

Control Structures - If

The simplest control structure is the if statement. An if statement tests a condition and then executes one sequence of expressions if the condition was true, or a different sequence if the it was false. For example:

```
$ name, letter, type, possibilities.
```

```
name := "John Jacob Jinglehiemer-Schmidt".
```

```
letter := name choose.
```

```
if ("aeiou" has letter)
```

```
then [
```

```
type := "vowels".
```

```
possibilities := 5.
```

```
]
```

```
else [
```

```
type := "consonants".
```

```
possibilities := 21.
```

].

letter && "is one of" && possibilities.name && "possible" && type.

Formatting

The two conditional sequences are between square brackets. There are several fine points about its formatting: Between the square brackets, each expression ends in a period, although the last one is optional. Outside the brackets, there is only a period after the last close bracket, as this is the end of the if statement. Even though this statement was written out over nine lines, it could be written in fewer: line breaks and spaces are interchangeable.

Blocks

The bracketed statements, along with the square brackets, are called a block. In other languages, blocks are often wrapped in curly brackets, or the words 'begin' and 'end'. Unlike some languages, in Glyphic Script, the square brackets are always required, even if the block has only one expression inside of it.

There are two other small points of interest in the above example:

»

First: The choose message was used in a different way. Instead of a number being the receiver, a string was the receiver. In this case, choose picks a random character of the string. In takes no arguments. Notice that the same message can do different (although often similar) actions when used with different receivers. Computer Science calls this polymorphism. It is one of the powerful features of an object-oriented, message passing language like Glyphic Script.

»

Second: In the last expression, the message name was used with a number (the expression was 'possibilities.name'). There is a difference between the number 21 and the string "21". The number is a value that can take part in arithmetic. The string is a sequence of characters that can be used to communicate with the user. We needed to turn the number into a string so that we can make it part of the final string result. Name is the message that results in a string which is the name of the receiver.

Nesting

If statements can be nested, like all control structures:

\$ n, msg.

n := choose 1 to 100.

if n > 33.3 then [

if n > 66.6

then [msg := "huge"]

else [msg := "medium"].

]

else [msg := "small"].

msg.

This example also demonstrates several other things: If the condition is just a simple expression (not a message) then it doesn't need to be in parentheses. If there is only one expression in the block, since it is the last expression, it doesn't need a period.