Руководство MotorXP-AFM Scripting API

None

MotorXP Team

Copyright © 2025 MotorXP

Table of contents

1. BE	ведение	4
1.1	Описание	4
1.2	Официальный сайт	4
1.3	Контакты	4
2. Гл	обальные функции	5
2.1	Глобальные функции	5
2.2	include()	6
2.3	require()	7
2.4	writeFile()	8
2.5	readFile()	9
3. Ко	энстанты	10
3.1	Список констант	10
3.2	Direction	12
3.3	Coil	13
3.4	Magnetization	14
3.5	PoleArrangement	15
3.6	Именованные цвета	16
4. Bo	строенные объекты	17
4.1	Встроенные объекты	17
4.2	Math	18
4.3	Geom	31
4.4	Material	76
4.5	QtWidgets	87
4.6	console	112
4.7	motor	120
5. Ти	пы объектов	134
5.1	Встроенные типы	134
5.2	Airgap	136
5.3	BoundingBox	147
5.4	Stator	162
5.5	StatorItem	191
5.6	Rotor	203
5.7	RotorItem	233
5.8	Winding	245
5.9	Mesh	282

5.10	EmptyMaterial	290
5.11	GeneralMaterial	291
5.12	IronMaterial	292
5.13	ConductorMaterial	296
5.14	WindingMaterial	297
5.15	EndturnMaterial	306
5.16	MagnetRadialMaterial	307
5.17	MagnetParallelMaterial	311
5.18	CustomMaterial	316
5.19	Point3	319
5.20	Vector3	354
5.21	Shape	364
5.22	Piece	405
5.23	UI-виджеты	409

1. Введение

1.1 Описание

Данная документация описывает программный интерфейс (API) для написания скриптов в приложении **MotorXP-AFM** для параметрического построения геометрии мотора. Включает подробную информацию о глобальных функциях, встроенных объектах, типах создаваемых объектов, материалах, константах и элементах управления (UI-виджеты).



Скриптовая машина приложения MotorXP-AFM реализована на базе скриптовой машины фреймворка Qt — QJSEngine . Для написания скриптов используется язык программирования JavaScript со спецификацией **ECMAScript 6** (ES2015).

1.2 Официальный сайт

Посетите наш веб-сайт: motorxp.com

1.3 Контакты

Если вы обнаружите какие-либо ошибки или неточности в документации, пожалуйста, сообщите о них по электронной почте: $nika_homework@mail.ru$, info@motorxp.com.

2. Глобальные функции

2.1 Глобальные функции

Название	Описание
include()	Включение содержимого одного файла скрипта в другой файл скрипта.
require()	Загрузка модуля.
writeFile()	Запись данных в файл.
readFile()	Чтение содержимого из файла.

2.2 include()

2.2.1 Описание

Функция include () предназначена для включения содержимого одного файла скрипта в другой файл скрипта. Это позволяет использовать код из одного файла (например, библиотечные функции или классы) в другом файле без необходимости копировать этот код вручную.

2.2.2 Синтаксис

```
include(path: string)
```

2.2.3 Аргументы

• path (string, обязательно):

Путь к файлу скрипта, который нужно включить:

а. Может быть абсолютным (полный путь к файлу на диске).

Например, "C:/scripts/library.js".

b. Может быть **относительным** (путь относительно текущего исполняемого файла).

Примеры:

- "./helpers/utils.js" файл находится в той же директории или поддиректории.
- "../config/settings.js" файл находится в родительской директории.

2.2.4 Возвращаемое значение

undefined

2.2.5 Пример

```
// Подключаем файл по полному пути
include("c:/scripts/script.js")

// Подключаем файл script2.js из той же директории
include("./script2.js")

// Подключаем файл script3.js из родительской директории
include("../script3.js")
```



require()

2.3 require()

2.3.1 Описание

Функция require() предназначена для загрузки модуля. В отличие от include(), функция require() возвращает объект, который содержит экспортируемые данные из подключаемого файла.

2.3.2 Синтаксис

```
require(path : string) : object
```

2.3.3 Аргументы

• path (string, обязательно):

Путь к файлу скрипта, который нужно включить:

а. Может быть абсолютным (полный путь к файлу на диске).

Например, "C:/scripts/library.js".

b. Может быть **относительным** (путь относительно текущего исполняемого файла).

Примеры

- "./helpers/utils.js" файл находится в той же директории или поддиректории.
- "../config/settings.js" файл находится в родительской директории.

Поддержка расширений файлов

Расширение .js можно опустить при указании пути. Например, require("utils") автоматически найдет файл utils.js.

2.3.4 Возвращаемое значение

оbject . Если файл не содержит явного экспорта, возвращается пустой объект ({ } }).

2.3.5 Пример

```
let path = "./helpers"
let helpers = require(path)

let library = require("C:/scripts/library")
```



include()

2.4 writeFile()

2.4.1 Описание

Запись в файл.

2.4.2 Синтаксис

```
writeFile(path : string, content : string, append : bool = true) : bool
```

2.4.3 Аргументы

- content (string, обязательно): содержимое файла.
- append (boolean, опционально, по умолчанию = true): равно true добавить в файл, иначе перед записью файл усекается.

2.4.4 Возвращаемое значение

bool : равно true — успешная запись, иначе равно false , если при записи возникли ошибки. Ошибки выводятся в консоль.

2.4.5 Пример

- let sampleText = "Sample text"
 let result = writeFile("c:/temp/sample.txt", sampleText)
 console.info(result)
 - 🔐. также

readFile()

2.5 readFile()

2.5.1 Описание

Чтение из файла.

2.5.2 Синтаксис

readFile(path : string) : string

2.5.3 Аргументы

• раth (string, обязательно): путь к файлу, может быть абсолютным или относительным, который указывает положение относительно <path_MotorXP-AFM>/bin/assets/scripts.

2.5.4 Возвращаемое значение

string: содержимое файла.

2.5.5 Пример

- let content = readFile("c:/temp/sample.txt")
 console.info(content)



writeFile()

3. Константы

3.1 Список констант

3.1.1 Direction

Константа	Описание	Значение
Direction.CW	Направление по часовой стрелке	[-1]
Direction.CCW	Направление против часовой стрелки	+1

3.1.2 Coil

Coil orientation

Константа	Описание	Значение
Coil.CW	Намотка по часовой стрелке	[-1]
Coil.CCW	Намотка против часовой стрелки	+1

Coil direction

Константа	Описание	Значение
Coil.Incoming	Incoming (входящий)	+1
Coil.Outcoming	Outcoming (исходящий)	-1

3.1.3 PoleArrangement

Константа	Описание	Значение
PoleArrangement.NN	NN	0
PoleArrangement.NS	NS	1
PoleArrangement.NSNS	NSNS	2
PoleArrangement.NSSN	NSSN	3
PoleArrangement.NNSS	NNSS	4
PoleArrangement.NNNN	NNNN	5

3.1.4 Magnetization

Константа	Описание	Значение
Magnetization.From	Направление "от" объекта	1
Magnetization.Toward	Направление "к" объекту	2
Magnetization.CW	По часовой стрелке	3
Magnetization.CCW	Против часовой стрелки	4

3.1.5 Именованные цвета

Список доступных цветов и их визуальное представление:

Константа	Название цвета	Цвет
Qt.black	Черный	
Qt.white	Белый	
Qt.darkGray	Темно-серый	
Qt.gray	Серый	
Qt.lightGray	Светло-серый	
Qt.red	Красный	
Qt.green	Зеленый	
Qt.blue	Синий	
Qt.cyan	Голубой	
Qt.magenta	Пурпурный	
Qt.yellow	Желтый	
Qt.darkRed	Темно-красный	
Qt.darkGreen	Темно-зеленый	
Qt.darkBlue	Темно-синий	
Qt.darkCyan	Темно-голубой	
Qt.darkMagenta	Темно-пурпурный	
Qt.darkYellow	Темно-желтый	

3.2 Direction

Константа	Описание	Значение
Direction.CW	Направление по часовой стрелке	[-1]
Direction.CCW	Направление против часовой стрелки	+1

3.3 Coil

3.3.1 Coil orientation

Константа	Описание	Значение
Coil.CW	Намотка по часовой стрелке	[-1]
Coil.CCW	Намотка против часовой стрелки	+1

3.3.2 Coil direction

Константа	Описание	Значение
Coil.Incoming	Incoming (входящий)	+1
Coil.Outcoming	Outcoming (исходящий)	-1

3.4 Magnetization

Константа	Описание	Значение
Magnetization.From	Направление "от" объекта	1
Magnetization.Toward	Направление "к" объекту	2
Magnetization.CW	По часовой стрелке	3
Magnetization.CCW	Против часовой стрелки	4

3.5 PoleArrangement

Константа	Описание	Значение
PoleArrangement.NN	NN	0
PoleArrangement.NS	NS	[1]
PoleArrangement.NSNS	NSNS	2
PoleArrangement.NSSN	NSSN	3
PoleArrangement.NNSS	NNSS	4
PoleArrangement.NNNN	NNNN	5

3.6 Именованные цвета

Список доступных цветов и их визуальное представление:

Константа	Название цвета	Цвет
Qt.black	Черный	
Qt.white	Белый	
Qt.darkGray	Темно-серый	
Qt.gray	Серый	
Qt.lightGray	Светло-серый	
Qt.red	Красный	
Qt.green	Зеленый	
Qt.blue	Синий	
Qt.cyan	Голубой	
Qt.magenta	Пурпурный	
Qt.yellow	Желтый	
Qt.darkRed	Темно-красный	
Qt.darkGreen	Темно-зеленый	
Qt.darkBlue	Темно-синий	
Qt.darkCyan	Темно-голубой	
Qt.darkMagenta	Темно-пурпурный	
Qt.darkYellow	Темно-желтый	

4. Встроенные объекты

4.1 Встроенные объекты

Название	Описание
console	Консоль приложения.
motor	Модель двигателя (генератора) с различными параметрами.
Math	Математические функции.
Geom	Геометрическое ядро.
<u>Material</u>	Материалы.
QtWidgets	UI-виджеты.

4.2 Math

4.2.1 Обзор объекта Math

Встроенный объект Math является расширением стандартного объекта JavaScript Math и предоставляет ряд полезных математических функций.

Основные возможности Math

- Сравнение чисел с плавающей запятой с заданной точностью (isEpsilon, isEqual, isLessEqual, isGreatEqual).
- Преобразование полярных координат в декартовые координаты (fromPolar).
- Функции для работы с угловыми значениями (rad , deg , normAngle , middleAngle).

Константы Math

• Math.EPSILON: Минимальное положительное число, такое что 1 + Math.EPSILON = 1 при работе с числами с плавающей точкой.

Методы Math

Ниже приведен список расширенных доступных методов, предоставляемых объектом Math . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Метод	Описание
Math.deg()	Перевод радиан в градусы.
Math.rad()	Перевод градусов в радианы.
Math.fromPolar()	Перевод координат точки из полярных координат (radius, angle) в декартовые (x, y) .
Math.normAngle()	Нормализует значение угла, возвращает значение в интервале (0°, 360°).
Math.middleAngle()	Расчет среднего угла между углами ат и аг с учетом направления.
Math.isEpsilon()	Возвращает true , если абсолютное значение ${\tt arg} \leq {\tt eps}$.
Math.isEqual()	Возвращает true , если разница между ${\tt arg_1}$ и ${\tt arg_2} \leq {\tt eps}$.
Math.isLessEqual()	Возвращает true , если $arg_1 \leq arg_2$ с учетом точности eps .
Math.isGreatEqual()	Возвращает true, если $arg_1 \ge arg_2$ с учетом точности eps.

4.2.2 Константы

Константы объекта Math

• $\underline{\mathtt{Math.EPSILON}}$ — Минимальное положительное число.

Math.EPSILON

ОПИСАНИЕ

Минимальное положительное число, такое что 1 + Math.EPSILON = 1 при работе с числами с плавающей точкой.

ЗНАЧЕНИЕ

1e-7

ПРИМЕР

```
let eps = Math.EPSILON
console.info(eps) // OUTPUT: 1e-7
```

4.2.3 Методы

Методы объекта Math

Ниже приведен список расширенных доступных методов, предоставляемых объектом Math . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Метод	Описание	
Math.deg()	Перевод радиан в градусы.	
Math.rad()	Перевод градусов в радианы.	
Math.fromPolar()	Перевод координат точки из полярных координат (radius, angle) в декартовые (x, y) .	
Math.normAngle()	Нормализует значение угла, возвращает значение в интервале (0°, 360°).	
Math.middleAngle()	Расчет среднего угла между углами ат и аz с учетом направления.	
Math.isEpsilon()	Возвращает true , если абсолютное значение ${\tt arg} \leq {\tt eps}$.	
Math.isEqual()	Возвращает true , если разница между arg_1 и $arg_2 \leq eps$.	
Math.isLessEqual()	Возвращает true , если ${\tt arg_1} \leq {\tt arg_2}$ с учетом точности ${\tt eps}$.	
Math.isGreatEqual()	Возвращает true, если $arg_1 \ge arg_2$ с учетом точности eps.	

Math.isEpsilon()

ОПИСАНИЕ

Возвращает true, если абсолютное значение \mbox{arg} меньше или $\mbox{paвнo}$ eps.

СИНТАКСИС

```
isEpsilon(arg : number, eps : number = Math.EPSILON) : bool
```

АРГУМЕНТЫ

- arg (number, обязательно): значение.
- eps (number, опционально): значение точности, по умолчанию равно Math.Epsilon.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

ьоо1: результат.

ПРИМЕР

```
let b1 = Math.isEpsilon(0.0001)
console.info(b1) // output: false
let b2 = Math.isEpsilon(1e-8, Math.EPSILON)
console.info(b2) // output: true
```

🔐. также

Math.isEqual()

Math.isLessEqual()

Math.isGreatEqual()

Math.isEqual()

ОПИСАНИЕ

Возвращает true , если разница абсолютных значений \arg_1 и \arg_2 меньше или равно $\,$ eps .

СИНТАКСИС

```
isEqual(arg1 : number, arg2 : number = Math.EPSILON) : bool
```

АРГУМЕНТЫ

- arg1 (number, обязательно): первое значение.
- arg2 (number, обязательно): второе значение.
- eps (number, опционально): значение точности, по умолчанию равно Math.EPSILON.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: результат.

ПРИМЕР

```
let ret = Math.isEqual(1/3, 1)
console.info(ret)
```

🗓. также

Math.isEqual()

Math.isLessEqual()

Math.isGreatEqual()

Math.isLessEqual()

ОПИСАНИЕ

Возвращает true , если значение \arg_1 меньше или равно значению \arg_2 с учетом точности \exp_3 .

СИНТАКСИС

```
isLessEqual(arg1 : number, arg2 : number, eps : number = Math.EPSILON) : number
```

АРГУМЕНТЫ

- arg1 (number, обязательно): значение 1.
- arg2 (number, обязательно): начение 2.
- eps (number, опционально): значение точности, по умолчанию равно Math.EPSILON.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: результат сравнения.

ПРИМЕР

1 let ret = Math.isLessEqual(1, 1.1)



Math.isEpsilon()

Math.isEqual()

Math.isLessEqual()

Math.isGreatEqual()

Math.isGreatEqual()

ОПИСАНИЕ

Возвращает true, если значение arg1 больше или равно значению arg2 с учетом точности eps.

СИНТАКСИС

```
isGreatEqual(arg1 : number, arg2 : number = Math.EPSILON) : number
```

АРГУМЕНТЫ

- arg1 (number, обязательно): первое значение.
- arg2 (number, обязательно): второе значение.
- ullet ерs (number, опционально): значение точности, по умолчанию равно Math.EPSILON .

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

ьоо1: результат сравнения.

ПРИМЕР

```
1 let ret = Math.isGreatEqual(1, 1.1)
```



Math.isEpsilon()

Math.isEqual()

Math.isGreatEqual()

Math.rad()

ОПИСАНИЕ

Перевод градусов в радианы.

СИНТАКСИС

```
rad(degrees : number): number
```

АРГУМЕНТЫ

degrees (number, обязательно): значение угла в градусах.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: значение угла в радианах.

ПРИМЕР

```
1 let radians = Math.rad(180)
2 console.info("180°=", radians)
```



Math.deg()

Math.normAngle()

Math.middleAngle()

Math.deg()

ОПИСАНИЕ

Перевод радиан в градусы.

СИНТАКСИС

```
deg(radians : number) : number
```

АРГУМЕНТЫ

radians (number, обязательно): значение угла в радианах.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: значение угла в градусах.

ПРИМЕР

```
let degrees = Math.deg(3.14157)
console.info("3.14157=", degrees, "°")
```



Math.rad()

Math.normAngle()

Math.middleAngle()

Math.fromPolar()

ОПИСАНИЕ

Перевод координат точки из полярных координат (radius, angle) в декартовые координаты (x, y).

СИНТАКСИС

```
fromPolar(radius : number, angle : number) : Point
```

АРГУМЕНТЫ

- radius (number, обязательно): радиус.
- angle (number, обязательно): угол в градусах.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Point: точка в декартовых координатах.

ПРИМЕР

```
let p = Math.fromPolar(100, 90)
console.info(p)
```

Math.normAngle()

ОПИСАНИЕ

Нормализирует значение угла, возвращает значение в интервале (0°, 360°) градусов.

СИНТАКСИС

```
normAngle(a : number) : number
```

АРГУМЕНТЫ

а (number, обязательно): значение угла в градусах.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number : значение угла в интервале $(0^{\circ}, 360^{\circ})$.

ПРИМЕР

```
1 let nAngle = Math.normAngle(3600)
2 console.info(nAngle)
```

🗓. также

Math.rad()

Math.deg()

Math.middleAngle()

Math.middleAngle()

ОПИСАНИЕ

Расчет среднего угла между углами a_1 и $a_{\sim 2}$ с учетом направления (${\tt CW}$ — по часовой стрелке, ${\tt CCW}$ — против часовой стрелки).

СИНТАКСИС

```
middleAngle(a1 : number, a2 : number, dir = [Direction.CW | Direction.CCW]) : number
```

АРГУМЕНТЫ

- a1 (number, обязательно): первый угол.
- a2 (number, обязательно): второй угол.
- ullet dir (number, обязательно): направление обхода, одно из констант <code>Direction.CW</code> | <code>Direction.CCW</code> .

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: средний угол.

ПРИМЕР

```
1 let mAngle = Math.middleAngle(0, 180, Direction.CW)
2 console.info(mAngle)
```

🔐. также

Math.rad()

Math.deg()

Math.normAngle()

4.3 Geom

4.3.1 Обзор объекта Geom

Встроенный объект Geom реализует геометрическое ядро.

Основные возможности Geom

Предоставляет функции для создания графических примитивов и операции с ними.

Методы объекта Geom

Ниже приведен список доступных методов, предоставляемых объектом [Geom . Щелкните имя метода, чтобы увидеть его подробное описание и примеры. СОЗДАНИЕ ГЕОМЕТРИЧЕСКИХ ПРИМИТИВОВ

- point3(): Создание точки.
- vector3(): Создание вектора.
- boundingBox(): Создание ограничивающей коробки.

2D-ПРИМИТИВЫ (ПЛОСКИЕ)

- <u>arc()</u>: Дуга.
- circle(): Круг или сектор круга.
- <u>ring()</u>: Кольцо.
- <u>ellipse()</u>: Эллипс или сектор эллипса.
- <u>square()</u>: Квадрат.
- rectangle(): Прямоугольник.
- ngon(): Правильный N-угольник.
- <u>infplane()</u>: Плоскость.
- segment(): Отрезок.
- <u>bspline()</u>: Плавная линия, проходящая через все заданные точки.
- polysegment(): Ломаная линия.
- polygon(): Многоугольник.

3D-ПРИМИТИВЫ (ОБЪЕМНЫЕ)

- <u>box()</u>: Параллелепипед.
- <u>collar()</u>: Втулка.
- <u>cone()</u>: Конус.
- <u>cylinder()</u>: Цилиндр.
- sphere(): Сфера.
- <u>torus()</u>: Top.

БУЛЕВЫ ОПЕРАЦИИ

- <u>unite()</u>: Объединение двух или более тел.
- intersect(): Пересечение двух или более тел.
- <u>diff()</u>: Разность двух или более тел.
- difference(): Разность двух или более тел.

ГЕОМЕТРИЧЕСКИЕ ОПЕРАЦИИ

- angle(): Получить угол точки.
- $\underline{angle X()}$: Получить угол точки относительно центра координат [0,0,0] в плоскости YOZ.
- angleY(): Получить угол точки относительно центра координат [0,0,0] в плоскости XOZ.
- <u>angleZ()</u>: Получить угол точки относительно центра координат [0,0,0] в плоскости XOY.
- radius(): Радиус точки в 2D-пространстве.
- radiusX(): Радиус точки в плоскости YOZ.
- radiusY(): Радиус точки в плоскости XOZ.
- radiusZ(): Радиус точки в плоскости XOY.
- angleBetweenVectors(): Угол между векторами.
- pointAtSegment(): Возвращает точку в параметризованной позиции сегмента.
- distance(): Рассчитывает расстояние между двумя точками.
- section(): Создать сечение.
- unify(): Упрощение геометрического представления объекта.
- <u>fillet()</u>: Операция скругления тела.
- <u>chamfer()</u>: Операция снятия фаски.

СОЗДАНИЕ ДЕТАЛИ

• ріесе(): Создание детали.

4.3.2 Методы

Методы объекта Geom

Ниже приведен список доступных методов, предоставляемых объектом Geom . Щелкните имя метода, чтобы увидеть его подробное описание и примеры. СОЗДАНИЕ ГЕОМЕТРИЧЕСКИХ ПРИМИТИВОВ

- point3(): Создание точки.
- vector3(): Создание вектора.
- boundingBox(): Создание ограничивающей коробки.

2D-ПРИМИТИВЫ (ПЛОСКИЕ)

- <u>arc()</u>: Дуга.
- circle(): Круг или сектор круга
- <u>ring()</u>: Кольцо.
- ellipse(): Эллипс или сектор эллипса.
- <u>square()</u>: Квадрат.
- rectangle(): Прямоугольник
- ngon(): Правильный N-угольник.
- <u>infplane()</u>: Плоскость.
- segment(): Отрезок.
- <u>bspline()</u>: Плавная линия, проходящая через все заданные точки.
- polysegment(): Ломаная линия.
- polygon(): Многоугольник.

3D-ПРИМИТИВЫ (ОБЪЕМНЫЕ)

- box(): Параллелепипед.
- collar(): Втулка.
- <u>cone()</u>: Конус
- cylinder(): Цилиндр.
- sphere(): Сфера.
- torus(): Top.

БУЛЕВЫ ОПЕРАЦИИ

- <u>unite()</u>: Объединение двух или более тел.
- intersect(): Пересечение двух или более тел.
- diff(): Разность двух или более тел.
- difference(): Разность двух или более тел.

ГЕОМЕТРИЧЕСКИЕ ОПЕРАЦИИ

- <u>angle()</u>: Получить угол точки.
- angle X(): Получить угол точки относительно центра координат [0,0,0] в плоскости YOZ.
- <u>angleY()</u>: Получить угол точки относительно центра координат [0,0,0] в плоскости XOZ.
- angleZ(): Получить угол точки относительно центра координат [0,0,0] в плоскости XOY.
- <u>radius()</u>: Радиус точки в 2D-пространстве.
- radiusX(): Радиус точки в плоскости YOZ.
- radiusY(): Радиус точки в плоскости XOZ.

- radiusZ(): Радиус точки в плоскости XOY.
- <u>angleBetweenVectors()</u>: Угол между векторами.
- pointAtSegment(): Возвращает точку в параметризованной позиции сегмента.
- distance(): Расчитывает расстояние между двумя точками.
- section(): Создать сечение.
- unify(): Упрощение геометрического представления объекта.
- <u>fillet()</u>: Операция скругления тела.
- chamfer(): Операция снятия фаски.

СОЗДАНИЕ ДЕТАЛИ

• ріесе(): Создание детали.

Geom.angle()

ОПИСАНИЕ

Получить угол точки.

СИНТАКСИС

```
Geom.angle(x: number, y: number) : number
```

АРГУМЕНТЫ

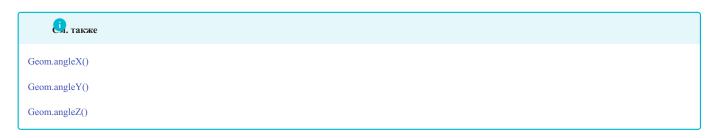
Аргумент	Тип	Значение по умолчанию	Обязательный	Описание
x	Number	_	Да	Значение координаты X
У	Number	_	Да	Значение координаты Ү

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Number: угол точки в градусах относительно центра координат [0,0] (ноль — на три часа, увеличение по часовой).

ПРИМЕР

```
let result = Geom.angle(100, 100)
console.info(result) // output: 45
```



Geom.angleBetweenVectors()

ОПИСАНИЕ

Угол между векторами.

СИНТАКСИС

```
Geom.angleBetweenVectors(v1: Vector3, v2: Vector3) : number
```

АРГУМЕНТЫ

- v1 (Vector3, обязательно): вектор 1.
- v2 (Vector3, обязательно): вектор 2.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number : угол между векторами в градусах.

ПРИМЕР

```
1 let angle = Geom.angleBetweenVectors(Geom.vetor3(10,10,10), Geom.vetor3(30,30,30))
2 console.info(angle) // output:
```



Geom.angleX()

ОПИСАНИЕ

Получить угол точки относительно центра координат [0,0,0] в плоскости YOZ.

СИНТАКСИС

```
Geom.angleX(point: Point3) : number
```

АРГУМЕНТЫ

• argument (Роіпt3, обязательно)

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: угол точки относительно центра координат [0,0,0] в плоскости YOZ(ноль - на три часа, увеличение по часовой).

ПРИМЕР

```
let point = Geom.point3(100,100,100)
let angle = Geom.angleX(point)
console.info(result) // output:
```



Geom.angleY()

ОПИСАНИЕ

Получить угол точки относительно центра координат [0,0,0] в плоскости XOZ.

СИНТАКСИС

```
Geom.angleX(point: Point3) : number
```

АРГУМЕНТЫ

• argument (Роіпt3, обязательно)

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: угол точки относительно центра координат [0,0,0] в плоскости XOZ (ноль - на три часа, увеличение по часовой).

ПРИМЕР

```
let point = Geom.point3(100,100,100)
let angle = Geom.angleY(point)
console.info(result) // output:
```



Geom.angleZ()

ОПИСАНИЕ

Получить угол точки относительно центра координат [0,0,0] в плоскости XOY.

СИНТАКСИС

```
Geom.angleX(point: Point3) : number
```

АРГУМЕНТЫ

• argument (Point3, обязательно): Точка.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: угол точки относительно центра координат [0,0,0] в плоскости XOY (ноль - на три часа, увеличение по часовой).

ПРИМЕР

```
let point = Geom.point3(100,100,100)
let angle = Geom.angleZ(point)
console.info(result) // output:
```



Geom.arc()

ОПИСАНИЕ

2D дуга, строится в плоскости XOY, углы задаются в градусах, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.arc(radius: number, angle: number) : Shape
Geom.arc(radius: number, angle1: number, angle2: number) : Shape
Geom.arc(p1: Point3, p2: Point3, p3: Point3) : Shape
```

АРГУМЕНТЫ

- radius (number, обязательно): Радиус дуги.
- angle (number, обязательно): Угол дуги в градусах в диапазоне [0...360], начальный угол равен 0.
- angle1 (number, обязательно): Начальный угол дуги в градусах в диапазоне [0...360].
- angle2 (number, обязательно): Конечный угол дуги в градусах в диапазоне [0...360].
- p1, p2, p3 (Point3, обязательно): Дуга проходящяя через три точки.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, дуга.

ПРИМЕР

```
1  let s1 = Geom.arc(100, 30)
2  console.info(s1) // output:
3  let s2 = Geom.arc(100, 30, 60)
4  console.info(s2) // output:
5  let s3 = Geom.arc(Geom.point3(0,0,0), Geom.point3(10,10,10), Geom.point3(20,20,20))
6  console.info(s3) // output:
```



Geom.boundingBox()

ОПИСАНИЕ

Возвращает ограничивающую коробку тела.

СИНТАКСИС

```
Geom.boundingBox(shp: Shape) : BoundingBox
```

АРГУМЕНТЫ

• shp (Shape, обязательно): Тело.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

BoundingBox : Объект типа BoundingBox , ограничивающая коробка тела.

ПРИМЕР

- let result = Geom.boundingBox(value)
 console.info(result)



Shape

Geom.box()

ОПИСАНИЕ

3D примитив(объемный): параллелипипед, задаётся с указанием трёх размеров х, у, z и строится относительно точки [0,0,0].

СИНТАКСИС

```
Geom.box(x: number, y: number, z: number) : Shape
Geom.box(a: number) : Shape
```

АРГУМЕНТЫ

- \times (number, обязательно): Размер параллелипипеда по оси X.
- у (number, обязательно): Размер параллелипипеда по оси Y.
- z (number, обязательно): Размер параллелипипеда по оси Z.
- а (number, обязательно): Размер куба.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Параллелипипед.

ПРИМЕР

```
let s1 = Geom.box(10, 20, 30)
console.info(s1) // output:
let s2 = Geom.box(10)
console.info(s2) // output:
```



Geom.bspline()

ОПИСАНИЕ

B-spline - 3D плавная линия, проходящая через все заданные точки. Задаётся массивом точек points .

СИНТАКСИС

```
Geom.bspline(points: array of Point3, closed: bool = false) : Shape
```

АРГУМЕНТЫ

- points (array of Point3, обязательно): Массив точек плавной кривой.
- closed (bool, опционально): Установка флага closed создает замкнутую плавную кривую.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, B-spline.

ПРИМЕР

```
let s = Geom.bspline([Geom.point3(10,10,10), Geom.point3(20,20,20), Geom.point3(30,30,30)])
console.info(s) // output:
```



Geom.chamfer()

ОПИСАНИЕ

Операция взятия фаски тела. В отличие от скругления применяется только к объёмным телам. Фаска задаётся расстоянием г, взятым от ребра до линии фаски и массивом ближайших точек refs.

СИНТАКСИС

```
Geom.chamfer(shp: Shape, r: number) : Shape
Geom.chamfer(shp: Shape, r: number, refs: array of Point3) : Shape
```

АРГУМЕНТЫ

- shp (Shape, обязательно): Тело для снятия фаски.
- г (number, обязательно): Расстояние, взятым от ребра до линии фаски.
- refs (Array of Point3, обязательно): массивом ближайших точек refs к ребрам, требующих снятия фаски.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, результат операции снятия фаски.

ПРИМЕР

```
let box = Geom.box(100, 100, 100)
let s = Geom.chamfer(box, 1);
console.info(result)
```



Geom.fillet

Geom.circle()

ОПИСАНИЕ

2D круг или сектор круга, строится в плоскости XOY, углы задаются в градусах, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.circle(radius: number) : Shape
Geom.circle(radius: number, angle: number) : Shape
Geom.circle(radius: number, angle1: number, angle2: number) : Shape
```

АРГУМЕНТЫ

- radius (number, обязательно): Радиус круга.
- angle (number, обязательно): угол сектора круга в градусах в диапазоне [0...360], тело центрируется относительно оси ОХ.
- angle1 (number, обязательно): Начальный угол сектора круга в градусах в диапазоне [0...360].
- angle2 (number, обязательно): Конечный угол сектора круга в градусах в диапазоне [0...360].

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, круг или сектор круга.

ПРИМЕР

```
1  let s1 = Geom.circle(100)
2  console.info(s1) // output:
3  let s2 = Geom.circle(100, 30)
4  console.info(s2) // output:
5  let s3 = Geom.circle(100, 0, 90)
6  console.info(s3) // output:
```



Geom.collar()

ОПИСАНИЕ

3D примитив(объемный): втулка (цилиндр с осевым отверстием), задаётся с указанием внешненго радиуса r1, внутреннего радиуса r2 и высоты h и строится относительно точки [0,0,0]. Возможно построение сектора втулки с использованием параметра angle.

СИНТАКСИС

```
Geom.collar(r1: number, r2: number, h: number) : Shape
Geom.collar(r1: number, r2: number, h: number, angle: number) : Shape
```

АРГУМЕНТЫ

- r1 (number, обязательно): Внешний радиус втулки.
- r2 (number, обязательно): Внутренний радиус втулки.
- h (number, обязательно): Высота втулки.
- angle (number, обязательно): Угол сектора втулки в градусах, тело центрируется относительно оси ОХ.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Втулка.

ПРИМЕР

```
1 let s1 = Geom.collar(200, 50, 100)
2 console.info(s1) // output:
3 let s2 = Geom.collar(200, 50, 100, 30)
4 console.info(s2) // output:
```



Geom.cone()

ОПИСАНИЕ

3D примитив(объемный): конус, задаётся с указанием нижнего радиуса r1, верхнего радиуса r2 и высоты h и строится относительно точки [0,0,0]. Возможно построение сектора конуса с использованием параметра angle.

СИНТАКСИС

```
Geom.cone(r1: number, r2: number, h: number) : Shape
Geom.cone(r1: number, r2: number, h: number, angle: number) : Shape
```

АРГУМЕНТЫ

- r1 (number, обязательно): Нижний радиус конуса.
- r2 (number, обязательно): Верхний радиус конуса.
- h (number, обязательно): Высота конуса.
- angle (number, обязательно): Угол сектора конуса в градусах, тело центрируется относительно оси ОХ.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Конус.

ПРИМЕР

```
let s1 = Geom.cone(100, 50, 100)
console.info(s1) // output:
let s2 = Geom.cone(100, 50, 100, 30)
console.info(s2) // output:
```



Geom.cylinder()

ОПИСАНИЕ

3D примитив(объемный): цилиндр, задаётся с указанием радиуса r и высоты h и строится относительно точки [0,0,0]. Возможно построение сектора цилиндра с использованием параметра angle.

СИНТАКСИС

```
Geom.cylinder(r: number, h: number) : Shape
Geom.cylinder(r: number, h: number, angle: number) : Shape
```

АРГУМЕНТЫ

- г (number, обязательно): Радиус цилиндра.
- h (number, обязательно): Высота цилиндра.
- \bullet angle (number, обязательно): Угол сектора цилиндра в градусах, тело центрируется относительно оси OX.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, цилиндр.

ПРИМЕР

```
let s1 = Geom.cylinder(100, 200)
console.info(s1) // output:
let s2 = Geom.cylinder(100, 200, 30)
console.info(s2) // output:
```



Geom.diff()

ОПИСАНИЕ

Булева операция: Разность двух или более тел.

СИНТАКСИС

```
Geom.diff(a: Shape, b: Shape,) : Shape
Geom.diff(shapes: array of Shape) : Shape
```

- а (Shape, обязательно): Тело 1.
- b (Shape, обязательно): Тело 2.
- shapes (Array of Shape, обязательно): Массив тел.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Объединение тел.

ПРИМЕР

```
1 let box = Geom.box(100, 100, 100);
2 let cyl = Geom.cylinder(50, 150);
3 let s1 = Geom.diff(box, cyl) // box - cyl
4 console.info(s1) // output:
5 let s2 = Geom.diff(cyl, box) // cyl - box
6 console.info(s2) // output:
```



Geom.unite

Geom.intersect

Geom.difference

Geom.difference()

ОПИСАНИЕ

Булева операция: Разность двух или более тел.

СИНТАКСИС

```
Geom.difference(a: Shape, b: Shape,) : Shape
Geom.difference(shapes: array of Shape) : Shape
```

- а (Shape, обязательно): Тело 1.
- b (Shape, обязательно): Тело 2.
- shapes (Array of Shape, обязательно): Массив тел.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, объединение тел.

ПРИМЕР

```
let box = Geom.box(100, 100, 100);
let cyl = Geom.cylinder(50, 150);
let s1 = Geom.difference(box, cyl) // box - cyl
console.info(s1) // output:
let s2 = Geom.difference(cyl, box) // cyl - box
console.info(s2) // output:
```

🛄. также

Geom.diff

Geom.distance()

ОПИСАНИЕ

Расчитывает расстояние между двумя точками.

СИНТАКСИС

```
Geom.distance(p1: Point3, p2: Point3) : number
```

АРГУМЕНТЫ

- p1 (Point3, обязательно): Точка 1.
- p2 (Point3, обязательно): Точка 2.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: расстояние между двумя точками.

ПРИМЕР

```
1 let p1 = Geom.point3(0, 0, 0)
2 let p2 = Geom.point3(100, 100, 100)
3 let dist = Geom.distance(p1, p2)
4 console.info(dist) // output:
```



Geom.ellipse()

ОПИСАНИЕ

2D эллипс или сектор эллипса, строится в плоскости ХОУ, углы задаются в градусах, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.ellipse(radius1: number, radius2: number) : Shape
Geom.ellipse(radius1: number, radius2: number, angle1: number, angle2: number) : Shape
```

АРГУМЕНТЫ

- radius1 (number, обязательно): Радиус 1 эллипса.
- radius2 (number, обязательно): Радиус 2 эллипса.
- angle1 (number, обязательно): Начальный угол сектора эллипса в градусах в диапазоне [0...360].
- angle2 (number, обязательно): Конечный угол сектора эллипса в градусах в диапазоне [0...360].

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, эллипс или сектор эллипса.

ПРИМЕР

```
let s1 = Geom.ellipse(100, 200)
console.info(s1) // output:
let s2 = Geom.ellipse(100, 200, 0, 90)
console.info(s2) // output:
```



Geom.fillet()

ОПИСАНИЕ

Операция скругления тела. Если тело объёмное - модификации подвергаются ребра. Если плоское - вершины. Скругления задаются радиусом г и массивом ближайших точек refs к модифицируемым ребрам/вершинам.

СИНТАКСИС

```
Geom.fillet(shp: Shape, r: number) : Shape
Geom.fillet(shp: Shape, r: number, refs: array of Point3) : Shape
```

АРГУМЕНТЫ

- shp (Shape, обязательно): Тело для скругления.
- г (number, обязательно): Радиус скругления.
- refs (Array of Point3, обязательно): массивом ближайших точек refs к ребрам/вершинам, требующих скругления.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, результат операции скругления.

ПРИМЕР

```
1 let box = Geom.box(100, 100, 100)
2 let s = Geom.fillet(box, 1);
3 console.info(s) // output:
```



Geom.chamfer

Geom.infplane()

ОПИСАНИЕ

Создается плоскость.

СИНТАКСИС

АРГУМЕНТЫ

- pos (Point3, обязательно): Позиция плоскости.
- dir (Vector3, обязательно): Наравление, вектор перпендикулярный плоскости.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Плоскость.

ПРИМЕР

```
let s = Geom.infplane(Geom.point(0,0,0), Geom.vector3(0,0,1));
console.info(s);
```



Geom.intersect()

ОПИСАНИЕ

Булева операция: получить пересечение двух или более тел.

СИНТАКСИС

```
Geom.intersect(a: Shape, b: Shape,) : Shape
Geom.intersect(shapes: array of Shape) : Shape
```

- а (Shape, обязательно): Тело 1.
- b (Shape, обязательно): Тело 2.
- shapes (Array of Shape, обязательно): Массив тел.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Объединение тел.

ПРИМЕР

```
1 let box = Geom.box(100, 100, 100);
2 let cyl = Geom.cylinder(50, 150);
3 let s1 = Geom.intersect(box, cyl) // box & cyl
4 console.info(s1) // output:
```

. также

Geom.unite

Geom.diff

Geom.difference

Geom.ngon()

ОПИСАНИЕ

2D Правильный N-угольник, строится в плоскости XOY, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.ngon(radius: number, n: number) : Shape
```

АРГУМЕНТЫ

- radius (number, обязательно): Радиус N-угольника.
- п (number, обязательно): Количество углов N-угольника (допустимое значение: 3 и более).

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape : объект типа Shape , Правильный N-угольник.

ПРИМЕР

```
let s = Geom.ngon(100, 6)
console.info(s)
```



Geom.piece()

ОПИСАНИЕ

Создает деталь мотора из геометрической формы и материала.

СИНТАКСИС

```
Geom.piece(shp: Shape, material: Material) : Piece
```

АРГУМЕНТЫ

- shp (Shape, обязательно): Геометрическая форма детали.
- material (Material, обязательно): Материал детали.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Ріесе: Объект типа Ріесе, деталь мотора.

ПРИМЕР

```
let shp = Geom.cylinder(100, 100)
let material = Material.general();
let piece = Geom.piece(shp, material)
console.info(piece) // output:
```



Geom

Material

Geom.point3()

ОПИСАНИЕ

Создание точки в 3D пространстве.

СИНТАКСИС

```
Geom.point3(x: number, y: number, z: number) : Point3
Geom.point3(p: QPointF, z: number) : Point3
```

АРГУМЕНТЫ

- \times (number, обязятельно): \times координата точки.
- у (number, обязятельно): у координата точки.
- z (number, обязятельно): z координата точки.
- р (QPointF, обязятельно): Точка с координатами [x,y].

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Point3: объект типа Point3.

ПРИМЕР

```
let p1 = Geom.point3()
console.info(p1) // output:
let p2 = Geom.point3(10, 10, 10)
console.info(p2) // output:
let pt = Qt.point(10, 10)
let p3 = Geom.point3(pt, 10)
console.info(p2) // output:
```



Geom.vector()

Geom.pointAtSegment()

ОПИСАНИЕ

Возвращает точку в параметризованной позиции сегмента, указанной t. Функция возвращает начальную точку сегмента, если t=0, и ее конечную точку, если t=1.

СИНТАКСИС

```
Geom.pointAtSegment(p1: Point3, p2: Point3, t : number) : Point3
```

АРГУМЕНТЫ

- p1 (Point3, обязательно): Первая точка сегмента.
- p2 (Point3, обязательно): Вторая точка сегмента.
- t (number, обязательно): Параметризованная позиция сегмента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Point3: Возвращает точку Point3

ПРИМЕР

```
let p1 = Geom.point3(0, 0, 0)
let p2 = Geom.point3(100, 100, 100)
let p3 = Geom.pointAtSegment(p1, p2, 0.5)
console.info(p3) // output:
```



Geom.polygon()

ОПИСАНИЕ

2D Многоугольник, строится в плоскости XOY, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.polygon(points: array of Point3) : Shape
```

АРГУМЕНТЫ

• points (array of Point3, обязательно): Массив вершин многоугольника.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, Многоугольник.

ПРИМЕР

```
let s = Geom.polygon([Geom.point3(0,0,0), Geom.point3(1,0,0), Geom.point3(1,1,0), Geom.point3(0,1,0), Geom.point3(0,0,0)]);
console.info(s);
```



Geom.polysegment()

ОПИСАНИЕ

Полисегмент - 3D ломанная линия. Задаётся массивом точек points.

СИНТАКСИС

```
Geom.polysegment(points: array of Point3, closed: bool = false) : Shape
```

АРГУМЕНТЫ

- points (array of Point3, обязательно): Массив точек полилинии.
- closed (bool, опционально): Установка флага closed создает замкнутую полилинию, добавляет сегмент полилинии, идущий от точки конца к точке начала.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, полилиния.

ПРИМЕР

```
let s = Geom.polysegment([Geom.point3(10,10,10), Geom.point3(20,20,20), Geom.point3(30,30,30)])
console.info(s) // output:
```



Geom.radius()

ОПИСАНИЕ

Радиус точки в 2D пространстве.

СИНТАКСИС

```
Geom.radius(x: number, y: number) : number
```

АРГУМЕНТЫ

- x (number, обязательно): x координата точки.
- у (number, обязательно): у координата точки.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Paдиус точки с координатами [x,y] относительно точки ноль [0,0].

ПРИМЕР

```
let result = Geom.radius(10, 10)
console.info(result) // output:
```



Geom.radiusX()

ОПИСАНИЕ

Радиус точки в плоскости YOZ.

СИНТАКСИС

```
Geom.radiusX(point: Point3) : number
Geom.radiusX(x: number, y: number, z: number,) : number
```

АРГУМЕНТЫ

- point (Point3, обязательно): точка в 3D координатах.
- х (х, обязательно): х координата точки.
- у (у, обязательно): у координата точки.
- z (z, обязательно): z координата точки.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Радиус точки относительно точки ноль [0,0,0].

ПРИМЕР

```
let radius1 = Geom.radiusX(Geom.point3(10,10,10))
console.info(radius1) // output:
let radius2 = Geom.radiusX(10,10,10)
console.info(radius2) // output:
```



Geom.radiusY()

ОПИСАНИЕ

Радиус точки в плоскости XOZ, ноль - [0,0,0].

СИНТАКСИС

```
Geom.radiusY(point: Point3) : number
Geom.radiusY(x: number, y: number, z: number,) : number
```

АРГУМЕНТЫ

- point (Point3, обязательно): точка в 3D координатах.
- х (х, обязательно): х координата точки.
- у (у, обязательно): у координата точки.
- z (z, обязательно): z координата точки.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Радиус точки.

ПРИМЕР

""javascript linenums="1" let radius1 = Geom.radiusY(Geom.point3(10,10,10)) console.info(radius1) // output: let radius2 = Geom.radiusY(10,10,10) console.info(radius2) // output:



Geom.radiusZ()

ОПИСАНИЕ

Радиус точки в плоскости ХОҮ, ноль - [0,0,0].

СИНТАКСИС

```
Geom.radiusZ(point: Point3) : number
Geom.radiusZ(x: number, y: number, z: number,) : number
```

АРГУМЕНТЫ

- point (Point3, обязательно): точка в 3D координатах.
- х (х, обязательно): х координата точки.
- у (у, обязательно): у координата точки.
- z (z, обязательно): z координата точки.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Радиус точки.

ПРИМЕР

""javascript linenums="1" let radius1 = Geom.radiusZ(Geom.point3(10,10,10)) console.info(radius1) // output: let radius2 = Geom.radiusZ(10,10,10) console.info(radius2) // output:



Geom.rectangle()

ОПИСАНИЕ

2D Прямоугольник, строится в плоскости XOY, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.rectangle(a: number, b: number) : Shape
```

АРГУМЕНТЫ

- ullet а (number, обязательно): Сторона прямоугольника по X.
- b (number, обязательно): Сторона прямоугольника по Y.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape : объект типа Shape , Прямоугольник.

ПРИМЕР

```
let s = Geom.rectangle(300, 200);
console.info(s) // output:
```



Geom.ring()

ОПИСАНИЕ

2D кольцо или сектор кольца, строится в плоскости ХОУ, углы задаются в градусах, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.ring(radius1: number, radius2: number) : Shape
Geom.ring(radius1: number, radius2: number, angle: number) : Shape
Geom.ring(radius1: number, radius2: number, angle1: number, angle2: number) : Shape
```

АРГУМЕНТЫ

- radius1 (number, обязательно): Наружний радиус кольца.
- radius2 (number, обязательно): Внутренний радиус кольца.
- angle (number, обязательно): угол сектора кольца в градусах в диапазоне [0...360], тело центрируется относительно оси ОХ.
- angle1 (number, обязательно): Начальный угол сектора кольца в градусах в диапазоне [0...360].
- angle2 (number, обязательно): Конечный угол сектора кольца в градусах в диапазоне [0...360].

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, кольцо или сектор кольца.

ПРИМЕР

```
1 let s1 = Geom.ring(100, 50)
2 console.info(s1) // output:
3 let s2 = Geom.ring(100, 50, 30)
4 console.info(s2) // output:
5 let s3 = Geom.circle(100, 50, 0, 90)
6 console.info(s3) // output:
```



Geom.section()

ОПИСАНИЕ

Создание сечения.

СИНТАКСИС

```
Geom.section(a: Shape, b: Shape) : Shape
```

АРГУМЕНТЫ

- а (Shape, обязательно): Тело 1.
- b (Shape, обязательно): Тело 2.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: Объект типа Shape, сечение.

ПРИМЕР

```
1 let a = Geom.box(100, 100, 100)
2 let b = Geom.inflate(Geom.point3(50,50,50), Geom.vector(100, 100, 100))
3 let result = Geom.section(a, b)
4 console.info(result) // output:
```



Geom.segment()

ОПИСАНИЕ

Создание 3D отрезка, задается двумя точками.

СИНТАКСИС

```
Geom.segment(argument: type) : Shape
```

АРГУМЕНТЫ

- p1 (Point3, обязательно): Начальная точка отрезка.
- p2 (Point3, обязательно): Конечная точка отрезка.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа | Shape, отрезок.

ПРИМЕР

```
1 let s = Geom.segment(Geom.point3(10,10,10), Geom.point3(20,20,20))
2 console.info(s) // output:
```



Geom.sphere()

ОПИСАНИЕ

3D примитив(объемный): сфера, строится относительно точки [0,0,0].

СИНТАКСИС

```
Geom.sphere(r: number) : Shape
Geom.sphere(r: number, an1: number) : Shape
Geom.sphere(r: number, an1: number, an2: number) : Shape
Geom.sphere(r: number, an1: number, an2: number) : Shape
```

АРГУМЕНТЫ

- г (number, обязательно): Радиус сферы.
- an1 (number, обязательно):
- an2 (number, обязательно):
- an3 (number, обязательно):

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

```
Shape: объект типа Shape, Сфера.
```

ПРИМЕР

```
1 let s = Geom.sphere(100)
2 console.info(s) // output:
```



Geom.square()

ОПИСАНИЕ

2D Квадрат, строится в плоскости XOY, центр фигуры в точке [0,0,0].

СИНТАКСИС

```
Geom.square(a: number) : Shape
```

АРГУМЕНТЫ

• a (number, обязательно): Сторона квадрата.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

```
Shape: объект типа Shape, Квадрат.
```

ПРИМЕР

```
1 let s = Geom.square(100)
2 console.info(s) // output:
```



Geom.torus()

ОПИСАНИЕ

3D примитив (объемный): тор, строится относительно точки [0,0,0].

СИНТАКСИС

```
Geom.torus(r1: number) : Shape
Geom.torus(r1: number, r2: number) : Shape
Geom.torus(r1: number, r2: number, ua: number) : Shape
Geom.torus(r1: number, r2: number, va1: number, va2: number) : Shape
Geom.torus(r1: number, r2: number, va1: number, va2: number) : Shape
```

АРГУМЕНТЫ

- r1 (number, обязательно):
- r2 (number, обязательно):
- ua (number, обязательно):
- val (number, обязательно):
- va2 (number, обязательно):

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

```
Shape: объект типа Shape, тор.
```

ПРИМЕР

```
1 let s = Geom.torus(100)
2 console.info(s) // output:
```



Geom.unify()

ОПИСАНИЕ

Упрощает геометрическое представление объекта, путём удаления лишних рёбер и объединения граней.

СИНТАКСИС

```
Geom.unify(shp: Shape) : Shape
```

АРГУМЕНТЫ

• shp (Shape, обязательно): Тело для операции упрощения.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Shape: объект типа Shape, результат операции упрощения.

ПРИМЕР

```
let box = Geom.box(100, 100, 100)
let s = Geom.unify(box)
console.info(s) // output:
```



Geom.unite

Geom.intersect

Geom.diff

Geom.difference

Geom.unite()

ОПИСАНИЕ

Булева операция: Объединить два или более тел.

СИНТАКСИС

```
Geom.unite(a: Shape, b: Shape
Geom.unite(shapes: array of Shape) : Shape
```

АРГУМЕНТЫ

- а (Shape, обязательно): Тело 1.
- b (Shape, обязательно): Тело 2.
- shapes (Array of Shape, обязательно): Массив тел.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

```
Shape: объект типа Shape, Объединение тел.
```

ПРИМЕР

```
let box = Geom.box(100, 100, 100);
let cyl = Geom.cylinder(50, 150);
let result = Geom.unite(box, cyl) // box + cyl
console.info(result) // output:
```

. также

Geom.intersect

Geom.diff

Geom.difference

Geom.vector3()

ОПИСАНИЕ

Создание вектора в 3D пространстве, задает направление и длину.

СИНТАКСИС

```
Geom.vector3(x: number, y: number) : Vector3
Geom.vector3(p: Point3) : Vector3
Geom.vector3(p1: Point3, p2: Point3) : Vector3
```

АРГУМЕНТЫ

- x (number, обязательно): х координата точки.
- у (number, обязательно): у координата точки.
- z (number, обязательно): z координата точки.
- р (Point3, обязательно): Точка.
- p1 (Point3, обязательно): Начальная точка.
- p2 (Point3, обязательно): Конечная точка.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Vector: объект тира Vector3.

ПРИМЕР

```
let v1 = Geom.vector3(10,10,10)
console.info(v1) // output:
let v2 = Geom.vector3(Geom.point3(10,10,10))
console.info(v2) // output:
let v3 = Geom.vector3(Geom.point3(20,20,20), Geom.point3(10,10,10))
console.info(v3) // output:
```



4.4 Material

4.4.1 Обзор объекта Material

Встроенный объект Material предназначен для создания объектов различных материалов.

Основные возможности Material

Создает различные материалы для деталей 'Ріесе' двигателя.

Методы объекта Material

Ниже приведен список доступных методов, предоставляемых объектом Material . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- Material.empty(): Создает пустой материал.
- Material.general(): Создает материал для воздушной среды.
- Material.iron(): Создает материал для железа.
- Material.winding(): Создает материал для обмотки.
- Material.endturn(): Создает материал для лобовых частей обмоток.
- Material.conductor(): Создает материал для проводника.
- Material.magnetParallel(): Создает материал для магнита с параллельным намагничиванием.
- Material.magnetRadial(): Создает материал для магнита с радиальным намагничиванием.
- Material.custom(): Создает пользовательский материал с заданным цветом.



<u>Piece</u>

4.4.2 Методы

Методы объекта Material

Ниже приведен список доступных методов, предоставляемых объектом Material . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- Material.empty(): Создает пустой материал.
- Material.general(): Создает материал для воздушной среды.
- Material.iron(): Создает материал для железа.
- Material.winding(): Создает материал для обмотки.
- Material.endturn(): Создает материал для лобовых частей обмоток.
- Material.conductor(): Создает материал для проводника.
- Material.magnetParallel(): Создает материал для магнита с параллельным намагничиванием.
- <u>Material.magnetRadial()</u>: Создает материал для магнита с радиальным намагничиванием.
- Material.custom(): Создает пользовательский материал с заданным цветом.

Material.empty()

ОПИСАНИЕ

Пустой материал.

СИНТАКСИС

Material.empty(): EmptyMaterial

АРГУМЕНТЫ

Отсутствуют

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

EmptyMaterial

ПРИМЕР

var m = Material.empty();
console.info(m) // output: Not assigned



Material.general()

Material.iron()

Material.winding()

Material.endturn()

Material.conductor()

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.general()

ОПИСАНИЕ

Материал для воздушной среды.

СИНТАКСИС

```
Material.general() : GenetalMaterial
```

АРГУМЕНТЫ

Отсутствуют

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

GeneralMaterial

ПРИМЕР

```
let m = Material.general()
console.info(m) // output: General
```

🛄. также

Material.empty()

Material.iron()

Material.winding()

Material.endturn()

Material.conductor()

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.iron()

ОПИСАНИЕ

Материал для железа.

СИНТАКСИС

```
Material.iron(fillCoefs: array of number = [], dsomaloy: number = 0) : IronMaterial
```

АРГУМЕНТЫ

- fillCoefs (array of number, опционально): Material filling coefficient
- dsomaloy (number, опционально): Smallest cross section of component [m] for Somaloy material

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

IronMaterial

ПРИМЕР

```
let m = Material.iron()
console.info(m) // output: Iron
```



Material.empty()

Material.general()

Material.winding()

Material.endturn()

Material.conductor()

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.conductor()

ОПИСАНИЕ

Материал для проводника.

СИНТАКСИС

Material.conductor() : ConductorMaterial

АРГУМЕНТЫ

Отсутствуют

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

ConductorMaterial

ПРИМЕР

1 let m = Material.conductor()
2 console.info(m) // output: Conductor



Material.empty()

Material.general()

Material.iron()

Material.winding()

Material.endturn()

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.winding()

ОПИСАНИЕ

Материал для обмотки.

СИНТАКСИС

АРГУМЕНТЫ

- layer (number, опционально): Homep слоя, одно из значений Winding.LayerAuto | Winding.LayerSingle | Winding.LayerDouble.
- turn (number, опционально): Номер витка.
- strand (number, опционально): Номер провода в витке.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

WindingMaterial

ПРИМЕР

```
let m1 = Material.winding()
console.info(m1) // output: Winding, layer None
let m2 = Material.winding(Winding.SingleLayer, 1, 1)
console.info(m2) // output: Winding, layer 1, turn 1, strand 1
```



Material.empty()

Material.general()

Material.iron()

Material.endturn()

<u>Material.conductor()</u>

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.endturn()

ОПИСАНИЕ

Материал для лобовых частей обмоток (Endturns).

СИНТАКСИС

```
Material.endturn() : EndturnMaterial
```

АРГУМЕНТЫ

Отсутствуют

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

EndturnMaterial

ПРИМЕР

1 let m = Material.endturn()
2 console.info(m) // output: EndTurn



Material.empty()

Material.general()

Material.iron()

Material.winding()

Material.conductor()

Material.magnetParallel()

Material.magnetRadial()

Material.custom()

Material.magnetParallel()

ОПИСАНИЕ

Материал для магнита с параллельным намагничиванием.

СИНТАКСИС

```
Material.magnetParallel(angle: number = 0, segmentRadiuses: array of number = [], savePoleBorder: bool = true) : MagnetParallelMaterial
```

АРГУМЕНТЫ

- angle (string, обязательно): Угол параллельного намагничивания
- segmentRadiuses (array of number, опционально): массив радиусов сегментов магнита
- savePoleBorder (bool, опционально): =true сохраняем границу между магнитами двух соседних полюсов, если =false магнит сливается с магнитом соседнего полюса.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MagnetParallelMaterial

ПРИМЕР

```
let m = Material.magnetParallel()
console.info(m) // output: Magnet, Parallel 0°, 1 segment(s)
```



Material.empty()

Material.general()

Material.iron()

Material.winding()

Material.endturn()

Material.conductor()

Material.magnetRadial()

Material.custom()

Material.magnetRadial()

ОПИСАНИЕ

Материал для магнита с радиальным намагничиванием.

СИНТАКСИС

```
Material.magnetRadial(direction: Magnetization.From, center: point = Qt.point(0, 0), segmentRadiuses: array of numbers = []) : MagnetRadialMaterial
```

АРГУМЕНТЫ

- direction (string, обязательно): Направление магнетизации.
- center (QPointF, опционально): Центр магнизации.
- segmentRadiuses (array of number, опционально): Массив радиусов сегментов магнита.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MagnetRadialMaterial

ПРИМЕР

```
let m = Material.magnetRadial();
console.info(m) // output: Magnet, Radial From center (0, 0), 1 segment(s)
```



Magnetization

Material.empty()

Material.general()

Material.iron()

Material.winding()

Material.endturn()

 $\underline{Material.conductor()}$

Material.magnetParallel()

Material.custom()

Material.custom()

ОПИСАНИЕ

Пользовательский материал с заданным цветом.

СИНТАКСИС

```
Material.custom(color : Qt.color = Qt.red) : CustomMaterial
```

АРГУМЕНТЫ

• color (Qt.color, опционально): цвет материала.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

CustomMaterial

ПРИМЕР

```
let m1 = Material.custom()
console.info(m1) // output: Custom, color #ff0000
let m2 = Material.custom(Qt.green)
console.info(m2) // output:
```



Material.empty()

Material.general()

Material.iron()

Material.winding()

Material.endturn()

Material.conductor()

Material.magnetParallel()

Material.magnetRadial()

4.5 QtWidgets

4.5.1 Обзор объекта QtWidgets

Встроенный объект QtWidgets создает различные виджеты для создания экранных форм пользовательского интерфейса.

Основные возможности QtWidgets

Создание виджетов для построения форм пользовательского интерфейса.

Методы объекта QtWidgets

Ниже приведен список доступных методов, предоставляемых объектом <code>QtWidgets</code> . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

4.5.2 Методы

Методы объекта QtWidgets

Ниже приведен список доступных методов, предоставляемых объектом QtWidgets . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

QtWidgets.createQGridLayout()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQGridLayout(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQGridLayout(value)
 console.info(result)



QtWidgets.createQFormLayout()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createQFormLayout(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQFormLayout(value)
 console.info(result)



QtWidgets.createQWidget()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQWidget(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQWidget(value)
 console.info(result)



QtWidgets.createQLabel()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQLabel(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQLabel(value)
 console.info(result)



QtWidgets.createQLineEdit()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQLineEdit(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQLineEdit(value)
 console.info(result)



QtWidgets.createQPushButton()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQPushButton(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQPushButton(value)
 console.info(result)



QtWidgets.createQSpinBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQSpinBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQSpinBox(value)
 console.info(result)



QtWidgets.createQDoubleSpinBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createQDoubleSpinBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQDoubleSpinBox(value)
 console.info(result)



QtWidgets.createQComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQComboBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQComboBox(value)
 console.info(result)



QtWidgets.createQGroupBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQGroupBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQGroupBox(value)
 console.info(result)



QtWidgets.createQCheckBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createQCheckBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createQCheckBox(value)
 console.info(result)



QtWidgets.createWarningIcon()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createWarningIcon(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createWarningIcon(value)
 console.info(result)



QtWidgets.createExclamationIcon()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createExclamationIcon(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createExclamationIcon(value)
 console.info(result)



QtWidgets.createNumberEdit()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createNumberEdit(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createNumberEdit(value)
 console.info(result)



QtWidgets.createWindingLayersComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createWindingLayersComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createWindingLayersComboBox(value)
 console.info(result)



QtWidgets.createWindingLayersOrientationComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createWindingLayersOrientationComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР



QtWidgets.createWindingTypeComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createWindingTypeComboBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createWindingTypeComboBox(value)
 console.info(result)



QtWidgets.createStatorTypeComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createStatorTypeComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createStatorTypeComboBox(value)
 console.info(result)



QtWidgets.createRotorTypeComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createRotorTypeComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createRotorTypeComboBox(value)
 console.info(result)



QtWidgets.createPoleArrangementComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createPoleArrangementComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР



QtWidgets.createNumberSlotSpinBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createNumberSlotSpinBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createNumberSlotSpinBox(value)
 console.info(result)



QtWidgets.createNumberPoleSpinBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createNumberPoleSpinBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = QtWidgets.createNumberPoleSpinBox(value)
 console.info(result)



QtWidgets.createWindingConnectionComboBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

createWindingConnectionComboBox(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- $\label{lem:console} \mbox{let result} = \mbox{QtWidgets.createWindingConnectionComboBox} \mbox{(value)} \\ \mbox{console.info} \mbox{(result)}$



4.6 console

4.6.1 Обзор объекта console

Объект console предоставляет методы для вывода информации, предупреждений, ошибок и отладочных данных в консоль приложения мотоххр-AFM. Это мощный инструмент для разработчиков, позволяющий отслеживать и отлаживать их скрипты.

Основные возможности console

- Вывод сообщений с различными уровнями (log, info, warn, error).
- Очистка вывода консоли (clear).
- Просмотр объектов и их свойств (dir).

Методы объекта console

Ниже приведен список доступных методов, предоставляемых объектом console. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- console.log(): Выводит сообщение серым цветом в консоль.
- console.info(): Выводит информационное сообщение зеленым цветом в консоль.
- console.warn(): Выводит предупреждающее сообщение желтым цветом в консоль.
- console.error(): Выводит сообщение об ошибке красным цветом в консоль.
- console.clear(): Очищает вывод консоли, удаляя все ранее зарегистрированные сообщения.
- console.dir(): Отображает список свойств и методов указанного JavaScript-объекта.

4.6.2 Методы

Методы объекта console

Ниже приведен список доступных методов, предоставляемых объектом console. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- console.log(): Выводит сообщение серым цветом в консоль.
- console.info(): Выводит информационное сообщение зеленым цветом в консоль.
- <u>console.warn()</u>: Выводит предупреждающее сообщение желтвм цветом в консоль.
- console.error(): Выводит сообщение об ошибке красным цветом в консоль.
- console.clear(): Очищает вывод консоли, удаляя все ранее зарегистрированные сообщения.
- console.dir(): Отображает список свойств и методов указанного JavaScript-объекта.

console.log()

ОПИСАНИЕ

Metod console.log() выводит сообщение серым цветом в консоль, каждое сообщение помечается иконкой bug. Он может принимать несколько аргументов и отображать их в одной строке.

СИНТАКСИС

```
console.log(message1 : any, message2 : any, ..., messageN : any)
```

АРГУМЕНТЫ

• messageN (апу, обязательно):

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefined

ПРИМЕР

```
// Один аргумент
console.log('log text')

// Несколько аргументов
console.log('text1', 'text2')
console.log('text1', 'text2', ..., 'text10')

// Различные типы аргументов
console.log('text1', 123, true, { key: 'value' })
```

console.info()
console.warn()
console.error()

console.info()

ОПИСАНИЕ

Metog console.info() выводит информационное сообщение зеленым цветом в консоль, каждое сообщение помечается иконкой info. Он похож на console.log(), но предназначен для информационных целей. Он может принимать несколько аргументов и отображать их в одной строке.

СИНТАКСИС

```
console.info(message1 : any, message2 : any, ..., messageN : any)
```

АРГУМЕНТЫ

• messageN (апу, обязательно):

возвращаемое значение

undefined

ПРИМЕР

```
// Один аргумент
console.info('info text')

// Несколько аргументов
console.info('text1', 'text2')
console.info('text1', 'text2', ..., 'text10')

// Различные типы аргументов
console.info('text1', 123, true, { key: 'value' })
```



console.warn()

ОПИСАНИЕ

Metod console.info() выводит предупреждающее сообщение желтым цветом в консоль, каждое сообщение помечается иконкой warn. Данный метод используется для указания на потенциальные проблемы, которые не являются критичными, но должны быть устранены. Метод может принимать несколько аргументов и отображать их в одной строке.

СИНТАКСИС

```
console.warn(message1 : any, message2 : any, ..., messageN : any)
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefined

ПРИМЕР

```
// Один аргумент
console.warn('warn text')

// Несколько аргументов
console.warn('text1', 'text2')
console.warn('text1', 'text2', ..., 'text10')

// Mixed types
console.warn('text1', 123, true, { key: 'value' })
```



console.error()

ОПИСАНИЕ

Metod console.error() выводит сообщение об ошибке красным цветом в консоль, каждое сообщение помечается иконкой error. Он используется для указания на критические проблемы, которые требуют немедленного внимания. Он может принимать несколько аргументов и отображать их в одной строке.

СИНТАКСИС

```
console.error(message1 : any, message2 : any, ..., messageN : any)
```

АРГУМЕНТЫ

• messageN (апу, обязательно):

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefined

ПРИМЕР

```
// Один аргумент
console.error('critical text')

// Несколько аргументов
console.error('text1', 'text2')
console.error('text1', 'text2', ..., 'text10')

// Различные типы аргументов
console.error('text1', 123, true, { key: 'value' })
```



console.log()

console.info()

console.warn()

console.clear()

ОПИСАНИЕ

Metod console.clear() очищает вывод консоли, удаляя все ранее зарегистрированные сообщения. Это полезно для сброса консоли и обеспечения чистого рабочего пространства при отладке или записи новой информации.

СИНТАКСИС

console.clear()

АРГУМЕНТЫ

Без аргументов

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefined

ПРИМЕР

1 console.clear()

console.dir()

ОПИСАНИЕ

Metod console.dir() отображает список свойств и методов указанного JavaScript-объекта. Он особенно полезен для проверки структуры и содержимого объектов, так как форматирует вывод в виде списка, что облегчает исследование вложенных свойств.

СИНТАКСИС

```
console.dir(object : any)
```

АРГУМЕНТЫ

• object (апу, обязательно):

возвращаемое значение

undefined

ПРИМЕР

```
// Define an object
const obj = {
   name: 'MotorXP',
   version: 1.0,
   features: ['AFM Design', 'Electromagnetic Analysis', 'Optimization API'],
   details: {
      developer: 'MotorXP Team',
      license: 'Combo'
      }
   }
   // Inspect the object
   console.dir(obj)
```

Output

```
▶ Object
name: "MotorXP"
version: 1

▶ features: Array(3)
0: "AFM Design"
1: "Electromagnetic Analysis"
2: "Optimization API"

▶ details: Object
developer: "MotorXP Team"
license: "Combo"
```

4.7 motor

4.7.1 Обзор объекта motor

Описание

Встроенный объект motor представляет собой модель двигателя (генератора) с различными параметрами.

Константы объекта motor

- $\underline{\text{Motor.SR}}$ Тип мотора: Статор-Ротор.
- Значение: 0
- Motor.SRS Тип мотора: Статор-Ротор-Статор.
- Значение: 1
- Motor. SRSRS Тип мотора: Статор-Ротор-Статор-Ротор-Статор.
- Значение: 2
- Motor.RSR Тип мотора: Ротор-Статор-Ротор.
- Значение: 3
- Motor.RSRSR Тип мотора: Ротор-Статор-Ротор-Статор-Ротор.

Значение: 4

Свойства объекта motor

Ниже приведен список доступных свойств, предоставляемых объектом motor . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- machineType: Type of the machine.
- stator: Returns the Stator object associated with the motor.
- \bullet $\underline{\mathtt{rotor}}$: Returns the Rotor object associated with the motor.
- airgap: Returns the Airgap object associated with the motor.
- winding: Returns the Winding object associated with the motor.
- mesh : Returns the Mesh object associated with the motor.

4.7.2 Константы

Константы объекта motor

- Motor.SR Тип мотора: Статор-Ротор. Значение: 0
- <u>Motor.SRS</u> Тип мотора: Статор-Ротор-Статор. Значение: 1
- Motor.SRSRS Тип мотора: Статор-Ротор-Статор-Ротор-Статор. Значение: 2
- Motor.RSR Тип мотора: Ротор-Статор-Ротор. Значение: 3
- Motor.RSRSR Тип мотора: Ротор-Статор-Ротор-Статор-Ротор. Значение: 4

Motor.SR

ОПИСАНИЕ

Тип мотора: Статор-Ротор.

ЗНАЧЕНИЕ

0

ПРИМЕР

- 1 let t = Motor.SR
 2 console.info(t)
 - вязанные свойства

motor.machine Type

Motor.SRS

ОПИСАНИЕ

Тип мотора: Статор-Ротор-Статор.

ЗНАЧЕНИЕ

1

ПРИМЕР

- 1 let t = Motor.SRS
 2 console.info(t)
 - Вязанные свойства

motor.machine Type

Motor.SRSRS

ОПИСАНИЕ

Тип мотора: Статор-Ротор-Статор-Ротор-Статор.

ЗНАЧЕНИЕ

2

ПРИМЕР

```
1 let t = Motor.SRSRS
2 console.info(t)
```



motor.machine Type

Motor.RSR

ОПИСАНИЕ

Тип мотора: Ротор-Статор-Ротор.

ЗНАЧЕНИЕ

3

ПРИМЕР

- 1 let t = Motor.RSR
 2 console.info(t)
 - Вязанные свойства

motor.machine Type

Motor.RSRSR

ОПИСАНИЕ

Тип мотора: Ротор-Статор-Ротор-Статор-Ротор.

ЗНАЧЕНИЕ

4

ПРИМЕР

- 1 let t = Motor.RSRSR
 2 console.info(t)
 - вязанные свойства

motor.machineType

4.7.3 Свойства

Свойства объекта motor

- $\underline{\text{machineType}}$: Type of the machine.
- \bullet $\,\underline{\text{\tt stator}}$: Returns the $\,\text{\tt Stator}\,$ object associated with the motor.
- $\underline{\mathtt{rotor}}$: Returns the Rotor object associated with the motor.
- airgap: Returns the Airgap object associated with the motor.
- \bullet $\underline{\text{winding}}$: Returns the Winding object associated with the motor.
- \bullet $\underline{\text{mesh}}$: Returns the Mesh object associated with the motor.

machineType

ОПИСАНИЕ

The $\mbox{machineType}$ property defines the configuration type of the motor.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- Motor.SR Тип мотора: Статор-Ротор. Значение: 0
- Motor. SRS Тип мотора: Статор-Ротор-Статор. Значение: 1
- Motor.SRSRS Тип мотора: Статор-Ротор-Статор-Ротор-Статор. Значение: 2
- Motor.RSR Тип мотора: Ротор-Статор-Ротор. Значение: 3
- Motor.RSRSR Тип мотора: Ротор-Статор-Ротор-Статор-Ротор. Значение: 4

ТИП ЗНАЧЕНИЯ СВОЙСТВА

number

доступ

Чтение\Запись

СИНТАКСИС

```
motor.machineType = Motor.<TYPE>
```

ПРИМЕР

```
// Set the machine type to Stator-Rotor-Stator configuration
motor.machineType = Motor.SRS;

let type = motor.machineType
// Check the current machine type
console.log(motor.type) // Output: 1
```



stator

ОПИСАНИЕ

Возвращает объект типа Stator.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Stator

доступ

Только Чтение

СИНТАКСИС

motor.stator

ПРИМЕР

1 let stator = motor.stator



rotor

ОПИСАНИЕ

Возвращает объект типа Rotor.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Rotor

доступ

Только Чтение

СИНТАКСИС

motor.rotor

ПРИМЕР

1 let rotor = motor.rotor



airgap

ОПИСАНИЕ

Возвращает объект типа Airgap.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Airgap

доступ

Только Чтение

СИНТАКСИС

motor.airgap

ПРИМЕР

1 let airgap = motor.airgap



Airgap

winding

ОПИСАНИЕ

Возвращает объект типа $\mbox{Winding}$.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Winding

доступ

Только Чтение

СИНТАКСИС

motor.winding

ПРИМЕР

1 let winding = motor.winding



mesh

ОПИСАНИЕ

Возвращает объект типа Mesh.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Mesh

доступ

Только Чтение

СИНТАКСИС

motor.mesh

ПРИМЕР

1 let mesh = motor.mesh



5. Типы объектов

5.1 Встроенные типы

5.1.1 Части мотора

- Stator:
- StatorItem:
- Rotor:
- RotorItem:
- Airgap:
- Winding:
- Mesh:

5.1.2 Геометрия

- <u>Point3</u>:
- <u>Vector3</u>:
- Shape:
- BoundingBox:
- Piece:

5.1.3 Материалы

- EmptyMaterial:
- GeneralMaterial:
- IronMaterial:
- ConductorMaterial:
- WindingMaterial:
- EndturnMaterial:
- $\bullet \, \underline{Magnet Radial Material};\\$
- MagnetParallelMaterial:
- CustomMaterial:

5.1.4 UI Widgets

Qt Widgets

- QWidget:
- QLabel:
- QLineEdit:
- <u>QPushButton</u>:
- QSpinBox:
- QDoubleSpinBox:
- QComboBox:
- QGroupBox:

- QCheckBox:
- QGridLayout:
- QFormLayout:

Custom Widgets

- WarningIcon:
- ExclamationIcon:
- NumberEdit:
- $\bullet \, \underline{NumberSlotSpinBox} :$
- <u>StatorTypeComboBox</u>:
- $\bullet \underline{WindingLayersComboBox}:$
- $\bullet \underline{Winding Layers Orientation Combo Box} :$
- $\bullet \underline{WindingTypeComboBox}:$
- $\bullet \, \underline{PoleArrangementComboBox} :$
- <u>StatorConnectionComboBox</u>:
- $\bullet \, \underline{RotorConnectionComboBox} :$

5.2 Airgap

5.2.1 Описание Airgap

Встроенный тип Аігдар является составной частью мотора и описывает свойства воздушного зазора мотора.

Константы Аігдар

• <u>Airgap.ID</u>: Идентификатор воздушного зазора.

Свойства Аігдар

Ниже приведен список доступных свойств, предоставляемых типом Airgap . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- id: Идентификатор воздушного зазора.
- thickness: Толщина воздушного зазора.
- <u>numberLayers</u>: Количество слоев воздушного зазора для расчета сетки.
- posBottom : Позиция нижней стороны воздушного зазора.
- розтор: Позиция верхней стороны воздушного зазора.
- <u>posMiddle</u>: Позиция середины воздушного зазора.



motor.airgap

5.2.2 Константы

Константы Аігдар

• $\underline{\mathtt{Airgap.ID}}$: Идентификатор воздушного зазора.

Airgap.ID

ОПИСАНИЕ

Идентификатор воздушного зазора.

ЗНАЧЕНИЕ

- Airgap.Empty : 0
- Airgap.ID1:1
- Airgap.ID2 : 2
- Airgap.ID3: 3
- Airgap.ID4: 4

ПРИМЕР

```
1 let id = Airgap.ID1
2 console.info(id) // OUTPUT: 1
```



id

5.2.3 Свойства

Свойства Аігдар

Ниже приведен список доступных свойств, предоставляемых типом Airgap . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>id</u> : Идентификатор воздушного зазора.
- <u>thickness</u>: Толщина воздушного зазора.
- <u>numberLayers</u>: Количество слоев воздушного зазора для расчета сетки.
- posBottom : Позиция нижней стороны воздушного зазора.
- розтор : Позиция верхней стороны воздушного зазора.
- <u>posMiddle</u> : Позиция середины воздушного зазора.

id

ОПИСАНИЕ

Идентификатор воздушного зазора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Airgap.ID

возможные значения

- Airgap.ID1
- Airgap.ID2
- Airgap.ID3
- Airgap.ID4

доступ

Только Чтение

СИНТАКСИС

id

ПРИМЕР

```
let airgap = motor.airgap
let result = airgap.id
console.info(result) // output: 1
```



thickness

numberLayers

posTop

posMiddle

thickness

ОПИСАНИЕ

Толщина воздушного зазора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

thickness

ПРИМЕР

- 1 let airgap = motor.airgap
 2 let t = airgap.thickness
 3 console.info(t) // output:



id

numberLayers

posBottom

posTop

posMiddle

numberLayers

ОПИСАНИЕ

Количество слоев воздушного зазора для расчета сетки.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberLayers

ПРИМЕР

```
let airgap = motor.airgap
let result = airgap.numberLayers
console.info(result) // output:
```



id

thickness

posBottom

posTop

posMiddle

posBottom

ОПИСАНИЕ

Позиция нижней стороны воздушного зазора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Только Чтение

СИНТАКСИС

posBottom

ПРИМЕР

```
let airgap = motor.airgap
let result = airgap.posBottom
console.info(result) // output:
```



id

thickness

numberLayers

posTop

posMiddle

posTop

ОПИСАНИЕ

Позиция верхней стороны воздушного зазора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Только Чтение

СИНТАКСИС

posTop

ПРИМЕР

```
let airgap = motor.airgap
let result = airgap.posTop
console.info(result) // output:
```



id

thickness

number Layers

posBottom

posMiddle

posMiddle

ОПИСАНИЕ

Позиция центра воздушного зазора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Только Чтение

СИНТАКСИС

posBottom

ПРИМЕР

```
let airgap = motor.airgap
let result = airgap.posMiddle
console.info(result) // output:
```



id

thickness

numberLayers

posBottom

posTop

5.2.4 Методы

Методы Airgap

Нет методов.

5.3 BoundingBox

5.3.1 Описание BoundingBox

Описывает ограничивающий ящик в трехмерном пространстве. Ограничивающий ящик параллелен осям системы координат.интервалами:

Свойства BoundingBox

Ниже приведен список доступных свойств, предоставляемых типом BoundingBox . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- $\underline{\mathtt{xMin}}$: Минимальное значение по оси X.
- $_{\underline{\times}\underline{\mathsf{Max}}}$: Максимальное значение по оси X.
- $\underline{\mathtt{xSize}}$: Размер по оси X.
- xCenter : Середина по оси X.
- уміп : Минимальное значение по оси Y.
- умах : Максимальное значение по оси Ү.
- $\underline{\mathtt{ySize}}$: Размер по оси Y.
- <u>yCenter</u> : Середина по оси Y.
- $\underline{\mathtt{zMin}}$: Минимальное значение по оси Z.
- дмах : Максимальное значение по оси Z.
- $\underline{\mathtt{zSize}}$: Размер по оси Z.
- \bullet **zCenter** : Середина по оси Z.

Методы BoundingBox

Нет методов

5.3.2 Свойства

Свойства BoundingBox

Ниже приведен список доступных свойств, предоставляемых типом BoundingBox . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- $\underline{\text{xMin}}$: Минимальное значение по оси X.
- $_{\underline{\times}\underline{\text{Max}}}$: Максимальное значение по оси X.
- $\underline{\mathtt{xSize}}$: Размер по оси X.
- \bullet <u>xCenter</u> : Середина по оси X.
- уміп : Минимальное значение по оси Ү.
- умах : Максимальное значение по оси Ү.
- уѕіде : Размер по оси Ү.
- <u>yCenter</u> : Середина по оси Y.
- $\underline{\mathtt{zMin}}$: Минимальное значение по оси Z.
- дмах : Максимальное значение по оси Z.
- $\underline{\mathtt{zSize}}$: Размер по оси Z.
- zCenter : Середина по оси Z.

xMin

ОПИСАНИЕ

Минимальное значение по оси X.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

xMin

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let mx = bbox.xMin
console.info(mx) // output:
```



- xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSize
- zCenter

xMax

ОПИСАНИЕ

Максимальное значение по оси Х.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

хМах

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let mx = bbox.xMax
console.info(mx) // output:
```



- xMin
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSize
- zCenter

xSize

ОПИСАНИЕ

Размер по оси X.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

xSize

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let sx = bbox.xSize
console.info(sx) // output:
```



- xMin
- \bullet xMax
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSize
- zCenter

xCenter

ОПИСАНИЕ

Середина по оси Х.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

xCenter

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let cx = bbox.xCenter
console.info(cx) // output:
```



- xMin
- \bullet xMax
- xSize
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSize
- zCenter

yMin

ОПИСАНИЕ

Минимальное значение по оси Ү.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

yMin

ПРИМЕР

```
1 let cyl = Geom.cylinder(100, 50)
2 let bbox = cyl.boundBox()
3 let my = bbox.yMin
4 console.info(my) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSizezCenter

yMax

ОПИСАНИЕ

Максимальное значение по оси Ү.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

yMax

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let my = bbox.yMax
console.info(my) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- ySize
- yCenter
- zMin
- zMax
- zSizezCenter

ySize

ОПИСАНИЕ

Размер по оси Ү.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

ySize

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let sy = bbox.ySize
console.info(sy) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- yMax
- yCenter
- zMin
- zMax
- zSize

• zCenter

yCenter

ОПИСАНИЕ

Середина по оси Ү.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

yCenter

ПРИМЕР

```
1 let cyl = Geom.cylinder(100, 50)
2 let bbox = cyl.boundBox()
3 let cy = bbox.yCenter
4 console.info(cy) // output:
```



- xMin
- xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- \bullet zMin
- zMax
- zSizezCenter

zMin

ОПИСАНИЕ

Минимальное значение по оси Z.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

zMin

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let mz = bbox.zMin
console.info(mz) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMax
- zSize
- zCenter

zMax

ОПИСАНИЕ

Максимальное значение по оси Z.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

zMax

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let mz = bbox.zMax
console.info(mz) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zSize
- zCenter

zSize

ОПИСАНИЕ

Размер по оси Z.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

zSize

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let sz = bbox.zSize
console.info(sz) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zCenter

zCenter

ОПИСАНИЕ

Середина по оси Z.

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

доступ

Только Чтение

СИНТАКСИС

zCenter

ПРИМЕР

```
let cyl = Geom.cylinder(100, 50)
let bbox = cyl.boundBox()
let cz = bbox.zCenter
console.info(cz) // output:
```



- xMin
- \bullet xMax
- xSize
- xCenter
- yMin
- yMax
- ySize
- yCenter
- zMin
- zMax
- zSize

5.3.3 Методы

Методы BoundingBox

Нет методов.

5.4 Stator

5.4.1 Описание Stator

Встроенный тип Stator является составной частью мотора и описывает свойства статора мотора.

Константы Stator

• Stator.MiddleТуре: Тип средней части статора.

Свойства Stator

Ниже приведен список доступных свойств, предоставляемых типом Stator . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- outerDiameter: Внешний диаметр.
- outerRadius : Внешний радиус.
- innerDiameter: Внутрений диаметр.
- innerRadius : Внутрений радиус.
- numberSlots: Количество пазов.
- slotAngleSpan : Угловой размер одного паза статора в градусах.
- typeMiddleItem: Тип средней части статора.
- countItems: Количество элементов статора.
- items: Массив элементов статора.
- <u>nameScript</u>: Имя скрипта.
- script: Имя файла скрипта.
- <u>ironMaterial</u>: Материал стали статора.
- ironStacking: Коеффициент укладки желез статора.
- windingMaterial: Материал обмотки статора.
- windingTemperature: Температура обмотки статора.
- conductorMaterial: Материал проводника статора.
- <u>conductorTemperature</u>: Температура порводника статора.

Методы Stator

Ниже приведен список доступных методов, предоставляемых типом Stator . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- isLower: Истина, если в моторе элемент статора нижний.
- isMiddle: Истина, если в моторе элемент статора средний.
- isUpper: Истина, если в моторе элемент статора верхний.
- <u>item</u>: Элемент статора.
- <u>itemAngularDisplacement</u>: Угловое смещение элемента статора.
- <u>setItemAngularDisplacement</u>: Задать угловое смещение элемента статора.



motor.stator

5.4.2 Константы

Константы Stator

• $\underline{\mathtt{Stator}.\mathtt{MiddleType}}$: Тип средней части статора.

Stator.MiddleType

ОПИСАНИЕ

Определяет тип средней части статора: с ярмом (Yoke) или без ярма (Yokeless).

ЗНАЧЕНИЕ

- Stator. Yoke: Средняя часть статора с ярмом.
- Stator. Yokeless: Средняя часть статора без ярма.

ПРИМЕР

```
1 let value = Stator.Yoke;
2 console.info(value);
```



middleType

5.4.3 Свойства

Свойства Stator

Ниже приведен список доступных свойств, предоставляемых типом Stator . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- outerDiameter: Внешний диаметр.
- outerRadius: Внешний радиус.
- <u>innerDiameter</u>: Внутрений диаметр.
- innerRadius: Внутрений радиус.
- <u>numberSlots</u>: Количество слотов.
- slotAngleSpan: Угловой размер одного паза статора в градусах.
- typeMiddleItem: Тип средней части статора
- countItems: Количество элементов статора
- items: Массив элементов статора
- nameScript: Имя скрипта
- | script : Имя файла скрипта
- <u>ironMaterial</u>: Материал стали статора
- <u>ironStacking</u>: Коеффициент укладки желез статора.
- windingMaterial: Материал обмотки статора.
- windingTemperature: Температура обмотки статора.
- conductorMaterial: Материал проводника статора.
- <u>conductorTemperature</u>: Температура порводника статора.

outerDiameter

ОПИСАНИЕ

Внешний диаметр.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

outerDiameter = value

ПРИМЕР

- let result = motor.outerDiameter
 console.info(result)



outerRadius

ОПИСАНИЕ

Внешний радиус

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

outerRadius = value

ПРИМЕР

let result = motor.outerRadius
console.info(result)



innerDiameter

ОПИСАНИЕ

Внутрений диаметр

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

innerDiameter = value

ПРИМЕР

- let result = motor.stator.innerDiameter
 console.info(result)



innerRadius

ОПИСАНИЕ

Внутрений радиус.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

innerRadius = value

ПРИМЕР

let result = motor.stator.innerRadius
console.info(result)



numberSlots

ОПИСАНИЕ

Количество пазов.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число из ряда значений: 3,6,9,12,15,...501.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberSlots = value

ПРИМЕР

let result = motor.stator.numberSlots
console.info(result)



slotAngleSpan

ОПИСАНИЕ

Угловой размер одного паза статора в градусах.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Тоько Чтение

СИНТАКСИС

slotAngleSpan

ПРИМЕР

- 1 let result = motor.stator.slotAngleSpan
 2 console.info(result)



typeMiddleItem

ОПИСАНИЕ

Тип средней части статора

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из зачений: - Stator.Yoke - Stator.Yokeless

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Stator.MiddleType

доступ

Чтение\Запись

СИНТАКСИС

typeMiddleItem = value

ПРИМЕР

- let result = motor.stator.typeMiddleItem
 console.info(result)



script

ОПИСАНИЕ

Имя файла скрипта

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Тоько Чтение

СИНТАКСИС

script

ПРИМЕР

- 1 let result = motor.stator.script
 2 console.info(result)



nameScript

ОПИСАНИЕ

Имя скрипта

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Тоько Чтение

СИНТАКСИС

nameScript

ПРИМЕР

- 1 let result = motor.stator.nameScript
 2 console.info(result)



countItems

ОПИСАНИЕ

Количество элементов статора, зависит от motor.machineType.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

number

доступ

Только Чтение

СИНТАКСИС

countItems

ПРИМЕР

- 1 let result = motor.stator.countItems
 2 console.info(result)



items

ОПИСАНИЕ

Массив элементов статора, количество элементов зависит от motor.machineType.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Array of StatorItem

доступ

Только Чтение

СИНТАКСИС

items

ПРИМЕР

- let result = motor.stator.items
 console.info(result)



ironMaterial

ОПИСАНИЕ

Материал стали статора

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

ironMaterial = value

ПРИМЕР

```
let result = motor.stator.ironMaterial
console.info(result)
```



ironStacking

ОПИСАНИЕ

Коеффициент укладки желез статора.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число в диапазоне от 0 до 1.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

ironStacking = value

ПРИМЕР

let result = motor.stator.ironStacking
console.info(result)



windingMaterial

ОПИСАНИЕ

Материал обмотки статора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

windingMaterial = value

ПРИМЕР

- 1 let result = motor.stator.windingMaterial
 2 console.info(result)



windingTemperature

ОПИСАНИЕ

Температура обмотки статора в градусах Цельсия.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

windingTemperature = value

ПРИМЕР

- 1 let result = motor.windingTemperature
 2 console.info(result)



conductorMaterial

ОПИСАНИЕ

Материал проводника статора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

conductorMaterial = value

ПРИМЕР

- 1 let result = motor.stator.conductorMaterial
 2 console.info(result)



conductorTemperature

ОПИСАНИЕ

Температура проводника статора в градусах Цельсия.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

conductorTemperature = value

ПРИМЕР

- 1 let result = motor.stator.conductorTemperature
 2 console.info(result)



5.4.4 Методы

Методы Stator

Ниже приведен список доступных методов, предоставляемых типом Stator . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>islower</u>: Истина, если в моторе элемент статора нижний..
- <u>isMiddle</u>: Истина, если в моторе элемент статора средний.
- <u>isUpper</u>: Истина, если в моторе элемент статора верхний.
- <u>item</u>: Элемент статора.
- $\underline{\text{itemAngularDisplacement}}$: Угловое смещение элемента статора.
- <u>setItemAngularDisplacement</u>: Задать угловое смещение элемента статора.

isLower()

ОПИСАНИЕ

Истина, если в моторе элемент статора нижний.

СИНТАКСИС

```
isLower(itemID: StatorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент статора нижний.

ПРИМЕР

```
let stator = motor.stator
let result = stator.isLower(StatorItem.ID1)
console.info(result)
```



isMiddle()

ОПИСАНИЕ

Истина, если в моторе элемент статора средний.

СИНТАКСИС

```
isLower(itemID: StatorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

ьоо1: Истина, если в моторе элемент статора средний.

ПРИМЕР

```
let stator = motor.stator
let result = stator.isMiddle(StatorItem.ID1)
console.info(result)
```



isUpper()

ОПИСАНИЕ

Истина, если в моторе элемент статора верхний.

СИНТАКСИС

```
isLower(itemID: StatorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент статора верхний.

ПРИМЕР

```
let stator = motor.stator
let result = stator.isUpper(StatorItem.ID1)
console.info(result)
```



item()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
item(itemID: StatorItem.ID) : StatorItem
```

АРГУМЕНТЫ

• itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

StatorItem: Объект типа StatorItem.

ПРИМЕР

```
let stator = motor.stator
let result = stator.item(StatorItem.ID1)
console.info(result)
```



itemAngularDisplacement()

ОПИСАНИЕ

Угловое смещение элемента статора в градусах.

СИНТАКСИС

```
itemAngularDisplacement(itemID: StatorItem.ID) : number
```

АРГУМЕНТЫ

• itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Угловое смещение элемента статора в градусах.

ПРИМЕР

```
let stator = motor.stator
let result = stator.itemAngularDisplacement(StatorItem.ID1)
console.info(result)
```



setItemAngularDisplacement()

ОПИСАНИЕ

Задать угловое смещение элемента статора в градусах.

СИНТАКСИС

```
setItemAngularDisplacement(itemID: StatorItem.ID, angle: number)
```

АРГУМЕНТЫ

- itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.
- angle (number, обязательно): Угловое смещение элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefine

ПРИМЕР

```
1 let stator = motor.stator
2 stator.setItemAngularDisplacement(StatorItem.ID1, 30)
```



5.5 StatorItem

5.5.1 Описание StatorItem

Встроенный тип StatorItem является составной частью статора и описывает свойства элемента статора мотора.

5.5.2 Константы StatorItem

- StatorItem.ID: Идентификатор элемента статора.
- StatorItem.Layer: Идентификатор слоя элемента статора.

Свойства StatorItem

Ниже приведен список доступных свойств, предоставляемых типом StatorItem. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>id</u>: Идентификатор элемента статора.
- <u>height</u>: Высота элемента статора.
- angularDisplacement: Угловое смещение элемента статора.

Mетоды StatorItem

Ниже приведен список доступных методов, предоставляемых типом Stator . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- isLower: Истина, если в моторе элемент статора нижний.
- <u>isMiddle</u>: Истина, если в моторе элемент статора средний.
- <u>isUpper</u>: Истина, если в моторе элемент статора верхний.



motor.stator

5.5.3 Константы

Константы StatorItem

- $\underline{\mathtt{StatorItem.ID}}$: Идентификатор элемента статора.
- <u>StatorItem.Layer</u>: Слой элемента статора.

StatorItem.ID

ОПИСАНИЕ

Идентификатор элемента статора.

ЗНАЧЕНИЕ

- StatorItem.ID1:1
- StatorItem.ID2:2
- StatorItem.ID3:3

ПРИМЕР

- let value = StatorItem.Lower;
 console.info(value);



StatorItem.Layer

ОПИСАНИЕ

Слой элемента статора.

ЗНАЧЕНИЕ

- StatorItem.Lower: 1
- StatorItem.Upper:2

ПРИМЕР

```
1 let value = StatorItem.ID1;
2 console.info(value);
```



ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

```
let value = Constants.ID2;
console.info(value);
```



5.5.4 Свойства

Свойства типа StatorItem

Ниже приведен список доступных свойств, предоставляемых типом StatorItem. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>id</u>: Идентификатор элемента статора.
- height: Высота элемента статора.
- \bullet <u>angularDisplacement</u>: Угловое смещение элемента статора.

id

ОПИСАНИЕ

Идентификатор элемента статора

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из констант:

- StatorItem.ID1
- StatorItem.ID2
- StatorItem.ID3

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Только Чтение

СИНТАКСИС

id = value

ПРИМЕР

- 1 let result = motor.stator.item(StatorItem.ID1).id
 2 console.info(result)



StatorItem.ID

height

ОПИСАНИЕ

Высота элемента.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Положительное число

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

height = value

ПРИМЕР

```
let result = motor.stator.item(StatorItem.ID1).height
console.info(result)
```



angularDisplacement

ОПИСАНИЕ

Угловое смещение элемента.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

```
angularDisplacement = value
```

ПРИМЕР

```
1 let result = motor.stator.item(StatorItem.ID1).angularDisplacement
2 console.info(result)
```



()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
angularDisplacement(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

returnТуре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = angularDisplacement(value)
2 console.info(result)
```



5.5.5 Методы

Методы объекта StatorItem

Ниже приведен список доступных методов, предоставляемых объектом StatorItem. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>islower</u>: Истина, если в моторе элемент статора нижний.
- <u>isMiddle</u>: Истина, если в моторе элемент статора средний.
- <u>isUpper</u>: Истина, если в моторе элемент статора верхний.

isUpper()

ОПИСАНИЕ

Истина, если в моторе элемент статора верхний.

СИНТАКСИС

```
isUpper() : bool
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент статора верхний.

ПРИМЕР

```
1 let result = motor.stator.item(StatorItem.ID1).isUpper(value)
2 console.info(result)
```



isMiddle()

ОПИСАНИЕ

Истина, если в моторе элемент статора средний.

СИНТАКСИС

```
isMiddle() : bool
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент статора средний.

ПРИМЕР

```
let result = motor.stator.item(StatorItem.ID1).isMiddle()
console.info(result)
```



isLower()

ОПИСАНИЕ

Истина, если в моторе элемент статора нижний.

СИНТАКСИС

```
isLower() : bool
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент статора нижний.

ПРИМЕР

```
let result = motor.stator.item(StatorItem.ID1).isLower()
console.info(result)
```



5.6 Rotor

5.6.1 Описание Rotor

Встроенный тип Rotor является составной частью мотора и описывает свойства ротора мотора.

Константы Rotor

• Rotor.MiddleТype: Тип средней части ротора.

Свойства Rotor

Ниже приведен список доступных свойств, предоставляемых типом Rotor. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- outerDiameter: Внешний диаметр.
- outerRadius: Внешний радиус.
- <u>innerDiameter</u>: Внутрений диаметр.
- <u>innerRadius</u>: Внутрений радиус.
- numberPolePairs : Количество пар полюсов.
- poleAngleSpan : Угловой размер одного полюса ротора в градусах.
- poleArrangement : Расположение полюсов.
- typeMiddleItem: Тип средней части ротора.
- countItems: Количество элементов ротора.
- items: Массив элементов ротора.
- nameScript : Имя скрипта.
- <u>script</u>: Имя файла скрипта.
- ironMaterial: Материал стали ротора.
- ironStacking: Коеффициент укладки желез ротора.
- magnetMaterial: Материал магнита ротора.
- magnetTemperature: Температура магнита ротора.
- conductorMaterial: Материал проводника ротора.
- <u>conductorTemperature</u>: Температура проводника ротора.

Методы Rotor

Ниже приведен список доступных методов, предоставляемых типом Rotor . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- isLower: Истина, если в моторе элемент ротора нижний.
- isMiddle: Истина, если в моторе элемент ротора средний.
- <u>isUpper</u>: Истина, если в моторе элемент ротора верхний.
- <u>item</u>: Элемент ротора.
- <u>itemAngularDisplacement</u>: Угловое смещение элемента ротора.
- setItemAngularDisplacement: Задать угловое смещение элемента ротора.



motor.rotor

5.6.2 Константы

Константы Rotor

• Rotor.MiddleType: Тип средней части ротора.

Rotor.MiddleType

ОПИСАНИЕ

Определяет тип средней части ротора: с ярмом (Yoke) или без ярма (Yokeless).

ЗНАЧЕНИЕ

- Rotor. Yoke: Средняя часть ротора с ярмом.
- Rotor. Yokeless: Средняя часть ротора без ярма.

ПРИМЕР

```
1 let value = Rotor.Yoke;
2 console.info(value);
```



middleType

5.6.3 Свойства

Свойства Rotor

Ниже приведен список доступных свойств, предоставляемых типом Rotor . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- outerDiameter: Внешний диаметр.
- outerRadius: Внешний радиус.
- <u>innerDiameter</u>: Внутрений диаметр.
- innerRadius: Внутрений радиус.
- <u>numberPolePairs</u>: Количество пар полюсов.
- poleAngleSpan : Угловой размер одного полюса ротора в градусах.
- poleArrangement : Расположение полюсов.
- typeMiddleItem: Тип средней части ротора.
- countItems: Количество элементов ротора.
- items: Массив элементов ротора.
- nameScript : Имя скрипта.
- <u>script</u>: Имя файла скрипта.
- <u>ironMaterial</u>: Материал стали ротора.
- ironStacking: Коеффициент укладки желез ротора.
- magnetMaterial: Материал магнита ротора.
- <u>magnetTemperature</u>: Температура магнита ротора.
- conductorMaterial: Материал проводника ротора.
- <u>conductorTemperature</u>: Температура проводника ротора.

outerDiameter

ОПИСАНИЕ

Внешний диаметр.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

outerDiameter = value

ПРИМЕР

- let result = motor.rotor.outerDiameter
 console.info(result)



outerRadius

ОПИСАНИЕ

Внешний радиус.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

outerRadius = value

ПРИМЕР

- let result = motor.rotor.outerRadius
 console.info(result)



innerDiameter

ОПИСАНИЕ

Внутрений диаметр

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

innerDiameter = value

ПРИМЕР

- let result = motor.rotor.innerDiameter
 console.info(result)



innerRadius

ОПИСАНИЕ

Внутрений радиус

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число больше нуля.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

innerRadius = value

ПРИМЕР

- let result = motor.rotor.innerRadius
 console.info(result)



numberPolePairs

ОПИСАНИЕ

Количество пар полюсов.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Положительное число.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberSlots = value

ПРИМЕР

- let result = motor.rotor.numberPolePairs
 console.info(result)



poleAngleSpan

ОПИСАНИЕ

Угловой размер одного полюса ротора в градусах.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Тоько Чтение

СИНТАКСИС

slotAngleSpan

ПРИМЕР

- 1 let result = motor.rotor.slotAngleSpan
 2 console.info(result)



poleArrangement

ОПИСАНИЕ

Расположение полюсов

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из значения - PoleArrangement.NN - PoleArrangement.NS - PoleArrangement.NSNS - PoleArrangement.NSSN - PoleAr PoleArrangement.NNNN

ТИП ЗНАЧЕНИЯ СВОЙСТВА

PoleArrangement

доступ

Чтение\Запись

СИНТАКСИС

poleArrangement = value

ПРИМЕР

- 1 let result = motor.rotor.poleArrangement
 2 console.info(result)



typeMiddleItem

ОПИСАНИЕ

Тип средней части ротора

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из зачений: - Rotor.Yoke - Rotor.Yokeless

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Rotor.MiddleType

доступ

Чтение\Запись

СИНТАКСИС

typeMiddleItem = value

ПРИМЕР

- let result = motor.rotor.typeMiddleItem
 console.info(result)



script

ОПИСАНИЕ

Имя файла скрипта

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Тоько Чтение

СИНТАКСИС

script

ПРИМЕР

- 1 let result = motor.rotor.script
 2 console.info(result)



nameScript

ОПИСАНИЕ

Имя скрипта

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Тоько Чтение

СИНТАКСИС

nameScript

ПРИМЕР

- 1 let result = motor.rotor.nameScript
 2 console.info(result)



countItems

ОПИСАНИЕ

Количество элементов ротора, зависит от motor.machineType.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

number

доступ

Только Чтение

СИНТАКСИС

countItems

ПРИМЕР

- 1 let result = motor.rotor.countItems
 2 console.info(result)



items

ОПИСАНИЕ

Массив элементов ротора, количество элементов зависит от motor.machineType.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Array of StatorItem

доступ

Только Чтение

СИНТАКСИС

items

ПРИМЕР

- let result = motor.rotor.items
 console.info(result)



ironStacking

ОПИСАНИЕ

Коеффициент укладки железа ротора.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Число в диапазоне от 0 до 1.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

ironStacking = value

ПРИМЕР

- let result = motor.rotor.ironStacking
 console.info(result)



ironMaterial

ОПИСАНИЕ

Материал стали ротора

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

ironMaterial = value

ПРИМЕР

let result = motor.rotor.ironMaterial
console.info(result)



magnetTemperature

ОПИСАНИЕ

Температура магнита ротора в градусах Цельсия.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

conductorTemperature = value

ПРИМЕР

- let result = motor.rotor.conductorTemperature
 console.info(result)



magnetMaterial

ОПИСАНИЕ

Материал магнита ротора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

windingMaterial = value

ПРИМЕР

- 1 let result = motor.rotor.windingMaterial
 2 console.info(result)



conductorTemperature

ОПИСАНИЕ

Температура проводника ротора в градусах Цельсия.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

conductorTemperature = value

ПРИМЕР

- let result = motor.rotor.conductorTemperature
 console.info(result)



conductorMaterial

ОПИСАНИЕ

Материал проводника ротора.

ТИП ЗНАЧЕНИЯ СВОЙСТВА

String

доступ

Чтение\Запись

СИНТАКСИС

conductorMaterial = value

ПРИМЕР

- 1 let result = motor.rotor.conductorMaterial
 2 console.info(result)



5.6.4 Методы

Методы Rotor

Ниже приведен список доступных методов, предоставляемых типом Rotor . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>islower</u>: Истина, если в моторе элемент ротора нижний.
- <u>isMiddle</u>: Истина, если в моторе элемент ротора средний.
- <u>isUpper</u>: Истина, если в моторе элемент ротора верхний.
- <u>item</u>: Элемент ротора.
- $\underline{\text{itemAngularDisplacement}}$: Угловое смещение элемента ротора.
- <u>setItemAngularDisplacement</u>: Задать угловое смещение элемента ротора.

Last update: 3 мая 2025, 11:59

item()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
item(itemID: RotorItem.ID) : RotorItem
```

АРГУМЕНТЫ

• itemID (RotorItem.ID, обязательно): Идентификатор элемента ротора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

RotorItem: Объект типа RotorItem.

ПРИМЕР

```
let rotor = motor.rotor
let result = rotor.item(RotorItem.ID1)
console.info(result)
```



isLower()

ОПИСАНИЕ

Истина, если в моторе элемент ротора нижний.

СИНТАКСИС

```
isLower(itemID: RotorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (RotorItem.ID, обязательно): Идентификатор элемента ротора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент ротора нижний.

ПРИМЕР

```
let rotor = motor.rotor
let result = rotor.isLower(RotorItem.ID1)
console.info(result)
```



isMiddle()

ОПИСАНИЕ

Истина, если в моторе элемент ротора средний.

СИНТАКСИС

```
isLower(itemID: RotorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (RotorItem.ID, обязательно): Идентификатор элемента ротора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

bool: Истина, если в моторе элемент ротора средний.

ПРИМЕР

```
let rotor = motor.rotor
let result = rotor.isMiddle(RotorItem.ID1)
console.info(result)
```



isUpper()

ОПИСАНИЕ

Истина, если в моторе элемент ротора верхний.

СИНТАКСИС

```
isLower(itemID: RotorItem.ID) : bool
```

АРГУМЕНТЫ

• itemID (RotorItem.ID, обязательно): Идентификатор элемента ротора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

ьоо1: Истина, если в моторе элемент ротора верхний.

ПРИМЕР

```
let rotor = motor.rotor
let result = rotor.isUpper(RotorItem.ID1)
console.info(result)
```



itemAngularDisplacement()

ОПИСАНИЕ

Угловое смещение элемента ротора в градусах.

СИНТАКСИС

```
itemAngularDisplacement(itemID: RotorItem.ID) : number
```

АРГУМЕНТЫ

• itemID (RotorItem.ID, обязательно): Идентификатор элемента ротора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

number: Угловое смещение элемента ротора в градусах.

ПРИМЕР

```
1 let rotor = motor.rotor
2 let result = rotor.itemAngularDisplacement(RotorItem.ID1)
3 console.info(result)
```



setItemAngularDisplacement()

ОПИСАНИЕ

Задать угловое смещение элемента статора в градусах.

СИНТАКСИС

```
setItemAngularDisplacement(itemID: StatorItem.ID, angle: number)
```

АРГУМЕНТЫ

- itemID (StatorItem.ID, обязательно): Идентификатор элемента статора.
- angle (number, обязательно): Угловое смещение элемента статора.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

undefine

ПРИМЕР

```
1 let stator = motor.stator
2 stator.setItemAngularDisplacement(StatorItem.ID1, 30)
```



5.7 RotorItem

5.7.1 Описание RotorItem

Встроенный тип RotorItem является составной частью ротора и описывает свойства элемента ротора мотора.

5.7.2 Константы RotorItem

- RotorItem.ID: Идентификатор элемента ротора.
- RotorItem. Layer: Идентификатор слоя элемента ротора.

Свойства RotorItem

Ниже приведен список доступных свойств, предоставляемых типом RotorItem. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

• іd: Идентификатор элемента ротора.

Методы RotorItem

Ниже приведен список доступных методов, предоставляемых типом Rotor . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>isLower</u>: Истина, если в моторе элемент ротора нижний.
- isMiddle: Истина, если в моторе элемент ротора средний.
- isUpper: Истина, если в моторе элемент ротора верхний.



motor.rotor

Last update: 3 мая 2025, 11:59

5.7.3 Константы

Константы RotorItem

- $\[\underline{\text{RotorItem.ID}}\]$: Идентификатор элемента ротора.
- $\underline{\mathtt{RotorItem, Layer}}$: Идентификатор слоя элемента ротора.

Last update: 3 мая 2025, 11:59

ID

ОПИСАНИЕ

Идентификатор элемента ротора.

ЗНАЧЕНИЕ

- RotorItem.ID1:1
- RotorItem.ID2:2
- RotorItem.ID2:3

ПРИМЕР

- 1 let value = RotorItem.ID1;
 2 console.info(value);
 - 🔝. также

RotorItem.Layer

ОПИСАНИЕ

Слой элемента ротора.

ЗНАЧЕНИЕ

- RotorItem.Lower: 1
- RotorItem.Upper: 2

ПРИМЕР

- 1 let value = RotorItem.Lower;
 2 console.info(value);
 - 🗓. также

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- let value = Constants.ID2;
 console.info(value);
 - 🤼 также

5.7.4 Свойства

Свойства типа RotorItem

Ниже приведен список доступных свойств, предоставляемых типом RotorItem . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>id</u>: Идентификатор элемента ротора.
- height: Высота элемента ротора.
- <u>angularDisplacement</u>: Угловое смещение элемента ротора.

Last update: 3 мая 2025, 11:59

id

ОПИСАНИЕ

Идентификатор элемента ротора.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из констант:

- RotorItem.ID1
- RotorItem.ID2
- RotorItem.ID3

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Только Чтение

СИНТАКСИС

id

ПРИМЕР

let result = motor.rotor.item(RotorItem.ID1).id
console.info(result)



StatorItem.ID

height

ОПИСАНИЕ

Высота элемента.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Положительное число

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

height = value

ПРИМЕР

- let result = motor.rotor.item(RotorItem.ID1).height
 console.info(result)



angularDisplacement

ОПИСАНИЕ

The angularDisplacement property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

angularDisplacement = value

ПРИМЕР

let result = motor.angularDisplacement
console.info(result)



5.7.5 Методы

Методы RotorItem

Ниже приведен список доступных методов, предоставляемых типом Rotor . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- $\underline{\mathtt{isLower}}$: Истина, если в моторе элемент ротора нижний.
- <u>isMiddle</u>: Истина, если в моторе элемент ротора средний.
- <u>isUpper</u>: Истина, если в моторе элемент ротора верхний.

Last update: 3 мая 2025, 11:59

isUpper()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isUpper(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = isUpper(value)
 2 console.info(result)
 - 🤼. также

isMiddle()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isMiddle(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = isMiddle(value)
 2 console.info(result)
 - 🤼. также

isLower()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isLower(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = isLower(value)
 2 console.info(result)
 - ім. также

5.8 Winding

5.8.1 Описание

Last update: 3 мая 2025, 11:59

5.8.2 Константы

Last update: 3 мая 2025, 11:59

Planar

ОПИСАНИЕ

Константа Planar.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.Planar;
 2 console.info(value);



Toroidal

ОПИСАНИЕ

Константа Toroidal.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

```
1 let value = Constants.Toroidal;
2 console.info(value);
```



SingleLayer

ОПИСАНИЕ

Kонстанта SingleLayer.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.SingleLayer;
 2 console.info(value);



DoubleLayer

ОПИСАНИЕ

Константа DoubleLayer.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.DoubleLayer;
 2 console.info(value);



UpperLower

ОПИСАНИЕ

Kонстанта UpperLower.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.UpperLower;
 2 console.info(value);



LeftRight

ОПИСАНИЕ

Константа LeftRight.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.LeftRight;
 2 console.info(value);



Star

ОПИСАНИЕ

Константа Star.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.Star;
 2 console.info(value);



Delta

ОПИСАНИЕ

Константа Delta.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.Delta;
 2 console.info(value);



Lumped

ОПИСАНИЕ

Kонстанта Lumped.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

- 1 let value = Constants.Lumped;
 2 console.info(value);



Full

ОПИСАНИЕ

Константа Full.

ЗНАЧЕНИЕ

Number: значение константы.

ПРИМЕР

```
1 let value = Constants.Full;
2 console.info(value);
```



5.8.3 Свойства

Свойства типа Winding

Ниже приведен список доступных свойств, предоставляемых типом Winding . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Last update: 3 мая 2025, 11:59

circuit

ОПИСАНИЕ

The circuit property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

circuit = value

ПРИМЕР

```
1 let result = motor.circuit
2 console.info(result)
```



statorConnection

ОПИСАНИЕ

The statorConnection property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

statorConnection = value

ПРИМЕР

let result = motor.statorConnection
console.info(result)



parallelPaths

ОПИСАНИЕ

The parallelPaths property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

parallelPaths = value

ПРИМЕР

```
let result = motor.parallelPaths
console.info(result)
```



numberTurns

ОПИСАНИЕ

The numberTurns property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberTurns = value

ПРИМЕР

let result = motor.numberTurns
console.info(result)



strandsConductor

ОПИСАНИЕ

The strandsConductor property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

strandsConductor = value

ПРИМЕР

- let result = motor.strandsConductor
 console.info(result)



numberOuterSegments

ОПИСАНИЕ

The numberOuterSegments property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberOuterSegments = value

ПРИМЕР

let result = motor.numberOuterSegments
console.info(result)



type

ОПИСАНИЕ

The type property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

type = value

ПРИМЕР

1 let result = motor.type
2 console.info(result)



numberLayers

ОПИСАНИЕ

The numberLayers property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberLayers = value

ПРИМЕР

let result = motor.numberLayers
console.info(result)



layersOrientation

ОПИСАНИЕ

The layersOrientation property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

layersOrientation = value

ПРИМЕР

let result = motor.layersOrientation
console.info(result)



autoCalcOverhandEndturns

ОПИСАНИЕ

The autoCalcOverhandEndturns property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

autoCalcOverhandEndturns = value

ПРИМЕР

let result = motor.autoCalcOverhandEndturns
console.info(result)



radialOverhandOuterEndturn

ОПИСАНИЕ

The radialOverhandOuterEndturn property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

radialOverhandOuterEndturn = value

ПРИМЕР

- let result = motor.radialOverhandOuterEndturn
 console.info(result)



radialOverhandInnerEndturn

ОПИСАНИЕ

The radialOverhandInnerEndturn property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

radialOverhandInnerEndturn = value

ПРИМЕР

- let result = motor.radialOverhandInnerEndturn
 console.info(result)



heightOuterEndturn

ОПИСАНИЕ

The heightOuterEndturn property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

heightOuterEndturn = value

ПРИМЕР

let result = motor.heightOuterEndturn
console.info(result)



heightInnerEndturn

ОПИСАНИЕ

 $The \ {\tt heightInnerEndturn} \ property...$

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

heightInnerEndturn = value

ПРИМЕР

let result = motor.heightInnerEndturn
console.info(result)



5.8.4 Методы

Last update: 3 мая 2025, 11:59

isPlanar()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isPlanar(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = isPlanar(value)
 2 console.info(result)
 - . также

isToroidal()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isToroidal(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isToroidal(value)
 console.info(result)



isSingleLayer()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isSingleLayer(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isSingleLayer(value)
 console.info(result)



isDoubleLayer()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isDoubleLayer(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isDoubleLayer(value)
 console.info(result)



isOrientationUpperLower()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

isOrientationUpperLower(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isOrientationUpperLower(value)
 console.info(result)



isOrientationLeftRight()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isOrientationLeftRight(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isOrientationLeftRight(value)
 console.info(result)



checkOverlapEndturns()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

checkOverlapEndturns(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = checkOverlapEndturns(value)
 console.info(result)



volumeOverlapEndturns()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

volumeOverlapEndturns(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = volumeOverlapEndturns(value)
 console.info(result)



distanceBetweenEndturns()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

distanceBetweenEndturns(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = distanceBetweenEndturns(value)
 console.info(result)



5.9 Mesh

5.9.1 Описание

Встроенный объект меsh представляет собой модель сетки с различными параметрами.

Свойства типа меsh

Ниже приведен список доступных свойств, предоставляемых типом Mesh . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- <u>autoSizeBound</u>:
- sizeBound:
- <u>numberSlices</u>:
- airgapQuality:
- horizontalSymmetry:
- boundCylinderRadius:
- boundCylinderAxialExtension:

Методы типа Mesh

Ниже приведен список доступных методов, предоставляемых типом Mesh . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Last update: 3 мая 2025, 11:59

5.9.2 Свойства

Свойства типа меsh

Ниже приведен список доступных свойств, предоставляемых типом Mesh . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- <u>autoSizeBound</u>:
- <u>sizeBound</u>:
- <u>numberSlices</u>:
- airgapQuality:
- horizontalSymmetry:
- boundCylinderRadius:
- $\underline{\text{boundCylinderAxialExtension}}$:

Last update: 3 мая 2025, 11:59

autoSizeBound

ОПИСАНИЕ

The autoSizeBound property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

bool

доступ

Чтение\Запись

СИНТАКСИС

autoSizeBound = value

ПРИМЕР

```
let mesh = motor.mesh
let result = mesh.autoSizeBound
console.info(result)
```



sizeBound

ОПИСАНИЕ

The sizeBound property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

sizeBound = value

ПРИМЕР

```
let mesh = motor.mesh
let result = mesh.sizeBound
console.info(result)
```



numberSlices

ОПИСАНИЕ

The numberSlices property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

numberSlices = value

ПРИМЕР

```
1 let mesh = motor.mesh
2 let result = mesh.numberSlices
3 console.info(result)
```



airgapQuality

ОПИСАНИЕ

The airgapQuality property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

```
airgapQuality = value
```

ПРИМЕР

```
let mesh = motor.mesh
let result = mesh.airgapQuality
console.info(result)
```



horizontalSymmetry

ОПИСАНИЕ

 $The \ {\tt horizontalSymmetry} \ property...$

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

bool

доступ

Чтение\Запись

СИНТАКСИС

horizontalSymmetry = value

ПРИМЕР

```
1 let mesh = motor.mesh
2 let result = mesh.horizontalSymmetry
3 console.info(result)
```



5.9.3 Методы

Методы Mesh

Ниже приведен список доступных методов, предоставляемых типом Mesh . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

5.10 EmptyMaterial

5.10.1 Описание EmptyMaterial

Материал пустой, предназначен для задания материала деталей мотора $\ \ \$ Ріесе .

Синтаксис

```
let m1 = Material.empty() // Cnoco6 1
let m2 = new EmptyMaterial() // Cnoco6 2
```

Свойства EmptyMaterial

Нет свойств.

Методы EmptyMaterial

Нет методов.



- Material
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- MagnetParallelMaterial
- MagnetRadialMaterial
- CustomMaterial

5.11 GeneralMaterial

5.11.1 Описание GeneralMaterial

Материал для воздуха, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.general() // Cnoco6 1
let m2 = new GeneralMaterial() // Cnoco6 2
```

Свойства GeneralMaterial

Нет свойств.

Методы GeneralMaterial

Нет методов.



- Material
- EmptyMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- $\bullet \ Magnet Parallel Material\\$
- MagnetRadialMaterial
- CustomMaterial

5.12 IronMaterial

5.12.1 Описание IronMaterial

Материал для сердечника, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.iron() // Cnoco6 1
let m2 = new IronMaterial() // Cnoco6 2
```

Свойства IronMaterial

Нет свойств.

Методы IronMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- WindingMaterial
- EndturnMaterial
- $\bullet \ Magnet Parallel Material\\$
- MagnetRadialMaterial
- CustomMaterial

5.12.2 Свойства

Свойства IronMaterial

Ниже приведен список доступных свойств, предоставляемых типом IronMaterial. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

fillCoefs

ОПИСАНИЕ

The fillCoefs property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

fillCoefs = value

ПРИМЕР

let result = motor.fillCoefs
console.info(result)



dsomaloy

ОПИСАНИЕ

The dsomaloy property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

```
dsomaloy = value
```

ПРИМЕР

```
let m = Material.iron()
m.dsomaloy = 0.9
let result = m.dsomaloy
console.info(result)
```



5.13 ConductorMaterial

5.13.1 Описание ConductorMaterial

Материал проводник, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.conductor() // Cnocof 1
let m2 = new ConductorMaterial() // Cnocof 2
```

Свойства ConductorMaterial

Нет свойств.

Методы ConductorMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- MagnetParallelMaterial
- MagnetRadialMaterial
- CustomMaterial

5.14 WindingMaterial

5.14.1 Описание WindingMaterial

Материал обмоток, предназначен для задания материала деталей мотора $\,\,$ Ріесе $\,$

Синтаксис

```
let m1 = Material.winding() // Cnoco6 1
let m2 = new WindingMaterial() // Cnoco6 2
```

Свойства WindingMaterial

Методы WindingMaterial



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- EndturnMaterial
- MagnetParallelMaterial
- MagnetRadialMaterial
- CustomMaterial

5.14.2 Свойства

Свойства WindingMaterial

Ниже приведен список доступных свойств, предоставляемых типом WindingMaterial. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

layer

ОПИСАНИЕ

The layer property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

layer = value

ПРИМЕР

1 let result = motor.layer
2 console.info(result)



turn

ОПИСАНИЕ

The turn property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

turn = value

ПРИМЕР

1 let result = motor.turn
2 console.info(result)



strand

ОПИСАНИЕ

The strand property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

strand = value

ПРИМЕР

let result = motor.strand
console.info(result)



windingModel

ОПИСАНИЕ

The windingModel property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

windingModel = value

ПРИМЕР

let result = motor.windingModel
console.info(result)



5.14.3 Методы

Методы WindingMaterial

isWindingModelLumped()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

isWindingModelLumped(argument: type) : returnType

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isWindingModelLumped(value)
 console.info(result)



isWindingModelFull()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isWindingModelFull(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = isWindingModelFull(value)
 console.info(result)



5.15 EndturnMaterial

5.15.1 Описание EndturnMaterial

Материал лобовых частей обмоток, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.endturn() // Cnoco6 1
let m2 = new EndturnMaterial() // Cnoco6 2
```

Свойства EndturnMaterial

Нет свойств.

Методы EndturnMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- $\bullet \ Magnet Parallel Material\\$
- MagnetRadialMaterial
- CustomMaterial

5.16 MagnetRadialMaterial

5.16.1 Описание MagnetRadialMaterial

Материал для магнитов с радиальным намагничиванием, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.magnetRadial() // Cποcοб 1
let m2 = new MagnetRadialMaterial() // Cποcοб 2
```

Свойства MagnetRadialMaterial

Методы MagnetRadialMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- MagnetParallelMaterial
- CustomMaterial

5.16.2 Свойства

Свойства типа MagnetRadialMaterial

Ниже приведен список доступных свойств, предоставляемых типом MagnetRadialMaterial. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

direction

ОПИСАНИЕ

The direction property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

direction = value

ПРИМЕР

let result = motor.direction
console.info(result)



center

ОПИСАНИЕ

The center property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

center = value

ПРИМЕР

let result = motor.center
console.info(result)



5.17 MagnetParallelMaterial

5.17.1 Описание MagnetParallelMaterial

Материал для магнитов с параллельным намагничиванием, предназначен для задания материала деталей мотора Ріесе .

Синтаксис

```
let m1 = Material.magnetParallel() // Cnoco6 1
let m2 = new MagnetParallelMaterial() // Cnoco6 2
```

Свойства MagnetParallelMaterial

Методы MagnetParallelMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- MagnetRadialMaterial
- CustomMaterial

5.17.2 Свойства

Свойства MagnetParallelMaterial

Ниже приведен список доступных свойств, предоставляемых типом MagnetParallelMaterial. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

angle

ОПИСАНИЕ

The angle property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

angle = value

ПРИМЕР

1 let result = motor.angle
2 console.info(result)



segmentRadiuses

ОПИСАНИЕ

The segmentRadiuses property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

segmentRadiuses = value

ПРИМЕР

let result = motor.segmentRadiuses
console.info(result)



savePoleBorder

ОПИСАНИЕ

The savePoleBorder property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

savePoleBorder = value

ПРИМЕР

let result = motor.savePoleBorder
console.info(result)



5.18 CustomMaterial

5.18.1 Описание CustomMaterial

Материал пользователя, предназначен для задания материала деталей мотора Ріесе.

Синтаксис

```
let m1 = Material.custom(Qt.red) // Cποco6 1
let m2 = new CustomMaterial(Qt.red) // Cπoco6 2
```

Свойства CustomMaterial

• color: Цвет материала.

Методы CustomMaterial

Нет методов.



- Material
- EmptyMaterial
- GeneralMaterial
- ConductorMaterial
- IronMaterial
- WindingMaterial
- EndturnMaterial
- MagnetParallelMaterial
- MagnetRadialMaterial

5.18.2 Свойства

Свойства типа CustomMaterial

Ниже приведен список доступных свойств, предоставляемых типом [CustomMaterial]. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

• color: Цвет материала.

color

ОПИСАНИЕ

Цвет материала.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

Одно из значений Qt.Colors

ТИП ЗНАЧЕНИЯ СВОЙСТВА

QColor

доступ

Чтение\Запись

СИНТАКСИС

```
color = value
```

ПРИМЕР

```
let m = Material.custom(Qt.red)
m.color = Qt.green
let result = m.color
console.info(result)
```



5.19 Point3

5.19.1 Описание

Встроенный тип Point3 представляет точку в 3D пространстве.

Свойства типа Point3

Ниже приведен список доступных свойств, предоставляемых типом Point3 . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

- <u>x</u>:
- <u>y</u>:
- <u>z</u> :

Методы типа Point3

Ниже приведен список доступных методов, предоставляемых типом Point3. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>distance</u>:
- mirrorO:
- mirrorX:
- mirrorXY:
- mirrorXZ:
- mirrorY:
- mirrorYZ:
- mirrorZ:
- move:
- moveX:
- moveY:
- moveZ:
- <u>rotate</u>:
- <u>rotateX</u>:
- <u>rotateY</u>:
- <u>rotateZ</u>:
- <u>scale</u>:
- scaleX:
- scaleXY:
- <u>scaleXYZ</u>:
- <u>scaleXZ</u>:
- <u>scaleY</u>:
- scaleYZ:
- <u>scaleZ</u>:
- transcale:
- <u>transcaleX</u>:
- <u>transcaleY</u>:
- <u>transcaleZ</u>:

5.19.2 Свойства

Свойства типа Point3

Ниже приведен список доступных свойств, предоставляемых типом Point3 . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>x</u>:
- <u>y</u>:
- <u>z</u>:

X

ОПИСАНИЕ

The x property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

x = value

ПРИМЕР

```
let result = motor.x
console.info(result)
```



у

ОПИСАНИЕ

The y property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

y = value

ПРИМЕР

```
1 let result = motor.y
2 console.info(result)
```



z

ОПИСАНИЕ

The z property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

z = value

ПРИМЕР

```
1 let result = motor.z
2 console.info(result)
```



5.19.3 Методы

Методы типа Point3

Ниже приведен список доступных методов, предоставляемых типом Point3 . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

- <u>distance</u>:
- <u>mirrorO</u>:
- mirrorX:
- mirrorXY:
- mirrorXZ:
- mirrorY:
- mirrorYZ:
- mirrorZ:
- move:
- moveX:
- moveY:
- moveZ:
- rotate:
- <u>rotateX</u>:
- <u>rotateY</u>:
- rotateZ:
- <u>scale</u>:
- <u>scaleX</u>:
- scaleXY:
- <u>scaleXYZ</u>:
- <u>scaleXZ</u>:
- <u>scaleY</u>:
- <u>scaleYZ</u>:
- scaleZ:
- <u>transcale</u>:
- <u>transcaleX</u>:
- <u>transcaleY</u>:
- transcaleZ:

Last update: 3 мая 2025, 11:59

distance()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
distance(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = distance(value)
 2 console.info(result)



translate()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translate(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = translate(value)
 console.info(result)



translateX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translateX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = translateX(value)
 2 console.info(result)



translateY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translateY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = translateY(value)
 console.info(result)



translateY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translateY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = translateY(value)
 console.info(result)



move()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
move(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = move(value)
 2 console.info(result)
 - 🤼. также

moveX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveX(value)
 2 console.info(result)



moveY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveY(value)
 2 console.info(result)



moveZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveZ(value)
 2 console.info(result)



rotate()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotate(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = rotate(value)
 2 console.info(result)
 - 🤼. также

rotateX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateX(value)
 console.info(result)



rotateY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateY(value)
 console.info(result)



rotateZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateZ(value)
 console.info(result)



mirrorO()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorO(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = mirrorO(value)
 console.info(result)



mirrorX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = mirrorX(value)
 console.info(result)
 - й. также

mirrorY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorY(value)
 2 console.info(result)



mirrorZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorZ(value)
 2 console.info(result)
 - і. также

mirrorXY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorXY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorXY(value)
 2 console.info(result)



mirrorYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorYZ(value)
 2 console.info(result)



mirrorXZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorXZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorXZ(value)
 2 console.info(result)



scale()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scale(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scale(value)
 2 console.info(result)



scaleX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleX(value)
 2 console.info(result)
 - 🤼 также

scaleY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = scaleY(value)
2 console.info(result)
```



scaleZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleZ(value)
 2 console.info(result)
 - 🤼. также

scaleXY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXY(value)
 2 console.info(result)



scaleYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleYZ(value)
 2 console.info(result)



scaleXZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXZ(value)
 2 console.info(result)



scaleXYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXYZ(value)
 2 console.info(result)



5.20 Vector3

5.20.1 Описание

Last update: 3 мая 2025, 11:59

5.20.2 Свойства

Свойства типа Vector3

Ниже приведен список доступных свойств, предоставляемых типом Vector3 . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Last update: 3 мая 2025, 11:59

X

ОПИСАНИЕ

The x property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

x = value

ПРИМЕР

```
1 let result = motor.x
2 console.info(result)
```



у

ОПИСАНИЕ

The y property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

y = value

ПРИМЕР

```
1 let result = motor.y
2 console.info(result)
```



z

ОПИСАНИЕ

The z property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

доступ

Чтение\Запись

СИНТАКСИС

z = value

ПРИМЕР

```
1 let result = motor.z
2 console.info(result)
```



5.20.3 Методы

Last update: 3 мая 2025, 11:59

lenght()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
lenght(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = lenght(value)
2 console.info(result)
```



lenght2()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
lenght2(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = lenght2(value)
2 console.info(result)
```



angle()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
angle(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = angle(value)
 console.info(result)



isZero()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isZero(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = isZero(value)
2 console.info(result)
```



5.21 Shape

5.21.1 Описание

Last update: 3 мая 2025, 11:59

5.21.2 Свойства

Свойства типа Shape

Ниже приведен список доступных свойств, предоставляемых типом [Shape]. Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

Last update: 3 мая 2025, 11:59

5.21.3 Методы

Last update: 3 мая 2025, 11:59

isEmpty()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
isEmpty(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

```
1 let result = isEmpty(value)
2 console.info(result)
```



unite()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
unite(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = unite(value)
 console.info(result)



intersect()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
intersect(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = intersect(value)
 2 console.info(result)
 - 🗓. также

difference()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
difference(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = difference(value)
 console.info(result)



diff()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
diff(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = diff(value)
 2 console.info(result)
 - 🤼. также

extrude()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
extrude(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = extrude(value)
 2 console.info(result)



extrudeX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
extrudeX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = extrudeX(value)
 2 console.info(result)



extrudeY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
extrudeY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = extrudeY(value)
 2 console.info(result)



extrudeZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
extrudeZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = extrudeZ(value)
 2 console.info(result)



unify()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
unify(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = unify(value)
 console.info(result)



center()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
center(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = center(value)
 2 console.info(result)
 - 🤼 также

boundBox()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
boundBox(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = boundBox(value)
 2 console.info(result)



translate()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translate(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = translate(value)
 console.info(result)



translateX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translateX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = translateX(value)
 console.info(result)



translateY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
translateY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = translateY(value)
 2 console.info(result)



move()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
move(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = move(value)
 2 console.info(result)
 - 🤼 также

moveX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveX(value)
 2 console.info(result)



moveY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveY(value)
 2 console.info(result)



moveZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
moveZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = moveZ(value)
 2 console.info(result)



rotate()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotate(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = rotate(value)
 2 console.info(result)
 - 🤼. также

rotateX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateX(value)
 console.info(result)



rotateY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateY(value)
 console.info(result)



rotateZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
rotateZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = rotateZ(value)
 console.info(result)



mirrorO()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorO(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = mirrorO(value)
 console.info(result)



mirrorX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = mirrorX(value)
 console.info(result)



mirrorY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorY(value)
 2 console.info(result)



mirrorZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorZ(value)
 2 console.info(result)
 - 🤼 также

mirrorXY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorXY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorXY(value)
 2 console.info(result)



mirrorYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorYZ(value)
 2 console.info(result)



mirrorXZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
mirrorXZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = mirrorXZ(value)
 2 console.info(result)



scale()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scale(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- let result = scale(value)
 console.info(result)



scaleX()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleX(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleX(value)
 2 console.info(result)
 - 🤼 также

scaleY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleY(value)
 2 console.info(result)
 - 🤼. также

scaleZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleZ(value)
 2 console.info(result)
 - 🤼 также

scaleXY()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXY(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXY(value)
 2 console.info(result)



scaleYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleYZ(value)
 2 console.info(result)



scaleXZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXZ(value)
 2 console.info(result)



scaleXYZ()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
scaleXYZ(argument: type) : returnType
```

АРГУМЕНТЫ

• argument (type, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

гетигл Туре: описание возвращаемого значения.

ПРИМЕР

- 1 let result = scaleXYZ(value)
 2 console.info(result)



5.22 Piece

5.22.1 Описание

Встроенный тип Ріесе представляет собой деталь двигателя.

Свойства типа Ріесе

Ниже приведен список доступных свойств, предоставляемых типом Ріесе . Щелкните имя свойства, чтобы увидеть её подробное описание и примеры.

Методы типа Ріесе

Ниже приведен список доступных методов, предоставляемых типом Ріесе . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

• toFileStep:

5.22.2 Свойства

Методы объекта Ріесе

Ниже приведен список доступных методов, предоставляемых объектом Piece . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

5.22.3 Методы

Методы объекта Ріесе

Ниже приведен список доступных методов, предоставляемых объектом Piece . Щелкните имя метода, чтобы увидеть его подробное описание и примеры.

• $\underline{\text{toFileStep}}$: Сохранить деталь в файле.

toFileStep()

ОПИСАНИЕ

Сохранить деталь в файле

СИНТАКСИС

toFileStep(pathFile: string)

АРГУМЕНТЫ

• pathFile (string, обязательно):

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

Ничего не возвлащает

ПРИМЕР

piece.toFileStep("d:/temp/piece.Step")



5.23 UI-виджеты

5.23.1 Описание

5.23.2 QWidget()

Описание

Описание метода.

Синтаксис

```
QWidget(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- 1 let result = QWidget(value)
 2 console.info(result)
 - 🛄. также

5.23.3 QLabel()

Описание

Описание метода.

Синтаксис

```
QLabel(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- 1 let result = QLabel(value)
 2 console.info(result)
 - 🗓. также

5.23.4 QLineEdit()

Описание

Описание метода.

Синтаксис

```
QLineEdit(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QLineEdit(value)
 console.info(result)



5.23.5 QPushButton()

Описание

Описание метода.

Синтаксис

QPushButton(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QPushButton(value)
 console.info(result)



5.23.6 QSpinBox()

Описание

Описание метода.

Синтаксис

```
QSpinBox(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QSpinBox(value)
 console.info(result)
 - іі. также

5.23.7 QDoubleSpinBox()

Описание

Описание метода.

Синтаксис

QDoubleSpinBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QDoubleSpinBox(value)
 console.info(result)



5.23.8 QComboBox()

Описание

Описание метода.

Синтаксис

```
QComboBox(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QComboBox(value)
 console.info(result)



5.23.9 QGroupBox()

Описание

Описание метода.

Синтаксис

```
QGroupBox(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QGroupBox(value)
 console.info(result)



5.23.10 QCheckBox()

Описание

Описание метода.

Синтаксис

```
QCheckBox(argument: type) : returnType
```

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QCheckBox(value)
 console.info(result)



5.23.11 QGridLayout()

Описание

Описание метода.

Синтаксис

QGridLayout(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QGridLayout(value)
 console.info(result)



5.23.12 QFormLayout()

Описание

Описание метода.

Синтаксис

QFormLayout(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = QFormLayout(value)
 console.info(result)



5.23.13 WarningIcon()

Описание

Описание метода.

Синтаксис

WarningIcon(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = WarningIcon(value)
 console.info(result)



5.23.14 ExclamationIcon()

Описание

Описание метода.

Синтаксис

ExclamationIcon(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = ExclamationIcon(value)
 console.info(result)



5.23.15 NumberEdit()

Описание

Описание метода.

Синтаксис

NumberEdit(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = NumberEdit(value)
 console.info(result)



5.23.16 NumberSlotSpinBox()

Описание

Описание метода.

Синтаксис

NumberSlotSpinBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = NumberSlotSpinBox(value)
 console.info(result)



5.23.17 StatorTypeComboBox()

Описание

Описание метода.

Синтаксис

StatorTypeComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = StatorTypeComboBox(value)
 console.info(result)



5.23.18 WindingLayersComboBox()

Описание

Описание метода.

Синтаксис

WindingLayersComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = WindingLayersComboBox(value)
 console.info(result)



5.23.19 WindingLayersOrientationComboBox()

Описание

Описание метода.

Синтаксис

WindingLayersOrientationComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = WindingLayersOrientationComboBox(value)
 console.info(result)



5.23.20 WindingTypeComboBox()

Описание

Описание метода.

Синтаксис

WindingTypeComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = WindingTypeComboBox(value)
 console.info(result)



5.23.21 PoleArrangementComboBox()

Описание

Описание метода.

Синтаксис

PoleArrangementComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = PoleArrangementComboBox(value)
 console.info(result)



5.23.22 StatorConnectionComboBox()

Описание

Описание метода.

Синтаксис

StatorConnectionComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = StatorConnectionComboBox(value)
 console.info(result)



5.23.23 RotorConnectionComboBox()

Описание

Описание метода.

Синтаксис

RotorConnectionComboBox(argument: type) : returnType

Аргументы

• argument (type, обязательно): описание аргумента.

Возвращаемое значение

гетигл Туре: описание возвращаемого значения.

Пример

- let result = RotorConnectionComboBox(value)
 console.info(result)

