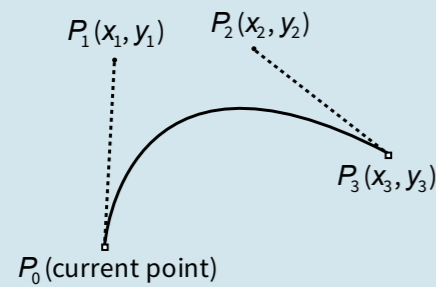


# PDF – Graphic Operators – Cheat Sheet

## Vector Graphics

### Path Construction

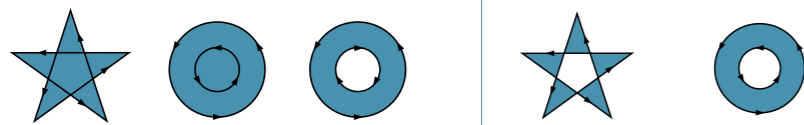
- A PDF vector *path* has a *current point* and can have multiple independent *subpaths*, each of which can have multiple *segments* (curves, lines). Subpaths can be open or closed.
- Cubic Bézier curve control points (**c** operator example):



$x_1 y_1 x_2 y_2 x_3 y_3$	<b>c</b>	Add a curved Bézier segment to the current subpath using 3 control points, as shown above. New current point is then $(x_3, y_3)$ .
	<b>h</b>	Close the current subpath by adding a straight-line segment from the current point to the start point of the current subpath. Next operator must be <b>m</b> , <b>re</b> to start new subpath, or a path painting operator to paint the path.
$x y$	<b>l</b>	<b>Line-to.</b> (Lowercase L) Add a straight-line segment from the current point to $(x, y)$ . New current point is then $(x, y)$ .
$x y$	<b>m</b>	<b>Move-to.</b> Begin a new subpath in the current path by setting the current point to $(x, y)$ .
$x y width height$	<b>re</b>	<b>Rectangle.</b> Add a closed rectangle subpath with lower-left corner at $(x, y)$ and dimensions <i>width</i> and <i>height</i> . Next operator must be <b>m</b> , <b>re</b> to start new subpath, or a path painting operator to paint the path.
$x_2 y_2 x_3 y_3$	<b>v</b>	Add a curved Bézier segment to the subpath from current point to $(x_3, y_3)$ , using the current point and $(x_2, y_2)$ as the Bézier control points. New current point is then $(x_3, y_3)$ .
$x_1 y_1 x_3 y_3$	<b>y</b>	Add a curved Bézier segment to the current subpath. The curve extends from current point to the point $(x_3, y_3)$ , using $(x_1, y_1)$ and $(x_3, y_3)$ as the Bézier control points. New current point is then $(x_3, y_3)$ .

### Path Painting

Combined fill and stroke painting must be treated as a single atomic graphics object.



Non-zero Winding Rule

Even-odd Winding Rule

<b>b</b>	Close, fill, and stroke path using the non-zero winding rule. Same as: <b>h B</b>
<b>B</b>	Fill and stroke path using the non-zero winding rule.
<b>b*</b>	Close, fill, and stroke path using the even-odd winding rule. Same as: <b>h B*</b>
<b>B*</b>	Fill and stroke path using the even-odd winding rule.
<b>f</b>	Close all subpaths then fill the current path using non-zero winding rule.
<b>f*</b>	Fill path using even-odd winding rule.
<b>F</b>	Fill the current path using non-zero winding number rule. (Deprecated in PDF 2.0)
<b>n</b>	End path without filling or stroking (“no op”). Used after <b>W</b> / <b>W*</b> operators to establish the new clipping path.
<b>s</b>	Close and stroke the current path. Same as <b>h S</b> .
<b>S</b>	Stroke the current path.

### Clipping

<b>W</b>	Modify the current clipping path by intersecting it with the current path, using the non-zero winding number rule to determine which regions lie inside the clipping path. Initial clipping path is the page <b>MediaBox</b> .
<b>W*</b>	Modify the current clip path by intersecting it with the current path using the even-odd winding rule to determine which regions lie inside the clipping path. Initial clipping path is the page <b>MediaBox</b> .

## Text

### Text Object

<b>BT</b>	Begin text object. Nested.
<b>ET</b>	End text object. After filling/stroking and if <b>Tr</b> was set to a text clipping mode (4 - 7), then change the clipping path to the intersection of the current clipping path and any glyph-based clipping path using the Non-zero Winding rule. Must be paired with <b>BT</b> operator.

### Text State

<b>Text knockout</b> can only be set via graphics state parameter dictionary <b>TK</b> entry.																													
<i>number</i>	<b>Tc</b>	Set character spacing in unscaled text space units to <i>number</i> . Character spacing is used by <b>Tj</b> , <b>TJ</b> and ' text showing operators. Initial value: 0.																											
<i>name size</i>	<b>Tf</b>	Set text font and size (number) in the graphics state. <i>name</i> is the name of a font resource in the <b>Font</b> subdictionary of the current resource dictionary. Zero sized text does not mark or clip any pixels. There are no default / initial values. Equivalent to the <b>Font</b> entry (array) in the graphics state parameter dictionary.																											
<i>number</i>	<b>TL</b>	Set text leading to <i>number</i> expressed in unscaled text space units. Text leading is only used by <b>T*</b> , " and ' text showing operators. Initial value: 0.																											
<i>mode</i>	<b>Tr</b>	Set text rendering mode (integer). Initial value is 0 (filled text). Once set to a clipping mode (4-7), cannot change back before <b>ET</b> .																											
		<table border="1"> <thead> <tr> <th>Mode</th> <th>Description</th> <th>Example</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Filled text.</td> <td>R</td> </tr> <tr> <td>1</td> <td>Stroked text.</td> <td>R</td> </tr> <tr> <td>2</td> <td>Fill, then stroke text.</td> <td>R</td> </tr> <tr> <td>3</td> <td>Invisible. Neither fill nor stroke text. Text will still be selectable/searchable.</td> <td></td> </tr> <tr> <td>4</td> <td>Fill text and add to path for clipping.</td> <td>R</td> </tr> <tr> <td>5</td> <td>Stroke text and add to path for clipping.</td> <td>R</td> </tr> <tr> <td>6</td> <td>Fill, then stroke and add to path for clipping.</td> <td>R</td> </tr> <tr> <td>7</td> <td>Add text to path for clipping.</td> <td>R</td> </tr> </tbody> </table>	Mode	Description	Example	0	Filled text.	R	1	Stroked text.	R	2	Fill, then stroke text.	R	3	Invisible. Neither fill nor stroke text. Text will still be selectable/searchable.		4	Fill text and add to path for clipping.	R	5	Stroke text and add to path for clipping.	R	6	Fill, then stroke and add to path for clipping.	R	7	Add text to path for clipping.	R
Mode	Description	Example																											
0	Filled text.	R																											
1	Stroked text.	R																											
2	Fill, then stroke text.	R																											
3	Invisible. Neither fill nor stroke text. Text will still be selectable/searchable.																												
4	Fill text and add to path for clipping.	R																											
5	Stroke text and add to path for clipping.	R																											
6	Fill, then stroke and add to path for clipping.	R																											
7	Add text to path for clipping.	R																											
<i>number</i>	<b>Ts</b>	Set text rise expressed in unscaled text space units. Initial value: 0.																											
<i>number</i>	<b>Tw</b>	Set word spacing in unscaled text space units. Word spacing is used by <b>Tj</b> , <b>TJ</b> and ' text showing operators. Initial value: 0.																											
<i>number</i>	<b>Tz</b>	Set horizontal text scaling specified as a percentage of normal width ( <i>number</i> > 0). Initial value: 100 (100% is normal width).																											

### Text Positioning

	<b>T*</b>	Move to start of next text line.
$t_x t_y$	<b>Td</b>	Move to the start of the next line, offset from the start of the current line by $(t_x, t_y)$ . $t_x$ and $t_y$ are numbers expressed in unscaled text space units.
$t_x t_y$	<b>TD</b>	Move to the start of the next line, offset from the start of the current line by $(t_x, t_y)$ . As a side effect, also set the leading parameter in the text state. <b>TD</b> is equivalent to: $-t_y$ <b>TL</b> $t_x t_y$ <b>Td</b>
$a b c d e f$	<b>Tm</b>	Set text matrix and text line matrix.

### Text Showing

<i>string</i>	<b>Tj</b>	Show text <i>string</i> . <i>string</i> comprises glyph IDs.
[ <i>string number ...</i> ]	<b>TJ</b>	Show text allowing individual glyph positioning. Each element in array is either a string (glyph IDs), or a number representing a text adjustment that is <b>subtracted</b> from the current horizontal or vertical coordinate, depending on the writing mode.
<i>string</i>	'	Move to the next line and show text <i>string</i> .
$a_w a_c string$	"	Set word and character spacing to $a_w$ and $a_c$ numbers respectively, move to next line, and show text <i>string</i> .

### Type 3 fonts

Must always be the first operator in a Type 3 glyph description content stream. The number $w_x$ is the horizontal displacement and the number $w_y$ is the vertical displacement in the glyph coordinate system.		
$w_x w_y$	<b>d0</b>	(Ends in digit zero). Set width information for a Type 3 glyph description and declare that it specifies <b>both its shape and color</b> .
$w_x w_y ll_x ll_y ur_x ur_y$	<b>d1</b>	(Ends in digit 1). Set width and bounding box information for a Type 3 glyph description and it specifies <b>only shape</b> and not color.

## Marked Content

### Marked Content Sequences and Points

See subclause 14.6 in ISO 32000-2:2020. Introduced in PDF 1.2.		
<b>EMC</b> must be paired with either a <b>BDC</b> or <b>BMC</b> operator and nested correctly with <b>BT</b> / <b>ET</b> text object, <b>BX</b> / <b>EX</b> compatibility operators, and <b>q</b> / <b>Q</b> paired operators.		
<i>name property</i>	<b>BDC</b>	Begin marked-content sequence with property list. Nested. <i>name</i> is a name object indicating the role or significance of the sequence. <i>property</i> is either an inline dictionary or a name of a resource in the <b>Properties</b> subdictionary of the current resource dictionary.
<i>name</i>	<b>BMC</b>	Begin a marked-content sequence. Nested. <i>name</i> is a name indicating the role or significance of the sequence
<i>name property</i>	<b>DP</b>	Define a marked-content point with property list. <i>name</i> is a name object indicating the role or significance of the point. <i>property</i> is either an inline dictionary or the name of a resource in the <b>Properties</b> subdictionary of the current resource dictionary.
	<b>EMC</b>	End marked-content sequence.
<i>name</i>	<b>MP</b>	Define a marked-content point. <i>tag</i> is a name indicating the role or significance of the sequence

## Graphics State

**Coordinate Systems**

Relationship among PDF coordinate systems:

$a\ b\ c\ d\ e\ f$	<b>cm</b>	Concatenate specified matrix to current transformation matrix. Operands are numbers. Matrix is: $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$ Initial value is a matrix that transforms default user coordinates ( $1/72$ inch) into device coordinates.
--------------------	-----------	---

**Graphics State Parameters**

$[on\ off\ \dots]\ phase$	<b>d</b>	Set stroking dash pattern to the specified dash array ( <i>on</i> , <i>off</i> , <i>on</i> , ... - numbers) and dash phase ( <i>number</i> ). Equivalent to the <b>D</b> entry in the graphics state parameter dictionary. Initial value is <b>[ ]</b> (solid undashed line).  <table border="1"> <tr> <td><math>[3]\ 0</math></td> <td></td> </tr> <tr> <td><math>[2\ 1\ 3]\ -2</math></td> <td></td> </tr> </table>	$[3]\ 0$		$[2\ 1\ 3]\ -2$									
$[3]\ 0$														
$[2\ 1\ 3]\ -2$														
<i>name</i>	<b>gs</b>	(PDF 1.2) Set parameters from the named graphics state parameter dictionary. <i>name</i> is the name of a graphics state parameter dictionary in the <b>ExtGState</b> subdictionary of the current resource dictionary.												
<i>number</i>	<b>i</b>	Set flatness tolerance to <i>number</i> . Equivalent to the <b>FL</b> entry in the graphics state parameter dictionary. Initial value is 1.0.												
<i>style</i>	<b>j</b>	(Lowercase "j") Set line join style (integer). Equivalent to the <b>LJ</b> entry in the graphics state parameter dictionary. Initial value is 0 (miter join).  <table border="1"> <thead> <tr> <th>Style</th> <th>Name</th> <th>Appearance</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Miter join</td> <td></td> </tr> <tr> <td>1</td> <td>Round join</td> <td></td> </tr> <tr> <td>2</td> <td>Bevel join</td> <td></td> </tr> </tbody> </table>	Style	Name	Appearance	0	Miter join		1	Round join		2	Bevel join	
Style	Name	Appearance												
0	Miter join													
1	Round join													
2	Bevel join													

<i>style</i>	<b>J</b>	(Uppercase J) Set line cap style (integer). Equivalent to the <b>LC</b> entry in the graphics state parameter dictionary. Initial value is 0 (butt cap).  <table border="1"> <thead> <tr> <th>Style</th> <th>Name</th> <th>Appearance</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Butt cap</td> <td></td> </tr> <tr> <td>1</td> <td>Round cap</td> <td></td> </tr> <tr> <td>2</td> <td>Projecting square cap</td> <td></td> </tr> </tbody> </table>	Style	Name	Appearance	0	Butt cap		1	Round cap		2	Projecting square cap	
Style	Name	Appearance												
0	Butt cap													
1	Round cap													
2	Projecting square cap													
<i>number</i>	<b>M</b>	Set the miter limit ratio. When line width is zero, the miter length is zero. Equivalent to the <b>ML</b> entry in the graphics state parameter dictionary. Initial value is 10.0, for a miter cutoff below approx. $\theta = 11.5^\circ$ .   $\text{miter limit ratio} = \frac{\text{miter length}}{\text{line width}} = \frac{1}{\sin \frac{\theta}{2}}$												
	<b>q</b>	Save graphics state ("push"). Nested. Needs to be paired with a <b>Q</b> operator.												
	<b>Q</b>	Restore graphics state ("pop").												
<i>name</i>	<b>ri</b>	Set the color render intent in the graphics state. <i>intent</i> is a name and usually one of <i>AbsoluteColorimetric</i> , <i>RelativeColorimetric</i> , <i>Saturation</i> , or <i>Perceptual</i> . Equivalent to the <b>RI</b> entry in the graphics state parameter dictionary. Initial value is <i>RelativeColorimetric</i> .												
<i>number</i>	<b>w</b>	Set line width to <i>number</i> in user space units ( $number \geq 0$ ). Initial value is 1.0.												

## Other operators

**Inline Images**

Only for very small images (<4KB). Otherwise use Image XObject and **Do** operator.

<b>BI</b>	Begin inline image object. Followed by Image XObject dictionary key value pairs. Certain key names and values may also be abbreviated. Abbreviated key names take precedence over full key names.
<b>ID</b>	Begin inline image data after a single whitespace character.
<b>EI</b>	Ends an inline image object.

**Object painting**

<i>name</i>	<b>Do</b>	Invoke (paint) the named XObject. <i>name</i> is the name of an XObject that is in the <b>XObject</b> subdictionary of the current resource dictionary.
<i>name</i>	<b>sh</b>	(PDF 1.3) Paint area defined by a shading pattern. <i>name</i> is the name of a shading dictionary resource in the <b>Shading</b> subdictionary of the current resource dictionary.

**Compatibility Sections**

Introduced in PDF 1.1.

<b>BX</b>	Begin compatibility section. Nested. Unrecognised operators (along with all operands) will be ignored <u>without error</u> until the balancing <b>EX</b> operator.
<b>EX</b>	End compatibility section. Must be paired with <b>BX</b> operator.

All paired operators must be nested correctly: **BDC/EMC** or **BMC/EMC** marked content; **BT/ET** text object, **BX/EX** compatibility section, and **q/Q** graphics stack.

## Color

**Color Operators**

UPPERCASE = stroking  
lowercase = filling (non-stroking)

<i>name</i>	<b>CS</b>	(PDF 1.1) Set color space for stroking ( <b>CS</b> ) or non-stroking ( <b>cs</b> ) operations. If the color space is one that can be specified by a name and no additional parameters ( <b>DeviceGray</b> , <b>DeviceRGB</b> , <b>DeviceCMYK</b> , and certain cases of <b>Pattern</b> ), that name may be specified directly. Otherwise, <i>name</i> is a resource in the <b>ColorSpace</b> subdictionary of the current resource dictionary. Initial color space is <b>DeviceGray</b> .
<i>gray</i>	<b>G</b>	Set gray level for stroking ( <b>G</b> ) or non-stroking ( <b>g</b> ) operations. $0.0 \text{ (black)} \leq \text{gray} \leq 1.0 \text{ (white)}$
$c\ m\ y\ k$	<b>K</b>	Set CMYK levels for stroking ( <b>K</b> ) or non-stroking ( <b>k</b> ) operations. $0.0 \leq c, m, y, k \leq 1.0$
$r\ g\ b$	<b>RG</b>	Set RGB levels for stroking ( <b>RG</b> ) or non-stroking ( <b>rg</b> ) operations. $0.0 \text{ (no color)} \leq r, g, b \leq 1.0 \text{ (max. color)}$
$C_1 \dots C_n$	<b>SC</b>	(PDF 1.1) Set the color to use for stroking operations ( <b>SC</b> ) or non-stroking ( <b>sc</b> ) to CIE-based (other than <b>ICCBased</b> ), or <b>Indexed</b> color space. The number of operands required, and their interpretation depends on the current stroking/non-stroking color space: For <b>DeviceGray</b> , <b>CalGray</b> , and <b>Indexed</b> color spaces, one operand is required ( $n = 1$ ). For <b>DeviceRGB</b> , <b>CalRGB</b> , and <b>Lab</b> color spaces, three operands are required ( $n = 3$ ). For <b>DeviceCMYK</b> , four operands are required ( $n = 4$ )
$C_1 \dots C_n$	<b>SCN</b>	(PDF 1.2) Same as <b>SC/sc</b> operators but also for <b>Pattern</b> , <b>Separation</b> , <b>DeviceN</b> and <b>ICCBased</b> color spaces. If the current stroking ( <b>SCN</b> ) or non-stroking ( <b>scn</b> ) color space is a <b>Separation</b> , <b>DeviceN</b> , or <b>ICCBased</b> color space, the operands $C_1 \dots C_n$ are numbers. The number of operands and their interpretation depends on the color space. If the current color space is a <b>Pattern</b> color space, then <i>name</i> is the name of an entry in the <b>Pattern</b> subdictionary of the current resource dictionary. For an uncolored tiling pattern ( <b>PatternType</b> =1 and <b>PaintType</b> =2), $C_1 \dots C_n$ are component values specifying a color in the pattern's underlying color space. For other types of patterns, these operands are not specified.