Pre-defined Global Constants

**TRUE**

The typical true value.

**FALSE**

The false itself.

**NIL**

The nil itself.

**STDIN**

The standard input. The default value for stdin.

**STDOUT**

The standard output. The default value for stdout.

**STDERR**

The standard error output. The default value for stderr.

**ENV**

The hash containing current environment variables.

**ARGV**

The alias to the $<.

**ARGV**

The alias to the $*.

**DATA**

The file object of the script, pointing just after __END__.

**RUBY_VERSION**

The ruby version string (VERSION was depreciated).

**RUBY_RELEASE_DATE**

The release date string.

**RUBY_PLATFORM**

The platform identifier.

Expressions

**Terms**

Terms are expressions that may be a basic type (listed above), a shell command, variable reference, constant reference, or method invocation.

**Operators and Precedence**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>:/</td>
<td>7</td>
</tr>
<tr>
<td>*</td>
<td>6</td>
</tr>
<tr>
<td>+</td>
<td>5</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>&lt;&lt;</td>
<td>4</td>
</tr>
<tr>
<td>&gt;&gt;</td>
<td>3</td>
</tr>
<tr>
<td>==</td>
<td>2</td>
</tr>
<tr>
<td>=~</td>
<td>2</td>
</tr>
<tr>
<td>!</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

Control Expressions

if bool-exp (then)
    body
else
    body
end

unless bool-exp (then)
    body
else
    body
end

expr if bool-exp
expr unless bool-exp

case target-exp

Pre-defined Global Constants

**TRUE**

The typical true value.

**FALSE**

The false itself.

**NIL**

The nil itself.

**STDIN**

The standard input. The default value for stdin.

**STDOUT**

The standard output. The default value for stdout.

**STDERR**

The standard error output. The default value for stderr.

**ENV**

The hash containing current environment variables.

**ARGV**

The alias to the $<.

**ARGV**

The alias to the $*.

**DATA**

The file object of the script, pointing just after __END__.

**RUBY_VERSION**

The ruby version string (VERSION was depreciated).

**RUBY_RELEASE_DATE**

The release date string.

**RUBY_PLATFORM**

The platform identifier.

Expressions

**Terms**

Terms are expressions that may be a basic type (listed above), a shell command, variable reference, constant reference, or method invocation.

**Operators and Precedence**

<table>
<thead>
<tr>
<th>Operator</th>
<th>Precedence</th>
</tr>
</thead>
<tbody>
<tr>
<td>:/</td>
<td>7</td>
</tr>
<tr>
<td>*</td>
<td>6</td>
</tr>
<tr>
<td>+</td>
<td>5</td>
</tr>
<tr>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>&lt;&lt;</td>
<td>4</td>
</tr>
<tr>
<td>&gt;&gt;</td>
<td>3</td>
</tr>
<tr>
<td>==</td>
<td>2</td>
</tr>
<tr>
<td>=~</td>
<td>2</td>
</tr>
<tr>
<td>!</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>1</td>
</tr>
</tbody>
</table>

Control Expressions

if bool-exp (then)
    body
else if bool-exp (then)
    body
else
    body
end

unless bool-exp (then)
    body
else
    body
end

expr if bool-exp
expr unless bool-exp

case target-exp

Defining a Class

One must begin with capital class.

```
class Identifier [< Superclass ]; ... ; end

# Singleton classes, or idclasses;
# add methods to a single instance
# @obj can be self
class << @obj; ... ; end
```

Defining a Module

```
module Identifier; ... ; end
```

Defining a Method

```
def method_name[arg_list]; ... ; end
def expr.method_name[arg_list]; ... ; end
```

Arguments may have default values (varname = expr).

## Method definitions may not be nested.

```
method_name may be an operator: =, ==, ===, =>, =~, <<, >>, +, -, *, /, \$, %, **, <<>>, +@, -@, [], [] (the last takes two arguments)
```

Access Restriction

```
public
totally accessible.
```

```
protected
accessible only by instances of class and direct descendants. Even through hash relationships (see below)
```

```
private
accessible only by instances of class.
```

```