jsPDF

jsPDF

new jsPDF(options_opt) → {jsPDF}

Source: jspdf.js, line 75

```
{
 orientation: 'p',
 unit: 'mm',
 format: 'a4',
 putOnlyUsedFonts:true
}
```

Parameters:

Name	Type	Attributes	Description
options	Object	<optional>	Collection of settings initializing the jsPDF-instance *Properties* **Name** / **Type** / **Attributes** / **Default** / **Description** orientation / string / <optional> / portrait / Orientation of the first page. Possible values are "portrait" or "landscape" (or shortcuts "p" or "l"). unit / string / <optional> / mm / Measurement unit (base unit) to be used when coordinates are specified. Possible values are "pt" (points), "mm", "cm", "m", "in" or "px". format / string/Array / <optional> / a4 / The format of the first page. Can be: • a0 - a10 • b0 - b10 • c0 - c10 • dl • letter • government-letter • legal • junior-legal • ledger • tabloid • credit-card. Default is "a4". If you want to use your own format just pass instead of one of the above predefined formats the size as an number-array, e.g. [595.28, 841.89] putOnlyUsedFonts / boolean / <optional> / false / Only put fonts into the PDF, which were used. compress / boolean / <optional> / false / Compress the generated PDF. precision / number / <optional> / 2 / Precision of the element-positions. userUnit / number / <optional> / 1.0 / Not to be confused with the base unit. Please inform yourself before you use it.

Returns:

jsPDF-instance

Type jsPDF

Members

(static) __bidiEngine__

Source: libs/bidiEngine.js, line 188

constructor (options)

Initializes Bidi engine

API

Source: jspdf.js, line 4872

jsPDF.API is a STATIC property of jsPDF class. jsPDF.API is an object you can add methods and properties to. The methods / properties you add will show up in new jsPDF objects.

One property is prepopulated. It is the 'events' Object. Plugin authors can add topics, callbacks to this object. These will be reassigned to all new instances of jsPDF.

Example

```
jsPDF.API.mymethod = function(){
  // 'this' will be ref to internal API object. see jsPDF source
  // , so you can refer to built-in methods like so:
  //     this.line(....)
  //     this.text(....)
}
var pdfdoc = new jsPDF()
```

```
pdfdoc.mymethod() // <- !!!!!!
```

CapJoinStyles

Source: jspdf.js, line 4190

Is an Object providing a mapping from human-readable to integer flag values designating the varieties of line cap and join styles.

clipEvenOdd

Source: jspdf.js, line 3138

Modify the current clip path by intersecting it with the current path using the even-odd rule. Note that this will NOT consume the current path. In order to only use this path for clipping call API.discardPath afterwards.

close

Source: jspdf.js, line 3206

Close the current path. The PDF "h" operator.

comment

Source: jspdf.js, line 4446

Inserts a debug comment into the pdf.

discardPath

Source: jspdf.js, line 3163

Consumes the current path without any effect. Mainly used in combination with clip or clipEvenOdd. The PDF "n" operator.

fill

Source: jspdf.js, line 3228

Fill the current path using the nonzero winding number rule. If a pattern is provided, the path will be filled with this pattern, otherwise with the current fill color. Equivalent to the PDF "f" operator.

fillEvenOdd

Source: jspdf.js, line 3241
See: API.fill

Fill the current path using the even-odd rule. The PDF f* operator.

fillStroke

Source: jspdf.js, line 3254
See: API.fill

Fill using the nonzero winding number rule and then stroke the current Path. The PDF "B" operator.

fillStrokeEvenOdd

Source: jspdf.js, line 3267
See: API.fill

Fill using the even-odd rule and then stroke the current Path. The PDF "B" operator.

#identityMatrix :Matrix

Source: jspdf.js, line 1048

The identity matrix (equivalent to new Matrix(1, 0, 0, 1, 0, 0)).

Type:

- Matrix

lineTo

Source: jspdf.js, line 3394

Append a straight line segment from the current point to the point (x, y). The PDF "l" operator.

matrixMult

Source: jspdf.js, line 1037

Multiplies two matrices. (see Matrix)

moveTo

Source: jspdf.js, line 3383

Begin a new subpath by moving the current point to coordinates (x, y). The PDF "m" operator.

stroke

Source: jspdf.js, line 3217

Stroke the path. The PDF "S" operator.

version :string

Source: jspdf.js, line 4898

The version of jsPDF.

Type:

- string

Methods

addFont()

Source: jspdf.js, line 3833

Properties:

Name	Type	Description
postScriptName	string	PDF specification full name for the font.
id	string	PDF-document-instance-specific label assinged to the font.
fontStyle	string	Style of the Font.
encoding	Object	Encoding_name-to-Font_metrics_object mapping.

Add a custom font to the current instance.

addGState(key, gState) → {jsPDF}

Source: jspdf.js, line 4374

Adds a new GState for later use. See setGState.

Parameters:

Name	Type	Description
key	String	
gState	GState	

Returns:

Type jsPDF

addPage(format, orientation) → {jsPDF}

Source: jspdf.js, line 2431

Adds (and transfers the focus to) new page to the PDF document.

Parameters:

Name	Type	Description
format	String/Array	The format of the new page. Can be: • a0 - a10 • b0 - b10 • c0 - c10 • dl • letter • government-letter • legal • junior-legal • ledger • tabloid • credit-card Default is "a4". If you want to use your own format just pass instead of one of the above predefined formats the size as an number-array, e.g. [595.28, 841.89]
orientation	string	Orientation of the new page. Possible values are "portrait" or "landscape" (or shortcuts "p" (Default), "l").

Returns:

Type jsPDF

beginFormObject(x, y, width, height, matrix) → {jsPDF}

Source: jspdf.js, line 4603

Starts a new pdf form object, which means that all consequent draw calls target a new independent object until endFormObject is called. The created object can be referenced and drawn later using doFormObject. Nested form objects are possible. x, y, width, height set the bounding box that is used to clip the content.

Parameters:

Name	Type	Description
x	number	
y	number	
width	number	
height	number	
matrix	Matrix	The matrix that will be applied to convert the form objects coordinate system to the parent's.

Returns:

Type jsPDF

---

circle(x, y, r, style) → {jsPDF}

Source: jspdf.js, line 3732

Adds an circle to PDF.

Parameters:

Name	Type	Description
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
r	number	Radius (in units declared at inception of PDF document).
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.

Returns:

Type jsPDF

---

clip(rule) → {jsPDF}

Source: jspdf.js, line 3118

All .clip() after calling drawing ops with a style argument of null.

Parameters:

Name	Type	Description
rule	string	Only possible value is 'evenodd'

Returns:

Type jsPDF

---

deletePage(targetPage) → {jsPDF}

Source: jspdf.js, line 2519

Deletes a page from the PDF.

Parameters:

Name	Type	Description
targetPage	number	

Returns:

Type jsPDF

---

doFormObject(key, matrix) → {jsPDF}

Source: jspdf.js, line 4646

Draws the specified form object by referencing to the respective pdf XObject created with API.beginFormObject and endFormObject. The location is determined by matrix.

Parameters:

Name	Type	Description
key	String	The key to the form object.
matrix	Matrix	The matrix applied before drawing the form object.

Returns:

Type jsPDF

---

ellipse(x, y, rx, ry, style) → {jsPDF}

Source: jspdf.js, line 3664

Adds an ellipse to PDF.

Parameters:

Name	Type	Description
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
rx	number	Radius along x axis (in units declared at inception of PDF document).
ry	number	Radius along y axis (in units declared at inception of PDF document).
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.

Returns:

Type jsPDF

---

endFormObject(key) → {jsPDF}

Source: jspdf.js, line 4633

Completes and saves the form object.

Parameters:

Name	Type	Description
key	String	The key by which this form object can be referenced.

Returns:

Type jsPDF

---

getCharSpace() → {number}

Source: jspdf.js, line 4158

Get global value of CharSpace.

Returns:

charSpace

Type number

---

getCreationDate(type) → {Object}

Source: jspdf.js, line 354

Parameters:

Name	Type	Description
type	Object	

Returns:

Type Object

---

getDrawColor() → {string}

Source: jspdf.js, line 3958

Gets the stroke color for upcoming elements.

Returns:

colorAsHex

Type string

---

getFileId() → {string}

Source: jspdf.js, line 267

Returns:

GUID.

Type string

---

getFillColor() → {string}

Source: jspdf.js, line 4026

Gets the fill color for upcoming elements.

Returns:

colorAsHex

Type string

getFont() → {Object}

Source: jspdf.js, line 3771

Gets text font face, variant for upcoming text elements.

Returns:

Type Object

getFontList() → {Object}

Source: jspdf.js, line 3803

Returns an object - a tree of fontName to fontStyle relationships available to active PDF document.

Returns:

Like {'times':['normal', 'italic', ...], 'arial':['normal', 'bold', ...], ... }

Type Object

getFontSize() → {number}

Source: jspdf.js, line 459

Gets the fontsize for upcoming text elements.

Returns:

Type number

getFormObject(key) → {Object|jsPDF}

Source: jspdf.js, line 4668

Returns the form object specified by key.

Parameters:

Name	Type	Description
key	String	

Returns:

- Type Object
- Type jsPDF

getLineHeightFactor() → {number}

Source: jspdf.js, line 3925

Gets the LineHeightFactor, default: 1.15.

Returns:

lineHeightFactor

Type number

getR2L() → {boolean}

Source: jspdf.js, line 490

Get value of R2L functionality.

Returns:

jsPDF-instance

Type boolean

getTextColor() → {string}

Source: jspdf.js, line 4092

Gets the text color for upcoming elements.

Returns:

colorAsHex

Type string

insertPage(beforePage) → {jsPDF}

Source: jspdf.js, line 2469

Parameters:

Name	Type	Description
beforePage	Object	

Returns:

Type jsPDF

---

line(x1, y1, x2, y2, style) → {jsPDF}

Source: jspdf.js, line 3405

Draw a line on the current page.

Parameters:

Name	Type	Description
x1	number	
y1	number	
x2	number	
y2	number	
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument. default: 'S'

Returns:

Type jsPDF

---

lines(lines, x, y, scale, style, closed) → {jsPDF}

Source: jspdf.js, line 3429

Adds series of curves (straight lines or cubic bezier curves) to canvas, starting at `x`, `y` coordinates. All data points in `lines` are relative to last line origin. `x`, `y` become x1,y1 for first line / curve in the set. For lines you only need to specify [x2, y2] - (ending point) vector against x1, y1 starting point. For bezier curves you need to specify [x2,y2,x3,y3,x4,y4] - vectors to control points 1, 2, ending point. All vectors are against the start of the curve - x1,y1.

Example

```
.lines([[2,2],[-2,2],[1,1,2,2,3,3],[2,1]], 212,110, [1,1], 'F', false) // line, line, bezier curve, line
```

Parameters:

Name	Type	Description
lines	Array	Array of *vector* shifts as pairs (lines) or sextets (cubic bezier curves).
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
scale	number	(Defaults to [1.0,1.0]) x,y Scaling factor for all vectors. Elements can be any floating number Sub-one makes drawing smaller. Over-one grows the drawing. Negative flips the direction.
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.
closed	boolean	If true, the path is closed with a straight line from the end of the last curve to the starting point.

Returns:

Type jsPDF

---

lstext(text, x, y, spacing) → {jsPDF}

Source: jspdf.js, line 3082

Deprecated: We'll be removing this function. It doesn't take character width into account.

Letter spacing method to print text with gaps

Parameters:

Name	Type	Description
text	String | Array	String to be added to the page.
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page
spacing	number	Spacing (in units declared at inception)

Returns:

Type jsPDF

---

movePage(targetPage, beforePage) → {jsPDF}

Source: jspdf.js, line 2484

Parameters:

Name	Type	Description
targetPage	number	

Name	Type	Description
beforePage	number	

Returns:

Type jsPDF

---

output(type, options) → {jsPDF}

Source: jspdf.js, line 2235

Generates the PDF document.

If `type` argument is undefined, output is raw body of resulting PDF returned as a string.

Parameters:

Name	Type	Description
type	string	A string identifying one of the possible output types. Possible values are 'arraybuffer', 'blob', 'bloburi'/'bloburl', 'datauristring'/'dataurlstring', 'datauri'/'dataurl', 'dataurlnewwindow', 'pdfobjectnewwindow', 'pdfjsnewwindow'.
options	Object	An object providing some additional signalling to PDF generator. Possible options are 'filename'.

Returns:

Type jsPDF

---

path(lines, style^{opt}, patternKey^{opt}, patternData^{opt}) → {jsPDF}

Source: jspdf.js, line 3519

Similar to API.lines but all coordinates are interpreted as absolute coordinates instead of relative.

Parameters:

Name	Type	Attributes	Description
lines	Array.<Object>		An array of {op: operator, c: coordinates} object, where op is one of "m" (move to), "l" (line to) "c" (cubic bezier curve) and "h" (close (sub)path)). c is an array of coordinates. "m" and "l" expect two, "c" six and "h" an empty array (or undefined).
style	String	<optional>	The style. Deprecated!
patternKey	String	<optional>	The pattern key for the pattern that should be used to fill the path. Deprecated!
patternData	Matrix | PatternData	<optional>	The matrix that transforms the pattern into user space, or an object that will modify the pattern on use. Deprecated!

Returns:

Type jsPDF

---

rect(x, y, w, h, style) → {jsPDF}

Source: jspdf.js, line 3557

Adds a rectangle to PDF.

Parameters:

Name	Type	Description
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
w	number	Width (in units declared at inception of PDF document).
h	number	Height (in units declared at inception of PDF document).
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.

Returns:

Type jsPDF

---

restoreGraphicsState() → {jsPDF}

Source: jspdf.js, line 4410

Restores a previously saved graphics state saved by saveGraphicsState ("pops the stack").

Returns:

Type jsPDF

---

roundedRect(x, y, w, h, rx, ry, style) → {jsPDF}

Source: jspdf.js, line 3625

Adds a rectangle with rounded corners to PDF.

Parameters:

Name	Type	Description
x	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
w	number	Width (in units declared at inception of PDF document).
h	number	Height (in units declared at inception of PDF document).
rx	number	Radius along x axis (in units declared at inception of PDF document).
ry	number	Radius along y axis (in units declared at inception of PDF document).
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.

Returns:

Type jsPDF

---

save(filename, options) → {jsPDF}

Source: jspdf.js, line 4688

Saves as PDF document. An alias of jsPDF.output('save', 'filename.pdf'). Uses FileSaver.js-method saveAs.

Parameters:

Name	Type	Description
filename	string	The filename including extension.
options	Object	An Object with additional options, possible options: 'returnPromise'.

Returns:

jsPDF-instance

Type jsPDF

---

saveGraphicsState() → {jsPDF}

Source: jspdf.js, line 4390

Saves the current graphics state ("pushes it on the stack"). It can be restored by restoreGraphicsState later. Here, the general pdf graphics state is meant, also including the current transformation matrix, fill and stroke colors etc.

Returns:

Type jsPDF

---

setCharSpace(charSpace) → {jsPDF}

Source: jspdf.js, line 4171

Set global value of CharSpace.

Parameters:

Name	Type	Description
charSpace	number	

Returns:

jsPDF-instance

Type jsPDF

---

setCreationDate(date) → {jsPDF}

Source: jspdf.js, line 341

Parameters:

Name	Type	Description
date	Object	

Returns:

Type jsPDF

---

setCurrentTransformationMatrix(matrix) → {jsPDF}

Source: jspdf.js, line 4431

Appends this matrix to the left of all previously applied matrices.

Parameters:

Name	Type	Description
matrix	Matrix	

Returns:

Type jsPDF

---

setDisplayMode(zoom, layout, pmode) → {jsPDF}

Source: jspdf.js, line 551

Set the display mode options of the page like zoom and layout.

Parameters:

Name	Type	Description
zoom	integer | String	You can pass an integer or percentage as a string. 2 will scale the document up 2x, '200%' will scale up by the same amount. You can also set it to 'fullwidth', 'fullheight', 'fullpage', or 'original'. Only certain PDF readers support this, such as Adobe Acrobat.
layout	string	Layout mode can be: 'continuous' - this is the default continuous scroll. 'single' - the single page mode only shows one page at a time. 'twoleft' - two column left mode, first page starts on the left, and 'tworight' - pages are laid out in two columns, with the first page on the right. This would be used for books.
pmode	string	'UseOutlines' - it shows the outline of the document on the left. 'UseThumbs' - shows thumbnails along the left. 'FullScreen' - prompts the user to enter fullscreen mode.

Returns:

Type jsPDF

---

setDocumentProperties(A) → {jsPDF}

Source: jspdf.js, line 602

Adds a properties to the PDF document.

Parameters:

Name	Type	Description
A	Object	property_name-to-property_value object structure.

Returns:

Type jsPDF

---

setDrawColor(ch1, ch2, ch3, ch4) → {jsPDF}

Source: jspdf.js, line 3971

Sets the stroke color for upcoming elements.

Depending on the number of arguments given, Gray, RGB, or CMYK color space is implied.

When only ch1 is given, "Gray" color space is implied and it must be a value in the range from 0.00 (solid black) to to 1.00 (white) if values are communicated as String types, or in range from 0 (black) to 255 (white) if communicated as Number type. The RGB-like 0-255 range is provided for backward compatibility.

When only ch1,ch2,ch3 are given, "RGB" color space is implied and each value must be in the range from 0.00 (minimum intensity) to to 1.00 (max intensity) if values are communicated as String types, or from 0 (min intensity) to to 255 (max intensity) if values are communicated as Number types. The RGB-like 0-255 range is provided for backward compatibility.

When ch1,ch2,ch3,ch4 are given, "CMYK" color space is implied and each value must be a in the range from 0.00 (0% concentration) to to 1.00 (100% concentration)

Because JavaScript treats fixed point numbers badly (rounds to floating point nearest to binary representation) it is highly advised to communicate the fractional numbers as String types, not JavaScript Number type.

Parameters:

Name	Type	Description
ch1	Number | String	Color channel value or {string} ch1 color value in hexadecimal, example: '#FFFFFF'.
ch2	Number	Color channel value.
ch3	Number	Color channel value.
ch4	Number	Color channel value.

Returns:

Type jsPDF

---

setFileId(value) → {jsPDF}

Source: jspdf.js, line 254

Parameters:

Name	Type	Description
value	string	GUID.

Returns:

Type jsPDF

---

setFillColor(ch1, ch2, ch3, ch4) → {jsPDF}

Source: jspdf.js, line 4038

Sets the fill color for upcoming elements.

Depending on the number of arguments given, Gray, RGB, or CMYK color space is implied.

When only ch1 is given, "Gray" color space is implied and it must be a value in the range from 0.00 (solid black) to to 1.00 (white) if values are communicated as String types, or in range from 0 (black) to 255 (white) if communicated as Number type. The RGB-like 0-255 range is provided for backward compatibility.

When only ch1,ch2,ch3 are given, "RGB" color space is implied and each value must be in the range from 0.00 (minimum intensity) to to 1.00 (max intensity) if values are communicated as String types, or from 0 (min intensity) to to 255 (max intensity) if values are communicated as Number types. The RGB-like 0-255 range is provided for backward compatibility.

When ch1,ch2,ch3,ch4 are given, "CMYK" color space is implied and each value must be a in the range from 0.00 (0% concentration) to to 1.00 (100% concentration)

Because JavaScript treats fixed point numbers badly (rounds to floating point nearest to binary representation) it is highly advised to communicate the fractional numbers as String types, not JavaScript Number type.

Parameters:

Name	Type	Description
ch1	Number | String	Color channel value or {string} ch1 color value in hexadecimal, example: '#FFFFFF'.
ch2	Number	Color channel value.
ch3	Number	Color channel value.
ch4	Number	Color channel value.

Returns:

Type jsPDF

---

setFont(fontName, fontStyle) → {jsPDF}

Source: jspdf.js, line 3752

Sets text font face, variant for upcoming text elements. See output of jsPDF.getFontList() for possible font names, styles.

Parameters:

Name	Type	Description
fontName	string	Font name or family. Example: "times".
fontStyle	string	Font style or variant. Example: "italic".

Returns:

Type jsPDF

---

setFontSize(size) → {jsPDF}

Source: jspdf.js, line 444

Sets font size for upcoming text elements.

Parameters:

Name	Type	Description
size	number	Font size in points.

Returns:

Type jsPDF

---

setFontStyle(style) → {jsPDF}

Source: jspdf.js, line 3784

Deprecated:

Switches font style or variant for upcoming text elements, while keeping the font face or family same. See output of jsPDF.getFontList() for possible font names, styles.

Parameters:

Name	Type	Description
style	string	Font style or variant. Example: "italic".

Returns:

Type jsPDF

---

setGState(gState) → {jsPDF}

Source: jspdf.js, line 4318

Sets a either previously added GState (via addGState) or a new GState.

Parameters:

Name	Type	Description
gState	String | GState	If type is string, a previously added GState is used, if type is GState it will be added before use.

Returns:

Type jsPDF

setLineCap(style) → {jsPDF}

Source: jspdf.js, line 4214

Sets the line cap styles. See {jsPDF.CapJoinStyles} for variants.

Parameters:

Name	Type	Description
style	String | Number	A string or number identifying the type of line cap.

Returns:

Type jsPDF

setLineDashPattern(dashArray, dashPhase) → {jsPDF}

Source: jspdf.js, line 3866

Sets the dash pattern for upcoming lines.

To reset the settings simply call the method without any parameters.

Parameters:

Name	Type	Description
dashArray	Array.<number>	An array containing 0-2 numbers. The first number sets the length of the dashes, the second number the length of the gaps. If the second number is missing, the gaps are considered to be as long as the dashes. An empty array means solid, unbroken lines.
dashPhase	number	The phase lines start with.

Returns:

Type jsPDF

setLineHeightFactor(value) → {jsPDF}

Source: jspdf.js, line 3907

Sets the LineHeightFactor of proportion.

Parameters:

Name	Type	Description
value	number	LineHeightFactor value. Default: 1.15.

Returns:

Type jsPDF

setLineJoin(style) → {jsPDF}

Source: jspdf.js, line 4237

Sets the line join styles. See {jsPDF.CapJoinStyles} for variants.

Parameters:

Name	Type	Description
style	String | Number	A string or number identifying the type of line join.

Returns:

Type jsPDF

setLineMiterLimit(length) → {jsPDF}

Source: jspdf.js, line 4260

Sets the miterLimit property, which effects the maximum miter length.

Parameters:

Name	Type	Description
length	number	The length of the miter

Returns:

Type jsPDF

setLineWidth(width) → {jsPDF}

Source: jspdf.js, line 3851

Sets line width for upcoming lines.

Parameters:

Name	Type	Description

Name	Type	Description
width	number	Line width (in units declared at inception of PDF document).

Returns:

Type jsPDF

setPage(page) → {jsPDF}

Source: jspdf.js, line 2447

Adds (and transfers the focus to) new page to the PDF document.

Example

```
doc = jsPDF()
doc.addPage()
doc.addPage()
doc.text('I am on page 3', 10, 10)
doc.setPage(1)
doc.text('I am on page 1', 10, 10)
```

Parameters:

Name	Type	Description
page	number	Switch the active page to the page number specified.

Returns:

Type jsPDF

setR2L(value) → {jsPDF}

Source: jspdf.js, line 475

Set value of R2L functionality.

Parameters:

Name	Type	Description
value	boolean	

Returns:

jsPDF-instance

Type jsPDF

setTextColor(ch1, ch2, ch3, ch4) → {jsPDF}

Source: jspdf.js, line 4104

Sets the text color for upcoming elements.

Depending on the number of arguments given, Gray, RGB, or CMYK color space is implied.

When only ch1 is given, "Gray" color space is implied and it must be a value in the range from 0.00 (solid black) to to 1.00 (white) if values are communicated as String types, or in range from 0 (black) to 255 (white) if communicated as Number type. The RGB-like 0-255 range is provided for backward compatibility.

When only ch1,ch2,ch3 are given, "RGB" color space is implied and each value must be in the range from 0.00 (minimum intensity) to to 1.00 (max intensity) if values are communicated as String types, or from 0 (min intensity) to to 255 (max intensity) if values are communicated as Number types. The RGB-like 0-255 range is provided for backward compatibility.

When ch1,ch2,ch3,ch4 are given, "CMYK" color space is implied and each value must be a in the range from 0.00 (0% concentration) to to 1.00 (100% concentration)

Because JavaScript treats fixed point numbers badly (rounds to floating point nearest to binary representation) it is highly advised to communicate the fractional numbers as String types, not JavaScript Number type.

Parameters:

Name	Type	Description
ch1	Number | String	Color channel value or {string} ch1 color value in hexadecimal, example: '#FFFFFF'.
ch2	Number	Color channel value.
ch3	Number	Color channel value.
ch4	Number	Color channel value.

Returns:

Type jsPDF

text(text, x, y, options_opt, transform) → {jsPDF}

Source: jspdf.js, line 2533

Adds text to page. Supports adding multiline text when 'text' argument is an Array of Strings.

Parameters:

Name	Type	Attributes	Description

Name	Type	Attributes	Description
text	String | Array		String or array of strings to be added to the page. Each line is shifted one line down per font, spacing settings declared before this call.
x	number		Coordinate (in units declared at inception of PDF document) against left edge of the page.
y	number		Coordinate (in units declared at inception of PDF document) against upper edge of the page.
options	Object	<optional>	Collection of settings signaling how the text must be encoded. *Properties* **Name** / **Type** / **Attributes** / **Default** / **Description** align — string — <optional> — left — The alignment of the text, possible values: left, center, right, justify. baseline — string — <optional> — alphabetic — Sets text baseline used when drawing the text, possible values: alphabetic, ideographic, bottom, top, middle, hanging angle — string — <optional> — 0 — Rotate the text clockwise or counterclockwise. Expects the angle in degree. rotationDirection — string — <optional> — 1 — Direction of the rotation. 0 = clockwise, 1 = counterclockwise. charSpace — string — <optional> — 0 — The space between each letter. lineHeightFactor — string — <optional> — 1.15 — The lineheight of each line. flags — string — <optional> — Flags for to8bitStream. *Properties* **Name** / **Type** / **Attributes** / **Default** / **Description** noBOM — string — <optional> — true — Don't add BOM to Unicode-text. autoencode — string — <optional> — true — Autoencode the Text. maxWidth — string — <optional> — 0 — Split the text by given width, 0 = no split. renderingMode — string — <optional> — fill — Set how the text should be rendered, possible values: fill, stroke, fillThenStroke, invisible, fillAndAddForClipping, strokeAndAddPathForClipping, fillThenStrokeAndAddToPathForClipping, addToPathForClipping.
transform	number | Matrix		If transform is a number the text will be rotated by this value around the anchor set by x and y. If it is a Matrix, this matrix gets directly applied to the text, which allows shearing effects etc.; the x and y offsets are then applied AFTER the coordinate system has been established by this matrix. This means passing a rotation matrix that is equivalent to some rotation angle will in general yield a DIFFERENT result. A matrix is only allowed in "advanced" API mode.

Returns:

Type jsPDF

---

triangle(x1, y1, x2, y2, x3, y3, style) → {jsPDF}

Source: jspdf.js, line 3591

Adds a triangle to PDF.

Parameters:

Name	Type	Description
x1	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y1	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
x2	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y2	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
x3	number	Coordinate (in units declared at inception of PDF document) against left edge of the page.
y3	number	Coordinate (in units declared at inception of PDF document) against upper edge of the page.
style	string	A string specifying the painting style or null. Valid styles include: 'S' [default] - stroke, 'F' - fill, and 'DF' (or 'FD') - fill then stroke. A null value postpones setting the style so that a shape may be composed using multiple method calls. The last drawing method call used to define the shape should not have a null style argument.

Returns:

Type jsPDF

Documentation generated by JSDoc 3.5.5 on Tue Apr 02 2019 00:25:07 GMT+0200 (GMT+02:00) using the docdash theme.