PUNCTUATION CHARACTERS

**,** // to separate parameters in a function signature, or the items of a tuple

**.** // (1) to call a field or a function on a variable (field access or method call)

// (2) to qualify a type/method with its package name

**'** // used as alternative name in parameters (prime)

**"** // to delineate a literal string

**"""**  // to delineate a documentation string

**(** // 1) start of line: start of a tuple

// 2) middle of line: method call

**( )** // parentheses, for function or behavior parameters,

//or making a tuple (values separated by ,),

// or making an enumeration (values separated by |)

**[** // 1) start of line: start of an array literal

// 2) middle of line: generic formal parameters

**[ ]** // 1) to indicate a generic type, for example Range

// 2) to indicate the return type of an FFI function call

**{ }** // function type (???)

**:** // 1) after a variable: is followed by the type name, or to indicate a function // return type

// a type constraint

**;** // only used to separate expressions on the same line

**=** // (1) (destructive) assignment

// (2) in use alias = packname

! // a type that is an *alias* of another type

// not

**?** // 1) T? type T or None

// 2) partial functions

// 3) a call to a C function that could raise an error

**-** // 1) start of line: unary negation

// 2) middle of line: subtraction

**\_** // 1) to indicate a private variable, constructor, function, behavior

// 2) to ignore a tuple item in a pattern match

**~** // partial application

**^** // denotes an *ephemeral* type

**|** // 1) separates the types in an enumeration (or): the value can be any of these // types

// 2) starts a matching branch

**&** // 1) separates the types in a complex type (and) ): the value is of all of these // types

// 2) intersection

**@** // FFI call

**//** // comments

**/\***

**\*/** // multi-line or block comments

**=>** // 1) starts a function body

// 2) starts the code of a matching branch

**->** // arrow type

// viewpoint

**.\_i** // where i = 1,2,… means the ith item of a tuple

**<:** // “is a subtype of” or “can be substituted for”