

# Control Structures - Switch

The switch statement lets you choose one of several different blocks of code to execute depending on some value. For example:

```
$ rank, suit.
```

```
rank := choose 1 to 13.
```

```
switch rank;
```

```
case 1 do [ rank := "Ace" ];
```

```
case 11 do [ rank := "Jack" ];
```

```
case 12 do [ rank := "Queen" ];
```

```
case 13 do [ rank := "King" ];
```

```
default do [ rank := rank name ].
```

```
switch (choose 1 to 4);
```

```
case 1 do [ suit := "Clubs" ];
```

```
case 2 do [ suit := "Hearts" ];
```

```
case 3 do [ suit := "Diamonds" ];
```

```
case 4 do [ suit := "Spades" ].
```

```
return rank && "of" && suit.
```

Consider the first switch statement in this example: The switch statement is actually a consists of a switch expression, four case expressions and a default expression. The switch expression specifies the value of rank as the test value used in the following case expressions.

Each case expression specifies a value to match against. If it equals the test value, then the block of the case expression is executed, and the switch statement is finished.

After all the case expressions, if none of them matched, then the block of the default expression is executed. The default expression is optional, as shown in the second switch statement. If it is missing, and no case expression matched the test value, then the switch statement does nothing.

!!

The punctuation of the switch statement must be followed carefully. Pay particular attention to the semicolons: There are semicolons between each part of the statement. There is a period only at the end of the whole switch statement.