

ГЛОБАЛЬНЫЕ ОБЪЕКТЫ, АТРИБУТЫ И МЕТОДЫ.	6
COMP (NAME)	6
FOOTAGE (NAME)	6
THISCOMP	6
THISLAYER	6
THISPROPERTY	6
TIME	6
COLORDEPTH	6
POSTERIZE TIME (FRAMESPERSECOND)	7
VALUE	7
МЕТОДЫ ПРЕОБРАЗОВАНИЯ ВРЕМЕНИ.	7
TIME TO FRAMES (T = TIME + THISCOMP.DISPLAYSTARTTIME, FPS = 1.0 / THISCOMP.FRAME DURATION, ISDURATION = FALSE)	7
FRAMES TO TIME (FRAMES, FPS = 1.0 / THISCOMP.FRAME DURATION)	7
TIME TO TIMECODE (T = TIME + THISCOMP.DISPLAYSTARTTIME, TIMECODEBASE = 30, ISDURATION = FALSE)	8
TIME TO NTSC TIMECODE (T = TIME + THISCOMP.DISPLAYSTARTTIME, NTSCDROPFRAME = FALSE, ISDURATION = FALSE)	8
TIME TO FEET AND FRAMES (T = TIME + THISCOMP.DISPLAYSTARTTIME, FPS = 1.0 / THISCOMP.FRAME DURATION, FRAMESPERFOOT = 16, ISDURATION = FALSE)	8
TIME TO CURRENT FORMAT (T = TIME + THISCOMP.DISPLAYSTARTTIME, FPS = 1.0 / THISCOMP.FRAME DURATION, ISDURATION = FALSE)	8
МЕТОДЫ МАТЕМАТИЧЕСКИХ ВЕКТОРОВ.	9
ADD (VEC1, VEC2)	9
SUB (VEC1, VEC2)	9
MUL (VEC, AMOUNT)	9
DIV (VEC, AMOUNT)	9
CLAMP (VALUE, LIMIT1, LIMIT2)	9
DOT (VEC1, VEC2)	9
CROSS (VEC1, VEC2)	10
NORMALIZE (VEC)	10
LENGTH (VEC)	10
LENGTH (POINT1, POINT2)	10
LOOK AT (FROMPOINT, ATPOINT)	10
МЕТОДЫ СЛУЧАЙНЫХ ЧИСЕЛ.	11
SEEDRANDOM (OFFSET, TIMELESS=FALSE)	11
RANDOM ()	11
RANDOM (MAXVALORARRAY)	12
RANDOM (MINVALORARRAY, MAXVALORARRAY)	12
GAUSSRANDOM ()	12
GAUSSRANDOM (MAXVALORARRAY)	12
GAUSSRANDOM (MINVALORARRAY, MAXVALORARRAY)	13
NOISE (VALORARRAY)	13
МЕТОДЫ ИНТЕРПОЛЯЦИИ.	13
LINEAR (T, TMIN, TMAX, VALUE1, VALUE2)	13
LINEAR (T, VALUE1, VALUE2)	14
EASE (T, VALUE1, VALUE2)	14
EASE (T, TMIN, TMAX, VALUE1, VALUE2)	14
EASEIN (T, VALUE1, VALUE2)	14
EASEIN (T, TMIN, TMAX, VALUE1, VALUE2)	15

EASEOUT (T, VALUE1, VALUE2)	15
EASEOUT (T, TMIN, TMAX, VALUE1, VALUE2)	15
МЕТОДЫ ПРЕОБРАЗОВАНИЯ ЦВЕТОВ	15
RGBTOHSL (RGBAARRAY)	15
HSLTORGB (HSLAARRAY)	15
ДРУГИЕ МАТЕМАТИЧЕСКИЕ МЕТОДЫ.....	16
DEGREESTORADIANS (DEGREES).....	16
RADIANSTODEGREES (RADIANS).....	16
АТТРИБУТЫ И МЕТОДЫ КОМПОЗИЦИИ.	16
LAYER (INDEX).....	16
LAYER (NAME)	16
LAYER (OTHERLAYER, RELINDEX).....	16
MARKER.....	17
MARKER.KEY (INDEX).....	17
MARKER.KEY (NAME).....	17
MARKER.NEARESTKEY (T)	17
MARKER.NUMKEYS	18
NUMLAYERS	18
ACTIVECAMERA.....	18
WIDTH.....	18
HEIGHT	18
DURATION.....	18
NTSCDROPFRAME	18
DISPLAYSTARTTIME.....	19
FRAMEDURATION	19
SHUTTERANGLE.....	19
SHUTTERPHASE	19
BGCOLOR.....	19
PIXELASPECT.....	19
NAME.....	19
АТТРИБУТЫ И МЕТОДЫ ВИДЕОРЕДА.	19
WIDTH.....	19
HEIGHT	20
DURATION.....	20
FRAMEDURATION	20
NTSCDROPFRAME	20
PIXELASPECT.....	20
NAME.....	20
АТТРИБУТЫ И МЕТОДЫ ВЛОЖЕННЫХ ОБЪЕКТОВ СЛОЯ.....	20
SOURCE	20
SOURCETIME (T = TIME).....	21
EFFECT (NAME)	21
EFFECT (INDEX)	21
MASK (NAME).....	21
MASK (INDEX)	21
ОБЩИЕ АТТРИБУТЫ И МЕТОДЫ СЛОЯ.....	21

WIDTH	21
HEIGHT	22
INDEX.....	22
PARENT	22
HASPARENT.....	22
INPOINT	22
OUTPOINT.....	22
STARTTIME.....	22
HASVIDEO.....	23
HASAUDIO	23
ACTIVE.....	23
ENABLED.....	23
AUDIOACTIVE	23
SAMPLEIMAGE (POINT, RADIUS = [.5, .5], POSTEFFECT=TRUE, T=TIME)	23
АТРИБУТЫ И МЕТОДЫ СВОЙСТВ СЛОЯ.....	24
ANCHORPOINT.....	24
POSITION	24
SCALE	24
ROTATION	24
OPACITY.....	24
AUDIOLEVELS.....	25
TIMEREMAP	25
MARKER.KEY (INDEX).....	25
MARKER.KEY (NAME).....	25
MARKER.NEARESTKEY (T)	25
MARKER.NUMKEYS	26
NAME.....	26
АТРИБУТЫ И МЕТОДЫ 3D-СЛОЯ.	26
ORIENTATION.....	26
ROTATIONX.....	26
ROTATIONY	26
ROTATIONZ	26
LIGHTTRANSMISSION.....	26
CASTSSHADOWS	27
ACCEPTSSHADOWS	27
ACCEPTSLIGHTS.....	27
AMBIENT.....	27
DIFFUSE	27
SPECULAR	27
SHININESS	27
METAL.....	27
МЕТОДЫ ПРЕОБРАЗОВАНИЯ ПРОСТРАНСТВА СЛОЯ.	27
TOCOMP (POINT, T=TIME)	28
FROMCOMP (POINT, T=TIME)	28
TOWORLD (POINT, T=TIME)	28
FROMWORLD (POINT, T=TIME)	28
TOCOMPVEC (VEC, T=TIME)	28
FROMCOMPVEC (VEC, T=TIME).....	29

TOWORLDVEC (VEC, T=TIME)	29
FROMWORLDVEC (VEC, T=TIME).....	29
FROMCOMPTOSURFACE (POINT, T=TIME).....	29
АТРИБУТЫ И МЕТОДЫ КАМЕРЫ.	29
POINTOFINTEREST	30
ZOOM	30
DEPTHOFFIELD.....	30
FOCUSDISTANCE.....	30
APERTURE	30
BLURLEVEL	30
ACTIVE.....	30
АТРИБУТЫ И МЕТОДЫ СВЕТА.	31
POINTOFINTEREST	31
INTENSITY	31
COLOR.....	31
CONEANGLE	31
CONEFEATHER	31
SHADOWDARKNESS	31
SHADOWDIFFUSION	31
АТРИБУТЫ И МЕТОДЫ ЭФФЕКТОВ.	31
ACTIVE.....	31
PARAM (NAME).....	32
PARAM (INDEX).....	32
АТРИБУТЫ И МЕТОДЫ МАСКИ.....	32
MASKOPACITY	32
MASKFEATHER	32
MASKEXPANSION.....	32
INVERT	32
АТРИБУТЫ И МЕТОДЫ СВОЙСТВА.....	33
VALUE	33
VALUEATTIME (T).....	33
VELOCITY	33
VELOCITYATTIME (T).....	33
SPEED.....	33
SPEEDATTIME (T)	33
WIGGLE (FREQ, AMP, OCTAVES=1, AMP_MULT=.5, T=TIME)	34
TEMPORALWIGGLE (FREQ, AMP, OCTAVES=1, AMP_MULT=.5, T=TIME)	34
SMOOTH (WIDTH=.2, SAMPLES=5, T=TIME).....	35
LOOPIN (TYPE="CYCLE", NUMKEYFRAMES=0).....	35
ТИП ЦИКЛА - РЕЗУЛЬТАТ	35
LOOPOUT (TYPE="CYCLE", NUMKEYFRAMES=0).....	36
LOOPINDURATION (TYPE="CYCLE", DURATION=0).....	36
LOOPOUTDURATION (TYPE="CYCLE", DURATION=0).....	36
KEY (INDEX).....	36
KEY (MARKERNAME)	37
NEARESTKEY (T)	37

NUMKEYS	37
PROPERTYGROUP (COUNTUP = 1)	37
PROPERTYINDEX	37
NAME.....	38
АТРИБУТЫ И МЕТОДЫ КЛЮЧА.....	38
VALUE.....	38
TIME.....	38
INDEX.....	38
АТРИБУТЫ MARKERKEY.....	39
DURATION.....	39
COMMENT	39
CHAPTER.....	39
URL.....	40
FRAMETARGET	40
EVENTCUEPOINT	40
CUEPOINTNAME	40
PARAMETERS	40
ДОСТУП ВЫРАЖЕНИЙ К ТОЧКАМ КОНТУРОВ ФИГУР, МАСОК И ШТРИХОВ КИСТИ.....	40
МЕТОДЫ ВЫРАЖЕНИЙ:	40
МЕТОД КОНТУРА POINTS ({}): {PATHPROPERTY}.POINTS (T = TIME)	40
МЕТОД INTANGENTS ({}): {PATHPROPERTY}.INTANGENTS (T = TIME).....	41
МЕТОД OUTTANGENTS ({}): {PATHPROPERTY}.OUTTANGENTS (T = TIME)	41
МЕТОД ISCLOSED ({}): {PATHPROPERTY}.ISCLOSED ({})	41
МЕТОД POINTONPATH ({}): {PATHPROPERTY}.POINTONPATH (PERCENTAGE = 0,5, T = TIME)	41
МЕТОД TANGENTONPATH ({}): {PATHPROPERTY}.TANGENTONPATH (PERCENTAGE = 0,5, T = TIME)	42
МЕТОД NORMALONPATH ({}): {PATHPROPERTY}.NORMALONPATH (PERCENTAGE = 0,5, T = TIME)	43
МЕТОД CREATEPATH ({}): {PATHPROPERTY}.CREATEPATH (POINTS = [[0,0], [100,0], [100,100], [0,100]], INTANGENTS = [], OUTTANGENTS = [], IS_CLOSED = TRUE).....	43
АНИМАЦИЯ, УПРАВЛЯЕМАЯ ДАННЫМИ.....	45
АТРИБУТ ВИДЕОРЕДА SOURCETEXT {FOOTAGEITEM}.SOURCETEXT	45
АТРИБУТ ВИДЕОРЕДА SOURCEDATA {FOOTAGEITEM}.SOURCEDATA	45
МЕТОД ВИДЕОРЕДА DATAVALUE ({}): {FOOTAGEITEM}.DATAVALUE (DATAPATH)	45
МЕТОД ВИДЕОРЕДА DATAKEYCOUNT ({}): {FOOTAGEITEM}.DATAKEYCOUNT (DATAPATH).....	46
МЕТОД ВИДЕОРЕДА DATAKEYTIMES ({}): {FOOTAGEITEM}.DATAKEYTIMES (DATAPATH, T0 = STARTTIME, T1=ENDTIME).....	46
FOOTAGE ("SAMPLE.MGJSON").DATAKEYTIMES ([0], 1, 3)	46
МЕТОД ВИДЕОРЕДА DATAKEYVALUES ({}): {FOOTAGEITEM}.DATAKEYVALUES (DATAPATH, T0 = STARTTIME, T1=ENDTIME)	47

Глобальные объекты, атрибуты и методы.

comp (name)

Тип возвращения: композиция.

Тип аргумента: *name* – это строка.

Извлекает другую композицию по имени.

footage (name)

Тип возвращения: видеоряд.

Тип аргумента: *name* – это строка.

Извлекает элемент видеоряда по имени.

thisComp

Тип возвращения: композиция.

Представляет композицию, содержащую выражение.

thisLayer

Тип возвращения: слой, свет или камеру.

Представляет слой, содержащий выражение. Поскольку **thisLayer** является объектом по умолчанию, использовать его не обязательно.

Пример: можно начать выражение с **thisLayer.width** или **width** и получить тот же результат.

thisProperty

Тип возвращения: свойство.

Представляет свойство, содержащее выражение.

Пример: при написании выражения по свойству «Поворот», можно начать выражение `cthisProperty` для обращения к свойству «Поворот».

time

Тип возвращения: число.

Представляет время композиции в секундах, в которых вычисляется выражение.

colorDepth

Тип возвращения: число.

Возвращает значение разрядности цвета проекта.

Пример: `colorDepth` возвращает число 16, когда разрядность цвета проекта составляет 16 бит на канал.

6

posterizeTime (framesPerSecond)

Тип возвращения: число.

Тип аргумента: *framesPerSecond* – это число.

Значение *framesPerSecond* становится частотой кадров, на основе которой работает оставшаяся часть выражения. Выражение позволяет установить частоту кадров для свойства ниже частоты кадров композиции.

Пример: следующее выражение обновляет значение свойства произвольным значением один раз в секунду: `posterizeTime (1); random ()`

value

Тип возвращения: число, массив или строка.

Представляет текущее значение для свойства, содержащего выражение.

Методы преобразования времени.

timeToFrames (t = time + thisComp.displayStartTime, fps = 1.0 / thisComp.frameDuration, isDuration = false)

Тип возвращения: число.

Тип аргумента: *t* и *fps* – это числа; *isDuration* – это логическое значение.

Преобразует значение *t*, которое возвращается к значению времени текущей композиции по умолчанию, то есть целому числу кадров. Число кадров в секунду указывается в аргументе *fps*, который возвращается к значению частоты кадров текущей композиции по умолчанию ($1.0/\text{thisComp.frameDuration}$).

Аргумент *isDuration* со значением по умолчанию «ложь» принимает значение «истина», если значение *t*

Представляет собой разницу между двумя значениями времени вместо абсолютного значения времени. Абсолютные значения времени округляются в меньшую сторону к отрицательной бесконечности, а продолжительность — в сторону противоположную нулю (к положительным значениям).

framesToTime (frames, fps = 1.0 / thisComp.frameDuration)

Тип возвращения: число.

Тип аргумента: *frames* и *fps* — это числа.

Инверсия `timeToFrames`.

Возвращает время, соответствующее требуемому аргументу *frames*. Оно не должно обязательно являться целым числом. Описание аргумента *fps* см. в `timeToFrames`.

timeToTimecode (t = time + thisComp.displayStartTime, timecodeBase = 30, isDuration = false)

Тип возвращения: строка.

Тип аргумента: *t* и *timecodeBase* – это числа; *isDuration* – это логическое значение.

Преобразует значение *t* в строку, представляющую тайм-код. Описание аргументов *isDuration* см. *timeToFrames*. Значение *timecodeBase*, которое возвращается к значению по умолчанию 30, определяет число кадров в одной секунде.

timeToNTSCTimecode (t = time + thisComp.displayStartTime, ntscDropFrame = false, isDuration = false)

Тип возвращения: строка.

Тип аргумента: *t* – это число, а *ntscDropFrame* и *isDuration* – это логические значения. **Преобразует** *t* в строку, представляющую собой тайм-код NTSC. Описание аргументов *isDuration* см. *timeToFrames*.

Если значение параметра *ntscDropFrame* – «ложь» (по умолчанию), то результирующая строка будет являться тайм-кодом NTSC без пропуска кадров. Если значение параметра *ntscDropFrame* – «истина», то результирующая строка будет являться тайм-кодом NTSC с пропуском кадров.

timeToFeetAndFrames (t = time + thisComp.displayStartTime, fps = 1.0 / thisComp.frameDuration, framesPerFoot = 16, isDuration = false)

Тип возвращения: строка.

Тип аргумента: *t*, *fps* и *framesPerFoot* – это числа, а *isDuration* – это логическое значение.

Преобразует значение *t* в строку, представляющую собой метраж пленки и кадры. Описание аргументов *t*, *fps* и *isDuration* см. *timeToFrames*. Аргумент *framesPerFoot* задает число кадров в одном футе пленки. По умолчанию он имеет значение 16 кадров — типичное значение для пленки 35 мм.

timeToCurrentFormat (t = time + thisComp.displayStartTime, fps = 1.0 / thisComp.frameDuration, isDuration = false)

Тип возвращения: строка.

Тип аргумента: *t* и *fps* – это числа; *isDuration* – это логическое значение.

Преобразует значение *t* в строку, представляющую собой время в текущем формате отображения «Настройки проекта». Описание всех аргументов см. **timeToFrames**.

В After Effects CS5.5 и более поздних версиях к функции **timeToCurrentFormat()** добавлен дополнительный аргумент **ntscDropFrame**. По умолчанию: **ntscDropFrame = thisComp.ntscDropFrame**.

Примечание. Если требуется внести во внешний вид тайм-кода более серьезные изменения, используйте метод **timeToCurrentFormat** или другие методы **timeTo** для создания тайм-кода вместо использования эффекта «Тайм-код» или «Числа». Создайте текстовый слой, добавьте выражение к свойству «Исходный текст» и введите **timeToCurrentFormat()** в поле выражения.

С помощью этого метода можно форматировать и анимировать текст тайм-кода. Кроме того, тайм-код использует тот же стиль отображения, который задан в параметрах текущего проекта.

Методы математических векторов.

Функции математических векторов – это глобальные методы, позволяющие выполнять действия с массивами, обрабатывая их как математические векторы. В отличие от встроенных методов JavaScript, таких как **Math.sin**, эти методы не используют префикс **Math**. Если не указано иное, методы математических векторов обрабатывают размеры и возвращают значение, являющееся размером самого большого объекта массива, заполняя недостающие элементы нулями.

Пример: выражение `add ([10, 20], [1, 2, 3])` возвращает значения `[11, 22,3]`.

add (vec1, vec2)

Тип возвращения: массив.

Тип аргумента: *vec1* и *vec2* – это массивы. Добавляет два вектора.

sub (vec1, vec2)

Тип возвращения: массив.

Тип аргумента: *vec1* и *vec2* – это массивы. Вычитает два вектора.

mul (vec, amount)

Тип возвращения: массив.

Тип аргумента: *vec* – это массив, а *amount* – это число.

Умножает каждый элемент вектора на число, определенное аргументом *amount*.

div (vec, amount)

Тип возвращения: массив.

Тип аргумента: *vec* – это массив, а *amount* – это число.

Делит каждый элемент вектора на число, определенное аргументом *amount*.

clamp (value, limit1, limit2)

Тип возвращения: число или массив.

Тип аргумента: *value*, *limit1* и *limit2* – это числа или массивы.

Значение каждого компонента *value* ограничено диапазоном значений соответствующих значений *limit1* и *limit2*.

dot (vec1, vec2)

Тип возвращения: число.

Тип аргумента: *vec1* и *vec2* – это массивы.

Возвращает скалярное (внутреннее) произведение аргументов вектора.

cross (vec1, vec2)

Тип возвращения: массив [2 или 3].

Тип аргумента: *vec1* и *vec2* – это массивы [2 или 3].

Возвращает векторное перекрестное произведение *vec1* и *vec2*. Более подробные сведения см. в справочнике математических векторов или руководстве JavaScript.

normalize (vec)

Тип возвращения: массив.

Тип аргумента: *vec* – это массив.

Нормализует вектор таким образом, что его длина становится равной 1,0.

Метод **normalize**– это простой способ выполнения операции **div (vec,length (vec))**.

length (vec)

Тип возвращения: число.

Тип аргумента: *vec* – это массив. **Возвращает** длину вектора *vec*.

length (point1, point2)

Тип возвращения: число.

Тип аргумента: *point1* и *point2* – это массивы.

Возвращает расстояние между двумя точками. Аргумент *point2* является дополнительным.

Пример: `length (point1, point2)` совпадает с `length (sub (point1,point2))`.

Пример: можно добавить это выражение к свойству камеры «Фокусное расстояние» для закрепления фокальной плоскости на точке обзора камеры, чтобы точка обзора находилась в фокусе:

```
length (position, pointOfInterest)
```

lookAt (fromPoint, atPoint)

Тип возвращения: массив [3].

Тип аргумента: *fromPoint* и *atPoint* — это массивы [3].

Аргумент *fromPoint* – это точка в пространстве слоя World, ориентацию которой требуется изменить. Аргумент *atPoint* – это точка в пространстве слоя World, к которой нужно направить слой. Возвращаемое значение можно использовать в качестве выражения для свойства «Ориентация», чтобы ось Z слоя была направлена в сторону аргумента *atPoint*.

Данный метод особенно удобен при работе с камерами и светом. При использовании этого выражения применительно к камере отключите функцию автоориентации.

Пример: при использовании этого выражения со свойством источника направленного света «Ориентация» источник света будет указывать на опорную точку слоя номер 1 в той же самой композиции:

```
lookAt (position, thisComp.layer (1).position)
```

Методы случайных чисел.

Примечание. Метод покачивания, используемый для произвольного изменения значения свойства, относится к категории атрибутов свойства и методов.

seedRandom (offset, timeless=false)

Тип возвращения: отсутствует.

Тип аргумента: *offset* – это число, а *timeless* – это логическое значение.

В методах **random** и **gaussRandom** используется начальное значение, управляющее последовательностью чисел. По умолчанию начальное значение вычисляется в качестве функции уникального идентификатора слоя, свойства внутри слоя, текущего времени и значения смещения равного 0. Используйте метод **seedRandom**, чтобы задать смещению значение, отличное от 0, для создания другой случайной последовательности.

Используйте значение **true** для аргумента *timeless*, чтобы текущее время не использовалось в качестве входного для случайного начального значения. С помощью значения **true**, используемого для аргумента *timeless*, можно создавать случайные числа, которые не будут изменяться в зависимости от времени расчета.

Значение *offset*, в отличие от *timeless* также может использоваться для управления начальным значением функции **покачивания**.

Пример: это выражение в свойстве «Непрозрачность» задает значению «Непрозрачность» случайное значение, которое не меняется с течением времени:

```
seedRandom (123456,true); random ()*100
```

При умножении на 100 в этом примере выполняется преобразование значения в диапазоне 0–1, возвращаемом по методу **random**, в число в диапазоне 0–100. Этот диапазон чаще используется при работе со свойством «Непрозрачность», диапазон значений которого составляет 0–100 %.

random ()

Тип возвращения: число.

Возвращает случайное число в диапазоне значений от 0 до 1.

В After Effects CC и CS6 поведение выражения **random ()** становится более произвольным, когда идентификаторы слоя расположены близко друг к другу. Поведение выражения **wiggle ()** не

затрагивается.

random (maxValOrArray)

Тип возвращения: число или массив.

Тип аргумента: *maxValOrArray* – это число или массив.

Если *maxValOrArray* является числом, этот метод **Возвращает** число в диапазоне от 0 до значения *maxValOrArray*.

Если *maxValOrArray* является массивом, этот метод **Возвращает** массив того же размера, что и *maxValOrArray*, в котором каждый компонент находится в диапазоне от 0 до соответствующего компонента *maxValOrArray*.

random (minValOrArray, maxValOrArray)

Тип возвращения: число или массив.

Тип аргумента: *minValOrArray* и *maxValOrArray* – это числа или массивы.

Если *minValOrArray* и *maxValOrArray* являются числами, этот метод **Возвращает** число в диапазоне от *minValOrArray* до *maxValOrArray*.

Если аргументы являются массивами, этот метод **Возвращает** массив того же размера, что и аргумент большего размера, в котором каждый компонент находится в диапазоне от соответствующего компонента *minValOrArray* до соответствующего компонента *maxValOrArray*.

Пример: выражение **random ([100, 200], [300, 400])** возвращает массив, первое значение которого находится в диапазоне от 100 до 300, а второе значение находится в диапазоне от 200 до 400. Если размеры двух входных массивов не совпадают, значения большего размера самого короткого массива заполняются нулями.

gaussRandom ()

Тип возвращения: число.

Возвращает случайное число. Результаты имеют распределение Гаусса (колоколообразное).

Примерно 90 % результатов находятся в диапазоне 0–1, а оставшиеся 10 % находятся вне этого диапазона.

gaussRandom (maxValOrArray)

Тип возвращения: число или массив.

Тип аргумента: *maxValOrArray* – это число или массив.

Когда *maxValOrArray* является числом, этот метод **Возвращает** случайное число. Примерно 90 % результатов находятся в диапазоне от 0 до *maxValOrArray*, а оставшиеся 10 % находятся вне этого диапазона.

Когда *maxValOrArray* является массивом, этот метод **Возвращает** массив случайных значений того

же размера, что *imaxValOrArray*. 90 % значений находятся в диапазоне от 0 до *maxValOrArray*, а оставшиеся 10 % находятся вне этого диапазона. Результаты имеют распределение Гаусса (колоколообразное).

gaussRandom (minValOrArray, maxValOrArray)

Тип возвращения: число или массив.

Тип аргумента: *minValOrArray* и *maxValOrArray* — это числа или массивы.

Если *minValOrArray* и *imaxValOrArray* являются числами, этот метод **Возвращает** случайное число. Примерно 90 % результатов находятся в диапазоне от *minValOrArray* до *maxValOrArray*, а оставшиеся 10 % находятся вне этого диапазона.

Если аргументы являются массивами, этот метод **Возвращает** массив случайных чисел того же размера, что и аргумент большего размера.

Для каждого компонента примерно 90 % результатов находятся в диапазоне от соответствующего компонента *minValOrArray* до соответствующего компонента *maxValOrArray*, а оставшиеся 10 % находятся вне этого диапазона. Результаты имеют распределение Гаусса (колоколообразное).

noise (valOrArray)

Тип возвращения: число.

Тип аргумента: *valOrArray* — это число или массив [2 или 3].

Возвращает число в диапазоне от -1 до 1. В действительности шум не случаен. Он основан на шуме Перлина. Это означает, что возвращаемые значения для двух входных значений, расположенных рядом, будут расположены рядом. Этот тип шума полезен при формировании последовательности чисел, которые, по-видимому, являются случайными и значительно не отличаются друг от друга, как в случае анимации любого естественного случайного движения.

Пример:

```
rotation + 360*noise (time)
```

Методы интерполяции.

Для всех методов интерполяции аргумент *t* часто является аргументом **time** или **value**, однако он может иметь и другие значения. Если *t* является аргументом **time**, интерполяция между значениями выполняется за указанное время. Если *t* является аргументом **value**, выражение преобразует один диапазон значений в другой.

linear (t, tMin, tMax, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: *t*, *tmin* и *tmax* — это числа, а *val1* и *val2* — это числа или массивы.

Возвращает *val1* при значении $t \leq tmin$.

Возвращает *val2* при значении $t \geq tmax$.

Возвращает линейную интерполяцию между *value1* и *value2* при выполнении условия $tMin < t < tMax$.

Пример: это выражение в свойстве «Непрозрачность» приводит к плавному линейному переходу значений свойства «Непрозрачность» от 20% до 80% в интервале от 0 до 6 секунд:

```
linear (time, 0, 6, 20, 80)
```

Этот метод, как и другие методы интерполяции, можно использовать для преобразования одного диапазона значений в другой.

Пример: это выражение в свойстве «Непрозрачность» преобразует значения «Непрозрачности» из диапазона 0–100 % в диапазон 20–80 %:

```
linear (value, 0, 100, 20, 80)
```

linear (t, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: *t* — это число, а *value1* и *value2* — это числа или массивы.

Возвращает значение, которое линейно интерполируется из *value1* в *value2*, а значения *t* заключаются в пределах от 0 до 1.

Возвращает *value1* при значении $t \leq 0$.

Возвращает *value2* при значении $t \geq 1$.

ease (t, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: *t* — это число, а *value1* и *value2* — это числа или массивы.

Аналогично методу **linear** с теми же аргументами, за исключением того, что интерполяция ускоряется и замедляется, а значение скорости равно 0 в начальной и конечной точках. В результате использования этого метода анимация становится плавной.

ease (t, tMin, tMax, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: *t*, *tMin* и *tMax* — это числа, а *value1* и *value2* — это числа или массивы.

Аналогично методу **linear** с теми же аргументами, за исключением того, что интерполяция ускоряется и замедляется, а значение скорости равно 0 в начальной и конечной точках. В результате использования этого метода анимация становится плавной.

easeIn (t, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: t – это число, $avalue1$ и $value2$ – это числа или массивы.

Аналогично методу **ease**, за исключением того, что значение касательной равно 0 только на стороне $value1$, а интерполяция линейна на стороне $value2$.

easeIn (t, tMin, tMax, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: t , $tMin$ и $tMax$ – это числа, а $value1$ и $value2$ – это числа или массивы.

Аналогично методу **ease**, за исключением того, что значение касательной равно 0 только на стороне $tMin$, а интерполяция линейна на стороне $tMax$.

easeOut (t, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: t – это число, $avalue1$ и $value2$ – это числа или массивы.

Аналогично методу **ease**, за исключением того, что значение касательной равно 0 только на стороне $value2$, а интерполяция линейна на стороне $value1$.

easeOut (t, tMin, tMax, value1, value2)

Тип возвращения: число или массив.

Тип аргумента: t , $tMin$ и $tMax$ – это числа, а $value1$ и $value2$ – это числа или массивы.

Аналогично методу **ease**, за исключением того, что значение касательной равно 0 только на стороне $tMax$, а интерполяция линейна на стороне $tMin$.

Методы преобразования цветов

rgbToHsl (rgbaArray)

Тип возвращения: массив [4].

Тип аргумента: $rgbaArray$ – это массив [4].

Преобразует цвет пространства RGBA в цвет пространства HSLA. Входными данными является массив нормализованных значений красных, зеленых, синих и альфа-каналов, находящихся в диапазоне от 0,0 до 1,0. Результирующим значением является массив значений каналов тона, насыщенности, освещенности и альфа-каналов, также находящихся в диапазоне от 0,0 до 1,0.

Пример:

```
rgbToHsl.effect ("Change Color") ("Color To Change")
```

hslToRgb (hslaArray)

Тип возвращения: массив [4].

Тип аргумента: $hslaArray$ – это массив [4].

Преобразует цвет пространства HSLA в цвет пространства RGBA. Это преобразование противоположно преобразованию, выполняемому по методу **rgbToHsl**.

Другие математические методы.

degreesToRadians (degrees)

Тип возвращения: число.

Тип аргумента: *degrees* – это число. Преобразование градусов в радианы.

radiansToDegrees (radians)

Тип возвращения: число.

Тип аргумента: *radians* – это число. **Преобразует** радианы в градусы.

Атрибуты и методы композиции.

layer (index)

Тип возвращения: слой, свет или камера. **Тип аргумента:** *index* – это число.

Извлекает слой по числу (порядковый номер на панели «Таймлайн»).

Пример:

```
thisComp.layer(3)
```

layer (name)

Тип возвращения: слой, свет или камера. **Тип аргумента:** *name* – это строка.

Извлекает слой по имени. Имена согласуются по имени слоя или имени источника при отсутствии имени слоя. При наличии повторяющихся имен After Effects использует первое (самое верхнее) имя на панели «Таймлайн».

Пример:

```
thisComp.layer ("Solid 1")
```

layer (otherLayer, relIndex)

Тип возвращения: слой, свет или камера.

Тип аргумента: *otherLayer* – это объект слоя, а *relIndex* – это число.

Извлекает слой, который находится над или под слоем *otherLayer* на расстоянии *relIndex* слоев.

Пример

```
thisComp.layer (thisLayer, 1).active
```

возвращает значение «истина», если следующий нижний слой на панели «Таймлайн» активен.

marker

Тип возвращения: MarkerProperty.

Примечание. Невозможно получить доступ к маркеру композиции по номеру маркера. При работе в проекте, созданном в предыдущей версии After Effects, в котором в выражениях используются номера маркеров композиции, необходимо изменить эти вызовы, чтобы вместо этого использовать `marker.key (name)`. Поскольку именем маркера композиции по умолчанию является номер, процесс преобразования ссылки для использования имени часто заключается лишь в заключении номера в кавычки.

marker.key (index)

Тип возвращения: MarkerKey.

Тип аргумента: *index* — это число.

Возвращает объект MarkerKey маркера с указанным индексом. Индекс ссылается на порядок маркера во времени композиции, а не на имя маркера.

Пример: это выражение возвращает время первого маркера композиции:

```
thisComp.marker.key (1).time
```

marker.key (name)

Тип возвращения: MarkerKey.

Тип аргумента: *name* — это строка.

Возвращает объект MarkerKey маркера с указанным именем. Значение *name* — это имя маркера, вводимое в поле комментариев диалогового окна маркера.

Пример: `marker.key ("1")`.

Именем маркера композиции по умолчанию является номер. Если несколько маркеров в композиции имеют одинаковое имя, этот метод **Возвращает** первый маркер во времени (во времени композиции). Значением ключа маркера является строка, а не число.

Пример: это выражение возвращает время маркера композиции с именем «0»:

```
thisComp.marker.key ("0").time
```

marker.nearestKey (t)

Тип возвращения: MarkerKey.

Тип аргумента: *t* — это число.

Возвращает маркер, по времени находящийся ближе всего к *t*.

Пример: это выражение возвращает время маркера композиции, находящееся рядом со временем равным 1 секунде:

`thisComp.marker.nearestKey (1).time`

Это выражение возвращает время маркера композиции, ближайшее к текущему времени:

`thisComp.marker.nearestKey (time).time`

marker.numKeys

Тип возвращения: число.

Возвращает общее количество маркеров композиции в композиции.

numLayers

Тип возвращения: число.

Возвращает количество слоев в композиции.

activeCamera

Тип возвращения: камера.

Возвращает объект камеры для камеры, через которую выполняется рендеринг композиции в текущем кадре. Эта камера не обязательно является камерой, через которую осуществляется просмотр на панели «Композиция».

width

Тип возвращения: число.

Возвращает ширину композиции в пикселах.

height

Тип возвращения: число.

Возвращает высоту композиции в пикселах.

Примените следующее выражение к свойству слоя «Положение» для расположения слоя в центре кадра композиции:

`[thisComp.width/2, thisComp.height/2]`

duration

Тип возвращения: число.

Возвращает продолжительность композиции в секундах.

ntscDropFrame

Тип возвращения: логическое значение.

Возвращает значение «истина», если тайм-код имеет формат пропуска кадра. (After Effects CS5.5 и более поздние версии.)

displayStartTime

Тип возвращения: число.

Возвращает время начала композиции в секундах.

frameDuration

Тип возвращения: число.

Возвращает продолжительность кадра в секундах.

shutterAngle

Тип возвращения: число.

Возвращает значение угла затвора композиции в градусах.

shutterPhase

Тип возвращения: число.

Возвращает фазу затвора композиции в градусах.

bgColor

Тип возвращения: массив [4].

Возвращает цвет фона композиции.

pixelAspect

Тип возвращения: число.

Возвращает пропорции пиксела композиции.

name

Тип возвращения: строка.

Возвращает имя композиции.

Атрибуты и методы видеоряда.

Для использования элемента видеоряда с панели «Проект» в качестве объекта в выражении воспользуйтесь глобальным методом **footage**, как в **footage ("имя_файла")**. Также можно вызвать объект видеоряда с помощью атрибута **источника** на слое, источник которого является элементом видеоряда.

width

Тип возвращения: число.

Возвращает ширину элемента видеоряда в пикселах.

height

Тип возвращения: число.

Возвращает высоту элемента видеоряда в пикселах.

duration

Тип возвращения: число.

Возвращает продолжительность элемента видеоряда в секундах.

frameDuration

Тип возвращения: число.

Возвращает продолжительность кадра в элементе видеоряда в секундах.

ntscDropFrame

Тип возвращения: логическое значение.

Возвращает значение «истина», если тайм-код имеет формат пропуска кадра. (After Effects CS5.5 и более поздние версии.)

pixelAspect

Тип возвращения: число.

Возвращает пропорции пиксела элемента видеоряда.

name

Тип возвращения: строка.

Возвращает имя элемента видеоряда, указанное на панели «Проект».

Атрибуты и методы вложенных объектов слоя.

Примечание. Для After Effects CC и CS6 пункты «Меню языка выражения», «Подобъекты слоя», «Общие атрибуты слоя», «Свойства слоя», «3D-слой» и «Преобразования пространства слоя» расположены в подменю «Слой».

source

Тип возвращения: композиция или видеоряд.

Возвращает исходный объект композиции или видеоряда для слоя. Время по умолчанию корректируется по времени в источнике.

Пример: `source.layer (1).position`

sourceTime (t = time)

Тип возвращения: число.

Возвращает источник слоя, соответствующий времени **t**.

effect (name)

Тип возвращения: эффект.

Тип аргумента: *name* – это строка.

After Effects находит эффект по имени на панели «Элементы управления эффектами». Имя может быть именем по умолчанию или пользовательским именем. Если несколько эффектов имеют одинаковое имя, будет использоваться самый верхний эффект на панели «Элементы управления эффектами».

Пример: effect ("Fast Blur") ("Blurriness")

effect (index)

Тип возвращения: эффект.

Тип аргумента: *index* – это число.

After Effects находит эффект по индексу на панели «Элементы управления эффектами», начиная с первого верхнего эффекта.

mask (name)

Тип возвращения: маска.

Тип аргумента: *name* – это строка.

Имя может быть именем по умолчанию или пользовательским именем. Если несколько масок имеют одинаковое имя, будет использоваться первая (самая верхняя) маска.

Пример: mask ("Mask 1")

mask (index)

Тип возвращения: маска.

Тип аргумента: *index* – это число.

After Effects находит маску по индексу на панели «Таймлайн», начиная с первого верхнего эффекта.

Общие атрибуты и методы слоя.

Width

Тип возвращения: число.

Возвращает ширину слоя в пикселах. Она совпадает с **source.width**.

Height

Тип возвращения: число.

Возвращает высоту слоя в пикселах. Она совпадает с **source.height**.

Index

Тип возвращения: число.

Возвращает порядковый номер слоя в композиции.

parent

Тип возвращения: слой, свет или камера.

Возвращает родительский объект слоя, если таковой имеется.

Пример: position[0]+parent.width

hasParent

Возвращает значение «истина», если слой обладает родительским слоем, и значение «ложь», если такого слоя нет.

Используйте атрибут **hasParent** для определения наличия родительского слоя в слое. Можно использовать этот атрибут, даже если слой не содержит ни одного родительского слоя.

Пример: следующее выражение указывает, что слой, к которому применяется выражение, покачивается с учетом положения родительского слоя. Если слой не имеет родительского объекта, он покачивается в зависимости от собственного положения. Если слой содержит родительский слой, поведение слоя будет соответствующим образом меняться:

```
idx = index; if (hasParent) { idx = parent.index;} thisComp.layer (idx).position.wiggle (5,20)
```

inPoint

Тип возвращения: число.

Возвращает точку входа слоя в секундах.

Примечание. Как правило, значение **outPoint** больше значения **inPoint**. Однако, если слой обращен во времени, значение **inPoint** будет больше значения **outPoint**. Аналогично, значение **startTime** может быть больше значения **inPoint**.

outPoint

Тип возвращения: число.

Возвращает точку выхода слоя в секундах.

startTime

Тип возвращения: число.

Возвращает время начала слоя в секундах.

hasVideo

Тип возвращения: логическое значение.

Возвращает значение «истина», если слой обладает видео, и значение «ложь», если такого видео нет.

hasAudio

Тип возвращения: логическое значение.

Возвращает значение «истина», если слой обладает аудио, и значение «ложь», если такого аудио нет.

active

Тип возвращения: логическое значение.

Возвращает значение «истина», если переключатель «Видео» включен для слоя, а текущее время находится в диапазоне от точки входа слоя до точки выхода слоя. При других условиях значением будет «ложь».

enabled

Тип возвращения: логическое значение.

Возвращает значение «истина», если переключатель «Видео» включен для слоя. При других условиях значением будет «ложь».

audioActive

Возвращает значение «истина», если переключатель «Аудио» включен для слоя, а текущее время находится в диапазоне от точки входа слоя до точки выхода слоя. При других условиях значением будет «ложь».

sampleImage (point, radius = [.5, .5], postEffect=true, t=time)

Тип возвращения: массив [4].

Тип аргумента: *point* – это массив [2], *radius* – это массив [2], *postEffect* – это логическое значение, а *t* – это число.

Отбирает значения канала цвета и альфа-канала слоя и возвращает среднее взвешенное значение альфа-канала в пикселах в указанном расстоянии точки в виде массива: [красный, зеленый, синий, альфа]. Если *postEffect* имеет значение «истина», это значит, что был выполнен рендеринг выборочных значений слоя после масок и эффектов в этом слое, если же *postEffect* имеет значение «ложь», это значит, что был выполнен рендеринг выборочных значений слоя перед масками и эффектами. Входное значение *point* находится в пространстве слоя. Точка [0,0] соответствует центру верхнего левого пиксела в слое. Входное значение *radius* определяет горизонтальное и

вертикальное расстояние от центра выборки до краев прямоугольника выборки. Значение по умолчанию использует в качестве образца один пиксел.

Примечание. Параметр `postEffect` ссылается на эффекты, применяемые непосредственно к слою, а не на эффекты, применяемые косвенно, как это происходит с корректирующим слоем.

Примечание. Использование `sampleImage` в выражении больше не отключает параллельную обработку.

В этом примере в качестве образца используется прямоугольник шириной четыре пиксела и высотой три пиксела, центрированный вокруг точки в 100 пикселов в нижней правой части от верхнего левого края слоя:

```
thisComp.layer (1).sampleImage ([100, 100], [2, 1.5])
```

Атрибуты и методы свойств слоя.

При добавлении масок, эффектов, цвета или текста к слою After Effects добавляет новые свойства на панель «Таймлайн». Таких свойств очень много, поэтому, чтобы не описывать их все в этом документе, используйте инструмент «Лассо» для изучения синтаксиса обращения к этим свойствам в выражении.

anchorPoint

Тип возвращения: свойство [2 или 3].

Возвращает значение опорной точки слоя в системе координат слоя (пространство слоя).

position

Тип возвращения: свойство [2 или 3].

Возвращает значение положения слоя в пространстве слоя World, если у слоя нет родительского слоя. Если у слоя есть родительский слой, возвращается значение положения слоя в системе координат родительского слоя (в пространстве родительского слоя).

scale

Тип возвращения: свойство [2 или 3].

Возвращает значение масштаба слоя, выраженное в процентах.

rotation

Тип возвращения: свойство.

Возвращает значение поворота слоя в градусах. Для 3D-слоя выполняется возврат значения поворота по оси Z в градусах.

opacity

Тип возвращения: свойство.

Возвращает значение непрозрачности слоя, выраженное в процентах.

audioLevels

Тип возвращения: свойство [2].

Возвращает значение свойства слоя «Уровни аудио» в децибелах. Это значение является 2D-значением. Первое значение относится к левому аудиоканалу, а второе – к правому. Значение не является амплитудой аудиодорожки исходного материала. Напротив, это значение свойства «Уровни аудио», на которое могут влиять ключевые кадры.

timeRemap

Тип возвращения: свойство.

Возвращает значение свойства «Перераспределение времени» в секундах, если параметр «Перераспределение времени» активен.

marker.key (index)

Тип возвращения: MarkerKey.

Тип аргумента: *index* – это число.

Возвращает объект MarkerKey маркера слоя с указанным индексом.

marker.key (name)

Тип возвращения: MarkerKey.

Тип аргумента: *name* – это строка.

Возвращает объект MarkerKey маркера слоя с указанным именем. Значение *name* – это имя маркера, вводимое в поле комментариев диалогового окна маркера.

Пример

```
marker.key ("ch1").
```

Если несколько маркеров слоя имеют одинаковое имя, этот метод возвращает первый маркер во времени (во времени слоя). Значением ключа маркера является строка, а не число.

При использовании этого выражения в свойстве выполняется плавный переход значения свойства от 0 до 100 между двумя маркерами, обозначенными именем:

```
m1=marker.key ("Start").time; m2=marker.key ("End").time; linear (time, m1, m2, 0, 100);
```

marker.nearestKey (t)

Тип возвращения: MarkerKey.

Тип аргумента: *t* – это число.

Возвращает маркер слоя, расположенный во времени ближе всего к *t*.

Пример: это выражение возвращает время маркера слоя, находящееся рядом со временем равным 1 секунде:

```
marker.nearestKey (1).time
```

Это выражение возвращает время маркера слоя, ближайшее к текущему времени:

```
marker.nearestKey (time).time
```

marker.numKeys

Тип возвращения: число.

Возвращает количество маркеров в слое.

name

Возвращает общее количество маркеров в слое.

Тип возвращения: строка.

Возвращает имя слоя.

Атрибуты и методы 3D-слоя.

orientation

Тип возвращения: свойство [3].

Возвращает значение 3D-ориентации в градусах для 3D-слоя.

rotationX

Тип возвращения: свойство.

Возвращает значение поворота по оси X в градусах для 3D-слоя.

rotationY

Тип возвращения: свойство.

Возвращает значение поворота по оси Y в градусах для 3D-слоя.

rotationZ

Тип возвращения: свойство.

Возвращает значение поворота по оси Z в градусах для 3D-слоя.

lightTransmission

Тип возвращения: свойство.

Возвращает значение свойства «Передача света» для 3D-слоя.

castsShadows

Тип возвращения: свойство.

Возвращает значение 1,0, если слой отбрасывает тени.

acceptsShadows

Тип возвращения: свойство.

Возвращает значение 1,0, если слой принимает тени.

acceptsLights

Тип возвращения: свойство.

Возвращает значение 1,0, если слой принимает свет.

ambient

Тип возвращения: свойство.

Возвращает значение компонента освещения в процентах.

diffuse

Тип возвращения: свойство.

Возвращает значение компонента диффузии в процентах.

specular

Тип возвращения: свойство.

Возвращает значение компонента зеркального блеска в процентах.

shininess

Тип возвращения: свойство.

Возвращает значение компонента блеска в процентах.

metal

Тип возвращения: свойство.

Возвращает значение компонента металла в процентах.

Методы преобразования пространства слоя.

Пространство слоя преобразует методы для преобразования значения из одного пространства в другое, на**Пример**: преобразование пространства слоя в пространство слоя World. Методы «from» преобразуют значения из именованного пространства (композиция или слой World) в пространство слоя. Методы «to» преобразуют значения из пространства слоя в именованное пространство

(композиция или слой World). Каждый метод преобразования использует дополнительный аргумент для определения времени, в течение которого вычисляется преобразование. Однако почти всегда можно использовать текущее время (время по умолчанию).

Используйте методы преобразования «Vec» при преобразовании вектора направления, такого как разница между двумя значениями положения. Используйте обычные методы преобразования (невекторные) для преобразования точки,

Пример: положения. Пространство композиции и пространство слоя World одинаковы для 2D-слоев. Однако для 3D-слоев пространство композиции относится к активной камере, а пространство слоя World не зависит от камеры.

toComp (point, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *point* – это массив [2 или 3], а *t* – это число. **Преобразует** точку из пространства слоя в пространство композиции.

fromComp (point, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *point* – это массив [2 или 3], а *t* – это число.

Преобразует точку из пространства композиции в пространство слоя. Результирующая точка в 3D-слое может обладать ненулевым значением, даже если она находится в пространстве слоя.

Пример: `fromComp (thisComp.layer (2).position)`

toWorld (point, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *point* — это массив [2 или 3], а *t* — это число.

Преобразует точку из пространства слоя в видонезависимое пространство слоя World.

Пример: `toWorld.effect ("Bulge") ("Bulge Center")`

fromWorld (point, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *point* – это массив [2 или 3], а *t* – это число.

Преобразует точку из пространства слоя World в пространство слоя.

Пример: `fromWorld (thisComp.layer (2).position)`

toCompVec (vec, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *vec* – это массив [2 или 3], а *t* – это число.

Преобразует вектор из пространства слоя в пространство композиции.

Пример: `toCompVec ([1,0])`

fromCompVec (vec, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *vec* – это массив [2 или 3], а *t* – это число.

Преобразует вектор из пространства композиции в пространство слоя.

Пример

(2D-слой): `dir=sub (position, thisComp.layer (2).position); fromCompVec (dir)`

toWorldVec (vec, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *vec* – это массив [2 или 3], а *t* – это число.

Преобразует вектор из пространства слоя в пространство слоя World.

Пример:

`p1 = effect ("Eye Bulge 1") ("Bulge Center"); p2 = effect ("Eye Bulge 2") ("Bulge Center"); toWorld (sub (p1, p2))`

fromWorldVec (vec, t=time)

Тип возвращения: массив [2 или 3].

Тип аргумента: *vec* – это массив [2 или 3], а *t* – это число.

Преобразует вектор из пространства слоя World в пространство слоя.

Пример: `fromWorld (thisComp.layer (2).position)`

fromCompToSurface (point, t=time)

Тип возвращения: массив [2].

Тип аргумента: *point* – это массив [2 или 3], а *t* – это число.

Проектирует точку, расположенную в пространстве композиции, в точку на поверхности слоя (нулевое значение по оси Z) в месте, где она отображается, при просмотре из активной камеры. Этот метод удобен для настройки точек управления эффектом. Следует использовать только с 3D-слоями.

Атрибуты и методы камеры.

Объекты камеры обладают теми же атрибутами и методами, что и объекты слоя, за исключением **source, effect, mask, width, height, anchorPoint, scale, opacity, audioLevels, timeRemap** и всех свойств материала.

pointOfInterest

Тип возвращения: свойство.

Возвращает значения точки обзора камеры в пространстве слоя World.

Тип возвращения: свойство.

zoom

Возвращает значения масштаба камеры в пикселах.

Ниже представлено выражение для свойства слоя «Масштаб», сохраняющее относительный размер слоя в кадре при изменении положения по оси Z (глубины) слоя или значения «Масштаб» камеры:

```
cam = thisComp.activeCamera; distance = length (sub (position, cam.position)); scale * distance/cam.zoom;
```

depthOfField

Тип возвращения: свойство.

Возвращает значение 1, если свойство камеры «Глубина поля» активно, или значение 0, если свойство камеры «Глубина поля» отключено.

focusDistance

Тип возвращения: свойство.

Возвращает значение фокусного расстояния камеры в пикселах.

aperture

Тип возвращения: свойство.

Возвращает значение диафрагмы камеры в пикселах.

blurLevel

Тип возвращения: свойство.

Возвращает значение уровня размытия камеры в процентах.

active

Тип возвращения: логическое значение.

Возвращает значение «истина», если камера является активной для композиции в текущем времени: переключатель «Видео» для слоя камеры включен, текущее время находится в диапазоне от точки входа слоя камеры до точки выхода слоя камера и этот слой камеры является первым (самым верхним) в списке на панели «Таймлайн». В противном случае возвращает значение «ложь».

Атрибуты и методы света.

Объекты света обладают теми же атрибутами и методами, что и объекты слоя, за исключением `source`, `effect`, `mask`, `width`, `height`, `anchorPoint`, `scale`, `opacity`, `audioLevels`, `timeRemap` и всех свойств материала.

`pointOfInterest`

Тип возвращения: свойство [3].

Возвращает значения точки обзора для света в пространстве слоя World.

`intensity`

Тип возвращения: свойство.

Возвращает значения интенсивности света в процентах.

`color`

Тип возвращения: свойство [4].

Возвращает значение цвета света.

`coneAngle`

Тип возвращения: свойство.

Возвращает угол конуса света в градусах.

`coneFeather`

Тип возвращения: свойство.

Возвращает значение растушевки конуса света в процентах.

`shadowDarkness`

Тип возвращения: свойство.

Возвращает значение темноты тени света в процентах.

`shadowDiffusion`

Тип возвращения: свойство.

Возвращает значение диффузии тени света в пикселах.

Атрибуты и методы эффектов.

`active`

Тип возвращения: логическое значение.

Возвращает значение «истина», если эффект включен (выбран переключатель «Эффект»)

param (name)

Тип возвращения: свойство.

Тип аргумента: *name* – это строка.

Возвращает свойство в пределах эффекта. Точки управления эффектом всегда расположены в пространстве слоя.

Пример:

`effect ("Bulge").param ("Bulge Height").`

param (index)

Тип возвращения: свойство.

Тип аргумента: *index* – это число.

Возвращает свойство в пределах эффекта. Точки управления эффектом всегда расположены в пространстве слоя.

Пример: `effect ("Bulge").param (4)` возвращает свойство «Высота купола».

Атрибуты и методы маски.

Примечание. Свойства «Пути маски» можно связывать с другими свойствами пути (контуры в слое-фигуре и мазках кисти), но свойства не подходят для прямого числового управления с помощью выражений.

maskOpacity

Тип возвращения: свойство.

Возвращает значение непрозрачности маски в процентах.

maskFeather

Тип возвращения: свойство.

Возвращает значение растушевки маски в пикселах.

maskExpansion

Тип возвращения: свойство.

Возвращает значение расширения маски в пикселах.

invert

Тип возвращения: логическое значение.

Возвращает значение «истина», если маска обращена, и значение «ложь», если маска не обращена.

Атрибуты и методы свойства.

value

Тип возвращения: число, массив или строка. **Возвращает** значение свойства в текущем времени.

valueAtTime (t)

Тип возвращения: число или массив. **Тип аргумента:** t – это число.

Возвращает значение свойства в указанном времени в секундах.

Пример: чтобы значение свойства каждого кадра выбиралось произвольно из набора из четырех значений, задайте четырем параметрам, используемым в качестве ключевых кадров, значения 0, 1, 2 и 3 секунды, а затем примените к свойству следующее выражение:

```
valueAtTime (random (4))
```

velocity

Тип возвращения: число или массив.

Возвращает временное значение скорости в текущем времени. Для пространственных свойств.

Пример: свойства «Положение» метод возвращает значение вектора касательной. Результат будет обладать тем же размером, что и свойство.

velocityAtTime (t)

Тип возвращения: число или массив.

speed

Тип аргумента: t – это число.

Возвращает временное значение скорости в указанном времени.

Тип возвращения: число.

Возвращает одномерное, положительное значение скорости, равное скорости, при которой свойство изменяется во времени по умолчанию. Этот элемент можно использовать только для пространственных свойств.

speedAtTime (t)

Тип возвращения: число. **Тип аргумента:** t – это число.

Возвращает пространственное значение скорости в заданном времени.

wiggle (freq, amp, octaves=1, amp_mult=.5, t=time)

Тип возвращения: число или массив.

Тип аргумента: *freq*, *amp*, *octaves*, *amp_mult* и *t* – это числа. Произвольно применяет эффект покачивания к значению свойства. Значение *freq* – это частота в покачиваниях в секунду.

Значение *amp* – это амплитуда в единицах измерения свойства, к которому оно применяется.

Значение *octaves* – это количество октав шума, которые требуется сложить. Это значение управляет объемом детализации в покачивании. Сделайте это значение больше значения по умолчанию (1), чтобы добавить более высокие частоты, или меньше, чтобы добавить амплитудную гармонику в покачивание.

Значение *amp_mult* – это число, на которое умножается значение *amp* для каждой октавы. Это значение управляет скоростью спада гармоник. Значение по умолчанию равно 0,5. приблизите его к 1, чтобы добавить гармонику при той же амплитуде, что и базовая частота, или приблизите его к 0, чтобы уменьшить детализацию.

Значение *t* – это базовое начальное время. Этому значению по умолчанию присваивается текущее время. Используйте этот параметр, чтобы выходные данные являлись покачиванием значения свойства, полученного в разное время.

Пример: `position.wiggle (5,20,3,.5)` приводит к формированию около пяти покачиваний в секунду со средним размером в 20 пикселей. В дополнение к главному покачиванию формируются еще два уровня подробных покачиваний с частотой от 10 до 20 покачиваний в секунду и размером от 10 до 5 пикселей соответственно.

Следующее выражение с двухмерным свойством, таким как «Масштаб», иницирует покачивание обоих измерений в одной и той же степени:

```
v = wiggle (5, 10); [v[0], v[0]]
```

Следующее выражение с двухмерным свойством иницирует покачивание только вдоль оси Y:

```
freq = 3; amp = 50; w = wiggle (freq,amp); [value[0],w[1]];
```

temporalWiggle (freq, amp, octaves=1, amp_mult=.5, t=time)

Тип возвращения: число или массив.

Тип аргумента: *freq*, *amp*, *octaves*, *amp_mult* и *t* – это числа.

Отбирает свойство во времени покачивания. Значение *freq* – частота покачиваний в секунду, *amp* – амплитуда в единицах измерения свойства, к которому применяется значение, *octaves* – количество октав шума, которые требуется сложить, *amp_mult* – количество, на которое умножается значение *amp* для каждой октавы и *t* – это базовое время начала. Чтобы эта функция работала, необходимо анимировать свойство, отбираемое функцией, так как функция изменяет только время выборки, а не само значение.

Пример: `scale.temporalWiggle (5,.2)`

Smooth (width=.2, samples=5, t=time)

Тип возвращения: число или массив.

Тип аргумента: *width*, *samples* и *t* – это числа.

Сглаживает значения свойства с течением времени, преобразуя большие, короткие отклонения в значении в меньшие, более равномерно распределенные отклонения. Это сглаживание выполняется путем применения усредняющего фильтра к значению свойства в указанном времени. Значение *width* – это диапазон времени (в секундах), в пределах которого усредняется фильтр. Значение *samples* – это число дискретных образцов, равномерно распределенных во времени. Используйте высокое значение для обеспечения большей степени сглаживания (приводит к уменьшению производительности). Как правило, значение *samples* должно быть нечетным, чтобы значение в текущем времени было добавлено в среднюю величину.

Пример: position.smooth (.1, 5)

loopIn (type="cycle", numKeyframes=0)

Тип возвращения: число или массив.

Повторяет сегмент времени, измеряемый от первого ключевого кадра в слое до точки выхода слоя. Цикл воспроизводится с точки входа слоя. Значение *numKeyframes* определяет, какой сегмент воспроизводится циклично: повторяющийся сегмент является частью слоя от первого ключевого кадра до *numKeyframes* + еще один ключевой кадр.

Пример: loopIn ("cycle", 3) повторяет сегмент, ограниченный первым и четвертым ключевыми кадрами. Значение по умолчанию 0 означает, что все ключевые кадры повторяются циклично.

Методы зацикливания ключевых кадров можно использовать для настройки повторного воспроизведения набора ключевых кадров. Эти методы можно применять к большинству свойств. К исключениям относятся свойства, которые не могут быть выражены простыми числовыми значениями на панели «Таймлайн», такие как свойство «Исходный текст», свойства формы пути и свойство «Гистограмма» для эффекта «Уровни». Слишком большие ключевые кадры или значения продолжительности отсекаются до максимально допустимых значений.

Слишком низкие значения приводят к непрерывному циклическому повторению.

тип цикла - результат

cycle- (По умолчанию) зацикливает указанный сегмент.

Pingpong- Зацикливает определенный сегмент с попеременным чередованием направления.

Offset- Зацикливает определенный сегмент, но смещает каждый цикл на разницу в значениях свойства в начале и конце сегмента, умноженную на количество зацикливаний сегмента.

Continue - Не зацикливает указанный сегмент, но продолжает анимировать свойство на основании скорости в первом или последнем ключевом кадре.

Пример: если последний ключевой кадр свойства слоя «Масштаб» равен 100 %, слой будет продолжать масштабироваться от 100 % до точки выхода вместо возврата обратно к началу цикла в 35

точке выхода. Этот тип не принимает аргумент *keyframes* или *duration*.

loopOut (type="cycle", numKeyframes=0)

Тип возвращения: число или массив.

Повторяет сегмент времени, измеряемый от последнего ключевого кадра в слое до точки входа слоя. Цикл воспроизводится до точки выхода слоя. Заданное число ключевых кадров определяет циклически повторяющийся сегмент. Значение *numKeyframes* используется для определения количества сегментов ключевых кадров для заикливания. Указанный диапазон измеряется в обратном направлении от последнего ключевого кадра.

Пример: значение **loopOut ("cycle", 1)** повторяет сегмент, ограниченный последним и предпоследним ключевыми кадрами. Значение по умолчанию 0 означает, что все ключевые кадры повторяются циклично. Дополнительную информацию см. **loopIn**.

loopInDuration (type="cycle", duration=0)

Тип возвращения: число или массив.

Повторяет сегмент времени, измеряемый от первого ключевого кадра в слое до точки выхода слоя. Цикл воспроизводится с точки входа слоя. Указанная длительность определяет циклически повторяемый сегмент. Значение *duration* используется для указания числа секунд композиции в повторяющемся сегменте. Указанный диапазон измеряется от первого ключевого кадра.

Пример: значение **loopInDuration ("cycle",1)** повторяет первый сегмент всей анимации. Значение по умолчанию 0 означает, что повторяющийся сегмент начинается в точке выхода слоя. Дополнительную информацию см. **loopIn**.

loopOutDuration (type="cycle", duration=0)

Тип возвращения: число или массив.

Повторяет сегмент времени, измеряемый от последнего ключевого кадра в слое до точки входа слоя. Цикл воспроизводится до точки выхода слоя. Указанная длительность определяет циклически повторяемый сегмент. Значение *duration* используется для указания числа секунд композиции в повторяющемся сегменте. Указанный диапазон измеряется в обратном направлении от последнего ключевого кадра.

Пример: значение **loopOutDuration ("cycle", 1)** повторяет последний сегмент всей анимации. Значение по умолчанию 0 означает, что повторяющийся сегмент начинается в точке входа слоя.

Дополнительную информацию см. **loopIn**.

key (index)

Тип возвращения: Key или MarkerKey.

Тип аргумента: *index* – это число.

Возвращает объект Key или MarkerKey по числу. **Пример:** значение **key (1)** возвращает первый

ключевой кадр.

key (markerName)

Тип возвращения: MarkerKey.

Тип аргумента: *markerName* – это строка.

Возвращает объект MarkerKey с этим именем. Этот аргумент следует использовать только со свойствами маркера.

nearestKey (t)

Тип возвращения: Key или MarkerKey.

Возвращает объект Key или MarkerKey, расположенный ближе всего к указанному времени.

numKeys

Тип возвращения: число.

Возвращает количество ключевых кадров в свойстве. **Возвращает** количество маркеров в свойстве маркера.

Примечание. При использовании команды «Разделить размеры» для разделения размеров свойства «Положение» на отдельные компоненты количество ключевых кадров изменяется, что также приводит к изменению возвращаемого этим методом значения.

propertyGroup (countUp = 1)

Тип возвращения: группа.

Возвращает группу свойств, относящихся к свойству, по которому написано выражение. **Пример:** при добавлении выражения **propertyGroup (1)** к свойству мазка кисти «Поворот» целью выражения становится группа свойств «Преобразование», содержащая свойство

«Поворот». А при добавлении выражения **propertyGroup (2)** целью выражения становится группа свойств «Кисть». Этот метод позволяет установить связи в иерархии свойств, независимые от имени. Этот метод особенно удобен при дублировании свойств, содержащих выражения.

Метод **numProperties** для **propertyGroup** **Возвращает** количество свойств в группе свойств.

Следующее выражение возвращает количество свойств в группе, содержащей свойство, по которому написано выражение: **thisProperty.propertyGroup (1).numProperties**

propertyIndex

Тип возвращения: число.

Возвращает индекс свойства, относящийся к другим свойствам в соответствующей группе свойств, включая группы свойств в масках, эффектах, аниматорах текста, инструментах выбора, формах, средствах и точках отслеживания.

name

Тип возвращения: строка.

Возвращает имя свойства или группы свойств.

Пример: анимация методом `propertyGroup` с помощью атрибута `propertyIndex`

Значения `propertyGroup`, относящиеся к свойству мазка кисти «Положение» A. `propertyGroup (4)` B. `propertyGroup (3)` C. `propertyGroup (2)` D. `propertyGroup (1)` E. Значение положения `propertyIndex` равно 2. Значение вращения `propertyIndex` равно 4.

В следующем примере использования метода `propertyGroup` целью каждого мазка кисти является группа свойств «Кисть», так как эта группа расположена двумя группами свойств выше свойства «Поворот». Затем атрибут `propertyIndex` в каждом мазке кисти возвращает уникальное значение для каждого мазка кисти. Затем итоговое значение умножается на время и на число 200, а потом применяется к любому значению поворота при различном повороте каждого мазка кисти, создавая закрученные мазки:

```
propertyGroup (2).propertyIndex * time * 200
```

Атрибуты и методы ключа.**Value**

Тип возвращения: число или массив.

Возвращает значение ключевого кадра.

Time

Тип возвращения: число.

Возвращает время ключевого кадра.

index

Тип возвращения: число.

Возвращает индекс ключевого кадра.

При переходе к объекту `Key` из него можно получить свойства времени, индекса и значения.

Пример: следующее выражение возвращает значение третьего ключевого кадра «Положение»: `position.key (3).value`.

Следующее выражение, написанное по свойству «Непрозрачность» с ключевыми кадрами, игнорирует значения ключевых кадров и использует только положение ключевых кадров о времени для определения точки вспышки:

```
d = Math.abs (time — nearestKey (time).time); easeOut (d, 0, .1, 100, 0)
```

Атрибуты MarkerKey.

С помощью одних и тех же методов можно получать доступ к значениям маркеров композиции и слоя. Доступ к маркерам слоя осуществляется через объект **thisLayer.marker**, а к маркерам композиции — через объект **thisComp.marker**.

Работа выражений осуществляется за счет того, что маркеры представляют собой особый тип объекта Key, позволяющий использовать такие методы, как **nearestKey (time)** для получения доступа к маркерам; маркеры также включают атрибуты **time** и **index**. Атрибут **index** не является номером (именем) маркера. Это порядковый номер ключевого кадра, представляющий собой порядок следования маркера на линейке времени.

Выражения имеют доступ ко всем значениям маркера, которые можно задать в диалоговом окне «Маркер композиции» или «Маркер слоя». Следующее выражение, написанное по свойству «Исходный текст» текстового слоя, позволяет отобразить время, продолжительность, индекс, комментарий (имя), главу, URL-адрес, цель кадра и имя точки разметки маркера слоя, расположенного ближе всего к текущему времени, а также сообщает предназначен ли маркер для точки разметки события:

```
m = thisLayer.marker.nearestKey (time);
s = "time:" + timeToCurrentFormat (m.time) + "\r" + "duration: " + m.duration + "\r" +
"key index: " + m.index + "\r" + "comment:" + m.comment + "\r" + "chapter:" + m.chapter + "\r" + "URL:" +
m.url + "\r" +
"frame target: " + m.frameTarget + "\r" + "cue point name: " + m.cuePointName + "\r" + "Event cue
point? " + m.eventCuePoint + "\r"; for(param in m.parameters){
s += "parameter: " + param + " value: " + m.parameters[param] + "\r";
}
s
```

Поскольку метаданные XMP в элементе видеоряда можно преобразовать в маркеры слоя для основанного на этом элементе слоя, выражения могут взаимодействовать с метаданными XMP.

duration

Тип возвращения: число. Продолжительность маркера в секундах.

comment

Тип возвращения: строка.

Содержимое поля «Комментарий» в диалоговом окне маркера.

chapter

Тип возвращения: строка.

Содержимое поля «Глава» в диалоговом окне маркера.

url

Тип возвращения: строка.

Содержимое поля «URL-адрес» в диалоговом окне маркера.

frameTarget

Тип возвращения: строка.

Содержимое поля «Цель кадра» в диалоговом окне маркера.

eventCuePoint

Тип возвращения: логическое значение.

Установка типа точки разметки в диалоговом окне маркера. Значение «истина» для свойства «Событие»; значение «ложь» для свойства «Навигация».

cuePointName

Тип возвращения: строка.

Содержимое поля точки разметки «Имя» в диалоговом окне маркера.

parameters

Тип возвращения: ассоциативный массив строковых значений.

Содержимое полей «Имя параметра» и «Значение параметра» в диалоговом окне маркера.

Пример: если имеется параметр с именем «цвет фона», для доступа к его значению в ближайшем маркере можно использовать следующее выражение:

```
thisComp.marker.nearestKey (time).parameters["background color"]
```

Доступ выражений к точкам контуров фигур, масок и штрихов кисти.

Выражения можно использовать для считывания и записи координат x и y точек контуров (или вершин) следующих объектов:

- слои-маски;
- фигуры Безье;
- Штрихи кисти на панели «Рисование» и эффекты «Кисть для ротоскопии» и «Уточнить края».

Методы выражений:

Метод контура `points ()`: `{pathProperty}.points (t = time)`

Получение координат (x, y) всех точек контура. Координаты точек контура маски слоя указываются относительно начала координат слоя в левом верхнем углу. Координаты точек контура фигур Безье указываются относительно опорной точки группы фигур контура, **Пример:** Преобразовать: Фигура

1 > Опорная точка. Координаты точек контура штрихов кистью задаются относительно начала штриха. Координаты первой точки: [0,0]. При желании можно определить время выборки на контуре. При дублировании контура этот метод можно передать методу **createPath ()** для параметра точек.

Параметры: t - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: time (текущее время).

Возвращает: массив массивов пар чисел, округленных до четвертого знака после запятой.

Метод **inTangents ()** контура: **{pathProperty}.inTangents (t = time)**

Получает координаты x, y дескриптора входящей касательной для всех точек на контуре. Значения координат касательной представляют собой смещения относительно координат родительской точки. То есть значение [0,0] не приводит к созданию кривизны на входящей касательной. Этот метод можно передать методу **createPath ()** для параметра **inTangents** дублировании контура. При желании можно определить время выборки на контуре.

Параметры: t - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: time (текущее время).

Возвращает: массив массивов пар чисел, округленных до четвертого знака после запятой.

Метод **outTangents ()** контура: **{pathProperty}.outTangents (t = time)**

Получает координаты x, y дескриптора исходящей касательной для всех точек на контуре. Значения координат касательной представляют собой смещение относительно координат родительской точки. Значение [0,0] не приводит к созданию кривизны на исходящей касательной. Этот метод можно передать методу **createPath()** для параметра **outTangents** при дублировании контура. При желании можно определить время выборки на контуре.

Параметры: t - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: time (текущее время).

Возвращает: массив массивов пар чисел, округленных до четвертого знака после запятой.

Метод **isClosed ()** контура: **{pathProperty}.isClosed ()**

Определяет, является ли контур незамкнутым или замкнутым.

Возвращает значение true, если контур замкнут, или false, если контур не замкнут. Этот метод можно передать методу **createPath ()** для параметра **is_closed** при дублировании контура.

Параметры: Нет

Возвращает: логическое значение

Метод **pointOnPath ()** контура **{pathProperty}.pointOnPath (percentage = 0,5, t = time)**

Определяет координаты x и y произвольной точки вдоль контура. Координаты точки выражены в виде процента от длины дуги контура. 0 % — первая точка, а 100 % — последняя точка. Если контур

замкнут, 0 % и 100 % возвращают одинаковые координаты. Процент от длины дуги используется для обеспечения постоянной скорости вдоль контура. За исключением значений 0

% и 100 % процентные значения не обязательно коррелируют с точками Безье на контуре. Для контура с тремя точками вторая точка не обязательно соответствует 50 %. Это также означает, что для незамкнутого и замкнутого контуров с идентичными точками одинаковое процентное значение вдоль незамкнутого и замкнутого контуров не возвращает одинаковые координаты из-за дополнительной длины замкнутого контура. При желании можно определить время выборки на контуре.

Параметры:

percentage - Число между 0 и 1, дополнительное. Процент от длины дуги контура, для которого делается выборка. Значения меньше 0 или больше 1 обрезаются. Значение по умолчанию: 0,5.

t - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: time (текущее время).

Возвращает: массив пар чисел.

Метод **tangentOnPath()** контура: **{pathProperty}.tangentOnPath (percentage = 0,5, t = time)**

Получает вычисленные координаты x, y дескриптора исходящей касательной для произвольной точки вдоль контура. Значения координат касательной представляют собой смещение относительно координат родительской точки. Значение [0,0] не ведет к созданию кривизны на исходящей касательной. Дескриптору входящей касательной соответствует инверсия этого значения (координаты x, y, умноженные на -1).

Родительская точка касательной выражена в виде процента от длины дуги контура. Координаты, возвращаемые методом **tangentOnPath ()**, вычисляются на основе родительской точки. Они будут отличаться от тех, которые возвращаются методом **outTangents ()**, если процент от длины дуги также соответствует точке, определенной пользователем. Линейное расстояние между родительской точкой и точкой, координаты которой возвращаются методом **tangentOnPath ()**, всегда равно 1. Возвращенные координаты можно умножить на некоторый множитель, чтобы создать более длинную касательную,

Пример:

`(myPath.tangentOnPath () * 100)`. При желании можно определить время выборки на контуре.

Параметры: Percentage - Число между 0 и 1, дополнительное. Процент от длины дуги контура, для которого делается выборка. Значения меньше 0 или больше 1 обрезаются. Значение по умолчанию: 0,5.

T - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: time (текущее время).

Возвращает: массив пар чисел.

Метод `normalOnPath ()` контура `{pathProperty}.normalOnPath (percentage = 0,5, t = time)`

Получает вычисленные координаты x , y нормали для произвольной точки вдоль контура. Значения координат нормалей представляют собой смещение относительно координат родительской точки. Значение $[0,0]$ соответствует родительской точке. Родительская точка нормали выражена в виде процента от длины дуги контура. Подробнее об использовании процента от длины дуги см. в описании метода `pointOnPath ()`. Координаты, возвращаемые методом `normalOnPath ()`, вычисляются на основе своей родительской точки. Линейное расстояние между родительской точкой и точкой, координаты которой возвращаются методом `normalOnPath ()`, всегда равно 1. Возвращенные координаты можно умножить на некоторый множитель, чтобы создать более длинную нормаль,

Пример:

`(myPath.normalOnPath () * 100)`. При желании можно определить время выборки на контуре.

Параметры:

`Percentage` - Число между 0 и 1, дополнительное. Процент от длины дуги контура, для которого делается выборка. Значения меньше 0 или больше 1 обрезаются. Значение по умолчанию: 0,5.

`t` - число, дополнительно. Время композиции (в секундах), для которого делается выборка контура. Значение по умолчанию: `time` (текущее время).

Возвращает: массив пар чисел.

Метод `createPath ()` контура `{pathProperty}.createPath (points = [[0,0], [100,0], [100,100], [0,100]], inTangents = [], outTangents = [], is_closed = true)`

Создает контурный объект из набора точек и касательных. Точки определены массивом массивов пар чисел, представляющих координаты x , y точек. Длина массива должна быть не меньше 1.

Дескрипторы входящих и исходящих касательных точек определяются массивом массивов пар чисел, представляющих их координаты смещения x , y . Длина массивов касательных должна быть в точности равна значению параметра `points`. Значения координат касательной представляют собой смещение относительно координат родительской точки. Значение $[0,0]$ не ведет к созданию кривизны на входящей касательной. Методы `points ()`, `inTangents ()`, `outTangents ()` и `isClosed ()` контура можно передавать в параметрах `points`, `inTangents`, `outTangents` и `is_closed` для дублирования контура. Точки и касательные одного контура можно передавать методу `createPath ()` с изменениями, чтобы получить другой результат.

Пример: следующее выражение удаляет кривые из Маски 1, не передавая параметры `inTangents` или `outTangents`:

```
myMask = mask ("Маска 1").path; myMask.createPath (myMask.points ());
```

В следующем примере передаются точки и касательные Маски 1 с преобразованием ее в незамкнутый контур, задав параметру `is_closed` значение `false`:

```
myMask = mask ("Маска 1").path; myMask.createPath (myMask.points (), myMask.inTangents (), myMask.outTangents (), false);
```

`points`

Массив длиной 1 или больше, который содержит массивы пар чисел, представляющие координаты [x, y] точек контура. Не обязателен, только если параметры не передаются (т. е. `createPath()`).
Значение по умолчанию: `[[0,0], [100,0], [100,100], [0,100]]`.

`is_closed`

Логический, необязательный. Определяет, замкнута ли маска. В случае значения `true` последняя точка будет соединена с первой точкой. Значение по умолчанию: `true`.

`inTangents`

Массив, который содержит массивы пар чисел, представляющих координаты смещения [x, y] дескрипторов исходящих касательных в точках контура. Не обязателен, только если параметры не передаются (т. е. `createPath ()`).

Длина массива должна быть равна значению параметра `points` или можно передать пустой массив (`[]`), что означает, что у всех касательных одинаковая длина (равна значению `points`) и исходная точка (координаты `[0,0]`).

Значение по умолчанию: пустой массив.

`outTangents`

Массив, который содержит массивы пар чисел, представляющих координаты смещения [x, y] дескрипторов входящих касательных в точках контура. Не обязателен, только если параметры не передаются (т. е. `createPath ()`). Длина массива должна быть равна значению параметра `points` или можно передать пустой массив (`[]`), что означает, что у всех касательных одинаковая длина (равна значению `points`) и исходная точка (координаты `[0,0]`). Значение по умолчанию: пустой массив.

Возвращает: контурный объект.

Пример 1

В данном примере реализован вывод в строку списка координат точек и касательных контура 1 фигуры 1 в слое-фигуре 1 слоя в момент `time=0`. Примените этот результат к исходному текстовому свойству текстового слоя для считанных координат и входящих, и исходящих касательных фигуры.

```
pointsList = ""; sampleTime = 0;
myShape = thisComp.layer ("Слой-фигура 1").content ("Фигура 1").content ("Контур 1").path;
for (i = 0; i < myShape.points (sampleTime).length; i++) { pointsList += "с: " + myShape.points
(sampleTime)[i].toString () + "
i: " + myShape.inTangents (sampleTime)[i].toString () + " о: " + myShape.outTangents
(sampleTime)[i].toString () + "\n";
}
pointsList;
```

Пример 2

В данном примере координаты первой вершины Маски 1 на Темно-серой заливке 1 считываются и преобразуются в координаты композиции. Примените этот результат к точкам управления 2D-эффектами, такими как «Надпись» и «Эффект систем частиц II СС», чтобы выполнить трассировку эффекта или отследить первую точку анимированной маски. Создайте дубликат эффект и измените значение индекса точек контура ([0]) для трассировки или отслеживания других точек маски.

```
myLayer = thisComp.layer ("Темно-серая заливка 1");
myLayer.toComp (myLayer.mask ("Маска 1").maskPath.points ()[0]);
```

Анимация, управляемая данными.**Атрибут видеоряда sourceText {footageItem}.sourceText**

Возвращает содержимое файла .JSON в виде строки. С помощью метода eval () эту строку можно преобразовать в массив объектов sourceData, идентичный результатам атрибута sourceData, из которого к отдельным потокам данных можно обращаться как к иерархическим атрибутам данных.

Пример:

```
var myData = eval (footage ("sample.json").sourceText); myData.sampleValue;
```

Тип: строка, содержимое файла .JSON; только для чтения.

Атрибут видеоряда sourceData {footageItem}.sourceData

Возвращает данные файла .JSON в виде массива объектов sourceData. Структура файла .JSON определит размер и сложность массива. К отдельным потокам данных можно обращаться как к иерархическим атрибутам данных.

Пример: если имеется поток данных с именем Color, следующий код вернет значение Color из первого объекта данных: footage ("sample.json").sourceData[0].Color

Стандартный вариант использования состоит в назначении атрибута sourceData файла .JSON переменной с последующим обращением к нужному потоку данных.

Пример:

```
var myData = footage ("sample.json").sourceData;
myData[0].Color;
```

Тип: Массив объектов sourceData; только для чтения.

Метод видеоряда dataValue () {footageItem}.dataValue (dataPath)

Возвращает значение указанного статического или динамического потока данных из файла .mgJSON. Принимает одно значение массива для определение пути в иерархии к нужному потоку данных.

Пример:

`footage ("sample.mgjson").dataValue ([0])` возвращает данные первого потомка.

`footage ("sample.mgjson").dataValue ([1][0])` возвращает данные первого потомка из второй группы.

Параметры: `dataPath` - Массив, обязательный. Путь в иерархии к статическому или динамическому потоку данных.

Возвращает: Значение потока данных.

Метод видеоряда `dataKeyCount () {footageItem}.dataKeyCount (dataPath)`

Возвращает количество выборок в указанном динамическом потоке данных из файла `.mgJSON`. Принимает одно значение массива, чтобы определить путь в иерархии к нужному динамическому потоку данных.

Пример:

`footage ("sample.mgjson").dataKeyCount ([0])` возвращает количество выборок для первого потомка.

`footage ("sample.mgjson").dataKeyCount ([1][0])` возвращает количество выборок для второй группы

Параметры: `dataPath` - Массив, обязательный. Путь в иерархии к статическому или динамическому потоку данных.

Возвращает: Количество выборок в динамическом потоке данных.

Метод видеоряда `dataKey Times () {footageItem}.dataKeyTimes (dataPath, t0 = startTime, t1=endTime)`

Возвращает время в секундах для выборок указанного динамического потока данных из файла `.mgJSON`. Также можно указать промежуток времени, из которого требуется вернуть выборки. По умолчанию возвращается время для всех выборок между `startTime` и `endTime` динамического потока данных в соответствии со свойством `samplesTemporalExtent` заданным в файле `.mgJSON`. Принимает одно значение массива, чтобы определить путь в иерархии к нужному динамическому потоку данных.

Следующий **Пример** возвращает время выборок между 1 секундой и 3 секундами для первого потомка:

`footage ("sample.mgjson").dataKeyTimes ([0], 1, 3)`

Параметры:

`dataPath` - Массив, обязательный. Путь в иерархии к динамическому потоку данных.

`t0` - число, дополнительно. Время начала (в секундах) промежутка времени, из которого требуется вернуть выборки. По умолчанию задано значение `startTime`.

`t1` - число, дополнительно. Время окончания (в секундах) промежутка времени, из которого требуется вернуть выборки. По умолчанию задано значение `endTime`.

Возвращает: Массив чисел, представляющих время выборок.

Метод видеоряда `dataKeyValues ()` {footageItem}.dataKeyValues (dataPath, t0 = startTime, t1=endTime)

Возвращает значения для выборок указанного динамического массива данных из файла .mgJSON. Также можно указать промежуток времени, из которого требуется вернуть выборки. По умолчанию возвращается время для всех выборок между `startTime` и `endTime` динамического потока данных в соответствии со свойством `samplesTemporalExtent` заданным в файле .mgJSON. Принимает одно значение массива, чтобы определить путь в иерархии к нужному динамическому потоку данных.

Пример:

`footage ("sample.mgjson").dataKeyTimes ([0], 1, 3)` возвращает значение образцов между 1 второй и 3 секунды для первого нижестоящего элемента.

Параметры:

`dataPath` - Массив, обязательный. Путь в иерархии к динамическому потоку данных.

`t0` -число, дополнительно. Время начала (в секундах) промежутка времени, из которого требуется вернуть выборки. По умолчанию задано значение `startTime`.

`t1` -число, дополнительно. Время окончания (в секундах) промежутка времени, из которого требуется вернуть выборки. По умолчанию задано значение `endTime`.

Возвращает: Массив чисел, представляющих значения выборки.