## private/1

### Description

```
private(Functor/Arity)
private(Functor1/Arity1, Functor2/Arity2, ...)
private([Functor1/Arity1, Functor2/Arity2, ...])

private(Functor//Arity)
private(Functor1//Arity1, Functor2//Arity2, ...)
private([Functor1//Arity1, Functor2//Arity2, ...])
```

Declares private predicates and private grammar rule non-terminals. A private predicate can only be called from the object containing the private directive. A private non-terminal can only be used in a call of the `phrase/2` and `phrase/3` methods from the object containing the private directive.

### Template and modes

```
private(+predicate_indicator_term)
private(+non_terminal_indicator_term)
```

### Examples

```
:- private(counter/1).

:- private(init/1, free/1).

:- private([data/3, key/1, keys/1]).
```