

1 Function Documentation

concat-counter

Concatenate two counters. The state of the resulting counter, obtained by `counter-at()`, is an array of both states.

Parameters

```
concat-counter(  
  counter-1: counter-like numbify-counter ,  
  counter-2: counter-like numbify-counter  
) -> numbify-counter
```

slice-counter

Slices a range of levels from a counter. The state of the resulting counter can be obtained using `counter-at()`.

Parameters

```
slice-counter(  
  counter: counter-like ,  
  end: integer none ,  
  start: integer  
) -> numbify-counter
```

end integer or none

The end-level of the slice (inclusive). If none, the slice goes to the last level of the counter.

start integer

The start-level of the slice.

Default: 1

counter-at

Retrieves the value of a counter-like or numbify-counter at the given location. Behaves similar to [counter.at\(\)](#).

This function is contextual.

Parameters

```
counter-at(  
  counter: counter-like numbify-counter ,  
  selector: label selector location function  
) -> array
```

selector label or selector or location or function

The place at which the counter's value should be retrieved.

counter-get

Retrieves the value of the counter at the current location.

This is equivalent to `counter-at(counter, here())`.

This function is contextual.

Parameters

`counter-get(counter: counter-like numbify-counter) -> array`

numbify-numbering

Applies a numbering to the value of a counter-like or numbify-counter. Since this function behaves similarly to [numbering\(\)](#) when provided with the state of a counter-like, only the behaviour of this function when provided with the state of a numbify-counter will be described here.

Parameters

```
numbify-numbering(  
  numbering: str function array ,  
  numbers: array ,  
  separator: content  
) -> function
```

numbering str or function or array

This parameter defines the numbering format. If it's a string or a function, the numbering will be applied to the concatenation of the two states. If it's an array of two numberings, each numbering will be applied to its corresponding state. The results will then be joined using the separator.

numbers array

The value of the counter-like or numbify-counter.

separator content

In case of an array-numbering, this separator is used to join the individual numberings.

Default: `"."`

override-counter

Creates a numbering function that replaces the usual counter with the provided counter. The numbering is applied to the counter using numbify-numbering().

The returned function is contextual.

Parameters

```
override-counter(  
  counter: counter-like numbify-counter ,  
  numbering: str function array ,  
  separator: content  
) -> function
```

counter counter-like or numbify-counter

The new counter.

numbering str or function or array

The numbering that will be applied to the counter.

separator content

The separator that will be used to join the individual numberings in case of an array-numbering.

Default: "."

heading-level-range

Returns a selector that selects a range of heading levels.

Parameters

```
heading-level-range(  
  end: integer ,  
  start: integer  
) -> selector
```

end integer

The end-level (inclusive).

start integer

The start-level (inclusive).

Default: 1

reset-counter

Reset the counter. The update only occurs if the resulting content is put into the document.

counter (counter-like):

Parameters

```
reset-counter(counter) -> content
```

reset-counter-rule

Reset the counter. This is a helper function that can be used in a show-rule.

Parameters

```
reset-counter-rule(counter: counter-like) -> function
```

is-numbify-counter

Checks if the argument was constructed by one of the counter-operations.

Parameters

```
is-numbify-counter(arg: any) -> boolean
```

is-concat-counter

Checks if the argument was constructed by the concat-counter operation.

Parameters

```
is-concat-counter(arg: any) -> boolean
```

is-slice-counter

Checks if the argument was constructed by the slice-counter operation.

Parameters

```
is-slice-counter(arg: any) -> boolean
```

from-counter-like

Turns a counter-like into a counter by constructing the counter from the argument if necessary. A counter-like is a counter or something that a counter can be constructed from.

Parameters

```
from-counter-like(counter: counter-like) -> counter
```

from-counter-like-or-numbify

Convert argument into counter or numbify-counter. If the argument is a numbify-counter, it is returned as is. Otherwise, the argument is converted into a counter if necessary.

Parameters

`from-counter-like-or-numbify(counter: counter-like numbify-counter)`