Knowledge Representation in the Wolfram Language

See “Knowledge Representation & Access” in the Wolfram Language Guide

Wolfram Language Data

The Wolfram Language itself is knowledge represented in the Wolfram Language:

```
WolframLanguageData["Classes"]
```

{ Wolfram Language experimental symbols ,
Wolfram Language atomic functions , Wolfram Language autoevaluating symbols ,
Wolfram Language curryable symbols , all Wolfram Language symbols }

Our WolframLanguageData input appears to be identical to this EntityClassList input:

```
EntityClassList["WolframLanguageSymbol"]
```

{ Wolfram Language experimental symbols ,
Wolfram Language atomic functions , Wolfram Language autoevaluating symbols ,
Wolfram Language curryable symbols , all Wolfram Language symbols }

The output are instances of EntityClass. Let’s see a Shallow list of Wolfram Language experimental symbols:

```
EntityList[ Wolfram Language experimental symbols (Wolfram Language symbols) ] // Shallow
```

{ AnatomyForm , AnatomyPlot3D , Ask , AskAppend , AskConfirm ,
AskDisplay , AskedQ , AskedValue , AskFunction , AskTemplateDisplay , <<111>> }

Use Dataset to scroll/browse through all of the symbols:

```
Dataset[EntityList[ Wolfram Language experimental symbols (Wolfram Language symbols) ]]
```

AnatomyForm	AnatomyPlot3D	Ask	AskAppend	As
Autocomplete	AutocompletionFunction	BatchNormalizationLayer	BatchSize	Ba
ChannelListen	ChannelListener	ChannelListeners	ChannelObject	Ch
ContentLocationFunction	ContentObject	ConvolutionLayer	CreateChannel	Cr
DimensionReduce	DimensionReducerFunction	DimensionReduction	DisableFormatting	Do
FeatureDistance	FeatureExtract	FeatureExtraction	FeatureExtractor	Fe
FoldPairList	FormControl	GalleryView	HandlerFunctions	Ha
NetDecoder	NetEncoder	NetExtract	NetGraph	Ne
ReshapeLayer	ResourceData	ResourceObject	ResourceRemove	Re
Snippet	SoftmaxLayer	SourceLink	SummationLayer	Ta
TotalLayer	UpdateSearchIndex	URLSubmit	WordTranslation	Zo

Use Select to target a subset:

```
Select[
 EntityList[ Wolfram Language experimental symbols (Wolfram Language symbols) ],
 StringMatchQ[EntityValue[#, name ], "Feature*"] &
]
```

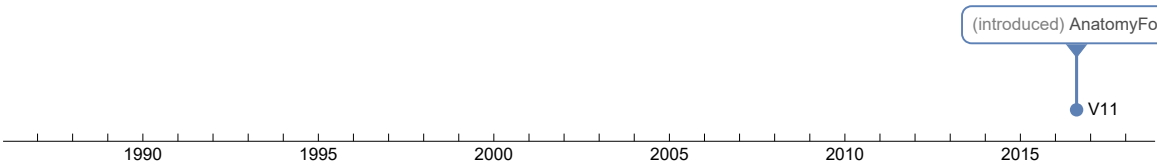
{ FeatureDistance , FeatureExtract ,
 FeatureExtraction , FeatureExtractor , FeatureExtractorFunction }

```
EntityProperties[ AnatomyForm (Wolfram Language symbol) ]
```

{ *attributes* , *common option values* , *date introduced* , *date last modified* , *dates modified* ,
 documentation basic examples , *documentation example counts* , *documentation example inputs* ,
 documentation example text , *eponymous people* , *equal-precedence symbols* , *external links* ,
 frequencies of usage , *full version introduced* , *full version last modified* , *full versions modified* ,
 functionality areas , *symbol background* , *keyboard shortcuts* , *link trails* , *memberships* ,
 name , *option names* , *options* , *plaintext usage* , *precedence ranks* , *ranks of usage* ,
 related entities , *related symbols* , *relationship community graph* , *relationship graph* ,
 short notations , *subject classifications* , *text strings* , *timeline* , *translations* ,
 typeset usage , *URL* , *version introduced* , *version last modified* , *versions modified* }

We see from the EntityProperties of the AnatomyForm Entity date-related properties that suggest that a TimelinePlot can be generated:

```
WolframLanguageData[ AnatomyForm (Wolfram Language symbol) , "Timeline"]
```



EntityValue from an EntityProperty:

```
EntityValue[ AnatomyForm (Wolfram Language symbol) , plaintext usage ]
```

```
AnatomyForm[g] is a graphics directive used in AnatomyPlot3D
  that specifies how anatomy entity-based graphics objects are to be
  drawn using the graphics directive or association of directives g.
```

```
EntityValue[ AnatomyForm (Wolfram Language symbol) , related symbols ]
```

```
{ AnatomyPlot3D , AnatomyData , HumanGrowthData , EntityValue }
```

We see that EntityValue is an explicit form of this equivalent expression:

```
AnatomyForm (Wolfram Language symbol) [ related symbols ]
```

```
{ AnatomyPlot3D , AnatomyData , HumanGrowthData , EntityValue }
```

Or the Input Text form:

```
Entity["WolframLanguageSymbol", "AnatomyForm"]["RelatedSymbols"]
```

```
{ AnatomyPlot3D , AnatomyData , HumanGrowthData , EntityValue }
```