

Pensieve Assembly Programs

Pensieve Header: This notebook contains the programs that assemble the web version of my Academic Pensieve.

■ Assemble All

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

■ Utilities / General

```

BeginPackage["AcademicPensieve`"];
{StripRootDir, LinkTarget};
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",
  "\"\" -> "' '", "<" -> "(", ">" -> ")", "\\\" -> "!", "*" -> "$"
}];

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

ShortcutTarget[lnk_String] := Module[
  {l},
  l = StringCases[
    FromCharacterCode[BinaryReadList[lnk]],
    (AcademicPensieveDirectory <> "\\\" ) ~~
    Shortest[u__] ~~ FromCharacterCode[0] -> u
  ];
  If[Head[l] != List, "", First[l]]
];

LinkTarget[lnk_String] := First[StringCases[
  FromCharacterCode[BinaryReadList[lnk]],
  "URL=" ~~ Shortest[u__] ~~ "\r"... ~~ EndOfLine -> u
]];

End[]; EndPackage[]

```

■ For .nb files

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
NB2PDF[NotebookFilename_String] := Module[
  {SplitName, PDFFilename, PDFDir,
   nb, TOCFilename, toc, summary, l, LinkToRoot, suffix},
  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, RootDir]];
l = Length[StringSplit[StripDir[NotebookFilename, RootDir], {"\\", "/"}]];
LinkToRoot = StringJoin[Table["../", {l-1}]];
If[FileType[PDFDir] === None, CreateDirectory[PDFDir]];
NotebookPrint[
nb = NotebookOpen[NotebookFilename], PDFFilename
];
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[
NotebookGet[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, {
"\"" → "\"", "\"" → "\"", "'" → "'",
"β" → "&beta;", "λ" → "&lambda;", "μ" → "&mu;",
"ν" → "&nu;", "ω" → "&omega;", "θ" → "&theta;", "η" → "&eta;"
}];
Print["... ", summary];
summary = StringReplace[summary,
Shortest[StringExpression[
protocol: ("pensieve" | "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
    ]
),
-, StringExpression[
    "<a href=\"", protocol,
    "://", url, "\">", protocol, "://", url, "</a>", w
]
]
];
AppendTo[toc, StringDrop[Last@SplitName, -3] → summary]
];
Put[toc, TOCfilename];
If[! MemberQ[OpenNotebooks, nb], NotebookClose[nb]];
];
PDFfilename
];
AcademicPensieveAssemble["nb"] := Module[
{legits, nbdirs, orphans, nbdir, files},
If[$FrontEnd === Null,
Print["\nNo Mathematica Front End -- no action on .nb files!!\n"],
OpenNotebooks = Notebooks[];
legits = NB2PDF /@ FileNames["*.nb", {RootDir}, Infinity];
(* Delete orphaned PDF files *)
nbdirs = Select[
    FileNames["nb", {RootDir}, Infinity],
    (FileType[#] === Directory) &
];
orphans = Complement[
    Flatten[FileNames["*", {#}] & /@ nbdirs],
    legit,
    FileNames["index.html" | "TOC.m", {RootDir}, 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/-CallingNETFromMathematica.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], "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];
legit = OnePage2PDF[#, PDFDirectory] & /@ PageDescriptors;
Put[XML, tocfile = ToFileName[PDFDirectory, "TOC.m"]];
AppendTo[legit, 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,
  KoshierFilename[name] <> ".pdf"
];
If[
Or[
  FileType[pdffilename] === None,
  AbsoluteTime[DatePlus[FileDate[pdffilename], {- $TimeZone, "Hour"}]] <

```

```

        AbsoluteTime[lastModifiedTime]
    ],
    Print["one: ", StripDir[pdffilename, RootDir]];
    If[FileType[pdffilename] != None, DeleteFile[pdffilename]];
    OneNoteLink@Publish[ID, pdffilename, 3, ""];
    (* SetFileDate[pdffilename,
        1+AbsoluteTime[lastModifiedTime]+$TimeZone*3600] *)
];
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["thumbs: ", StripDir[ThumbFilename, RootDir]];
    If[FileType[ThumbDir] === None, CreateDirectory[ThumbDir]];
    Export[ThumbFilename, ImageResize[Import[ImageFilename], {200}]];
];
ThumbFilename
];
AcademicPensieveAssemble["thumbs"] := Module[
    {legits, thumbsdirs, orphans, files, thumbsdir},
    legit = MakeThumb /@ FileNames[ImageTypes, {RootDir}, Infinity];
    thumbsdirs = Select[
        FileNames["thumbs", {RootDir}, Infinity],
        (FileType[#] === Directory) &
    ];
    orphans = Complement[

```

```

    Flatten[FileNames["*", {#}] & /@ thumbsdirs],
    legits,
    FileNames["index.html", {RootDir}, 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}
];
End[]; EndPackage[]

```

■ Template Extraction

```

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

```

■ Assemble Index Pages

```

BeginPackage["AcademicPensieve`"]; 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, i, j, OneNoteTOC, OneNotePages, OneNoteDir,
    OneNoteData, pL, NBDir, NBTOCFilename, NBTOC, MathematicaNotebooks,
    NBFileNames, links, Links, ThumbsDir, Images, ImageFileNames,
    DirectoryListing, UserDirectoryListing, SubfoldersAndShortcuts,
    style, FileListing, OtherFiles, last, htmlfile, count, shortcuts
  },
  dir = StripRootDir[s];
  fulldir = ToFileName[RootDir, dir];

```

```

InternalLinks = {};
Customizations = ToFileName[fulldir, "index.m"];
If[FileType[Customizations] === File,
  SetDirectory[fulldir];
  Customizations = Get[Customizations];
  If[Head[Customizations] != List, Customizations = {}];
  ResetDirectory[],
  (* else *) 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["../", {l - 1}]];
  Navigator = StringJoin[Table[
    StringJoin[
      "<a href=\"",
      StringJoin[Table["../", {l - i}]],
      "index.html\">" <> SplitPath[[i]] <> "</a>: "
    ],
    {i, l - 1}
  ]];
  TitleNotes = "TitleNotes" /. Customizations /. "TitleNotes" -> "";
  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[AcademicPensieveDirectory]]
  ]
}]];
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;&nbsp;&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\">",
      "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, "",
  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;</h2></a>\n",
  "MathematicaNotebooksNotes" /. Customizations /.
  "MathematicaNotebooksNotes" → "",
  "<table class=sortable border=1 cellpadding=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"];
Images = If[FileType[ThumbsDir] != Directory, "",
  AppendTo[InternalLinks, {"#Images", "Images"}];
ImageFileNames = StringDrop[StringReplace[#, ThumbsDir -> ""], 1] & /@
  FileNames[ImageTypes, ThumbsDir];
StringJoin[
  "<a 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",
  "<table border=0 cellpadding=5>\n",
  StringJoin[
    i = 0;
    StringJoin[
      ++i;
      If[1 == (i~Mod~4), " <tr>", ""],
      "<td width=25% align=center><a href=\"",
      #, "\"><img src=\"thumbs/", #, "\">",
      "<br><span style=\"font-size: 80%;\">", #, "</span></a></td>",
      If[0 == (i~Mod~4), "</tr>\n", ""]
    ] & /@ ImageFileNames
  ],
  If[0 != (i~Mod~4), "</tr>\n", ""],
  "</table>\n"
]
];
DirectoryListing = Select[FileListing, FileType[#] == Directory &];
shortcuts = {
  Last[FileNameSplit[#]],
  ShortcutTarget[#]
} & /@ FileNames["*.lnk", {fulldir}];
shortcuts = DeleteCases[shortcuts, {_, ""}];
OtherFiles = FileListing = Last[StringSplit[#, {"\\", "/"}]] & /@
  Complement[FileListing, DirectoryListing];
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,
    #[[2]] & /@ links,
    First /@ shortcuts
  }]]];
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"}];
UserDirectoryListing = Complement[UserDirectoryListing,
  "ExcludeDirectories" /. Customizations /. "ExcludeDirectories" -> {}
];
SubfoldersAndShortcuts = If[UserDirectoryListing === {} && shortcuts === {}, "",
  StringJoin[
    "<a name=\"SAS\"/><h2><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></h2>\n",
    StringJoin[(
      If[# == ThisMonth,
        style = "style=\"background-color: yellow;\"",
        style = ""
      ]
    );
    " <a " <> style <> " href=\"" <>
    # <> \"/index.html\">" <> # <> "</a>&nbsp;&nbsp;&nbsp;\n"
  ) & /@ UserDirectoryListing],
  If[UserDirectoryListing != {} && shortcuts != {}, " /&nbsp;&nbsp;&nbsp;", ""],
  StringJoin[StringJoin[
    " <a href=\"" <> LinkToRoot, StringReplace#[[2]], "\\\" -> "/">,
    If[FileType[FileNameJoin[{RootDir, #[[2]]}]] === Directory,
      \"/index.html", ""
    ]
  ]
];

```

```

    ],
    "\"" > ", StringReplace#[[2]], "\"\" -> \": \", "</a>&nbsp;&nbsp;&nbsp;\n"
  ] & /@ shortcuts]
]
];
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,
    "<#next#" -> next,
    "<#previous#" -> previous,
    "<#Navigator#" -> Navigator,
    "<#LinkToRoot#" -> LinkToRoot,
    "<#OneNotePages#" -> OneNotePages,
    "<#MathematicaNotebooks#" -> MathematicaNotebooks,
    "<#Links#" -> Links,
    "<#Images#" -> Images,
    "<#OtherFiles#" -> OtherFiles,
    "<#FileListing#" -> FileListing,
    "<#DirectoryListing#" -> DirectoryListing,
    "<#SubfoldersAndShortcuts#" -> SubfoldersAndShortcuts,
    "<#InternalLinks#" -> InternalLinks
  }]
];
Print["index: ", StripDir[Close[htmlfile], RootDir]]
]
];
AcademicPensieveAssemble["indexes"] := (
  AssembleIndexPage /@
  Select[FileNames["*", RootDir, Infinity], FileType[#] == Directory &];
  AssembleIndexPage[""];
);
End[]; EndPackage[]

```

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

```

BeginPackage["AcademicPensieve`"]; Begin["`Private`"];
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;
AcademicPensieveAssemble["random"] := Module[
  {db, s, DB, IndexCount, Z, htmlfile, DocCount, rdb, LinkTo, RDB},
  db = FileNames[
    (*. "<>#) & /@ Alternatives[
      "agda", "docx", "gif",
      "html", "jpg", "m", "odt", "pdf", "png", "svg", "txt"
    ], {RootDir}, Infinity
  ];
  db = StringDrop[StringReplace[#, {RootDir → "", "\\\" → "/"}], 1] & /@ db;
  db = Select[db, !(
    s = FileNameSplit[#];
    Or[
      Length[s] ≥ 2 && MemberQ[{"thumbs"}, 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[
      "\\\"", #, "\", ", ToString[StatisticalWeight[#]], ",\n"
    ] & /@ db
  ], -2];
  htmlfile = OpenWrite[ToFileName[RootDir, "random.html"]];
  WriteString[htmlfile,
    StringReplace[ExtractTemplate["random.html"], {
      "<#DB#>" → DB, "<#Z#>" → ToString[Z]
    }]
  ];
  Close[htmlfile];
  DocCount = Round[Z - IndexCount];
  htmlfile = OpenWrite[ToFileName[RootDir, "About.html"]];
  WriteString[htmlfile,
    StringReplace[ExtractTemplate["About.html"], {
      "<#DocCount#>" → ToString[DocCount]
    }]
  ];
  Close[htmlfile];
  rdb = Select[db, (Last[FileNameSplit[#]] != "index.html") &];
  rdb = Complement[rdb, {
    "About.html", "random.html", "RecentChanges.html", "ThisMonth.html"
  }];
  rdb = Select[rdb, (StatisticalWeight[#] > Random[]) &];
  rdb = Reverse[Sort[
    {FileDate[FileNameJoin[{AcademicPensieveDirectory, #}]], #} & /@ rdb
  ]];
  Print /@ 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 = path <> If[i > 1, "/", ""] <> sp[[i]];
    StringJoin[
      If[i > 1, "/", ""],
      "<a href=\"", path, If[i < 1, "/index.html", ""], "\">",
      StringReplace[sp[[i]], "_" -> " "], "</a>"
    ],
    {i, l}
  ]
];
RDB = StringJoin[
  StringJoin[
    "<tr>",
    "<td>", StringReplace[DateString[#[[1]]], " " -> "&nbsp;"], "</td>",
    "<td>", LinkTo[#[[2]]], "</td>",
    "<td align=right>",
    ToString[NumberForm[
      FileByteCount[FileNameJoin[{AcademicPensieveDirectory, #[[2]]}]],
      DigitBlock -> 3
    ]],
    "</td>",
    "</tr>\n"
  ] & /@ rdb
];
htmlfile = OpenWrite[ToFileName[RootDir, "RecentChanges.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["RecentChanges.html"], {
    "<#DocCount#" -> ToString[DocCount],
    "<#RDB#" -> RDB
  }]
];
Close[htmlfile];
htmlfile = OpenWrite[ToFileName[RootDir, "ThisMonth.html"]];
WriteString[htmlfile,
  StringReplace[ExtractTemplate["ThisMonth.html"], {
    "<#ThisMonth#" -> ThisMonth
  }]
];
Close[htmlfile];
Print["random: N=", Length[db], ", Z=", Z,
  ", DocCount=", DocCount, ", and ThisMonth=", ThisMonth];
]
End[]; EndPackage[]

```

Experiments

- Do I need an image browser?

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