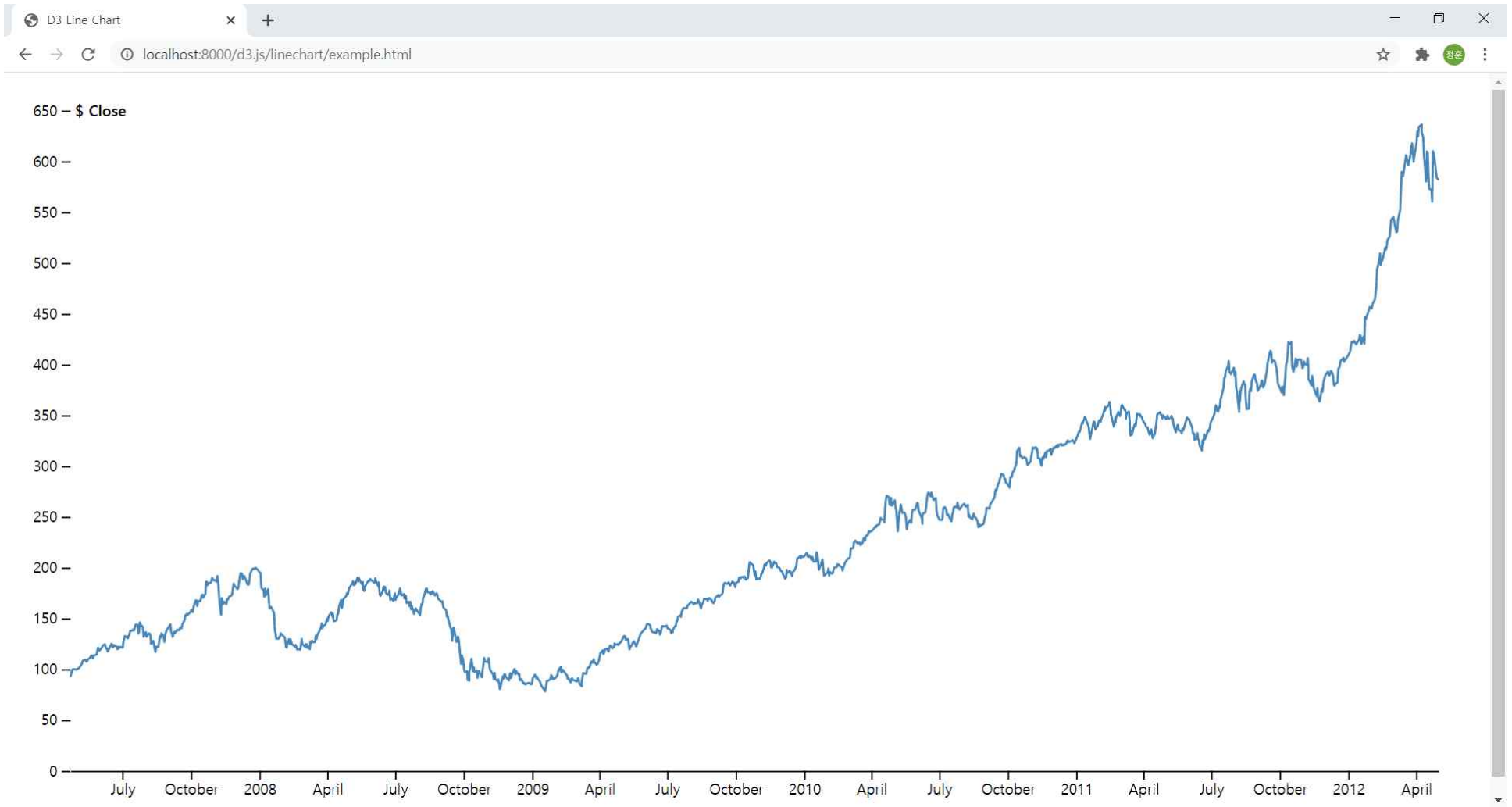
2021-Python/d3.js Study(week 3)

이정훈

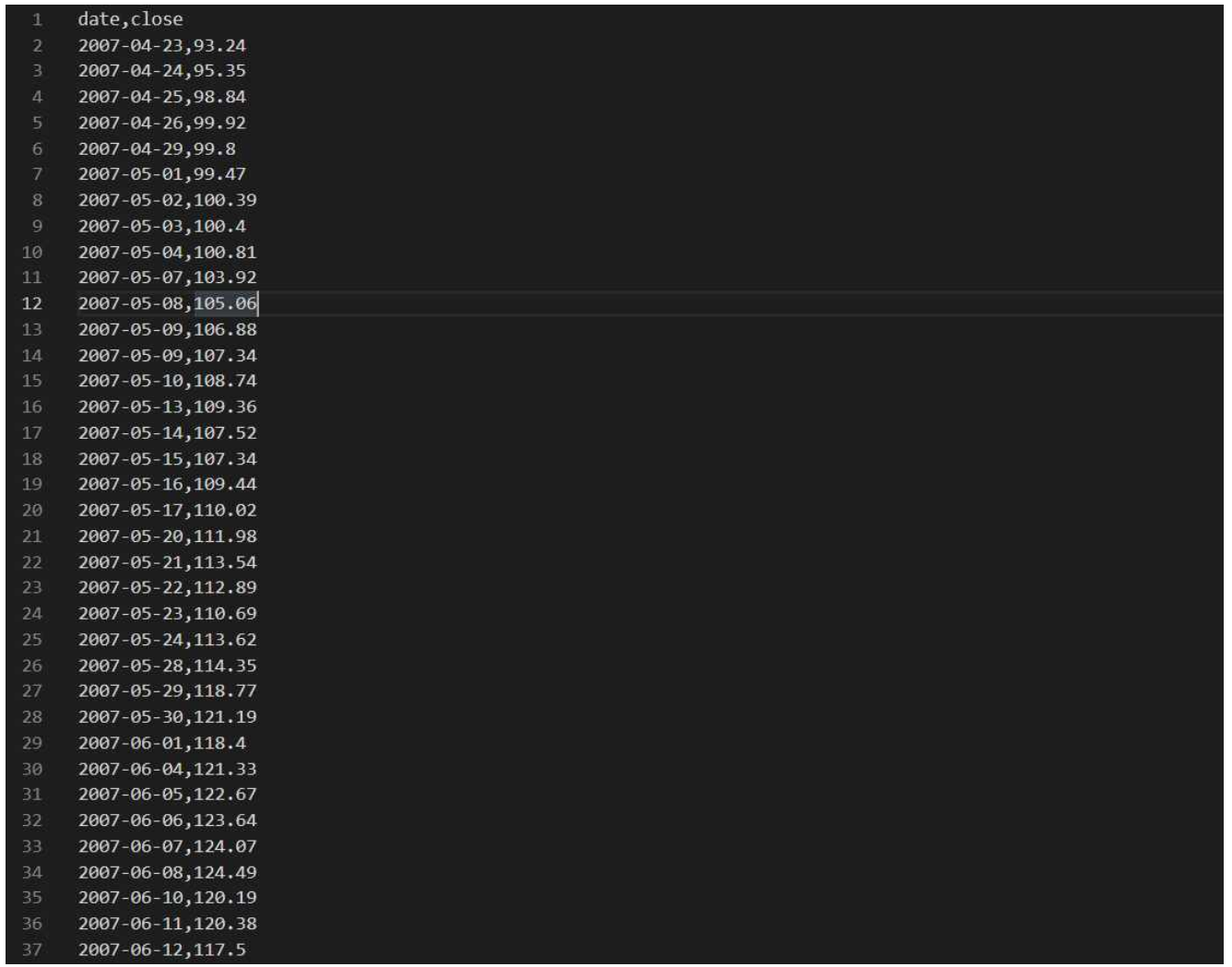
Week 3 Study Topics

- Line chart
  - Code analysis
  - Studying functions
- 나만의 예제 만들기(CVOID-19 누적 확진자 추세)

Line chart


D3 Line Chart
localhost:8000/d3.js/linechart/example.html
650 – $ Close
600 –
550 –
500 –
450 –
400 –
350 –
300 –
250 –
200 –
150 –
100 –
50 –
0
July
October
2008
April
July
October
2009
April
July
October
2010
April
July
October
2011
April
July
October
2012
April

Input



```
date,close
2007-04-23,93.24
2007-04-24,95.35
2007-04-25,98.84
2007-04-26,99.92
2007-04-29,99.8
2007-05-01,99.47
2007-05-02,100.39
2007-05-03,100.4
2007-05-04,100.81
2007-05-07,103.92
2007-05-08,105.06
2007-05-09,106.88
2007-05-09,107.34
2007-05-10,108.74
2007-05-13,109.36
2007-05-14,107.52
2007-05-15,107.34
2007-05-16,109.44
2007-05-17,110.02
2007-05-20,111.98
2007-05-21,113.54
2007-05-22,112.89
2007-05-23,110.69
2007-05-24,113.62
2007-05-28,114.35
2007-05-29,118.77
2007-05-30,121.19
2007-06-01,118.4
2007-06-04,121.33
2007-06-05,122.67
2007-06-06,123.64
2007-06-07,124.07
2007-06-08,124.49
2007-06-10,120.19
2007-06-11,120.38
2007-06-12,117.5
```

Line chart - Code 1

```
d3.csv("aapl.csv", ({ date, close }) => ({ date, value: +close }))
    .then((d) => {
        const data = Object.assign(d, { y: "$ Close" });
        const margin = { top: 20, right: 30, bottom: 30, left: 40 };
        const height = 500;
        const width = 1000;

        const parseDate = d3.timeParse("%Y-%m-%d");

        x = d3.scaleUtc()
            .domain(d3.extent(data, d => parseDate(d.date)))
            .range([margin.left, width - margin.right])

        y = d3.scaleLinear()
            .domain([0, d3.max(data, d => d.value)]).nice()
            .range([height - margin.bottom, margin.top]);

        line = d3.line()
            .defined(d => !isNaN(d.value))
            .x(d => x(parseDate(d.date)))
            .y(d => y(d.value));

        xAxis = g => g
            .attr("transform", `translate(0,${height - margin.bottom})`)
            .call(d3.axisBottom(x).ticks(width/80).tickSizeOuter(0));

        yAxis = g => g
            .attr("transform", `translate(${margin.left},0)`)
            .call(d3.axisLeft(y))
            .call(g => g.select(".domain").remove())
            .call(g => g.select(".tick:last-of-type text").clone()
                .attr("x", 3)
                .attr("text-anchor", "start")
                .attr("font-weight", "bold")
                .text(data.y));
```

Code analysis

- const parseDate = d3.timeParse("%Y-%m-%d");
  - %Y-%m-%d 형식의 string input을 d3에서 사용하는 time format으로 parsing
  - 이후 x축 값에 대해서 모두 parseDate(d.date)로 적용

```
8       const parseDate = d3.timeParse("%Y-%m-%d");
9
10      x = d3.scaleUtc()
11          .domain(d3.extent(data, d => parseDate(d.date)))
12          .range([margin.left, width - margin.right])
```

Code analysis

- line = d3.line()
  - line generator : line chart에서의 line을 표시하는 함수
- .defined(d => !isNaN(d.value))
  - defined는 define된 data에 대해서만 뒤의 x, y 접근자를 실행시킴
  - missing data에 대한 표현 가능(undefined data는 표시하지 않아 끊어진 line chart 표현 가능)
  - 이 예제에서는 NaN인 데이터를 filtering

```
18          line = d3.line()
19              .defined(d => !isNaN(d.value))
20              .x(d => x(parseDate(d.date)))
21              .y(d => y(d.value));
```

Line chart - Code 2

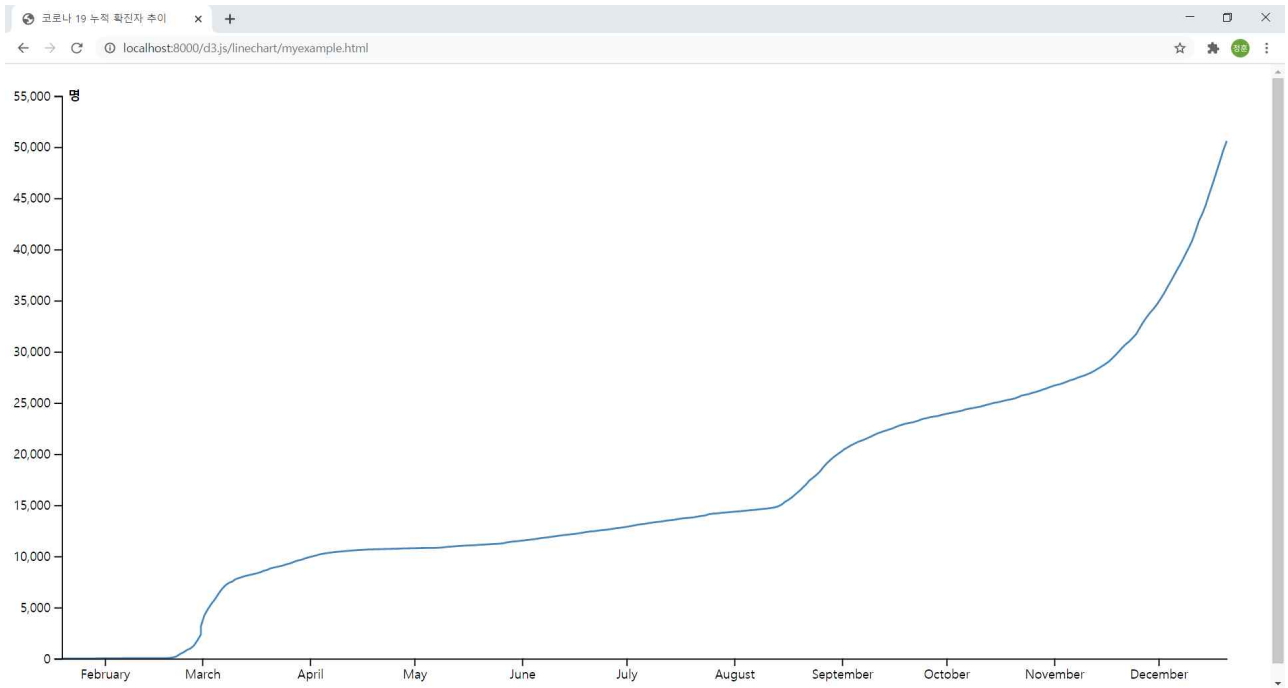
```
37          const svg = d3.select("svg")
38              .attr("viewBox", [0, 0, width, height]);
39
40          svg.append("path")
41              .datum(data)
42              .attr("fill", "none")
43              .attr("stroke", "steelblue")
44              .attr("stroke-width", "1.5px")
45              .attr("stroke-linejoin", "round")
46              .attr("stroke-linecap", "round")
47              .attr("d", line);
48
49          svg.append("g")
50              .call(xAxis);
51
52          svg.append("g")
53              .call(yAxis);
54      })
```

Code analysis

- .datum(data)
  - .datum() vs .data()
    - .datum() : 단일 요소에 단일 데이터 지정
    - .data() : 여러 요소에 데이터셋을 지정
    - bar chart에서는 selectAll을 통해 여러 rect에 data를 연결했지만
    - line chart에서는 path에 datum으로만 넣어줘도 path의 d 속성의 line(line generator)를 통해 시각화 가능
- .attr("d", line);
  - d 속성 : path의 모양을 결정짓는 속성
  - line(line generator)를 넣어주고 datum을 통해 값을 받아 표시

나만의 예제(COVID-19 누적 확진자 추세)

- 데이터 출처 : https://ko.wikipedia.org/wiki/코로나19_범유행
- line chart, x-axis : 날짜, y-axis : 누적 확진자 수



Input

```
date,value
01-20,1
01-24,1
01-26,2
01-27,3
01-30,4
01-31,6
02-01,11
02-02,12
02-04,15
02-05,16
02-06,19
02-07,23
02-09,24
02-10,27
02-11,27
02-12,28
02-15,28
02-16,28
02-17,29
02-18,30
02-19,31
02-20,51
02-21,104
02-22,204
02-23,433
02-24,602
02-25,833
02-26,977
02-27,1261
```

Code - 1

```
d3.csv("corona.csv", ({ date, value }) => ({ date, value: +value }))
    .then((d) => {
        const data = Object.assign(d, { y: "명" });
        const margin = { top: 20, right: 30, bottom: 30, left: 40 };
        const height = 500;
        const width = 1000;

        const parseDate = d3.timeParse("%m-%d");

        x = d3.scaleUtc()
            .domain(d3.extent(data, d => parseDate(d.date)))
            .range([margin.left, width - margin.right])

        y = d3.scaleLinear()
            .domain([0, d3.max(data, d => d.value)]).nice()
            .range([height - margin.bottom, margin.top]);

        line = d3.line()
            .defined(d => !isNaN(d.value))
            .x(d => x(parseDate(d.date)))
            .y(d => y(d.value));
```

Code - 2

```
23          xAxis = g => g
24              .attr("transform", `translate(0,${height - margin.bottom})`)
25              .call(d3.axisBottom(x).ticks(width/80).tickSizeOuter(0));
26
27          yAxis = g => g
28              .attr("transform", `translate(${margin.left},0)`)
29              .call(d3.axisLeft(y))
30              .call(g => g.select(".tick:last-of-type text").clone()
31                  .attr("x", 5)
32                  .attr("text-anchor", "start")
33                  .attr("font-weight", "bold")
34                  .text(data.y));
35
36          const svg = d3.select("svg")
37              .attr("viewBox", [0, 0, width, height]);
38
39          svg.append("path")
40              .datum(data)
41              .attr("fill", "none")
42              .attr("stroke", "steelblue")
43              .attr("stroke-width", "1.5px")
44              .attr("d", line);
45
46          svg.append("g")
47              .call(xAxis);
48
49          svg.append("g")
50              .call(yAxis);
51      })
```