JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/DiscourseFunction.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/features/ClauseStatus.html)
* [Next Class](http://docs.google.com/simplenlg/features/Feature.html)
* [Frames](http://docs.google.com/index.html?simplenlg/features/DiscourseFunction.html)
* [No Frames](http://docs.google.com/DiscourseFunction.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Enum Constants](#id.3znysh7)|
* Field |
* [Method](#id.2et92p0)
* Detail:
* [Enum Constants](#id.1t3h5sf)|
* Field |
* [Method](#id.4i7ojhp)

simplenlg.features

## Enum DiscourseFunction

* java.lang.Object
  + java.lang.Enum<[DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html)>
    - simplenlg.features.DiscourseFunction

All Implemented Interfaces: java.io.Serializable, java.lang.Comparable<[DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html)>

* public enum DiscourseFunction  
  extends java.lang.Enum<[DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html)>  
  An enumeration representing the grammatical function that an element might take. The discourse function is recorded under the Feature.DISCOURSE\_FUNCTION feature and applies to any type of NLGElement.Version: 4.0 Author: A. Gatt and D. Westwater, University of Aberdeen.

### Enum Constant SummaryEnum Constants

|  |
| --- |
| * + Enum Constant and Description |
| * + [**AUXILIARY**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#AUXILIARY) Auxiliaries are the additional verbs added to a verb phrase to alter the meaning being described. |
| * + [**COMPLEMENT**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#COMPLEMENT) Complements are additional components that are required to complement the meaning of a sentence. |
| * + [**CONJUNCTION**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#CONJUNCTION) A conjunction is a word that links items together in a coordinated phrase. |
| * + [**CUE\_PHRASE**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#CUE_PHRASE) Cue phrases are added to sentence to indicate document structure or flow. |
| * + [**FRONT\_MODIFIER**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#FRONT_MODIFIER) Front modifiers are modifiers that apply to clauses. |
| * + [**HEAD**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#HEAD) This represents the main item of the phrase. |
| * + [**INDIRECT\_OBJECT**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#INDIRECT_OBJECT) This is the indirect object of a verb phrase or an additional object that is affected by the action performed. |
| * + [**OBJECT**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#OBJECT) This is the object of a verb phrase and represents the item that the action is performed upon. |
| * + [**POST\_MODIFIER**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#POST_MODIFIER) Post-modifiers, typically adjectives and adverbs, are added after the head of the phrase. |
| * + [**PRE\_MODIFIER**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#PRE_MODIFIER) Pre-modifiers, typically adjectives and adverbs, appear before the head of a phrase. |
| * + [**SPECIFIER**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#SPECIFIER) The specifier, otherwise known as the determiner, is a word that can be placed before a noun in a noun phrase. |
| * + [**SUBJECT**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#SUBJECT) This is the subject of a verb phrase and represents the entity performing the action. |
| * + [**VERB\_PHRASE**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#VERB_PHRASE) The verb phrase highlights the part of a clause that forms the verb phrase. |

### Method SummaryMethods

|  |  |
| --- | --- |
| * + Modifier and Type | * + Method and Description |
| * + static [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) | * + [**valueOf**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#valueOf(java.lang.String))(java.lang.String name) Returns the enum constant of this type with the specified name. |
| * + static [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html)[] | * + [**values**](http://docs.google.com/simplenlg/features/DiscourseFunction.html#values())() Returns an array containing the constants of this enum type, in the order they are declared. |

### Methods inherited from class java.lang.EnumcompareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.ObjectgetClass, notify, notifyAll, wait, wait, wait

### Enum Constant Detail

#### AUXILIARY public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) AUXILIARY Auxiliaries are the additional verbs added to a verb phrase to alter the meaning being described. For example, *will* can be added as an auxiliary to a verb phrase to represent the future tense of the verb, *John* ***will*** *kiss Mary*.

#### COMPLEMENT public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) COMPLEMENT Complements are additional components that are required to complement the meaning of a sentence. For example, *put the bread* ***on the table*** requires the complement *on the table* to make the clause meaningful.

#### CONJUNCTION public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) CONJUNCTION A conjunction is a word that links items together in a coordinated phrase. The most common conjunctions are *and* and *but*.

#### CUE\_PHRASE public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) CUE\_PHRASE Cue phrases are added to sentence to indicate document structure or flow. They normally do not add any semantic information to the phrase. For example, ***Firstly****, let me just say it is an honour to be here.* ***Incidentally****, John kissed Mary last night.*

#### FRONT\_MODIFIER public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) FRONT\_MODIFIER Front modifiers are modifiers that apply to clauses. They are placed in the syntactical structure after the cue phrase but before the subject. For example, *However,* ***last night*** *John kissed Mary.*

#### HEAD public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) HEAD This represents the main item of the phrase. For verb phrases, the head will be the main verb. For noun phrases, the head will be the subject noun. For adjective, adverb and prepositional phrases, the head will be the adjective, adverb and preposition respectively.

#### INDIRECT\_OBJECT public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) INDIRECT\_OBJECT This is the indirect object of a verb phrase or an additional object that is affected by the action performed. This is typically the recipient of *give*. For example, Mary is the indirect object in the phrase *John gives* ***Mary*** *the flower*.

#### OBJECT public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) OBJECT This is the object of a verb phrase and represents the item that the action is performed upon. For example, the flower is the object in the phrase *John gives Mary* ***the flower***.

#### PRE\_MODIFIER public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) PRE\_MODIFIER Pre-modifiers, typically adjectives and adverbs, appear before the head of a phrase. They can apply to noun phrases and verb phrases. For example, *the* ***beautiful*** *woman*, *the* ***ferocious*** *dog*.

#### POST\_MODIFIER public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) POST\_MODIFIER Post-modifiers, typically adjectives and adverbs, are added after the head of the phrase. For example, *John walked* ***quickly***.

#### SPECIFIER public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) SPECIFIER The specifier, otherwise known as the determiner, is a word that can be placed before a noun in a noun phrase. Example specifiers include: *the*, *some*, *a* and *an* as well as the personal pronouns such as *my*, *your*, *their*.

#### SUBJECT public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) SUBJECT This is the subject of a verb phrase and represents the entity performing the action. For example, John is the subject in the phrase ***John*** *gives Mary the flower.*

#### VERB\_PHRASE public static final [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) VERB\_PHRASE The verb phrase highlights the part of a clause that forms the verb phrase. Verb phrases can be formed of a single verb or from a verb with a particle, such as *kiss*, *talk*, *bark*, *fall down*, *pick up*.

### Method Detail

#### values public static [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html)[] values() Returns an array containing the constants of this enum type, in the order they are declared. This method may be used to iterate over the constants as follows: for (DiscourseFunction c : DiscourseFunction.values())   System.out.println(c); Returns:an array containing the constants of this enum type, in the order they are declared

#### valueOf public static [DiscourseFunction](http://docs.google.com/simplenlg/features/DiscourseFunction.html) valueOf(java.lang.String name) Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)Parameters:name - the name of the enum constant to be returned. Returns:the enum constant with the specified name Throws: java.lang.IllegalArgumentException - if this enum type has no constant with the specified name java.lang.NullPointerException - if the argument is null

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Use](http://docs.google.com/class-use/DiscourseFunction.html)
* [Tree](http://docs.google.com/package-tree.html)
* [Deprecated](http://docs.google.com/deprecated-list.html)
* [Index](http://docs.google.com/index-files/index-1.html)
* [Help](http://docs.google.com/help-doc.html)
* [Prev Class](http://docs.google.com/simplenlg/features/ClauseStatus.html)
* [Next Class](http://docs.google.com/simplenlg/features/Feature.html)
* [Frames](http://docs.google.com/index.html?simplenlg/features/DiscourseFunction.html)
* [No Frames](http://docs.google.com/DiscourseFunction.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Enum Constants](#id.3znysh7)|
* Field |
* [Method](#id.2et92p0)
* Detail:
* [Enum Constants](#id.1t3h5sf)|
* Field |
* [Method](#id.4i7ojhp)