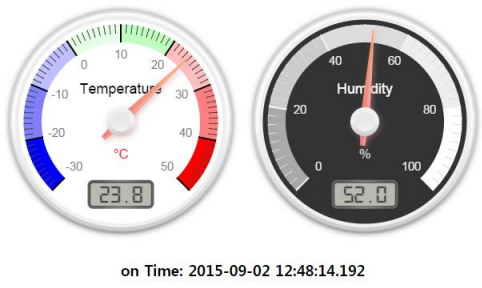
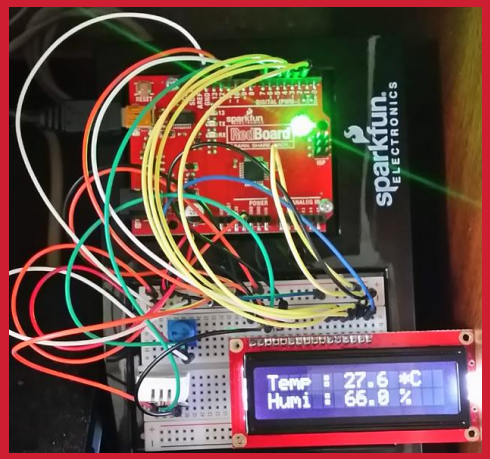
Healthcare-IOT
[wk09]





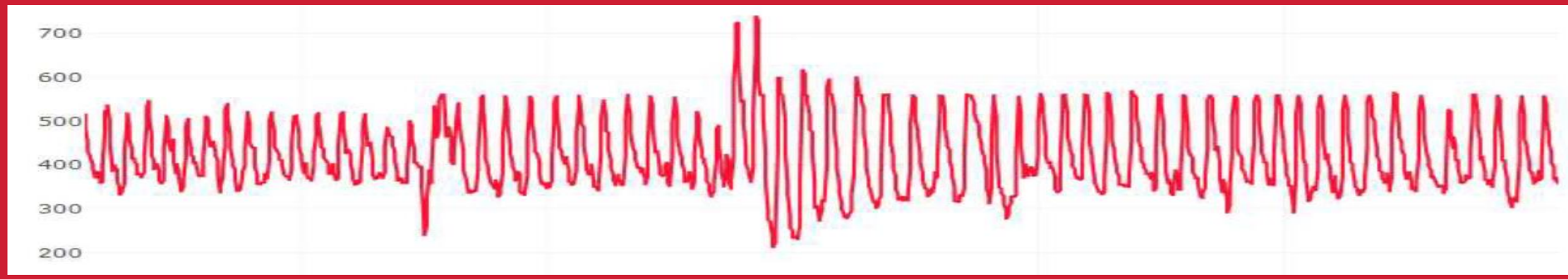
Data visualization using plotly.js I.

Visualization of Healthcare Signals using Arduino & Node.js

HCit, INJE University

1st semester, 2018

Email : chaos21c@gmail.com



My ID

오전

성명	ID
김민선	HS01
김영걸	HS02
김주란	HS03
김주현	HS04
김태민	HS05
여준하	HS06
이수민	HS07
정민지	HS08
정유현	HS09
정재은	HS10
주하영	HS11
한준영	HS12

오후

성명	ID
신영주	HS21
오가영	HS22
윤민수	HS23
윤진아	HS24
이진영	HS25
임상은	HS26
임재형	HS27
최민영	HS28
황유빈	HS29

주간계획서

주간계획서			
주차	수업방법	수업내용	과제물
1	강의/실습	수업 및 실습 안내 -포터블 소프트웨어 설치	
2	강의/실습	Node.js I - Node.js 코드의 기본 구조 - 기초 Node 서버 및 클라이언트	실습확인
3	강의/실습	Node.js II - Node.js Express 서버	실습확인
4	강의/실습/발표	Arduino I - 아날로그 신호 회로 - LCD를 이용한 센서 신호 모니터링	실습확인
5	강의/실습	Arduino II - 단일 센서 회로와 Node.js 연결 - 다중 센서 회로와 Node.js 연결	실습확인
6	강의/실습	프로젝트1 - 생체 센서 회로와 Node.js 연결 - 생체 신호 소개	프로젝트1
7	강의/실습/발표	IOT 데이터 시각화 I (Plotly.js) - 데이터 및 시계열 차트 - 데이터 스트리밍	실습확인
8	시험	중간고사	
9	강의/실습	IOT 데이터 시각화 II (Plotly.js) - 다중 센서 데이터 시각화 - 다중 센서 데이터 스트리밍	실습확인
10	강의/실습/발표	프로젝트II - 생체 센서 데이터 시각화 - 생체 센서 데이터 스트리밍	프로젝트II
11	강의/실습	IOT 데이터 저장과 처리 - MongoDB 설치 및 Mongo shell - MongoDB와 Node.js 연결 및 데이터 저장	실습확인
12	강의/실습	프로젝트III - MongoDB에 IOT 데이터 저장 및 모니터링 - 생체 센서 데이터 저장 및 시각화	프로젝트III
13	강의/실습	IOT 데이터 마이닝 - 아두이노에서 발생된 데이터 관리 - 데이터마이닝 소개	실습확인
14	강의/실습/발표	프로젝트IV - 생체 센서 데이터 관리 - 생체 센서 데이터 마이닝	프로젝트IV
15	시험	기말고사	

Purpose of HS

주요 수업 목표는 다음과 같다.

1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리
4. 생체 센서 발생 신호 처리, 시각화 및 저장
5. 생체 센서 발생 신호 저장 및 분석
6. 생체 신호 장비 활용 능력

[Review]

- [wk07]
  - Arduino + Node.js II. multi sensors
  - Complete your project
  - Submit file : HSnn_Rpt06.zip

wk07 : Practice-mid : HSnn_Rpt06.zip

- [Target of this week – 10 points]
  - Complete your works
  - Save your outcomes and compress 5 outputs

제출파일명 : **HSnn_Rpt06.zip**

- 압축할 파일들

① **HSnn_cds_tmp36_lcd.png**

② **HSnn_cds_tmp36_IOT.png**

③ **HSnn_multi_signals.ino**

④ **hsnn_multi_signals_code.png**

⑤ **HSnn_multi_signals_node.png**

Email : chaos21c@gmail.com

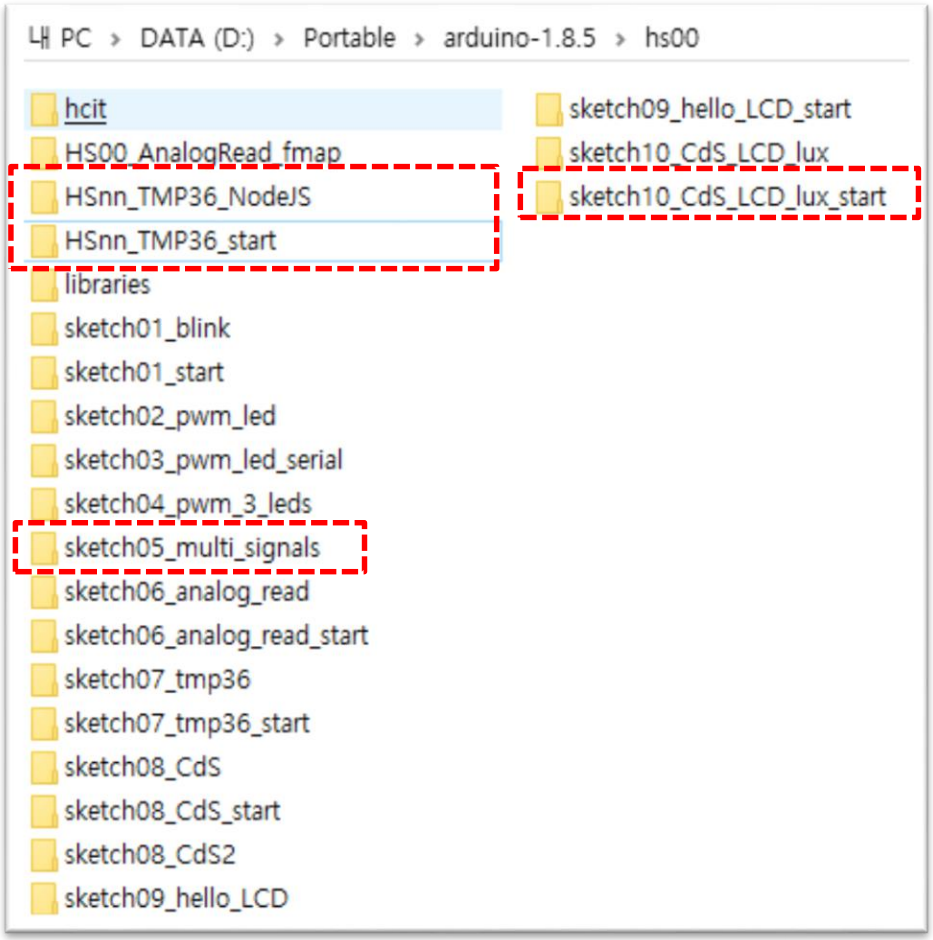
[제목 : id, 이름 (수정)]

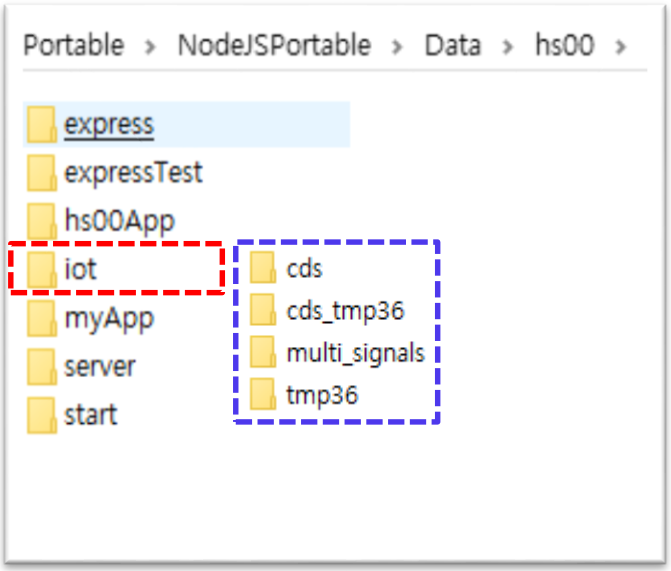
중간고사
Arduino & Node

[1] 과목명 : 헬스케어신호처리개론 (헬스케어IT 학과 2학년)
[2] 4월 26일(목), 5교시: 오후 1시~2시 (장소 : F1002)
[3] 필기 시험 (20점)
- Arduino & Node Coding에서 20문제 (선다형/단답형/서술형)
- 수업 시간에 배운 Code의 이해와 활용 능력 평가

**** 잘 준비해서 웃기를 ... ^_^

[My working folder – wk07]

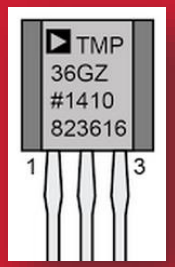


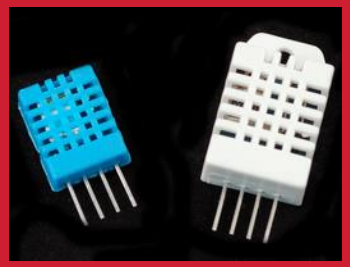


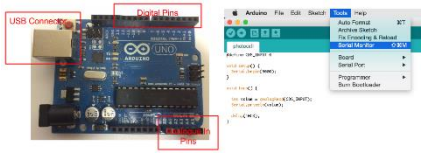
Arduino

+ Node.js

+ plotly.js







IOT: HSC

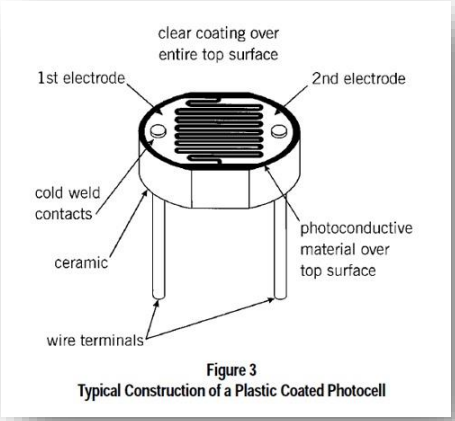
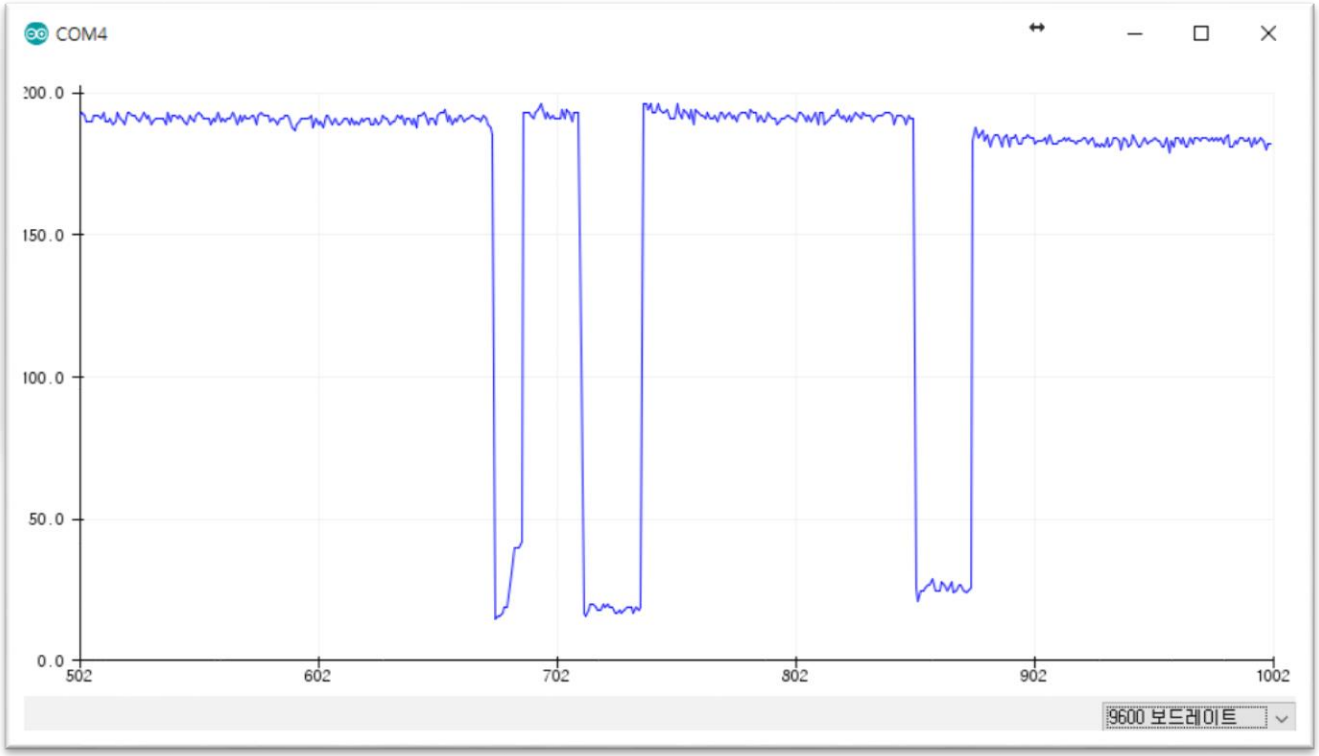
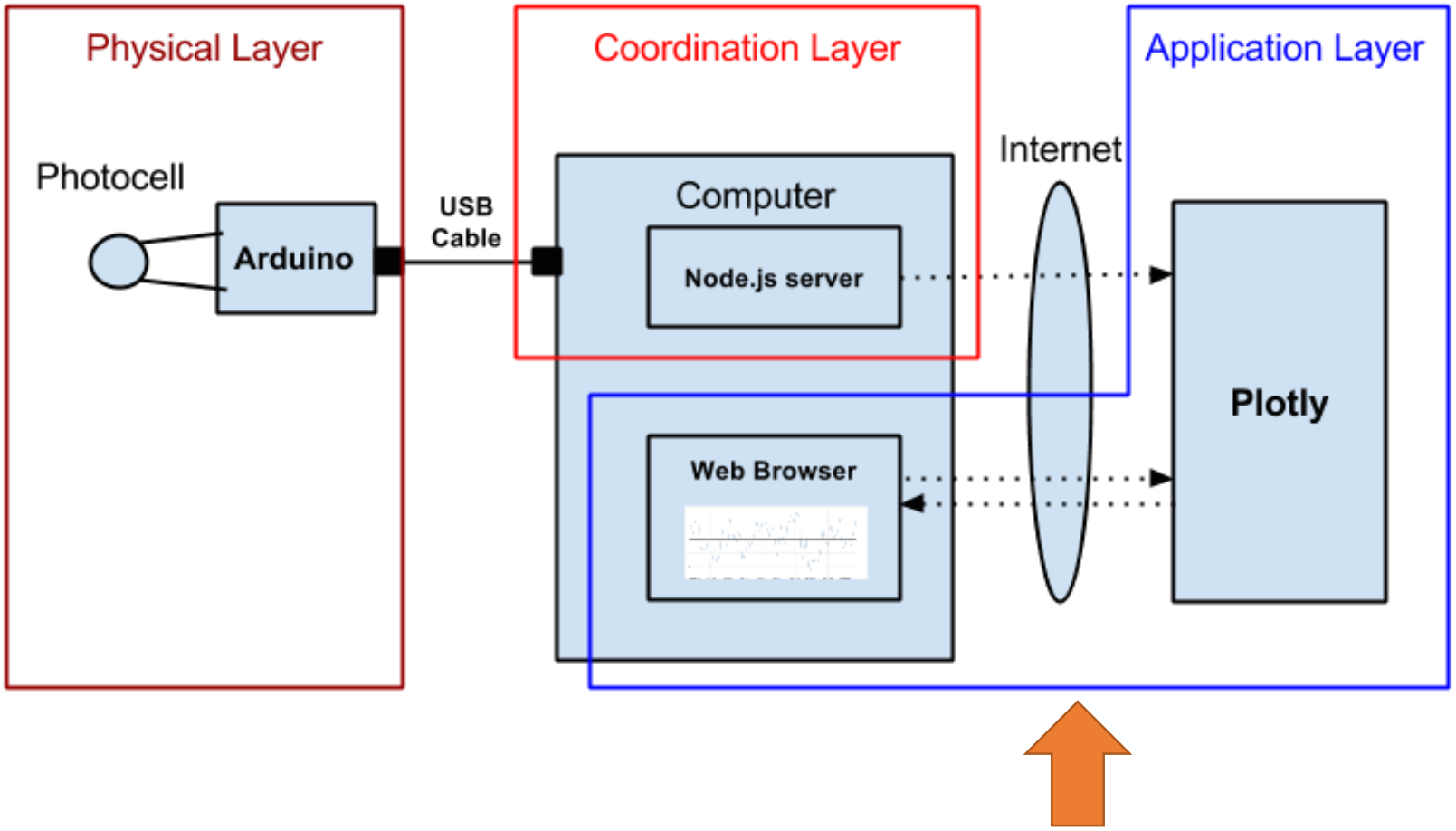


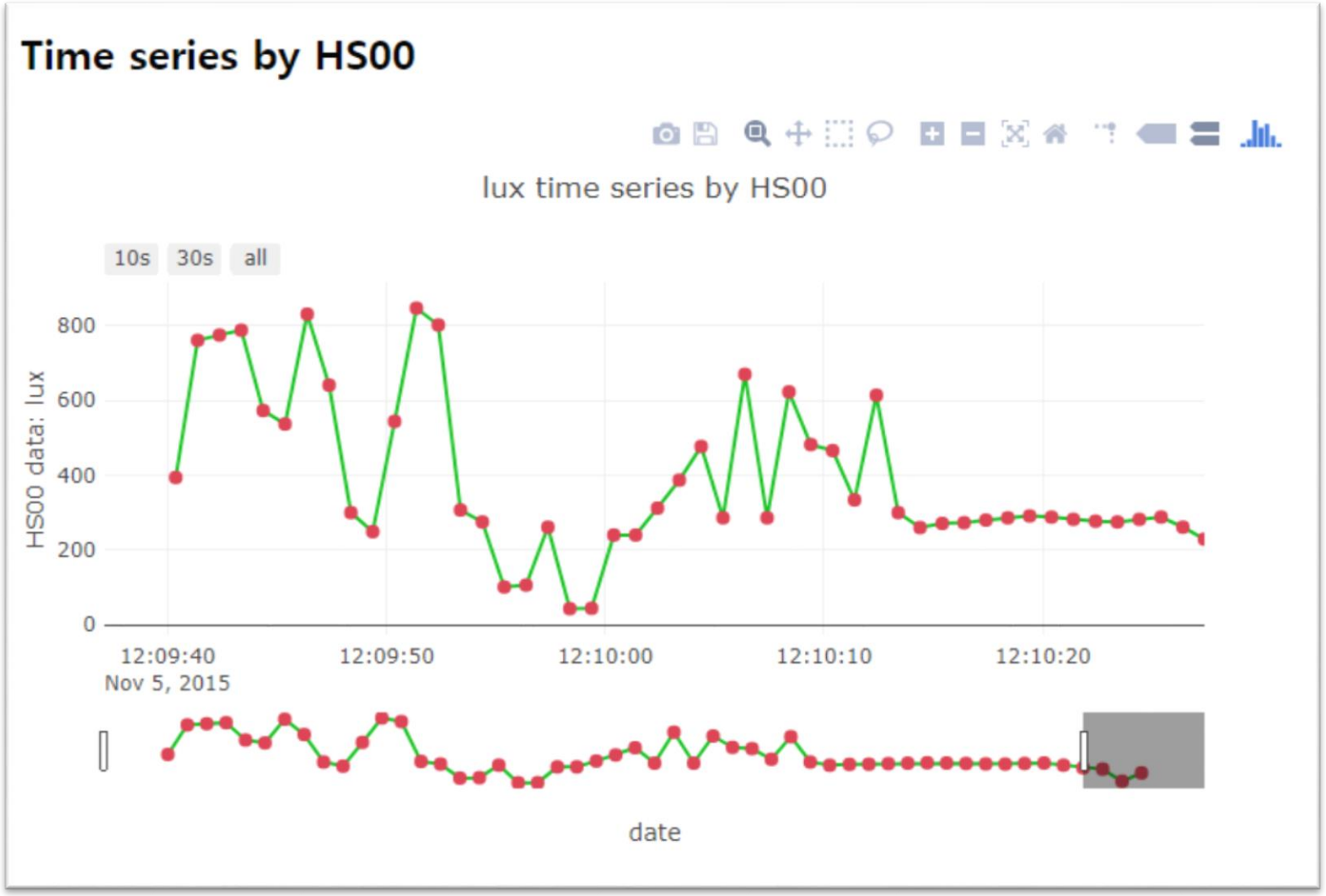
Figure 3
Typical Construction of a Plastic Coated Photocell



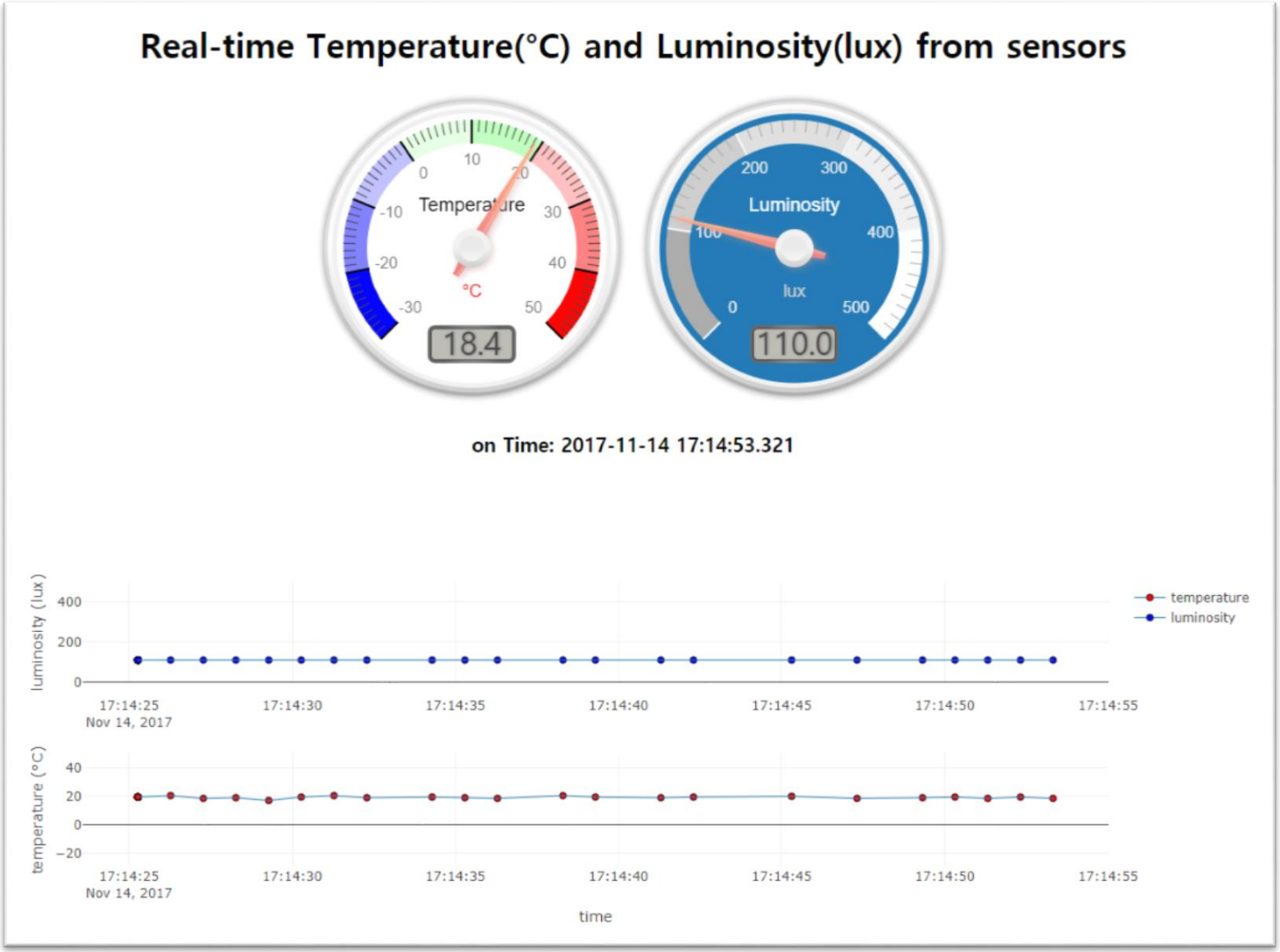
Layout [H S C]

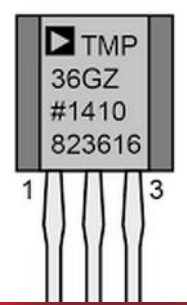

Physical Layer
Photocell
Arduino
USB Cable
Coordination Layer
Computer
Node.js server
Web Browser
Internet
Application Layer
Plotly

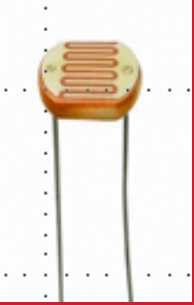
Arduino data + plotly



Arduino: node.js + plotly

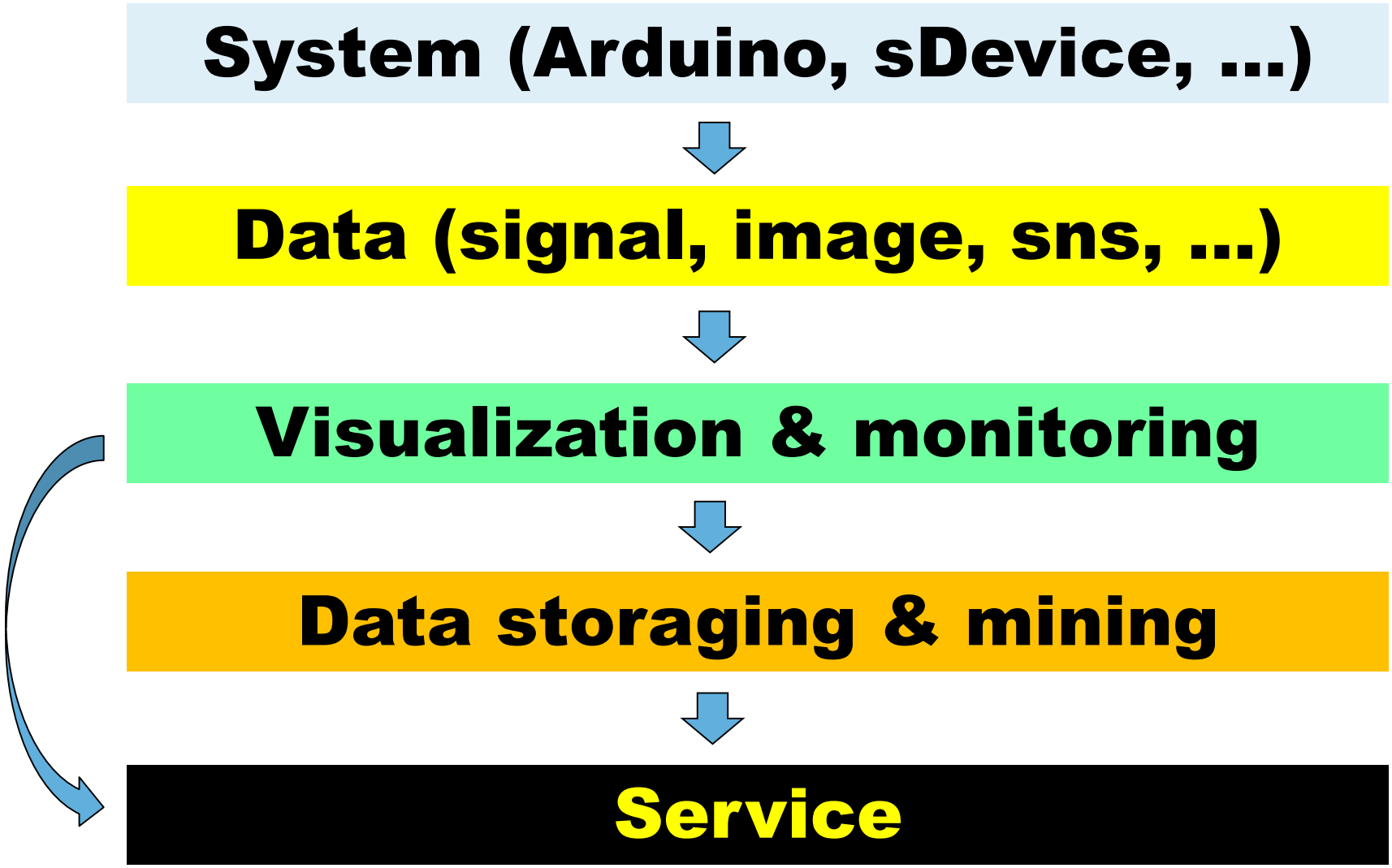




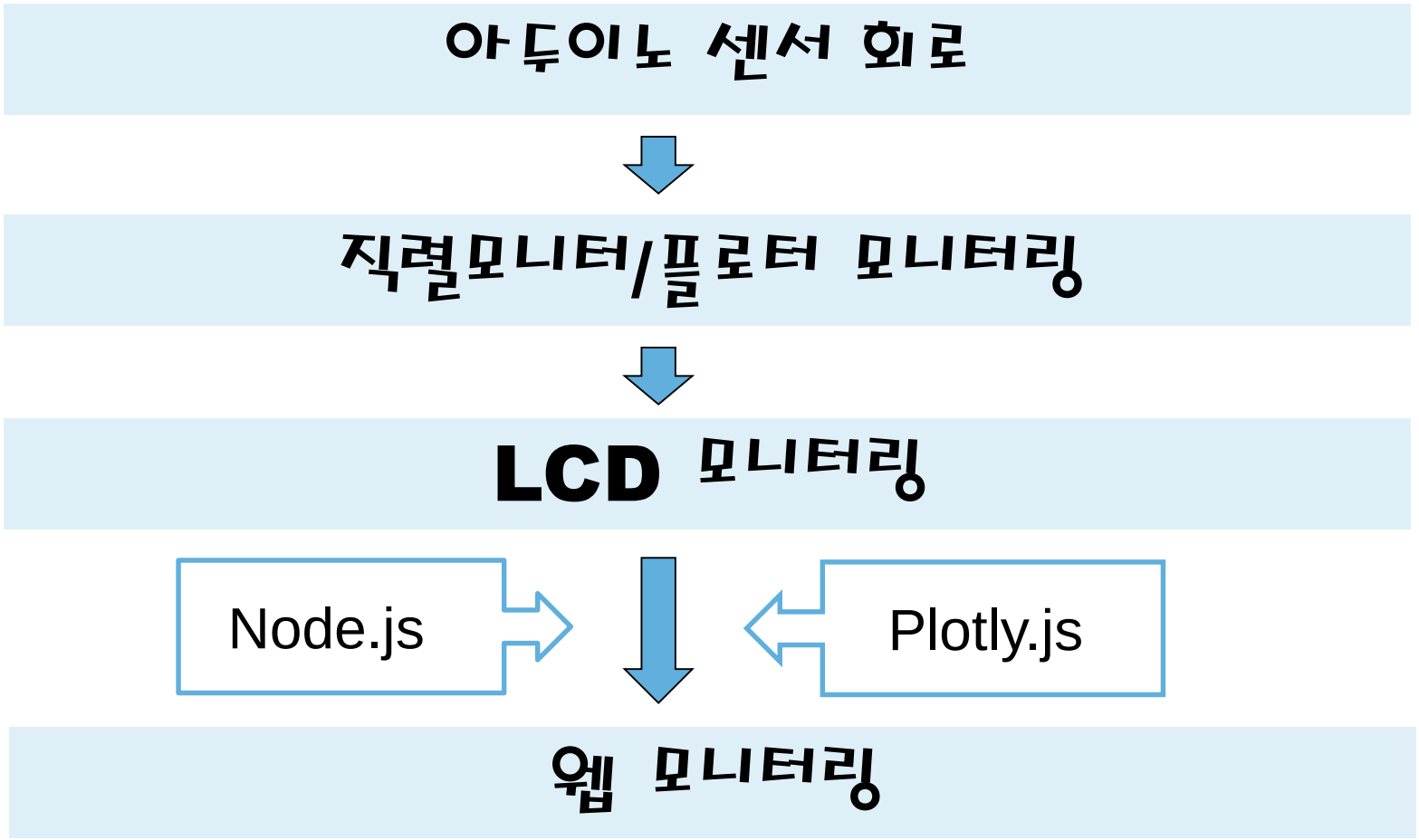


Data visualization using plotly.js

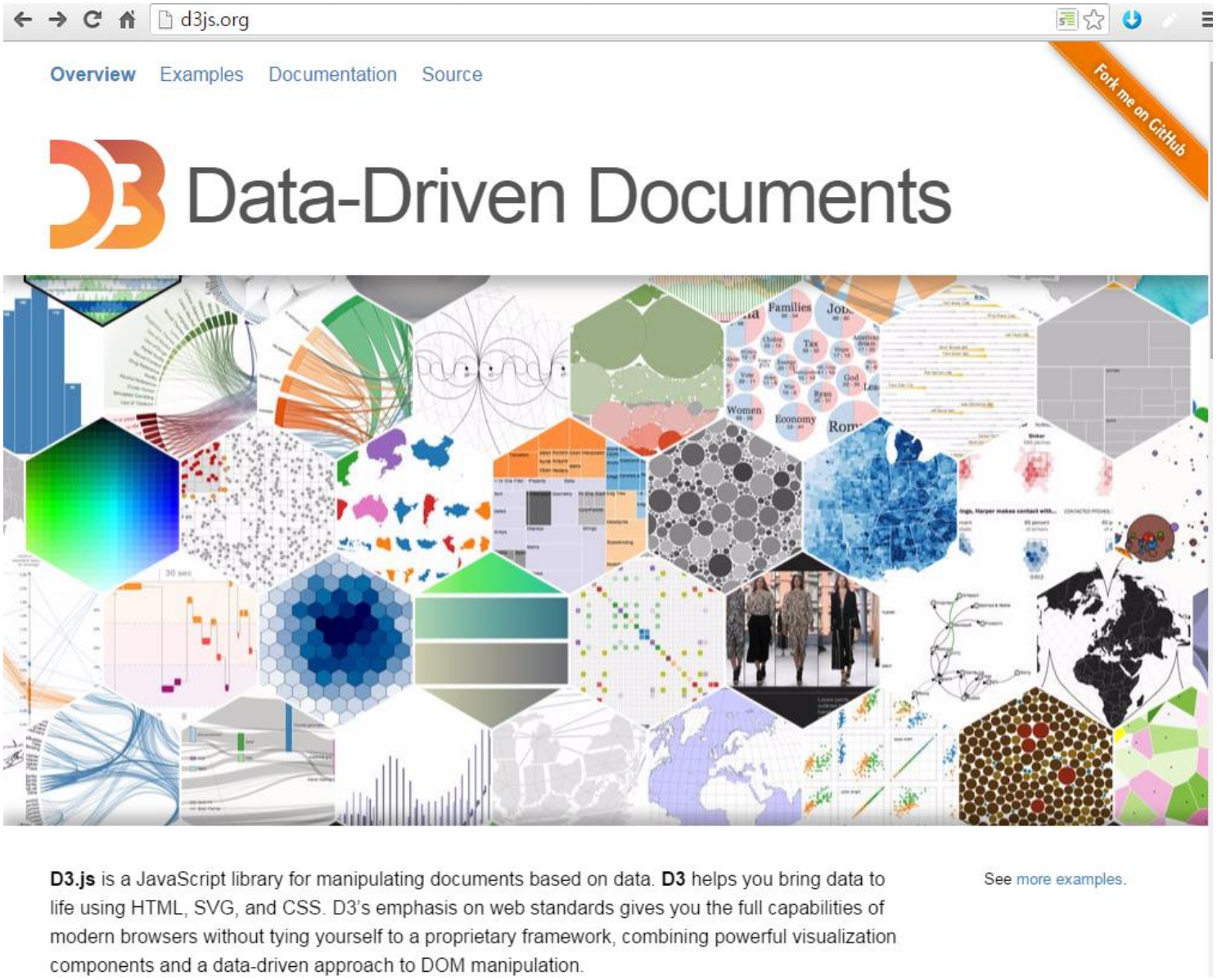
A5. Introduction to visualization



A5.1. Introduction to data visualization

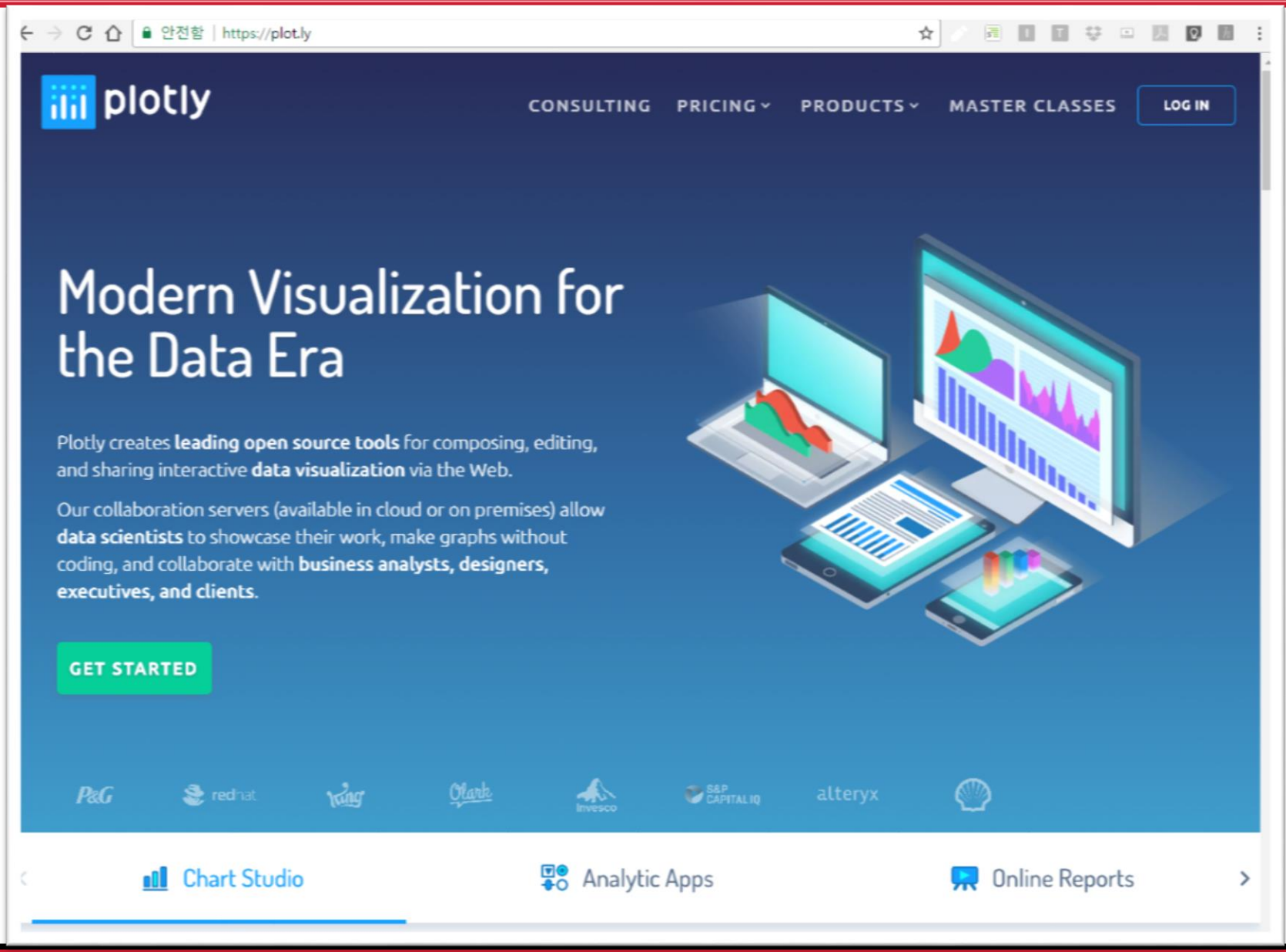


A5.1.1 D3.js



http://d3js.org/

A5.1.2 plot.ly



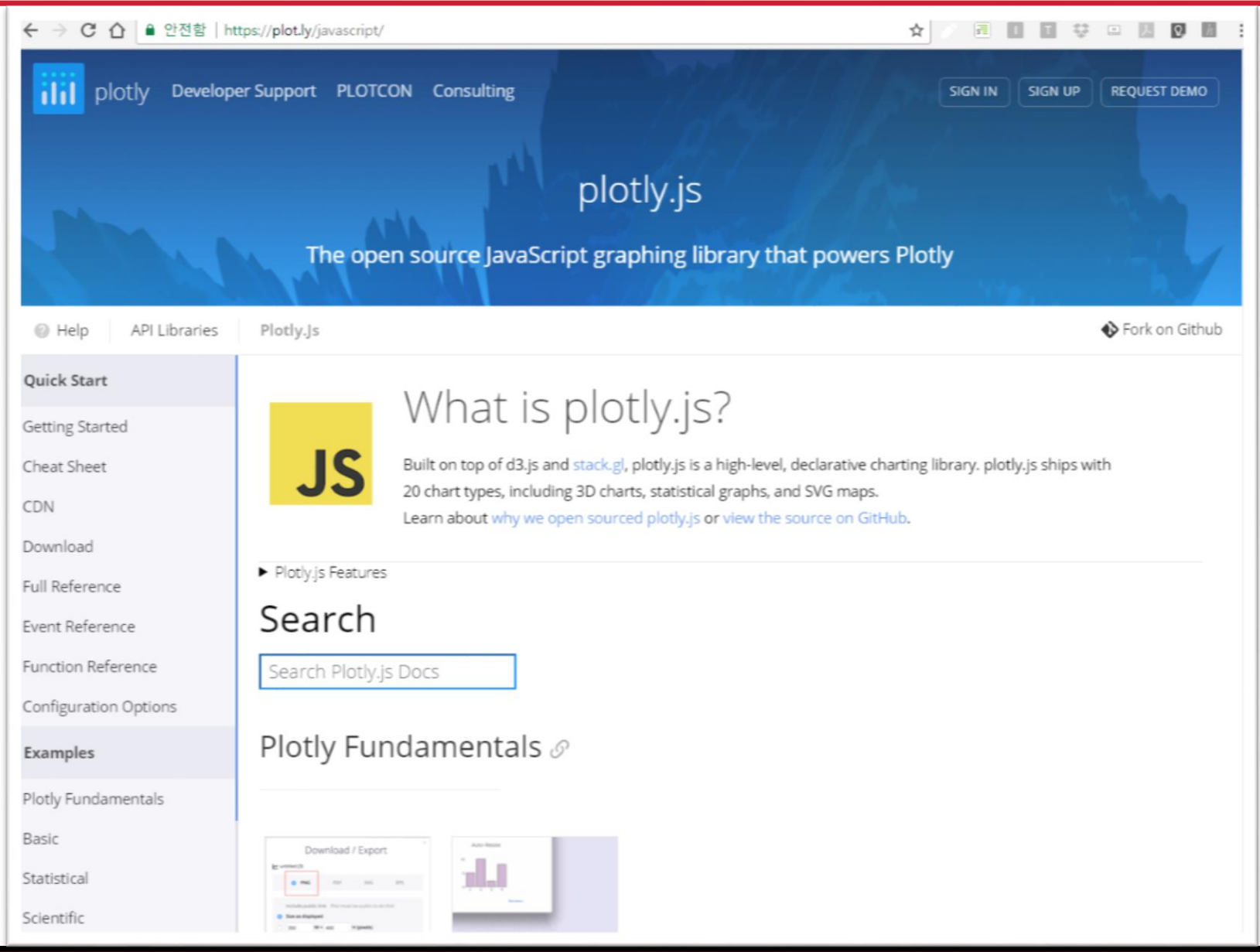
https://plot.ly/

A5.1.3 plotly.js

plotly.js is Plotly's **client-side, interactive JavaScript charting library**, built on top of **D3.js, stack.gl**, and **jQuery**.

https://plot.ly/javascript/

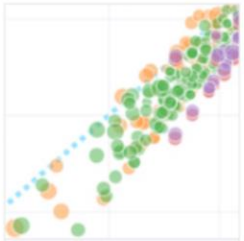
A5.1.4 Introduction to plotly.js



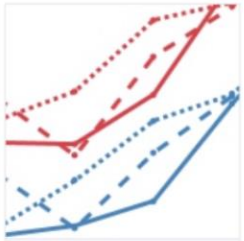
https://plot.ly/javascript/getting-started/#download

A5.1.5 Introduction to plotly.js charts

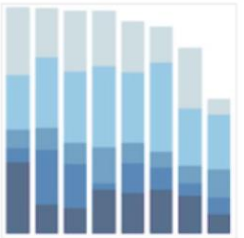
Basic Charts



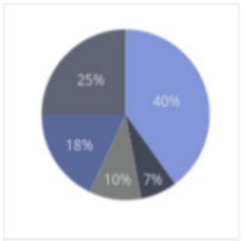
Scatter Plots



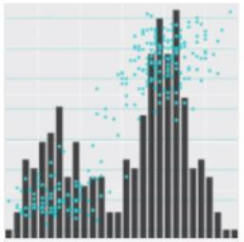
Line Charts



Bar Charts

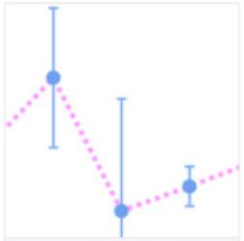


Pie Charts

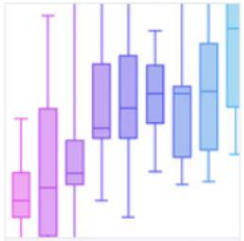


More Basic Charts

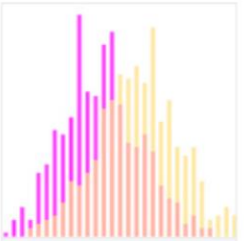
Statistical Charts



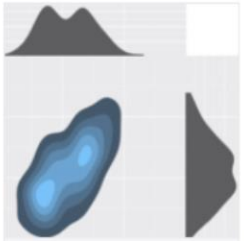
Error Bars



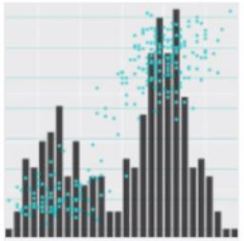
Box Plots



Histograms



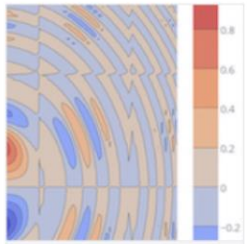
2d Density Plots



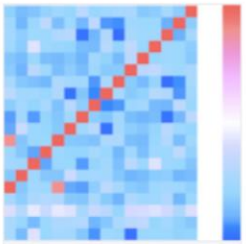
More Statistical

A5.1.6 Introduction to plotly.js charts

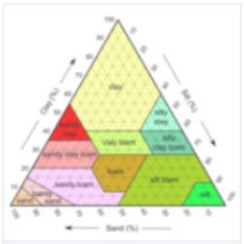
Scientific Charts



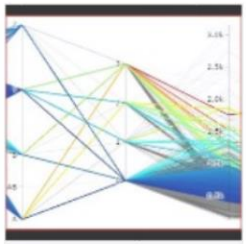
Contour Plots



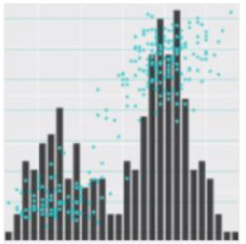
Heatmaps



Ternary Plots



Parallel Coordinates Plot

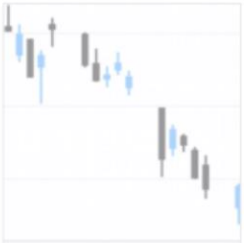


More Scientific Charts

Financial Charts

Time Series

OHLC Charts



Candlestick Charts

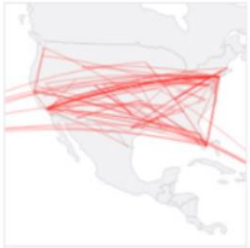
A5.1.7 Introduction to plotly.js charts

Maps

Choropleth Maps

Scatter Plots on Maps

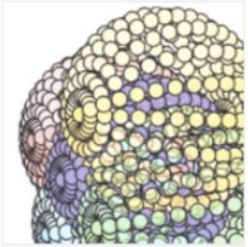
Bubble Maps



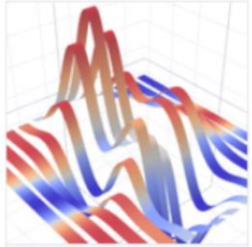
Lines on Maps

Scatter Plots on Mapbox

3D Charts

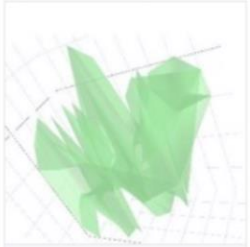


3D Scatter Plots

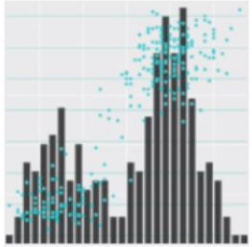


Ribbon Plots

3D Surface Plots



3D Mesh Plots



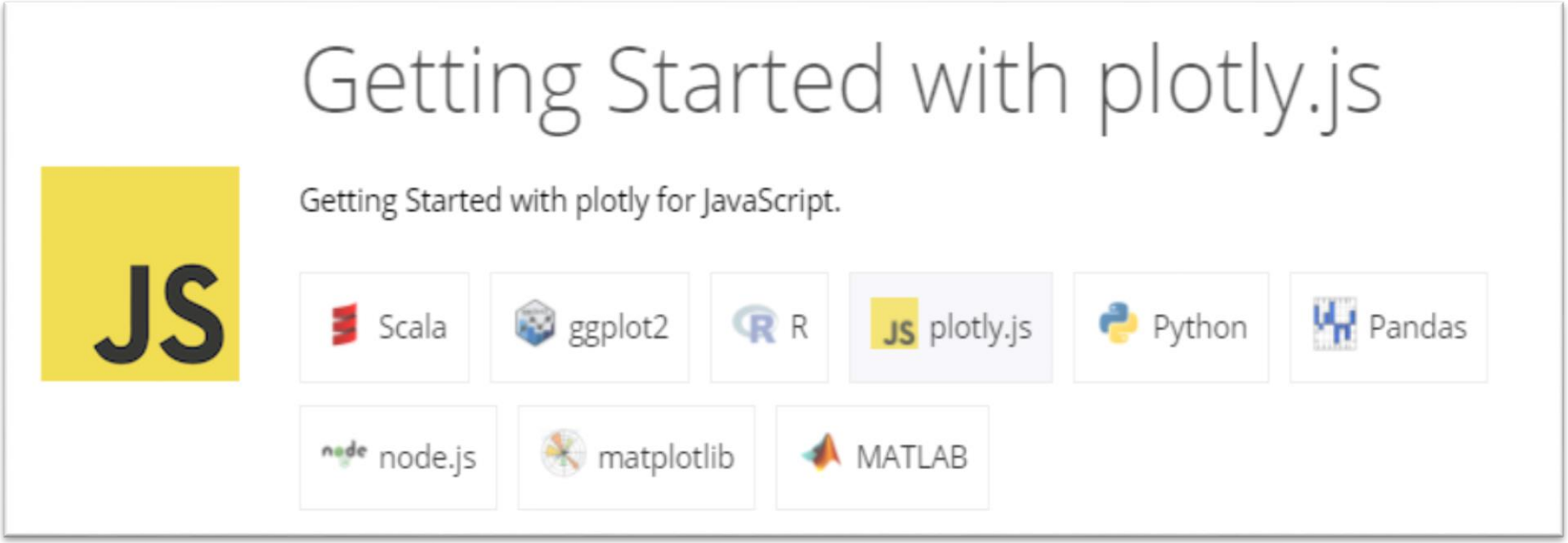
More 3D Charts

A5.1.8 plotly.js: time series & streaming

https://plot.ly/javascript/time-series/

https://plot.ly/javascript/streaming/

A5.1.9 Getting started: plotly.js



https://plot.ly/javascript/getting-started/

https://plot.ly/

A5.1.10 Getting started: plotly.js

plotly.js CDN

You can also use the ultrafast plotly.js CDN link. This CDN is graciously provided by the incredible team at Fastly.

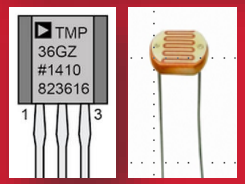
```
<head>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
</head>
```

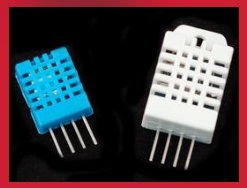
Else, if you want to get a specific version of plotly.js, say 1.2.0:

```
<head>
    <script src="https://cdn.plot.ly/plotly-1.2.0.min.js"></script>
</head>
```

<script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

https://plot.ly/javascript/getting-started/





Data charts using plotly.js

A5.2 Data charts

Navigation

- Basic Line Plot
- Line and Scatter Plot
- Adding Names to Line and Scatter Plot
- Line and Scatter Styling
- Styling Line Plot
- Colored and Styled Scatter Plot
- Line Shape Options for Interpolation
- Graph and Axes Titles
- Line Dash
- Connect Gaps Between Data
- Labelling Lines with Annotations

← Back To Plotly.Js

Line Charts in plotly.js

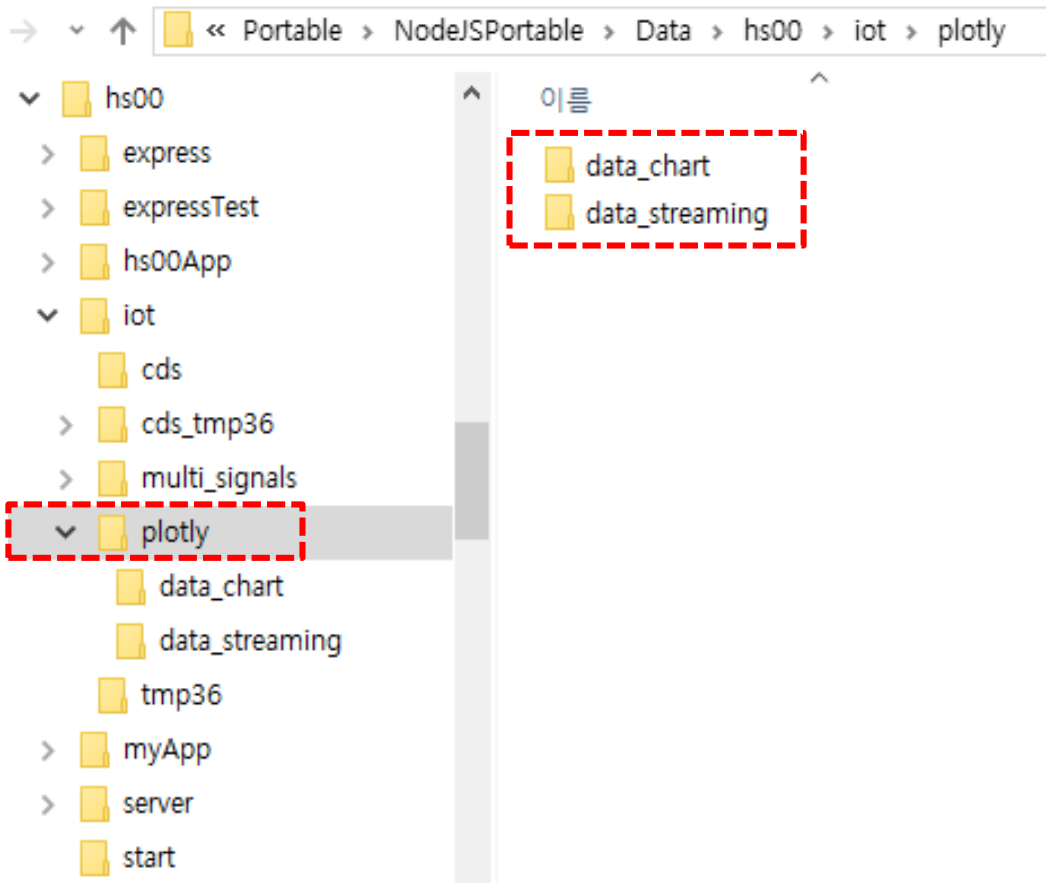
How to make D3.js-based line charts in JavaScript.

R | plotly.js | Python | Pandas

Basic Line Plot

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  type: 'scatter'
};

var trace2 = {
  x: [1, 2, 3, 4],
  y: [16, 5, 11, 9],
  type: 'scatter'
};
```

A5.2.1 Working folders



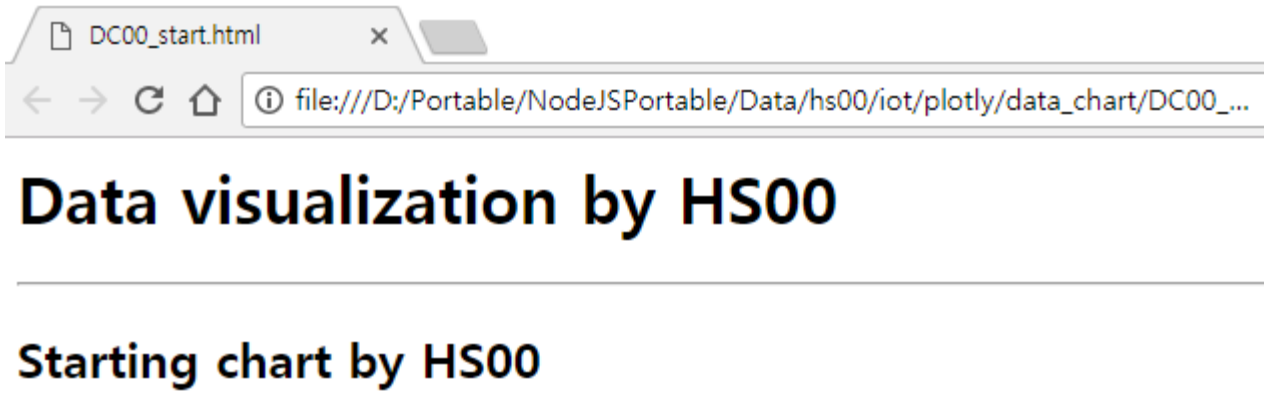
A5.2.2.1 Starting plotly basic chart

DC00_start.html

Starting chart!

```
1  <html>
2  <head>
3      <meta charset="utf-8">
4      <!-- Plotly.js -->
5      <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
6  </head>
7  <body>
8      <h1>Data visualization by HS00</h1>
9      <hr>
10     <h2>Starting chart by HS00</h2>
11
12     <!-- Plotly chart will be drawn inside this DIV -->
13     <div id="myDiv" style="width: 500px;height: 300px"></div>
14
15     <script>
16         <!-- JAVASCRIPT CODE GOES HERE -->
17
18
19     </script>
20 </body>
21 </html>
22
```

A5.2.2.2 Starting plotly basic chart



[Tip] Using WEB browser in SB text3

[Tool] Sublime Text – 현재 작업 중인 파일을 웹브라우저로 열기

1. Tool > Developer > New Plugin을 실행 한 후 아래 내용으로 덮어 씌운 후 **'open_browser'**으로 저장한다**.**

```
import sublime, sublime_plugin
import webbrowser

class OpenBrowserCommand(sublime_plugin.TextCommand):
    def run(self,edit):
        url = self.view.file_name()
        webbrowser.open_new(url)
```

2. Preferences -> Key Bindings - User로 이동한 후 단축키를 할당한다**.**

```
{ "keys": ["f10"], "command": "open_browser" }
```

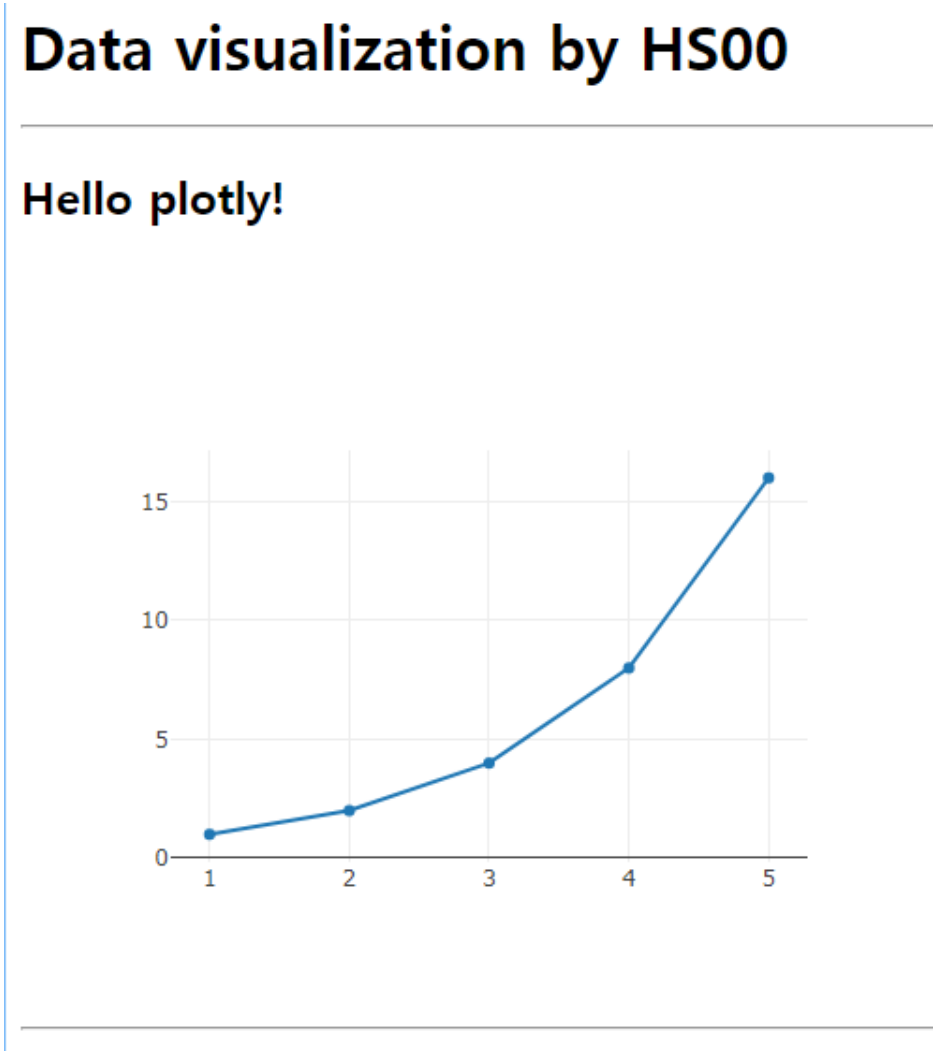
A5.2.3.1 Hello plotly basic chart

Hello plotly data chart!

```
1  <html>
2  <head>
3      <meta charset="utf-8">
4      <!-- Plotly.js -->
5      <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
6  </head>
7  <body>
8      <h1>Data visualization by HS00</h1>
9      <hr>
10     <h2>Hello plotly!</h2>
11     <!-- Plotly chart will be drawn inside this DIV -->
12     <div id="myDiv" style="width: 500px;height: 400px"></div>
13     <hr>
14     <script>
15         <!-- JAVASCRIPT CODE GOES HERE -->
16         var data = [
17         {
18             x: [1, 2, 3, 4, 5],
19             y: [1, 2, 4, 8, 16],
20             type: 'scatter'
21         }];
22
23         Plotly.newPlot('myDiv', data);
24
25     </script>
26 </body>
27 </html>
```

A5.2.3.2 Hello plotly basic chart

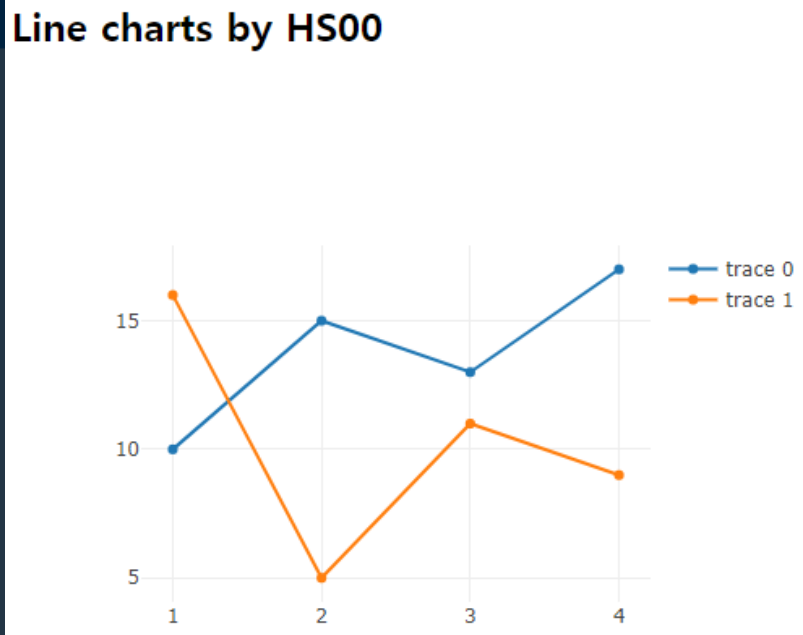
Graph : Hello plotly chart!



A5.2.4 plotly.js: Line Charts

[1] Basic multi-line charts

```
<script>
    <!-- JAVASCRIPT CODE GOES HERE -->

    var trace1 = {
        x: [1, 2, 3, 4],
        y: [10, 15, 13, 17],
        type: 'scatter'
    };

    var trace2 = {
        x: [1, 2, 3, 4],
        y: [16, 5, 11, 9],
        type: 'scatter'
    };

    var data = [trace1, trace2];

    Plotly.newPlot('myDiv', data);

</script>
```

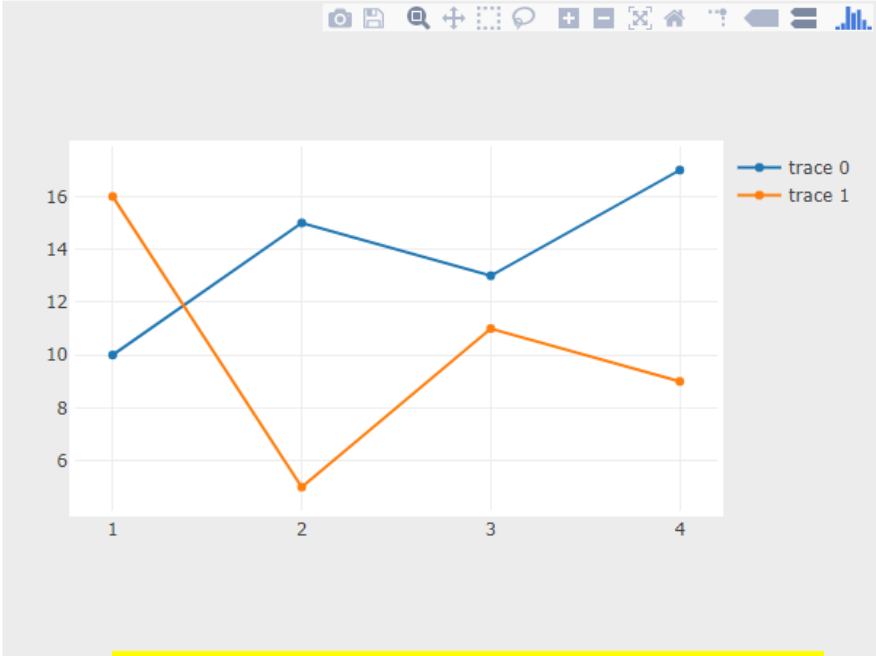


A5.2.5 plotly.js: Line Charts

[2] Basic line charts with layout

```
var layout = {
    autosize: false,
    width: 600,
    height: 450,
    margin: {
        l: 50,  // left
        r: 50,  // right
        b: 100, // bottom
        t: 100, // top
        pad: 4  // padding
    },
    paper_bgcolor: '#ececec',
    plot_bgcolor: '#ffffff' //'#rrggbb'
};

Plotly.newPlot('myDiv', data, layout);
```

Line charts with layout by HS00



HSnn_Chart_Layout.png

Test: pad → 40

A5.2.6.1 plotly.js: Line & Scatter plot

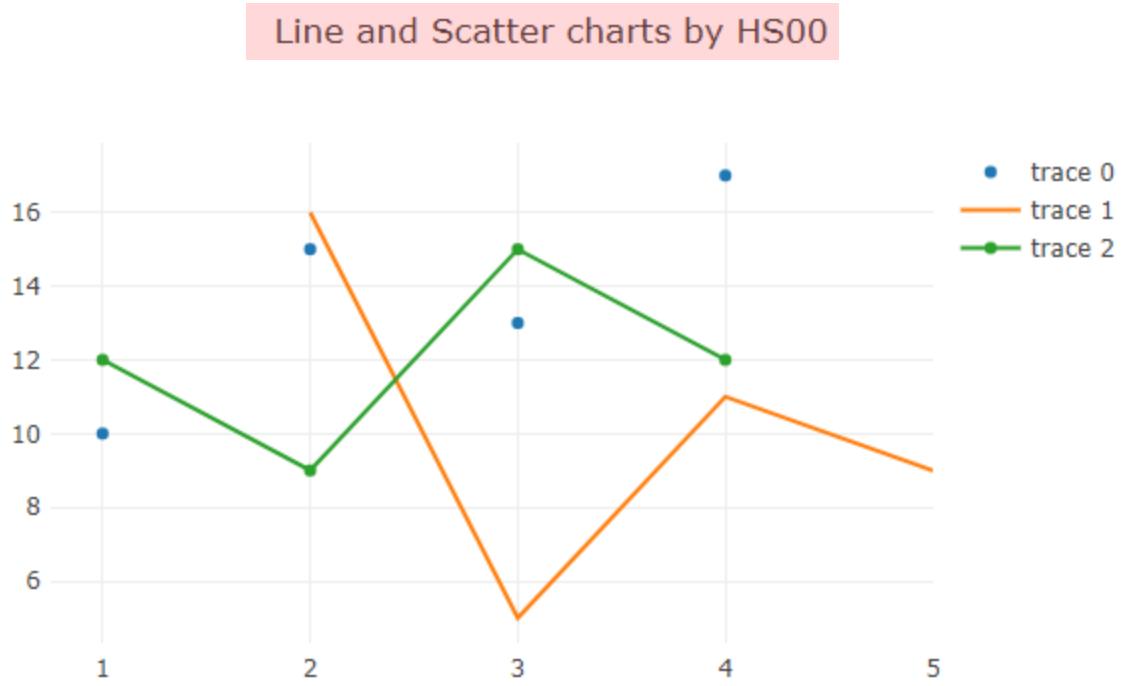
[3] Line & scatter plot

```
var trace1 = {
    x: [1, 2, 3, 4],
    y: [10, 15, 13, 17],
    mode: 'markers'
};

var trace2 = {
    x: [2, 3, 4, 5],
    y: [16, 5, 11, 9],
    mode: 'lines'
};

var trace3 = {
    x: [1, 2, 3, 4],
    y: [12, 9, 15, 12],
    mode: 'lines+markers'
};
```

```
var data = [ trace1, trace2, trace3 ];

var layout = {
    title:'Line and Scatter charts by HS00',
    width: 600,
    height: 450,
    margin: {
        l: 50,
        r: 50,
        b: 100,
        t: 100,
        pad: 4
    },
};

Plotly.newPlot('myDiv', data, layout);
```

A5.2.6.2 plotly.js: Line & Scatter plot

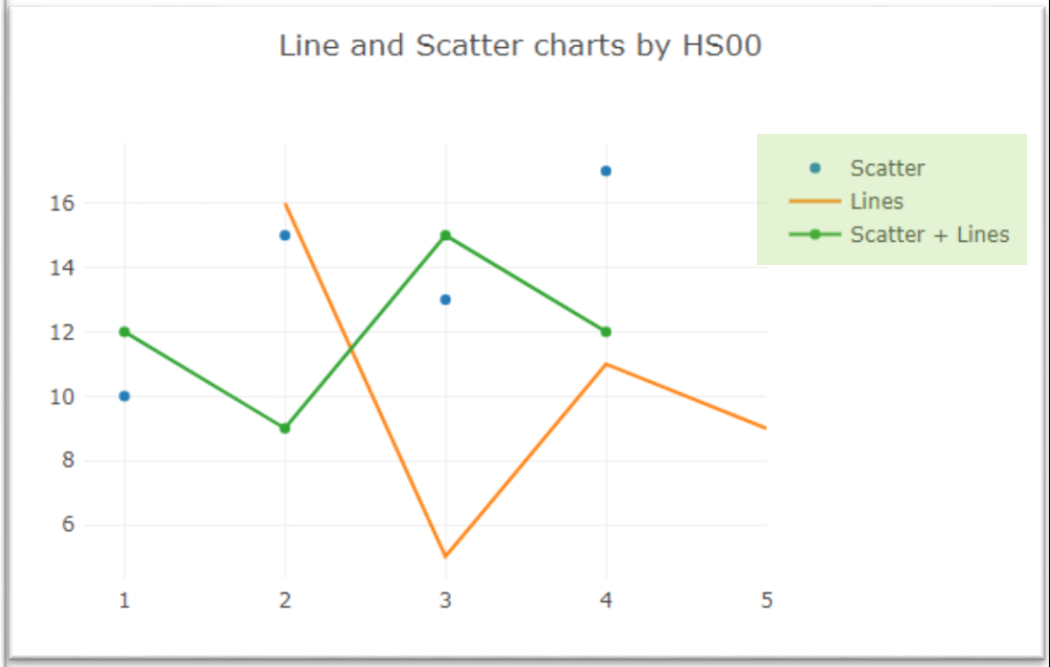
[3.1] Line & scatter plot with title



A5.2.6.3 plotly.js: Line & Scatter plot

[3.2] Line & scatter plot with axis name

```
var trace1 = {
    x: [1, 2, 3, 4],
    y: [10, 15, 13, 17],
    mode: 'markers',
    name: 'Scatter'
};

var trace2 = {
    x: [2, 3, 4, 5],
    y: [16, 5, 11, 9],
    mode: 'lines',
    name: 'Lines'
};

var trace3 = {
    x: [1, 2, 3, 4],
    y: [12, 9, 15, 12],
    mode: 'lines+markers',
    name: 'Scatter + Lines'
};
```

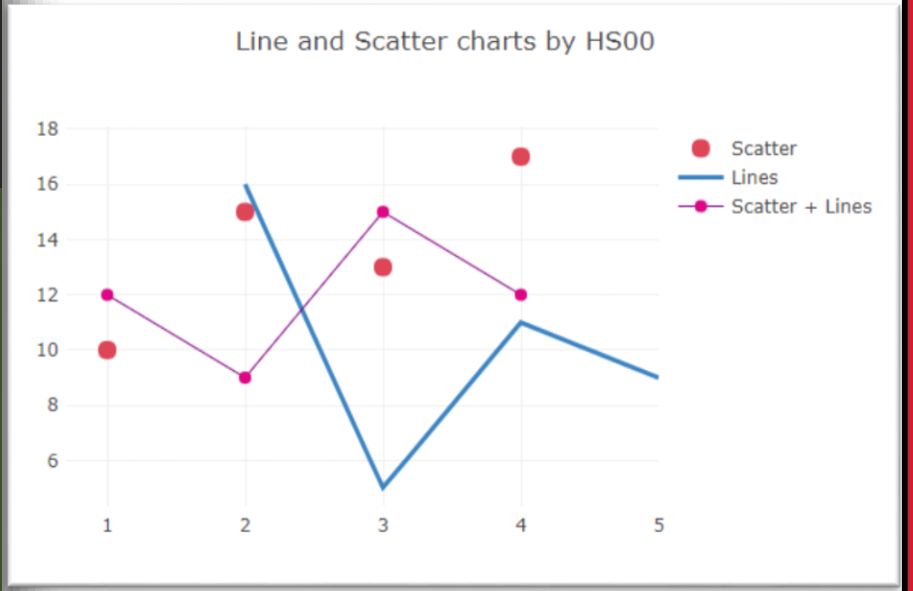


A5.2.6.4 plotly.js: Line & Scatter plot

[3.3] Line & scatter plot with style

```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'markers',
  name: 'Scatter',
  marker: {
    color: 'rgb(219, 64, 82)',
    size: 12
  }
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'Lines',
  line: {
    color: 'rgb(55, 128, 191)',
    width: 3
  }
};
```

```
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines+markers',
  name: 'Scatter + Lines',
  marker: {
    color: 'rgb(128, 0, 128)',
    size: 8
  },
  line: {
    color: 'rgb(128, 0, 128)',
    width: 1
  }
};
```

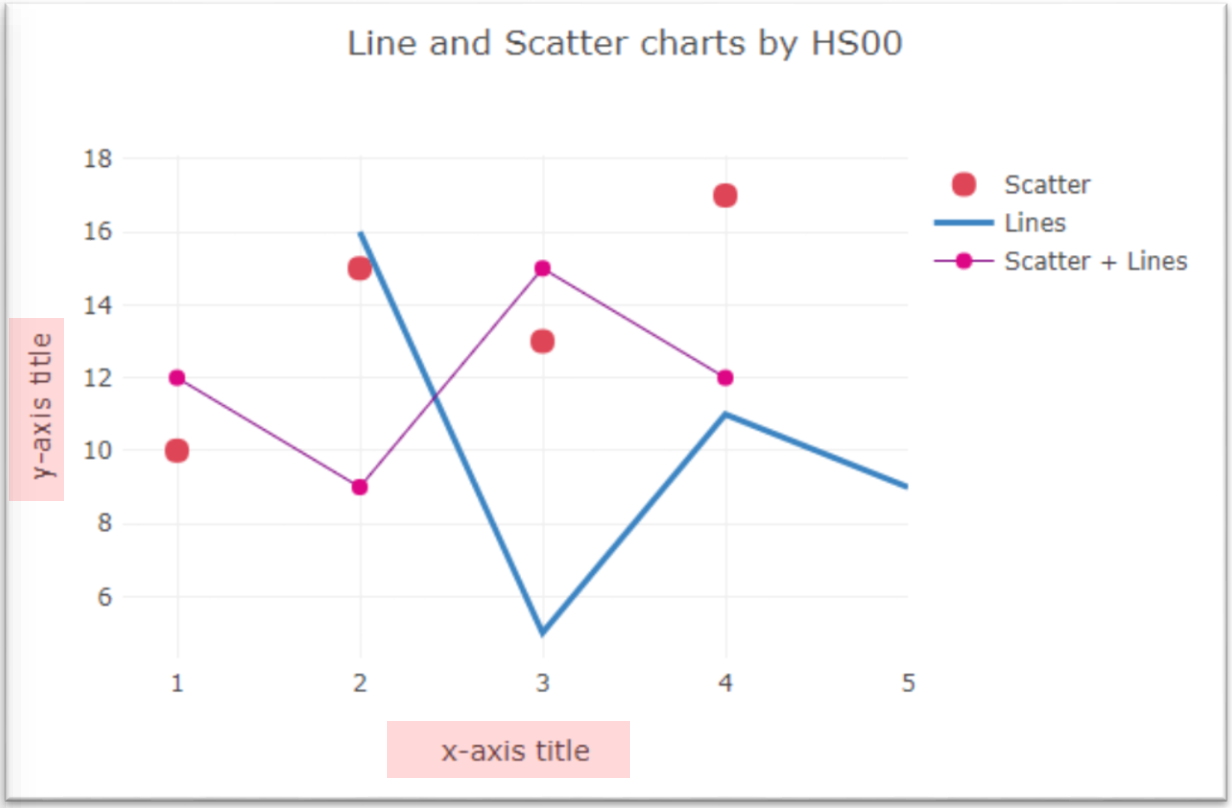


HSnn_Plot_Style.png

A5.2.6.5 plotly.js: Line & Scatter plot

[3.4] Line & scatter plot with axis titles

```
var layout = {
  title:'Line and Scatter Plot',
  width: 600, height: 450,
  margin: {
    l: 50,
    r: 50,
    b: 100,
    t: 100,
    pad: 4
  },
  xaxis: {
    title: 'x-axis title'
  },
  yaxis: {
    title: 'y-axis title'
  }
};
```



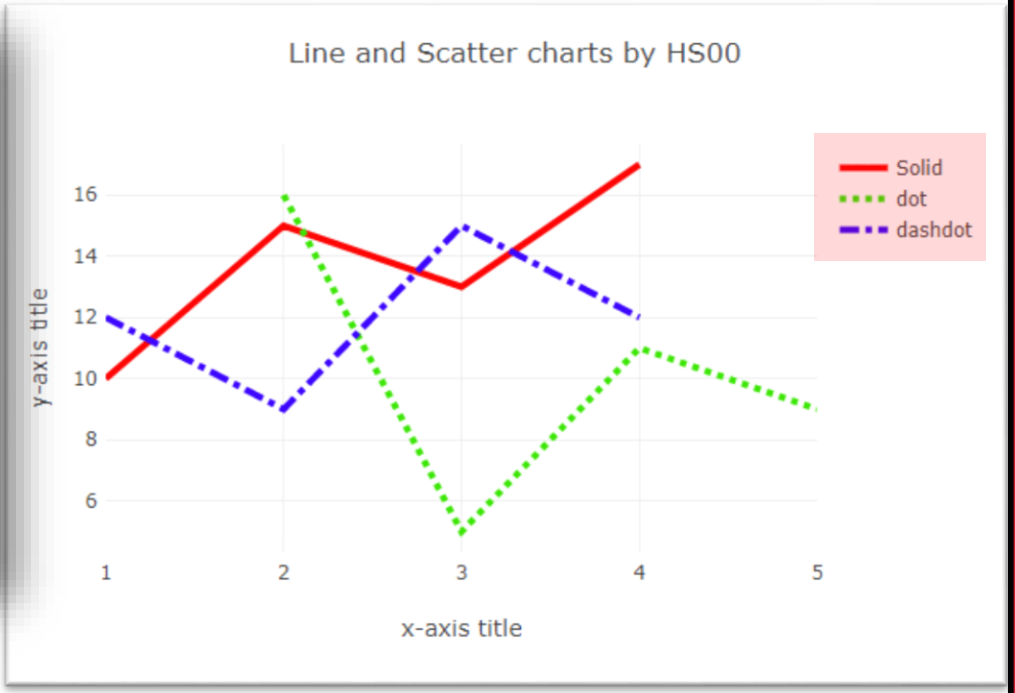
HSnn_Axis_Title.png

A5.2.6.6 plotly.js: Line & Scatter plot

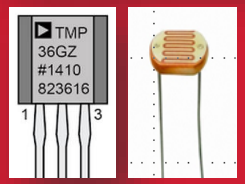
[3.5] Line & scatter plot with dash and dot

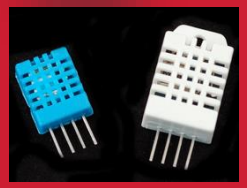
```
var trace1 = {
  x: [1, 2, 3, 4],
  y: [10, 15, 13, 17],
  mode: 'lines',
  name: 'Solid',
  line: {
    color: 'rgb(255, 0, 0)',
    dash: 'solid',
    width: 4
  }
};

var trace2 = {
  x: [2, 3, 4, 5],
  y: [16, 5, 11, 9],
  mode: 'lines',
  name: 'dot',
  line: {
    color: 'rgb(55, 228, 0)',
    dash: 'dot',
    width: 4
  }
};
```

```
var trace3 = {
  x: [1, 2, 3, 4],
  y: [12, 9, 15, 12],
  mode: 'lines',
  name: 'dashdot',
  line: {
    color: 'rgb(55, 0, 255',
    dash: 'dashdot',
    width: 4
  }
};
```

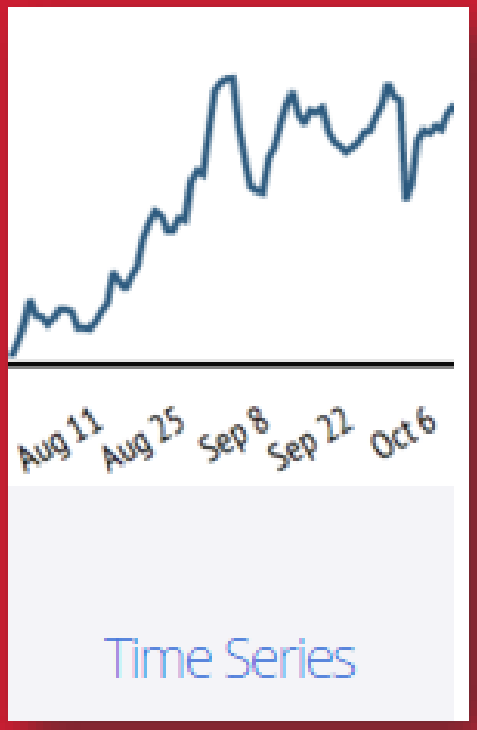


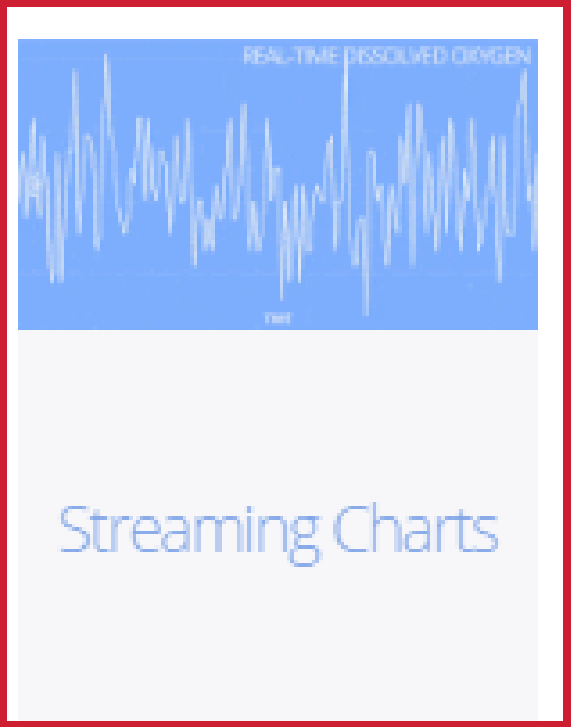
HSnn_Line_Dash_Dot.png



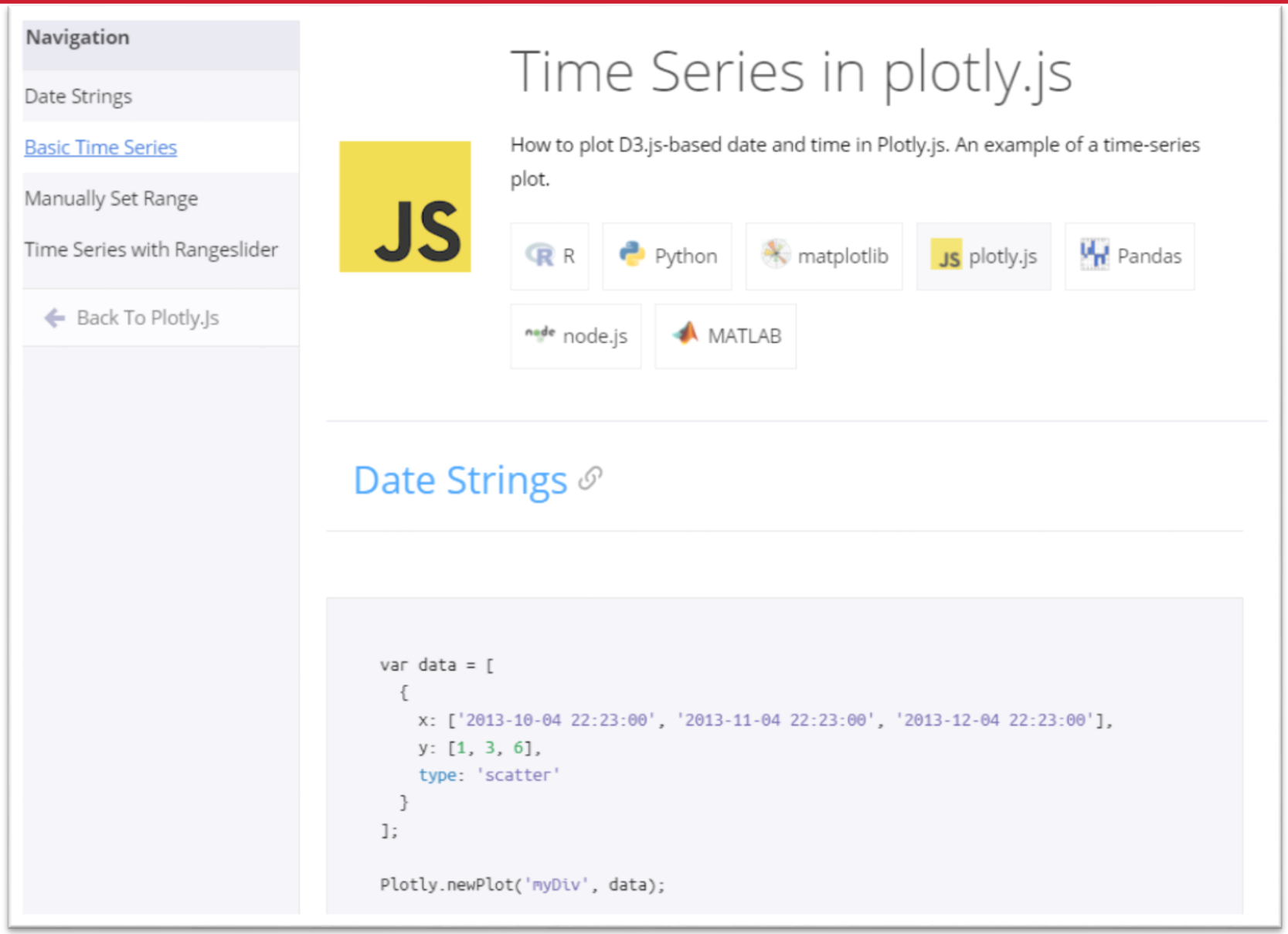


Data visualization using plotly.js





A5.3. Time series



Navigation

Date Strings

Basic Time Series

Manually Set Range

Time Series with Rangeslider

Back To Plotly.Js

Time Series in plotly.js

How to plot D3.js-based date and time in Plotly.js. An example of a time-series plot.

R Python matplotlib plotly.js Pandas

node.js MATLAB

Date Strings

```
var data = [
  {
    x: ['2013-10-04 22:23:00', '2013-11-04 22:23:00', '2013-12-04 22:23:00'],
    y: [1, 3, 6],
    type: 'scatter'
  }
];

Plotly.newPlot('myDiv', data);
```

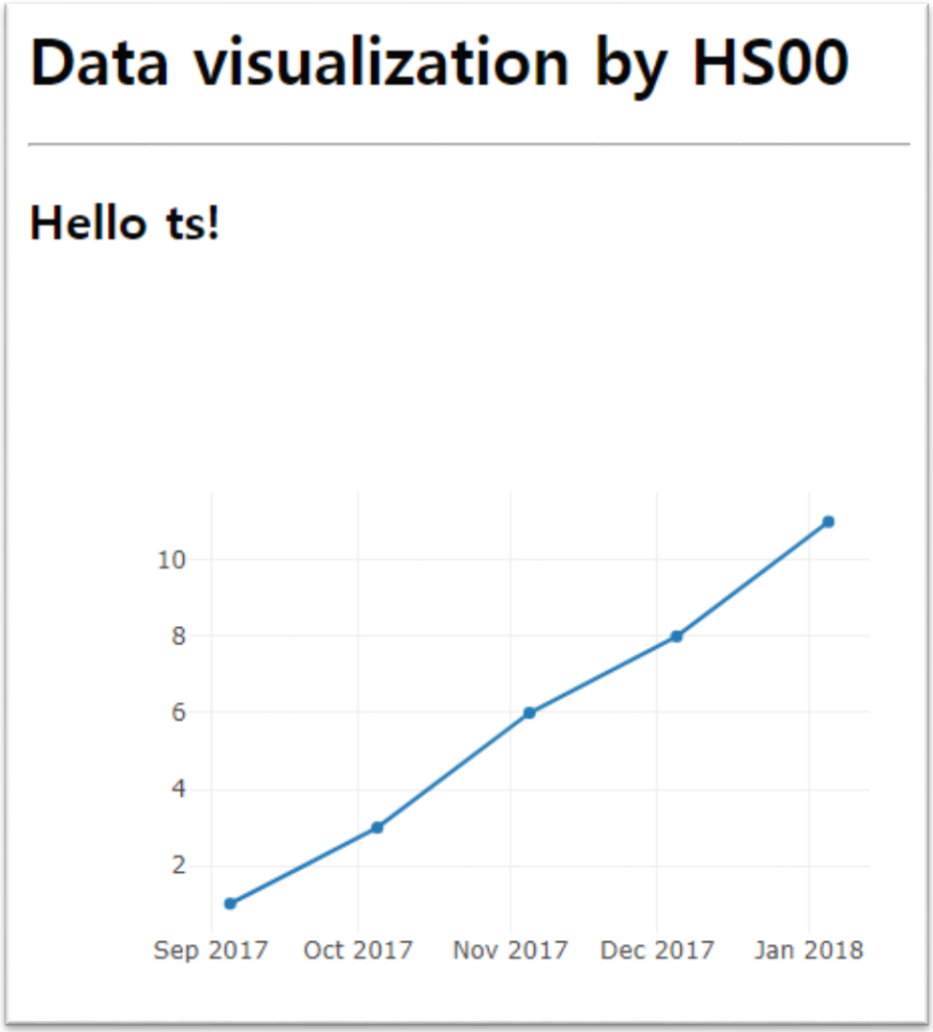
A5.3.1 plotly.js: Time series

[1] Time series : date strings

```
<!-- Plotly chart will be drawn inside this DIV -->
<div id="myDiv" style="width: 500px;height: 400px"></div>

<script>
    <!-- JAVASCRIPT CODE GOES HERE -->

    var data = [
    {
        x: ['2017-9-04 22:23:00',
        '2017-10-04 22:23:00',
        '2017-11-04 22:23:00',
        '2017-12-04 22:23:00'],
        y: [1, 3, 6, 8],
        type: 'scatter'
    }
    ];

    Plotly.newPlot('myDiv', data);

</script>
```

A5.3.2 plotly.js: Time series

Time series : date strings – result



A5.3.3.1 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

안전함 | https://raw.githubusercontent.com/plotly/datasets/master/finance-charts-a...

```
Date,AAPL.Open,AAPL.High,AAPL.Low,AAPL.Close,AAPL.Volume,AAPL.Adjusted,dn,mavg,up,direction
2015-02-17,127.489998,128.880005,126.919998,127.830002,63152400,122.905254,106.7410523,117.9276669,129.1142814,Increasing
2015-02-18,127.629997,128.779999,127.449997,128.720001,44891700,123.760965,107.842423,118.9403335,130.0382439,Increasing
2015-02-19,128.479996,129.029999,128.330002,128.449997,37362400,123.501363,108.8942449,119.8891668,130.8840887,Decreasing
2015-02-20,128.619995,129.5,128.050003,129.5,48948400,124.510914,109.7854494,120.7635001,131.7415509,Increasing
2015-02-23,130.020004,133,129.660004,133,70974100,127.876074,110.3725162,121.7201668,133.0678174,Increasing
2015-02-24,132.940002,133.600006,131.169998,132.169998,69228100,127.078049,111.0948689,122.6648335,134.2347981,Decreasing
2015-02-25,131.559998,131.600006,128.149994,128.789993,74711700,123.828261,113.2119183,123.6296667,134.0474151,Decreasing
2015-02-26,128.789993,130.869995,126.610001,130.419998,91287500,125.395469,114.1652991,124.2823333,134.3993674,Increasing
2015-02-27,130,130.570007,128.240005,128.460007,62014800,123.510987,114.9668484,124.8426669,134.7184854,Decreasing
2015-03-02,129.25,130.279999,128.300003,129.089996,48096700,124.116706,115.8770904,125.4036668,134.9302432,Decreasing
2015-03-03,128.960007,129.520004,128.089996,129.360001,37816300,124.376308,116.9535132,125.9551669,134.9568205,Increasing
2015-03-04,129.100006,129.559998,128.320007,128.539993,31666300,123.587892,118.0874253,126.4730002,134.8585751,Decreasing
2015-03-05,128.580002,128.75,125.760002,126.410004,56517100,121.539962,119.1048311,126.848667,134.5925029,Decreasing
2015-03-06,128.399994,129.369995,126.260002,126.599998,72842100,121.722637,120.190797,127.2288335,134.26687,Decreasing
2015-03-09,127.959999,129.570007,125.059998,127.139999,88528500,122.241834,121.6289771,127.631167,133.6333568,Decreasing
2015-03-10,126.410004,127.220001,123.800003,124.510002,68856600,119.71316,123.1164763,127.9235004,132.7305246,Decreasing
2015-03-11,124.75,124.769997,122.110001,122.239998,68939000,117.530609,123.592756,128.0033337,132.4139113,Decreasing
2015-03-12,122.309998,124.900002,121.629997,124.449997,48362700,119.655466,123.4894559,127.9813337,132.4732114,Increasing
2015-03-13,124.400002,125.400002,122.580002,123.589996,51827300,118.828598,123.045606,127.8490003,132.6523946,Decreasing
2015-03-16,123.879997,124.949997,122.870003,124.949997,35874300,120.136203,122.6967016,127.7283335,132.7599655,Increasing
2015-03-17,125.900002,127.32,125.650002,127.040001,51023100,122.145688,122.616033,127.6680002,132.7199674,Increasing
2015-03-18,127,129.160004,126.370003,128.470001,65270900,123.520597,122.6064498,127.652167,132.6978842,Increasing
2015-03-19,128.75,129.25,127.400002,127.5,45809500,122.587966,122.5939029,127.6245004,132.6550979,Decreasing
2015-03-20,128.25,128.399994,125.160004,125.900002,68695100,121.049608,122.4865925,127.4980004,132.5094083,Decreasing
2015-03-23,127.120003,127.849998,126.519997,127.209999,37709700,122.309137,122.6741703,127.2633335,131.8524968,Increasing
2015-03-24,127.230003,128.039993,126.559998,126.690002,32842300,121.809174,123.0410183,127.0025001,130.9639818,Decreasing
2015-03-25,126.540001,126.82,123.379997,123.379997,51655200,118.626689,122.8276392,126.7531667,130.6786943,Decreasing
2015-03-26,122.760002,124.879997,122.599998,124.239998,47572900,119.453558,122.5538523,126.4835001,130.4131478,Increasing
2015-03-27,124.57,124.699997,122.910004,123.25,39546200,118.5017,122.2826504,126.2099998,130.1373491,Decreasing
2015-03-30,124.050003,126.400002,124,126.370003,47099700,121.501502,122.346906,126.0283332,129.7097604,Increasing
2015-03-31,126.089996,126.489998,124.360001,124.43,42090600,119.63624,122.395242,125.8334998,129.2717577,Decreasing
2015-04-01,124.82,125.120003,123.099998,124.25,40621400,119.463174,122.3761274,125.6009999,128.8258723,Decreasing
```

A5.3.3.2 plotly.js: Time series

[2] Time series : financial data strings – AAPL stock price

```
Plotly.d3.csv("https://raw.githubusercontent.com/plotly/datasets/master/
    finance-charts-apple.csv", function(err, rows){

    function unpack(rows, key) {
        return rows.map(function(row) { return row[key]; });
    }


    var trace1 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL High',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.High'),
        line: {color: '#17BECF'}
    }

    var trace2 = {
        type: "scatter",
        mode: "lines",
        name: 'AAPL Low',
        x: unpack(rows, 'Date'),
        y: unpack(rows, 'AAPL.Low'),
        line: {color: '#7F7F7F'}
    }

    var data = [trace1,trace2];
```

A5.3.3.3 plotly.js: Time series

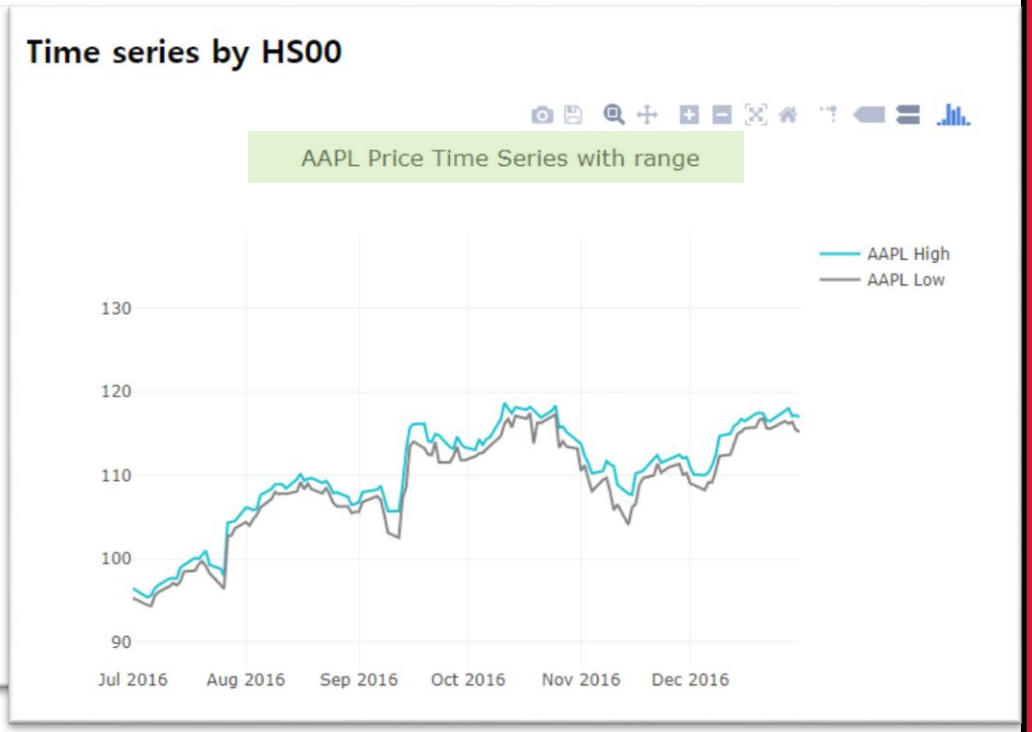
[2] Time series : financial data strings – AAPL stock price

```
var data = [trace1,trace2];

var layout = {
    title: 'AAPL Price Time Series',
};

Plotly.newPlot('myDiv', data, layout);
```

A5.3.3.4 plotly.js: Time series

[2] Time series : financial data strings – set range

```
var data = [trace1,trace2];

var layout = {
    title: 'AAPL Price Time Series with range',
    xaxis: {
        range: ['2016-07-01', '2016-12-31'],
        type: 'date'
    },
    yaxis: {
        autorange: true,
        range: [86.8700008333, 138.870004167],
        type: 'linear'
    }
};

Plotly.newPlot('myDiv', data, layout);
```



날짜와 주가의 범위를 지정

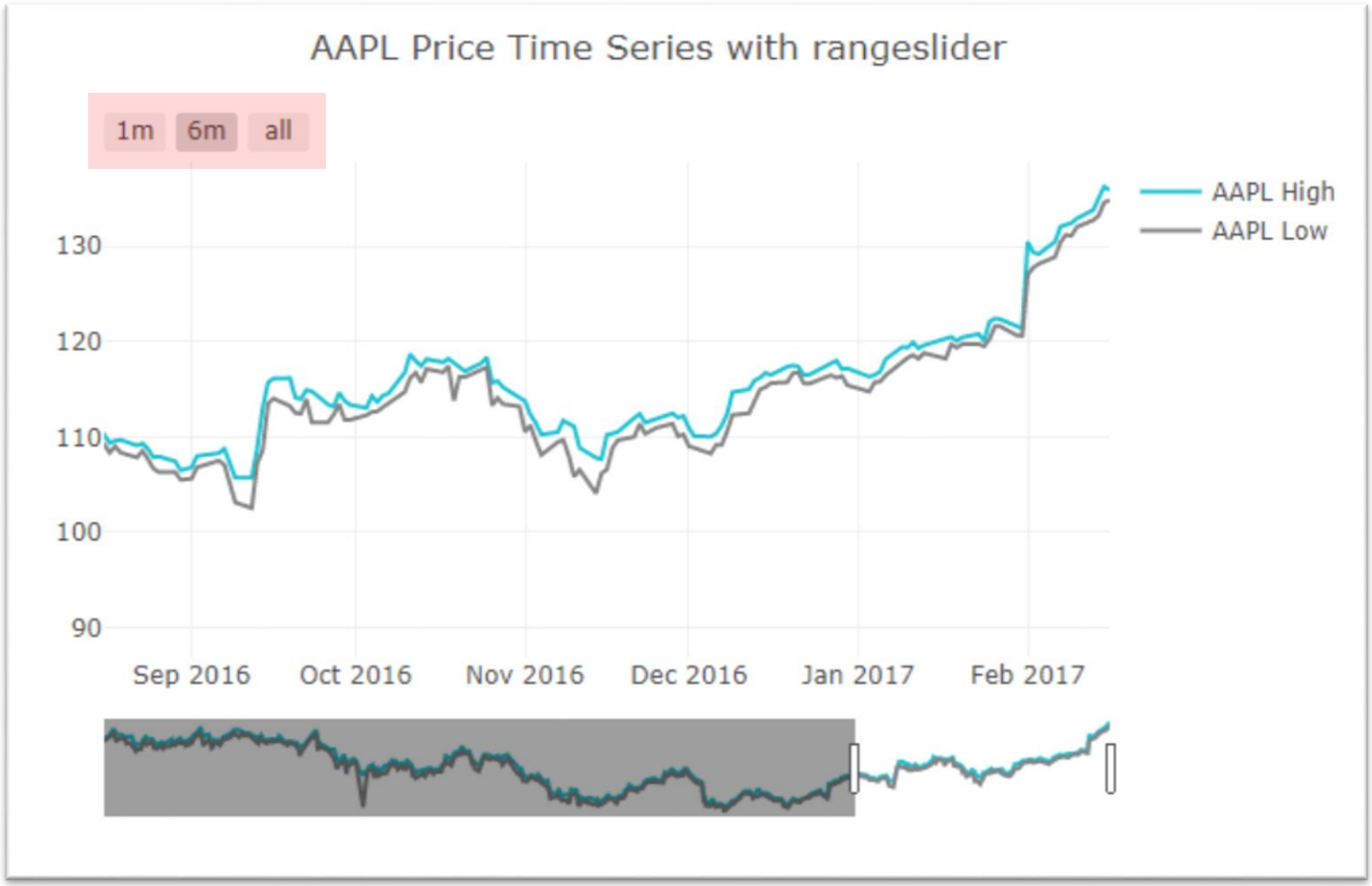
A5.3.3.5 plotly.js: Time series

[2] Time series : financial data strings – Range slider

```
var layout = {
    title: 'AAPL Price Time Series with rangeslider',
    xaxis: {
        autorange: true,
        range: ['2015-02-17', '2017-02-16'],
        rangeselector: {buttons: [
            {
                count: 1,
                label: '1m',
                step: 'month',
                stepmode: 'backward'
            },
            {
                count: 6,
                label: '6m',
                step: 'month',
                stepmode: 'backward'
            },
            {step: 'all'}
            ]},
            rangeslider: {range: ['2015-02-17', '2017-02-16']},
            type: 'date'
        },
        yaxis: {
            autorange: true,
            range: [86.8700008333, 138.870004167],
            type: 'linear'
        }
    };
```

A5.3.3.6 plotly.js: Time series

[2] Time series : financial data strings – Range slider

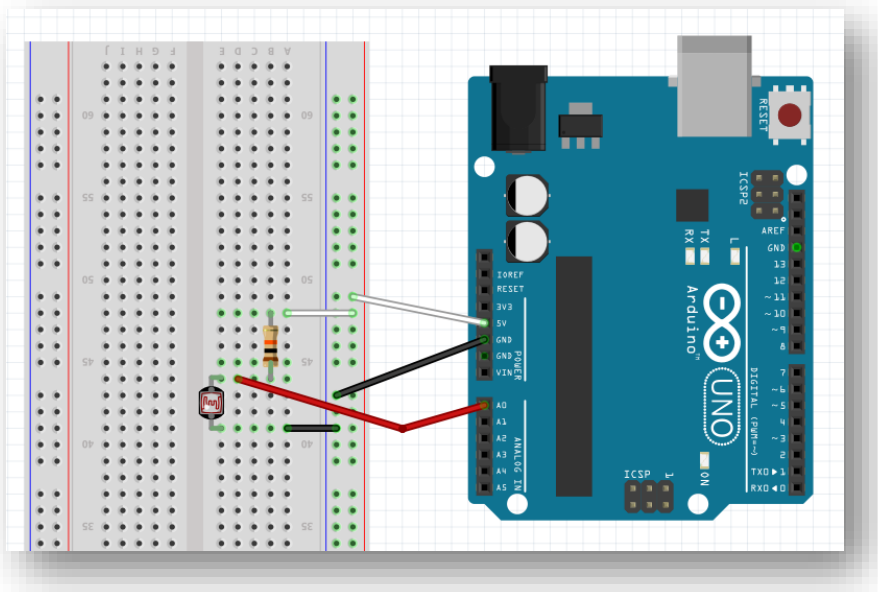


A5.3.4.1 plotly.js: Sensor time series

[3] Time series : my lux data

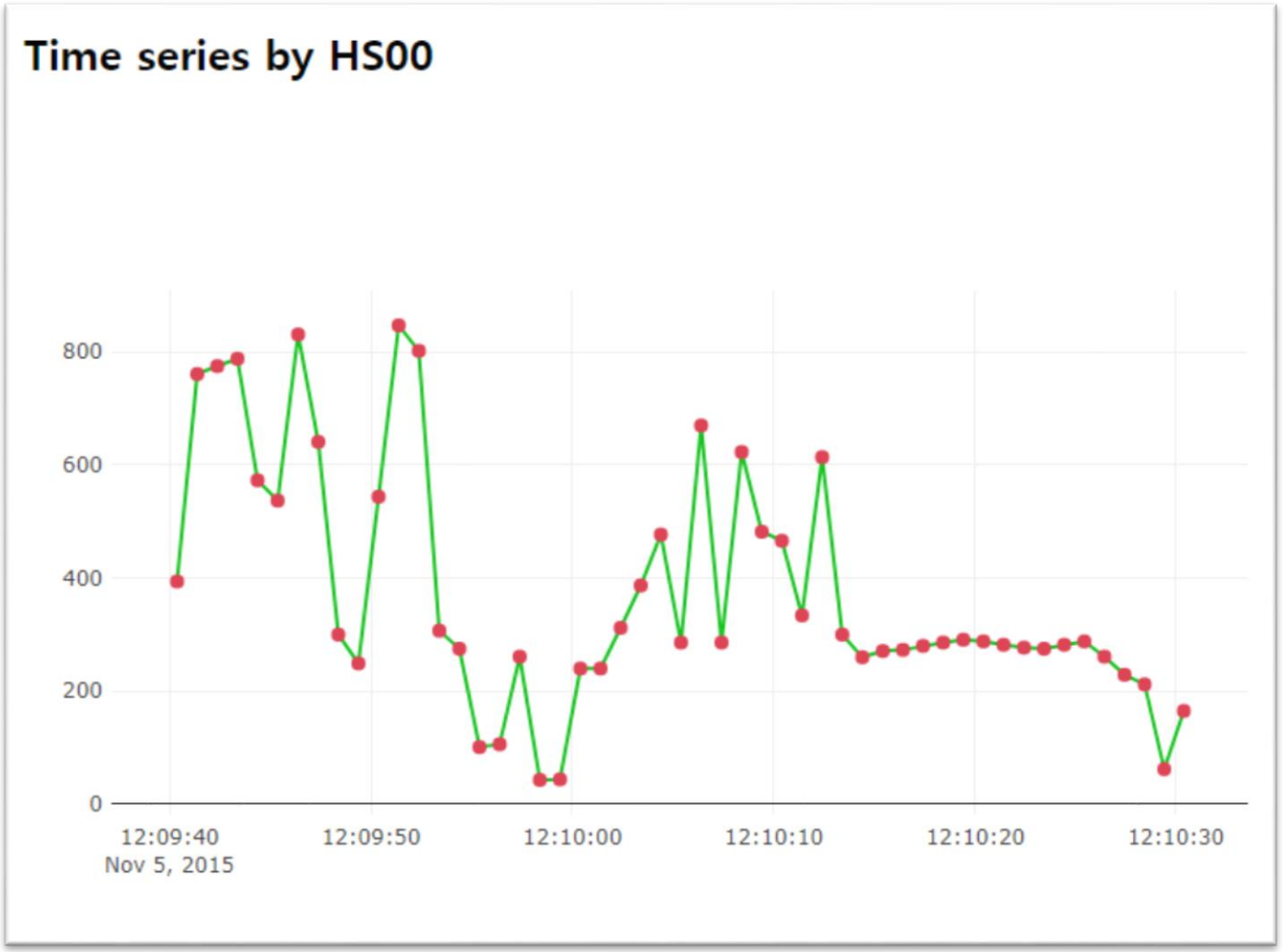
```
'2015-11-05 12:09:41.382',
'2015-11-05 12:09:42.380',
'2015-11-05 12:09:43.378',
'2015-11-05 12:09:44.377',
'2015-11-05 12:09:45.375',
'2015-11-05 12:09:46.389',
'2015-11-05 12:09:47.388',
'2015-11-05 12:09:48.386',
'2015-11-05 12:09:49.384',
'2015-11-05 12:09:50.383',
'2015-11-05 12:09:51.381',
'2015-11-05 12:09:52.380',
'2015-11-05 12:09:53.394',
'2015-11-05 12:09:54.392',
'2015-11-05 12:09:55.391',
'2015-11-05 12:09:56.389',
'2015-11-05 12:09:57.387',
'2015-11-05 12:09:58.386',
'2015-11-05 12:09:59.384',
'2015-11-05 12:10:00.398',
'2015-11-05 12:10:01.397',
```

Data :
date,value



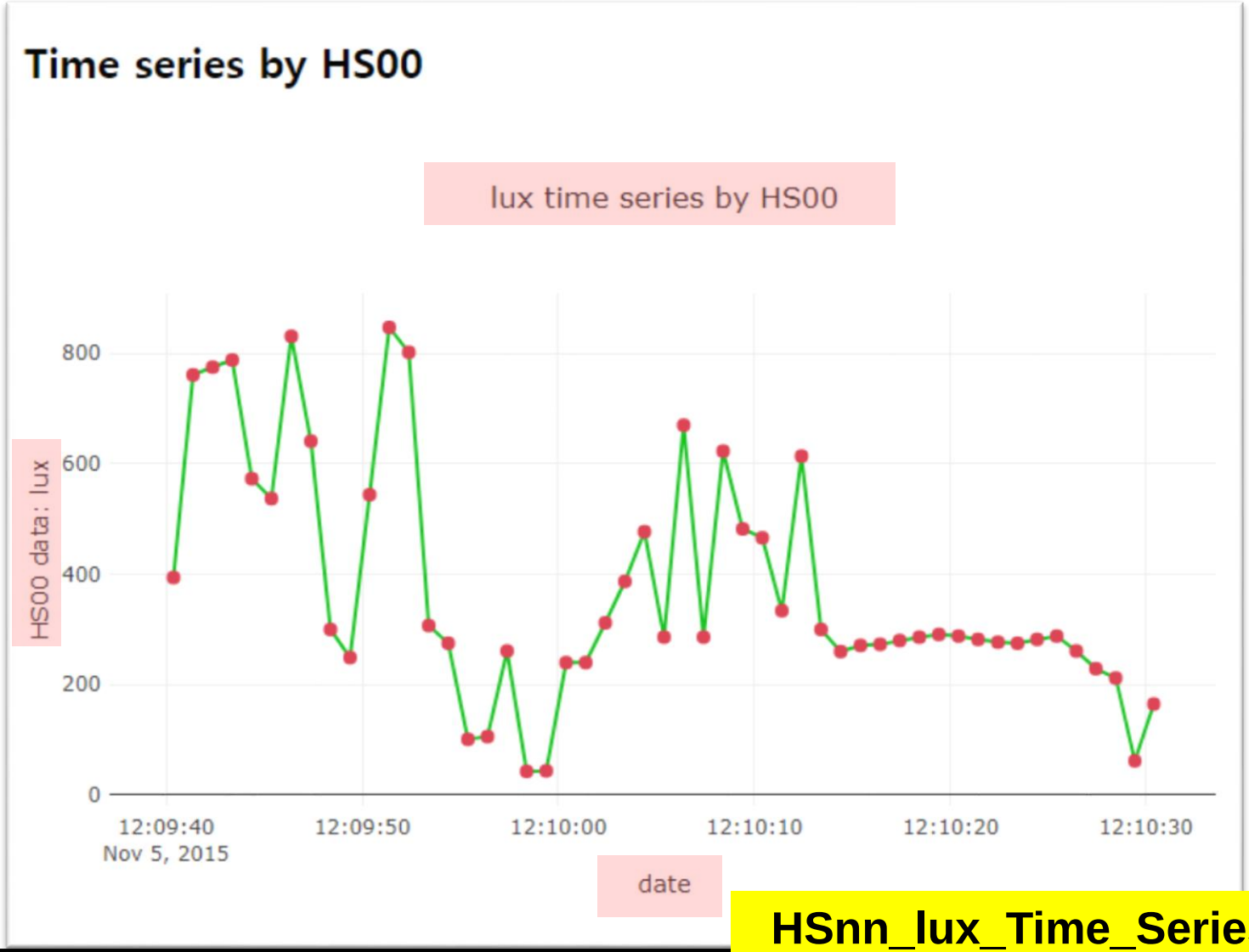
A5.3.4.2 plotly.js: Time series

[3] Time series : my lux data → DV_ts03_sensor_chart.html



A5.3.4.3 plotly.js: Time series

[3] Time series : my lux data – [DIY] → Set title and axis title



HSnn_lux_Time_Series.png

Project: Time series with Rangeslider

[Project-DIY] HSnn_lux_Rangelslider.html

HSnn_lux_Rangelslider.png

[Practice]

- [wk09]
  - Charts by plotly
  - Complete your plotly chart project
  - Upload file name : HSnn_Rpt07.zip

[wk09] Practice-07 HSnn_Rpt07.zip

- [Target of this week]
  - Complete your charts
  - Save your outcomes and compress them.

제출파일명 : **HSnn_Rpt07.zip**

- 압축할 파일들

① HSnn_Chart_Layout.png

② HSnn_Plot_Style.png

③ HSnn_Axis_Title.png

④ HSnn_Line_Dash_Dot.png

⑤ HSnn_lux_Time_Series.png

⑥ HSnn_lux_Rangeslider.png

Email : chaos21c@gmail.com

[제목 : id, 이름 (수정)]

Lecture materials

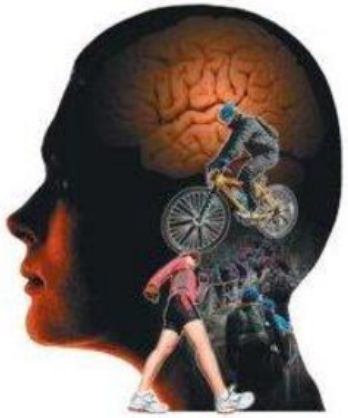
References & good sites

- ✓ http://www.nodejs.org/ko Node.js
- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub
- ✓ http://www.google.com Googling

Github.com/Redwoods

GitHub, Inc. [US] | https://github.com Redwoods (Redwoods Yi) · GitHub

Features Business Explore Marketplace Pricing Search GitHub Sign in or Sign up



Redwoods Yi

Redwoods

Block or report user

GimHae, Republic of Korea

Overview Repositories 5 Stars 2 Followers 0 Following 0

Pinned repositories

dht22-iot-project

Iot project to monitor data streaming from DHT22 wired at Arduino.

HTML

Lec

All lectures by Redwoods in Inje University

arduino-nodejs-plotly-streaming

This repo introduces a simple and efficient way to plot the streaming data from Arduino with Easy Pulse ppg sensor or DHT11 sensor.

HTML

hw-coding

Resource for lecture of Hardware Programming (2017, Inje university)

Arduino

Github.com/Redwoods/healthcare-signal-iot

Redwoods / Lec

Unwatch 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

All lectures by Redwoods in Inje University

Add topics

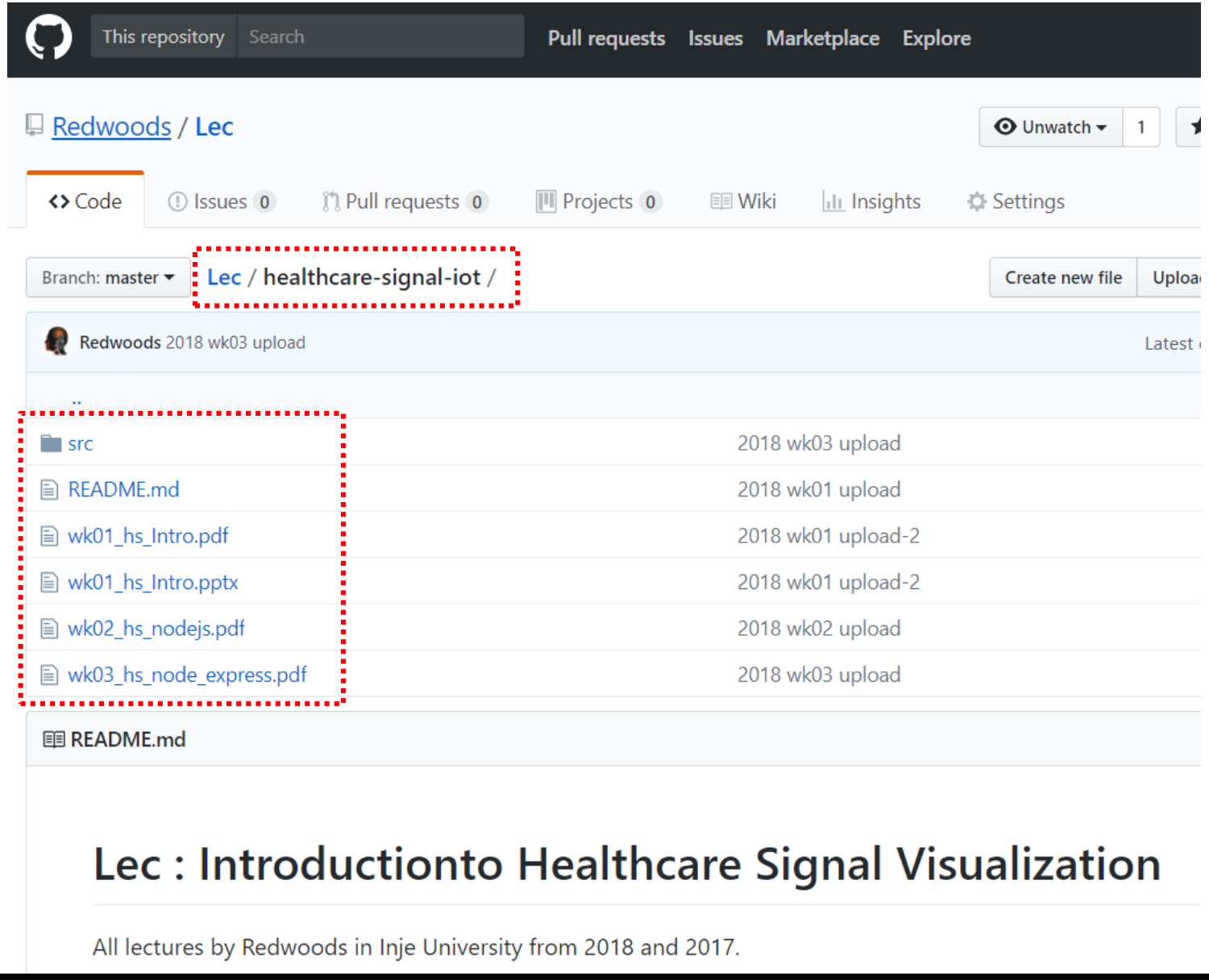
81 commits | 1 branch | 0 releases

Branch: master | New pull request | Create new file | Upload files

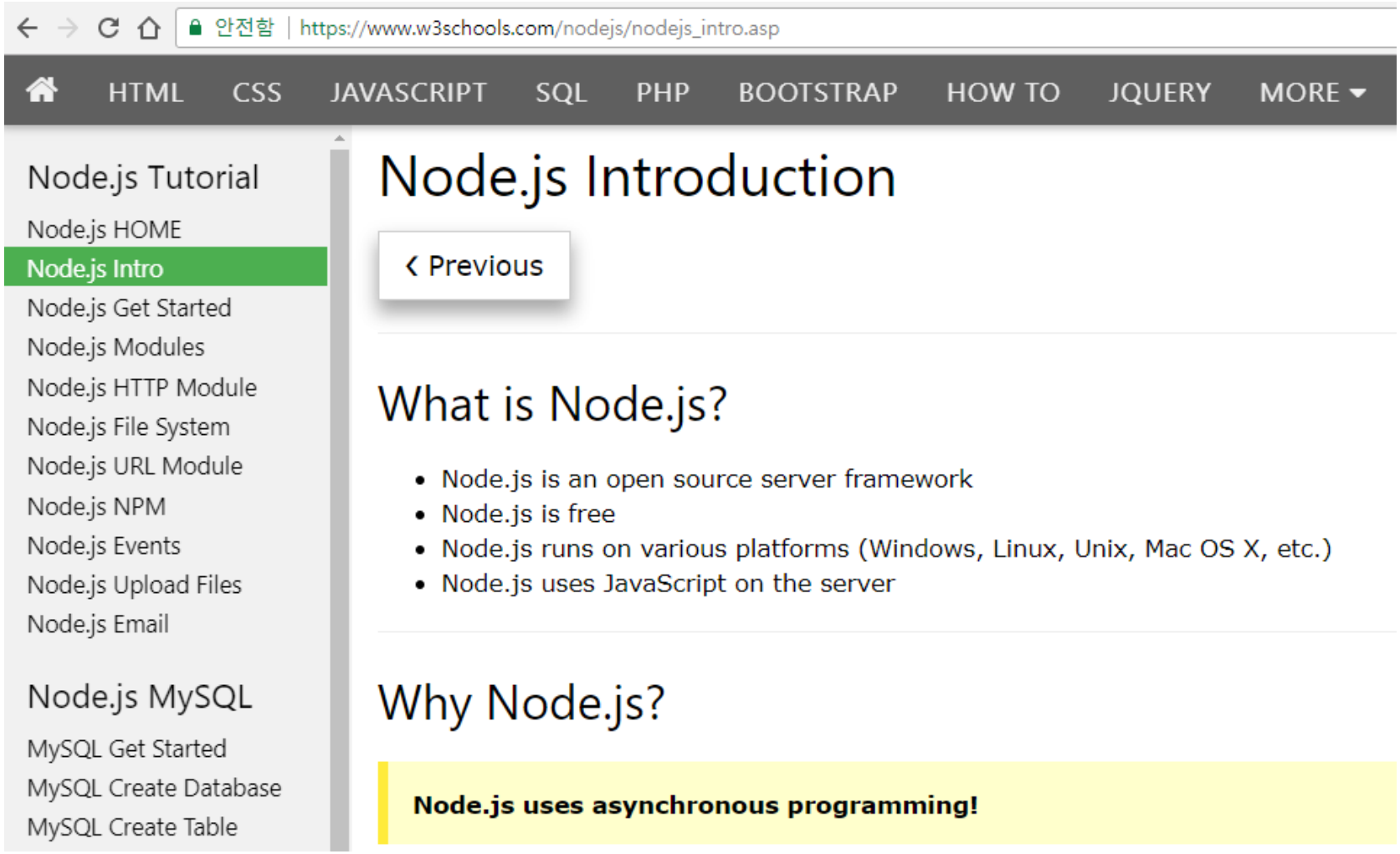
Redwoods 2018 wk01 upload

advanced-Arduino-iot	wk16 exam upload
ev3	wk16 final exam. answers
healthcare-signal-iot	2018 wk01 upload
html5-basic	2018 wk01 upload
html5-mobile-simulation	wk15 lec upload
Lec.Rproj	2018 wk01 upload
README.md	wk03 upload and fix links

Github.com/Redwoods/healthcare-signal-iot

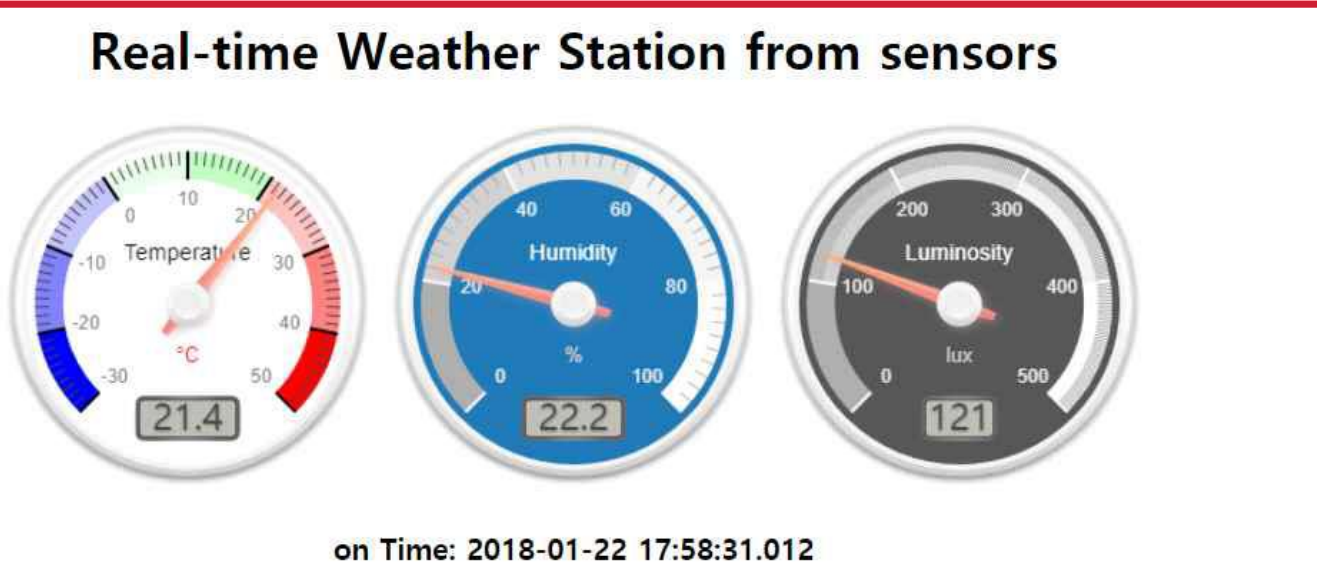


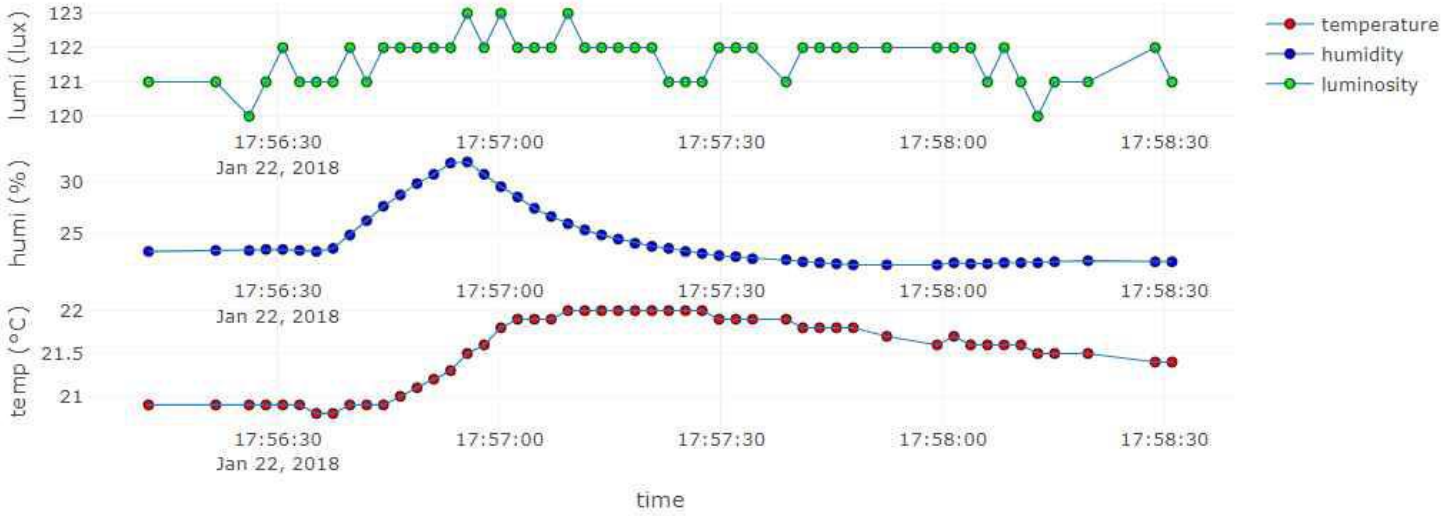
1.0 What is node.js?



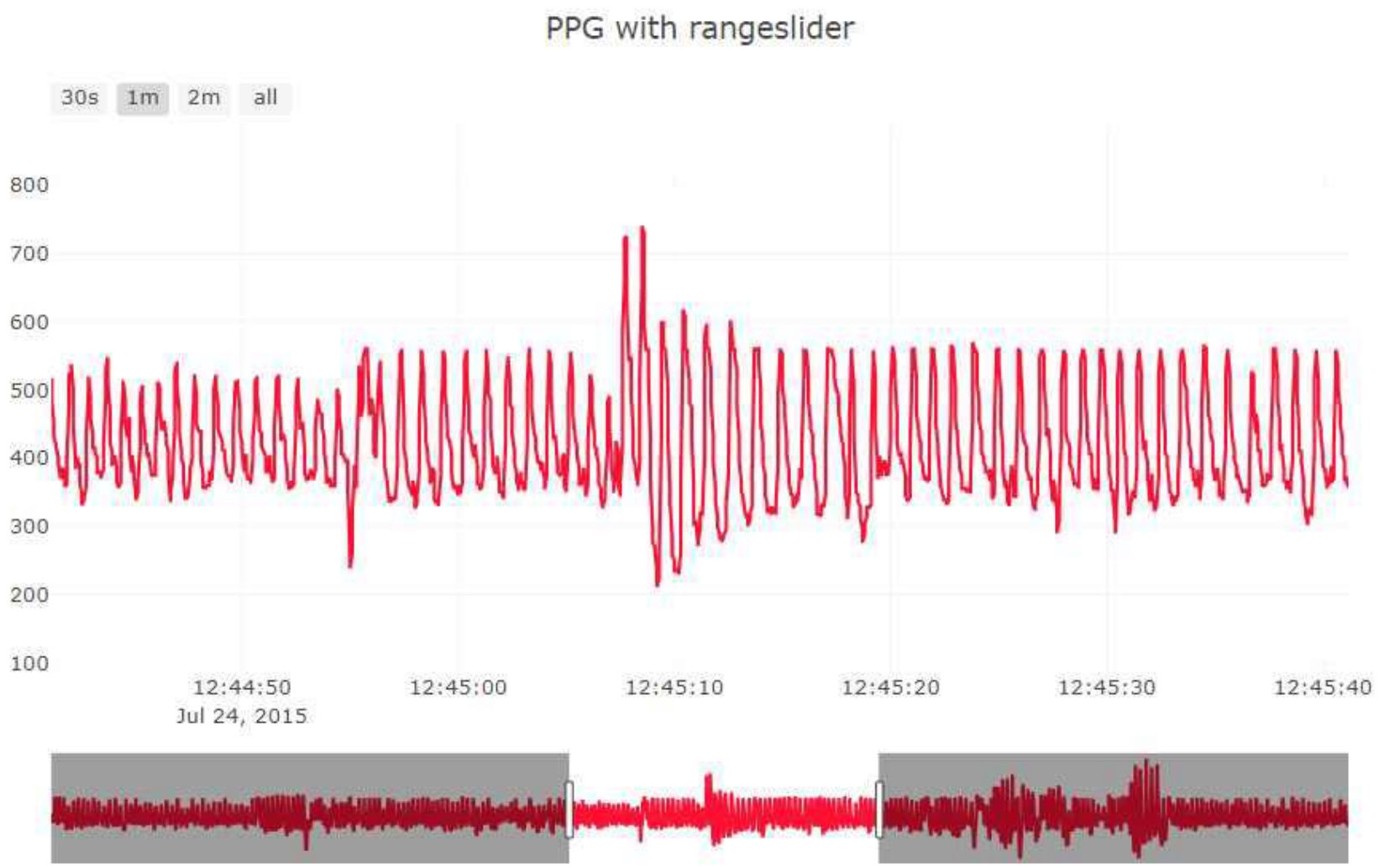
https://www.w3schools.com/nodejs/nodejs_intro.asp

Target of this class





Project of this class



주교재