Package 'stencilaschema'

October 14, 2022

Title Bindings for Stencila Schema

Description Provides R bindings for the Stencila Schema <https://schema.stenci.la>. This package is primarily aimed at R developers wanting to programmatically generate, or modify, executable documents.

URL https://github.com/stencila/schema#readme

BugReports https://github.com/stencila/schema/issues

License Apache License (>= 2)

Language en-gb

Version 1.0.0

Encoding UTF-8

Suggests covr, devtools, lintr, pkgdown, rhub, roxygen2, spelling, testthat

RoxygenNote 7.1.1

Collate 'typing.R' 'types.R'

NeedsCompilation no

Author Nokome Bentley [aut, cre]

Maintainer Nokome Bentley <nokome@stenci.la>

Repository CRAN

Date/Publication 2021-02-22 10:40:03 UTC

R topics documented:

Any 4
Array 4
ArrayValidator 5
Article 6
as_scalar 8
as_type 8
AudioObject 9
BlockContent 11

BooleanValidator . . . 12
Brand . . . 12
check_property . . . 13
Cite . . . 14
CiteGroup . . . 15
Code . . . 15
CodeBlock . . . 16
CodeBlockTypes . . . 17
CodeChunk . . . 17
CodeError . . . 19
CodeExpression . . . 20
CodeFragment . . . 20
CodeFragmentTypes . . . 21
CodeTypes . . . 22
Collection . . . 22
Comment . . . 24
ConstantValidator . . . 26
ContactPoint . . . 27
ContactPointTypes . . . 28
CreativeWork . . . 28
CreativeWorkTypes . . . 30
Datatable . . . 31
DatatableColumn . . . 33
Date . . . 34
DefinedTerm . . . 34
Delete . . . 35
Emphasis . . . 36
Entity . . . 36
EntityTypes . . . 37
Enum . . . 37
EnumValidator . . . 38
Figure . . . 38
Function . . . 40
Grant . . . 41
GrantTypes . . . 42
Heading . . . 43
ImageObject . . . 43
Include . . . 46
InlineContent . . . 46
IntegerValidator . . . 47
is_class . . . 48
is_type . . . 48
last_class . . . 49
Link . . . 49
List . . . 50
ListItem . . . 51
Mark . . . 52
MarkTypes . . . 52

Math . . . 53
MathBlock . . . 53
MathFragment . . . 54
MathTypes . . . 55
MediaObject . . . 55
MediaObjectTypes . . . 58
MonetaryGrant . . . 58
Node . . . 59
NontextualAnnotation . . . 60
NumberValidator . . . 60
NumberValidatorTypes . . . 61
Organization . . . 62
Paragraph . . . 63
Parameter . . . 64
Periodical . . . 65
Person . . . 67
PostalAddress . . . 68
Product . . . 70
PropertyValue . . . 71
PublicationIssue . . . 72
PublicationVolume . . . 74
Quote . . . 76
QuoteBlock . . . 77
Review . . . 78
SoftwareApplication . . . 80
SoftwareEnvironment . . . 82
SoftwareSession . . . 83
SoftwareSourceCode . . . 85
StringValidator . . . 87
Strong . . . 88
Subscript . . . 88
Superscript . . . 89
Table . . . 89
TableCell . . . 92
TableRow . . . 92
ThematicBreak . . . 93
Thing . . . 93
ThingTypes . . . 94
TupleValidator . . . 95
Union . . . 95
ValidatorTypes . . . 96
Variable . . . 96
VariableTypes . . . 97
VideoObject . . . 97
VolumeMount . . . 100

Index **102**

Any *Any type*

Description

Any type

Usage

```
Any()
```

Value

A `list` of class `Any`

Examples

```
Any()
```

Array *Array type*

Description

Array type

Usage

```
Array(items)
```

Arguments

`items`	The type that items in the array should be

Value

A `list` of class `Array` describing the valid `items` of an array

Examples

```
Array(numeric)
Array("numeric")
Array(Person)
Array(Union(character, Person))
```

`ArrayValidator` *A validator specifying constraints on an array node.*

Description

A validator specifying constraints on an array node.

Usage

```
ArrayValidator(
  contains,
  id,
  itemsValidator,
  maxItems,
  meta,
  minItems,
  uniqueItems
)
```

Arguments

`contains`	An array node is valid if at least one of its items is valid against the `contains` schema.
`id`	The identifier for this item.
`itemsValidator`	Another validator node specifying the constraints on all items in the array.
`maxItems`	An array node is valid if its size is less than, or equal to, this value.
`meta`	Metadata associated with this item.
`minItems`	An array node is valid if its size is greater than, or equal to, this value.
`uniqueItems`	A flag to indicate that each value in the array should be unique.

Value

A `list` of class `ArrayValidator`

See Also

`Entity`

`Article` *An article, including news and scholarly articles.*

Description

An article, including news and scholarly articles.

Usage

```
Article(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  pageEnd,
  pageStart,
  pagination,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
pageEnd	The page on which the article ends; for example "138" or "xvi".
pageStart	The page on which the article starts; for example "135" or "xiii".
pagination	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55".
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `Article`

See Also

`CreativeWork`

`as_scalar` *Declare that a node is scalar*

Description

Declare that a node is scalar

Usage

```
as_scalar(node)
```

Arguments

`node`	The node to declare as a scalar

Value

The node with the `scalar` class added to it

`as_type` *Coerce a value to conform to the type Principally, marks values as scalar where possible*

Description

Coerce a value to conform to the type Principally, marks values as scalar where possible

Usage

```
as_type(value, type)
```

Arguments

`value`	The value to coerce
`type`	The type to coerce it to

Value

The value, possibly coerced to the type

AudioObject *An audio file*

Description

An audio file

Usage

```
AudioObject(
  contentUrl,
  about,
  alternateNames,
  authors,
  bitrate,
  caption,
  comments,
  content,
  contentSize,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  embedUrl,
  format,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  transcript,
  url,
```

```
  version
)
```

Arguments

contentUrl	URL for the actual bytes of the media object, for example the image file or video file. **Required**.
about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
bitrate	Bitrate in megabits per second (Mbit/s, Mb/s, Mbps).
caption	The caption for this audio recording.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
contentSize	File size in megabits (Mbit, Mb).
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
embedUrl	URL that can be used to embed the media on a web page via a specific media player.
format	Media type (MIME type) as per http://www.iana.org/assignments/media-types/media-types.xhtml.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.

parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
transcript	The transcript of this audio recording.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class AudioObject

See Also

MediaObject

BlockContent *Union type for valid block content.*

Description

Union type for valid block content.

Usage

```
BlockContent
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

BooleanValidator *A schema specifying that a node must be a boolean value.*

Description

A schema specifying that a node must be a boolean value.

Usage

```
BooleanValidator(id, meta)
```

Arguments

id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class BooleanValidator

See Also

Entity

Brand *A brand used by an organization or person for labeling a product, product group, or similar.*

Description

A brand used by an organization or person for labeling a product, product group, or similar.

Usage

```
Brand(
  name,
  alternateNames,
  description,
  id,
  identifiers,
  images,
  logo,
  meta,
  reviews,
  url
)
```

Arguments

name	The name of the item. **Required**.
alternateNames	Alternate names (aliases) for the item.
description	A description of the item.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
logo	A logo associated with the brand.
meta	Metadata associated with this item.
reviews	Reviews of the brand.
url	The URL of the item.

Value

A list of class Brand

See Also

Thing

---

check_property *Check that a value is present if required and conforms to the specified type for a property.*

Description

Check that a value is present if required and conforms to the specified type for a property.

Usage

```
check_property(type_name, property_name, is_required, is_missing, type, value)
```

Arguments

type_name	The name of the type that they property is on
property_name	The name of the property
is_required	Is a value for the property required?
is_missing	Is a value for the property missing?
type	The type to check against
value	The value to check

Value

The value, if it is valid, nothing otherwise

`Cite` *A reference to a CreativeWork that is cited in another CreativeWork.*

Description

A reference to a CreativeWork that is cited in another CreativeWork.

Usage

```
Cite(
  target,
  citationMode,
  content,
  id,
  meta,
  pageEnd,
  pageStart,
  pagination,
  prefix,
  suffix
)
```

Arguments

`target`	The target of the citation (URL or reference ID). **Required**.
`citationMode`	How the cite is rendered in the surrounding text.
`content`	Optional structured content/text of this citation.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.
`pageEnd`	The page on which the work ends; for example "138" or "xvi".
`pageStart`	The page on which the work starts; for example "135" or "xiii".
`pagination`	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55".
`prefix`	A prefix to show before the citation.
`suffix`	A suffix to show after the citation.

Value

A `list` of class `Cite`

See Also

`Entity`

CiteGroup *A group of* Cite *nodes*

Description

A group of Cite nodes

Usage

```
CiteGroup(items, id, meta)
```

Arguments

items	One or more Cites to be referenced in the same surrounding text. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class CiteGroup

See Also

Entity

Code *Base type for code nodes e.g.* CodeBlock, CodeExpression.

Description

Base type for code nodes e.g. CodeBlock, CodeExpression.

Usage

```
Code(text, format, id, meta, programmingLanguage)
```

Arguments

text	The text of the code. **Required**.
format	Media type, typically expressed using a MIME format, of the code.
id	The identifier for this item.
meta	Metadata associated with this item.
programmingLanguage	The programming language of the code.

Value

A `list` of class `Code`

See Also

`Entity`

`CodeBlock` *A code block.*

Description

A code block.

Usage

```
CodeBlock(text, exportFrom, format, id, importTo, meta, programmingLanguage)
```

Arguments

`text`	The text of the code. **Required**.
`exportFrom`	A compilation directive giving the name of the variable to export into the content of the code block.
`format`	Media type, typically expressed using a MIME format, of the code.
`id`	The identifier for this item.
`importTo`	A compilation directive giving the name of the variable to import the content of the code block as.
`meta`	Metadata associated with this item.
`programmingLanguage`	The programming language of the code.

Value

A `list` of class `CodeBlock`

See Also

`Code`

CodeBlockTypes *All type schemas that are derived from CodeBlock*

Description

All type schemas that are derived from CodeBlock

Usage

```
CodeBlockTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

CodeChunk *A executable chunk of code.*

Description

A executable chunk of code.

Usage

```
CodeChunk(
  text,
  alters,
  assigns,
  caption,
  declares,
  duration,
  errors,
  exportFrom,
  format,
  id,
  importTo,
  imports,
  label,
  meta,
  outputs,
  programmingLanguage,
  reads,
  uses
)
```

Arguments

text	The text of the code. **Required**.
alters	Names of variables that the code chunk alters.
assigns	Variables that the code chunk assigns to.
caption	A caption for the CodeChunk.
declares	Variables that the code chunk declares.
duration	Duration in seconds of the last execution of the chunk.
errors	Errors when compiling or executing the chunk.
exportFrom	A compilation directive giving the name of the variable to export into the content of the code block.
format	Media type, typically expressed using a MIME format, of the code.
id	The identifier for this item.
importTo	A compilation directive giving the name of the variable to import the content of the code block as.
imports	Software packages that the code chunk imports
label	A short label for the CodeChunk.
meta	Metadata associated with this item.
outputs	Outputs from executing the chunk.
programmingLanguage	The programming language of the code.
reads	Filesystem paths that this code chunk reads from.
uses	Names of variables that the code chunk uses (but does not alter).

Value

A `list` of class `CodeChunk`

See Also

`CodeBlock`

Examples

```
CodeChunk(
  programmingLanguage = "r",
  text = "plot(1)",
  label = "Figure 1",
  id = "fig1",
  caption = list(
    Heading(
      content = list("Figure title"),
      depth = 2
    ),
    Paragraph(
```

```
        content = list(
          "A paragraph with some",
          Strong(content = list("strong emphasis")),
          "in it."
        )
      )
    )
  )
```

`CodeError` *An error that occurred when parsing, compiling or executing a Code node.*

Description

An error that occurred when parsing, compiling or executing a Code node.

Usage

```
CodeError(errorMessage, errorType, id, meta, stackTrace)
```

Arguments

`errorMessage`	The error message or brief description of the error. **Required**.
`errorType`	The type of error e.g. "SyntaxError", "ZeroDivisionError".
`id`	The identifier for this item.
`meta`	Metadata associated with this item.
`stackTrace`	Stack trace leading up to the error.

Value

A `list` of class `CodeError`

See Also

`Entity`

CodeExpression *An expression defined in programming language source code.*

Description

An expression defined in programming language source code.

Usage

```
CodeExpression(text, errors, format, id, meta, output, programmingLanguage)
```

Arguments

`text`	The text of the code. **Required**.
`errors`	Errors when compiling or executing the chunk.
`format`	Media type, typically expressed using a MIME format, of the code.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.
`output`	The value of the expression when it was last evaluated.
`programmingLanguage`	The programming language of the code.

Value

A `list` of class `CodeExpression`

See Also

`CodeFragment`

CodeFragment *Inline code.*

Description

Inline code.

Usage

```
CodeFragment(text, format, id, meta, programmingLanguage)
```

Arguments

`text`	The text of the code. **Required**.
`format`	Media type, typically expressed using a MIME format, of the code.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.
`programmingLanguage`	The programming language of the code.

Value

A `list` of class `CodeFragment`

See Also

`Code`

`CodeFragmentTypes` *All type schemas that are derived from CodeFragment*

Description

All type schemas that are derived from CodeFragment

Usage

```
CodeFragmentTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

CodeTypes *All type schemas that are derived from Code*

Description

All type schemas that are derived from Code

Usage

```
CodeTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

Collection *A created collection of CreativeWorks or other artefacts.*

Description

A created collection of CreativeWorks or other artefacts.

Usage

```
Collection(
  parts,
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
```

```
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more. **Required**.
about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property `text`.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the `CreativeWork`.
fundedBy	Grants that funded the `CreativeWork`; reverse of `fundedItems`.
funders	People or organizations that funded the `CreativeWork`.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.

maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class Collection

See Also

CreativeWork

Comment *A comment on an item, e.g on a Article, or SoftwareSourceCode.*

Description

A comment on an item, e.g on a Article, or SoftwareSourceCode.

Usage

```
Comment(
  about,
  alternateNames,
  authors,
  commentAspect,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
```

```
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parentItem,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
commentAspect	The part or facet of the item that is being commented on.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.

licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parentItem	The parent comment of this comment.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `Comment`

See Also

`CreativeWork`

`ConstantValidator` *A validator specifying a constant value that a node must have.*

Description

A validator specifying a constant value that a node must have.

Usage

```
ConstantValidator(id, meta, value)
```

Arguments

id	The identifier for this item.
meta	Metadata associated with this item.
value	The value that the node must have.

Value

A `list` of class `ConstantValidator`

See Also

`Entity`

`ContactPoint` *A contact point, for example, a R&D department.*

Description

A contact point, for example, a R&D department.

Usage

```
ContactPoint(
  alternateNames,
  availableLanguages,
  description,
  emails,
  id,
  identifiers,
  images,
  meta,
  name,
  telephoneNumbers,
  url
)
```

Arguments

`alternateNames`	Alternate names (aliases) for the item.
`availableLanguages`	Languages (human not programming) in which it is possible to communicate with the organization/department etc.
`description`	A description of the item.
`emails`	Email address for correspondence.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`telephoneNumbers`	Telephone numbers for the contact point.
`url`	The URL of the item.

Value

A `list` of class `ContactPoint`

See Also

Thing

ContactPointTypes *All type schemas that are derived from ContactPoint*

Description

All type schemas that are derived from ContactPoint

Usage

```
ContactPointTypes
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

CreativeWork *A creative work, including books, movies, photographs, software programs, etc.*

Description

A creative work, including books, movies, photographs, software programs, etc.

Usage

```
CreativeWork(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
```

```
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.

keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `CreativeWork`

See Also

`Thing`

`CreativeWorkTypes` *All type schemas that are derived from CreativeWork*

Description

All type schemas that are derived from CreativeWork

Usage

```
CreativeWorkTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

`Datatable` *A table of data.*

Description

A table of data.

Usage

```
Datatable(
  columns,
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

`columns`	The columns of data. **Required**.

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class Datatable

See Also

CreativeWork

DatatableColumn *A column of data within a Datatable.*

Description

A column of data within a Datatable.

Usage

```
DatatableColumn(
  name,
  values,
  alternateNames,
  description,
  id,
  identifiers,
  images,
  meta,
  url,
  validator
)
```

Arguments

name	The name of the item. **Required**.
values	The data values of the column. **Required**.
alternateNames	Alternate names (aliases) for the item.
description	A description of the item.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
meta	Metadata associated with this item.
url	The URL of the item.
validator	The validator to use to validate data in the column.

Value

A list of class DatatableColumn

See Also

Thing

Date *A date encoded as a ISO 8601 string.*

Description

A date encoded as a ISO 8601 string.

Usage

```
Date(value, id, meta)
```

Arguments

value	The date as an ISO 8601 string. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Date

See Also

Entity

DefinedTerm *A word, name, acronym, phrase, etc. with a formal definition.*

Description

A word, name, acronym, phrase, etc. with a formal definition.

Usage

```
DefinedTerm(
  name,
  alternateNames,
  description,
  id,
  identifiers,
  images,
  meta,
  termCode,
  url
)
```

Arguments

name	The name of the item. **Required**.
alternateNames	Alternate names (aliases) for the item.
description	A description of the item.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
meta	Metadata associated with this item.
termCode	A code that identifies this DefinedTerm within a DefinedTermSet
url	The URL of the item.

Value

A list of class DefinedTerm

See Also

Thing

Delete *Content that is marked for deletion*

Description

Content that is marked for deletion

Usage

```
Delete(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Delete

See Also

Mark

Emphasis — *Emphasised content.*

Description

Emphasised content.

Usage

```
Emphasis(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Emphasis

See Also

Mark

Entity — *The most simple compound (ie. non-atomic like* number*,* string *etc) type.*

Description

The most simple compound (ie. non-atomic like number, string etc) type.

Usage

```
Entity(id, meta)
```

Arguments

id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Entity

EntityTypes *All type schemas that are derived from Entity*

Description

All type schemas that are derived from Entity

Usage

```
EntityTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

Enum *An enumeration*

Description

An enumeration

Usage

```
Enum(...)
```

Arguments

`...`	The values in the enumeration

Value

A `list` of class `Enum` describing the valid `values` in an enumeration

Examples

```
Enum("apple", "pear", "orange")
```

EnumValidator *A schema specifying that a node must be one of several values.*

Description

A schema specifying that a node must be one of several values.

Usage

```
EnumValidator(id, meta, values)
```

Arguments

id	The identifier for this item.
meta	Metadata associated with this item.
values	A node is valid if it is equal to any of these values.

Value

A list of class EnumValidator

See Also

Entity

Figure *Encapsulates one or more images, videos, tables, etc, and provides captions and labels for them.*

Description

Encapsulates one or more images, videos, tables, etc, and provides captions and labels for them.

Usage

```
Figure(
  about,
  alternateNames,
  authors,
  caption,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
```

```
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  label,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

`about`	The subject matter of the content.
`alternateNames`	Alternate names (aliases) for the item.
`authors`	The authors of this creative work.
`caption`	A caption for the figure.
`comments`	Comments about this creative work.
`content`	The structured content of this creative work c.f. property `text`.
`dateAccepted`	Date/time of acceptance.
`dateCreated`	Date/time of creation.
`dateModified`	Date/time of most recent modification.
`datePublished`	Date of first publication.
`dateReceived`	Date/time that work was received.
`description`	A description of the item.
`editors`	People who edited the `CreativeWork`.
`fundedBy`	Grants that funded the `CreativeWork`; reverse of `fundedItems`.
`funders`	People or organizations that funded the `CreativeWork`.
`genre`	Genre of the creative work, broadcast channel or group.

id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
label	A short label for the figure.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `Figure`

See Also

`CreativeWork`

`Function` *A function with a name, which might take Parameters and return a value of a certain type.*

Description

A function with a name, which might take Parameters and return a value of a certain type.

Usage

```
Function(id, meta, name, parameters, returns)
```

Arguments

id	The identifier for this item.
meta	Metadata associated with this item.
name	The name of the function.
parameters	The parameters of the function.
returns	The return type of the function.

Value

A list of class Function

See Also

Entity

Grant *A grant, typically financial or otherwise quantifiable, of resources.*

Description

A grant, typically financial or otherwise quantifiable, of resources.

Usage

```
Grant(
  alternateNames,
  description,
  fundedItems,
  id,
  identifiers,
  images,
  meta,
  name,
  sponsors,
  url
)
```

Arguments

alternateNames	Alternate names (aliases) for the item.
description	A description of the item.
fundedItems	Indicates an item funded or sponsored through a Grant.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.

images	Images of the item.
meta	Metadata associated with this item.
name	The name of the item.
sponsors	A person or organization that supports a thing through a pledge, promise, or financial contribution.
url	The URL of the item.

Value

A list of class Grant

See Also

Thing

GrantTypes *All type schemas that are derived from Grant*

Description

All type schemas that are derived from Grant

Usage

```
GrantTypes
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

Heading *Heading*

Description

Heading

Usage

```
Heading(content, depth, id, meta)
```

Arguments

content	Content of the heading. **Required**.
depth	The depth of the heading.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Heading

See Also

Entity

ImageObject *An image file.*

Description

An image file.

Usage

```
ImageObject(
  contentUrl,
  about,
  alternateNames,
  authors,
  bitrate,
  caption,
  comments,
  content,
  contentSize,
```

```
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  embedUrl,
  format,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  thumbnail,
  title,
  url,
  version
)
```

Arguments

`contentUrl`	URL for the actual bytes of the media object, for example the image file or video file. **Required**.
`about`	The subject matter of the content.
`alternateNames`	Alternate names (aliases) for the item.
`authors`	The authors of this creative work.
`bitrate`	Bitrate in megabits per second (Mbit/s, Mb/s, Mbps).
`caption`	The caption for this image.
`comments`	Comments about this creative work.
`content`	The structured content of this creative work c.f. property `text`.
`contentSize`	File size in megabits (Mbit, Mb).
`dateAccepted`	Date/time of acceptance.
`dateCreated`	Date/time of creation.
`dateModified`	Date/time of most recent modification.

datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
embedUrl	URL that can be used to embed the media on a web page via a specific media player.
format	Media type (MIME type) as per http://www.iana.org/assignments/media-types/media-types.xhtml.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
thumbnail	Thumbnail image of this image.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class ImageObject

See Also

MediaObject

Include — *A directive to include content from an external source (e.g. file, URL) or content.*

Description

A directive to include content from an external source (e.g. file, URL) or content.

Usage

```
Include(source, content, format, id, meta)
```

Arguments

source	The source of the content, a URL or file path, or the content itself. **Required**.
content	The content to be included.
format	Media type, typically expressed using a MIME format, of the source content.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Include

See Also

Entity

InlineContent — *Union type for valid inline content.*

Description

Union type for valid inline content.

Usage

```
InlineContent
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

`IntegerValidator` *A validator specifying the constraints on an integer node.*

Description

A validator specifying the constraints on an integer node.

Usage

```
IntegerValidator(
  exclusiveMaximum,
  exclusiveMinimum,
  id,
  maximum,
  meta,
  minimum,
  multipleOf
)
```

Arguments

`exclusiveMaximum`	The exclusive upper limit for a numeric node.
`exclusiveMinimum`	The exclusive lower limit for a numeric node.
`id`	The identifier for this item.
`maximum`	The inclusive upper limit for a numeric node.
`meta`	Metadata associated with this item.
`minimum`	The inclusive lower limit for a numeric node.
`multipleOf`	A number that a numeric node must be a multiple of.

Value

A `list` of class `IntegerValidator`

See Also

`NumberValidator`

is_class *Is a value of a particular class*

Description

Is a value of a particular class

Usage

```
is_class(value, clas)
```

Arguments

value	The value to check
clas	The class the check against

Value

TRUE if the value has the class as its last class, FALSE otherwise

is_type *Does a value conform to the type?*

Description

Does a value conform to the type?

Usage

```
is_type(value, type)
```

Arguments

value	The value to check
type	The type to check against

Value

TRUE if the value conforms to the type, FALSE otherwise

last_class — *Get the last class for an object The last class is usually the "highest" in the inheritance tree*

Description

Get the last class for an object The last class is usually the "highest" in the inheritance tree

Usage

```
last_class(obj)
```

Arguments

obj	The object to get the last class for

Value

The class that is last in the list of classes for the object

Link — *A hyperlink to other pages, sections within the same document, resources, or any URL.*

Description

A hyperlink to other pages, sections within the same document, resources, or any URL.

Usage

```
Link(content, target, exportFrom, id, importTo, meta, relation, title)
```

Arguments

content	The textual content of the link. **Required**.
target	The target of the link. **Required**.
exportFrom	A compilation directive giving the name of the variable to export to the link target.
id	The identifier for this item.
importTo	A compilation directive giving the name of the variable to import the link target as.
meta	Metadata associated with this item.
relation	The relation between the target and the current thing.
title	A title for the link.

Value

A `list` of class `Link`

See Also

`Entity`

`List` *A list of items.*

Description

A list of items.

Usage

```
List(items, id, meta, order)
```

Arguments

`items`	The items in the list **Required**.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.
`order`	Type of ordering.

Value

A `list` of class `List`

See Also

`Entity`

`ListItem` *A single item in a list.*

Description

A single item in a list.

Usage

```
ListItem(
  alternateNames,
  content,
  description,
  id,
  identifiers,
  images,
  isChecked,
  item,
  meta,
  name,
  position,
  url
)
```

Arguments

`alternateNames`	Alternate names (aliases) for the item.
`content`	The content of the list item.
`description`	A description of the item.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`isChecked`	A flag to indicate if this list item is checked.
`item`	The item represented by this list item.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`position`	The position of the item in a series or sequence of items.
`url`	The URL of the item.

Value

A `list` of class `ListItem`

See Also

`Thing`

Mark — *A base class for nodes that mark some other inline content in some way (e.g. as being emphasised, or quoted).*

Description

A base class for nodes that mark some other inline content in some way (e.g. as being emphasised, or quoted).

Usage

```
Mark(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Mark

See Also

Entity

MarkTypes — *All type schemas that are derived from Mark*

Description

All type schemas that are derived from Mark

Usage

```
MarkTypes
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

Math *A mathematical variable or equation.*

Description

A mathematical variable or equation.

Usage

```
Math(text, errors, id, mathLanguage, meta)
```

Arguments

text	The text of the equation in the language. **Required**.
errors	Errors that occurred when parsing the math equation.
id	The identifier for this item.
mathLanguage	The language used for the equation e.g tex, mathml, asciimath.
meta	Metadata associated with this item.

Value

A list of class Math

See Also

Entity

MathBlock *A block of math, e.g an equation, to be treated as block content.*

Description

A block of math, e.g an equation, to be treated as block content.

Usage

```
MathBlock(text, errors, id, mathLanguage, meta)
```

Arguments

text	The text of the equation in the language. **Required**.
errors	Errors that occurred when parsing the math equation.
id	The identifier for this item.
mathLanguage	The language used for the equation e.g tex, mathml, asciimath.
meta	Metadata associated with this item.

Value

A `list` of class `MathBlock`

See Also

`Math`

`MathFragment` *A fragment of math, e.g a variable name, to be treated as inline content.*

Description

A fragment of math, e.g a variable name, to be treated as inline content.

Usage

```
MathFragment(text, errors, id, mathLanguage, meta)
```

Arguments

`text`	The text of the equation in the language. **Required**.
`errors`	Errors that occurred when parsing the math equation.
`id`	The identifier for this item.
`mathLanguage`	The language used for the equation e.g tex, mathml, asciimath.
`meta`	Metadata associated with this item.

Value

A `list` of class `MathFragment`

See Also

`Math`

MathTypes *All type schemas that are derived from Math*

Description

All type schemas that are derived from Math

Usage

```
MathTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

MediaObject *A media object, such as an image, video, or audio object embedded in a web page or a downloadable dataset.*

Description

A media object, such as an image, video, or audio object embedded in a web page or a downloadable dataset.

Usage

```
MediaObject(
  contentUrl,
  about,
  alternateNames,
  authors,
  bitrate,
  comments,
  content,
  contentSize,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
```

```
  embedUrl,
  format,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

contentUrl	URL for the actual bytes of the media object, for example the image file or video file. **Required**.
about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
bitrate	Bitrate in megabits per second (Mbit/s, Mb/s, Mbps).
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property `text`.
contentSize	File size in megabits (Mbit, Mb).
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the `CreativeWork`.
embedUrl	URL that can be used to embed the media on a web page via a specific media player.

format	Media type (MIME type) as per http://www.iana.org/assignments/media-types/media-types.xhtml.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class MediaObject

See Also

CreativeWork

MediaObjectTypes *All type schemas that are derived from MediaObject*

Description

All type schemas that are derived from MediaObject

Usage

```
MediaObjectTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

MonetaryGrant *A monetary grant.*

Description

A monetary grant.

Usage

```
MonetaryGrant(
  alternateNames,
  amounts,
  description,
  fundedItems,
  funders,
  id,
  identifiers,
  images,
  meta,
  name,
  sponsors,
  url
)
```

Arguments

alternateNames	Alternate names (aliases) for the item.
amounts	The amount of money.
description	A description of the item.
fundedItems	Indicates an item funded or sponsored through a Grant.
funders	A person or organization that supports (sponsors) something through some kind of financial contribution.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
meta	Metadata associated with this item.
name	The name of the item.
sponsors	A person or organization that supports a thing through a pledge, promise, or financial contribution.
url	The URL of the item.

Value

A list of class MonetaryGrant

See Also

Grant

Node *Union type for all valid nodes.*

Description

Union type for all valid nodes.

Usage

```
Node
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

NontextualAnnotation *Inline text that has a non-textual annotation.*

Description

Inline text that has a non-textual annotation.

Usage

```
NontextualAnnotation(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class NontextualAnnotation

See Also

Mark

NumberValidator *A validator specifying the constraints on a numeric node.*

Description

A validator specifying the constraints on a numeric node.

Usage

```
NumberValidator(
  exclusiveMaximum,
  exclusiveMinimum,
  id,
  maximum,
  meta,
  minimum,
  multipleOf
)
```

Arguments

exclusiveMaximum	The exclusive upper limit for a numeric node.
exclusiveMinimum	The exclusive lower limit for a numeric node.
id	The identifier for this item.
maximum	The inclusive upper limit for a numeric node.
meta	Metadata associated with this item.
minimum	The inclusive lower limit for a numeric node.
multipleOf	A number that a numeric node must be a multiple of.

Value

A list of class NumberValidator

See Also

Entity

NumberValidatorTypes *All type schemas that are derived from NumberValidator*

Description

All type schemas that are derived from NumberValidator

Usage

```
NumberValidatorTypes
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

`Organization` *An organization such as a school, NGO, corporation, club, etc.*

Description

An organization such as a school, NGO, corporation, club, etc.

Usage

```
Organization(
  address,
  alternateNames,
  brands,
  contactPoints,
  departments,
  description,
  funders,
  id,
  identifiers,
  images,
  legalName,
  logo,
  members,
  meta,
  name,
  parentOrganization,
  url
)
```

Arguments

`address`	Postal address for the organization.
`alternateNames`	Alternate names (aliases) for the item.
`brands`	Brands that the organization is connected with.
`contactPoints`	Correspondence/Contact points for the organization.
`departments`	Departments within the organization. For example, Department of Computer Science, Research & Development etc.
`description`	A description of the item.
`funders`	Organization(s) or person(s) funding the organization.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`legalName`	Legal name for the Organization. Should only include letters and spaces.
`logo`	The logo of the organization.

members	Person(s) or organization(s) who are members of this organization.
meta	Metadata associated with this item.
name	The name of the item.
parentOrganization	Entity that the Organization is a part of. For example, parentOrganization to a department is a university.
url	The URL of the item.

Value

A list of class Organization

See Also

Thing

Paragraph *Paragraph*

Description

Paragraph

Usage

```
Paragraph(content, id, meta)
```

Arguments

content	The contents of the paragraph. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Paragraph

See Also

Entity

`Parameter` *A parameter that can be set and used in evaluated code.*

Description

A parameter that can be set and used in evaluated code.

Usage

```
Parameter(
  name,
  default,
  id,
  isExtensible,
  isReadonly,
  isRequired,
  isVariadic,
  meta,
  validator,
  value
)
```

Arguments

`name`	The name of the variable. **Required**.
`default`	The default value of the parameter.
`id`	The identifier for this item.
`isExtensible`	Indicates that this parameter is variadic and can accept multiple named arguments.
`isReadonly`	Whether or not a property is mutable. Default is false.
`isRequired`	Is this parameter required, if not it should have a default or default is assumed to be null.
`isVariadic`	Indicates that this parameter is variadic and can accept multiple arguments.
`meta`	Metadata associated with this item.
`validator`	The validator that the value is validated against.
`value`	The value of the variable.

Value

A `list` of class `Parameter`

See Also

`Variable`

`Periodical` *A periodical publication.*

Description

A periodical publication.

Usage

```
Periodical(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateEnd,
  dateModified,
  datePublished,
  dateReceived,
  dateStart,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  issns,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateEnd	The date this Periodical ceased publication.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
dateStart	The date this Periodical was first published.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
issns	The International Standard Serial Number(s) (ISSN) that identifies this serial publication.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `Periodical`

See Also

`CreativeWork`

`Person` *A person (alive, dead, undead, or fictional).*

Description

A person (alive, dead, undead, or fictional).

Usage

```
Person(
  address,
  affiliations,
  alternateNames,
  description,
  emails,
  familyNames,
  funders,
  givenNames,
  honorificPrefix,
  honorificSuffix,
  id,
  identifiers,
  images,
  jobTitle,
  memberOf,
  meta,
  name,
  telephoneNumbers,
  url
)
```

Arguments

`address`	Postal address for the person.
`affiliations`	Organizations that the person is affiliated with.
`alternateNames`	Alternate names (aliases) for the item.
`description`	A description of the item.
`emails`	Email addresses for the person.

familyNames	Family name. In the U.S., the last name of a person.
funders	A person or organization that supports (sponsors) something through some kind of financial contribution.
givenNames	Given name. In the U.S., the first name of a person.
honorificPrefix	An honorific prefix preceding a person's name such as Dr/Mrs/Mr.
honorificSuffix	An honorific suffix after a person's name such as MD/PhD/MSCSW.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
jobTitle	The job title of the person (for example, Financial Manager).
memberOf	An organization (or program membership) to which this person belongs.
meta	Metadata associated with this item.
name	The name of the item.
telephoneNumbers	Telephone numbers for the person.
url	The URL of the item.

Value

A list of class Person

See Also

Thing

Examples

```
Person(
  honorificPrefix = "Dr",
  givenNames = list("Jane", "J"),
  familyNames = list("Jamieson")
)
```

PostalAddress *A physical mailing address.*

Description

A physical mailing address.

Usage

```
PostalAddress(
  addressCountry,
  addressLocality,
  addressRegion,
  alternateNames,
  availableLanguages,
  description,
  emails,
  id,
  identifiers,
  images,
  meta,
  name,
  postOfficeBoxNumber,
  postalCode,
  streetAddress,
  telephoneNumbers,
  url
)
```

Arguments

addressCountry	The country.
addressLocality	The locality in which the street address is, and which is in the region.
addressRegion	The region in which the locality is, and which is in the country.
alternateNames	Alternate names (aliases) for the item.
availableLanguages	Languages (human not programming) in which it is possible to communicate with the organization/department etc.
description	A description of the item.
emails	Email address for correspondence.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
meta	Metadata associated with this item.
name	The name of the item.
postOfficeBoxNumber	The post office box number.
postalCode	The postal code.
streetAddress	The street address.
telephoneNumbers	Telephone numbers for the contact point.
url	The URL of the item.

Value

A `list` of class `PostalAddress`

See Also

`ContactPoint`

`Product` *Any offered product or service. For example, a pair of shoes; a haircut; or an episode of a TV show streamed online.*

Description

Any offered product or service. For example, a pair of shoes; a haircut; or an episode of a TV show streamed online.

Usage

```
Product(
  alternateNames,
  brands,
  description,
  id,
  identifiers,
  images,
  logo,
  meta,
  name,
  productID,
  url
)
```

Arguments

`alternateNames`	Alternate names (aliases) for the item.
`brands`	Brands that the product is labelled with.
`description`	A description of the item.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`logo`	The logo of the product.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`productID`	Product identification code.
`url`	The URL of the item.

Value

A `list` of class `Product`

See Also

`Thing`

`PropertyValue` *A property-value pair.*

Description

A property-value pair.

Usage

```
PropertyValue(
  value,
  alternateNames,
  description,
  id,
  identifiers,
  images,
  meta,
  name,
  propertyID,
  url
)
```

Arguments

`value`	The value of the property. **Required**.
`alternateNames`	Alternate names (aliases) for the item.
`description`	A description of the item.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`propertyID`	A commonly used identifier for the characteristic represented by the property.
`url`	The URL of the item.

Value

A `list` of class `PropertyValue`

See Also

Thing

PublicationIssue *A part of a successively published publication such as a periodical or publication volume, often numbered.*

Description

A part of a successively published publication such as a periodical or publication volume, often numbered.

Usage

```
PublicationIssue(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  issueNumber,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  pageEnd,
  pageStart,
  pagination,
  parts,
  publisher,
  references,
```

```
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
issueNumber	Identifies the issue of publication; for example, "iii" or "2".
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
pageEnd	The page on which the issue ends; for example "138" or "xvi".
pageStart	The page on which the issue starts; for example "135" or "xiii".
pagination	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55".
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.

publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class PublicationIssue

See Also

CreativeWork

PublicationVolume *A part of a successively published publication such as a periodical or multi-volume work.*

Description

A part of a successively published publication such as a periodical or multi-volume work.

Usage

```
PublicationVolume(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
```

```
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  pageEnd,
  pageStart,
  pagination,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version,
  volumeNumber
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.

maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
pageEnd	The page on which the volume ends; for example "138" or "xvi".
pageStart	The page on which the volume starts; for example "135" or "xiii".
pagination	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55".
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.
volumeNumber	Identifies the volume of publication or multi-part work; for example, "iii" or "2".

Value

A `list` of class `PublicationVolume`

See Also

`CreativeWork`

`Quote` *Inline, quoted content.*

Description

Inline, quoted content.

Usage

```
Quote(content, cite, id, meta)
```

Arguments

content	The content that is marked. **Required**.
cite	The source of the quote.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A `list` of class `Quote`

See Also

`Mark`

`QuoteBlock` *A section quoted from somewhere else.*

Description

A section quoted from somewhere else.

Usage

```
QuoteBlock(content, cite, id, meta)
```

Arguments

`content`	The content of the quote. **Required**.
`cite`	The source of the quote.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.

Value

A `list` of class `QuoteBlock`

See Also

`Entity`

`Review` *A review of an item, e.g of an Article, or SoftwareSourceCode.*

Description

A review of an item, e.g of an Article, or SoftwareSourceCode.

Usage

```
Review(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  itemReviewed,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  reviewAspect,
  text,
  title,
  url,
  version
)
```

Arguments

`about`	The subject matter of the content.

`alternateNames`	Alternate names (aliases) for the item.
`authors`	The authors of this creative work.
`comments`	Comments about this creative work.
`content`	The structured content of this creative work c.f. property `text`.
`dateAccepted`	Date/time of acceptance.
`dateCreated`	Date/time of creation.
`dateModified`	Date/time of most recent modification.
`datePublished`	Date of first publication.
`dateReceived`	Date/time that work was received.
`description`	A description of the item.
`editors`	People who edited the `CreativeWork`.
`fundedBy`	Grants that funded the `CreativeWork`; reverse of `fundedItems`.
`funders`	People or organizations that funded the `CreativeWork`.
`genre`	Genre of the creative work, broadcast channel or group.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`isPartOf`	An item or other CreativeWork that this CreativeWork is a part of.
`itemReviewed`	The item that is being reviewed.
`keywords`	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
`licenses`	License documents that applies to this content, typically indicated by URL.
`maintainers`	The people or organizations who maintain this CreativeWork.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`parts`	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
`publisher`	A publisher of the CreativeWork.
`references`	References to other creative works, such as another publication, web page, scholarly article, etc.
`reviewAspect`	The part or facet of the item that is being reviewed.
`text`	The textual content of this creative work.
`title`	The title of the creative work.
`url`	The URL of the item.
`version`	The version of the creative work.

Value

A `list` of class `Review`

See Also

`CreativeWork`

`SoftwareApplication` *A software application.*

Description

A software application.

Usage

```
SoftwareApplication(
  about,
  alternateNames,
  authors,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  softwareRequirements,
  softwareVersion,
  text,
  title,
  url,
  version
)
```

Arguments

about	The subject matter of the content.

alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
softwareRequirements	Requirements for application, including shared libraries that are not included in the application distribution.
softwareVersion	Version of the software.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `SoftwareApplication`

See Also

`CreativeWork`

`SoftwareEnvironment` *A computational environment.*

Description

A computational environment.

Usage

```
SoftwareEnvironment(
  name,
  adds,
  alternateNames,
  description,
  extends,
  id,
  identifiers,
  images,
  meta,
  removes,
  url
)
```

Arguments

`name`	The name of the item. **Required**.
`adds`	The packages that this environment adds to the base environments listed under `extends` (if any).,
`alternateNames`	Alternate names (aliases) for the item.
`description`	A description of the item.
`extends`	Other environments that this environment extends by adding or removing packages.,
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`meta`	Metadata associated with this item.
`removes`	The packages that this environment removes from the base environments listed under `extends` (if any).,
`url`	The URL of the item.

Value

A `list` of class `SoftwareEnvironment`

See Also

`Thing`

`SoftwareSession` *Definition of a compute session, including its software and compute resource requirements and status.*

Description

Definition of a compute session, including its software and compute resource requirements and status.

Usage

```
SoftwareSession(
  alternateNames,
  clientsLimit,
  clientsRequest,
  cpuLimit,
  cpuRequest,
  dateEnd,
  dateStart,
  description,
  durationLimit,
  durationRequest,
  environment,
  id,
  identifiers,
  images,
  memoryLimit,
  memoryRequest,
  meta,
  name,
  networkTransferLimit,
  networkTransferRequest,
  status,
  timeoutLimit,
  timeoutRequest,
  url,
  volumeMounts
)
```

Arguments

alternateNames	Alternate names (aliases) for the item.
clientsLimit	The maximum number of concurrent clients the session is limited to.
clientsRequest	The maximum number of concurrent clients requested for the session.
cpuLimit	The amount of CPU the session is limited to.
cpuRequest	The amount of CPU requested for the session.
dateEnd	The date-time that the session ended.
dateStart	The date-time that the session began.
description	A description of the item.
durationLimit	The maximum duration (seconds) the session is limited to.
durationRequest	The maximum duration (seconds) requested for the session.
environment	The software environment to execute this session in.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
memoryLimit	The amount of memory that the session is limited to.
memoryRequest	The amount of memory requested for the session.
meta	Metadata associated with this item.
name	The name of the item.
networkTransferLimit	The amount of network data transfer (GiB) that the session is limited to.
networkTransferRequest	The amount of network data transfer (GiB) requested for the session.
status	The status of the session (starting, stopped, etc).
timeoutLimit	The inactivity timeout (seconds) the session is limited to.
timeoutRequest	The inactivity timeout (seconds) requested for the session.
url	The URL of the item.
volumeMounts	Volumes to mount in the session.

Value

A `list` of class `SoftwareSession`

See Also

`Thing`

SoftwareSourceCode	*Computer programming source code. Example: Full (compile ready) solutions, code snippet samples, scripts, templates.*

Description

Computer programming source code. Example: Full (compile ready) solutions, code snippet samples, scripts, templates.

Usage

```
SoftwareSourceCode(
  about,
  alternateNames,
  authors,
  codeRepository,
  codeSampleType,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  programmingLanguage,
  publisher,
  references,
  runtimePlatform,
  softwareRequirements,
  targetProducts,
  text,
  title,
```

```
  url,
  version
)
```

Arguments

about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
codeRepository	Link to the repository where the un-compiled, human readable code and related code is located.
codeSampleType	What type of code sample: full (compile ready) solution, code snippet, inline code, scripts, template.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property text.
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the CreativeWork.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
programmingLanguage	The computer programming language.
publisher	A publisher of the CreativeWork.

references	References to other creative works, such as another publication, web page, scholarly article, etc.
runtimePlatform	Runtime platform or script interpreter dependencies (Example - Java v1, Python2.3, .Net Framework 3.0).
softwareRequirements	Dependency requirements for the software.
targetProducts	Target operating system or product to which the code applies.
text	The textual content of this creative work.
title	The title of the creative work.
url	The URL of the item.
version	The version of the creative work.

Value

A `list` of class `SoftwareSourceCode`

See Also

`CreativeWork`

`StringValidator` *A schema specifying constraints on a string node.*

Description

A schema specifying constraints on a string node.

Usage

```
StringValidator(id, maxLength, meta, minLength, pattern)
```

Arguments

id	The identifier for this item.
maxLength	The maximum length for a string node.
meta	Metadata associated with this item.
minLength	The minimum length for a string node.
pattern	A regular expression that a string node must match.

Value

A `list` of class `StringValidator`

See Also

`Entity`

Strong *Strongly emphasised content.*

Description

Strongly emphasised content.

Usage

```
Strong(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Strong

See Also

Mark

Subscript *Subscripted content.*

Description

Subscripted content.

Usage

```
Subscript(content, id, meta)
```

Arguments

content	The content that is marked. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.

Value

A list of class Subscript

See Also

Mark

`Superscript` *Superscripted content.*

Description

Superscripted content.

Usage

```
Superscript(content, id, meta)
```

Arguments

`content`	The content that is marked. **Required**.
`id`	The identifier for this item.
`meta`	Metadata associated with this item.

Value

A `list` of class `Superscript`

See Also

Mark

`Table` *A table.*

Description

A table.

Usage

```
Table(
  rows,
  about,
  alternateNames,
  authors,
  caption,
  comments,
  content,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  label,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  title,
  url,
  version
)
```

Arguments

`rows`	Rows of cells in the table. **Required**.
`about`	The subject matter of the content.
`alternateNames`	Alternate names (aliases) for the item.
`authors`	The authors of this creative work.
`caption`	A caption for the table.
`comments`	Comments about this creative work.
`content`	The structured content of this creative work c.f. property `text`.

`dateAccepted`	Date/time of acceptance.
`dateCreated`	Date/time of creation.
`dateModified`	Date/time of most recent modification.
`datePublished`	Date of first publication.
`dateReceived`	Date/time that work was received.
`description`	A description of the item.
`editors`	People who edited the `CreativeWork`.
`fundedBy`	Grants that funded the `CreativeWork`; reverse of `fundedItems`.
`funders`	People or organizations that funded the `CreativeWork`.
`genre`	Genre of the creative work, broadcast channel or group.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`isPartOf`	An item or other CreativeWork that this CreativeWork is a part of.
`keywords`	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
`label`	A short label for the table.
`licenses`	License documents that applies to this content, typically indicated by URL.
`maintainers`	The people or organizations who maintain this CreativeWork.
`meta`	Metadata associated with this item.
`name`	The name of the item.
`parts`	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
`publisher`	A publisher of the CreativeWork.
`references`	References to other creative works, such as another publication, web page, scholarly article, etc.
`text`	The textual content of this creative work.
`title`	The title of the creative work.
`url`	The URL of the item.
`version`	The version of the creative work.

Value

A `list` of class `Table`

See Also

`CreativeWork`

TableCell *A cell within a* Table.

Description

A cell within a Table.

Usage

```
TableCell(content, cellType, colspan, id, meta, name, rowspan)
```

Arguments

content	Contents of the table cell. **Required**.
cellType	Indicates whether the cell is a header or data.
colspan	How many columns the cell extends.
id	The identifier for this item.
meta	Metadata associated with this item.
name	The name of the cell.
rowspan	How many columns the cell extends.

Value

A list of class TableCell

See Also

Entity

TableRow *A row within a Table.*

Description

A row within a Table.

Usage

```
TableRow(cells, id, meta, rowType)
```

Arguments

cells	An array of cells in the row. **Required**.
id	The identifier for this item.
meta	Metadata associated with this item.
rowType	If present, indicates that all cells in this row should be treated as header cells.

Value

A `list` of class `TableRow`

See Also

`Entity`

`ThematicBreak` *A thematic break, such as a scene change in a story, a transition to another topic, or a new document.*

Description

A thematic break, such as a scene change in a story, a transition to another topic, or a new document.

Usage

```
ThematicBreak(id, meta)
```

Arguments

`id`	The identifier for this item.
`meta`	Metadata associated with this item.

Value

A `list` of class `ThematicBreak`

See Also

`Entity`

`Thing` *The most generic type of item.*

Description

The most generic type of item.

Usage

```
Thing(alternateNames, description, id, identifiers, images, meta, name, url)
```

Arguments

alternateNames	Alternate names (aliases) for the item.
description	A description of the item.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
meta	Metadata associated with this item.
name	The name of the item.
url	The URL of the item.

Value

A list of class Thing

See Also

Entity

ThingTypes *All type schemas that are derived from Thing*

Description

All type schemas that are derived from Thing

Usage

```
ThingTypes
```

Format

An object of class Union of length 1.

Value

A list of class Union describing valid subtypes of this type

TupleValidator *A validator specifying constraints on an array of heterogeneous items.*

Description

A validator specifying constraints on an array of heterogeneous items.

Usage

```
TupleValidator(id, items, meta)
```

Arguments

id	The identifier for this item.
items	An array of validators specifying the constraints on each successive item in the array.
meta	Metadata associated with this item.

Value

A list of class TupleValidator

See Also

Entity

Union *Union type*

Description

Union type

Usage

```
Union(...)
```

Arguments

...	The types in the union

Value

A list of class Union describing the valid sub types of a union type

Examples

```
Union(character, Person)
```

`ValidatorTypes` *Union type for all validator types.*

Description

Union type for all validator types.

Usage

```
ValidatorTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

`Variable` *A variable representing a name / value pair.*

Description

A variable representing a name / value pair.

Usage

```
Variable(name, id, isReadonly, meta, validator, value)
```

Arguments

`name`	The name of the variable. **Required**.
`id`	The identifier for this item.
`isReadonly`	Whether or not a property is mutable. Default is false.
`meta`	Metadata associated with this item.
`validator`	The validator that the value is validated against.
`value`	The value of the variable.

Value

A `list` of class `Variable`

See Also

`Entity`

VariableTypes *All type schemas that are derived from Variable*

Description

All type schemas that are derived from Variable

Usage

```
VariableTypes
```

Format

An object of class `Union` of length 1.

Value

A `list` of class `Union` describing valid subtypes of this type

VideoObject *A video file.*

Description

A video file.

Usage

```
VideoObject(
  contentUrl,
  about,
  alternateNames,
  authors,
  bitrate,
  caption,
  comments,
  content,
  contentSize,
  dateAccepted,
  dateCreated,
  dateModified,
  datePublished,
  dateReceived,
  description,
  editors,
  embedUrl,
```

```
  format,
  fundedBy,
  funders,
  genre,
  id,
  identifiers,
  images,
  isPartOf,
  keywords,
  licenses,
  maintainers,
  meta,
  name,
  parts,
  publisher,
  references,
  text,
  thumbnail,
  title,
  transcript,
  url,
  version
)
```

Arguments

contentUrl	URL for the actual bytes of the media object, for example the image file or video file. **Required**.
about	The subject matter of the content.
alternateNames	Alternate names (aliases) for the item.
authors	The authors of this creative work.
bitrate	Bitrate in megabits per second (Mbit/s, Mb/s, Mbps).
caption	The caption for this video recording.
comments	Comments about this creative work.
content	The structured content of this creative work c.f. property `text`.
contentSize	File size in megabits (Mbit, Mb).
dateAccepted	Date/time of acceptance.
dateCreated	Date/time of creation.
dateModified	Date/time of most recent modification.
datePublished	Date of first publication.
dateReceived	Date/time that work was received.
description	A description of the item.
editors	People who edited the `CreativeWork`.

embedUrl	URL that can be used to embed the media on a web page via a specific media player.
format	Media type (MIME type) as per http://www.iana.org/assignments/media-types/media-types.xhtml.
fundedBy	Grants that funded the CreativeWork; reverse of fundedItems.
funders	People or organizations that funded the CreativeWork.
genre	Genre of the creative work, broadcast channel or group.
id	The identifier for this item.
identifiers	Any kind of identifier for any kind of Thing.
images	Images of the item.
isPartOf	An item or other CreativeWork that this CreativeWork is a part of.
keywords	Keywords or tags used to describe this content. Multiple entries in a keywords list are typically delimited by commas.
licenses	License documents that applies to this content, typically indicated by URL.
maintainers	The people or organizations who maintain this CreativeWork.
meta	Metadata associated with this item.
name	The name of the item.
parts	Elements of the collection which can be a variety of different elements, such as Articles, Datatables, Tables and more.
publisher	A publisher of the CreativeWork.
references	References to other creative works, such as another publication, web page, scholarly article, etc.
text	The textual content of this creative work.
thumbnail	Thumbnail image of this video recording.
title	The title of the creative work.
transcript	The transcript of this video recording.
url	The URL of the item.
version	The version of the creative work.

Value

A list of class VideoObject

See Also

MediaObject

`VolumeMount` *Describes a volume mount from a host to container.*

Description

Describes a volume mount from a host to container.

Usage

```
VolumeMount(
  mountDestination,
  alternateNames,
  description,
  id,
  identifiers,
  images,
  meta,
  mountOptions,
  mountSource,
  mountType,
  name,
  url
)
```

Arguments

`mountDestination`	The mount location inside the container. **Required**.
`alternateNames`	Alternate names (aliases) for the item.
`description`	A description of the item.
`id`	The identifier for this item.
`identifiers`	Any kind of identifier for any kind of Thing.
`images`	Images of the item.
`meta`	Metadata associated with this item.
`mountOptions`	A list of options to use when applying the mount.
`mountSource`	The mount source directory on the host.
`mountType`	The type of mount.
`name`	The name of the item.
`url`	The URL of the item.

Value

A `list` of class `VolumeMount`

See Also

Thing

Index

∗ **datasets**
BlockContent, 11
CodeBlockTypes, 17
CodeFragmentTypes, 21
CodeTypes, 22
ContactPointTypes, 28
CreativeWorkTypes, 30
EntityTypes, 37
GrantTypes, 42
InlineContent, 46
MarkTypes, 52
MathTypes, 55
MediaObjectTypes, 58
Node, 59
NumberValidatorTypes, 61
ThingTypes, 94
ValidatorTypes, 96
VariableTypes, 97

Any, 4
Array, 4
ArrayValidator, 5
Article, 6
as_scalar, 8
as_type, 8
AudioObject, 9

BlockContent, 11
BooleanValidator, 12
Brand, 12

check_property, 13
Cite, 14
CiteGroup, 15
Code, 15, *16*, *21*
CodeBlock, 16, *18*
CodeBlockTypes, 17
CodeChunk, 17
CodeError, 19
CodeExpression, 20
CodeFragment, *20*, 20
CodeFragmentTypes, 21
CodeTypes, 22
Collection, 22
Comment, 24
ConstantValidator, 26
ContactPoint, 27, *70*
ContactPointTypes, 28
CreativeWork, *8*, *24*, *26*, 28, *32*, *40*, *57*, *67*, *74*, *76*, *79*, *82*, *87*, *91*
CreativeWorkTypes, 30

Datatable, 31
DatatableColumn, 33
Date, 34
DefinedTerm, 34
Delete, 35

Emphasis, 36
Entity, *5*, *12*, *14–16*, *19*, *26*, *34*, 36, *38*, *41*, *43*, *46*, *50*, *52*, *53*, *61*, *63*, *77*, *87*, *92–96*
EntityTypes, 37
Enum, 37
EnumValidator, 38

Figure, 38
Function, 40

Grant, 41, *59*
GrantTypes, 42

Heading, 43

ImageObject, 43
Include, 46
InlineContent, 46
IntegerValidator, 47
is_class, 48
is_type, 48

last_class, 49
Link, 49
List, 50
ListItem, 51

Mark, *35, 36*, 52, *60*, *77*, *88, 89*
MarkTypes, 52
Math, 53, *54*
MathBlock, 53
MathFragment, 54
MathTypes, 55
MediaObject, *11*, *45*, 55, *99*
MediaObjectTypes, 58
MonetaryGrant, 58

Node, 59
NontextualAnnotation, 60
NumberValidator, *47*, 60
NumberValidatorTypes, 61

Organization, 62

Paragraph, 63
Parameter, 64
Periodical, 65
Person, 67
PostalAddress, 68
Product, 70
PropertyValue, 71
PublicationIssue, 72
PublicationVolume, 74

Quote, 76
QuoteBlock, 77

Review, 78

SoftwareApplication, 80
SoftwareEnvironment, 82
SoftwareSession, 83
SoftwareSourceCode, 85
StringValidator, 87
Strong, 88
Subscript, 88
Superscript, 89

Table, 89
TableCell, 92
TableRow, 92
ThematicBreak, 93
Thing, *13*, *28*, *30*, *33*, *35*, *42*, *51*, *63*, *68*, *71, 72*, *83, 84*, 93, *101*
ThingTypes, 94
TupleValidator, 95

Union, 95

ValidatorTypes, 96
Variable, *64*, 96
VariableTypes, 97
VideoObject, 97
VolumeMount, 100