

Σημειώσεις D3

♦ Είδαμε ότι το D3 είναι μία JavaScript βιβλιοθήκη που μας επιτρέπει να επεξεργαζόμαστε δυναμικά στοιχεία του DOM. Το D3 δίνει την δυνατότητα στον προγραμματιστή να επιλέγει πολλά στοιχεία DOM με μία select. Με απλή JavaScript, χωρίς D3, μπορούμε να προσθέσουμε ένα div:

```
<body>
  <script type="text/javascript">
    var div = document.createElement("div");
    div.innerHTML = "Hello, world!";
    document.body.appendChild(div);
  </script>
</body>
```

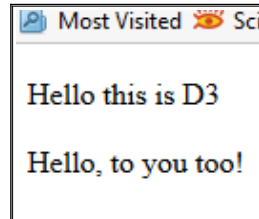
♦ Με την D3 μπορούμε να επιλέξουμε έναν ή πολλά υπαρκτά ή όχι στοιχεία του DOM. Στο παρακάτω παράδειγμα, η select καλείται να επιλέξει μία παράγραφο και να της αλλάξει το text. Επειδή παράγραφος δεν υπάρχει στο DOM, το αποτέλεσμα είναι να μην κάνει τίποτα.

```
<!doctype html>
<html>
<head>
  <title>D3 tutorial</title>
  <script src="d3/d3.min.js" charset="utf-8"></script>
</head>
<body>
  <script type="text/javascript">
    d3.select("p").text("Hello this is D3");
  </script>
</body>
```

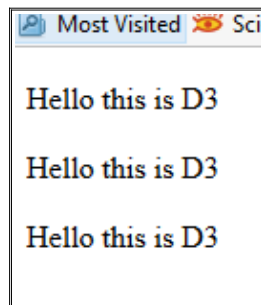
♦ Προφανώς εάν προσθέταμε μία ή περισσότερες παραγράφους, πριν το script, θα άλλαζε το text στην πρώτη.

```
<body>
  <p>Hello, world!
  <p>Hello, to you too!
  <p>
  <script type="text/javascript">
    d3.select("p").text("Hello this is D3");
  </script>
</body>
```

Το αποτέλεσμα είναι αυτό. Το κείμενο από την πρώτη παράγραφο αντικαθίσταται αμέσως από το text που ορίζεται στο D3 script. Η δεύτερη και η τρίτη παράγραφος παραμένουν όπως ορίστηκαν στην HTML. Η τρίτη είναι κενή.



Αντικαθιστώντας την select με την **selectAll**, αντικαθίσταται το text σε όλα τα στοιχεία του DOM:



♦ Με την **append** μπορούμε να προσθέσουμε νέα στοιχεία στην σελίδα μας:

```
<body>
  <script type="text/javascript">
    d3.select("body").append("p").text("Hello this is D3");
  </script>
</body>
```

Χρησιμοποιώντας αλυσίδες μεθόδων μπορούμε να γλυτώσουμε πολύ γράψιμο. Με τα method chains ο παρακάτω κώδικας:

```
var body = d3.select("body");
body.style("color", "black");
body.style("background-color", "white");
```

μπορεί να γραφεί καλύτερα έτσι:

```
d3.select("body")
  .style("color", "black")
  .style("background-color", "white");
```

και:

```
d3.select("body")
  .append("p")
  .style("color", "red")
  .text("Hello this is D3");
```

- ◆ Μπορούμε να προσθέσουμε γραφικά στοιχεία **SVG** στην σελίδα μας όπως στο επόμενο παράδειγμα:

```
<!doctype html>
<html>

<head>
  <title>D3 tutorial 3a</title>
  <script type="text/javascript" src="d3/d3.min.js"></script>
</head>
<body>
  <script type="text/javascript">

    var canvas = d3.select("body")
      .append("svg")
      .attr("width", "500")
      .attr("height", "500");

    var circle = canvas.append("circle")
      .attr("cx", "250")
      .attr("cy", "250")
      .attr("r", 50)
      .attr("fill", "red");

    var rect = canvas.append("rect")
      .attr("width", "100")
      .attr("height", "50");

    var line = canvas.append("line")
      .attr("x1", "0")
      .attr("y1", "100")
      .attr("x2", "400")
      .attr("y2", "400")
      .attr("stroke", "green")
      .attr("stroke-width", "10");

  </script>
</body>
```

Βλέπουμε ότι στο SVG αντί για style δηλώνουμε attributes.

◆ Data Join Concept¹

Αν θέλουμε να κατασκευάσουμε μερικά circle elements για να οπτικοποιήσουμε δεδομένα, θα διαπιστώσουμε ότι δεν υπάρχει στο D3 βασική εντολή για να δημιουργηθούν πολλαπλά DOM elements. Προφανώς, υφίσταται η μέθοδος append, αλλά αυτή μας δίνει την δυνατότητα να κατασκευάσουμε ένα στοιχείο.

```
svg.append("circle")
  .attr("cx", d.x)
  .attr("cy", d.y)
```

1 <https://bost.ocks.org/mike/join/>

```
.attr("r", 2.5);
```

Εάν θέλουμε να αποκτήσουμε την δυνατότητα να κατασκευάσουμε πολλούς κύκλους: έναν για κάθε δεδομένο, αντί να βάλουμε τον κώδικα μας μέσα σε ένα for ή while loop ακολουθούμε την παρακάτω προσέγγιση:

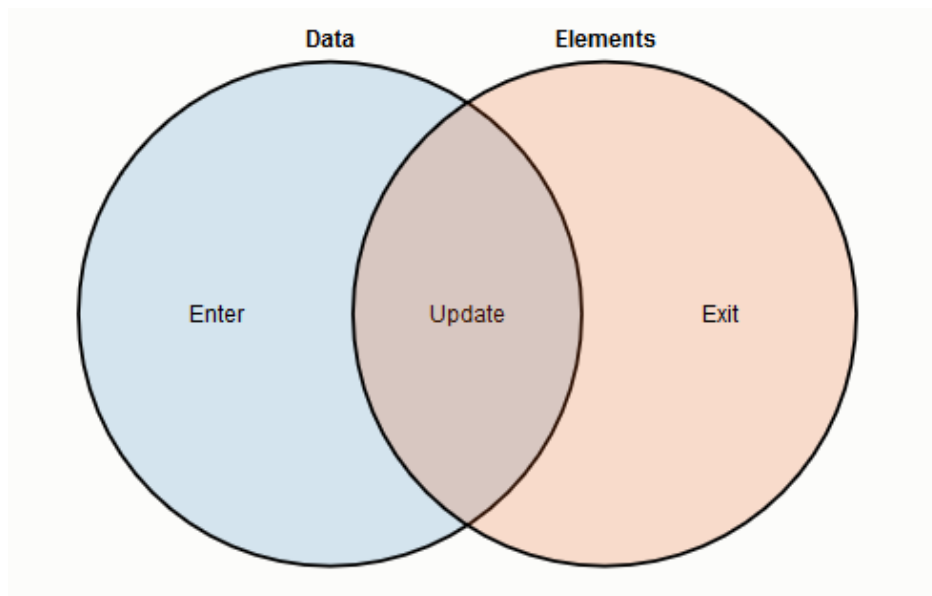
```
svg.selectAll("circle")  
  .data(data)  
  .enter().append("circle")  
  .attr("cx", function(d) { return d.x; })  
  .attr("cy", function(d) { return d.y; })  
  .attr("r", 2.5);
```

Εδώ τα δεδομένα προέρχονται από ένα JSON array με x και y ιδιότητες:

```
[{"x": 1.0, "y": 1.1}, {"x": 2.0, "y": 2.5}, ...].
```

Ο κώδικας κατασκευάζει έναν κύκλο για κάθε στοιχείο του πίνακα, χρησιμοποιώντας τις ιδιότητες x και y για την τοποθέτησή τους. Το `selectAll("circle")` επιλέγει στοιχεία που **δεν υπάρχουν ακόμα**.

Η προσέγγιση του D3 αντίθετα με πολλές τεχνικές προγραμματισμού που ο προγραμματιστής καθορίζει με εντολές πως πρέπει να γίνει μία διαδικασία, ζητάει από τον προγραμματιστή να πει τι θέλει. Εδώ θέλει ο κάθε κύκλος να αντιστοιχεί σε ένα ζεύγος δεδομένα. Αντί να καθοδηγήσουμε το D3 να κατασκευάζει κύκλους απλά του λέμε ότι η επιλογή "circle" αντιστοιχεί σε δεδομένα. Αυτή η τεχνική ονομάζεται **Data Join**.



Τα σημεία που ενώνονται με υπάρχοντα στοιχεία δημιουργούν την **update** (inner) selection. Όσα δεν έχουν ενωθεί παράγουν την **enter** selection, η οποία αντιπροσωπεύει στοιχεία που λείπουν. Και η επιλογή **exit** αντιπροσωπεύει τα στοιχεία που πρέπει να σβηστούν.

1. Η `svg.selectAll("circle")` επιστρέφει μια κενή επιλογή, επειδή ο SVG container ήταν κενός.
2. Η πιο πάνω επιλογή ενώνεται με ένα array από data, με αποτέλεσμα να δημιουργηθούν τρεις νέες καταστάσεις: **enter**, **update** και **exit**. Επειδή η επιλογή ήταν αρχικά άδεια, οι επιλογές **update** και **exit** είναι επίσης άδειες. Η επιλογή **enter** περιέχει ένα placeholder για κάθε δεδομένο.
3. Η επιλογή **update** επιστρέφεται από το `selection.data`, ενώ οι επιλογές **enter** και **exit** εξαρτώνται από την **update selection**, ενώ η `selection.enter` επιστρέφει την επιλογή **enter**.
4. Τα στοιχεία που λείπουν προστίθενται στον SVG container καλώντας την `selection.append` μετά την **enter selection**. Έτσι προστίθεται ένας νέος κύκλος για κάθε data point στον SVG container.

Το σκεπτικό των Joins είναι ο ορισμός μίας σχέσης μεταξύ της επιλογής και των δεδομένων και η υλοποίηση αυτής της σχέσης μέσω των τριών καταστάσεων: **enter**, **update** και **exit**.

Θα μπορούσε εύκολα να υπάρχει ένα primitive (θεμελιακό στοιχείο) που να δημιουργεί πολλαπλά elements, αλλά το πλεονέκτημα της προσέγγισης μέσω Join είναι αποτελεί πιο γενικότερη λύση. Ο κώδικας πιο πάνω καλύπτει τις ανάγκες των στατικών οπτικοποιήσεων. Είναι όμως πολύ εύκολο να επεκταθεί με ελάχιστες προσθήκες έτσι ώστε να καλύπτει και τις ανάγκες των δυναμικών οπτικοποιήσεων για **update** και **exit**. Έτσι επιτρέπει διαδραστικές αναζητήσεις και οπτικοποιήσεις σε πραγματικό χρόνο καθώς και ομαλές εναλλαγές δεδομένων.

Εδώ είναι ένα παράδειγμα που διαχειρίζεται και τις τρεις καταστάσεις:

```
var circle = svg.selectAll("circle")
    .data(data);

circle.exit().remove();

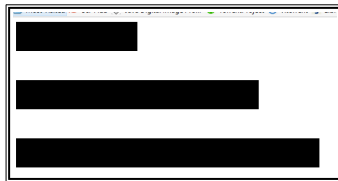
circle.enter().append("circle")
    .attr("r", 2.5)
    .merge(circle)
    .attr("cx", function(d) { return d.x; })
    .attr("cy", function(d) { return d.y; });
```

Όταν τρέξει αυτός ο κώδικας, επανυπολογίζει το Join των δεδομένων και υποστηρίζει την αντιστοιχία μεταξύ στοιχείων και δεδομένων. Εάν το νέο data-set είναι μικρότερο από το προηγούμενο, όλα τα επιπλέον στοιχεία καταλήγουν στην επιλογή **exit** και διαγράφονται. Εάν το νέο data-set είναι μεγαλύτερο τα επιπλέον στοιχεία καταλήγουν στην **enter** επιλογή και νέοι κόμβοι προστίθενται. Εάν το

νέο data-set είναι ίδιου μεγέθους με το προηγούμενο, τότε όλα τα στοιχεία ενημερώνονται σε νέες θέσεις και δεν διαγράφεται ούτε προστίθεται τίποτα.

Τα Joins επιτρέπουν τον κώδικα να είναι περισσότερο περιγραφικός: η διαχείριση των τριών καταστάσεων γίνεται χωρίς επιλογές (if) ή επαναλήψεις (for, while). Ο προγραμματιστής περιγράφει πως τα στοιχεία πρέπει να αντιστοιχούν σε δεδομένα. Εάν μία επιλογή enter, update ή exit τύχει να είναι κενή, τότε ο αντίστοιχος κώδικας είναι no-op.

◆ Ένα πολύ απλό παράδειγμα που δείχνει πως μπορεί να κατασκευαστεί ένα barchart όπως αυτό:



Ο κώδικας:

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>D3 tutorial</title>
5     <script src="d3/d3.min.js" charset="utf-8"></script>
6   </head>
7
8   <body>
9     <script type="text/javascript">
10
11       var dataArray = [20, 40, 50];
12
13       var canvas = d3.select("body")
14         .append("svg")
15         .attr("width", 500)
16         .attr("height", 500);
17
18       var bars = canvas.selectAll("rect")
19         .data(dataArray)
20         .enter()
21         .append("rect")
22         .attr("width", function(d) { return d * 10;})
23         .attr("height", 50)
24         .attr("y", function(d, i) { return i * 100 });
25     </script>
26   </body>
27 </html>
```

Άσκηση: Να τροποποιήσετε τον κώδικα, έτσι ώστε οι μπάρες του ιστογράμματος να γίνουν πέντε και η κατεύθυνση τους να γίνει κάθετη.