D3 Data Loading with Promises

1: Design and Data Loading with Promises

2: Scales

3. Axes

4: Drawing SVG and Interactivity

JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

```
d3.csv( <file>, <row conversion> )
.then( <callback function> )
.catch( <callback function> );
```

JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

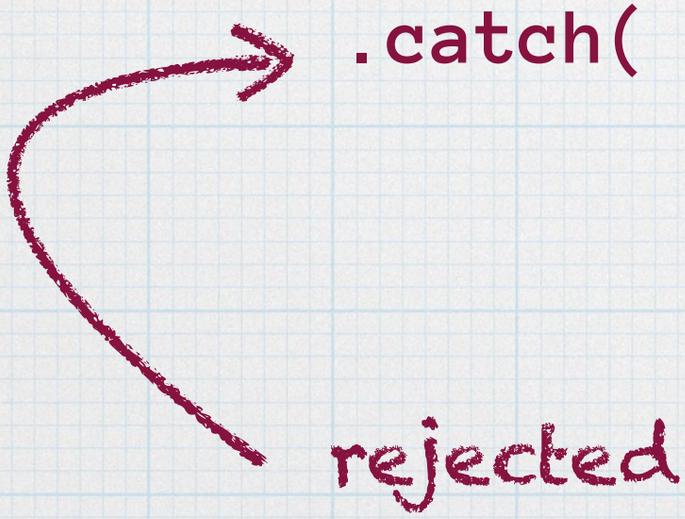
```
d3.csv( <file>, <row conversion> )
    .then( <callback function> )
    .catch( <callback function> );
```

fulfilled

JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

```
d3.csv( <file>, <row conversion> )
    .then( <callback function> )
    .catch( <callback function> );
```



JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

```
d3.csv( <file>, <row conversion> )
.then( <callback function> )
.catch( <callback function> );
```

a file to load

JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

```
d3.csv( <file>, <row conversion> )
  .then( <callback function> )
  .catch( <callback function> );
```

"An optional row conversion function may be specified to map and filter row objects to a more-specific representation."

D3 API documentation

JAVASCRIPT PROMISES

D3 version 5 has abandoned the Ajax model of data loading and uses a new Javascript function called **`fetch()`**, which handles asynchronicity differently. A function like **`d3.csv()`** now returns a "***promise***," an obligation that will eventually either be ***fulfilled*** (if the data loads properly, for example) or ***rejected*** (if the CSV file doesn't exist, for example). While in progress, the promise is ***pending***.

```
d3.csv('data/parents_1550.csv', clean_rows)
.then( function(d) {} )
.catch( function(e) {} );
```

D3 Scatterplot Project

1: Design and Data Loading with Promises

2: Scales

3. Axes

4: Drawing SVG and Interactivity