D3 Tips and Tricks

```
<!DOCTYPE html>
<meta charset="utf-8">

<style>
   The CSS is in here
</style>

<body>
   <script src="https://d3js.org/d3.v5.min.js"></script>
   <script>
      The D3 JavaScript code is here
   </script>
</body>
```

Το HTML μέρος του κώδικα

Τα δύο τμήματα που πρέπει να συμπληρωθούν για να κατασκευαστεί μία οπτικοποίηση με D3 είναι το CSS κομμάτι και ο κώδικας D3 με JavaScript.

Θα κατασκευάσουμε μία γραφική παράσταση όπως αυτή στο παρακάτω σχήμα, με πηγή δεδομένων ένα απλό CSV αρχείο:

Το CSS τμήμα αφορά αποκλειστικά και μόνο την εμφάνιση της γραμμής.

```
.line {
  fill: none;
  stroke: steelblue;
  stroke-width: 2px;
}
```

Στα CSS έχουμε “κανόνες” (rules). Κάθε κανόνας έχει έναν “επιλογέα” (selector) και έναν ή περισσότερους “ορισμούς” (declarations). Κάθε declaration έχει μία ιδιότητα και μία τιμή.

line είναι ο selector. Η τελεία μπροστά του μπαίνει επειδή ο selector αυτός αφορά μία κλάση.

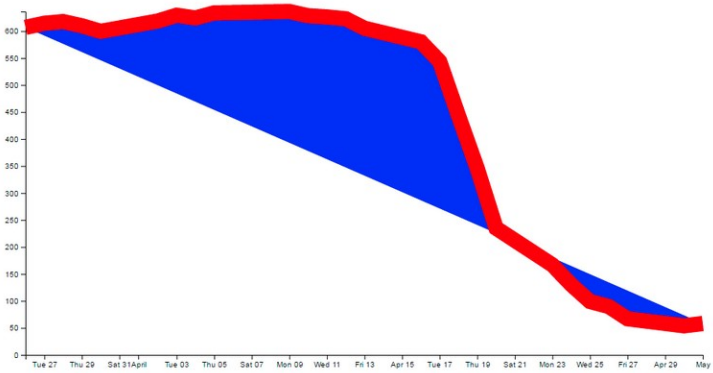
Εάν αλλάξουμε το stoke-width σε 20px:

```
stroke-width: 20px;
```

Η γραμμή γίνεται έτσι...

Εάν αλλάξουμε το stroke σε red...

Εάν αλλάξουμε το fill από none σε blue...



Βλέπουμε ότι το fill property αλλάζει το χρώμα της περιοχής που περικλείεται από την γραμμή.

```
// set the dimensions and margins of the graph
var margin = {top: 20, right: 20, bottom: 30, left: 50},
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top - margin.bottom;

// parse the date / time
var parseTime = d3.timeParse("%d-%b-%y");

// set the ranges
var x = d3.scaleTime().range([0, width]);
var y = d3.scaleLinear().range([height, 0]);

// define the line
var valueline = d3.line()
    .x(function(d) { return x(d.date); })
    .y(function(d) { return y(d.close); });

// append the svg obgect to the body of the page
// appends a 'group' element to 'svg'
// moves the 'group' element to the top left margin
var svg = d3.select("body").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
          "translate(" + margin.left + "," + margin.top + ")");

// Get the data
d3.csv("data.csv").then(function(data) {

  // format the data
  data.forEach(function(d) {
      d.date = parseTime(d.date);
      d.close = +d.close;
  });

  // Scale the range of the data
  x.domain(d3.extent(data, function(d) { return d.date; }));
  y.domain([0, d3.max(data, function(d) { return d.close; })]);

  // Add the valueline path.
  svg.append("path")
      .data([data])
      .attr("class", "line")
      .attr("d", valueline);

  // Add the X Axis
  svg.append("g")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x));

  // Add the Y Axis
  svg.append("g")
      .call(d3.axisLeft(y));

});
```

Ο κώδικας αυτός περιέχει αρκετές λεπτομέρειες, τις οποίες και θα δούμε κομμάτι-κομμάτι.

Αρχικά ορίζουμε τα margins και την περιοχή του γραφήματος.
Ο κώδικας για αυτή την διαδικασία καθώς και η καλύτερη προσέγγιση περιγράφονται στην ιστοσελίδα του Mike Bostock: http://bl.ocks.org/3019563

```
// ορισμός διαστάσεων και περιθωρίων του γραφήματος
var margin = {top: 20, right: 20, bottom: 30, left: 50},
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top – margin.bottom;
```

Η πρώτη γραμμή ορίζει το margin object, το οποίο περιέχει τέσσερα περιθώρια, ορισμένα από το πάνω μέρος και σύμφωνα με τη φορά των δεικτών του ρολογιού γύρω από την περιοχή στην οποία θα τοποθετηθεί το γράφημα.

```
var margin = {top: 20, right: 20, bottom: 30, left: 50},
```

Με αυτόν τον τρόπο το πάνω περιθώριο είναι 20 pixels, το δεξί 20, το κάτω 30 και το αριστερό 50.

> Στην JavaScript μπορούμε τις μεταβλητές να τις καταχωρήσουμε σε arrays και αντικείμενα. Μία απλή τεχνική για να ορίσουμε array είναι με square brackets ή λέγοντας new Array() και για να ορίσουμε αντικείμενα, η χρήση από curly brackets. Ο ορισμός του margin περιλαμβάνει τέσσερις ιδιότητες. Κάθε ιδιότητα πρέπει να έχει ένα ξεχωριστό όνομα και αναφερόμαστε σε αυτή μέσω dot syntax. Π.χ. margin.right. Τα στοιχεία των arrays είναι ανώνυμα και αναφερόμαστε σε αυτά μέσω της θέσης τους, όπως στη C πχ. myArray[0], myArray[1] κλπ.
>
> Όταν δηλώνουμε ένα array με την χρήση του new πρέπει να προσέχουμε επειδή:
>
> var nodes = new Array(100, 200, 300); // κατασκευάζει ένα array με τρία στοιχεία ενώ
>
> var nodes = new Array(100); // κατασκευάζει ένα array με 100 undefined στοιχεία

Ακολουθεί ο ορισμός του εύρους και του ύψους, σαν εσωτερικές διαστάσεις της περιοχής. Με αυτόν τον τρόπο το width και το height αφορούν καθαρές διαστάσεις του γραφήματος μας.

```
    width = 960 - margin.left - margin.right,
    height = 500 - margin.top – margin.bottom;
```

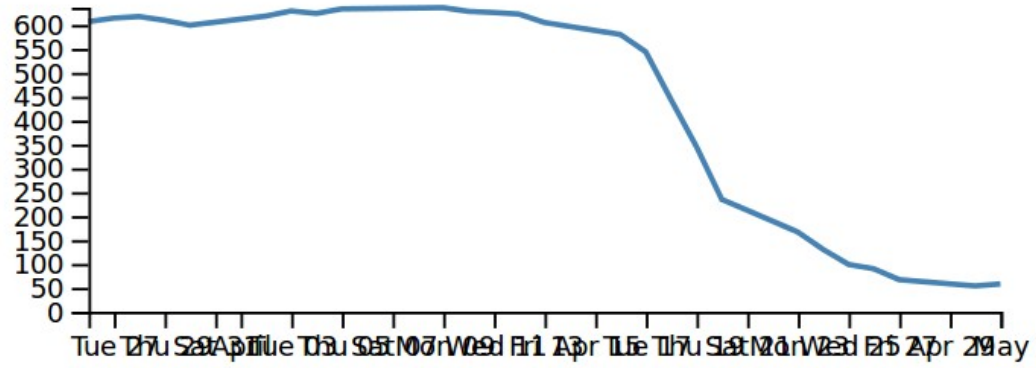
Τελικά ορίζουμε το svg σαν στοιχείο G, που μεταφράζει την αριστερή πάνω γωνία της περιοχής του γραφήματος.

```
var svg = d3.select("body").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
          "translate(" + margin.left + "," + margin.top + ")");
```

Αν τροποποιήσουμε τα μεγέθη και δώσουμε τις επόμενες τιμές:

```
var margin = {top: 80, right: 20, bottom: 80, left: 50},
   width = 400 - margin.left – margin.right,
   height = 270 - margin.top - margin.bottom;
```

Και το διάγραμμα γίνεται έτσι:



Έγινε δηλαδή στενότερο (400 pixels) αλλά κράτησε τα left και right margins, ενώ μεγάλωσαν τα top και bottom margins. Προσέξτε ότι οι άξονες προσαρμόζονται αυτόματα στις αλλαγές. Προφανώς υπάρχει θέμα με τον άξονα των x.

Φορτώνοντας δεδομένα

Αυτός ο κώδικας διαβάζει τα δεδομένα:

```
// Get the data
d3.csv("data.csv").then(function(data) {

  // format the data
  data.forEach(function(d) {
      d.date = parseTime(d.date);
      d.close = +d.close;
  });
```

Αν έχουμε λίγα δεδομένα θα μπορούσαμε να τα φορτώσουμε και από τον JavaScript κώδικα χρησιμοποιώντας μία JSON δομή:

```
var data = [
    {date:"1-May-12",close:"58.13"},
    {date:"30-Apr-12",close:"53.98"},
    {date:"27-Apr-12",close:"67.00"},
    {date:"26-Apr-12",close:"89.70"},
    {date:"25-Apr-12",close:"99.00"}
];

// Get the data
```

```
d3.csv("data.csv", function(error, data) {
  if (error) throw error;

  // format the data
  data.forEach(function(d) {
      d.date = parseTime(d.date);
      d.close = +d.close;
  });
```

Η πρώτη γραμμή κάνει ένα d3.csv request και η συνάρτηση επεξεργάζεται τα δεδομένα τα οποία βρίσκονται στο URL data.csv.

Το data.csv έχει αυτό το format:

```
date,close
1-May-12,58.13
30-Apr-12,53.98
27-Apr-12,67.00
26-Apr-12,89.70
25-Apr-12,99.00
```

Ξεκινάει με επικεφαλίδες και ακολουθούν τα δεδομένα, χωρισμένα με κόμμα.

Το δεύτερο τμήμα της γραμμής καλεί την συνάρτηση function για επεξεργασία των δεδομένων τα οποία λέγονται data.

```
function(error, data) {
```

Η συνάρτηση θα πιάσει (catch) οποιοδήποτε error συμβεί στην διαδικασία. Εάν υπήρχε catch block θα μπορούσαμε να χειριστούμε το error. Εαν δεν υπάρχει τερματίζεται η λειτουργία του προγράμματος.

```
  data.forEach(function(d) {
      d.date = parseTime(d.date);
      d.close = +d.close;
  });
```

Ο κώδικας αυτός εξασφαλίζει ότι όλες οι τιμές που θα βγουν από το CSV αρχείο θα γίνουν formatted με τον κατάλληλο τρόπο. Η πρώτη γραμμή καθορίζει ότι κάθε ομάδα από δεδομένα που βγαίνουν από το αρχείο πρέπει να επεξεργαστούν. Κάθε γραμμή λέγεται d.

Για τα dates καλείται η parseTime και για το close το + εξασφαλίζει ότι θα αντιμετωπιστεί σαν αριθμός.

Το άγκιστρο που ανοίγει στην γραμμή:

```
d3.csv("data.csv", function(error, data) {
```

κλείνει πολύ πιο κάτω.

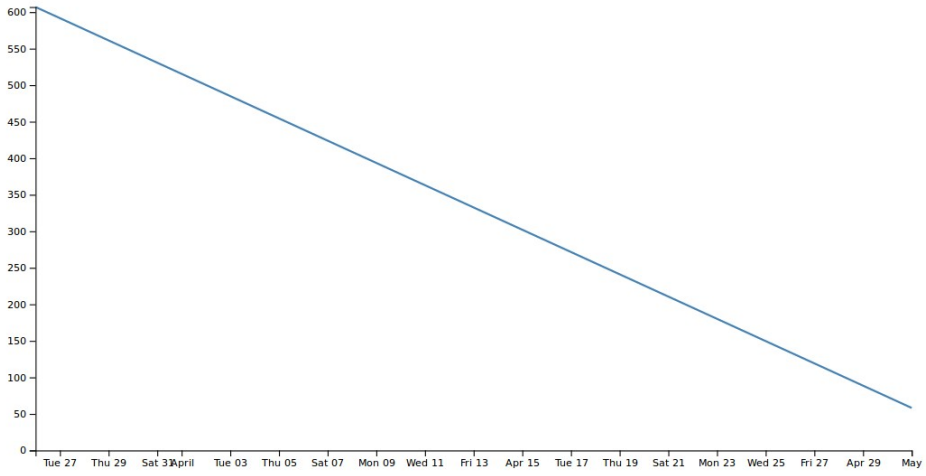
Το αρχείο data.csv

```
date,close
1-May-12,58.13
30-Apr-12,53.98
27-Apr-12,67.00
26-Apr-12,89.70
25-Apr-12,99.00
24-Apr-12,130.28
23-Apr-12,166.70
20-Apr-12,234.98
19-Apr-12,345.44
18-Apr-12,443.34
17-Apr-12,543.70
16-Apr-12,580.13
13-Apr-12,605.23
12-Apr-12,622.77
11-Apr-12,626.20
10-Apr-12,628.44
9-Apr-12,636.23
5-Apr-12,633.68
4-Apr-12,624.31
3-Apr-12,629.32
2-Apr-12,618.63
30-Mar-12,599.55
29-Mar-12,609.86
28-Mar-12,617.62
27-Mar-12,614.48
26-Mar-12,606.98
```

Εάν διαγράψουμε όλες τις ενδιάμεσες τιμές από το αρχείο

```
date,close
1-May-12,58.13
26-Mar-12,606.98
```

Το γράφημα γίνεται έτσι:



Αν όμως αλλάξουμε την πιο πρόσφατη μέρα από 1 Μαΐου σε 29 Μαρτίου τότε βλέπουμε ότι το D3 αυτόματα αναγνωρίζει ότι πρέπει να αλλάξει τον άξονα των x, να δείχνει ώρα.

Αν αλλάξουμε τα έτη, πάλι αλλάζει ο άξονας των x αυτόματα δείχνοντας έτη.

```
date,close
29-Mar-21,58.13
26-Mar-12,606.98
```

Πίνακας κωδικών που μπορούμε να χρησιμοποιήσουμε για dates

- %a - abbreviated weekday name.
- %A - full weekday name.
- %b - abbreviated month name.
- %B - full month name.
- %c - date and time, as “%a %b %e %H:%M:%S %Y”.
- %d - zero-padded day of the month as a decimal number [01,31].
- %e - space-padded day of the month as a decimal number [1,31].
- %H - hour (24-hour clock) as a decimal number [00,23].
- %I - hour (12-hour clock) as a decimal number [01,12].
- %j - day of the year as a decimal number [001,366].
- %m - month as a decimal number [01,12].
- %M - minute as a decimal number [00,59].
- %p - either AM or PM.
- %S - second as a decimal number [00,61].
- %U - week number of the year (Sunday as the first day of the week) as a decimal number [00,53].
- %w - weekday as a decimal number [0(Sunday),6].
- %W - week number of the year (Monday as the first day of the week) as a decimal number [00,53].
- %x - date, as “%m/%d/%y”.
- %X - time, as “%H:%M:%S”.
- %y - year without century as a decimal number [00,99].
- %Y - year with century as a decimal number.
- %Z - time zone offset, such as “-0700”.
- There is also a a literal “%” character that can be presented by using double % signs.

Για να επεξεργαστούμε generic MySQL dates, ‘YYYY-MM-DD HH:MM:SS’ TIMESTAMP format το format του D3 πρέπει να είναι έτσι: `parseTime = d3.timeParse("%Y-%m-%d %H:%M:%S");`

Το format που χρησιμοποιήσαμε ήταν:

d για μέρα του μήνα [01-31],
b για abbreviated όνομα μήνα και
y για έτος διψήφιο.
- αντικαθιστά τις υπάρχουσες παύλες.

Scales και Ranges

Το σημείο αυτό του κώδικα:

```
// set the ranges
var x = d3.scaleTime().range([0, width]);
var y = d3.scaleLinear().range([height, 0]);
```

Ο κώδικας φροντίζει τα όρια των τιμών που θέλουμε να απεικονίσουμε σε ένα γράφημα να χωράνε σε αυτό. Επειδή στο γράφημα μας περιλαμβάνονται δύο σειρές τιμών: αριθμοί και ημερομηνίες πρέπει να τις αντιμετωπίσουμε ξεχωριστά.

```
  // Scale the range of the data
  x.domain(d3.extent(data, function(d) { return d.date; }));
  y.domain([0, d3.max(data, function(d) { return d.close; })]);
```

Εδώ φροντίζουμε να πάρουμε τα όρια των τιμών που θα αναπαραστήσουμε έτσι ώστε να τηρηθούν οι αναλογίες της απεικόνισης.

Αν έχουμε data από 53.98 έως 636.23, όπως συμβαίνει στο CSV αρχείο μας, αλλά έχουμε ένα γράφημα με μόνο 450 pixels ύψος πρέπει να αλλάξουμε κλίμακα.

Η scaleTime() προσαρμόζει τον άξονα x.