Languages-beta: SIMPLE-Funcons-Index

The PLanCompS Project

`Languages-beta/SIMPLE/SIMPLE-Funcons-Index/SIMPLE-Funcons-Index.cbs`[*]

Computations

Normal computation

Flowing

```
[ Funcon sequential
    Alias seq
  Funcon effect
  Funcon if-true-else
    Alias if-else
  Funcon while-true
    Alias while ]
```

Giving

```
[ Funcon initialise-giving
  Funcon give
  Funcon given
  Funcon left-to-right-repeat ]
```

[*] Suggestions for improvement: `plancomps@gmail.com`.
Issues: `https://github.com/plancomps/CBS-beta/issues`.

Binding

```
  [ Type     environments
    Alias    envs
  Datatype   identifiers
    Alias    ids
   Funcon    initialise-binding
   Funcon    bind-value
    Alias    bind
   Funcon    bound-value
    Alias    bound
   Funcon    scope
   Funcon    collateral  ]
```

Storing

```
[ Funcon    initialise-storing
Datatype    variables
   Alias    vars
  Funcon    allocate-variable
   Alias    alloc
  Funcon    allocate-initialised-variable
   Alias    alloc-init
  Funcon    assign
  Funcon    assigned  ]
```

Interacting

Input

```
[ Funcon  read  ]
```

Output

```
[ Funcon  print  ]
```

Abnormal computation

Failing

```
[ Funcon  finalise-failing
  Funcon  checked  ]
```

Throwing

```
[ Funcon finalise-throwing
  Funcon throw
  Funcon handle-thrown ]
```

Returning

```
[ Funcon return
  Funcon handle-return ]
```

Values

Value Types

```
 [ Type values
  Alias vals
Funcon is-equal
  Alias is-eq ]
```

Primitive values

Booleans

```
[ Funcon true
  Funcon false
  Funcon not ]
```

Integers

```
[ Type    natural-numbers
   Alias  nats
  Funcon  integer-add
   Alias  int-add
  Funcon  integer-subtract
   Alias  int-sub
  Funcon  integer-multiply
   Alias  int-mul
  Funcon  integer-divide
   Alias  int-div
  Funcon  integer-modulo
   Alias  int-mod
  Funcon  integer-negate
   Alias  int-neg
  Funcon  integer-is-less
   Alias  is-less
  Funcon  integer-is-less-or-equal
   Alias  is-less-or-equal
  Funcon  integer-is-greater
   Alias  is-greater
  Funcon  integer-is-greater-or-equal
   Alias  is-greater-or-equal
  Funcon  decimal-natural
   Alias  decimal ]
```

The null value

```
[ Datatype  null-type
    Funcon  null-value
     Alias  null ]
```

Composite values

Sequences of values

```
[ Funcon  length
  Funcon  index ]
```

Tuples

[*Datatype* tuples]

Vectors

[*Datatype* vectors
  Funcon vector
  Funcon vector-elements]

Abstraction values

Generic abstractions

[*Funcon* closure]

Functions

[*Datatype* functions
  Funcon function
  Funcon apply]

Patterns

[*Datatype* patterns
  Funcon pattern
  Funcon match]