```
. then () always returns a new promise. What happens to that promise?
                                     promiseB = promiseA.then( [successHandler], [errorHandler] );
                                                                          when
                                                                       promiseA
                                                                            is
// promiseA is fulfilled with 'hello'
                                                                                                                         // promiseA is rejected with 'bad request'
                                                 fulfilled with
                                                                                              rejected with
                                                                                                                         promiseA
                                                   data
                                                                                               reason
                                                                                                                           .then( myFunc1, null, myFunc2 )
                                                                                                                           .then()
                                                     and
                                                                                                  and
                                                                                                                           .then( null, console.log );
                                     has no success.
                                                                                                        has no error
                                                                                          has an
                                                          has a
                                         handler
                                                                                                          handler
                                                                                           error
                                                        success
                                                                        handler
                                                                                          handler
                                                         handler
                                                                                                       promiseB reject(reason);
                                                                          which
                                                                                                           REJECTION BUBBLED
                                          returns
                                                                         returns
                                                                                                         throws
                                          result
                                                                      promiseZ
                                                                                                          err
                                                                promiseB ← promiseZ
                                                                                                         promiseB reject(err);
                                                                     ASSIMILATION
                                                                                                              ERROR CAUGHT
                                                                                                               // promiseForVal2 will be rejected with error '404':
                                                          promiseForMessages = promiseForUser
                                                                                                               promiseForVal2 = promiseForVal1
                                                            .then( function success (user) {
                                                                                                                .then( function success (val1) {
                                                             // do some code to make a new promise
                                                                                                                  // THROWN ERROR '404' trying to make val2
                                                                                                                  return val2:
```

```
promiseB resolve(data);
                         FULFILLMENT BUBBLED
                     promiseB resolve(result);
                          PROMISE FOR RESULT
// promiseForVall is fulfilled with 999
promiseForVal2 = promiseForVal1
  .then( function success (val1) {
   val2 = val1++;
   return val2:
                                                                  return promiseForMessages;
 });
                                                                });
                                                                                                                          });
// result: promiseForVal2 is fulfilled with 1000.
                                                              // same idea, shown in a direct chain:
                                                                                                                         // same idea, shown in a direct chain:
// same idea, shown in a direct chain:
                                                              promiseForUser
                                                                                                                         promiseForVal1
                                                                .then( function success (user) {
                                                                                                                           .then( function success (val1) {
promiseForVal1
                                                                  // do some code to make a new promise
                                                                                                                            // THROWN ERROR '404' trying to make val2
  .then( function success (val1) {
                                                                  return promiseForMessages;
                                                                                                                            return val2;
    // do some code to make val2
                                                               })
    return val2; -
                                                                .then( function success (messages) {
                                                                                                                           .then( null, function failed (err) {
                                                                  console.log( messages );
                                                                                                                            // handle the error as you prefer
  .then( function success (val2) {
                                                                                                                            console.log('Oops!', err);
    console.log( val2 );
                                                                                                                          });
  });
                                                                                                                         // console shows 'Oops! 404'
                                                                        © 2014 GABRIEL LEBEC
```

promiseA

.then()

.then(null, | myFunc1, myFunc2)

// result: console shows 'hello'

.then(console.log);