

Elmer Bulthuis

- Rotterdam
- LuvDaSun Software
- +15 jaar JavaScript
- @elmerbulthuis
- luvdasun.slack.com



JavaScript callbacks

hoe zit dat?

Wat is een callback?

“In computer programming, a callback is any executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at a given time.”

Wikipedia

```
setTimeout(function() {  
    alert("hello");  
}, 1000);
```

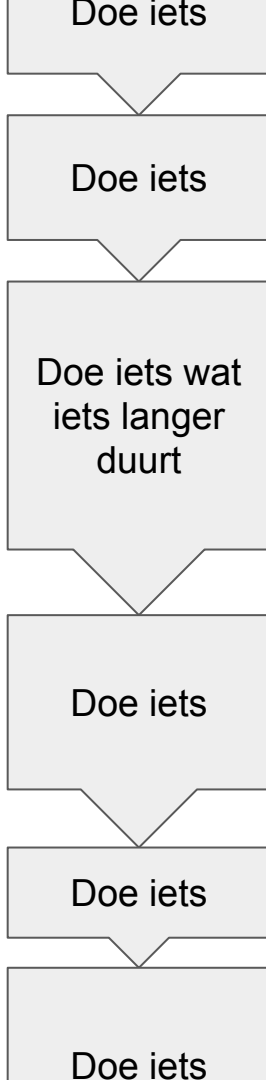
Waarom callbacks?

- JavaScript heeft maar 1 thread *
- Is een efficiënte manier om je code uit te voeren!
- Er wordt niet gewacht tot iets klaar is (Asynchroon)
- Past goed bij een user interface (event === callback)
- Dankzij callbacks wordt die thread niet geblokkeerd (DEMO)

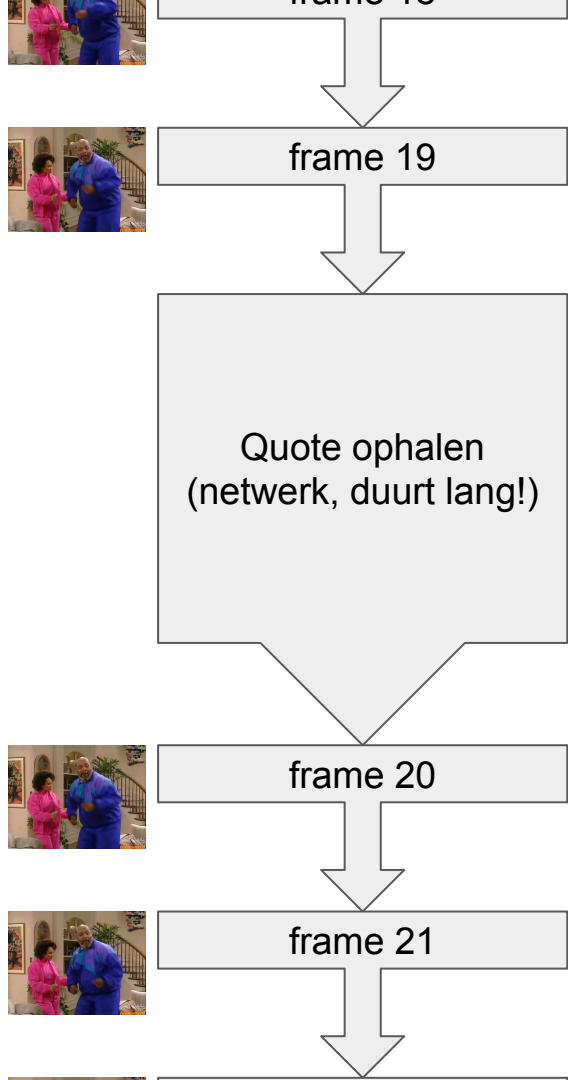
Wat is een thread?

“A thread is an execution context, which is all the information a CPU needs to execute a stream of instructions.”

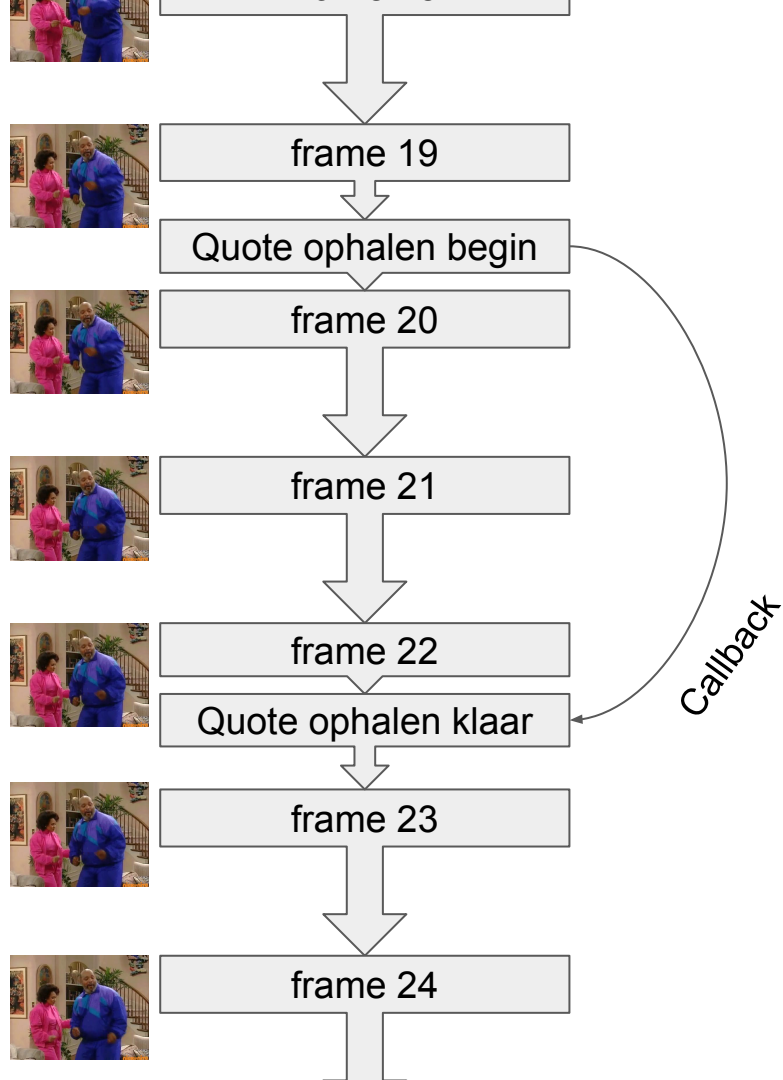
pwnall, Stack Overflow



Synchroon



Asynchroon



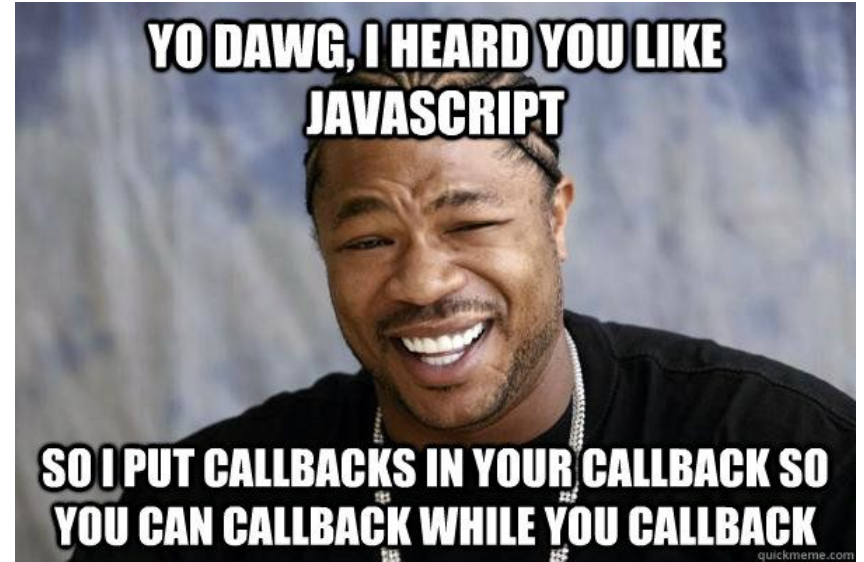
Wat als het fout gaat?

- try / catch
- Callback moet altijd exact één keer aangeroepen worden!



Het is niet zo makkelijk...

- Foutafhandeling
- Callback hell
- Pyramid of doom (DEMO)
- ...of toch wel?



Voorbeeld applicatie

1. Kies een categorie
 2. Druk op de knop
 3. ...event wachten
 - a. Als het goed gaat, zie je een quote!
 - b. Als het niet goed gaat, dan krijg je een nette melding (alert)
- Een beetje HTML
 - ... initialisatie code (events setup)
 - `getJSON`
 - `getQuote`

Synchrone JavaScript code

- Overzichtelijk
- Automatische correcte foutafhandeling
- Doet precies wat je verwacht dat het doet
- ...maar blokkeert de thread

Asynchrone code met callbacks

- Blokkeert de thread niet, soepele animatie!
- Minder overzichtelijk
- Minder makkelijk te begrijpen
- Worden alle potentiële fouten wel afgehandeld?

HTML

```
<p>
  <select id="quote-type">
    <option value="none"></option>
    <option value="explicit">explicit</option>
    <option value="dev">dev</option>
    ...
  </select>
  <button id="get-quote">Get quote!</button>
</p>
<p id="quote-display">
</p>
```

getQuote

```
function getQuote(category, cb) {  
    var url = "https://api.chucknorris.io/jokes/random" +  
        "?category=" + encodeURIComponent(category) + "";  
  
    getJSON(url, function (err, data) {  
        if (err) return cb(err);  
        cb(null, data.value);  
    });  
}
```

```
function getQuote(category) {  
    var url = "https://api.chucknorris.io/jokes/random" +  
        "?category=" + encodeURIComponent(category) + "";  
  
    var data = getJSON(url);  
  
    return data.value;  
}
```

initialisatie code...

```
getQuoteElement.addEventListener("click", function (e) {  
    getQuote(quoteTypeElement.value, function (err, quoteText) {  
        if (err) return alert("Something went wrong!");  
  
        displayQuoteElement.innerText = quoteText;  
    });  
}, false);
```

```
getQuoteElement.addEventListener("click", function (e) {  
    try {  
        var quoteText = getQuote(quoteTypeElement.value);  
        displayQuoteElement.innerText = quoteText;  
    }  
    catch (err) {  
        alert("Something went wrong!");  
    }  
}, false);
```

Promises

- callback v2
- Foutafhandeling gedeeltelijk automatisch
- letwat beter te begrijpen

getQuote

```
function getQuote(category) {  
  var url = "https://api.chucknorris.io/jokes/random" +  
    "?category=" + encodeURIComponent(category) + "";  
  
  return getJSON(url).then(function (data) {  
    return data.value;  
  });  
}
```

```
function getQuote(category) {  
  var url = "https://api.chucknorris.io/jokes/random" +  
    "?category=" + encodeURIComponent(category) + "";  
  
  var data = getJSON(url);  
  
  return data.value;  
}
```


initialisatie code...

```
getQuoteElement.addEventListener("click", function (e) {  
  try {  
    var quoteText = getQuote(quoteTypeElement.value);  
    displayQuoteElement.innerText = quoteText;  
  }  
  catch (err) {  
    alert("Something went wrong!");  
  }  
}, false);
```

```
getQuoteElement.addEventListener("click", function (e) {  
  getQuote(quoteTypeElement.value).then(function (quoteText) {  
    displayQuoteElement.innerText = quoteText;  
  }, function (err) {  
    alert("Something went wrong!");  
  });  
}, false);
```

Async / await

- Lijkt wel heel erg veel op synchrone code!
- Dus ook alle voordelen van synchrone code
- try / catch werkt gewoon!

getQuote

```
async function getQuote(category) {  
    var url = "https://api.chucknorris.io/jokes/random" +  
        "?category=" + encodeURIComponent(category) + "";  
  
    var data = await getJSON(url);  
  
    return data.value;  
}
```

```
function getQuote(category) {  
    var url = "https://api.chucknorris.io/jokes/random" +  
        "?category=" + encodeURIComponent(category) + "";  
  
    var data = getJSON(url);  
  
    return data.value;  
}
```

initialisatie code...

```
getQuoteElement.addEventListener("click", function (e) {  
  try {  
    var quoteText = getQuote(quoteTypeElement.value);  
    displayQuoteElement.innerText = quoteText;  
  }  
  catch (err) {  
    alert("Something went wrong!");  
  }  
}, false);
```

```
getQuoteElement.addEventListener("click", async function (e) {  
  try {  
    var quoteText = await getQuote(quoteTypeElement.value);  
    displayQuoteElement.innerText = quoteText;  
  }  
  catch (err) {  
    alert("Something went wrong!");  
  }  
}, false);
```

Async / await modern (bonus)

- Nu met string templates
- Arrow functions
- Block scoped variables (const)

getQuote

```
async function getQuote(category) {  
  var url = "https://api.chucknorris.io/jokes/random" +  
    "?category=" + encodeURIComponent(category) + "";  
  var data = await getJSON(url);  
  return data.value;  
}
```

```
async function getQuote(category) {  
  const url = `https://api.chucknorris.io/jokes/random?category=${encodeURIComponent(category)}`;  
  const data = await getJSON(url);  
  return data.value;  
}
```

initialisatie code...

```
getQuoteElement.addEventListener("click", async e => {  
  try {  
    const quoteText = await getQuote(quoteTypeElement.value);  
    displayQuoteElement.innerText = quoteText;  
  }  
  catch (err) {  
    alert("Something went wrong!");  
  }  
}, false);
```

```
getQuoteElement.addEventListener("click", async function (e) {  
  try {  
    var quoteText = await getQuote(quoteTypeElement.value);  
    displayQuoteElement.innerText = quoteText;  
  }  
  catch (err) {  
    alert("Something went wrong!");  
  }  
}, false);
```

Ik wil dat!

- edge 15
- firefox 52
- chrome 55
- safari 10.1
- opera 42
- ios safari 10.3
- android browser 56
- chrome for android 57
- node.js 7.10.0
- Babel
- Typescript <3

Bedankt voor je aandacht!

- Elmer Bulthuis
- @elmerbulthuis
- lvdasun.slack.com
- Bij vragen, gewoon vragen!