Reading Files

Asynchronous Programming

Events

```
.on(”click”, function () {...})
```

Callbacks `d3.csv() v4`

Promises `d3.csv() v5`

Callbacks (v4)

```
d3.csv(”mydata.csv”, function () {

    all the stuff that uses the data

});
```

Multiple line graphs with labels

bl.ocks.org/d3noob/8603837

Multi-line graph 2 with v4: Colours

https://bl.ocks.org/d3noob/ae9786c26d6a821eefeabe60dec350a9

Promises (v5)

```
d3.csv(”mydata.csv” [, row converter
function]).then(function(data) {

    all the stuff that uses the data

})
.catch(function(error){
    what to do if there's an error
});
```

Row converter function

```
var rowConverter = function (d) {
return {
   grad: +d.graduation_rate,
   attend: +d.attendance_rate,
   name: d.school_name
   };
 }
```

Row converter function

arrow functions

```
const rowConverter = d => ({
    grad: +d.graduation_rate,
    attend: +d.attendance_rate,
    name: d.school_name
    });
```

parens needed so it's clear that { } means object

Promises (v5)

```
d3.csv(”mydata.csv” [, row converter
function]).then(function(data) {

    all the stuff that uses the data

})
.catch(function(error){
    what to do if there's an error
});
```

promise is resolved

```
.then( function

// executes if promise is resolved

)
```

promise is rejected

`.catch(` *function*

`//` *executes if promise is rejected*

`)`

Example

```
d3.csv(”mydata.csv”, rowConverter).then(function(data) {
   var allCircles = svg.selectAll(”circle”)
   .data(data)
   .enter()
   .append(”circle”)
   .attr(”cx”, d => d.attend)
   .attr(”cy”, d => d.grad)
   .attr(”r”, ”3”);
})
.catch(function(error){
   d3.select(”text#message”).text(”Error loading data.”);
});
```

Loading files

1. Use a local server to read local files (***will not work with CourseWorks***)

2. Work online blockbuilder.org, GitHub Pages, etc.

3. Upload files and then read from URL