

Senet Language Reference Manual

Program structure

A Senet program is composed of two parts like so:

```

setup{
...
}
turn{
...
}

```

The first, setup part, is akin to C's main function; it is a sequence of statements and control flows, to be executed until the end of the part is reached. The second part is a sequence of phase definitions, each describing how to interpret player input to affect the game state.

Everything is an object

In Senet, everything built into the language, and everything the user defines, is an object. This means that member functions and variables for all variables can be accessed using the dot syntax

VARIABLE.MEMBER

There are four categories of objects: primitive types, classes, functions, and phases.

Primitive types

This is a closed category, no new primitive types can be created.

Type	Meaning
int	32-bit integer
char	Character
list	Linked list
str	String (list of char)
bool	Boolean (True or False)
set	Unique set of objects
void	Type of None, a value used to represent the absence of a value

Classes

Classes are essentially compound, user-defined types. A class object may have multiple class or primitive type member variables, in addition to functions.

Functions

Functions are callable objects that take as input a list of variables and output exactly one. Functions are defined using the syntax

```

TYPE FUNCTIONNAME (TYPE1 VAR1, TYPE2 VAR2, ...) {
...
}

```

where the ellipsis in the input list stands in for any number of additional variables, and the ellipsis in the body stands in for a series of statements, which must end with a return statement *return**x*;;, where x is the result of the function, of the return type of the function.

Phases

Phases are only defined in the turn part of a program. They are defined using function-like syntax, but without any return type, and must have at least one “int” input variable, representing the index of the active player. One phase must be named “begin”, this is called after the setup part is complete. Like functions, no phase can reach the end of its statements, it must call itself or another phase, or, alternatively, declare the game over and exit (or restart).