Make

In[ ]:=
```
Make::usage =
  "Make[targets, sources, Hold[action]] makes a target, or a list of targets, given sources,
    or a list of sources, in the style of the unix 'make' command.";
Make[target_String, sources_, action_Hold] := Make[Evaluate@{target}, sources, action];
Make[targets_, source_String, action__Hold] := Make[targets, Evaluate@{source}, action];
Make[targets_List, sources_List, action_Hold] := Module[{},
   If[
    (And @@ ((FileType[#] =!= None) & /@ sources)) &&
     Or[
      Or @@ ((FileType[#] === None) & /@ targets),
      Min[AbsoluteTime[FileDate[#]] & /@ targets] < Max[AbsoluteTime[FileDate[#]] & /@ sources]
     ],
    Print["Making ", targets, " ..."];
    ReleaseHold[action]
   ]
  ];
```

WordCloud

In[ ]:=
```
sources = {"index.m"};
target = "WordCloud.png";
```

In[ ]:=
```
DBNDictionaryWords = StringSplit[
   "aarhus abelian acknowledgements adjoint adjoints albert alekseev alexander antipode anton archibald
     artin arxiv associator associators bardakov basepoint behaviour berceanu bialgebra
     bialgebras bijection borromean brendle brochier cablings centres chern chterental chu
     claspers coadjoint cocommutative cocycle coface cofactor colour coloured colourful
     colourings colours combinatorially combinatorics componentwise conjecturally crans dancso
     det diffeomorphism drinfeld dror duflo enriquez equivariant etingof exp exponentiate
     fenn fheorem fibre flavours formulae framings frivial froof frove functionals functor
     functorial functoriality functors furusho gluings goussarov grothendieck grouplike
     habiro halacheva harinck hatcher haviv homfly homomorphic homomorphicity homonymous
     homotopic homotopies hopf ihx injective isometries isotopies isotopy jacobian kamnitzer
     kanenobu karene kashiwara kauffman kazhdan kishino kneissler knottings kohno kontsevich
     kricker kuperberg kurlin lescop leung lieberum linearization linearizations loday
     mccool meilhan meinrenken metrized milnor moded moding modulo multicategory multinary
     multiplicatively naot natan ohtsuki operad overcrossing overcrossings papadima parametrizing
     parentehesized parentetization parenthesization parenthesizations parenthetization
     perturbative planarity postfix preprint projectivization projectivizations proven
     quadrivalent quandle quandles reassociate reidemeister reutenauer rimanyi rolfsen
     roukema saito sanderson satoh sder selflinking semidirect semivirtual shima simons
     sinh skeleta skype subalgebra subalgebras subring surjection surjections surjective
     symmetrized tder teichmuller thurston torossian tr trivalence trivolution unbraided
     undercrossing undercrossings unfavourably unforbidden unignoring unipotent unital
     unitarity unitrivalent unknot unoriented usb valent vassiliev vergne verma versa vertices
     virtuals voldemort warmup watanabe wirings wirtinger wko zhang zsuzsanna zsuzsi"];
```

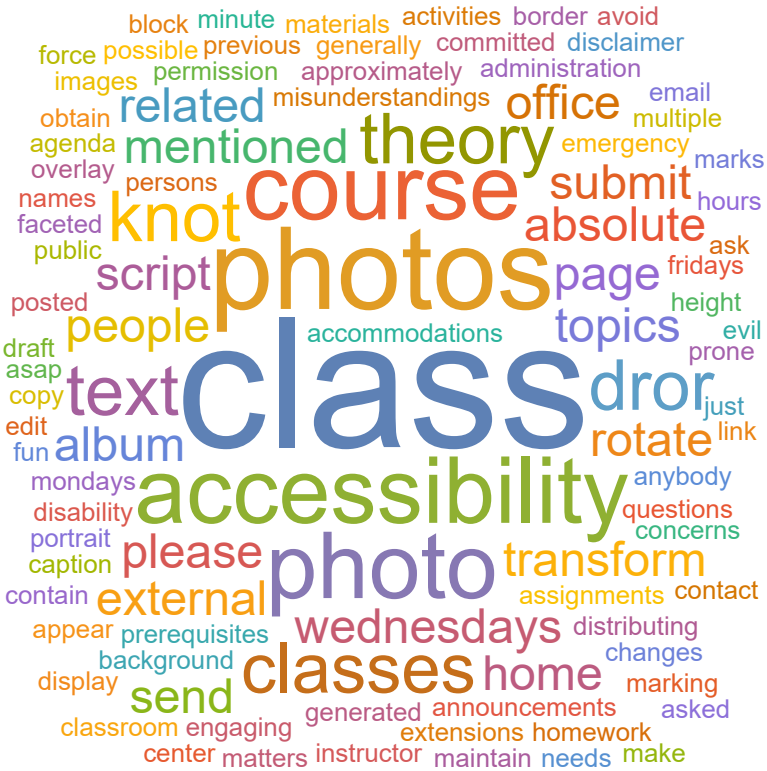
In[ ]:=
```
AvoidedWords =
  StringSplit["ac aft align aligned alpha bar begin beta bullet cali color corollary definition
     disc discussion dj em end equation eta example examples fa fig figure following font
     given hence ill indeed index left lemma let like math minus natan non note plus
     position proof red ref remark rh right rs section theorem ts tv ty way width xi"];
```

In[ ]:=
```
WordSubstitutions = {"algebras" → "algebra", "braids" → "braid", "cases" → "case",
   "crosses" → "cross", "crossings" → "crossing", "diagrams" → "diagram", "follows" → "follow",
   "generators" → "generator", "graphs" → "graph", "groups" → "group", "intervals" → "interval",
   "invariants" → "invariant", "knots" → "knot", "lines" → "line", "moves" → "move", "numbers" → "number",
   "objects" → "object", "options" → "option", "parts" → "part", "quotients" → "quotient",
   "relations" → "relation", "strands" → "strand", "tangles" → "tangle", "words" → "word"};
```

In[ ]:=
```
MakeWC[opts___] := Module[{words, words1, dict, T, dict1},
  words = ToLowerCase@DeleteStopwords@Flatten[
       StringSplit[TextWords[StringDelete[Longest["\\" ~~ LetterCharacter ...]]@ReadString[#]], "-"] & /@
        {"http://www.math.toronto.edu/~drorbn/classes/20-1350-KnotTheory/About.html"}
      ] /. WordSubstitutions;
  dict = Complement[Union[ToLowerCase@DictionaryLookup[], DBNDictionaryWords], AvoidedWords];
  dict1 = Dispatch[(# → T[#]) & /@ dict];
  words1 = Cases[words /. dict1, T[w_] :> w, {1}];
  WordCloud[words1, opts]
 ]
```

In[ ]:=
```
SetDirectory["C:\\drorbn\\AcademicPensieve\\Classes\\20-1350-KnotTheory\\Album"];
MakeWC[ImageSize → 400]
```

Out[ ]=



In[ ]:=
```
Make[target, sources, Hold[
  Export[target, MakeWC[ImageSize → 420]];
  MakeThumb@target;
 ]]
```

Output

```
{
 "TitleNotes" -> "Fall 2020 MAT1350 Public Album."
}
```

Out[•]= {TitleNotes → 2020–21 MAT257 Public Album.}