

Copyright | Dror Bar-Natan: Academic Pensieve: Projects: Academic Pensieve:

Pensieve Assembly Programs

Pensieve Header: This notebook contains the programs that assemble the web versions of my pensieves.

Requires `cpdf` in `C:\drorbn\bin`.

To do

Incorporate WinSort from <https://mathematica.stackexchange.com/questions/10619/sort-strings-by-natural-ordering>:

(Alt) In[]:=

```
WinSort[names_List] := names[[Ordering[Characters[names]]]];
```

Assemble All

```
In[ ]:= BeginPackage["Pensieve`"];
PensieveAssemble["all"] := (
  Print["Recognized Customizations:", Sort@{
    "ImportFiles", "CleanTeX", "TitleNotes", "FootNotes",
    "NotebookPagesNotes", "MathematicaNotebooksNotes", "ImageComments", "IgnoreShortcuts",
    "ExcludeDirectories", "DirectoryListingSorter", "DirectoryListingSeparator"
  }];
  InitializeTemplates[];
  PensieveAssemble /@ {"nb", (*"one",*) "thumbs", "previews", "indexes", "random"};
)
EndPackage[]
```

Utilities / General

```

BeginPackage["Pensieve`"];
{StripRootDir, LinkTarget, PensieveName, PensieveDirectory, PensieveURL,
 DoNotIndex, DoNotStamp, MakeThumb, MakePreview, AssembleIndexPage, ShortcutTarget};
Begin["`Private`"];

If[Head[DoNotIndex] == Symbol, DoNotIndex = ""];
If[Head[DoNotStamp] == Symbol, DoNotStamp = ""];

AcademicPensieveDirectory = "C:\\drorbn\\AcademicPensieve";
StripRootDir[s_] := StripDir[s, PensieveDirectory];
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",
  "\" -> "'", "<" -> "(", ">" -> ")", "\\\" -> "!", "*" -> "$", "ж" -> "$zhe"
}];

ThisMonth = StringJoin[
  ToString[Date[]][[1]],
  "-",
  IntegerString[Date[]][[2], 10, 2]
];

ShortcutTarget::failed = "Failed to detect target for `1`.";
ShortcutTarget[Lnk_String] := Module[
  {l, z},
  z = FromCharacterCode[0];
  l = StringCases[
    FromCharacterCode[BinaryReadList[Lnk]],
    {PensieveDirectory <> "\\\", "\\DROR-X220\\drorbn" <> z <> "AcademicPensieve\\\",
     "\\DROR-SP5\\drorbn" <> z <> "AcademicPensieve\\"} ~ Shortest[u_] ~ z -> u
  ];
  If[Head[l] === List && Length[l] > 0, First[l],
    Message[ShortcutTarget::failed, Lnk]; ""
  ]
];

LinkTarget[Lnk_String] := First[StringCases[
  FromCharacterCode[BinaryReadList[Lnk]],
  "URL=" ~ Shortest[u_] ~ "\\r" ... ~ EndOfLine -> u
]];

End[]; EndPackage[]

```

For .nb files

```

BeginPackage["Pensieve`"]; Begin["`Private`"];
NB2PDF[NotebookFilename_String] := Module[
  {SplitName, PDFFilename, PDFDir, nb,
   TOCFilename, toc, summary, l, LinkToRoot, suffix, navigator, url},
  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["nb: ", StripDir[PDFFilename, PensieveDirectory]];
  l = Length[StringSplit[StripDir[NotebookFilename, PensieveDirectory], {"\\", "/"}]];
  LinkToRoot = StringJoin[Table["../", {l - 1}]];
  If[FileType[PDFDir] === None, CreateDirectory[PDFDir]];
  navigator = StringJoin[Flatten[{
    "Dror Bar-Natan: Academic Pensieve: ",
    Riffle[FileNameSplit[StripRootDir[NotebookFilename]], ": "]
  }]];
  url = If[ValueQ[PensieveURL],
    StringJoin[Flatten[{
      PensieveURL,
      Riffle[
        Most[FileNameSplit[StripRootDir[NotebookFilename]],
        "/"
      ],
      "/#MathematicaNotebooks"
    }]],
    "private"
  ];
  nb = Get[NotebookFilename];
  AppendTo[nb, #] & /@ {
    PageHeaders → 1,
    PageFooters → 1,
    StyleDefinitions → Notebook[{
      Cell[StyleData[StyleDefinitions → "Default.nb"]],
      Cell[StyleData[All, "Printout"], ShowSyntaxStyles → True]
    }]
  ];
  If[FreeQ[nb, PrintingOptions → _, ∞], AppendTo[nb, PrintingOptions → {}]];
  nb = nb /. {
    (PageHeaders → _) → (PageHeaders → Table[{navigator, None, DateString[]}, {2}]),
    (PageFooters → _) → (PageFooters → Table[{None, url, None}, {2}]),
    (PrintingOptions → popts_) ⇒ (PrintingOptions →
      Join[popts,
        {"FirstPageHeader" → 1, "FirstPageFooter" → 1, "EmbedStandardPostScriptFonts" → 1}] /. {
          ("FirstPageHeader" → _) → ("FirstPageHeader" → True),

```

```

    ("FirstPageFooter" → _) → ("FirstPageFooter" → True),
    ("EmbedStandardPostScriptFonts" → _) → ("EmbedStandardPostScriptFonts" → False),
    ("PrintingMargins" → _) → ("PrintingMargins" → {{12, 12}, {36, 12}})
  })
};
Export[PDFFilename, nb];
If[AbsoluteTime[FileDate[PDFFilename]] < AbsoluteTime[FileDate[NotebookFilename]],
  Print["PDF creation failed for ", NotebookFilename],
  Null (* SetFileDate[PDFFilename, FileDate[NotebookFilename]] *)
];
TOCFilename = ToFileName[PDFDir, "TOC.m"];
If[FileType[TOCFilename] != File, toc = {},
  toc = Get[TOCFilename]
];
summary = Cases[nb,
  cc_String?(StringMatchQ[#, ("Pensieve Header: " | "Pensieve header: ") ~~ ___] &),
  Infinity, 1
];
toc = DeleteCases[toc, StringDrop[Last@SplitName, -3] → _];
If[summary != {},
  summary = StringDrop[First[summary], 17];
  summary = StringReplace[summary, {"Ж" → "&#x416;", "ж" → "&#x436;", "á" → "&acute;",
    "é" → "&eacute;", "α" → "&alpha;", "β" → "&beta;", "γ" → "&gamma;", "Γ" → "&Gamma;",
    "δ" → "&delta;", "Δ" → "&Delta;", "ε" → "&epsilon;", "η" → "&eta;", "θ" → "&theta;",
    "Θ" → "&Theta;", "λ" → "&lambd;", "Λ" → "&Lambda;", "μ" → "&mu;", "ν" → "&nu;", "ρ" → "&rho;",
    "ϕ" → "&Phi;", "ω" → "&omega;", "ħ" → "&hbar;", "" → "\", "" → "'", "" → "\"}];
  Print["... ", summary];
  summary = StringReplace[summary,
    Shortest[StringExpression[
      protocol : ("pensieve" | "arXiv" | "http" | "https"),
      "://", url__, w : (( "." | "," | ")" ... ) ~~ (Whitespace | EndOfString)
    ]] →
    Switch[protocol,
      "pensieve", (
        suffix = If[StringTake[url, -1] === "/", "index.html", ""];
        StringExpression[
          "<a href=\"", LinkToRoot, url, suffix, "\">", protocol, "://", url, "</a>", w
        ]
      ),
      "arXiv", (
        StringExpression[
          "<a href=\"https://arxiv.org/abs/\", url, "\">", protocol, "://", url, "</a>", w
        ]
      ),
      _, StringExpression[
        "<a href=\"", protocol, "://", url, "\">", protocol, "://", url, "</a>", w
      ]
    ]
  ];
  AppendTo[toc, StringDrop[Last@SplitName, -3] → summary]
];
Put[toc, TOCFilename];
];
PDFFilename

```

```

In[*]:= PensieveAssemble["nb"] := Module[
  {legits, nbdirs, orphans, nbdir, files},
  legit = NB2PDF /@ FileNames["*.nb", {PensieveDirectory}, Infinity];
  (* Delete orphaned PDF files *)
  nbdirs = Select[
    FileNames["nb", {PensieveDirectory}, Infinity],
    (FileType[#] === Directory) &
  ];
  orphans = Complement[
    Flatten[FileNames["*", {#}] & /@ nbdirs],
    legit,
    FileNames["index.html" | "TOC.m", {PensieveDirectory}, Infinity]
  ];
  DeleteFile[orphans];
  (* Delete empty nb directories *)
  Do[
    files = Select[
      FileNames["*", {nbdir}],
      (! MemberQ[{"index.html", "TOC.m"}, Last[FileNameSplit[#]]]) &
    ];
    If[files == {},
      Print["nb: Deleting ", nbdir];
      DeleteDirectory[nbdir, DeleteContents -> True]
    ],
    {nbdir, nbdirs}
  ]
];
End[];
EndPackage[]

```

For .one files

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

```

BeginPackage["Pensieve`"]; 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], "ByDate" | "ByTheme" | "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, "."],
False && 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]]
];
OneNoteFilename
];
OnePage2PDF[desc_List, dir_String] := Module[
  {ID, name, dateTime, lastModifiedTime, pdffilename, spdffilename, header, url, cl},
  {ID, name, dateTime, lastModifiedTime} = {"ID", "name", "dateTime", "lastModifiedTime"} /. desc;
  pdffilename = ToFileName[dir,
    KosherFilename[name] <> ".pdf"
  ];
  spdffilename = StringReplace[StripDir[pdffilename, PensieveDirectory], "\\\" -> "/"];
  If[
    Or[
      FileType[pdffilename] === None,
      AbsoluteTime[DatePlus[FileDate[pdffilename], {- $TimeZone, "Hour"}]] <
        AbsoluteTime[lastModifiedTime]
    ],
    Print["one: ", StripDir[pdffilename, PensieveDirectory]];
    If[FileType[pdffilename] != None, DeleteFile[pdffilename]];
    OneNoteLink@Publish[ID, pdffilename, 3, ""];
    header = StringJoin[
      "Dror Bar-Natan: ", PensieveName, ": ",
      StringJoin @@
        ((# <> ": ") & /@ Drop[FileNameSplit[StripDir[pdffilename, PensieveDirectory]], -2])
    ];
    url = PensieveURL <> spdffilename;
    cl = StringJoin[
      "c:\\drorbn\\bin\\cpdf.exe ", pdffilename,
      " -add-text \"", url, "\" -font-size 6 -topright 10 ",
      " AND -range 1 -add-text \"", header, "\" -font-size 10 -topleft 25 ",
      " -o ", pdffilename
    ];
    If[! StringMatchQ[spdffilename, DoNotStamp], Read@("! " <> cl)];
  ];
  pdffilename
];
PensieveAssemble["one"] := If[! NETObjectQ[OneNoteLink],
  $Failed, One2PDF /@ FileNames["*.one", {PensieveDirectory}, Infinity];];
PensieveAssemble["one"] := If[! NETObjectQ[OneNoteLink], $Failed,

```

```

One2PDF /@ FileNames [{"*.one", {PensieveDirectory}, Infinity}
];
End[];
EndPackage[]

```

Make Thumbnails

```
In[*]:= BeginPackage["Pensieve`"]; Begin["`Private`"];
```

```
In[*]:= ImageTypes = {"jpg", "JPG", "jpeg", "JPEG", "gif", "GIF", "png", "PNG"} ∪
  (VideoTypes = {"mp4", "MP4", "avi", "AVI", "ogg", "OGG"}); (* must be consistent with index.js *)
```

```
In[*]:= ThumbPlace[ImageFilename_String] := Module[{SplitName, ThumbDir, exttext},
  SplitName = StringSplit[ImageFilename, {"\\", "/"}];
  exttext = If[MemberQ[VideoTypes, FileExtension[ImageFilename]], ".gif", ""];
  {ThumbDir = ToFileName[Append[Most@SplitName, "thumbs"]],
  ToFileName[ThumbDir, Last[SplitName] <> exttext]};

```

```
In[*]:= Thumbnail2ImageName[thumbnail_String] :=
  StringReplace["." ~~ (vt : Alternatives @@ VideoTypes) ~~ ".gif" ~~ EndOfString => "." <> vt] @ thumbnail;
```

```

MakeThumb[ImageFilename_String] := Module[
  {SplitName, ThumbFilename, ThumbDir, img},
  {ThumbDir, ThumbFilename} = ThumbPlace[ImageFilename];
  SplitName = StringSplit[ImageFilename, {"\\", "/"}];
  If[
    And[
      Length[SplitName] < 2 || !MemberQ[{"thumbs", "previews"}, SplitName[[-2]],
      Or[
        FileType[ThumbFilename] === None,
        AbsoluteTime[FileDate[ThumbFilename]] < AbsoluteTime[FileDate[ImageFilename]]
      ]
    ],
    Print["thumbs: ", StripDir[ThumbFilename, PensieveDirectory]];
    If[FileType[ThumbDir] === None, CreateDirectory[ThumbDir]];
    If[MemberQ[VideoTypes, FileExtension[ImageFilename]],
      Export[ThumbFilename, ImageResize[#, 400] & /@ VideoFrameList[Import[ImageFilename], 10],
        "DisplayDurations" → 0.5, "AnimationRepetitions" → ∞],
      (* Else *) img = Import[ImageFilename];
      If[Head[img] === List, img = img[[Round[Length[img] / 2]]];
      Export[ThumbFilename, ImageResize[img, {400}]];
    ]
  ];
  ThumbFilename
];

```

```

In[*]:= PensieveAssemble["thumbs"] := Module[
  {legits, thumbsdirs, orphans, files, thumbsdir},
  Print["Making thumbs..."];
  legit = MakeThumb /@ FileNames[
    {"*." <> #} & /@ (Alternatives @@ ImageTypes), {PensieveDirectory}, Infinity
  ];
  thumbsdirs = Select[
    FileNames["thumbs", {PensieveDirectory}, Infinity],
    (FileType[#] === Directory) &
  ];
  orphans = Complement[
    Flatten[FileNames["*", {#}] & /@ thumbsdirs],
    legit,
    FileNames["index.html", {PensieveDirectory}, Infinity]
  ];
  DeleteFile[orphans];
  (* Delete empty thumbs directories *)
  Do[
    files = Select[
      FileNames["*", {thumbsdir}],
      (! MemberQ[{"index.html"}, Last[FileNameSplit[#]]]) &
    ];
    If[files == {},
      Print["thumbs: Deleting ", thumbsdir];
      DeleteDirectory[thumbsdir, DeleteContents -> True]
    ],
    {thumbsdir, thumbsdirs}
  ]
];

```

```

In[*]:= End[]; EndPackage[]

```

Make Previews

```

In[*]:= BeginPackage["Pensieve`"]; Begin["`Private`"];

```

```

In[*]:= ImageTypes = {"jpg", "JPG", "jpeg", "JPEG", "gif", "GIF", "png", "PNG"} ∪
  (VideoTypes = {"mp4", "MP4", "avi", "AVI", "ogg", "OGG"}); (* must be consistent with index.js *)

```

```

In[*]:= PreviewPlace[ImageFilename_String] := Module[{SplitName, PreviewDir, extext},
  SplitName = StringSplit[ImageFilename, {"\\", "/" }];
  extext = If[MemberQ[VideoTypes, FileExtension[ImageFilename]], ".gif", ""];
  {PreviewDir = ToFileName[Append[Most@SplitName, "previews"]],
   ToFileName[PreviewDir, Last[SplitName] <> extext]}];

```

```

In[*]:= Previewname2Imagename[previewname_String] :=
  StringReplace["." ~ (vt : Alternatives @@ VideoTypes) ~ ".gif" ~ EndOfString -> "." <> vt] @ previewname;

```



```

MakePreview[ImageFilename_String] := Module[
  {SplitName, PreviewFilename, PreviewDir, img},
  {PreviewDir, PreviewFilename} = PreviewPlace[ImageFilename];
  SplitName = StringSplit[ImageFilename, {"\\", "/" }];
  If[
    And[
      Length[SplitName] < 2 || !MemberQ[{"thumbs", "previews"}, SplitName[[-2]],
      Or[
        FileType[PreviewFilename] === None,
        AbsoluteTime[FileDate[PreviewFilename]] < AbsoluteTime[FileDate[ImageFilename]]
      ]
    ],
    img = Import[ImageFilename];
    If[MemberQ[VideoTypes, FileExtension[ImageFilename]] ^ Max[ImageDimensions@img] > 1920,
      If[FileType[PreviewDir] === None, CreateDirectory[PreviewDir]];
      Export[PreviewFilename, ImageResize[#, 1920] & /@ VideoFrameList[img, 10],
        "DisplayDurations" → 0.5, "AnimationRepetitions" → ∞];
      Print["previews: ", StripDir[PreviewFilename, PensieveDirectory]],
      (* Else *)
      If[Head[img] === List, img = img[Round[Length[img] / 2]];
      If[Max[ImageDimensions@img] > 1920,
        If[FileType[PreviewDir] === None, CreateDirectory[PreviewDir]];
        Export[PreviewFilename, ImageResize[img, {1920}]];
        Print["previews: ", StripDir[PreviewFilename, PensieveDirectory]],
        (* Else *)
        CreateFile[PreviewFilename, OverwriteTarget → True]
      ]
    ]
  ];
  PreviewFilename
];

```

```

PensieveAssemble["previews"] := Module[
  {legits, previewsdirs, orphans, files, previewsdir},
  Print["Making previews..."];
  legit = MakePreview /@ FileNames[
    {"*." <> #} & /@ (Alternatives @@ ImageTypes), {PensieveDirectory}, Infinity
  ];
  previewsdirs = Select[
    FileNames["previews", {PensieveDirectory}, Infinity],
    (FileType[#] === Directory) &
  ];
  orphans = Complement[
    Flatten[FileNames["*", {#}] & /@ previewsdirs],
    legit,
    FileNames["index.html", {PensieveDirectory}, Infinity]
  ];
  DeleteFile[orphans];
  (* Delete empty previews directories *)
  Do[
    files = Select[
      FileNames["*", {previewsdir}],
      (! MemberQ[{"index.html"}, Last[FileNameSplit[#]]) &
    ];
    If[files == {},
      Print["previews: Deleting ", previewsdir];
      DeleteDirectory[previewsdir, DeleteContents -> True]
    ],
    {previewsdir, previewsdirs}
  ]
];

```

```
In[*]:= End[]; EndPackage[]
```

Template Extraction

```

In[*]:= BeginPackage["Pensieve`"]; Begin["`Private`"];
InitializeTemplates[] := (
  Clear[ExtractTemplate];
  ExtractTemplate[tn_String] := ExtractTemplate[tn] = Import[
    FileNameJoin[{
      AcademicPensieveDirectory, "Projects", "AcademicPensieve",
      tn <> ".txt"
    }]
  ];
);
InitializeTemplates[];
End[];
EndPackage[]

```

Assemble Index Pages

```

BeginPackage["Pensieve`"];
Begin["`Private`"];

```

```

DateStringFormat =
{"Year", "-", "Month", "-", "Day", " ", "Hour24", ":", "Minute", ":", "Second"};
AssembleIndexPage[s_String] := Module[
{
dir, fulldir, InternalLinks, Customizations, parentdir, next, previous,
siblings, p, fname, t, rand, d, SplitPath, l, FullTitle, Title, LinkToRoot,
Navigator, TitleNotes, FootNotes, i, j, OneNoteTOC, OneNotePages, OneNoteDir,
OneNoteData, pl, NBDir, NBTOCFilename, NBTOC, MathematicaNotebooks, NBFileNames,
links, Links, ThumbsDir, Images, ImageFileNames, ImageComments, DirectoryListing,
UserDirectoryListing, DirectoryListingSorter, DirectoryListingSeparator,
SubfoldersAndShortcuts, style, FileListing, OtherFiles, last, htmlfile, count,
shortcuts, HPT, HTMLs, PDFs, TXTs, cleantexfiles, importfiles, SubfolderCount, FileCount
},
dir = StripRootDir[s];
fulldir = ToFileName[PensieveDirectory, dir];
InternalLinks = {};
Customizations = ToFileName[fulldir, "index.m"];
If[FileType[Customizations] === File,
SetDirectory[fulldir];
Customizations = Get[Customizations];
If[Head[Customizations] != List, Customizations = {}];
importfiles = "ImportFiles" /. Customizations /. "ImportFiles" → {};
If[importfiles != {},
ImportFile[ff_] := Module[
{f = FileNameTake[ff]},
If[FileExistsQ[ff] && (! FileExistsQ[f] || ! OrderedQ[{FileDate[ff], FileDate[f]}]),
Print["In ", dir, " copying ", ff → f];
If[FileExistsQ[f], DeleteFile[f]];
CopyFile[ff, f]
]
];
ImportFile /@ importfiles
];
ResetDirectory[],
(* else *) Customizations = {}
];
If["CleanTeX" /. Customizations /. "CleanTeX" → True,
SetDirectory[fulldir];
cleantexfiles =
FileNames[{"*.log", "*.bbl", "*.blg", "*.out", "*.bak", "*.toc", "*.upa", "*.upb"}];
If[cleantexfiles != {},
Print["In ", dir, " deleting ", cleantexfiles];
DeleteFile[cleantexfiles]
];
ResetDirectory[]
];
If[dir === "", next = previous = Last[FileNameSplit[PensieveDirectory]],
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[
  dir === "",
  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, {"\\", "/"}], Last[FileNameSplit[PensieveDirectory]]]
];
FullTitle = StringJoin[({# <> ": "}) & /@ SplitPath];
Title = Last[SplitPath];
LinkToRoot = StringJoin[Table["../", {l - 1}]];
Navigator = StringJoin[Table[
  StringJoin[
    "<a href=\"",
    StringJoin[Table["../", {l - i}]],
    "index.html\">" <> SplitPath[[i]] <> "</a>: "
  ],
  {i, l - 1}
]];
TitleNotes = "TitleNotes" /. Customizations /. "TitleNotes" -> "";
FootNotes = "FootNotes" /. Customizations /. "FootNotes" -> "";
FileListing = FileNames["*", {fulldir}];
OneNotePages = If[FileType[ToFileName[{fulldir, "one"}, "TOC.m"]] === None, "",
  AppendTo[InternalLinks, {"#NotebookPages", "Notebook Pages"}];
  OneNoteTOC = Get[ToFileName[{fulldir, "one"}, "TOC.m"]];
  OneNoteDir = FileNameJoin[Flatten[{
    LinkToRoot,
    Drop[
      FileNameSplit["path" /. OneNoteTOC[[2, 2]],
      Length[FileNameSplit[PensieveDirectory]]
    ]
  }]];
OneNoteData = OneNoteTOC[[2, 3, All, 2]];
count = 0;
StringJoin[
  "<a name=\"NotebookPages\"/><h2><a href=\"../\", previous,
  \"/index.html#NotebookPages\">&lt;&lt;/a> Notebook Pages <a href=\"../\",
  next, \"/index.html#NotebookPages\">&gt;&gt;/a></h2></a>\n",
  "NotebookPagesNotes" /. Customizations /. "NotebookPagesNotes" -> "",
  "<table class=sortable border=1 cellpadding=0>\n",
  "<tr><th>&nbsp;</th><th>Page</th><th>Created
  (UT)</th><th>Last Modified (UT)</th></tr>\n",
  StringJoin[StringJoin[
    "<tr align=left>\n",
    "<td align=right>", ToString[+count], "</td>\n",

```

```

StringJoin[
  " <td>",
  Which[
    (pL = ("pageLevel" /. #)) != "pageLevel", (
      pL /. {
        "1" → "", "2" → "---- ",
        "3" → "----- ", "4" → "----- ", _ → "----- "
      }
    ),
    ("isSubPage" /. #) == "true", "---- ",
    True, ""
  ],
  ],
  "<a href=\"one/",
  KoshersFilename["name" /. #],
  ".pdf\">",
  StringReplace["name" /. #, {"ж" → "&#1078;"}],
  "</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, "",
AppendTo[InternalLinks, {"#MathematicaNotebooks", "Mathematica Notebooks"}];
NBTOCFilename = ToFileName[NBDir, "TOC.m"];
NBTOC = If[FileType[NBTOCFilename] != File, {}, Get[NBTOCFilename]];
l = Length[
  NBFileNames =
  StringTake[StringReplace[#, NBDir → ""], {2, -5}] & /@ FileNames["*.pdf", NBDir]
];
count = 0;
StringJoin[
  "<a name=\"MathematicaNotebooks\"/><h2><a href=\"../\", previous,
  "/index.html#MathematicaNotebooks\">&lt;&lt;/a> Mathematica Notebooks <a href=\"../\",
  next, "/index.html#MathematicaNotebooks\">&gt;&gt;/a></h2></a>\n",
  "MathematicaNotebooksNotes" /. Customizations /. "MathematicaNotebooksNotes" → "",
  "<table class=sortable border=1 cellspacing=0>\n",
  "<tr><th>&nbsp;</th><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 align=right>", ToString[++count], "</td>\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"
]
];
links = Select[FileListing,
    (StringLength[#] ≥ 4 && ToLowerCase[StringTake[#, -4]] === ".url") &
];
links = Sort[{FileDate[#], Last[FileNameSplit[#]], LinkTarget[#]} & /@ links];
If[links === {},
    Links = "",
    (* else *) AppendTo[InternalLinks, {"#Links", "Links"}];
Links = StringJoin[
    "<a name=\"Links\"/><h2><a href=\"../\",
    previous, "/index.html#Links\">&lt;&lt;/a> Links <a href=\"../\",
    next, "/index.html#Links\">&gt;&gt;</a></h2></a>\n",
    "<ol>\n",
    StringJoin[
        "<li> Studied ", DateString[#[[1]],
        ": <a href=\"", #[[3]], "\">", StringDrop[#[[2]], -4], "</a>.\n"
    ] & /@ links,
    "</ol>\n"
]
];
ThumbsDir = ToFileName[fulldir, "thumbs"];
ImageFileNames = {};
ImageComments = "ImageComments" /. Customizations /. "ImageComments" -> {};
Images = If[FileType[ThumbsDir] != Directory, "",
    AppendTo[InternalLinks, {"#Images", "Images"}];
ImageFileNames = Thumbname2Imagename /@ (StringDrop[StringReplace[#, ThumbsDir -> ""], 1] & /@
    FileNames[{"*." <> #} & /@ (Alternatives @@ ImageTypes), ThumbsDir]);
StringJoin[
    "<a id=\"Images\" name=\"Images\"/><h2><a href=\"../\",
    previous, "/index.html#Images\">&lt;&lt;/a> Images <a href=\"../\",
    next, "/index.html#Images\">&gt;&gt;</a></h2></a>\n",
    "<div style=\"float:right; width: 60%;\"><div id=\"fotorama\" class=\"fotorama\"
    data-auto=\"false\" data-hash=\"true\" data-keyboard=\"true\"
    data-nav=\"false\" data-allowfullscreen=\"native\" data-width=\"100%\"
    data-ratio=\"16/9\" data-thumbheight=36 data-loop=\"true\">\n",
    StringJoin[
        "<a href=\"",
        If[
            FileExistsQ[ToFileName[fulldir, Last@PreviewPlace@#] &

```

```

FileByteCount[ToFileName[fulldir, Last@PreviewPlace@#]] > 0,
Last@PreviewPlace@#, #
], "\",
If[MemberQ[VideoTypes, FileExtension[#]], " data-video=\"true\"", ""],
" data-caption=\"<a target=_blank href=\", #, \">#10515;</a>",
If[(# /. ImageComments) === #, "", "&nbsp;"] <> (# /. ImageComments)],
"\"> <img src=\"thumbs/\", Last@FileNameSplit@Last@ThumbPlace@#, "\"> </a>\n"
] & /@ ImageFileNames,
"\n</div>
<table width=100% border=0 cellspacing=0 cellpadding=1 style=\"font-size: 150%\"><tr>
<td align=left><a href=\"javascript: show('<');\">previous</a></td>
<td align=center><a href=\"javascript: requestFullScreen();\">full screen</a></td>
<td align=right><a href=\"javascript: show('>');\">next</a></td>
</tr></table>
</div>
<p>\n\n",
k = 0; StringJoin[
"<div class=\"thumb\">",
"<a href=\"javascript:show(",
ToString[k++], ");\"><img class=\"thumbimg\" src=\"thumbs/\",
Last@FileNameSplit@Last@ThumbPlace@#, "\"></a>",
"<br><span style=\"font-size: 80%;\"><a href=\"\", #, \">\", #, \"</a>",
If[(# /. ImageComments) === #, "", "<br/>"] <> (# /. ImageComments)],
"</span></div>\n"
] & /@ ImageFileNames
]
];
DirectoryListing = Select[FileListing, FileType[#] == Directory &];
IgnoreShortcuts = "IgnoreShortcuts" /. Customizations /. "IgnoreShortcuts" -> {};
shortcuts = Select[
FileNames["*.lnk", {fulldir}],
!TrueQ[MemberQ[IgnoreShortcuts, Last[FileNameSplit[#]]]] &
];
shortcuts = {
Last[FileNameSplit[#]],
ShortcutTarget[#]
} & /@ shortcuts;
shortcuts = DeleteCases[shortcuts, {_, ""}];
OtherFiles = FileListing =
Last[StringSplit[#, {"\\", "/"}]] & /@ Complement[FileListing, DirectoryListing];
FileListing = Union[{"index.html"}, FileListing];
HTMLs = Complement[
Select[FileListing, MemberQ[{"html", "htm", "HTML", "HTM"}, FileExtension[#]] &],
{"index.html"}
];
PDFs = Select[FileListing, MemberQ[{"pdf", "PDF"}, FileExtension[#]] &];
TXTs = Select[FileListing, MemberQ[{"txt", "TXT"}, FileExtension[#]] &];
HPT = If[HTMLs === {} && PDFs === {} && TXTs === {}, "",
StringJoin[
"<p style=\"clear:left;\">",
"<div style=\"float: left; font-size: 150%; font-weight: bold;\">\n",
" <a name=\"HPT\"/><a href=\"../\", previous, "/index.html#HPT\">&lt;&lt;</a> ",
If[HTMLs != {}, "HTML", ""],
If[HTMLs != {} && PDFs != {}, " / ", ""],

```

```

If[PDFs != {}, "PDF", ""],
If[{HTMLs != {} || PDFs != {}} && TXTs != {}, " / ", ""],
If[TXTs != {}, "TXT", ""],
" <a href=\"../\", next, "/index.html#HPT\">&gt;</a>&nbsp;&nbsp;&nbsp;\n",
"</div>\n",
StringJoin[(
" <a href=\"\" <> # <> \"\">\" <> # <> \"</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ HTMLs],
If[HTMLs != {} && PDFs != {}, " /&nbsp;&nbsp;&nbsp;", ""],
StringJoin[(
" <a href=\"\" <> # <> \"\">\" <> # <> \"</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ PDFs],
If[{HTMLs != {} || PDFs != {}} && TXTs != {}, " /&nbsp;&nbsp;&nbsp;", ""],
StringJoin[(
" <a href=\"\" <> # <> \"\">\" <> # <> \"</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ TXTs]
]
];
FileCount = ToString@Length@FileListing;
FileListing = StringJoin[(
" <a href=\"\" <> # <> \"\">\" <> # <> \"</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ FileListing];
OtherFiles = Complement[OtherFiles,
Union[Flatten[{
"index.html", "index.m",
(# <> ".nb") & /@ NBFileNames,
ImageFileNames,
#[[2]] & /@ links,
First /@ shortcuts,
HTMLs, PDFs, TXTs
}]]];
OtherFiles = Select[OtherFiles, (StringTake[#, -1] != "~") &];
OtherFiles = If[OtherFiles === {}, "",
AppendTo[InternalLinks, {"#OtherFiles", "Other Files"}];
StringJoin[
"<a name=\"OtherFiles\"><h2><a href=\"../\",
previous, "/index.html#OtherFiles\">&lt;&lt;/a> Other Files <a href=\"../\",
next, "/index.html#OtherFiles\">&gt;&gt;</a></h2></a>\n",
StringJoin[(
" <a href=\"\" <> # <> \"\">\" <> # <> \"</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ OtherFiles]
]
];
DirectoryListing = Last[StringSplit[#, {"\\", "/"}]] & /@ DirectoryListing;
UserDirectoryListing = Complement[DirectoryListing, {"nb", "one", "thumbs", "previews"}];
UserDirectoryListing = Complement[UserDirectoryListing,
"ExcludeDirectories" /. Customizations /. "ExcludeDirectories" -> {}];
SubfoldersAndShortcuts = If[UserDirectoryListing === {} && shortcuts === {}, "",
DirectoryListingSorter =
"DirectoryListingSorter" /. Customizations /. "DirectoryListingSorter" -> Identity;
UserDirectoryListing = DirectoryListingSorter@UserDirectoryListing;

```



```

DirectoryListingSeparator = "DirectoryListingSeparator" /. Customizations /.
  "DirectoryListingSeparator" → "&nbsp;&nbsp;&nbsp;";
StringJoin[
  "<p style=\"clear:left;\>",
  "<div style=\"float: left; font-size: 150%; font-weight: bold;\>\n",
  "  <a name=\"SAS\"/><a href=\"../\", previous, "/index.html#SAS\">&lt;&lt;/a> ",
  If[UserDirectoryListing != {}, "Subfolders", ""],
  If[UserDirectoryListing != {} && shortcuts != {}, " / ", ""],
  If[shortcuts != {}, "Shortcuts", ""],
  "  <a href=\"../\", next, "/index.html#SAS\">&gt;&gt;</a>&nbsp;&nbsp;&nbsp;\n",
  "</div>\n",
  StringJoin[(
    style = If[# == ThisMonth, "background-color: yellow;", ""];
    "  <a style=\"font-size:120%; \"< style<> \"\" href=\"\"< #<> "/index.html\">"<>
      StringReplace[#, "_" → "_&shy;"]<> "</a>"<> DirectoryListingSeparator <> "\n"
    ) & /@ UserDirectoryListing],
  If[UserDirectoryListing != {} && shortcuts != {}, " /&nbsp;&nbsp;&nbsp;", ""],
  StringJoin[StringJoin[
    "  <a href=\"\", LinkToRoot, StringReplace[#[[2]], "\\\" → "/"",
    If[FileType[FileNameJoin[{PensieveDirectory, #[[2]]}]] == Directory,
      "/index.html", ""
    ],
    "\\\">, StringReplace[#[[2]], "\\\" → ": ", "</a>&nbsp;&nbsp;&nbsp;\n"
  ] & /@ shortcuts]
]
];
SubfolderCount = ToString@Length@DirectoryListing;
DirectoryListing = StringJoin[(
  "  <a href=\"\"< #<> "/index.html\">"<> #<> "</a>&nbsp;&nbsp;&nbsp;\n"
) & /@ DirectoryListing];
InternalLinks = StringJoin[
  StringJoin["<a href=\"\", #[[1]], \"\">, #[[2]], "</a> | "] & /@ InternalLinks
];
InternalLinks = If[InternalLinks == "", "-", StringDrop[InternalLinks, -3]];
htmlfile = OpenWrite[fname];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["index.html"], {
    "<#FullTitle#>" → FullTitle,
    "<#dir#>" → StringReplace[dir, "\\\" → "/"",
    "<#Title#>" → Title,
    "<#TitleNotes#>" → TitleNotes,
    "<#FootNotes#>" → FootNotes,
    "<#next#>" → next,
    "<#previous#>" → previous,
    "<#Navigator#>" → Navigator,
    "<#LinkToRoot#>" → LinkToRoot,
    "<#OneNotePages#>" → OneNotePages,
    "<#MathematicaNotebooks#>" → MathematicaNotebooks,
    "<#Links#>" → Links,
    "<#ImageFileNames#>" → StringJoin[Riffle[{"\", #, "\""}] & /@ ImageFileNames, ", "],
    "<#Images#>" → Images,
    "<#OtherFiles#>" → OtherFiles,
    "<#DirectoryListing#>" → DirectoryListing,
    "<#SubfolderCount#>" → SubfolderCount,

```

```

    "<#FileListing#" → FileListing,
    "<#FileCount#" → FileCount,
    "<#SubfoldersAndShortcuts#" → SubfoldersAndShortcuts,
    "<#InternalLinks#" → InternalLinks,
    "<#HPT#" → HPT
  ]]
];
Print["index: ", StripDir[Close[htmlfile], PensieveDirectory]]
]
];
PensieveAssemble["indexes"] := (
  Print["Making indexes..."];
  AssembleIndexPage /@
    Select[FileNames["*", PensieveDirectory, Infinity], FileType[#] == Directory &];
  AssembleIndexPage[""];
);
End[];
EndPackage[]

```

Assemble random.html, About.html, RecentChanges.html, AllDocs.html, and ThisMonth.html.

```

In[ ]:= BeginPackage["Pensieve`"]; Begin["`Private`"];
WrappedURL[fname_String] := If[! MemberQ[ImageTypes, Last[StringSplit[fname, "."]]],
  fname,
  StringReplace[
    fname,
    d : Longest[___] ~~ "/" ~~ f : Shortest[Except["/"] ..] :> d ~~ "/index.html?im=" ~~ f
  ]
];
StatisticalWeight[fname_String] := 1;
StatisticalWeight[fname_String] /;
  StringMatchQ[fname, "Projects/PlanetHopf/Frames/" ~~ DigitCharacter .. ~~ ".png"] := 0.01;
StatisticalWeight[fname_String] /;
  StringMatchQ[fname, "Projects/PlanetHopf/Frames2/" ~~ DigitCharacter .. ~~ ".png"] := 0.01;
StatisticalWeight[fname_String] /;
  StringMatchQ[fname, "2013-04/Elephants/x-" ~~ DigitCharacter .. ~~ ".png"] := 3 / 25;
StatisticalWeight[fname_String] /;
  StringMatchQ[fname, "2013-04/Elephants/y-" ~~ DigitCharacter .. ~~ ".png"] := 3 / 48;
StatisticalWeight[fname_String] /;
  StringMatchQ[fname, "2013-04/Elephants/z-" ~~ DigitCharacter .. ~~ ".png"] := 3 / 31;
PensieveAssemble["random"] := Module[
  {db, s, DB, IndexCount, Z, htmlfile,
   DocCount, rdb, LinkTo, RDB, AllLinks, NumberOfLinks, AllLinksDB},
  db = FileNames[
    ("*" <> #) & /@ Alternatives[
      "agda", "avi", "docx", "dvi", "eps", "gif", "html", "jpg",
      "m", "mp3", "mp4", "odt", "ogg", "pdf", "png", "ps", "svg", "tex", "txt", "zip"
    ], {PensieveDirectory}, Infinity, IgnoreCase → True
  ];
  db = StringDrop[StringReplace[#, {PensieveDirectory → "", "\\\" → "/}], 1] & /@ db;
  db = Select[db, ! (
    s = FileNameSplit[#];
    Or[

```

```

StringMatchQ[#, DoNotIndex],
Length[s] ≥ 2 && MemberQ[{"thumbs", "previews"}, s[[-2]],
Length[s] ≥ 2 && MemberQ[{"one", "nb"}, s[[-2]] && s[[-1]] == "TOC.m",
s[[-1]] == "index.m"
]
) &];
IndexCount = Length[Select[db, (Last[FileNameSplit[#]] == "index.html") &]];
Z = Total[StatisticalWeight /@ db];
DB = StringDrop[StringJoin[
StringJoin[
"["", WrappedURL[#, "\", "\", ToString[StatisticalWeight[#]], "\", \n"
] & /@ db
], -2];
NumberOfLinks = Length[
AllLinks = FileNames["*.url", {PensieveDirectory}, Infinity]
];
Print["random.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "random.html"]];
WriteString[htmlfile,
StringReplace[ExtractTemplate["random.html"], {
"<#DB#>" → DB,
"<#Z#>" → ToString[Z],
"<#LinkFraction#>" → ToString[N[NumberOfLinks / (Z + NumberOfLinks)]]
}]
];
Close[htmlfile];
DocCount = Round[Z - IndexCount];
Print["About.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "About.html"]];
WriteString[htmlfile,
StringReplace[ExtractTemplate["About.html"], {
"<#DocCount#>" → ToString[NumberForm[DocCount, DigitBlock → 3]]
}]
];
Close[htmlfile];
rdb = Select[db, (Last[FileNameSplit[#]] ≠ "index.html") &];
rdb = Complement[rdb, {
"About.html", "random.html", "RecentChanges.html",
"AllDocs.html", "ThisMonth.html", "RandomLink.html", "RandomLinkTop.html",
"RandomLinkMain.html", "AllLinks.html", "PostMortems.html"
}];
rdb = Select[rdb, (StatisticalWeight[#] > Random[]) &];
rdb = Reverse[Sort[
{FileDate[FileNameJoin[{PensieveDirectory, #}]], #} & /@ rdb
]];
Print[DateString[#[[1]], " ", #[[2]] & /@ Reverse@Take[rdb, 16];
(* LinkTo[f_String] := StringJoin[
" <a href=\"", f, "\">", f, "</a>"
]; *)
LinkTo[f_String] := Module[
{sp, l, path, i},
l = Length[sp = FileNameSplit[f]];
path = "";
StringJoin @@ Table[

```

```

path = StringJoin[
  path, If[i > 1, "/", ""],
  If[i == 1 && MemberQ[ImageTypes, Last[StringSplit[sp[[1], "."]]],
    "index.html?im=" <> sp[[1],
    sp[[i]
  ]
];
StringJoin[
  If[i > 1, "/", ""],
  "<a href=\"", path, If[i < 1, "/index.html", ""], "\">",
  StringReplace[sp[[i], "_" → " "], "</a>"
],
{i, 1}
]
];
RDB[rdb_] := StringJoin[
  StringJoin[
    "<tr>",
    "<td>", StringReplace[DateString[#[[1]]], " " → "&nbsp;"], "</td>",
    "<td>", LinkTo[#[[2]]], "</td>",
    "<td align=right>",
    ToString[NumberForm[
      FileByteCount[FileNameJoin[{PensieveDirectory, #[[2]]}],
      DigitBlock → 3
    ]],
    "</td>",
    "</tr>\n"
  ] & /@ rdb
];
Print["RecentChanges.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "RecentChanges.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["RecentChanges.html"], {
    "<#Today#>" → DateString[],
    "<#DocCount#>" → ToString[NumberForm[DocCount, DigitBlock → 3]],
    "<#RDB#>" → RDB[Take[rdb, Min[256, Length@rdb]]]
  }]
];
Close[htmlfile];
Print["PostMortems.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "PostMortems.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["PostMortems.html"], {
    "<#POMO#>" →
      RDB[Select[rdb, StringMatchQ[ToLowerCase[#[[2]]], ___ ~~ "post" ~~ ___ ~~ "mortem" ~~ ___] &]]
  }]
];
Close[htmlfile];
Print["AllDocs.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "AllDocs.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["AllDocs.html"], {
    "<#Today#>" → DateString[],
    "<#DocCount#>" → ToString[NumberForm[DocCount, DigitBlock → 3]],

```

```

    "<#RDB#>" → RDB[rdb]
  }
];
Close[htmlfile];
Print["ThisMonth.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "ThisMonth.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["ThisMonth.html"], {
    "<#ThisMonth#>" → If[FileType[FileNameJoin[{PensieveDirectory, ThisMonth}]] === Directory,
      ThisMonth,
      Print["The folder ", ThisMonth,
        " does not exist, ThisMonth link redirected to pensieve root."]; "."
    ]
  }
];
Close[htmlfile];
Print["RandomLink.html..."];
htmlfile = OpenWrite[ToFileName[PensieveDirectory, "RandomLinkMain.html"]];
AllLinksDB = StringJoin @@ Flatten[{
  "\n",
  Riffle[
    Table[
      {
        "[",
        Riffle[
          {
            "\"",
            StringReplace[
              ToString[#, CharacterEncoding → "PrintableASCII"],
              "\"" → "'"
            ],
            ],
            "\""
          } & /@ FileNameSplit[StripRootDir[link]],
        ],
        ],
        ], \", \"",
        LinkTarget[link],
        "\""]
      },
      {link, AllLinks}
    ],
    ",\n"
  ],
  "\n"]
  }];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["RandomLinkMain.html"], {
    "<#AllLinksDB#>" → AllLinksDB
  }
];
Close[htmlfile];
Print[]; Print[
  "Random: N=", Length[db],
  ", Z=", Z,

```

```
    ", DocCount=", NumberForm[DocCount, DigitBlock → 3],  
    ", LinkCount=", Length[AllLinks],  
    ", and ThisMonth=", ThisMonth];  
  ]  
End[];  
EndPackage[]
```

Experiments