資料更新與呈現

- 喜愛水果清單
  - 切換不同人
  - 更新清單
    - 新增
    - 更新
    - 刪除
  - 動畫效果

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
       <h3 class="Header">Favorite fruit list</h3>
       <div class="general-update-container">
           <svg width="150" height="200"></svg>
       </div>
       <div class="controls">
        <button class="john" data-name="john">John</button>
        <button class="marry" data-name="marry">Marry</button>
        <button class="ryan" data-name="ryan">Ryan</button>
       </div>
        <script src="//unpkg.com/d3"></script>
        <script src="main.js"></script>
    </body>
</html>
```

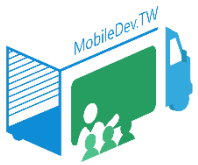
style.css

```css
body{
    font-weight: bold;
    margin: 2rem;
}

h3{
    font-weight: 100;
    color: #555;
    margin: 0.4rem 0;
}

svg{
    border: 1px solid #ccc;
}
```

```css
button{
    outline: none;
    color: #555;
    border-color: #eee;
    background: #fefefe;
    padding: 3px 0px;
    width: 48px;
    text-align: center;
    transition: all 200ms;
}

button:hover{
    background: lightgoldenrodyellow;
}

button:focus{
    background: lightpink;
}
```

main.js

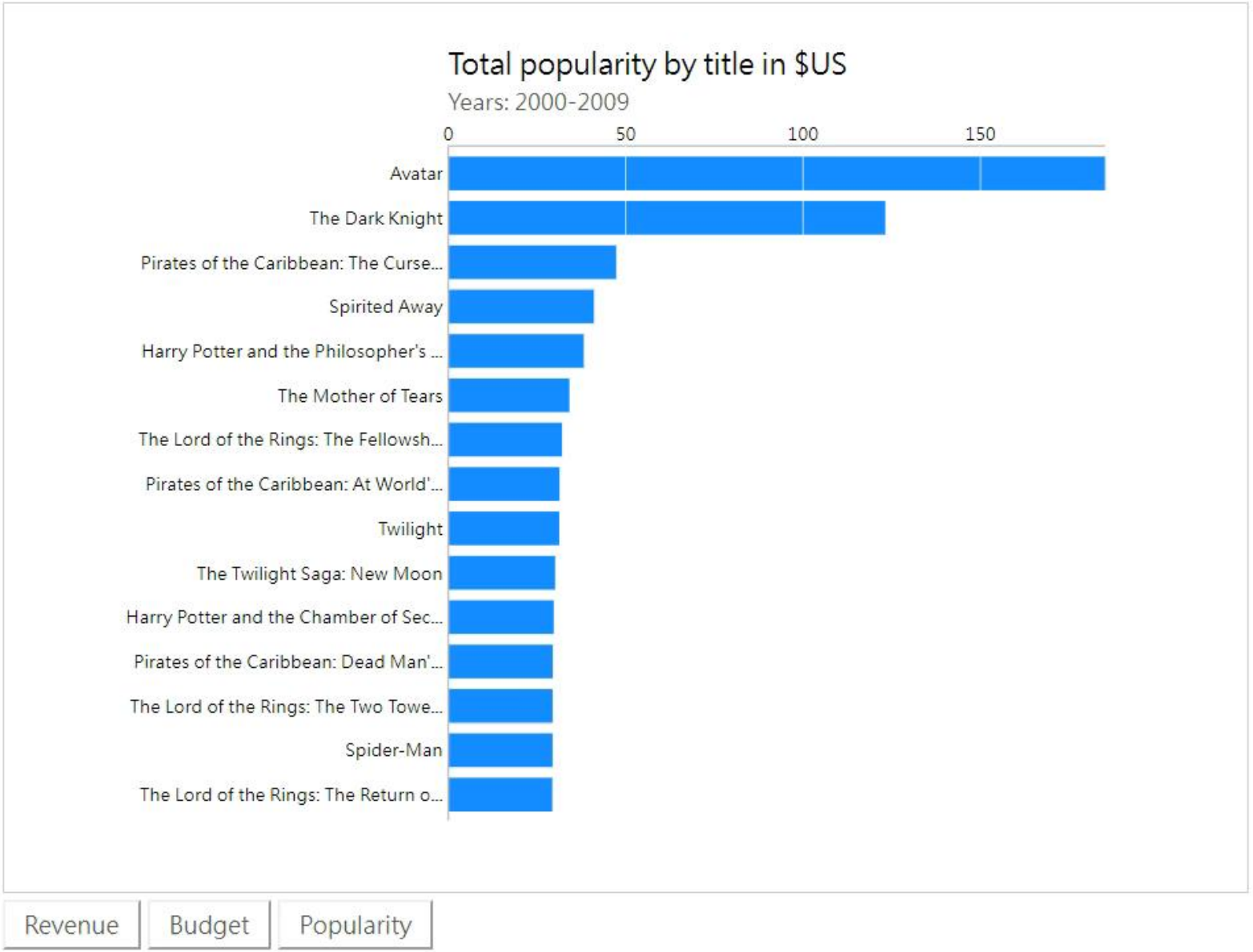
```
const friends = {
    john:['Apple','Orange','Lemon'],
    marry:['Apple','Orange'],
    ryan:['Apple','Cherry','Peach','Orange']
};

const thisSVG = d3.select('svg');
d3.selectAll('button').on('click',click);

function click(){
    const thisFruitList = friends[this.dataset.name];
    update(thisFruitList);
}
```



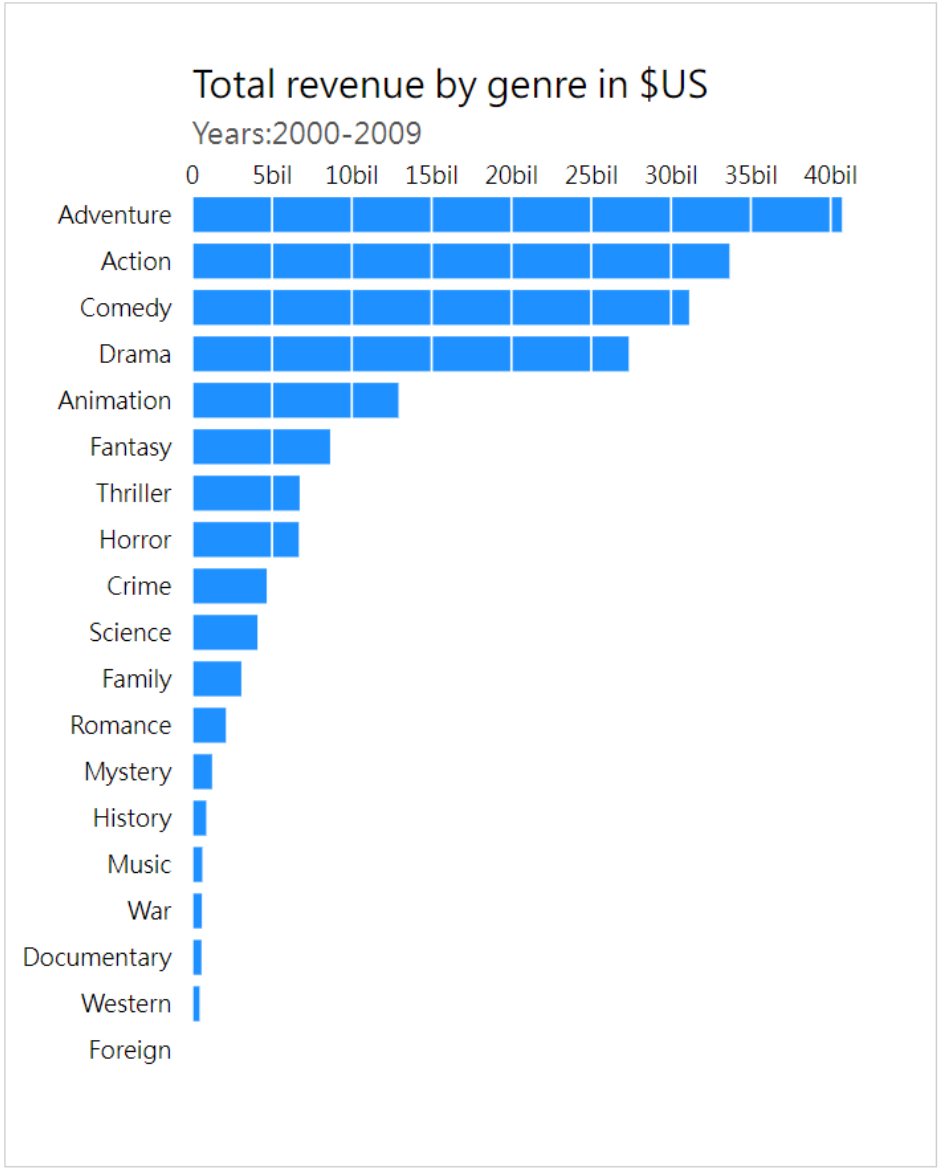
update function

```
function update(data){
    thisSVG.selectAll('text')
    .data(data, d=>d)
    .join(
        enter => {
            enter.append('text').text(d=>d)
                .attr('x',-100).attr('y',(d,i)=>50+i*30)
                .style('fill','green')
                .transition().attr('x',30)
        },
        update => {
            update.transition()
                .style('fill','red').attr('y',(d,i)=>50+i*30)
        },
        exit => {
            exit.transition()
                .attr('x',150)
                .style('fill','yellow').remove()
        }
    );
}
```

套用資料更新於Bar Chart



複製先前長條圖的專案開始修改



index.html

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <div class="bar-chart-container"></div>
        <div class="controls">
            <button data-name="revenue">Revenue</button>
            <button data-name="budget">Budget</button>
            <button data-name="popularity">Popularity</button>
        </div>
        <script src="//unpkg.com/d3"></script>
        <script src="main.js"></script>
    </body>
</html>
```

三種想要呈現的資料

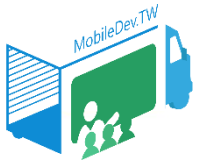
- Top 15 收益
  - 先改成這個版本
- Top 15 預算
- Top 15 熱門

main.js

```
function ready(movies){
    const movieClean = filterData(movies);
    //Get Top 15 revenue movies
    const revenueData = chooseData("revenue",movieClean);
    setupCanvas(revenueData, movieClean);
}

d3.csv('data/movies.csv',type).then(
    res => {
        ready(res);
    }
);

function chooseData(metric, movieClean){
    const thisData = movieClean.sort((a,b)=>b[metric]-a[metric]).filter((d,i)=>i<15);
    return thisData;
}
```

setupCanvas Function

```
function setupCanvas(barChartData, movieClean){

    //一開始預設指標是revenue
    let metric = 'revenue';

    function click(){
        metric = this.dataset.name;
        const thisData = chooseData(metric, movieClean);
        update(thisData);
    }

    d3.selectAll('button').on('click',click);

    function update(data){
    }
```



setupCanvas Function

```
const svg_width = 700;
const svg_height = 500;
const barchart_margin = {top:80,right:80,bottom:40,left:250};
const barchart_width = svg_width - (barchart_margin.left + barchart_margin.right);
const barchart_height = svg_height - (barchart_margin.top + barchart_margin.bottom);

const this_svg = d3.select('.bar-chart-container').append('svg')
  .attr('width', svg_width).attr('height',svg_height)
  .append('g')
  .attr('transform',`translate(${barchart_margin.left},${barchart_margin.top})`);
```

setupCanvas Function

```
//scale
//V1.d3.extent find the max & min in revenue
const xExtent = d3.extent(barChartData, d=>d.revenue);
const xScale_v1 = d3.scaleLinear().domain(xExtent).range([0,barchart_width]);
//V2.0 ~ max
let xMax = d3.max(barChartData, d=>d.revenue);
const xScale_v2 = d3.scaleLinear().domain([0, xMax]).range([0,barchart_width]);
//V3.Short writing for v2
let xScale_v3 = d3.scaleLinear([0,xMax],[0, barchart_width]);

//垂直空間的分配 - 平均分布給各種類
// const yScale = d3.scaleBand().domain(barChartData.map(d=>d.genre))
//                  .rangeRound([0, barchart_height])
//                  .paddingInner(0.25);

//垂直空間的分配 - 平均分布給Top 15
let yScale = d3.scaleBand().domain(barChartData.map(d=>d.title))
               .rangeRound([0, barchart_height])
               .paddingInner(0.25);
```

setupCanvas Function

```
//Draw bars
// const bars = this_svg.selectAll('.bar').data(barChartData).enter()
//                      .append('rect')
//                        .attr('class','bar')
//                        .attr('x',0).attr('y',d=>yScale(d.genre))
//                        .attr('width',d=>xScale_v3(d.revenue))
//                        .attr('height',yScale.bandwidth())
//                        .style('fill','dodgerblue');
const bars = this_svg.append('g').attr('class','bars');
```

setupCanvas Function

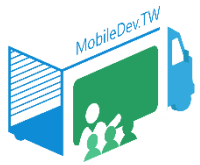
```
//Draw header
let header = this_svg.append('g').attr('class','bar-header')
                .attr('transform',`translate(0,${-barchart_margin.top/2})`)
                .append('text');
//header.append('tspan').text('Total revenue by genre in $US');
header.append('tspan').text('Top 15 XXX movies');
header.append('tspan').text('Years:2000-2009')
      .attr('x',0).attr('y',20).style('font-size','0.8em').style('fill','#555');
```

setupCanvas Function

```
    //tickSizeInner : the length of the tick lines
    //tickSizeOuter : the length of the square ends of the domain path
    let xAxis = d3.axisTop(xScale_v3).ticks(5).tickFormat(formatTicks)
                  .tickSizeInner(-barchart_height).tickSizeOuter(0);
    //this_svg.append('g').attr('class','x axis').call(xAxis);
    let xAxisDraw = this_svg.append('g').attr('class','x axis');

    //tickSize : set tickSizeInner & Outer
    let yAxis = d3.axisLeft(yScale).tickSize(0);
    //const yAxisDraw = this_svg.append('g').attr('class','y axis').call(yAxis);
    let yAxisDraw = this_svg.append('g').attr('class','y axis');
    yAxisDraw.selectAll('text').attr('dx','-0.6em');
    update(barChartData);
}
```

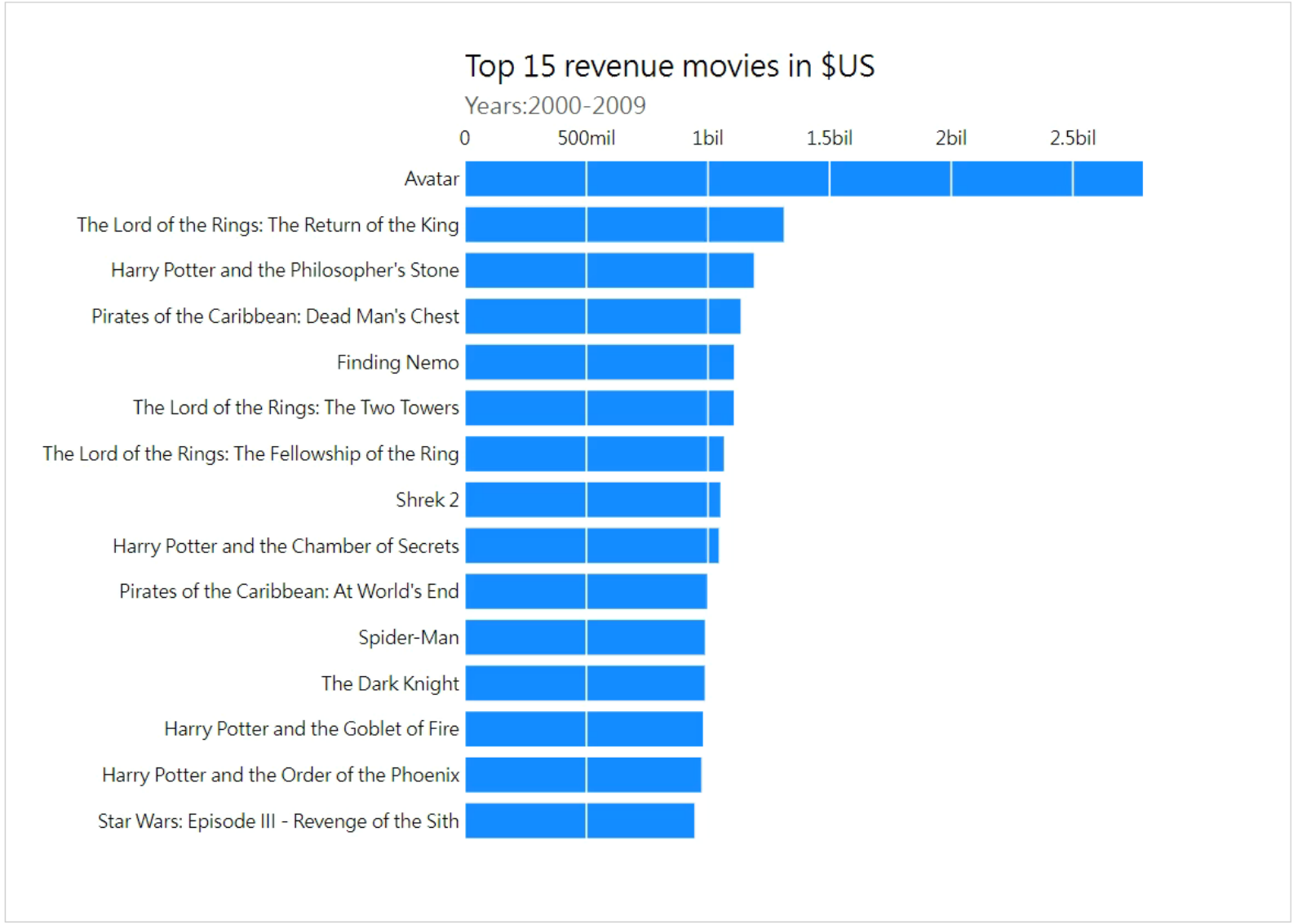
setupCanvas Function - update

```
function update(data){
        console.log(data);
        //Update Scale
        xMax = d3.max(data, d=>d[metric]);
        xScale_v3 = d3.scaleLinear([0, xMax],[0,barchart_width]);

        yScale = d3.scaleBand().domain(data.map(d=>d.title))
        .rangeRound([0, barchart_height])
        .paddingInner(0.25);

        //Transition Settings
        const defaultDelay = 1000
        const transitionDelay = d3.transition().duration(defaultDelay);

        //Update axis
        xAxisDraw.transition(transitionDelay).call(xAxis.scale(xScale_v3));
        yAxisDraw.transition(transitionDelay).call(yAxis.scale(yScale));

        //Update Header
        header.select('tspan').text(`Top 15 ${metric} movies ${metric==='popularity'?'':'in $US'}`);
```




```
        //Update Bar
        bars.selectAll('.bar').data(data, d=>d.title).join(
            enter => {
                enter.append('rect').attr('class','bar')
                    .attr('x',0).attr('y',d=>yScale(d.title))
                    .attr('height',yScale.bandwidth())
                    .style('fill','lightcyan')
                    .transition(transitionDelay)
                    .delay((d,i)=>i*20)
                    .attr('width',d=>xScale_v3(d[metric]))
                    .style('fill', 'dodgerblue')
            },
            update => {
                update.transition(transitionDelay)
                    .delay((d,i)=>i*20)
                    .attr('y',d=>yScale(d.title))
                    .attr('width',d=>xScale_v3(d[metric]))
            },
            exit => {
                exit.transition().duration(defaultDelay/2)
                    .style('fill-opacity',0)
                    .remove()
            }
        );
    }
```

測試結果


Top 15 revenue movies in $US
Years:2000-2009
0
500mil
1bil
1.5bil
2bil
2.5bil
Avatar
The Lord of the Rings: The Return of the King
Harry Potter and the Philosopher's Stone
Pirates of the Caribbean: Dead Man's Chest
Finding Nemo
The Lord of the Rings: The Two Towers
The Lord of the Rings: The Fellowship of the Ring
Shrek 2
Harry Potter and the Chamber of Secrets
Pirates of the Caribbean: At World's End
Spider-Man
The Dark Knight
Harry Potter and the Goblet of Fire
Harry Potter and the Order of the Phoenix
Star Wars: Episode III - Revenge of the Sith
Revenue
Budget
Popularity

項目名稱太長的解決方案

- 把所有d.title跟空間有關的部分，套上這個方法

  cutText(d.title)

```
function cutText(string){
    return string.length<35 ? string : string.substring(0,35)+"...";
}
```