Example of Text Analysis

```
In[•]:= webpage = Import["https://www.fooledbyrandomness.com/notebook.htm"];

In[•]:= WordCloud[DeleteStopwords[webpage]]
Out[•]=
```

This word cloud makes me think that this conversation is not very interesting. There seems to be no real discussion other than an assortment of people and topics

```
In[•]:= contents = TextContents[webpage, VerifyInterpretation → True];

In[•]:= counts = ReverseSort@CountsBy[contents, Type &]
```

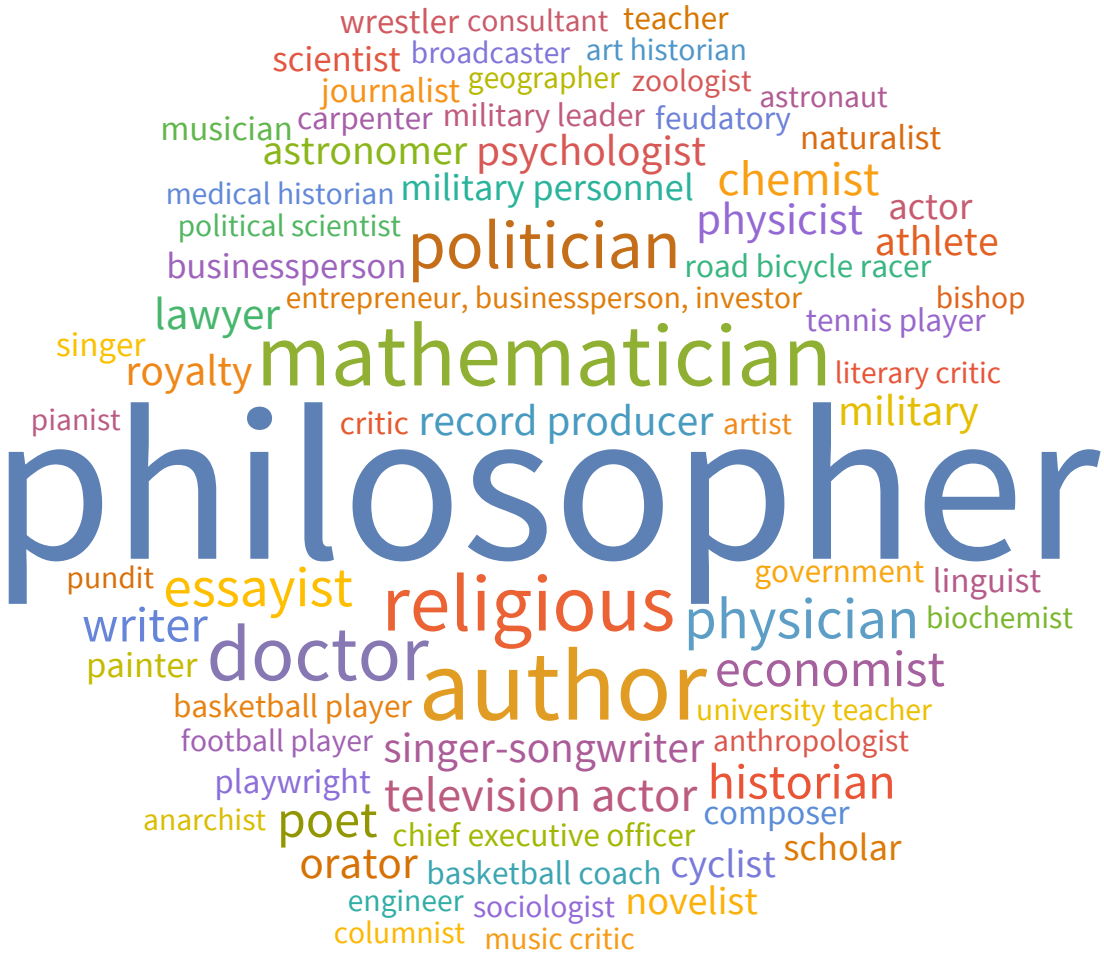
Out[•]=

```
Dataset[Association[Person → 423, Language → 317, Occupation → 292, GivenName → 262,
  NonperiodicTiling → 214, Country → 191, Quantity → 182, Religion → 152,
  GovernmentAgency → 122, Date → 117, Surname → 107, Nationality → 104, Year → 99,
  PhysicalConstant → 96, TopologicalSpaceType → 84, Color → 76, City → 71, Unit → 66,
  OrdinalNumber → 60, Organization → 58, HistoricalCountry → 47, Book → 27, Time → 24,
  AstronomicalObjectType → 21, SportEvent → 16, AdministrativeDivision → 15,
  TropicalStorm → 13, Financial → 12, Company → 11, Food → 11, GeographicRegion → 11,
  AstronomicalObject → 10, CurrencyAmount → 10, FoodManufacturer → 9,
  EthnicGroup → 9, Species → 8, University → 8, Disease → 8, Continent → 7, Plant → 6,
  Periodical → 6, Planet → 6, Season → 6, Island → 5, Element → 5, WaterBodyType → 5,
  Ocean → 4, FictionalCharacter → 3, Building → 3, WritingScript → 3, Movie → 3,
  Airport → 2, Mythology → 2, ProgrammingLanguage → 2, MeasurementDevice → 1,
  MilitaryConflict → 1, Cemetery → 1, Neighborhood → 1, Protein → 1, USState → 1,
  HistoricalEvent → 1, FileFormat → 1, Mountain → 1, Alphabet → 1, Gene → 1,
  SportObject → 1, PlanetaryMoon → 1, PersonTitle → 1, MathematicalFunction → 1,
  Mineral → 1, DogBreed → 1, Cave → 1, MusicWork → 1, BoardGame → 1],
 TypeSystem`Assoc[TypeSystem`Atom[String], TypeSystem`Atom[Integer],
  TypeSystem`AnyLength], Association[]]
```

In[•]:=
```
persons = Normal[Select[contents, Type === "Person" &][[All, "Interpretation"]]];
```

In[•]:=
```
WordCloud[Counts[Flatten@EntityValue[persons, EntityProperty["Person", "Occupation"]]],
 ImageSize → Large]
```

Out[ ]=

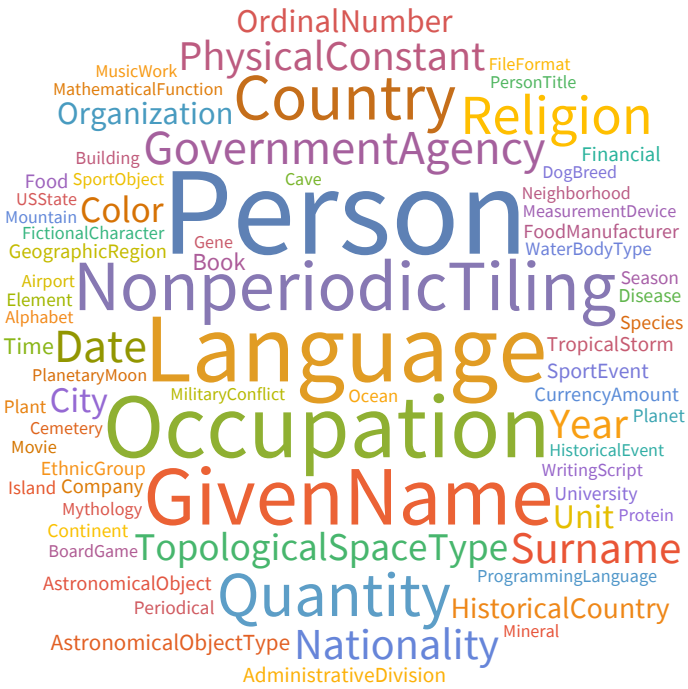


In[ ]:= **WordCloud[Counts[persons], ImageSize → 500]**

Out[•]=

In[•]:= **Show[WordCloud[counts]]**

Out[ ]=



In[ ]:= **countries = Normal[Select[contents, Type === "Country" &][[All, "Interpretation"]]]**

Out[•]=

{Greece, Greece, Greece, Italy, Italy, Spain, Jordan, Egypt, Lebanon, Lebanon, Syria, Lebanon, Greece, Greece, Armenia, Pakistan, Serbia, Croatia, Lebanon, Singapore, China, West Bank, Iran, Iran, Lebanon, United States, France, Austria, Russia, Cyprus, Armenia, Armenia, France, Spain, Portugal, Brazil, Japan, Japan, Syria, Saudi Arabia, France, Luxembourg, Austria, United Kingdom, France, Syria, France, Lebanon, Greece, France, Germany, Germany, Lebanon, United States, Switzerland, United States, Switzerland, China, Germany, France, United Kingdom, United Kingdom, United States, United States, United Kingdom, Philippines, Lebanon, India, Greece, Greece, Syria, Lebanon, Syria, Syria, Syria, Lebanon, Syria, India, Greece, Syria, Greece, Greece, Syria, Syria, Greece, Greece, Greece, France, Greece, France, France, France, France, France, France, France, Greece, Croatia, Serbia, Croatia, Lebanon, Romania, Serbia, Croatia, Russia, Bulgaria, Poland, Pakistan, Saudi Arabia, Malta, Turkey, Syria, France, India, France, Portugal, Brazil, Spain, Lebanon, Greece, Greece, Greece, Greece, Egypt, Syria, Greece, Greece, Syria, United States, Greece, Syria, Greece, Syria, Greece, Syria, Syria, Greece, Greece, Greece, Greece, Greece, Greece, Greece, Greece, Greece, France, Turkey, France, France, Sweden, Russia, Greece, Spain, Syria, France, France, France, Spain, United States, Lebanon, Lebanon, Poland, United Kingdom, France, China, China, China, Italy, Iraq, Egypt, Lebanon, Lebanon, Iraq, Saudi Arabia, Saudi Arabia, Saudi Arabia, Iran, United States, Lebanon, Colombia, Afghanistan, Iraq, Iraq, Lebanon, Greece, Greece, Lebanon, Sudan, Greece, Romania, Romania, Greece, Greece}

In[•]:= `WordCloud[countries, ImageSize → 500]`

Out[•]=

In[•]:=
```
WordCloud[Normal[Select[contents, Type === "MilitaryConflict" &][All, "Interpretation"]],
 ImageSize → Large]
```

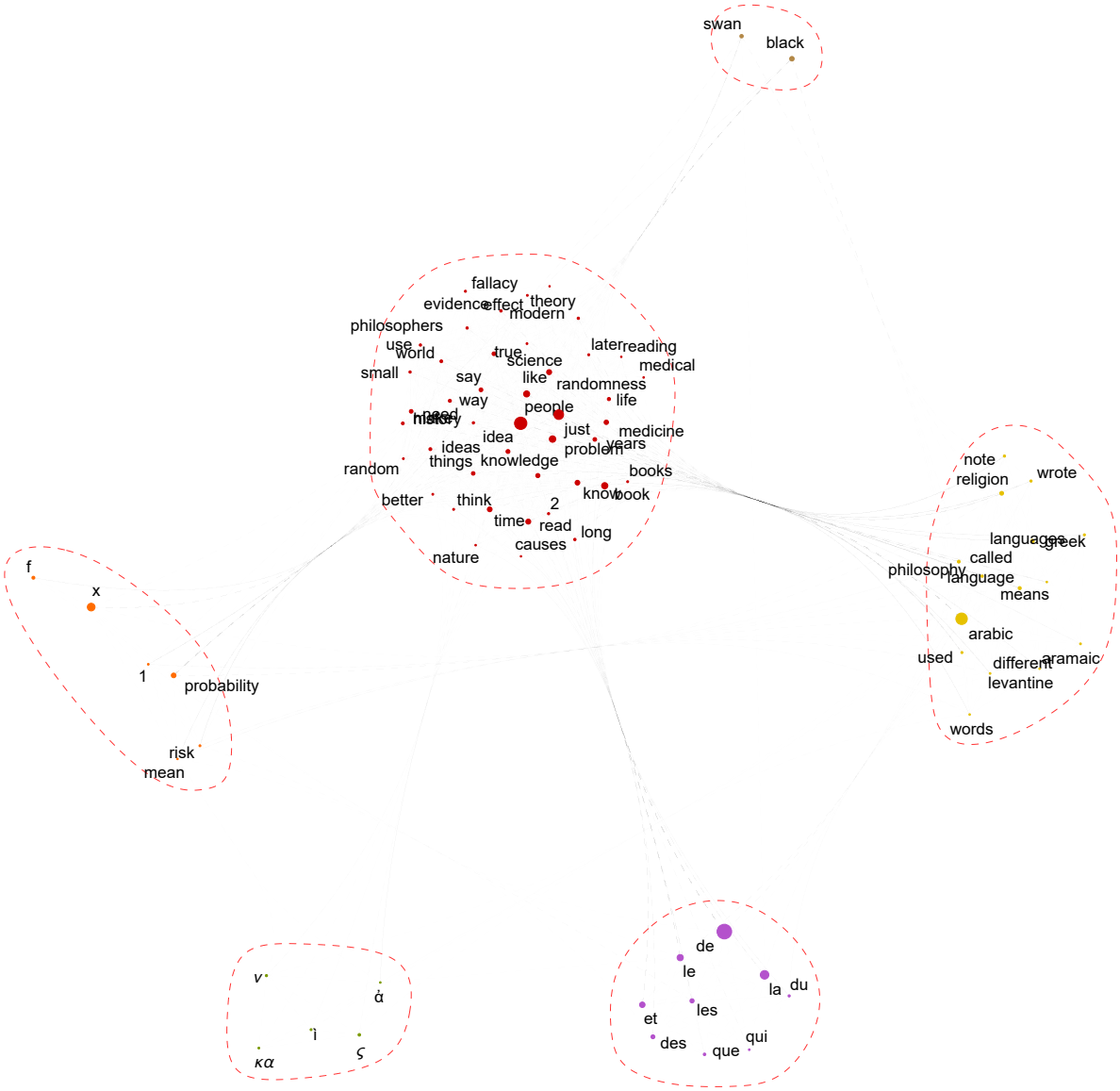
Out[•]=

KeywordGraph

```
In[ ]:= Options[KeywordsGraph] = {DirectedEdges → False, EdgeWeight → Automatic, "LowerCase" → True,
          "StopWords" → True, VertexLabels → Automatic, VertexWeight → Automatic};

        KeywordsGraph[text_String, number_Integer?Positive, blist_List : {},
          rlist_List : {}, opts : OptionsPattern[{KeywordsGraph, Graph}]] :=
         Module[{keycounts, keywords, edges, edgeCount, data =
            Replace[DeleteCases[TextWords[If[TrueQ[OptionValue["StopWords"]], DeleteStopwords,
                  Identity][If[TrueQ[OptionValue["LowerCase"]], ToLowerCase, Identity][text]]],
              Alternatives @@ blist], rlist, {1}]}, keycounts = Counts[data];
          Quiet[Check[keywords = TakeLargest[keycounts, number],
            Return[Failure["KeywordCount", <|"MessageTemplate" →
                 "Number of specified keywords `1` exceeds the actual number of keywords `2`.",
                "MessageParameters" → {number, Length[keycounts]}|>], Module]]];
          edges = Partition[Cases[data, Alternatives @@ Keys[keywords]], 2, 1];
          edgeCount = If[TrueQ[Replace[OptionValue[DirectedEdges], Automatic :> False]],
            KeySelect[Counts[DirectedEdge @@@ edges], #[[1]] ≠ #[[2]] &],
            KeySelect[Counts[Sort /@ UndirectedEdge @@@ edges], #[[1]] ≠ #[[2]] &]];
          Graph[Keys[keywords], Keys[edgeCount], FilterRules[Flatten[Join[{opts,
                VertexWeight → Replace[OptionValue[VertexWeight], Automatic :> Values[keywords]],
                EdgeWeight → Replace[OptionValue[EdgeWeight], Automatic :> Values[edgeCount]]},
              Options[KeywordsGraph]]], Options[Graph]]]]
```

```
In[ ]:= CommunityGraphPlot[KeywordsGraph[DeleteStopwords[webpage], 80,
          VertexSize → "VertexWeight", EdgeStyle → Directive[Black, Dashed, Opacity[0.01]]],
         CommunityBoundaryStyle → Directive[Red, Dashed, Opacity[0.8]],
         ImageSize → Full, GraphLayout → "RadialEmbedding"]
```

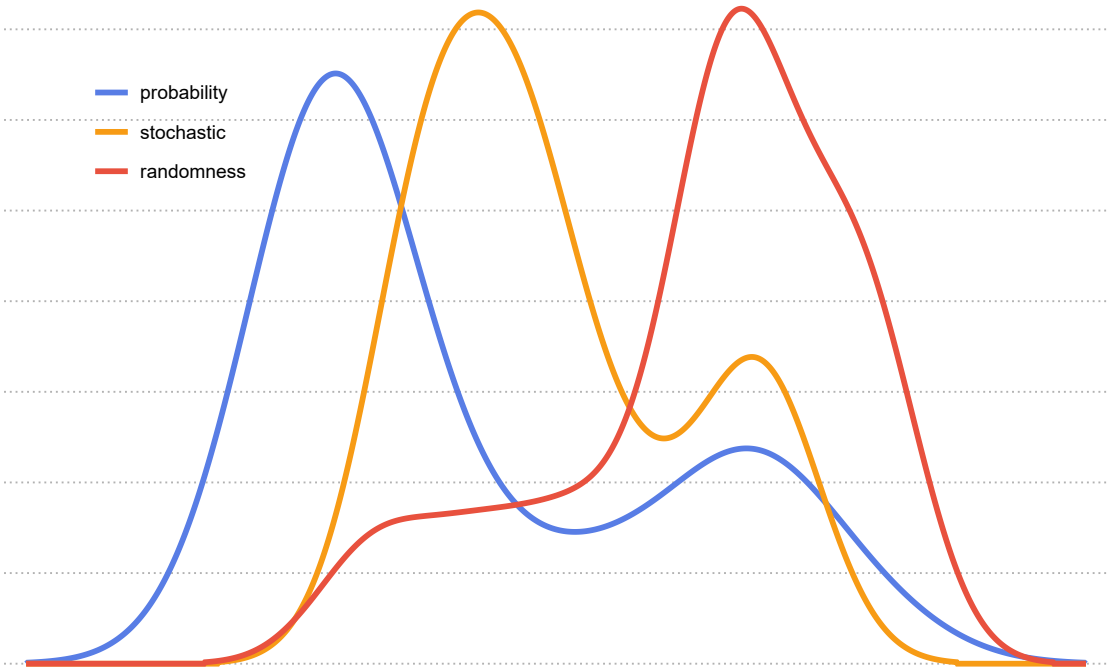
Out[•]=


swan
black
fallacy
evidence
effect
theory
modern
philosophers
use
world
true
science
later
reading
medical
small
say
like
randomness
life
way
people
history
just
medicine
idea
years
problem
ideas
things
knowledge
random
books
know
book
better
think
2
time
read
long
causes
nature
note
religion
wrote
languages
greek
called
philosophy
language
means
arabic
used
different
aramaic
levantine
words
f
x
1
probability
risk
mean
de
le
la
du
les
et
des
que
qui

KeywordPlot

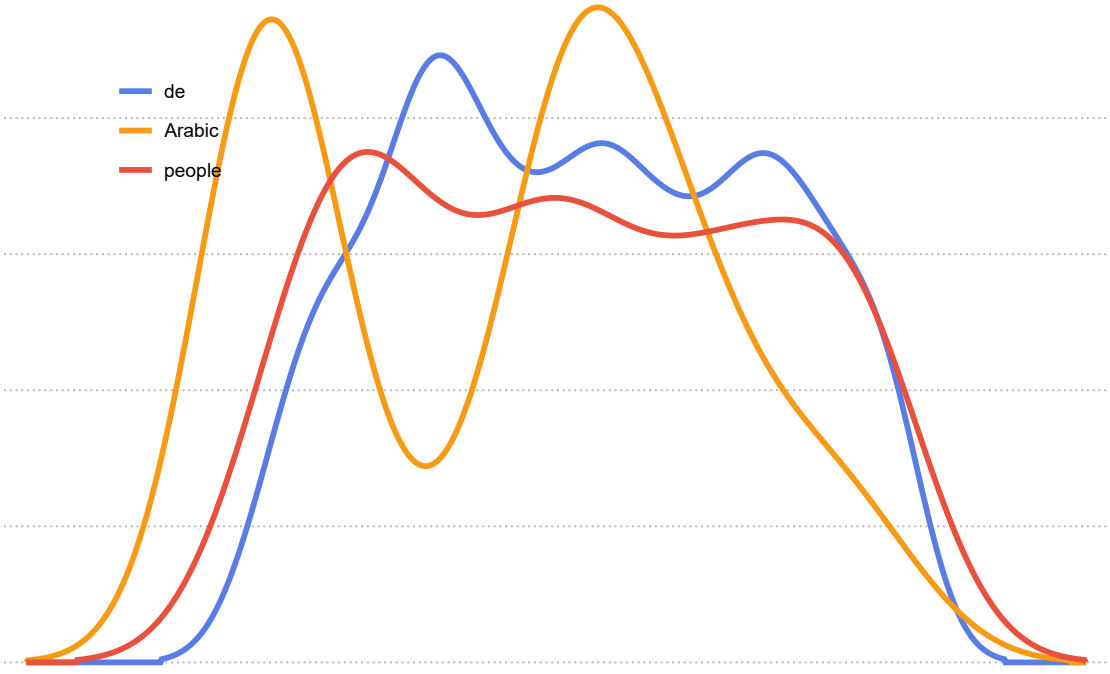
```
In[•]:= KeywordPlot // ClearAll;
        Options[KeywordPlot] =
          Options[SmoothHistogram] ~Join~ {"PlotFunction" → Automatic, "TopN" → All};
        KeywordPlot::nokwd = "Keyword \"`1`\" not found in text.";

        (*Operatorforms*)
        KeywordPlot[keyword_String, opts : OptionsPattern[]] := KeywordPlot[{keyword}, opts];
        KeywordPlot[keywords_List, opts : OptionsPattern[]] :=
          Function[text, KeywordPlot[text, keywords, opts]];

        (*Mainform*)
        KeywordPlot[text_String, keyword_String, opts : OptionsPattern[]] :=
          KeywordPlot[text, {keyword}, opts];
        KeywordPlot[text_String, keywords_List, opts : OptionsPattern[]] :=
         Module[{pltFun, topN, txt, kws, pos, badpos},
          pltFun = OptionValue["PlotFunction"] /. Automatic → SmoothHistogram;
          topN = OptionValue["TopN"];
          txt = RemoveDiacritics@ToLowerCase@text;
          kws = RemoveDiacritics@ToLowerCase@keywords;
          pos = StringPosition[txt, #][[All, 1]] & /@ kws;
          badpos = Position[pos, {}];
          If[badpos =!= {}, ResourceFunction["ResourceSystemMessage"][KeywordPlot::nokwd, #] & /@
            Extract[kws, badpos];
           If[Flatten[pos] == {}, Return@$Failed, (*nokeywordstoplot*)kws = Delete[kws, badpos];
            pos = Delete[pos, badpos];]];
          If[IntegerQ@topN, ord = OrderingBy[pos, -Length[#] &];
           pos = Take[pos[[ord]], UpTo@topN];
           kws = Take[kws[[ord]], UpTo@topN];];
          pltFun @@ {pos, Sequence @@ FilterRules[{opts}, Options[pltFun]],
            PlotLegends → kws, AspectRatio → 1 / GoldenRatio,
            PlotTheme → "Minimal", ImageSize → Automatic, Frame → None}]

In[•]:= keywords = {"probability", "stochastic", "randomness"};

        KeywordPlot[keywords, ImageSize → Large,
          PlotLegends → Placed[keywords, {{.15, .8}}], PlotTheme → "Business"]@webpage
```

Out[•]=



In[•]:=
```
KeywordPlot[
  Normal[Keys[TakeLargest[Counts[TextCases[DeleteStopwords[webpage], "Word"]], 3]]],
  ImageSize → Large, PlotLegends → Placed[
    Normal[Keys[TakeLargest[Counts[TextCases[DeleteStopwords[webpage], "Word"]], 3]]],
    {{.15, .8}}], PlotTheme → "Business"]@webpage
```

Out[•]=



In[•]:=
```
WordCloud[DeleteStopwords[webpage], ImageSize → Large]
```

Out[•]=

Nonfunctioning Resource Functions

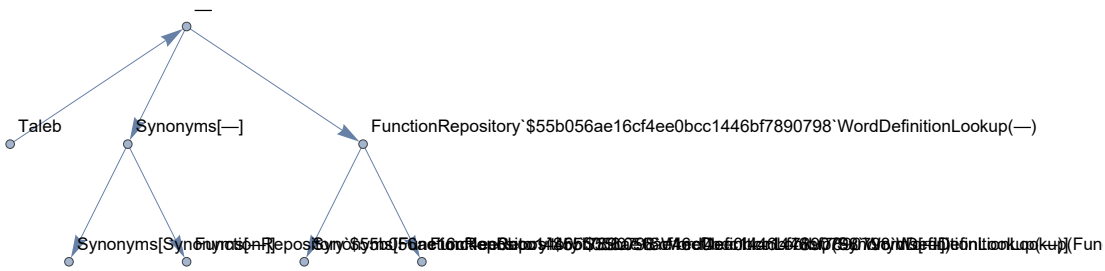
In[•]:=
```
ResourceFunction["ReadabilityScore"][webpage]
```

Out[•]=

11.6762

In[•]:=
```
NestGraph[{ResourceFunction["WordDefinitionLookup"], Synonyms} /* Through /* Flatten,
 "probability", 3, VertexLabels → "Name", ImageSize → Large]
```

Out[•]=



In[•]:= **ResourceFunction["SynonymGraph"]["probability", 2, ImageSize → Full]**

Out[•]=