parse-entities

Parse HTML character references: fast, spec-compliant, positional information.

Install

npm:

```
npm install parse-entities
```

Use

```
var decode = require('parse-entities')

decode('alpha &amp; bravo')
// => alpha & bravo

decode('charlie &copycat; delta')
// => charlie ©cat; delta

decode('echo &copy; foxtrot &#8800; golf &#x1D306; hotel')
// => echo © foxtrot ≠ golf 𝌆 hotel
```

API

`parseEntities(value[, options])`

`options`

`OPTIONS.ADDITIONAL`

Additional character to accept (`string?` , default: `''`). This allows other characters, without error, when following an ampersand.

`OPTIONS.ATTRIBUTE`

Whether to parse `value` as an attribute value (`boolean?` , default: `false`).

`OPTIONS.NONTERMINATED`

Whether to allow non-terminated entities (`boolean` , default: `true`). For example, `&copycat` for `©cat` . This behaviour is spec-compliant but can lead to unexpected results.

`OPTIONS.WARNING`

Error handler (`Function?`).

`OPTIONS.TEXT`

Text handler (`Function?`).

`options.reference`

Reference handler (`Function?`).

`options.warningContext`

Context used when invoking `warning` (`'*'` , optional).

`options.textContext`

Context used when invoking `text` (`'*'` , optional).

`options.referenceContext`

Context used when invoking `reference` (`'*'` , optional)

`options.position`

Starting `position` of `value` (`Location` or `Position` , optional). Useful when dealing with values nested in some sort of syntax tree. The default is:

```
{
  start: {line: 1, column: 1, offset: 0},
  indent: []
}
```

Returns

`string` — Decoded `value` .

`function warning(reason, position, code)`

Error handler.

Context

`this` refers to `warningContext` when given to `parseEntities` .

Parameters

`reason`

Human-readable reason for triggering a parse error (`string`).

`position`

Place at which the parse error occurred (`Position`).

`code`

Identifier of reason for triggering a parse error (`number`).

The following codes are used:

Code	Example	Note
1	`foo &amp bar`	Missing semicolon (named)
2	`foo &#123 bar`	Missing semicolon (numeric)
3	`Foo &bar baz`	Ampersand did not start a reference
4	`Foo &#`	Empty reference
5	`Foo &bar; baz`	Unknown entity

6	`Foo &#128; baz`	Disallowed reference
7	`Foo &#xD800; baz`	Prohibited: outside permissible unicode range

`function text(value, location)`

Text handler.

Context

`this` refers to `textContext` when given to `parseEntities`.

Parameters

`VALUE`

String of content (`string`).

`LOCATION`

Location at which `value` starts and ends (`Location`).

`function reference(value, location, source)`

Character reference handler.

Context

`this` refers to `referenceContext` when given to `parseEntities`.

Parameters

`VALUE`

Encoded character reference (`string`).

`LOCATION`

Location at which `value` starts and ends (`Location`).

`SOURCE`

Source of character reference (`Location`).

Related

- `stringify-entities` — Encode HTML character references
- `character-entities` — Info on character entities
- `character-entities-html4` — Info on HTML4 character entities
- `character-entities-legacy` — Info on legacy character entities
- `character-reference-invalid` — Info on invalid numeric character references

License