
```

## ■ Assemble Index Pages

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
DateStringFormat =
  {"Year", "-", "Month", "-", "Day", " ", "Hour24", ":", "Minute", ":", "Second"};
AssembleIndexPage[s_String] := Module[
  {
    dir, fulldir, Customizations, parentdir, next, previous, siblings, p, fname, t, rand,
    d, SplitPath, l, FullTitle, Title, LinkToRoot, Navigator, TitleNotes, i, j,
    OneNoteTOC, OneNotePages, OneNoteDir, OneNoteData, NBDir, NBTOCFilename,
    NBTOC, MathematicaNotebooks, NBFileNames, ThumbsDir, Images, ImageFileNames,
    DirectoryListing, style, ThisMonth, FileListing, OtherFiles, last, htmlfile
  },
  dir = StripRootDir[s];
  fulldir = ToFileName[RootDir, dir];
  Customizations = ToFileName[fulldir, "index.m"];
  Customizations = If[FileType[Customizations] === File, Get[Customizations], {}];
  If[dir === "", next = previous = "AcademicPensieve",
    parentdir =
      StringReplace[fulldir, par__ ~~ Shortest[{"\\", "/"} ~~ __ ~~ EndOfString] -> par];
  siblings = Select[FileNames["*", parentdir], FileType[#] === Directory &];
  l = Length[siblings];
  {{p}} = Position[siblings, fulldir];
  next = StripDir[siblings[[1 + Mod[p, l]]], parentdir];
  previous = StripDir[siblings[[1 + Mod[p - 2, l]]], parentdir];
  ];
  fname = ToFileName[fulldir, "index.html"];
  BlockRandom[SeedRandom[fname]; rand = RandomReal[]];
  If[
    ! Or[
      FileType[fname] === None,
      (t = AbsoluteTime[FileDate[fname]]) < AbsoluteTime[FileDate[fulldir]],
      FileType[d = ToFileName[fulldir, "nb"]] === Directory && t < Max[
        AbsoluteTime[FileDate[#]] & /@ FileNames["*.pdf", d]
      ],
      FileType[d = ToFileName[fulldir, "one"]] === Directory &&
        t < AbsoluteTime[FileDate[d]],
      AbsoluteTime[] - t > (12 + 4 * rand) * 24 * 60 * 60
    ],
    False && Print["Skipping ", fulldir, "."],
    l = Length[SplitPath = Prepend[StringSplit[dir, {"\\", "/"}], "AcademicPensieve"]];

```



```

FullTitle = StringJoin[ (# <> ": ") & /@ SplitPath];
Title = Last[SplitPath];
LinkToRoot = StringJoin[Table["../", {1 - 1}]];
Navigator = StringJoin[Table[
  StringJoin[
    "<a href=\"",
    StringJoin[Table["../", {1 - i}]],
    "index.html\">" <> SplitPath[[i]] <> "</a>: "
  ],
  {i, 1 - 1}
]];
TitleNotes = "TitleNotes" /. Customizations /. "TitleNotes" -> "";
OneNotePages = If[FileType[ToFileName[{fulldir, "one"}, "TOC.m"]] === None, "",
  OneNoteTOC = Get[ToFileName[{fulldir, "one"}, "TOC.m"]];
  OneNoteDir = FileNameJoin[Flatten[{
    "..",
    Drop[
      FileNameSplit["path" /. OneNoteTOC[[2, 2]],
      Length[FileNameSplit[AcademicPensieveDirectory]]
    ]
  }]];
OneNoteData = OneNoteTOC[[2, 3, All, 2]];
StringJoin[
  "<h2>Notebook Pages</h2>\n",
  "NotebookPagesNotes" /. Customizations /. "NotebookPagesNotes" -> "",
  "<table border=1 cellspacing=0>\n",
  "<tr><th>Page</th><th>Created  
(UT)</th><th>Last Modified (UT)</th></tr>\n",
  StringJoin[StringJoin[
    "<tr align=left>\n",
    StringJoin[
      " <td>",
      If[("isSubPage" /. #) == "true", "--- ", ""],
      "<a href=\"one/",
      KoshersFilename["name" /. #],
      ".pdf\">",
      "name" /. #,
      "</a></td>\n"
    ],
    ],
    " <td>" <>
    StringReplace["dateTime" /. #, {"T" -> "&nbsp;", ".000Z" -> ""}] <> "</td>\n",
    " <td>" <> StringReplace["lastModifiedTime" /. #,
      {"T" -> "&nbsp;", ".000Z" -> ""}] <> "</td>\n",

```

```

        "</tr>\n"
    ] & /@ OneNoteData],
"</table>",
"(<a href=", OneNoteDir, ">.one source file</a> for all pages above)\n"
]
];
NBDir = ToFileName[fulldir, "nb"];
MathematicaNotebooks = If[FileType[NBDir] != Directory, "",
NBTOCFilename = ToFileName[NBDir, "TOC.m"];
NBTOC = If[FileType[NBTOCFilename] != File, {}, Get[NBTOCFilename]];
l = Length[
    NBFileNames =
        StringTake[StringReplace[#, NBDir -> ""], {2, -5}] & /@ FileNames["*.pdf", NBDir]
];
StringJoin[
    "<h2>Mathematica Notebooks</h2>\n",
    "MathematicaNotebooksNotes " /. Customizations /.
    "MathematicaNotebooksNotes " -> "",
    "<table border=1 cellpadding=0>\n",
    "<tr><th>Notebook
    (.pdf)</th><th>Source (.nb)</th><th>Created</th><th>Last
    Modified</th><th>Summary</th></tr>\n",
StringJoin[
    StringJoin[
        "<tr align=left>\n",
        "    <td><a href=\"nb/", #, ".pdf\">", #, "</a></td>\n",
        "    <td align=center><a href=\"", #, ".nb\">source</a></td>\n",
StringJoin[
        "    <td align=center>",
        DateString[FileDate[
            ToFileName[fulldir, # <> ".nb"],
            "Creation"
        ]],
        "</td>\n"
    ],
StringJoin[
        "    <td align=center>",
        DateString[FileDate[ToFileName[fulldir, # <> ".nb"]]],
        "</td>\n"
    ],
    "    <td>", (# /. NBTOC) /. # -> "&nbsp;", "</td>\n"
] & /@ NBFileNames
],

```

```

    "</table>\n"
  ]
];
ThumbsDir = ToFileName[fulldir, "thumbs"];
Images = If[FileType[ThumbsDir] != Directory, "",
  ImageFileNames = StringDrop[StringReplace[#, ThumbsDir -> ""], 1] & /@
  FileNames[ImageTypes, ThumbsDir];
StringJoin[
  "<h2>Images</h2>\n",
  "<table border=0 cellpadding=5 width=100%>\n",
  StringJoin[
    i = 0;
    StringJoin[
      ++i;
      If[OddQ[i], " <tr>", ""],
      "<td width=125 align=center><a href=\"",
      #, "\"><img src=\"thumbs/", #, "\"></a></td>",
      "<td align=left><a href=\"", #, "\">", #, "</a></td>",
      If[EvenQ[i], "</tr>\n", ""]
    ] & /@ ImageFileNames
  ],
  If[OddQ[i], "</tr>\n", ""],
  "</table>\n"
]
];
FileListing = FileNames["*", {fulldir}];
DirectoryListing = Select[FileListing, FileType[#] == Directory &];
OtherFiles = FileListing =
  Last[StringSplit[#, {"\\", "/"}]] & /@ Complement[FileListing, DirectoryListing];
OtherFiles = FileListing;
FileListing = Union[{"index.html"}, FileListing];
FileListing = StringJoin[(
  " <a href=\"\" <>#<> \"\"> <>#<> </a>&nbsp;&nbsp;&nbsp;\n"
) & /@ FileListing];
OtherFiles = Complement[OtherFiles,
  Union[Flatten[{
    "index.html", "index.m", (#<> ".nb") & /@ NBFileNames, ImageFileNames
  }]]
];
OtherFiles = StringJoin[(
  " <a href=\"\" <>#<> \"\"> <>#<> </a>&nbsp;&nbsp;&nbsp;\n"
) & /@ OtherFiles];
ThisMonth = StringJoin[

```

```

ToString[Date][[1]],
"-",
IntegerString[Date][[2]], 10, 2]
];
DirectoryListing = StringJoin[(
  last = Last[StringSplit[#, {"\\", "/"}]];
  If[last == ThisMonth,
    style = "style=\"background-color: yellow;\"",
    style = ""
  ];
  " <a " <> style <> " href=\"" <>
  last <> "/index.html\">" <> last <> "</a>&nbsp;&nbsp;&nbsp;"
) & /@ DirectoryListing];
htmlfile = OpenWrite[fname];
WriteString[htmlfile,
StringReplace[ExtractDataCell["Index Template"], {
  "<#FullTitle#" -> FullTitle,
  "<#Title#" -> Title,
  "<#TitleNotes#" -> TitleNotes,
  "<#next#" -> next,
  "<#previous#" -> previous,
  "<#Navigator#" -> Navigator,
  "<#LinkToRoot#" -> LinkToRoot,
  "<#OneNotePages#" -> OneNotePages,
  "<#MathematicaNotebooks#" -> MathematicaNotebooks,
  "<#Images#" -> Images,
  "<#OtherFiles#" -> OtherFiles,
  "<#FileListing#" -> FileListing,
  "<#DirectoryListing#" -> DirectoryListing
}]]
];
Print[Close[htmlfile]]
]
];
AcademicPensieveAssemble["indexes"] := (
  AssembleIndexPage /@
  Select[FileNames["*", RootDir, Infinity], FileType[#] == Directory &];
  AssembleIndexPage[""];
);
End[]; EndPackage[]

```

## ■ Assemble Random.php and Random.html

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
AcademicPensieveAssemble["random"] := Module[
  {db, phpfile, htmlfile},
  db = StringDrop[StringJoin[
    StringJoin[
      "\"",
      StringDrop[StringReplace[#, {RootDir -> "", "\\\" -> "/}], 1],
      "\",\n"
    ] & /@ FileNames["*.html" | "*.pdf", {RootDir}, Infinity]
  ], -2];
  phpfile = OpenWrite[ToFileName[RootDir, "random.php"]];
  WriteString[phpfile,
    StringReplace[ExtractDataCell["random.php"], "<#DB#>" -> db]
  ];
  Close[phpfile];
  htmlfile = OpenWrite[ToFileName[RootDir, "random.html"]];
  WriteString[htmlfile,
    StringReplace[ExtractDataCell["random.html"], "<#DB#>" -> db]
  ];
  Close[htmlfile]
]
End[]; EndPackage[]

```



```

<#MathematicaNotebooks#>
<#Images#>

<h2>Other Files</h2>
<#OtherFiles#>

<h2>All Files</h2>
<#FileListing#>

</body></html>

```

### ■ Random.php

Data Cell random.php:

```

<! Script generated - do not edit! >
<?php
$db = array (
<#DB#>
);
$rand= $db[array_rand($db)];
echo "<meta http-equiv=refresh content=\"0; url=$rand\">";
?>

```

### ■ Random.html

Data Cell random.html:

```

<! Script generated - do not edit! >
<script language=JavaScript>
db = [
<#DB#>
];
document.location = db[Math.floor(Math.random()*db.length)];
</script>

```

---

## Experiments

### ■ Do I need an image browser?

```

FileNames ["*.jpg" | "*.png" | "*.gif", {AcademicPensieveDirectoty}, Infinity]
{}

```