



JavaScript promises



Michał Okulewicz

Warsaw University of Technology

Faculty of Mathematics and Information Science

M.Okulewicz@mini.pw.edu.pl

<http://www.mini.pw.edu.pl/~okulewicz>

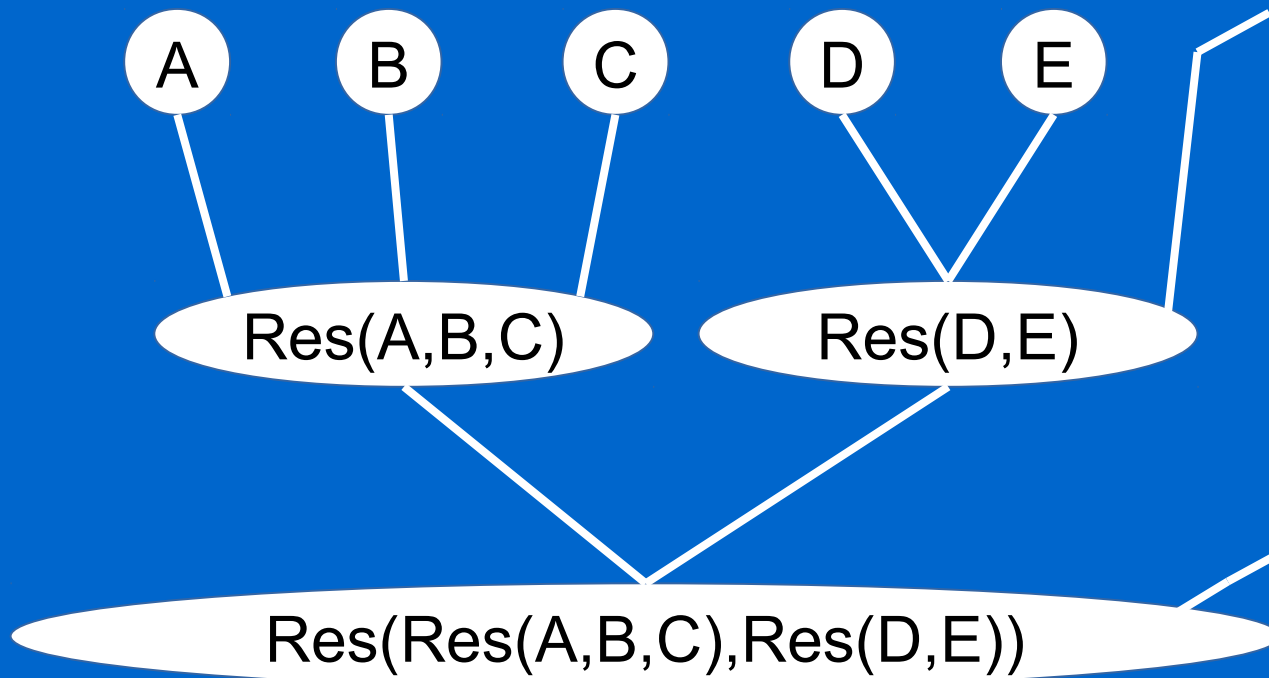


Asynchronous programming

- General idea:
 - The code is executed in order defined by available resources and available data
 - The task of the developer is to define the data necessary to compute a given value
 - One may think about it as an event-driven programming



Asynchronous programming



Developer defines the relation between data and results

The final result is computed when A, B, C, D, E are available



JavaScript Promise object

- After `developer.mozilla.org`:
 - The **Promise** object is used for asynchronous computations
 - A **Promise** represents a value which may be available now, or in the future, or never



Promise example

```
var promise = new Promise(function (resolve, reject) {
  var result = Math.random();
  setTimeout(function () {
    if (result < 0.8) {
      resolve(result);
    } else {
      reject("No luck!");
    }
  }, 1000);
});
promise
  .then(function (val) {
    console.log('Promise fulfilled with ' + val);
  })
  .catch(function (msg) {
    console.error('Promise failed because: ' + msg);
  });
```



Promises in jQuery

- Promises are wrapped up in a jQuery **Deferred** object
- jQuery nicely supports waiting for multiple promises with a jQuery **when ()** function



Promise example in jQuery

```
promise = $.Deferred();
var result = Math.random();
setTimeout(function () {
    if (result < 0.8) {
        promise.resolve(result);
    } else {
        promise.reject("No luck!");
    }
}, 1000);
promise
    .then(function (val) {
        console.log('Promise fulfilled with ' + val);
    })
    .catch(function (msg) {
        console.error('Promise failed because: ' + msg);
    });
```



Multiple promises

```
$(document).ready(function () {
    var val1 = $.Deferred();
    var val2 = $.Deferred();
    $.when(val1, val2)
        .then(printResult)
    $.when(val1)
        .then(console.log);
    $.when(val2)
        .then(console.log);
    $('#val1').change(function () {
        val1.resolve(this.value);
    });
    $('#val2').change(function () {
        val2.resolve(this.value);
    });
});

function printResult(a1, a2) {
    $('#result').html(Number(a1) + Number(a2));
}
```




Animation

```
$(document).ready(function () {
    $('#square').click(function () {
        var anim = $(this).animate(
            { left: '+=10px', top: '+=5px' },
            1000);
        $
        .when(anim)
        .then(function () {
            console.log('Animation finished ('
                + $(this).position().top + ', '
                + $(this).position().left + ')');
        });
    });
});
```