
E-HowNet

Release v0.5.2

Mu Yang

Jun 17, 2020

OVERVIEW

1	CKIP E-HowNet Tools	1
1.1	Git	1
1.2	PyPI	1
1.3	Documentation	1
1.4	Author	1
1.5	Requirements	1
1.6	Installation	2
2	Usage	3
2.1	E-HowNet Parser	3
3	License	5
4	E-HowNet Grammar	7
4.1	Text	7
4.2	Node	7
4.3	Valid Expressions	9
5	Parse Nodes	11
5.1	Major Nodes	12
5.2	Partial Nodes	14
6	ehn package	17
6.1	ehn.parse package	17
7	Index	23
8	Module Index	25
	Python Module Index	27
	Index	29

CKIP E-HOWNET TOOLS

1.1 Git

<https://github.com/emfomy/ehownet>

1.2 PyPI

<https://pypi.org/project/ehownet>

1.3 Documentation

<https://ehownet.readthedocs.io/>

1.4 Author

- Mu Yang <<https://muyang.pro>>

1.5 Requirements

- Python 3.6+
- PLY (Python Lex-Yacc) 3.11+
- TreeLib 1.5.5+
- wcwidth 0.1.7+

1.6 Installation

```
pip install -U ehownet
```

USAGE

- See <https://ehownet.readthedocs.io/en/latest/main/grammar.html> for E-HowNet grammar.
- See https://ehownet.readthedocs.io/en/latest/main/parse_node.html for E-HowNet parsing nodes usage.

2.1 E-HowNet Parser

2.1.1 CLI

```
# Usage
ehn-parser <text> [<text> ...]

# Example
ehn-parser \
    "{MusicTool|_x:predication={own|:possession={|PushingButton:whole={x}}}}}" \
    "{InstitutePlace|:telic={or({experiment|:location={~}},{research|:location={~}})}" \
    "TimePoint={},manner={urgent|}"
```

Output:

```
#1
[Entity $x] MusicTool|
├─ [Feature] predication
│   └─ [Entity] own|
│       └─ [Feature] possession
│           └─ [Entity] |PushingButton
│               └─ [Feature] whole
│                   └─ $x

#2
[Entity] InstitutePlace|
├─ [Feature] telic
│   └─ [FunctionEntity]
│       └─ [Function] or
│           ├── [Entity] experiment|
│           │   └─ [Feature] location
│           │       └─ [TildeEntity]
│           └─ [Entity] research|
│               └─ [Feature] location
│                   └─ [TildeEntity]
```

(continues on next page)

(continued from previous page)

```
#3
[Root]
├── [Feature] TimePoint
│   └── [AnyEntity]
├── [Feature] manner
│   └── [Entity] urgent|
```

2.1.2 Python API

```
from ehn.parse import EhnParser

text = '{MusicTool|_x:predication={own|:possession={|PushingButton:whole={x}}}}'

parser = EhnParser()
ress = parser(text, debug=False)
for res in ress:
    res.tree().show()
```

Output:

```
[Entity $x] MusicTool|
├── [Feature] predication
│   └── [Entity] own|
│       └── [Feature] possession
│           └── [Entity] |PushingButton
│               └── [Feature] whole
│                   └── $x
```

CHAPTER THREE

LICENSE



Copyright (c) 2018-2020 CKIP Lab under the [CC BY-NC-SA 4.0 License](#).

E-HOWNET GRAMMAR

This section describes the grammar of the E-HowNet expression.

4.1 Text

- TEXT
 - Any non empty string containing the following characters:
 - * Alphabets and Numbers (A-Za-z0-9)
 - * Unicode Characters (\x80-\U0010FFFF)
 - * |, +, -, ., ?, #.
- NUMBER
 - e.g. 1, 0.1, 1e-4
- COINDEX
 - x, x1, x2, ...
 - x? (refer to the not mentioned subject)

4.2 Node

4.2.1 Entity

- *EhnParseNormalEntity*
 - {TEXT}
 - {TEXT:FEATURE}
 - {TEXT:FEATURE,FEATURE}
 - {TEXT:FEATURE,FEATURE, ...}
 - {TEXT_COINDEX:FEATURE}
 - {TEXT_COINDEX:FEATURE,FEATURE}
 - {TEXT_COINDEX:FEATURE,FEATURE, ...}
- *EhnParseFunctionEntity*

- {FUNCTION}
 - {FUNCTION:FEATURE}
 - {FUNCTION:FEATURE,FEATURE}
 - {FUNCTION:FEATURE,FEATURE,...}
 - {FUNCTION_COINDEX:FEATURE}
 - {FUNCTION_COINDEX:FEATURE,FEATURE}
 - {FUNCTION_COINDEX:FEATURE,FEATURE,...}
 - *EhnParseAnyEntity*
 - { }
 - *EhnParseNameEntity*
 - { "TEXT" }
 - *EhnParseNumberEntity*
 - { NUMBER }
 - *EhnParseTildeEntity*
 - { ~ }
- Deprecated since version 0.6.
- *EhnParseCoindexEntity*
 - { COINDEX }

4.2.2 Feature

- *EhnParseNormalFeature*
 - TEXT=ENTITY
 - TEXT=RESTRICTION
- *EhnParseFunctionFeature*
 - FUNCTION=ENTITY
 - FUNCTION=RESTRICTION

4.2.3 Function

- *EhnParseFunction*
 - TEXT ()
 - TEXT (RESTRICTION)
 - TEXT (ENTITY)
 - TEXT (ENTITY,ENTITY)
 - TEXT (ENTITY,ENTITY,...)

4.2.4 Restriction

- *EhnParseRestriction*
 - /ENTITY
 - /ENTITY_COINDEX

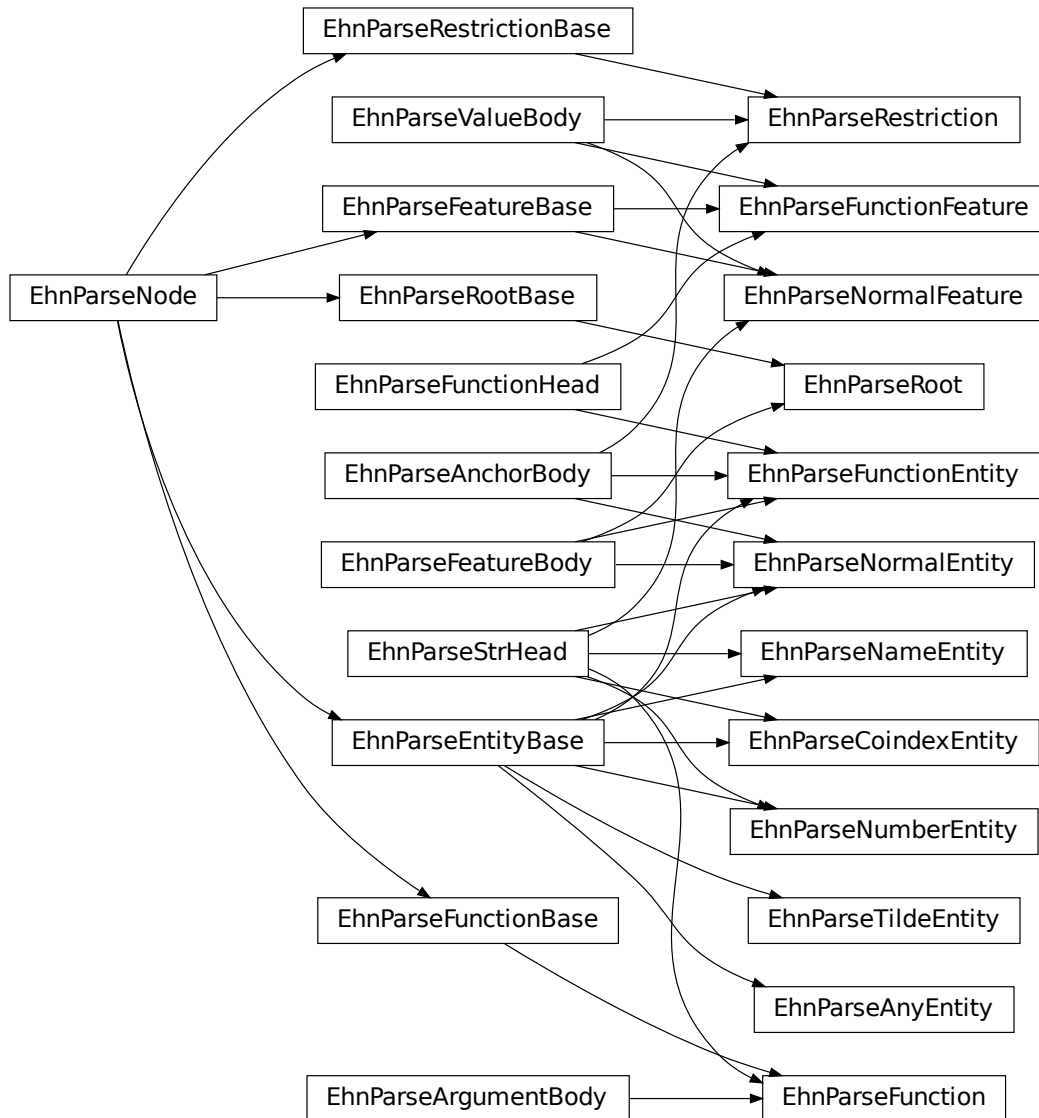
4.3 Valid Expressions

ENTITY or any number of FEATURES joined by , s.

- ENTITY
- FEATURE
- FEATURE, FEATURE
- FEATURE, FEATURE, ...

PARSE NODES

There are five types of nodes in E-HowNet expression — *Entity*, *Feature*, *Function*, *Restriction*, and *Root*.



5.1 Major Nodes

class EhnParseNode

The prototype of E-HowNet parsing nodes.

head: `str`

The head of this node.

get_features()

Get the features (or [] if not exists)

get_arguments ()
Get the arguments (or [] if not exists)

get_value ()
Get the value (or None if not exists)

get_function ()
Get the function (or None if not exists)

get_anchor ()
Get the anchor (or None if not exists)

children ()
Yields all direct child nodes of this node.

descendant ()
Yields all descendant nodes (including self) of this node.

decode ()
Converts to text representation.

tree () → `treelib.Tree`
Generates a tree representation of this node and all its descendant nodes.

5.1.1 Entity

class EhnParseEntityBase

The base class of E-HowNet parsing entity nodes.

Subclasses:

- *EhnParseNormalEntity* A normal entity. Can be an *anchor*.
- *EhnParseFunctionEntity* An entity with *function head*. Can be an *anchor*.
- *EhnParseAnyEntity* A placeholder entity.
- *EhnParseNameEntity* A name entity.
- *EhnParseNumberEntity* A number entity.
- *EhnParseTildeEntity* An entity refers to the parent entity.
- *EhnParseCoindexEntity* An entity refers to an anchor entity.

property features

A list of *Features*.

5.1.2 Feature

class EhnParseFeatureBase

The base class of E-HowNet parsing feature nodes.

Subclasses:

- *EhnParseNormalFeature* A normal feature.
- *EhnParseFunctionFeature* An feature with *function head*.

property value

Can be either *Entity* or *Restriction*.

5.1.3 Function

class EhnParseFunctionBase

The base class of E-HowNet parsing function nodes.

Subclasses:

- *EhnParseFunction*.

property arguments

A list of *Entities* or *Restriction*

5.1.4 Restriction

class EhnParseRestrictionBase

The base class of E-HowNet parsing function nodes.

Subclasses:

- *EhnParseRestriction*. Can be an *anchor*.

property value

Must be an *Entity*.

5.1.5 Root

class EhnParseRootBase

The base class of E-HowNet parsing root nodes. Works similar to entities but is not an entity. Used only in feature-based expressions.

Subclasses:

- *EhnParseRoot*.

property features

A list of *Features*.

5.2 Partial Nodes

5.2.1 Function Head

class EhnParseFunctionHead

The base class of nodes with a function as its head.

Note that the attribute **obj.head** of this object **obj** returns **obj.function.head**.

Subclasses:

- *EhnParseFunctionEntity*
- *EhnParseFunctionFeature*

property function

Must be a *Function*.

5.2.2 Anchor Body

class EhnParseAnchorBody

The base class of anchor nodes.

Subclasses:

- *EhnParseNormalEntity*
- *EhnParseFunctionEntity*
- *EhnParseRestriction*

property anchor

The *Anchor*.

5.2.3 Anchor

class EhnParseAnchor

The coindex target.

head: str

The coindex of this anchor.

EHN PACKAGE

Subpackages

6.1 ehnp.parse package

Subpackages

6.1.1 ehnp.parse.node package

Please refer the tutorial “*Parse Nodes*”.

Submodules

ehnp.parse.node.base module

Please refer the tutorial “*Parse Nodes*”.

class ehnp.parse.node.base.EhnParseNode

Bases: object

E-HowNet Parsing: Base Node

class ehnp.parse.node.base.EhnParseTree (*tree=None, deep=False, node_class=None, identifier=None*)

Bases: treelib.tree.Tree

show (*args, data_property='_tree_label', **kwargs)

Print the tree structure in hierarchy style.

You have three ways to output your tree data, i.e., stdout with `show()`, plain text file with `save2file()`, and json string with `to_json()`. The former two use the same backend to generate a string of tree structure in a text graph.

- Version $\geq 1.2.7a^*$: you can also specify the `line_type` parameter, such as ‘ascii’ (default), ‘ascii-ex’, ‘ascii-exr’, ‘ascii-em’, ‘ascii-emv’, ‘ascii-emh’) to the change graphical form.

Parameters

- **nid** – the reference node to start expanding.
- **level** – the node level in the tree (root as level 0).
- **idhidden** – whether hiding the node ID when printing.

- **filter** – the function of one variable to act on the Node object. When this parameter is specified, the traversing will not continue to following children of node whose condition does not pass the filter.
- **key** – the key param for sorting Node objects in the same level.
- **reverse** – the `reverse` param for sorting Node objects in the same level.
- **line_type** –
- **data_property** – the property on the node data object to be printed.

Returns None

class `ehn.parse.node.base.EhnParseEntityBase`

Bases: `ehn.parse.node.base.EhnParseNode`

E-HowNet Parsing: Base Entity Node

class `ehn.parse.node.base.EhnParseFeatureBase`

Bases: `ehn.parse.node.base.EhnParseNode`

E-HowNet Parsing: Base Feature Node

class `ehn.parse.node.base.EhnParseFunctionBase`

Bases: `ehn.parse.node.base.EhnParseNode`

E-HowNet Parsing: Base Function Node

class `ehn.parse.node.base.EhnParseRestrictionBase`

Bases: `ehn.parse.node.base.EhnParseNode`

E-HowNet Parsing: Base Function Node

class `ehn.parse.node.base.EhnParseRootBase`

Bases: `ehn.parse.node.base.EhnParseNode`

E-HowNet Parsing: Base Root Node

class `ehn.parse.node.base.EhnParseAnchor` (*head=None*)

Bases: object

E-HowNet Parsing: Node Anchor

class `ehn.parse.node.base.EhnParseStrHead` (*head*)

Bases: object

E-HowNet Parsing: Base Node with String Head

class `ehn.parse.node.base.EhnParseFunctionHead` (*function*)

Bases: object

E-HowNet Parsing: Base Node with Function Head

class `ehn.parse.node.base.EhnParseValueBody` (*value*)

Bases: object

E-HowNet Parsing: Base Node with Value

class `ehn.parse.node.base.EhnParseFeatureBody` (**features*)

Bases: object

E-HowNet Parsing: Base Node with Feature

```
class ehnp.parse.node.base.EhnParseArgumentBody (*arguments)
    Bases: object

    E-HowNet Parsing: Base Node with Argument
```

```
class ehnp.parse.node.base.EhnParseAnchorBody (anchor=None)
    Bases: object

    E-HowNet Parsing: Base Node with Anchor
```

ehnp.parse.node.entity module

Please refer the tutorial “*Parse Nodes*”.

```
class ehnp.parse.node.entity.EhnParseNormalEntity (head, *features, anchor=None)
    Bases: ehnp.parse.node.base.EhnParseEntityBase, ehnp.parse.node.base.
            EhnParseStrHead, ehnp.parse.node.base.EhnParseFeatureBody, ehnp.parse.node.
            base.EhnParseAnchorBody
```

E-HowNet Parsing: Normal Entity Node

```
feature_type
    alias of ehnp.parse.node.base.EhnParseFeatureBase
```

```
class ehnp.parse.node.entity.EhnParseFunctionEntity (function, *features, an-
                                                    chor=None)
    Bases: ehnp.parse.node.base.EhnParseEntityBase, ehnp.parse.node.base.
            EhnParseFunctionHead, ehnp.parse.node.base.EhnParseFeatureBody, ehnp.parse.
            node.base.EhnParseAnchorBody
```

E-HowNet Parsing: Function Entity Node

```
feature_type
    alias of ehnp.parse.node.base.EhnParseFeatureBase
```

```
class ehnp.parse.node.entity.EhnParseAnyEntity
    Bases: ehnp.parse.node.base.EhnParseEntityBase
```

E-HowNet Parsing: Any Entity Node

```
class ehnp.parse.node.entity.EhnParseNameEntity (head)
    Bases: ehnp.parse.node.base.EhnParseEntityBase, ehnp.parse.node.base.
            EhnParseStrHead
```

E-HowNet Parsing: Name Entity Node

```
class ehnp.parse.node.entity.EhnParseNumberEntity (head)
    Bases: ehnp.parse.node.base.EhnParseEntityBase, ehnp.parse.node.base.
            EhnParseStrHead
```

E-HowNet Parsing: Number Entity Node

```
class ehnp.parse.node.entity.EhnParseTildeEntity
    Bases: ehnp.parse.node.base.EhnParseEntityBase
```

E-HowNet Parsing: Tilde Entity Node

```
class ehnp.parse.node.entity.EhnParseCoindexEntity (head)
    Bases: ehnp.parse.node.base.EhnParseEntityBase, ehnp.parse.node.base.
            EhnParseStrHead
```

E-HowNet Parsing: Coindex Entity Node

ehn.parse.node.feature module

Please refer the tutorial “*Parse Nodes*”.

```
class ehn.parse.node.feature.EhnParseNormalFeature (head, value)
    Bases:      ehn.parse.node.base.EhnParseFeatureBase,      ehn.parse.node.base.
                EhnParseStrHead, ehn.parse.node.base.EhnParseValueBody
```

E-HowNet Parsing: Normal Feature Node

```
class ehn.parse.node.feature.EhnParseFunctionFeature (function, value)
    Bases:      ehn.parse.node.base.EhnParseFeatureBase,      ehn.parse.node.base.
                EhnParseFunctionHead, ehn.parse.node.base.EhnParseValueBody
```

E-HowNet Parsing: Function Feature Node

ehn.parse.node.other module

Please refer the tutorial “*Parse Nodes*”.

```
class ehn.parse.node.other.EhnParseRoot (*features)
    Bases:      ehn.parse.node.base.EhnParseRootBase,      ehn.parse.node.base.
                EhnParseFeatureBody
```

E-HowNet Parsing: Root Node

```
feature_type
    alias of ehn.parse.node.base.EhnParseFeatureBase
```

```
class ehn.parse.node.other.EhnParseFunction (head, *arguments)
    Bases:      ehn.parse.node.base.EhnParseFunctionBase,      ehn.parse.node.base.
                EhnParseArgumentBody, ehn.parse.node.base.EhnParseStrHead
```

E-HowNet Parsing: Function Node

```
class ehn.parse.node.other.EhnParseRestriction (value, anchor=None)
    Bases:      ehn.parse.node.base.EhnParseRestrictionBase,      ehn.parse.node.base.
                EhnParseValueBody, ehn.parse.node.base.EhnParseAnchorBody
```

E-HowNet Parsing: Restriction Node

```
value_type
    alias of ehn.parse.node.base.EhnParseEntityBase
```

Submodules

6.1.2 ehn.parse.parser module

```
exception ehn.parse.parser.EhnSyntaxError (*args, pos=None)
    Bases: SyntaxError
```

E-HowNet Syntax Error.

```
show_pos (text)
    Show error position.
```

Parameters **text** (*str*) – original input text


```
class ehnl.parse.parser.EhnLexer (**kwargs)
    Bases: ehnl.parse.parser._EhnLexer

    E-HowNet Lexer.

    __call__(data)
        Run tokenization.

class ehnl.parse.parser.EhnParser (lexer=None, **kwargs)
    Bases: ehnl.parse.parser._EhnParser

    E-HowNet Parser.

    __call__(data)
        Run parsing.
```

CHAPTER
SEVEN

INDEX

MODULE INDEX

PYTHON MODULE INDEX

e

- `ehn`, [17](#)
- `ehn.parse`, [17](#)
- `ehn.parse.node`, [17](#)
- `ehn.parse.node.base`, [17](#)
- `ehn.parse.node.entity`, [19](#)
- `ehn.parse.node.feature`, [20](#)
- `ehn.parse.node.other`, [20](#)
- `ehn.parse.parser`, [20](#)

Symbols

`__call__()` (*ehn.parse.parser.EhnLexer method*), 21
`__call__()` (*ehn.parse.parser.EhnParser method*), 21

A

`anchor()` (*EhnParseAnchorBody property*), 15
`arguments()` (*EhnParseFunctionBase property*), 14

C

`children()` (*EhnParseNode method*), 13

D

`decode()` (*EhnParseNode method*), 13
`descendant()` (*EhnParseNode method*), 13

E

`ehn`
 module, 17
`ehn.parse`
 module, 17
`ehn.parse.node`
 module, 17
`ehn.parse.node.base`
 module, 17
`ehn.parse.node.entity`
 module, 19
`ehn.parse.node.feature`
 module, 20
`ehn.parse.node.other`
 module, 20
`ehn.parse.parser`
 module, 20
`EhnLexer` (*class in ehn.parse.parser*), 20
`EhnParseAnchor` (*class in ehn.parse.node.base*), 18
`EhnParseAnchorBody` (*class in ehn.parse.node.base*), 19
`EhnParseAnyEntity` (*class in ehn.parse.node.entity*), 19
`EhnParseArgumentBody` (*class in ehn.parse.node.base*), 18
`EhnParseCoindexEntity` (*class in ehn.parse.node.entity*), 19

`EhnParseEntityBase` (*class in ehn.parse.node.base*), 18
`EhnParseFeatureBase` (*class in ehn.parse.node.base*), 18
`EhnParseFeatureBody` (*class in ehn.parse.node.base*), 18
`EhnParseFunction` (*class in ehn.parse.node.other*), 20
`EhnParseFunctionBase` (*class in ehn.parse.node.base*), 18
`EhnParseFunctionEntity` (*class in ehn.parse.node.entity*), 19
`EhnParseFunctionFeature` (*class in ehn.parse.node.feature*), 20
`EhnParseFunctionHead` (*class in ehn.parse.node.base*), 18
`EhnParseNameEntity` (*class in ehn.parse.node.entity*), 19
`EhnParseNode` (*class in ehn.parse.node.base*), 17
`EhnParseNormalEntity` (*class in ehn.parse.node.entity*), 19
`EhnParseNormalFeature` (*class in ehn.parse.node.feature*), 20
`EhnParseNumberEntity` (*class in ehn.parse.node.entity*), 19
`EhnParser` (*class in ehn.parse.parser*), 21
`EhnParseRestriction` (*class in ehn.parse.node.other*), 20
`EhnParseRestrictionBase` (*class in ehn.parse.node.base*), 18
`EhnParseRoot` (*class in ehn.parse.node.other*), 20
`EhnParseRootBase` (*class in ehn.parse.node.base*), 18
`EhnParseStrHead` (*class in ehn.parse.node.base*), 18
`EhnParseTildeEntity` (*class in ehn.parse.node.entity*), 19
`EhnParseTree` (*class in ehn.parse.node.base*), 17
`EhnParseValueBody` (*class in ehn.parse.node.base*), 18
`EhnSyntaxError`, 20

F

`feature_type` (*ehn.parse.node.entity.EhnParseFunctionEntity attribute*), 19
`feature_type` (*ehn.parse.node.entity.EhnParseNormalEntity attribute*), 19
`feature_type` (*ehn.parse.node.other.EhnParseRoot attribute*), 20
`features()` (*EhnParseEntityBase property*), 13
`features()` (*EhnParseRootBase property*), 14
`function()` (*EhnParseFunctionHead property*), 14

G

`get_anchor()` (*EhnParseNode method*), 13
`get_arguments()` (*EhnParseNode method*), 12
`get_features()` (*EhnParseNode method*), 12
`get_function()` (*EhnParseNode method*), 13
`get_value()` (*EhnParseNode method*), 13

H

`head` (*EhnParseAnchor attribute*), 15
`head` (*EhnParseNode attribute*), 12

M

`module`
 `ehn`, 17
 `ehn.parse`, 17
 `ehn.parse.node`, 17
 `ehn.parse.node.base`, 17
 `ehn.parse.node.entity`, 19
 `ehn.parse.node.feature`, 20
 `ehn.parse.node.other`, 20
 `ehn.parse.parser`, 20

S

`show()` (*ehn.parse.node.base.EhnParseTree method*), 17
`show_pos()` (*ehn.parse.parser.EhnSyntaxError method*), 20

T

`tree()` (*EhnParseNode method*), 13

V

`value()` (*EhnParseFeatureBase property*), 13
`value()` (*EhnParseRestrictionBase property*), 14
`value_type` (*ehn.parse.node.other.EhnParseRestriction attribute*), 20