
Deferreds
and
Promises
Edwin Martin, oktober 2012

Deferreds en promises maken het omgaan met asynchrone code makkelijker.

- Schone code
- Leesbare code
- Beter onderhoudbaar
- Minder bugs

Asynchroon

- Wat is dat?

Chronos, Griekse god van de tijd

Asynchroon

- Chronos = tijd
- Syn = samen
- Synchroon = gelijktijdig
- Asynchroon = niet gelijktijdig

Tijd een miljard keer vertraagd

- Berekeningen en vergelijkingen duren enkele seconden

Peter Burns https://plus.google.com/112493031290529814667/posts/LvhVwngPqSC

Tijd een miljard keer vertraagd

- Berekeningen en vergelijkingen duren enkele seconden
- Gegevens uit het geheugen ophalen kost acht minuten

Tijd een miljard keer vertraagd

- Berekeningen en vergelijkingen duren enkele seconden
- Gegevens uit het geheugen ophalen kost acht minuten
- Gegevens van de harddisk lezen kost drie jaar

blocking

...

```
read( fileHandle, buffer, 1024 );
```

...


TheMaveSite.Com

JavaScript / jQuery

```
$.get(“http://fronteers.nl/”, callback);

…

function callback(data) {
…
}
```

AJAX

Asynchronous

JavaScript / jQuery

```
$.get(“http://fronteers.nl/”, callback);

…

function callback(data) {
…
}
```

```
element.onclick = function (evt) {
… // Doe A
}
```

```
element.onclick = function (evt) {
… // Doe A
}

…

element.onclick = function (evt) {
… // Doe B
}
```

Traditionele jQuery Ajax

```
$.ajax( {
  url: "/myServerScript",
  success: mySuccessFunction,
  error: myErrorFunction
} );
…
function mySuccessFunction( data ) {
…
}
```

jQuery Ajax met een promise

```
var promise = $.ajax({ url: "/myServerScript" });

promise.done(mySuccessFunction);
promise.fail(myErrorFunction);
```

meerdere callbacks

```
var promise = $.ajax({ url: "/myServerScript" });

promise.done(myStopAnimationFunction);
promise.done(myOtherAjaxFunction);
promise.done(myShowInfoFunction);
promise.fail(myErrorFunction);
```

onveranderbaar

```
var promise = $.ajax({ url: "/myServerScript" });

promise.done(myStopAnimationFunction);
promise.done(myOtherAjaxFunction);

…

promise.done(myShowInfoFunction);
promise.fail(myErrorFunction);
```

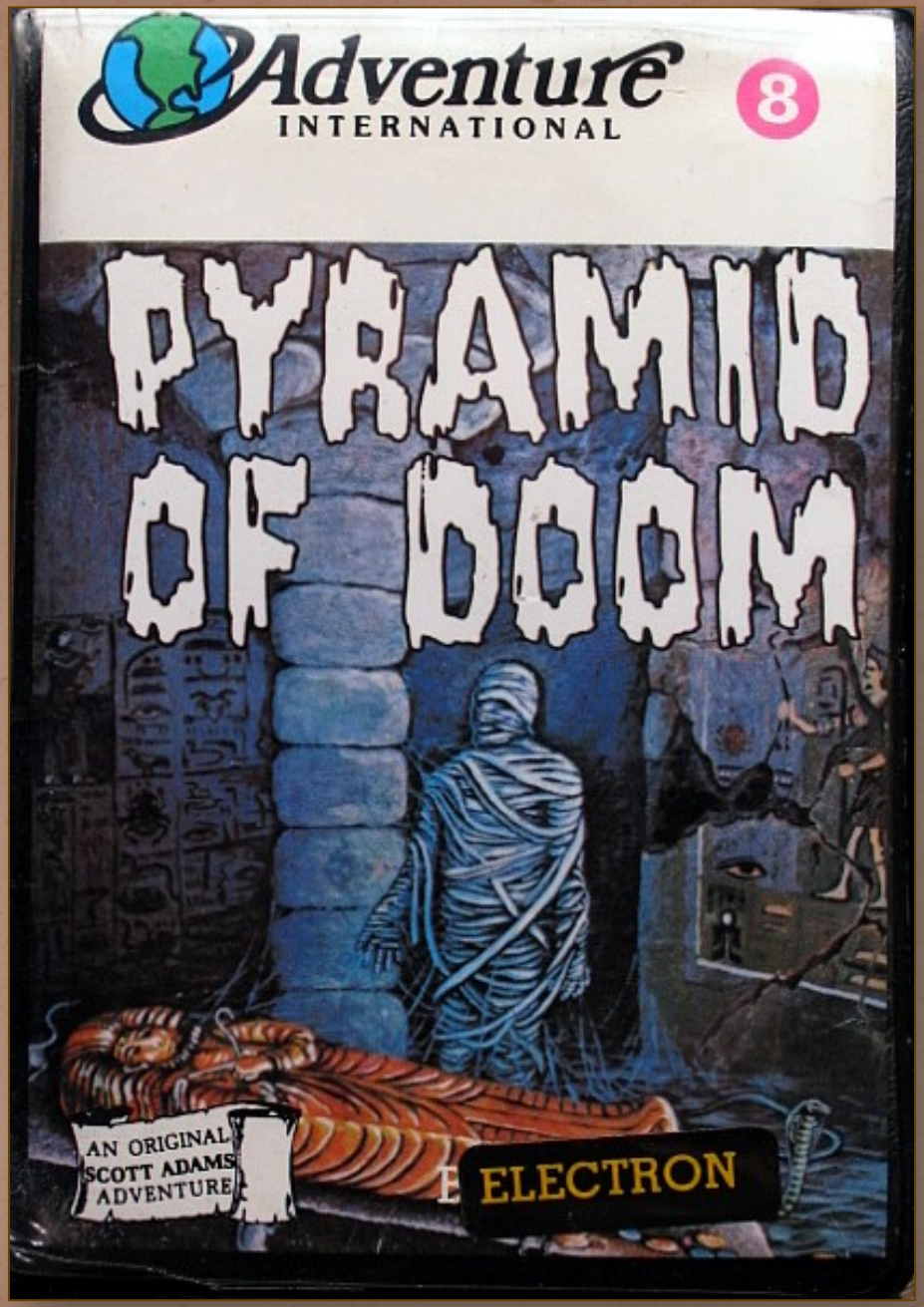
parallel

when

```
var promise1 = $.ajax("/myServerScript1");
var promise2 = $.ajax("/myServerScript2");

$.when( promise1, promise2 ).done( function( respon
se1, response2 ){
        // Handle both responses
});
```

```
doThis(param1, function (data1) {
    doThat(param2, function(data2) {
        doSomethingMore(param3, function(data3) {
            // Do something with data3
        });
    });
});
```


Adventure
INTERNATIONAL
8
PYRAMID OF DOOM
AN ORIGINAL SCOTT ADAMS ADVENTURE
ELECTRON


doThis(param1, function (data1) {
doThat(param2, function(data2) {
doSomethingMore(param3, function(data3) {
// Do something with data3
});
});
});

then

```
var promise = initialise();

function myAsync() { // returns promise
        return $.ajax("/myServerScript2");
}

promise.then(myAsync).then(function(ajaxResponse) {
        // Both promises are resolved
});
```


USED PIPED PROMISES
DEFEATED PYRAMID OF DOOM!
memegenerator.net

```
promise.done(myShowInfoFunction);
promise.fail(myErrorFunction);
```

Zelfde als

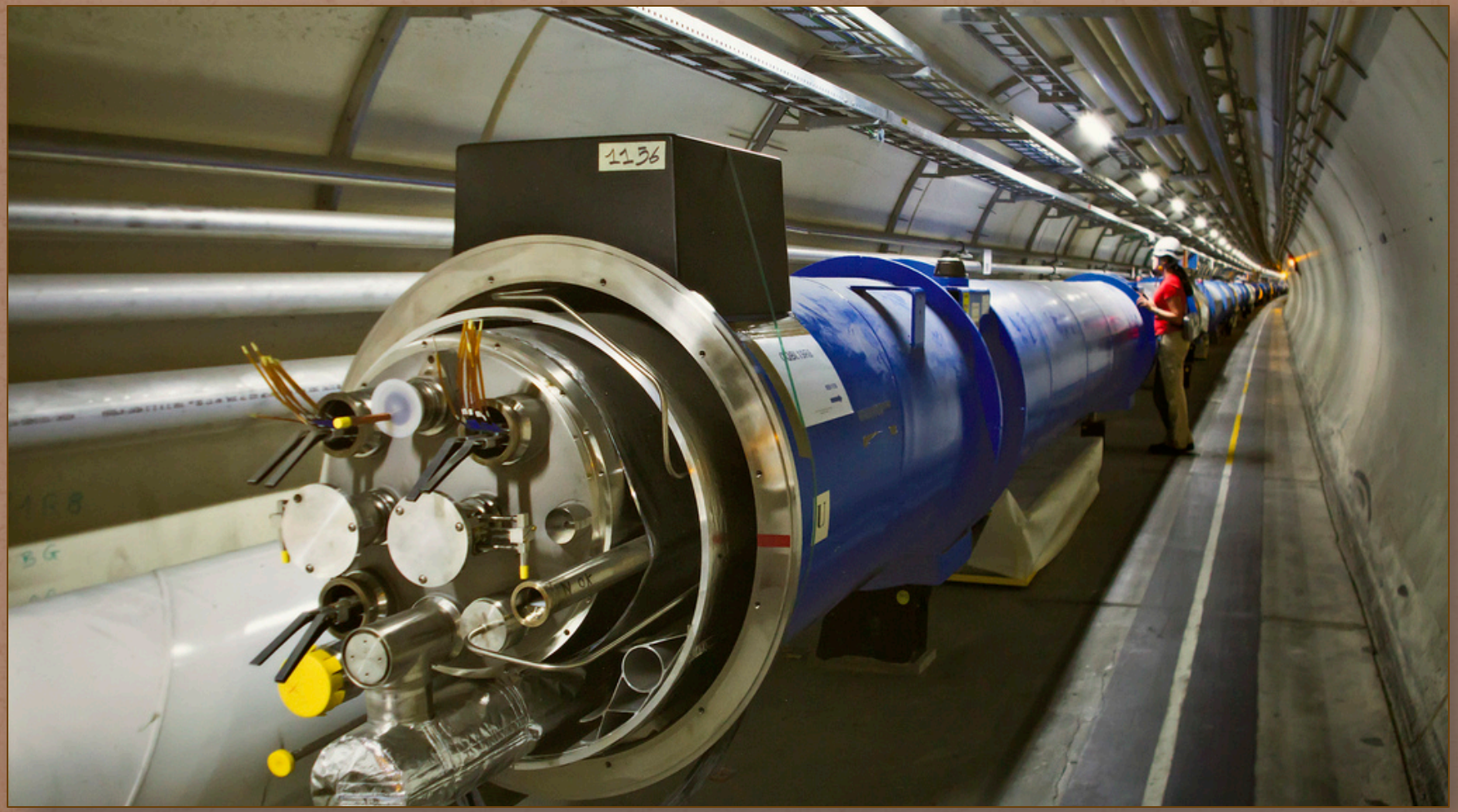
```
Promise.then(myShowInfoFunction, myErrorFunction);
```

Met when() en then() kan je elk asynchroon probleem oplossen.

Promises, deferred

Wat is het verschil?

zelf maken



```
$('#result').html('waiting...');
var promise = wait();
promise.done(result);
function result() {
        $('#result').html('done');
}
function wait() {
        var deferred = $.Deferred();
        setTimeout(function() {
                deferred.resolve();
        }, 2000);
        return deferred.promise();
}
```

Meer meer meer!

- state()
- pipe()
- progress()

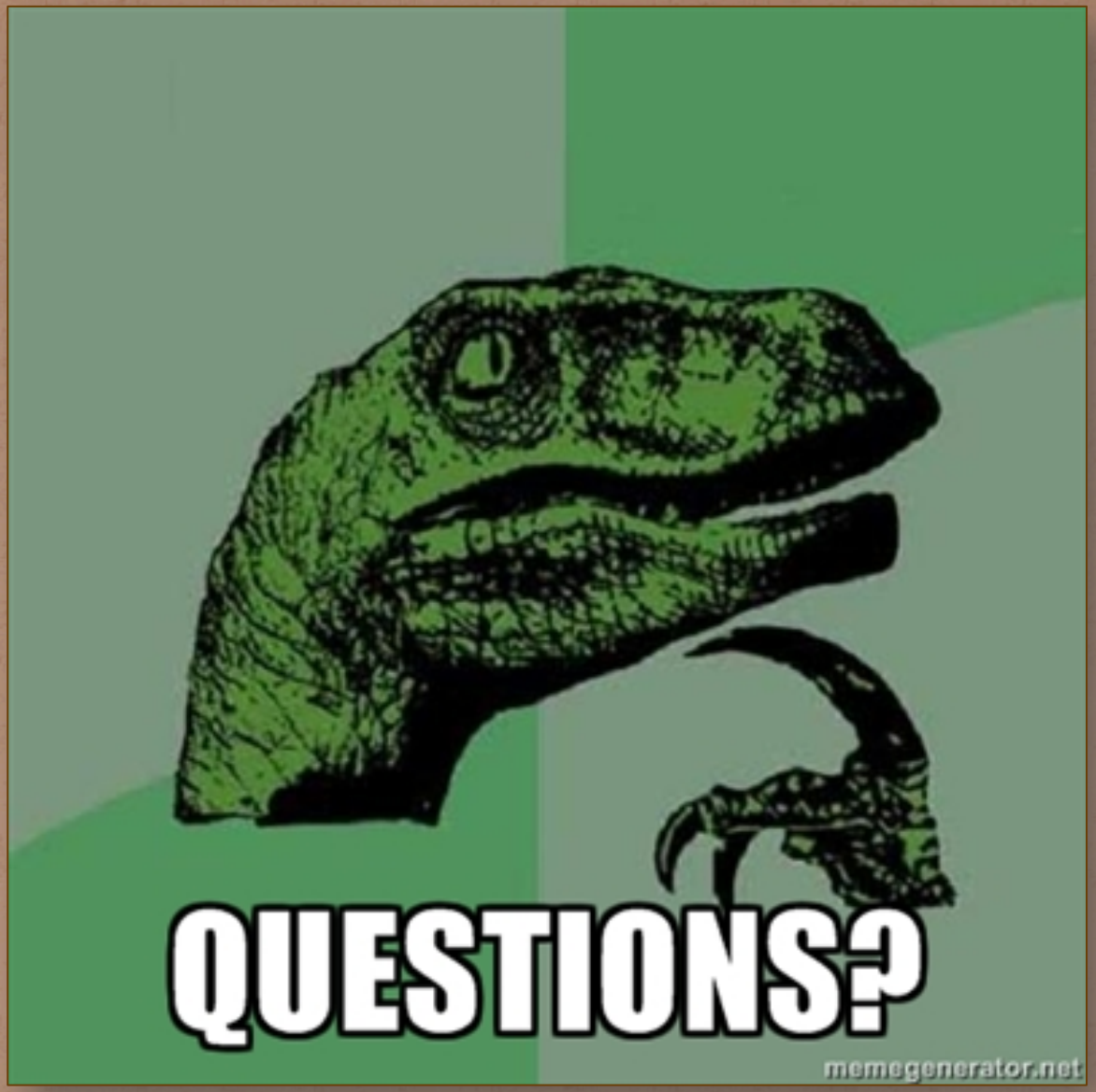
You're Missing the Point of Promises

Domenic Denicola https://gist.github.com/3889970

Dit werkt niet in jQuery:
promise.then(fn1).then(fn2).then(fn3, fail);

Promises/A: Als in fn1 een fout optreedt (throw), dan moet de fail-functie worden aangeroepen.

Beter: Q door @kriskowal


QUESTIONS?
memegenerator.net

Credits

- http://www.flickr.com/photos/malingering/351740549/
- http://www.flickr.com/photos/hdptcar/2266885580/
- http://www.flickr.com/photos/elpablo/164703609/
- http://www.flickr.com/photos/averagejane/485938956/
- http://www.flickr.com/photos/bisgovuk/6979060787/