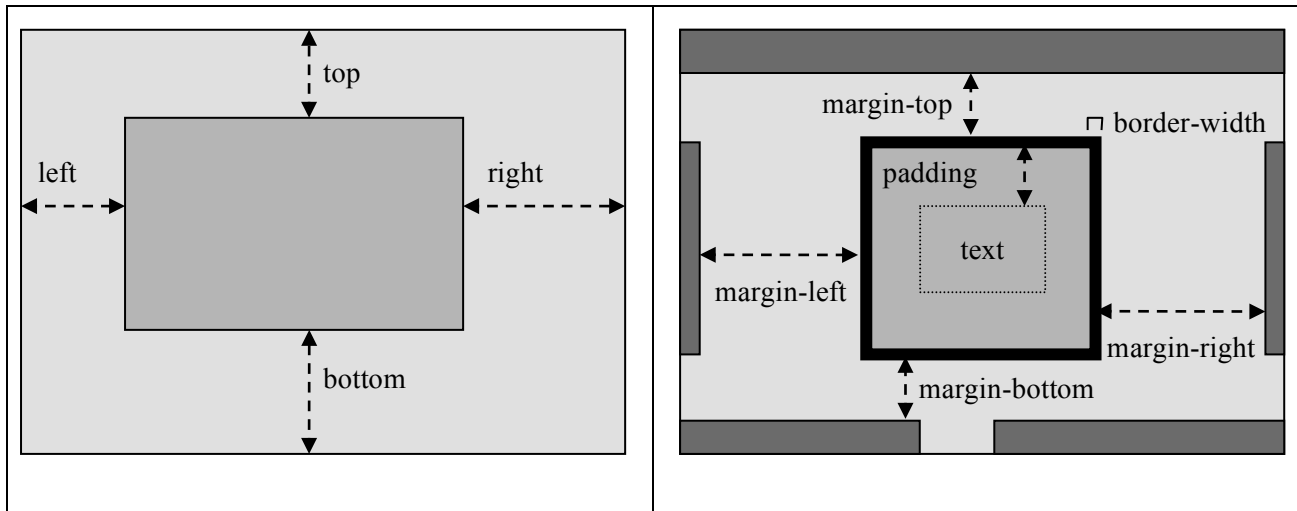


# d3 cheat sheet

## HTML

### Position an element :



### Positioning :

			style	attr
position	static	The default mode, block element is positioned in the flow. Top, left etc. are ignored.	•	
position	relative	The block element is positioned relative to its position in the flow.	•	
position	absolute	Block element is positioned relative to its container.	•	
position	fixed	Block element is positioned relative to the window and won't scroll.	•	
top	Number [px, cm, in...]	Positions the block <i>down</i> from the reference point at the specified distance.	•	
bottom	Number [px, cm, in...]	Positions the block <i>up</i> from the reference point at the specified distance.	•	
left	Number [px, cm, in...]	Positions the block <i>right</i> from the reference point at the specified distance.	•	
right	Number [px, cm, in...]	Positions the block <i>left</i> from the reference point at the specified distance.	•	

```
{position: absolute; top: 10px; left: 50px;}
```

### Sizing :

			style	attr
width	Number [+unit]	Width of the block.	•	
height	Number [+unit]	Height of the block.	•	
padding	1-4 numbers [+unit]	Additional space added inside the block, from the border in. If only one argument is provided, the same amount of space is applied in 4 directions. If 2 arguments are given, the first applies to top and bottom, the second to left/right. If 4 arguments are given they are in order top right bottom left.	•	
margin	1-4 numbers [+unit]	Additional space added outside the block, from the border out.	•	

```
{width: 100px; padding: 10px; margin: 5 10px;}
```

### Colors :

			style	attr
color	name, #RRGGBB, rgb(R, G, B)	Specifies the color of the text. The color can be a name ("steelblue"), a hex number ("4682b4"), or a RGB triplet ("rgb(70,130,180)")	•	
border-color	color	Specifies the color of the border, or outline of the block element.	•	
background-color	color	Specifies the color of the background, or fill of the block element.	•	

```
{color: red; border-color: "#FF8C00"; background-color: "rgb(255, 192, 203)";}
```

### Lines :

			style	attr
border-width	Number [+unit]	Specifies the thickness of the border, or outline, of the block element.	•	
border-style	none hidden dotted dashed solid double groove ridge inset outset	Specifies the style of the line. By default, this is none (no border).	•	

```
{border-style: dashed; border-width: 2px;}
```

The diagram illustrates the structure of an SVG document. It starts with a root node labeled **svg**. This node contains a **defs** group, which in turn contains five elements: **clipPath**, **pattern**, **linearGradient**, and **radialGradient**. The **svg** node also contains a **g** (group) element. This **g** element contains a collection of basic shapes: **circle**, **line**, **ellipse**, **path**, **polygon**, **text**, **polyline**, and **rect**. The diagram uses brackets to show the containment relationships between these elements.

Positioning :		svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
transform	"translate(x,y)"		●	●	●	●	●	●	●	●	●		●
x,y	number			●	●								●
x1,x2,y1,y2	number					●							●
d	(special)						●						●
points	"x,y x,y x,y"							●	●				●
cx,cy	number									●	●		●

Sizing:		svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
transform	"scale(k) "		●	●	●	●	●	●	●	●	●		●
width, height	number	●		●									●
r	number									●			●
rx,ry	number										●		●

[illegible]

Lines :		svg	g	rect	text	line	path	polygon	polyline	circle	ellipse	style	attr
stroke-width	number			●	●	●	●	●	●	●	●	●	
stroke-dasharray	numbers (separated by commas, blanks)*			●	●	●	●	●	●	●	●	●	
stroke-linecap	butt round square					●	●	●	●			●	
stroke-linejoin	miter round bevel						●	●				●	
stroke-miterlimit	number					●	●	●	●			●	

Text :			style	attr
text-anchor	start middle end	Determine if the text is aligned to the left, centered or aligned to the right.		●
writing-mode	tb	If specified, text is written vertically	●	
glyph-orientation-vertical	0 1	If 0, and writing-mode is set to tb, letters are horizontal (stacked), else if set to 1 when text mode is set to tb, the text will be rotated (written from top to bottom).	●	
textLength	Number	If specified, the text will be adjusted to fit the specified length.		●
lengthAdjust	spacing   spacingAndGlyphs	Which strategy is used to adjust the length?		●
textPath	xlink:href (#id)	Writes the text along a path (which should be in the defs and given an id attribute)		

[illegible]

# d3 cheat sheet

Select > data > enter > append :

select	CSS selector (string)	Selects the <i>first</i> element that matches the selector.
selectAll	CSS selector (string)	Selects <i>all</i> the elements that match the selector.
Selections can be chained : each new statement will look <i>inside</i> the current selection.		
data	array of items	Provides the data that will be matched against the selection
enter	nothing - ()	Selects elements to be added: the items in the data which are not yet matched by elements.
exit	Nothing - ()	Selects elements to be removed: the elements which are no longer matched by items in the data
append	name of element	creates specified element inside the selection, either once (if data/enter is not used) or as many times as there are elements in enter().
remove	Nothing - ()	removes selected elements.

The following constructs are valid.

d3.select("body").append("svg");	Creates an svg element in the HTML document (assuming there is a body element)
var myRects=d3.select("body").select("svg").selectAll("rect");	Looks for all rect elements within the first svg element found within the body element.
myRects=myRects.data([1,2,3]).enter();	If there are fewer than 3 rect elements in myRects, this prepares the missing elements to be added.
myRects.append("rect");	Create these missing elements. Extra methods can be added to initiate them (attr, style...) will only apply to the elements just created
d3.selectAll("rect");	This selects all rect elements. Methods added there will apply to all of the rect elements, not only to those just created.
d3.selectAll("rect").data([4,5]).exit().remove();	This passes new data to the rect elements. Then, if there are more than 2 of them, the rest – exit() – are deleted – remove().
d3.selectAll("rect").remove();	This deletes all rect elements.

The following are not:

d3.append("svg");	All elements must be added within a container element. <i>First</i> select, <i>then</i> append.
d3.selectAll("rect").enter().append("rect"); d3.data([1,2,3]).enter().append("rect");	enter() requires <i>first</i> a selection, <i>then</i> data.
d3.select("rect").data([1,2,3]).enter().append("rect");	To use data and enter, you need to use selectAll first. (else elements are created, but not where you'd expect.)

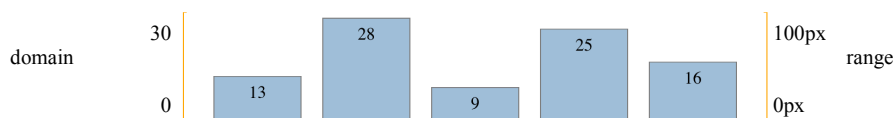
Path – what goes in the d attribute:

M	x,y	Begins the string. M moves the cursor to the designated position to draw a shape. If more than one pair of coordinates are provided, it's as if the other pairs are preceded by an L.
m	x,y	Same as above, the only difference is that if more than one pair of coordinates are provided, extra pairs are processed as if preceded by an l.
L	x,y	Draws a line from the last position to the specified position.
l	x,y	Draws a line from the last position to a relative position: x pixels to the right and y to the bottom.
H	x	Draws a horizontal line to the specified, absolute x position (y stays the same)
h	x	Draws a horizontal line x pixels to the right (if x is negative, the line is drawn to the left).
V	y	Draws a vertical line to the specified, absolute y position (x stays the same)
v	y	Draws a vertical line x pixels to the bottom (if y is negative, the line is drawn to the top).
A	rx,ry,alpha,large,sweep,x,y	Draws an elliptical arc: rx and ry are the radius of the ellipse; alpha is the x-axis rotation of the ellipse; large is 0 if the arc should be the shorter arc (less than 180°), 1 if it should be the longer arc; sweep is 0 is the arc is to be drawn clockwise, 1 if counter-clockwise; x,y are the coordinate of the end point of the arc.
Q	cX,cY,eX,eY	Draws a Bézier quadratic curve. cX,cY are the coordinates of the control point, eX,eY that of the endpoint. More pairs can be provided (extra control points and endpoints).
q	cX,cY,eX,eY	Draws a Bézier quadratic curve, with the coordinates of the points relative to the current point.
T	eX,eY	Draws a Bézier quadratic curve, using the last provided control point (or failing that the current point).
t	eX,eY	Same as above, with the coordinates of the end point relative to the current point.
C	cX1,cY1,cX2,cY2,eX,eY	Draws a Bézier cubic curve, with cX1,cY1, cX2, cY2 being the coordinates of the control points.
c	cX1,cY1,cX2,cY2,eX,eY	Same as above, with the coordinates of the points relative to the current point.
S	cX2,cY2,eX,eY	Draws a Bézier cubic curve, using the previously provided control point (cX1,cY1) or failing that the current point, cX2,cY2 as the next control point, and eX, eY as the end point
s	cX2,cY2,eX,eY	Same as above, with the coordinates of the points relative to the current point.
Z		Optional at the end the string for a closed shape (ie line to the first point)

## Quantitative scales

d3.scale	The beginning of every scale.
domain	The interval of values to transform. It is an array of 2 or more ordered values.
range	The interval of values in which to be transformed. It is an array with the same number of values than the domain. If there are more than 2 values, values of the first segment of the domain are turned into values of the first segment of the range, and subsequently values in the $k^{\text{th}}$ segment of the domain are turned into values of the $k^{\text{th}}$ segment of the range.
	<i>You must select one type of scale. These 5 scales have continuous outputs.</i>
Linear	The linear scale transforms one value in the domain interval into a value in the range interval (without transformation)
pow	The pow scale transforms one value in the domain interval, raised to a certain power, into one value in the range interval.
Exponent	For the pow scale, the exponent method allows to specify an exponent (1 by default, equivalent to linear).
sqrt	Transforms the square root of one value in the domain interval into one value in the range interval. Equivalent to <code>d3.scale.pow().exponent(.5)</code> .
log	Transforms the log of one value in the domain interval into one value in the range interval.
identity	The identity scale doesn't transform a value, but is useful when you need a scale object, specify a range etc.
	<i>The following 3 scales have discrete outputs.</i>
quantize	If the value is between the $k^{\text{th}}$ and the $k+1^{\text{th}}$ value of the domain, returns the $k^{\text{th}}$ value of the range.
threshold	If the value is between the $k^{\text{th}}$ and the $k+1^{\text{th}}$ value of the domain, returns the $k+1^{\text{th}}$ value of the range.
quantile	This is the one scale that doesn't require the domain and range to have the same cardinality. It divides the domain into $i$ intervals, where $i$ is the cardinality of the range array. Then, if a value is in the $k^{\text{th}}$ interval, it returns the $k^{\text{th}}$ value of the range.
	These other methods apply to all quantitative scales
invert	The opposite of the scale. If $s$ is a scale and $s(x)=y$ , then $s.invert()(y) = x$ .
nice	Extends the domain of the scale so that its bounds are round values.
rangeRound	(use instead of range). Makes it so that the output of the range are rounded to integer values.
interpolate	Takes a function ("factory"). Allows to override how d3 maps values from the domain to the range.
clamp	If set to <code>[true]</code> , if a value is outside the domain, it will be transformed into either the lower or the upper bound of the range.

```
var s1=d3.scale.linear().domain([0,10]).range([50,100]); s1(5) // 75
var s2=d3.scale.pow().domain([0,10]).rangeRound([50,100]).exponent(2); s2(9) // 91
var s3=d3.scale.threshold().domain([0,2,5,10]).range([50,100,150,200]); s3(3) // 150
var s4=d3.scale.quantile().domain([0,10]).range([0,1,2,3]); s4(4) // 1
var s5=d3.scale.linear().domain([0,10]).range([50,100]).clamp(true); s5(15) // 100
```



## Transitions

<code>selection.transition()</code>	Starts a transition on this selection.
<code>.duration</code>	Number of milliseconds
<code>.delay</code>	Number of milliseconds
<code>.attr</code>	String (attr. name), value
<code>.style</code>	String (style name), value
<code>.each</code>	"end", function

```
d3.select("rect").transition().delay(100).duration(1000).attr("x",100) // moves that rectangle to 100 pixels from the left of its container
d3.select("rect").transition().style("fill","red"); // turns it to red
d3.select("rect").transition().style("opacity",0).each("end",function() {d3.select(this).remove();}) // makes it transparent, and when it's completely transparent, delete it.
```

## Interactions

on	Event listener, function	Launches <i>function</i> when the specified event occurs. Event listeners include: click, dblclick, mouseover, mouseout, Form controls can use event listener "change" which is triggered when the control changes values.
d3.select(this)		Within a function triggered by an event listener, this refers to the element where the event occurred. For instance, the rect element on which one clicked, etc.
d3.select(window).on("keydown",function)		Launches a function when a key is pressed. Within this function, <code>d3.event.keycode</code> will hold the value of that key.