

# 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 clasplers
  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 theorem fibre
  flavours formulae framings trivial 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 mcool meilhan meinrenken metrized milnor moded moding
  modulo multicategory multinary multiplicatively naot natan ohtsuki operad overcrossing
  overcrossings papadima parametrizing parenthesized 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 univalent unknot unoriented usb valent vassiliev vergne verma versa
  vertices virtuals voldemort warmup watanabe wirings wirtinger wko zhang zszuzanna zszuzi"];
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"];

```

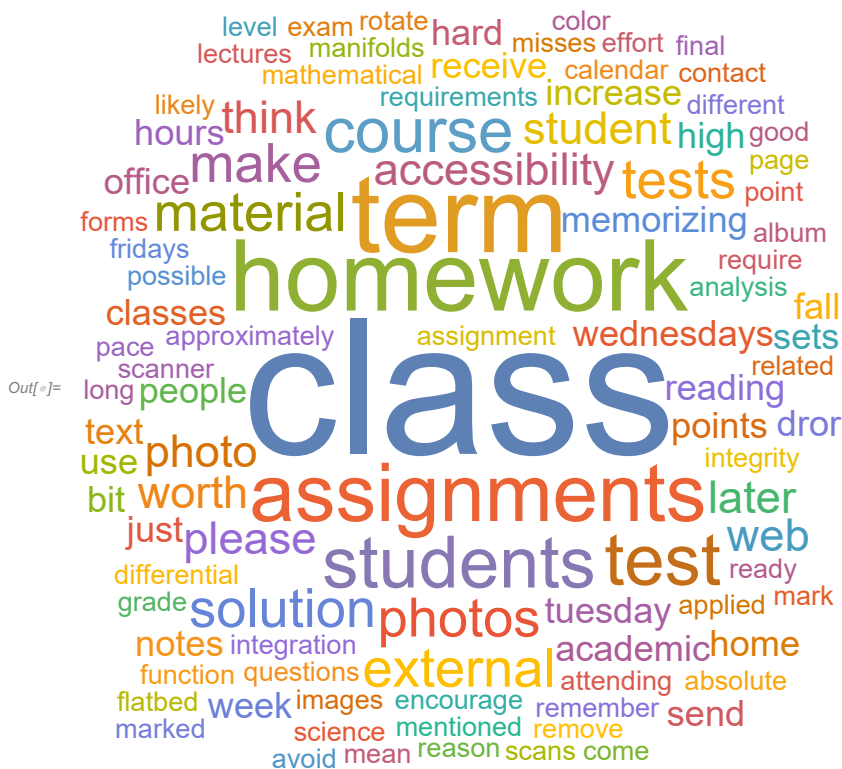
```

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/2021-257-AnalysisII/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\\2021-257-AnalysisII\\Album"];
MakeWC[ImageSize → 400]

```



```

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

```

## Collage

```

In[ ]:= SetDirectory["C:\\drorbn\\AcademicPensieve\\Classes\\2021-257-AnalysisII\\Album"];

In[ ]:= FileNames[{"*.jpg", "*.jpeg", "*.png", "*.gif"}]

Out[ ]:= {Collage.jpg, Gadgets.jpg, MyBookshelf.jpg,
  Painting-Andrew-Ho.jpeg, Process-Andrew-Ho.png, Team.jpg, WordCloud.png}

```

```
In[*]:= target = "Collage.jpg";  
sources = Complement[FileNames[{"*.jpg", "*.jpeg", "*.png", "*.gif"}], {target}];  
  
In[*]:= Export[target, ImageCollage[  
  1 → ImagePad[#, 8, White] & /@ ImageCrop /@ Import /@ sources,  
  "Fit", {2400, 2400},  
  Method → "ClosestPacking", Background → White, Padding → Red, ImagePadding → 4  
  ]];  
MakeThumb@target  
Out[*]:= MakeThumb[Collage.jpg]
```

```
In[*]:= Make[target, sources, Hold[  
  Export[target, ImageCollage[  
    1 → ImagePad[#, 8, White] & /@ ImageCrop /@ Import /@ sources,  
    "Fit", {2400, 2400},  
    Method → "ClosestPacking", Background → White, Padding → Red, ImagePadding → 4  
  ]  
  ]];  
MakeThumb@target;  
]]
```

## Output

```
In[*]:= {  
  "TitleNotes" -> "2020-21 MAT257 Public Album."  
}
```

```
Out[*]:= {TitleNotes → 2020-21 MAT257 Public Album.}
```