

# Руководство MotorXP- AFM Scripting API

---

None

*MotorXP Team*

*Copyright © 2025 MotorXP*

# Table of contents

---

1. Введение	4
1.1 Описание	4
1.2 Официальный сайт	4
1.3 Контакты	4
2. Глобальные функции	5
2.1 Глобальные функции	5
2.2 <code>include()</code>	6
2.3 <code>require()</code>	7
2.4 <code>writeFile()</code>	8
2.5 <code>readFile()</code>	9
3. Константы	10
3.1 Список констант	10
3.2 <code>Direction</code>	12
3.3 <code>Coil orientation</code>	13
3.4 <code>Magnetization</code>	14
3.5 <code>PoleArrangement</code>	15
3.6 Именованные цвета	16
4. Встроенные объекты	17
4.1 Встроенные объекты	17
4.2 <code>Math</code>	18
4.3 <code>Geom</code>	31
4.4 <code>Material</code>	76
4.5 <code>QtWidgets</code>	87
4.6 <code>console</code>	112
4.7 <code>motor</code>	120
5. Типы объектов	134
5.1 <code>Stator</code>	135
5.2 <code>StatorItem</code>	164
5.3 <code>Rotor</code>	178
5.4 <code>RotorItem</code>	210
5.5 <code>Airgap</code>	225
5.6 <code>Winding</code>	228
5.7 <code>Mesh</code>	265
5.8 <code>EmptyMaterial</code>	273
5.9 <code>GeneralMaterial</code>	274

5.10	IronMaterial	275
5.11	ConductorMaterial	279
5.12	WindingMaterial	280
5.13	EndturnMaterial	289
5.14	MagnetRadialMaterial	290
5.15	MagnetParallelMaterial	294
5.16	CustomMaterial	299
5.17	Point3	302
5.18	Vector3	335
5.19	Shape	345
5.20	BoundingBox	387
5.21	Piece	402
5.22	UI-виджеты	405

# 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).

### Информация

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](http://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](mailto:nika_homework@mail.ru), [info@motorxp.com](mailto:info@motorxp.com).

Last update: 16 апреля 2025, 16:38

## 2. Глобальные функции

---

### 2.1 Глобальные функции

---

- [include\(\)](#): Включение содержимого одного файла скрипта в другой файл скрипта.
- [require\(\)](#): Загрузка модуля.
- [writeFile\(\)](#): Запись в файл.
- [readFile\(\)](#): Чтение файла.

Last update: 15 апреля 2025, 14:48

## 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:

- a. 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

```
1 // Подключаем файл по полному пути
2 include("c:/scripts/script.js")
3
4 // Подключаем файл script2.js из той же директории
5 include("./script2.js")
6
7 // Подключаем файл script3.js из родительской директории
8 include("../script3.js")
```



См. также

[require\(\)](#)

Last update: 15 апреля 2025, 14:48

## 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:
  - Может быть **абсолютным** (полный путь к файлу на диске).  
Например, `"C:/scripts/library.js"`.
  - 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

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

### 2.3.5 Пример

```
1 let path = "./helpers"
2 let helpers = require(path)
3
4 let library = require("C:/scripts/library")
```



#### См. также

[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

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

```
1 let sampleText = "Sample text"
2 let result = writeFile("c:/temp/sample.txt", sampleText)
3 console.info(result)
```



См. также

[readFile\(\)](#)

Last update: 15 апреля 2025, 14:48



## 2.5 readFile()

### 2.5.1 Описание

Чтение из файла.

### 2.5.2 Синтаксис

```
readFile(path : string) : string
```

### 2.5.3 Аргументы

- **path** (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

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



См. также

[writeFile\(\)](#)

Last update: 15 апреля 2025, 14:48

## 3. Константы

---

### 3.1 Список констант

---

#### 3.1.1 Direction

- `Direction.CW` — Направление по часовой стрелке.  
Значение: `-1`
- `Direction.CCW` — Направление часовой стрелки.  
Значение: `+1`

#### 3.1.2 Coil orientation

- `Coil.CW` — Намотка по часовой стрелке.  
Значение: `-1`
- `Coil.CCW` — Намотка против часовой стрелки.  
Значение: `+1`

#### 3.1.3 Coil direction

- `Coil.Incoming` — Incoming.  
Значение: `+1`
- `Coil.Outcoming` — Outcoming.  
Значение: `-1`

#### 3.1.4 PoleArrangement

- `PoleArrangement.NN` — NN.  
Значение: `0`
- `PoleArrangement.NS` — NS.  
Значение: `1`
- `PoleArrangement.NSNS` — NSNS.  
Значение: `2`
- `PoleArrangement.NSSN` — NSSN.  
Значение: `3`
- `PoleArrangement.NNSS` — NNSS.  
Значение: `4`
- `PoleArrangement.NNNN` — NNNN.  
Значение: `5`

#### 3.1.5 Magnetization

- `Magnetization.From` — направление "от".  
Значение: `1`
- `Magnetization.Toward` — направление "к".  
Значение: `2`
- `Magnetization.CW` — по часовой стрелке.  
Значение: `3`
- `Magnetization.CCW` — против часовой стрелки.  
Значение: `4`

### 3.1.6 Именованные цвета

---

Список доступных цветов и их визуальное представление:

- `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  


Last update: 15 апреля 2025, 21:44

## 3.2 Direction

---

- `Direction.CW` — Направление по часовой стрелке.

Значение: `-1`

- `Direction.CCW` — Направление часовой стрелки.

Значение: `+1`

Last update: 16 апреля 2025, 12:45

## 3.3 Coil orientation

---

- `Coil.CW` — Намотка по часовой стрелке.  
Значение: `-1`
- `Coil.CCW` — Намотка против часовой стрелки.  
Значение: `+1`

### 3.3.1 Coil direction

---

- `Coil.Incoming` — Incoming.  
Значение: `+1`
- `Coil.Outcoming` — Outcoming.  
Значение: `-1`

Last update: 16 апреля 2025, 12:45

## 3.4 Magnetization

---

- `Magnetization.From` — направление "от".  
Значение: 1
- `Magnetization.Toward` — направление "к".  
Значение: 2
- `Magnetization.CW` — по часовой стрелке.  
Значение: 3
- `Magnetization.CCW` — против часовой стрелки.  
Значение: 4

Last update: 16 апреля 2025, 12:45

## 3.5 PoleArrangement

---

- `PoleArrangement.NN` — NN.  
Значение: 0
- `PoleArrangement.NS` — NS.  
Значение: 1
- `PoleArrangement.NSNS` — NSNS.  
Значение: 2
- `PoleArrangement.NSSN` — NSSN.  
Значение: 3
- `PoleArrangement.NNSS` — NNSS.  
Значение: 4
- `PoleArrangement.NNNN` — NNNN.  
Значение: 5

Last update: 16 апреля 2025, 12:45

## 3.6 Именованные цвета

---

Список доступных цветов и их визуальное представление:

- `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  


Last update: 16 апреля 2025, 12:45



## 4. Встроенные объекты

---

### 4.1 Встроенные объекты

---

- [console](#): Консоль приложения.
- [motor](#): Модель двигателя (генератора) с различными параметрами.
- [Math](#): Математические функции.
- [Geom](#): Геометрическое ядро.
- [Material](#): Материалы.
- [UI Widgets](#): UI-виджеты.

Last update: 16 апреля 2025, 12:45

## 4.2 Math

---

### 4.2.1 Обзор объекта Math

---

Встроенный объект `Math` дополняет стандартный объект `Math` и предоставляет ряд полезных математических функций.

#### Основные возможности

- 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 объекта Math

- `Math.Epsilon`: Минимальное положительное число, такое что  $1 + \text{eps} = 1$  при работе с числами с плавающей точкой.

#### Методы объекта Math

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

- `Math.deg()`: Перевод радиан в градусы.
- `Math.rad()`: Перевод градусов в радианы.
- `Math.fromPolar()`: Перевод координат точки из полярных координат (`radius`, `angle`) в декартовые координаты (`x`, `y`).
- `Math.normAngle()`: Нормализует значение угла, возвращает значение в интервале  $(0^\circ, 360^\circ)$ .
- `Math.middleAngle()`: Расчет среднего угла между углами  $a_1$  и  $a_2$  с учетом направления.
- `Math.isEpsilon()`: Возвращает `true`, если абсолютное значение `arg` меньше или равно `eps`.
- `Math.isEqual()`: Возвращает `true`, если разница абсолютных значений `arg1` и `arg2` меньше или равно `eps`.
- `Math.isLessEqual()`: Возвращает `true`, если значение `arg1` меньше или равно значению `arg2` с учетом точности `eps`.
- `Math.isGreatEqual()`: Возвращает `true`, если значение `arg1` больше или равно значению `arg2` с учетом точности `eps`.

Last update: 16 апреля 2025, 10:38

## 4.2.2 Константы

---

### Math.Epsilon

#### ОПИСАНИЕ

Минимальное положительное число, такое что  $1 + \text{eps} = 1$  при работе с числами с плавающей точкой.

#### ЗНАЧЕНИЕ

1e-7

#### ПРИМЕР

```
let eps = Math.Epsilon
console.info(eps)
```

Last update: 16 апреля 2025, 10:38

## Math.Epsilon

### ОПИСАНИЕ

Минимальное положительное число, такое что  $1 + \text{eps} = 1$  при работе с числами с плавающей точкой.

### ЗНАЧЕНИЕ

1e-7

### ПРИМЕР

```
1 let eps = Math.Epsilon
2 console.info(eps)
```

Last update: 16 апреля 2025, 10:38

## 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.

- [Math.deg\(\)](#): Перевод радиан в градусы.
- [Math.rad\(\)](#): Перевод градусов в радианы.
- [Math.fromPolar\(\)](#): Перевод координат точки из полярных координат `(radius, angle)` в декартовые координаты `(x, y)`.
- [Math.normAngle\(\)](#): Нормализует значение угла, возвращает значение в интервале  $(0^\circ, 360^\circ)$ .
- [Math.middleAngle\(\)](#): Расчет среднего угла между углами  $a_1$  и  $a_2$  с учетом направления.
- [Math.isEpsilon\(\)](#): Возвращает `true`, если абсолютное значение `arg` меньше или равно `eps`.
- [Math.isEqual\(\)](#): Возвращает `true`, если разница абсолютных значений `arg1` и `arg2` меньше или равно `eps`.
- [Math.isLessEqual\(\)](#): Возвращает `true`, если значение `arg1` меньше или равно значению `arg2` с учетом точности `eps`.
- [Math.isGreatEqual\(\)](#): Возвращает `true`, если значение `arg1` больше или равно значению `arg2` с учетом точности `eps`.

Last update: 16 апреля 2025, 10:38

## isEpsilon()

### ОПИСАНИЕ

Возвращает `true`, если абсолютное значение `arg` меньше или равно `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

- `bool`: результат.

### ПРИМЕР

```
1 let b = Math.isEpsilon(0.0001)
```



См. также

[Math.isEqual\(\)](#)

[Math.isLessEqual\(\)](#)

[Math.isGreatEqual\(\)](#)

Last update: 16 апреля 2025, 10:38

## isEqual()

### ОПИСАНИЕ

Возвращает `true`, если разница абсолютных значений `arg1` и `arg2` меньше или равно `eps`.

### СИНТАКСИС

```
isEqual(arg1 : number, arg2 : number, eps : 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

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



См. также

[Math.isEqual\(\)](#)

[Math.isLessEqual\(\)](#)

[Math.isGreatEqual\(\)](#)

Last update: 16 апреля 2025, 10:38

## isLessEqual()

### ОПИСАНИЕ

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

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `arg1` (`number`, обязательно): значение 1.
- `arg2` (`number`, обязательно): значение 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)
```



См. также

[Math.isEpsilon\(\)](#)

[Math.isEqual\(\)](#)

[Math.isLessEqual\(\)](#)

[Math.isGreatEqual\(\)](#)

Last update: 16 апреля 2025, 10:38



## isGreatEqual()

### ОПИСАНИЕ

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

### СИНТАКСИС

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

### 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 сравнения.

### ПРИМЕР

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



См. также

[Math.isEpsilon\(\)](#)

[Math.isEqual\(\)](#)

[Math.isGreatEqual\(\)](#)

Last update: 16 апреля 2025, 10:38

## 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)
```



См. также

[Math.deg\(\)](#)

[Math.normAngle\(\)](#)

[Math.middleAngle\(\)](#)

Last update: 16 апреля 2025, 10:38

## 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

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



#### См. также

[Math.rad\(\)](#)

[Math.normAngle\(\)](#)

[Math.middleAngle\(\)](#)

Last update: 16 апреля 2025, 10:38

## fromPolar()

### ОПИСАНИЕ

Перевод координат точки из полярных координат `(radius, angle)` в декартовы 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

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

Last update: 16 апреля 2025, 10:38

## normAngle()

### ОПИСАНИЕ

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

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `a` (number, обязательно): значение угла в градусах.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

- `number`: значение угла в интервале (0°, 360°).

### ПРИМЕР

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



См. также

[Math.rad\(\)](#)

[Math.deg\(\)](#)

[Math.middleAngle\(\)](#)

Last update: 16 апреля 2025, 10:38

## middleAngle()

### ОПИСАНИЕ

Расчет среднего угла между углами  $a_1$  и  $a_2$  с учетом направления ( `CW` — по часовой стрелке, `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.
- `dir` (number, обязательно): направление обхода, одно из констант `Direction.CW` | `Direction.CCW`.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

- `number`: средний угол.

### ПРИМЕР

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



См. также

[Math.rad\(\)](#)

[Math.deg\(\)](#)

[Math.normAngle\(\)](#)

Last update: 16 апреля 2025, 10:38

## 4.3 Geom

---

### 4.3.1 Обзор объекта Geom

---

Встроенный объект `Geom` реализует геометрическое ядро.

#### Основные возможности

#### Методы объекта Geom

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

Last update: 16 апреля 2025, 10:38

### 4.3.2 Методы

---

Last update: 16 апреля 2025, 10:38



## angle()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**angleX()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
angleX(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

**angleY()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
angleY(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

**angleZ()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
angleZ(argument: type) : returnType
```

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## angleBetweenVectors()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
angleBetweenVectors(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## radius()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
radius(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**radiusX()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
radiusX(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

**radiusY()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
radiusY(argument: type) : returnType
```

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38



**radiusZ()**

**ОПИСАНИЕ**

Описание метода.

**СИНТАКСИС**

```
radiusZ(argument: type) : returnType
```

**АРГУМЕНТЫ**

- `argument` (`type`, обязательно): описание аргумента.

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

`returnType` : описание возвращаемого значения.

**ПРИМЕР**

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



См. также

Last update: 16 апреля 2025, 10:38

**vector3()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
vector3(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

```
1 let result = vector3(value)
2 console.info(result)
```

 **См. также**

Last update: 16 апреля 2025, 10:38

## point3()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
point3(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

```
1 let result = point3(value)
2 console.info(result)
```



См. также

Last update: 16 апреля 2025, 10:38

## segment()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
segment(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## polysegment()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
polysegment(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**bspline()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
bspline(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## arc()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
arc(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**circle()**

**ОПИСАНИЕ**

Описание метода.

**СИНТАКСИС**

```
circle(argument: type) : returnType
```

**АРГУМЕНТЫ**


- `argument` (`type`, обязательно): описание аргумента.

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

`returnType` : описание возвращаемого значения.

**ПРИМЕР**

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

 **См. также**

Last update: 16 апреля 2025, 10:38



## ellipse()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
ellipse(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

ring()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
ring(argument: type) : returnType
```

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## ngon()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
ngon(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**square()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
square(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## rectangle()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
rectangle(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## polygon()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
polygon(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## infpplane()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
infpplane(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**box()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
box(argument: type) : returnType
```

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38



## cylinder()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
cylinder(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**collar()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
collar(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

cone()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
cone(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## sphere()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
sphere(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

**torus()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
torus(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## unite()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## difference()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## difference()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25



## intersect()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## section()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
section(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## fillet()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
fillet(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## fillet()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
fillet(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

**chamfer()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
chamfer(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 10:38

## unify()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## boundingBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
boundingBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## shape()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
shape(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38



## piece()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
piece(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## distance()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## pointAtSegment()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
pointAtSegment(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 10:38

## 4.4 Material

---

### 4.4.1 Обзор объекта Material

---

Встроенный объект `Material` предназначен для создания объектов различных материалов.

#### Основные возможности

#### Методы объекта Material

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

Last update: 16 апреля 2025, 10:38

## 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.

Last update: 16 апреля 2025, 10:38

**empty()**

ОПИСАНИЕ

Пустой материал.

СИНТАКСИС

```
empty(): MaterialEmpty
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MaterialEmpty

ПРИМЕР

```
1 var m = Material.empty();
```

 **См. также**

Last update: 16 апреля 2025, 10:38

**general()**

ОПИСАНИЕ

Материал для воздушной среды

СИНТАКСИС

```
general() : MaterialGeneral
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MaterialGeneral

ПРИМЕР

```
1 let m = Material.general()
```

 **См. также**

Last update: 16 апреля 2025, 10:38

**iron()**

ОПИСАНИЕ

Материал для железа

СИНТАКСИС

```
iron() : MaterialIron
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MaterialIron

ПРИМЕР

```
1 let m = Material.iron()
```

 **См. также**

Last update: 16 апреля 2025, 10:38



**conductor()**

ОПИСАНИЕ

Материал для проводника.

СИНТАКСИС

```
conductor() : MaterialConductor
```

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

MaterialConductor

ПРИМЕР

```
1 let m = Material.conductor()
```

 **См. также**

Last update: 16 апреля 2025, 10:38

## winding()

### ОПИСАНИЕ

Материал для обмотки

### СИНТАКСИС

```
winding(layer : number = Winding.LayerAuto, turn : number = 0, strand : number = 0) : MaterialWinding
```

### АРГУМЕНТЫ

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

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`MaterialWinding`

### ПРИМЕР

```
1 let m1 = Material.winding()  
2 let m2 = Material.winding(Winding.LayerSingle, 1, 1)
```



См. также

Last update: 16 апреля 2025, 10:38

## endturn()

### ОПИСАНИЕ

Материал для лобовых частей обмоток (Endturns).

### СИНТАКСИС

```
endturn() : MaterialEndturn
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`MaterialEndturn`

### ПРИМЕР

```
1 let m = Material.endturn()
```



См. также

Last update: 16 апреля 2025, 10:38

## magnetParallel()

### ОПИСАНИЕ

Материал для магнита с параллельным намагничиванием.

### СИНТАКСИС

```
magnetParallel() : MaterialMagnetParallel
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

[MaterialMagnetParallel](#)

### ПРИМЕР

```
1 let m = Material.magnetParallel()
```



См. также

[Material.magnetRadial\(\)](#)

Last update: 16 апреля 2025, 10:38

## magnetRadial()

### ОПИСАНИЕ

Материал для магнита с радиальным намагничиванием.

### СИНТАКСИС

```
magnetRadial() : MaterialMagnetRadial
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

[MaterialMagnetRadial](#)

### ПРИМЕР

```
1 let m = Material.magnetRadial();
```



См. также

[Material.magnetParallel\(\)](#)

Last update: 16 апреля 2025, 10:38

## custom()

### ОПИСАНИЕ

Пользовательский материал с заданным цветом.

### СИНТАКСИС

```
custom(color : Qt.color = Qt.red) : MaterialCustom
```

### АРГУМЕНТЫ

- `color` (`Qt.color`, опционально): цвет материала.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`MaterialCustom`

### ПРИМЕР

```
1 let m1 = Material.custom()
2 let m2 = Material.custom(Qt.green)
```



См. также

Last update: 16 апреля 2025, 10:38

## 4.5 QtWidgets

---

### 4.5.1 Обзор объекта QtWidgets

---

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

#### Основные возможности

Создание виджетов для построения форм пользовательского интерфейса

#### Методы объекта QtWidgets

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

Last update: 16 апреля 2025, 10:38

## 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.

Last update: 16 апреля 2025, 10:38



## createQGridLayout()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQFormLayout()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createQFormLayout(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

**createQWidget()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 12:45

## createQLabel()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQLineEdit()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQPushButton()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQSpinBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQDoubleSpinBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createQDoubleSpinBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## createQComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (type, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQGroupBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createQCheckBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createWarningIcon()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createExclamationIcon()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createNumberEdit()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createWindingLayersComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createWindingLayersComboBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

**createWindingLayersOrientationComboBox()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

```
createWindingLayersOrientationComboBox(argument: type) : returnType
```

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 12:45



## createWindingTypeComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (type, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createStatorTypeComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createStatorTypeComboBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createRotorTypeComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createRotorTypeComboBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (type, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createPoleArrangementComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createPoleArrangementComboBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (type, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createNumberSlotSpinBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createNumberSlotSpinBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createNumberPoleSpinBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createNumberPoleSpinBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## createWindingConnectionComboBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
createWindingConnectionComboBox(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## 4.6 console

---

### 4.6.1 Обзор объекта console

---

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

#### Основные возможности

- 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.

Last update: 16 апреля 2025, 10:38



## 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.
- [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.

Last update: 16 апреля 2025, 10:38

## console.log()

### ОПИСАНИЕ

Метод `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)
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`undefined`

### ПРИМЕР

```
1 // Один аргумент
2 console.log('log text')
3
4 // Несколько аргументов
5 console.log('text1', 'text2')
6 console.log('text1', 'text2', .. , 'text10')
7
8 // Различные типы аргументов
9 console.log('text1', 123, true, { key: 'value' })
```



См. также

[console.info\(\)](#)

[console.warn\(\)](#)

[console.error\(\)](#)

Last update: 16 апреля 2025, 10:38

## console.info()

### ОПИСАНИЕ

Метод `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)
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`undefined`

### ПРИМЕР

```
1 // Один аргумент
2 console.info('info text')
3
4 // Несколько аргументов
5 console.info('text1', 'text2')
6 console.info('text1', 'text2', .. , 'text10')
7
8 // Различные типы аргументов
9 console.info('text1', 123, true, { key: 'value' })
```

#### See also

[console.info\(\)](#)

[console.warn\(\)](#)

[console.error\(\)](#)

Last update: 16 апреля 2025, 10:38

## console.warn()

### ОПИСАНИЕ

Метод `console.warn()` 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

```
1 // Один аргумент
2 console.warn('warn text')
3
4 // Несколько аргументов
5 console.warn('text1', 'text2')
6 console.warn('text1', 'text2', .. , 'text10')
7
8 // Mixed types
9 console.warn('text1', 123, true, { key: 'value' })
```



См. также

`console.log()`

`console.info()`

`console.error()`

Last update: 16 апреля 2025, 10:38

## console.error()

### ОПИСАНИЕ

Метод `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)
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`undefined`

### ПРИМЕР

```
1 // Один аргумент
2 console.error('critical text')
3
4 // Несколько аргументов
5 console.error('text1', 'text2')
6 console.error('text1', 'text2', .. , 'text10')
7
8 // Различные типы аргументов
9 console.error('text1', 123, true, { key: 'value' })
```



См. также

[console.log\(\)](#)

[console.info\(\)](#)

[console.warn\(\)](#)

Last update: 16 апреля 2025, 10:38

## **console.clear()**

### ОПИСАНИЕ

Метод `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()
```

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

```
undefined
```

### ПРИМЕР

```
1 console.clear()
```

Last update: 16 апреля 2025, 10:38

**console.dir()**

**ОПИСАНИЕ**

Метод `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)
```

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

```
undefined
```

**ПРИМЕР**

```
1 // Define an object
2 const obj = {
3   name: 'MotorXP',
4   version: 1.0,
5   features: ['AFM Design', 'Electromagnetic Analysis', 'Optimization API'],
6   details: {
7     developer: 'MotorXP Team',
8     license: 'Combo'
9   }
10 };
11
12 // Inspect the object
13 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"

Last update: 16 апреля 2025, 10:38

## 4.7 motor

---

### 4.7.1 Motor Overview

---

#### Description

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

Last update: 16 апреля 2025, 10:38



## 4.7.2 Константы

---

### Константы объекта `motor`

- `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

Last update: 16 апреля 2025, 10:38

## Motor.SR

### ОПИСАНИЕ

Тип мотора: Статор-Ротор.

### ЗНАЧЕНИЕ

0

### ПРИМЕР

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



#### Связанные свойства

`motor.machineType`

Last update: 16 апреля 2025, 10:38

## Motor.SRS

### ОПИСАНИЕ

Тип мотора: Статор-Ротор-Статор.

### ЗНАЧЕНИЕ

1

### ПРИМЕР

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



#### Связанные свойства

`motor.machineType`

Last update: 16 апреля 2025, 10:38

**Motor.SRSRS**

ОПИСАНИЕ

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

ЗНАЧЕНИЕ

2

ПРИМЕР

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



Связанные свойства

motor.machineType

Last update: 16 апреля 2025, 10:38

## Motor.RSR

### ОПИСАНИЕ

Тип мотора: Ротор-Статор-Ротор.

### ЗНАЧЕНИЕ

3

### ПРИМЕР

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



#### Связанные свойства

`motor.machineType`

Last update: 16 апреля 2025, 10:38

## Motor.RSRSR

### ОПИСАНИЕ

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

### ЗНАЧЕНИЕ

4

### ПРИМЕР

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



#### Связанные свойства

`motor.machineType`

Last update: 16 апреля 2025, 10:38

## 4.7.3 Свойства

---

### Motor Properties

#### PROPERTIES OF THE `motor` OBJECT

- `machineType` : Type of the machine.
- `stator` : Returns the `Stator` object associated with the motor.
- `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.

Last update: 16 апреля 2025, 10:38

## machineType

### ОПИСАНИЕ

The `machineType` property defines the configuration type of the motor.

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- `Motor.SR` (0): stator - rotor
- `Motor.SRS` (1): stator - rotor - stator
- `Motor.SRSRS` (2): stator - rotor - stator - rotor - stator
- `Motor.RSR` (3): rotor - stator - rotor
- `Motor.RSRSR` (4): rotor - stator - rotor - stator - rotor

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

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

### ПРИМЕР

```
1 // Set the machine type to Stator-Rotor-Stator configuration
2 motor.machineType = Motor.SRS;
3
4 let type = motor.machineType
5
6 // Check the current machine type
7 console.log(motor.type) // Output: 1
```



См. также

Last update: 16 апреля 2025, 10:38



## stator

### ОПИСАНИЕ

The `propertyName` property defines the configuration type of the motor.

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- `Motor.SR` (0): stator - rotor

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

### ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

rotor

ОПИСАНИЕ

The `propertyName` property defines the configuration type of the motor.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- `Motor.SR` (0): stator - rotor

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

ДОСТУП

Только Чтение

СИНТАКСИС

ПРИМЕР

1

См. также

Last update: 16 апреля 2025, 10:38

## airgap

### ОПИСАНИЕ

Объект типа `Airgap`, который определяет свойства воздушных зазоров мотора.

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

`Airgap`

### ДОСТУП

Только Чтение

### СИНТАКСИС

```
motor.airgap
```

### ПРИМЕР

```
1 let airgap = motor.airgap
```

 **См. также**

Last update: 16 апреля 2025, 10:38

## winding

### ОПИСАНИЕ

The `machineType` property defines the configuration type of the motor.

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- `Motor.SR` (0): stator - rotor
- `Motor.SRS` (1): stator - rotor - stator
- `Motor.SRSRS` (2): stator - rotor - stator - rotor - stator
- `Motor.RSR` (3): rotor - stator - rotor
- `Motor.RSRSR` (4): rotor - stator - rotor - stator - rotor

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Только Чтение

### СИНТАКСИС

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

### ПРИМЕР

```
1 // Set the machine type to Stator-Rotor-Stator configuration  
2 motor.machineType = Motor.SRS;  
3  
4 // Check the current machine type  
5 console.log(motor.machineType); // Output: 1
```

 См. также

Last update: 16 апреля 2025, 10:38

mesh

ОПИСАНИЕ

The `propertyName` property defines the configuration type of the motor.

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ

- `Motor.SR` (0): stator - rotor

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

ДОСТУП

Только Чтение

СИНТАКСИС

ПРИМЕР

1

См. также

Last update: 16 апреля 2025, 10:38

## 5. Типы объектов

---

Last update: 12 апреля 2025, 14:03

## 5.1 Stator

---

Last update: 15 апреля 2025, 20:18

## 5.1.1 Константы

---

Last update: 15 апреля 2025, 20:23



## Yokeless

### ОПИСАНИЕ

Константа `Yokeless`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Yoke

### ОПИСАНИЕ

Константа `Yoke`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## 5.1.2 Свойства

---

Last update: 15 апреля 2025, 20:18

## outerDiameter

### ОПИСАНИЕ

The `outerDiameter` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
outerDiameter = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## outerRadius

### ОПИСАНИЕ

The `outerRadius` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
outerRadius = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## innerDiameter

### ОПИСАНИЕ

The `innerDiameter` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
innerDiameter = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## innerRadius

### ОПИСАНИЕ

The `innerRadius` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
innerRadius = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## numberSlots

### ОПИСАНИЕ

The `numberSlots` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberSlots = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## slotAngleSpan

### ОПИСАНИЕ

The `slotAngleSpan` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
slotAngleSpan = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## typeMiddleItem

### ОПИСАНИЕ

The `typeMiddleItem` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
typeMiddleItem = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## ironMaterial

### ОПИСАНИЕ

The `ironMaterial` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
ironMaterial = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## ironStacking

### ОПИСАНИЕ

The `ironStacking` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
ironStacking = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## windingMaterial

### ОПИСАНИЕ

The `windingMaterial` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
windingMaterial = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## windingTemperature

### ОПИСАНИЕ

The `windingTemperature` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
windingTemperature = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## script

### ОПИСАНИЕ

The `script` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
script = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## nameScript

### ОПИСАНИЕ

The `nameScript` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
nameScript = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## countItem

### ОПИСАНИЕ

The `countItem` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
countItem = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## items

### ОПИСАНИЕ

The `items` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
items = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## simDomain

### ОПИСАНИЕ

The `simDomain` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
simDomain = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## simRadialOuterDomain

### ОПИСАНИЕ

The `simRadialOuterDomain` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
simRadialOuterDomain = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

### 5.1.3 Методы

---

Last update: 15 апреля 2025, 20:18

**item()**

**ОПИСАНИЕ**

Описание метода.

**СИНТАКСИС**

```
item(argument: type) : returnType
```

**АРГУМЕНТЫ**

- `argument` (`type`, обязательно): описание аргумента.

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

`returnType` : описание возвращаемого значения.

**ПРИМЕР**

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isLower()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isMiddle()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



**isUpper()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

## itemAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
itemAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## setItemAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
setItemAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## 5.2 StatorItem

---

Last update: 15 апреля 2025, 20:18

## 5.2.1 Константы

---

Last update: 15 апреля 2025, 20:23

## ID1

### ОПИСАНИЕ

Константа `ID1`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## ID2

### ОПИСАНИЕ

Константа `ID2`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## ID3

### ОПИСАНИЕ

Константа `ID3`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

```
1 let value = Constants.ID3;  
2 console.info(value);
```



См. также

Last update: 15 апреля 2025, 20:23



## 5.2.2 Свойства

---

Last update: 15 апреля 2025, 20:18

**id**

## ОПИСАНИЕ

The `id` property...

## ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

## ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

## ДОСТУП

Чтение\Запись

## СИНТАКСИС

```
id = value
```

## ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## height

### ОПИСАНИЕ

The `height` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
height = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## 5.2.3 Методы

---

### Методы объекта `StatorItem`

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

Last update: 15 апреля 2025, 21:44

isUpper()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isMiddle()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isLower()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

## angularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## setAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
setAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## 5.3 Rotor

---

Last update: 15 апреля 2025, 20:18

## 5.3.1 Константы

---

Last update: 15 апреля 2025, 20:23

## Yokeless

### ОПИСАНИЕ

Константа `Yokeless`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Yoke

### ОПИСАНИЕ

Константа `Yoke`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## 5.3.2 Свойства

---

Last update: 15 апреля 2025, 20:18

## outerDiameter

### ОПИСАНИЕ

The `outerDiameter` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
outerDiameter = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## outerRadius

### ОПИСАНИЕ

The `outerRadius` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
outerRadius = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## innerDiameter

### ОПИСАНИЕ

The `innerDiameter` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
innerDiameter = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## innerRadius

### ОПИСАНИЕ

The `innerRadius` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
innerRadius = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## numberPolePairs

### ОПИСАНИЕ

The `numberPolePairs` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberPolePairs = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## poleAngleSpan

### ОПИСАНИЕ

The `poleAngleSpan` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
poleAngleSpan = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## poleArrangement

### ОПИСАНИЕ

The `poleArrangement` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
poleArrangement = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## typeMiddleItem

### ОПИСАНИЕ

The `typeMiddleItem` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
typeMiddleItem = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## ironStacking

### ОПИСАНИЕ

The `ironStacking` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
ironStacking = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## ironMaterial

### ОПИСАНИЕ

The `ironMaterial` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
ironMaterial = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## magnetTemperature

### ОПИСАНИЕ

The `magnetTemperature` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
magnetTemperature = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## magnetMaterial

### ОПИСАНИЕ

The `magnetMaterial` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
magnetMaterial = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## conductorTemperature

### ОПИСАНИЕ

The `conductorTemperature` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
conductorTemperature = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## conductorMaterial

### ОПИСАНИЕ

The `conductorMaterial` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
conductorMaterial = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## script

### ОПИСАНИЕ

The `script` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
script = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## nameScript

### ОПИСАНИЕ

The `nameScript` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
nameScript = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## countItem

### ОПИСАНИЕ

The `countItem` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
countItem = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## items

### ОПИСАНИЕ

The `items` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
items = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## simDomain

### ОПИСАНИЕ

The `simDomain` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
simDomain = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## simRadialOuterDomain

### ОПИСАНИЕ

The `simRadialOuterDomain` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
simRadialOuterDomain = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

### 5.3.3 Методы

---

Last update: 15 апреля 2025, 20:18

**item()**

**ОПИСАНИЕ**

Описание метода.

**СИНТАКСИС**

```
item(argument: type) : returnType
```

**АРГУМЕНТЫ**


- `argument` (`type`, обязательно): описание аргумента.

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

`returnType` : описание возвращаемого значения.

**ПРИМЕР**

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isLower()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isMiddle()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isUpper()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

## itemAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
itemAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## setItemAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
setItemAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

# 5.4 RotorItem

---

Last update: 15 апреля 2025, 20:18

## 5.4.1 Константы

---

Last update: 15 апреля 2025, 20:23

## ID1

### ОПИСАНИЕ

Константа `ID1`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## ID2

### ОПИСАНИЕ

Константа `ID2`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## ID3

### ОПИСАНИЕ

Константа `ID3`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

```
1 let value = Constants.ID3;  
2 console.info(value);
```



См. также

Last update: 15 апреля 2025, 20:23

## 5.4.2 Свойства

---

Last update: 15 апреля 2025, 20:18

**id**

## ОПИСАНИЕ

The `id` property...

## ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

## ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

## ДОСТУП

Чтение\Запись

## СИНТАКСИС

```
id = value
```

## ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## height

### ОПИСАНИЕ

The `height` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
height = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## angularDisplacement

### ОПИСАНИЕ

The `angularDisplacement` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
angularDisplacement = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

### 5.4.3 Методы

---

Last update: 15 апреля 2025, 20:18

isUpper()

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isMiddle()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isLower()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

## angularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## setAngularDisplacement()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
setAngularDisplacement(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## 5.5 Airgap

---

### 5.5.1 Motor Overview

---

**Description**

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

Last update: 15 апреля 2025, 13:33

## 5.5.2 Свойства

---

### Методы объекта `Geom`

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

Last update: 15 апреля 2025, 21:44

## 5.5.3 Методы

---

### Методы объекта `Airgap`

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

Last update: 15 апреля 2025, 21:44

## 5.6 Winding

---

Last update: 15 апреля 2025, 20:18

## 5.6.1 Константы

---

Last update: 15 апреля 2025, 20:23

## Planar

### ОПИСАНИЕ

Константа `Planar`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Toroidal

### ОПИСАНИЕ

Константа `Toroidal`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## SingleLayer

### ОПИСАНИЕ

Константа `SingleLayer`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23



## DoubleLayer

### ОПИСАНИЕ

Константа `DoubleLayer`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## UpperLower

### ОПИСАНИЕ

Константа `UpperLower`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## LeftRight

### ОПИСАНИЕ

Константа `LeftRight`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Star

### ОПИСАНИЕ

Константа `Star`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Delta

### ОПИСАНИЕ

Константа `Delta`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Lumped

### ОПИСАНИЕ

Константа `Lumped`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## Full

### ОПИСАНИЕ

Константа `Full`.

### ЗНАЧЕНИЕ

`Number`: значение константы.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:23

## 5.6.2 Свойства

---

Last update: 15 апреля 2025, 20:18



## 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)
```



См. также

Last update: 15 апреля 2025, 20:18

## statorConnection

### ОПИСАНИЕ

The `statorConnection` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
statorConnection = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## parallelPaths

### ОПИСАНИЕ

The `parallelPaths` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
parallelPaths = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## numberTurns

### ОПИСАНИЕ

The `numberTurns` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

`Number`

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberTurns = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## strandsConductor

### ОПИСАНИЕ

The `strandsConductor` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
strandsConductor = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## numberOuterSegments

### ОПИСАНИЕ

The `numberOuterSegments` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberOuterSegments = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## 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)
```



См. также

Last update: 15 апреля 2025, 20:18

## numberLayers

### ОПИСАНИЕ

The `numberLayers` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberLayers = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## layersOrientation

### ОПИСАНИЕ

The `layersOrientation` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
layersOrientation = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## autoCalcOverhandEndturns

### ОПИСАНИЕ

The `autoCalcOverhandEndturns` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
autoCalcOverhandEndturns = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## radialOverhandOuterEndturn

### ОПИСАНИЕ

The `radialOverhandOuterEndturn` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
radialOverhandOuterEndturn = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## radialOverhandInnerEndturn

### ОПИСАНИЕ

The `radialOverhandInnerEndturn` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
radialOverhandInnerEndturn = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## heightOuterEndturn

### ОПИСАНИЕ

The `heightOuterEndturn` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
heightOuterEndturn = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## heightInnerEndturn

### ОПИСАНИЕ

The `heightInnerEndturn` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
heightInnerEndturn = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

### 5.6.3 Методы

---

Last update: 15 апреля 2025, 20:18

**isPlanar()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18



**isToroidal()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ


- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isSingleLayer()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isDoubleLayer()**

ОПИСАНИЕ

Описание метода.

СИНТАКСИС

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

АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

ПРИМЕР

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

 **См. также**

Last update: 15 апреля 2025, 20:18

**isOrientationUpperLower()**

**ОПИСАНИЕ**

Описание метода.

**СИНТАКСИС**

```
isOrientationUpperLower(argument: type) : returnType
```

**АРГУМЕНТЫ**


- `argument` (`type`, обязательно): описание аргумента.

**ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ**

`returnType` : описание возвращаемого значения.

**ПРИМЕР**

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

 **См. также**

Last update: 15 апреля 2025, 20:18

## isOrientationLeftRight()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## checkOverlapEndturns()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
checkOverlapEndturns(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## volumeOverlapEndturns()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
volumeOverlapEndturns (argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## distanceBetweenEndturns()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
distanceBetweenEndturns(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18



## 5.7 Mesh

---

Last update: 15 апреля 2025, 20:18

## 5.7.1 Свойства

---

Last update: 15 апреля 2025, 20:18

## autoSizeBound

### ОПИСАНИЕ

The `autoSizeBound` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
autoSizeBound = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## sizeBound

### ОПИСАНИЕ

The `sizeBound` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
sizeBound = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## numberSlices

### ОПИСАНИЕ

The `numberSlices` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

`Number`

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
numberSlices = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## airgapQuality

### ОПИСАНИЕ

The `airgapQuality` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
airgapQuality = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## horizontalSymmetry

### ОПИСАНИЕ

The `horizontalSymmetry` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
horizontalSymmetry = value
```

### ПРИМЕР

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



См. также

Last update: 15 апреля 2025, 20:18

## 5.7.2 Методы

---

### Методы объекта Mesh

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

Last update: 15 апреля 2025, 21:44



## 5.8 EmptyMaterial

---

Last update: 16 апреля 2025, 12:45

## 5.9 GeneralMaterial

---

Last update: 16 апреля 2025, 10:38

## 5.10 IronMaterial

---

Last update: 16 апреля 2025, 10:38

## 5.10.1 Свойства

---

Last update: 16 апреля 2025, 10:38

## fillCoefs

### ОПИСАНИЕ

The `fillCoefs` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
fillCoefs = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:55

**dsomaloy**

ОПИСАНИЕ

The `dsomaloy` property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

ДОСТУП

Чтение\Запись

СИНТАКСИС

```
dsomaloy = value
```

ПРИМЕР

```
1 let m = Material.iron()
2 m.dsomaloy = 0.9
3 let result = m.dsomaloy
4 console.info(result)
```



См. также

Last update: 16 апреля 2025, 15:55

## 5.11 ConductorMaterial

---

Last update: 16 апреля 2025, 12:45

## 5.12 WindingMaterial

---

Last update: 16 апреля 2025, 10:38



### 5.12.1 Свойства

---

Last update: 16 апреля 2025, 10:38

## 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)
```



См. также

Last update: 16 апреля 2025, 15:25

## 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)
```



См. также

Last update: 16 апреля 2025, 15:25

## strand

### ОПИСАНИЕ

The `strand` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
strand = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## windingModel

### ОПИСАНИЕ

The `windingModel` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
windingModel = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.12.2 Методы

---

Last update: 16 апреля 2025, 10:38

## isWindingModelLumped()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
isWindingModelLumped(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## isWindingModelFull()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25



## 5.13 EndturnMaterial

---

Last update: 16 апреля 2025, 10:38

## 5.14 MagnetRadialMaterial

---

Last update: 16 апреля 2025, 10:38

### 5.14.1 Свойства

---

Last update: 16 апреля 2025, 10:38

## direction

### ОПИСАНИЕ

The `direction` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
direction = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## center

### ОПИСАНИЕ

The `center` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
center = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.15 MagnetParallelMaterial

---

Last update: 16 апреля 2025, 10:38

### 5.15.1 Свойства

---

Last update: 16 апреля 2025, 10:38

## 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)
```



См. также

Last update: 16 апреля 2025, 15:25



**segmentRadiuses**

ОПИСАНИЕ

The `segmentRadiuses` property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

ДОСТУП

Чтение\Запись

СИНТАКСИС

```
segmentRadiuses = value
```

ПРИМЕР

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

 **См. также**

Last update: 16 апреля 2025, 15:25

## savePoleBorder

### ОПИСАНИЕ

The `savePoleBorder` property...

### ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

### ТИП ЗНАЧЕНИЯ СВОЙСТВА

Number

### ДОСТУП

Чтение\Запись

### СИНТАКСИС

```
savePoleBorder = value
```

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.16 CustomMaterial

---

Last update: 16 апреля 2025, 12:45

## 5.16.1 Свойства

---

Last update: 16 апреля 2025, 12:45

color

ОПИСАНИЕ

The `color` property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА

`QColor`

ДОСТУП

Чтение\Запись

СИНТАКСИС

```
color = value
```

ПРИМЕР

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

 См. также

Last update: 16 апреля 2025, 15:55

## 5.17 Point3

---

Last update: 16 апреля 2025, 12:45

5.17.1 Свойства

---

Last update: 16 апреля 2025, 12:45

**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)
```



См. также

Last update: 16 апреля 2025, 12:45



## y

### 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)
```



См. также

Last update: 16 апреля 2025, 12:45

**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)
```

 **См. также**

Last update: 16 апреля 2025, 12:45

## 5.17.2 Методы

---

Last update: 16 апреля 2025, 12:45

## distance()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## translate()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## translateX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## translateY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## move()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## moveX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## moveY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## moveZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotate()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotateX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotateY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotateZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorO()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## mirrorX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorXY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorXZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scale()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## scaleY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleXY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleXZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleXYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## 5.18 Vector3

---

Last update: 16 апреля 2025, 12:45

### 5.18.1 Свойства

---

Last update: 16 апреля 2025, 12:45



**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)
```

 **См. также**

Last update: 16 апреля 2025, 12:45

## y

### 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)
```



См. также

Last update: 16 апреля 2025, 12:45

**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)
```

 **См. также**

Last update: 16 апреля 2025, 12:45

## 5.18.2 Методы

---

Last update: 16 апреля 2025, 12:45

## length()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
length(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## length2()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
length2(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

```
1 let result = length2(value)
2 console.info(result)
```



См. также

Last update: 16 апреля 2025, 12:45

## angle()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## isZero()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## 5.19 Shape

---

Last update: 16 апреля 2025, 12:45

### 5.19.1 Свойства

---

Last update: 16 апреля 2025, 12:45

## 5.19.2 Методы

---

Last update: 16 апреля 2025, 12:45

## isEmpty()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## toFileSTEP()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

```
toFileSTEP(argument: type) : returnType
```

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## unite()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## intersect()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## difference()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## diff()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## extrude()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## extrudeX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## extrudeY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## extrudeZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## unify()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## center()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## boundBox()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## translate()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## translateX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## translateY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## move()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## moveX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## moveY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## moveZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotate()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## rotateX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotateY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## rotateZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorO()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorXY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## mirrorYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## mirrorXZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scale()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleX()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleXY()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45



## scaleXZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## scaleXYZ()

### ОПИСАНИЕ

Описание метода.

### СИНТАКСИС

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

### АРГУМЕНТЫ

- `argument` (`type`, обязательно): описание аргумента.

### ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

`returnType` : описание возвращаемого значения.

### ПРИМЕР

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



См. также

Last update: 16 апреля 2025, 12:45

## 5.20 BoundingBox

---

### 5.20.1 Motor Overview

---

#### Description

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

Last update: 15 апреля 2025, 14:50

## 5.20.2 Свойства

---

### Методы объекта **BoundingBox**

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

Last update: 15 апреля 2025, 21:44

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38



propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The `propertyName` property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number

ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38

propertyName

ОПИСАНИЕ

The propertyName property...

ПРИНИМАЕМЫЕ ЗНАЧЕНИЯ:

...

ТИП ЗНАЧЕНИЯ СВОЙСТВА:

Number


ДОСТУП

Чтение\Запись

СИНТАКСИС

ПРИМЕР

1

 См. также

Last update: 16 апреля 2025, 10:38



## 5.20.3 Методы

---

### Методы объекта **BoundingBox**

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

Last update: 15 апреля 2025, 21:44

## 5.21 Piece

---

### 5.21.1 Motor Overview

---

#### Description

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

Last update: 16 апреля 2025, 10:38

## 5.21.2 Свойства

---

### Методы объекта Piece

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

Last update: 16 апреля 2025, 10:38

## 5.21.3 Методы

---

### Методы объекта `Piece`

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

Last update: 16 апреля 2025, 10:38

## 5.22 UI-виджеты

---

Last update: 16 апреля 2025, 15:25

## 5.22.1 WarningIcon()

---

### Описание

Описание метода.

### Синтаксис

```
WarningIcon(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.2 ExclamationIcon()

---

### Описание

Описание метода.

### Синтаксис

```
ExclamationIcon(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.3 NumberEdit()

### Описание

Описание метода.

### Синтаксис

```
NumberEdit(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25



## 5.22.4 NumberSlotSpinBox()

### Описание

Описание метода.

### Синтаксис

```
NumberSlotSpinBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.5 StatorTypeComboBox()

### Описание

Описание метода.

### Синтаксис

```
StatorTypeComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.6 WindingLayersComboBox()

### Описание

Описание метода.

### Синтаксис

```
WindingLayersComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.7 WindingLayersOrientationComboBox()

---

### Описание

Описание метода.

### Синтаксис

```
WindingLayersOrientationComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.8 WindingTypeComboBox()

### Описание

Описание метода.

### Синтаксис

```
WindingTypeComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.9 PoleArrangementComboBox()

---

### Описание

Описание метода.

### Синтаксис

```
PoleArrangementComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.10 StatorConnectionComboBox()

---

### Описание

Описание метода.

### Синтаксис

```
StatorConnectionComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.11 RotorConnectionComboBox()

---

### Описание

Описание метода.

### Синтаксис

```
RotorConnectionComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25



## 5.22.12 QWidget()

---

### Описание

Описание метода.

### Синтаксис

```
QWidget(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.13 QLabel()

---

### Описание

Описание метода.

### Синтаксис

```
QLabel(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.14 QLineEdit()

---

### Описание

Описание метода.

### Синтаксис

```
QLineEdit(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.15 QPushButton()

---

### Описание

Описание метода.

### Синтаксис

```
QPushButton(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.16 QSpinBox()

---

### Описание

Описание метода.

### Синтаксис

```
QSpinBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.17 QDoubleSpinBox()

---

### Описание

Описание метода.

### Синтаксис

```
QDoubleSpinBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.18 QComboBox()

---

### Описание

Описание метода.

### Синтаксис

```
QComboBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.19 QGroupBox()

---

### Описание

Описание метода.

### Синтаксис

```
QGroupBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25



## 5.22.20 QCheckBox()

---

### Описание

Описание метода.

### Синтаксис

```
QCheckBox(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.21 QGridLayout()

---

### Описание

Описание метода.

### Синтаксис

```
QGridLayout(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25

## 5.22.22 QFormLayout()

---

### Описание

Описание метода.

### Синтаксис

```
QFormLayout(argument: type) : returnType
```

### Аргументы

- `argument` (type, обязательно): описание аргумента.

### Возвращаемое значение

`returnType` : описание возвращаемого значения.

### Пример

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



См. также

Last update: 16 апреля 2025, 15:25