

# Стандартные функции языка CLIPS

Стандартные арифметические  
функции

## Арифметические функции с одним аргументом

<i>(abs &lt;numeric-expression&gt;)</i>	Возвращает модуль указанного значения
<i>(float &lt;numeric-expression&gt;)</i>	Возвращает аргумент, приведенный к типу float.
<i>(integer &lt;numeric-expression&gt;)</i>	Возвращает аргумент, приведенный к типу integer.

## Арифметические функции с несколькими аргументами

<p><i>(div &lt;numeric-expression&gt;</i> <i>&lt;numeric-expression&gt;+)</i></p>	<p>Возвращает значение первого аргумента, разделенного на каждый из остальных аргументов. Деление производится с помощью операций над целыми.</p>
<p><i>(max &lt;numeric-expression&gt;</i> <i>&lt;numeric-expression&gt;+)</i></p>	<p>Возвращает значение самого большого аргумента.</p>
<p><i>(min &lt;numeric-expression&gt;</i> <i>&lt;numeric-expression&gt;+)</i></p>	<p>Возвращает значение самого маленького аргумента.</p>

## Арифметические функции с несколькими аргументами

$(+ \text{ <numeric-expression> } \text{ <numeric-expression>+})$	Возвращает сумму аргументов
$(- \text{ <numeric-expression> } \text{ <numeric-expression>+})$	Возвращает разницу между первым аргументом и всеми остальными.
$(* \text{ <numeric-expression> } \text{ <numeric-expressions>})$	Возвращает произведение аргументов
$(/ \text{ <numeric-expression> } \text{ <numeric-expression>+})$	Возвращает значение первого аргумента, разделенного на все остальные.

# Стандартные функции языка CLIPS

Математические функции

## Математические функции с одним аргументом

$(\text{acos } \langle \text{numeric-expression} \rangle)$	Возвращает арккосинус аргумента
$(\text{acosh } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арккосинус аргумента
$(\text{acot } \langle \text{numeric-expression} \rangle)$	Возвращает арккотангенс аргумента
$(\text{acoth } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арккотангенс аргумента

# Математические функции с одним аргументом

$(\text{acsc } \langle \text{numeric-expression} \rangle)$	Возвращает арккосеканс аргумента
$(\text{acsch } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арккосеканс аргумента
$(\text{asec } \langle \text{numeric-expression} \rangle)$	Возвращает арксеканс аргумента
$(\text{asech } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арксеканс аргумента

# Математические функции с одним аргументом

$(\text{asin } \langle \text{numeric-expression} \rangle)$	Возвращает арксинус аргумента
$(\text{asinh } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арксинус аргумента
$(\text{atan } \langle \text{numeric-expression} \rangle)$	Возвращает арктангенс аргумента
$(\text{atanh } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический арктангенс аргумента

# Математические функции с одним аргументом

$(\cos \langle \text{numeric-expression} \rangle)$	Возвращает косинус аргумента
$(\cosh \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический косинус аргумента
$(\cot \langle \text{numeric-expression} \rangle)$	Возвращает котангенс аргумента (в радианах)
$(\coth \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический котангенс аргумента
$(\csc \langle \text{numeric-expression} \rangle)$	Возвращает косеканс

# Математические функции с одним аргументом

$(\text{csch } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический косеканс аргумента
$(\text{sec } \langle \text{numeric-expression} \rangle)$	Возвращает секанс аргумента
$(\text{sech } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический косеканс аргумента
$(\text{sin } \langle \text{numeric-expression} \rangle)$	Возвращает синус аргумента
$(\text{sinh } \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический синус аргумента

# Математические функции с одним аргументом

$(\tan \langle \text{numeric-expression} \rangle)$	Возвращает тангенс аргумента
$(\tanh \langle \text{numeric-expression} \rangle)$	Возвращает гиперболический тангенс аргумента

# Стандартные функции языка CLIPS

Функции конвертации

## Функции конвертации

<i>(deg-grad &lt;numeric-expression&gt;)</i>	Возвращает значение аргумента, переведенного из градусов в градиенты.
<i>(deg-rad &lt;numeric-expression&gt;)</i>	Возвращает значение аргумента, переведенного из градусов в радианы.
<i>(exp &lt;numeric-expression&gt;)</i>	Возвращает экспоненту аргумента.

# Функции конвертации

<i>(grad-deg &lt;numeric-expression&gt;)</i>	Возвращает значение аргумента, переведенного из градиентов в градусы.
<i>(log &lt;numeric-expression&gt;)</i>	Возвращает логарифм по основанию e аргумента.
<i>(log10 &lt;numeric-expression&gt;)</i>	Возвращает логарифм по основанию 10 аргумента.

# Функции конвертации

<i>(mod &lt;numeric-expression&gt; &lt;numeric-expression&gt;)</i>	Возвращает остаток от деления первого аргумента на второй.
<i>(pi)</i>	Возвращает число $\pi$ .
<i>(rad-deg &lt;numeric-expression&gt;)</i>	Возвращает значение аргумента, переведенного из радианов в градусы

# Функции конвертации

<i>(round &lt;numeric-expression&gt;)</i>	Возвращает значение аргумента округленного до ближайшего целого.
<i>(sqrt &lt;numeric-expression&gt;)</i>	Возвращает корень квадратный аргумента
<i>(** &lt;numeric-expression&gt; &lt;numeric-expression&gt;)</i>	Возвращает первый аргумент, возведенный в степень второго аргумента.

# Стандартные функции языка CLIPS

Логические (предикатные) функции

## Логические функции с одним аргументом

(evenp <expression>)	TRUE, если <expression> четное целое, иначе FALSE
(floatp <expression>)	TRUE, если <expression> типа float, иначе FALSE
(integerp <expression>)	TRUE, если <expression> типа integer, иначе FALSE
(lexemp <expression>)	TRUE, если <expression> типа string или symbol, иначе FALSE

## Логические функции с одним аргументом

<i>(multifieldp &lt;expression&gt;)</i>	TRUE, если <expression> – multifield, иначе FALSE
<i>(not &lt;expression&gt;)</i>	TRUE, если значение аргумента FALSE, иначе TRUE.
<i>(numberp &lt;expression&gt;)</i>	TRUE, если <expression> типа float или integer, иначе FALSE
<i>(oddp &lt;expression&gt;)</i>	TRUE, если <expression> нечетное целое, иначе FALSE
<i>(stringp &lt;expression&gt;)</i>	TRUE, если <expression> является строкой, иначе FALSE
<i>(symbolp &lt;expression&gt;)</i>	TRUE, если <expression> – идентификатор, иначе FALSE

## Логические функции с несколькими аргументами

<i>(and &lt;expression&gt;+)</i>	TRUE, если каждый из аргументов равен TRUE, иначе FALSE
<i>(eq &lt;expression&gt; &lt;expression&gt;+)</i>	TRUE, если первый аргумент равен по типу и по значению всем остальным аргументам, иначе FALSE
<i>(neq &lt;expression&gt; expression&gt;+)</i>	TRUE, если его первый аргумент не равен по типу или по значению всем остальным, иначе FALSE
<i>(or &lt;expression&gt;+)</i>	TRUE, если хотя бы один из аргументов равен TRUE, иначе FALSE

## Логические функции с несколькими аргументами

<p><math>(= \langle \textit{numeric-expression} \rangle</math> <math>\langle \textit{numeric-expression} \rangle +)</math></p>	<p>TRUE, если его первый аргумент равен по числовому значению всем остальным аргументам, иначе FALSE</p>
<p><math>(\langle \rangle \langle \textit{numeric-expression} \rangle</math> <math>\langle \textit{numeric-expressions} \rangle)</math></p>	<p>TRUE, если его первый аргумент не равен по числовому значению всем остальным аргументам, иначе FALSE</p>
<p><math>(&gt; \langle \textit{numeric-expression} \rangle</math> <math>\langle \textit{numeric-expressions} \rangle)</math></p>	<p>TRUE, если для всех аргументов, аргумент <math>n - 1</math> больше, чем аргумент <math>n</math>, иначе FALSE</p>

# Стандартные функции языка CLIPS

Строковые функции

# Строковые функции

<i>(<b>lowercase</b> &lt;string-or-symbol-expression&gt;)</i>	Возвращает аргумент, приведенный к нижнему регистру
<i>(<b>str-cat</b> &lt;expression&gt;*)</i>	Возвращает все аргументы, объединенные в строку
<i>(<b>str-compare</b> &lt;string-or-symbol-expression&gt; &lt;string-or-symbol-expression&gt;)</i>	Возвращает ноль, если оба аргумента равны, иначе отличное от нуля значение

# Строковые функции

<p><i>(str-index &lt;lexeme-expression&gt; &lt;lexeme-expression&gt;)</i></p>	<p>Возвращает позицию первого аргумента во втором аргументе, если первый аргумент подстрока второго, иначе FALSE</p>
<p><i>(str-length &lt;string-or-symbol-expression&gt;)</i></p>	<p>Возвращает длину строки</p>

# Строковые функции

<p><i>(sub-string</i> <i>&lt;begin-integer-expression&gt;</i> <i>&lt;end-integer-expression&gt;</i> <i>&lt;string-expression&gt;</i>)</p>	<p>Возвращает подстроку из <i>&lt;string-expression&gt;</i>, начинающуюся в позиции <i>&lt;begin-integer-expression&gt;</i> и заканчивающуюся в позиции <i>&lt;end-integer-expression&gt;</i></p>
<p><i>(sym-cat &lt;expression&gt;*)</i></p>	<p>Возвращает все аргументы, соединенные в идентификатор</p>

# Строковые функции

*(`uppercase` <`string-or-symbol-expression`>)*

Возвращает  
аргумент,  
приведенный к  
верхнему регистру

## Логические функции с несколькими аргументами

$(\geq \langle \text{numeric-expression} \rangle \langle \text{numeric-expression} \rangle +)$	TRUE, если для всех аргументов, аргумент $n - 1$ больше или равен аргументу $n$ , иначе FALSE
$(< \langle \text{numeric-expression} \rangle \langle \text{numeric-expression} \rangle +)$	TRUE, если для всех аргументов, аргумент $n - 1$ меньше, чем аргумент $n$ , иначе FALSE
$(<= \langle \text{numeric-expression} \rangle \langle \text{numeric-expression} \rangle +)$	TRUE, если для всех аргументов, аргумент $n - 1$ меньше или равен аргументу $n$ , иначе FALSE

# Стандартные функции языка CLIPS

Функции ввода-вывода

## Функции ввода-вывода

<p><i>(close</i> <i>[&lt;logical-name&gt;])</i></p>	<p>Закрывает файл, ассоциированный с логическим именем <i>&lt;logical-name&gt;</i> (или все файлы, если аргумент опущен). Возвращает TRUE, если файл был успешно закрыт, иначе FALSE</p>
<p><i>(format &lt;logical-name&gt;</i> <i>&lt;string-expression&gt;</i> <i>&lt;expression&gt;*)</i></p>	<p>Вычисляет и печатает в <i>&lt;logical-name&gt;</i> ноль или более выражений, отформатированных согласно <i>&lt;string-expression&gt;</i></p>

## Функции ввода-вывода

<p><i>(open &lt;file-name&gt; &lt;logical-name&gt; [&lt;mode&gt;])</i></p>	<p>Открывает файл &lt;file-name&gt; в указанном режиме («r», «w», «r+» или «a») и ассоциирует с ним логическое имя &lt;logical-name&gt;. Возвращает TRUE, если файл успешно открыт, иначе FALSE.</p>
<p><i>(printout &lt;logical-name&gt; &lt;expression&gt;*)</i></p>	<p>Вычисляет и печатает в &lt;logical-name&gt; ноль или более выражений.</p>

## Функции ввода-вывода

<p><i>(read [<i>&lt;logical-name&gt;</i>] )</i></p>	<p>Читает указанное поле из указанного <i>&lt;logical-name&gt;</i> (<i>stdin</i>, если ничего не указано). Возвращает поле в случае успеха или EOF, если нет никаких входных значений.</p>
<p><i>(readline [<i>&lt;logical-name&gt;</i>] )</i></p>	<p>Читает строку из указанного <i>&lt;logical-name&gt;</i> (<i>stdin</i>, если ничего не указано). Возвращает строку в случае успеха или EOF, если нет никаких входных значений.</p>

## Функции ввода-вывода

<i>(remove &lt;file-name&gt;)</i>	Удаляет файл <file-name>.
<i>(rename &lt;old-file-name&gt; &lt;new-file-name&gt;)</i>	Переименовывает файл <old-file-name> в <new-file-name>