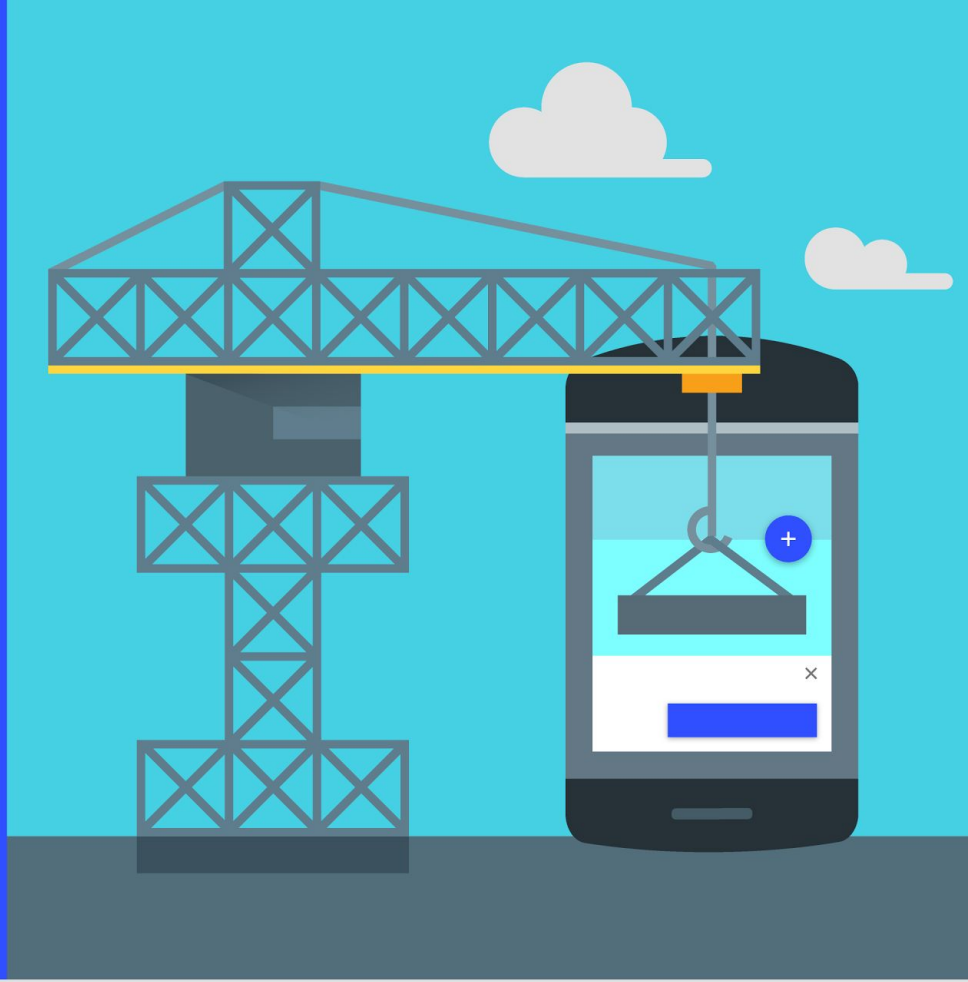
Working with Promises



Agenda

- What is a Promise?
  - Compare with callbacks
- Other uses for Promises
  - Executing promises concurrently
  - Running a promise against a timer

What is a Promise?

```
function isUserTooYoung(id, callback) {
  openDatabase(db => {
      getCollection(db, 'users', col => {
        find(col, {'id': id}, result => {
                result.filter(user => {
                  callback(user.age < cutoffAge);
                });
              });
         });
    });
}
```

What is a Promise?

```
function isUserTooYoung(id, callback) {
  openDatabase(db => {getCollection(db, 'users', col => {
      find(col, {'id': id}, result => {
        result.filter(user => {
          callback(user.age < cutoffAge);
        });
      });
    });
  });
}
```

What is a Promise?

```
var url = 'https://foo.com/foo.json';

function getJSON(url) {
  fetch(url)
  .then(function(response) {return response.json();})
  .then(function(json) {console.log('JSON: ', json);})
  .catch(function(error) {console.log('Error:', error);});
}

getJSON(url);
```

Promise Terminology

```
var p = new Promise(function(resolve, reject) {
  // Promise is pending
  // Perform an async task

  if (1 === 1) { // Did the task succeed?
    resolve('Success!');
    // It succeeded. Promise has been fulfilled
  }
  else {
    reject('Failure!');
    // It failed. Promise has been rejected
  }
});
```

Immutable results

```
function loadImage(url) {
  return new Promise((resolve, reject) => {
    var image = new Image();
    image.src = url;
    image.onload = () => {
      resolve(image);
    };
    image.onerror = () => {
      reject(new Error('Could not load image at ' + url));
    };
  });
}
```

Promise chains, then and catch

```
function processImage(image) {
  loadImage(image)
    .then(image => {
      document.body.appendChild(image);
      return scaleToFit(150, 225, image);
    })
    .then(image => {
      return watermark('Google Chrome', image);
    })
    .catch(error => {
      console.log('we had a problem in running processImage ', error);
    });
}
```

Running Promises concurrently

```
var request1 = fetch('/users.json');
var request2 = fetch('/articles.json');

Promise.all([request1, request2]) // Array of Promises to complete
.then((results) => {
  console.log('All data has loaded');
})
.catch((error) => {
  console.log('One or more requests have failed: ', error);
});
```

Running a Promise under a certain time

```
var promise1 = new Promise((resolve, reject) => {
  setTimeout(() => resolve('promise1'), 100);
});

var promise2 = new Promise((resolve, reject) => {
  setTimeout(() => resolve('promise2'), 200);
});

Promise.race([promise1, promise2]).then(function(winner) {
  console.log('First to resolve:', winner);
});
```