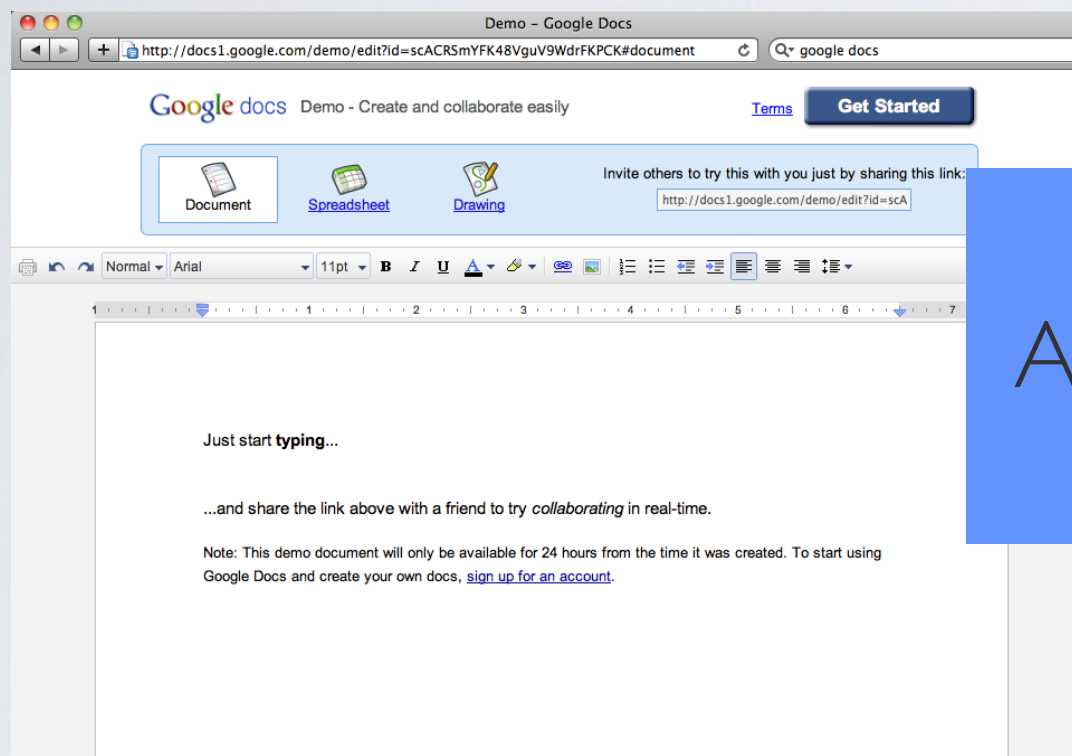


Ajax

Thierry Sans

Ajax - fetching data without refreshing the page

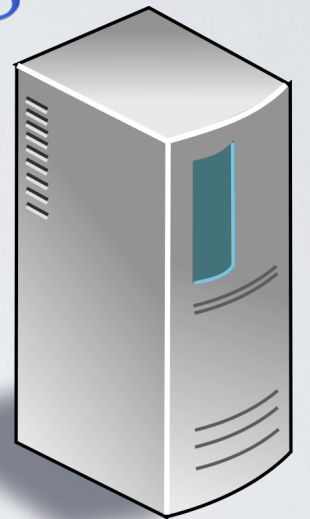


Ajax

id=scACRSm...

anything

Google



Javascript

Why do we need Ajax?

So far, when we wanted to

- send data to the server
- or retrieve data from the server
- we had to refresh the entire page
(i.e reloading HTML, CSS, JS and all media files)
- ✓ But, why not using Javascript to process the data
and perform the necessary page changes?

Ajax - Asynchronous Javascript And XML

Fetch/push content from/to the server asynchronously
i.e without having to refresh the page

- ⦿ Ajax is not a language

- ✓ It is a simple **Javascript command**

History of Ajax

- Patent from Microsoft (filled in 2000, granted in 2006)
 - XMLHttpRequest ActiveX control (Internet Explorer 5)
- Adopted and adapted by Opera, Mozilla and Apple
 - XMLHttpRequest Javascript object (standard)
- Before / After IE7
 - ◉ Different code for different browser (emergence of the javascript framework *Prototype*)
 - ✓ Javascript Object was adopted by IE7

Ajax revolutionized the Web

✓ Started with Gmail and Google Maps

- Advantages
 - Low latency
 - Rich interactions
- Consequences
 - Webapp center of gravity moved to the client side
 - Javascript engine performance race

Standard Ajax

```
var xhr = new XMLHttpRequest();  
xhr.onreadystatechange = function () {  
    ...  
}  
xhr.setRequestHeader(key, value);  
xhr.open(method, url, true);  
xhr.send(body);
```

(always) asynchronous

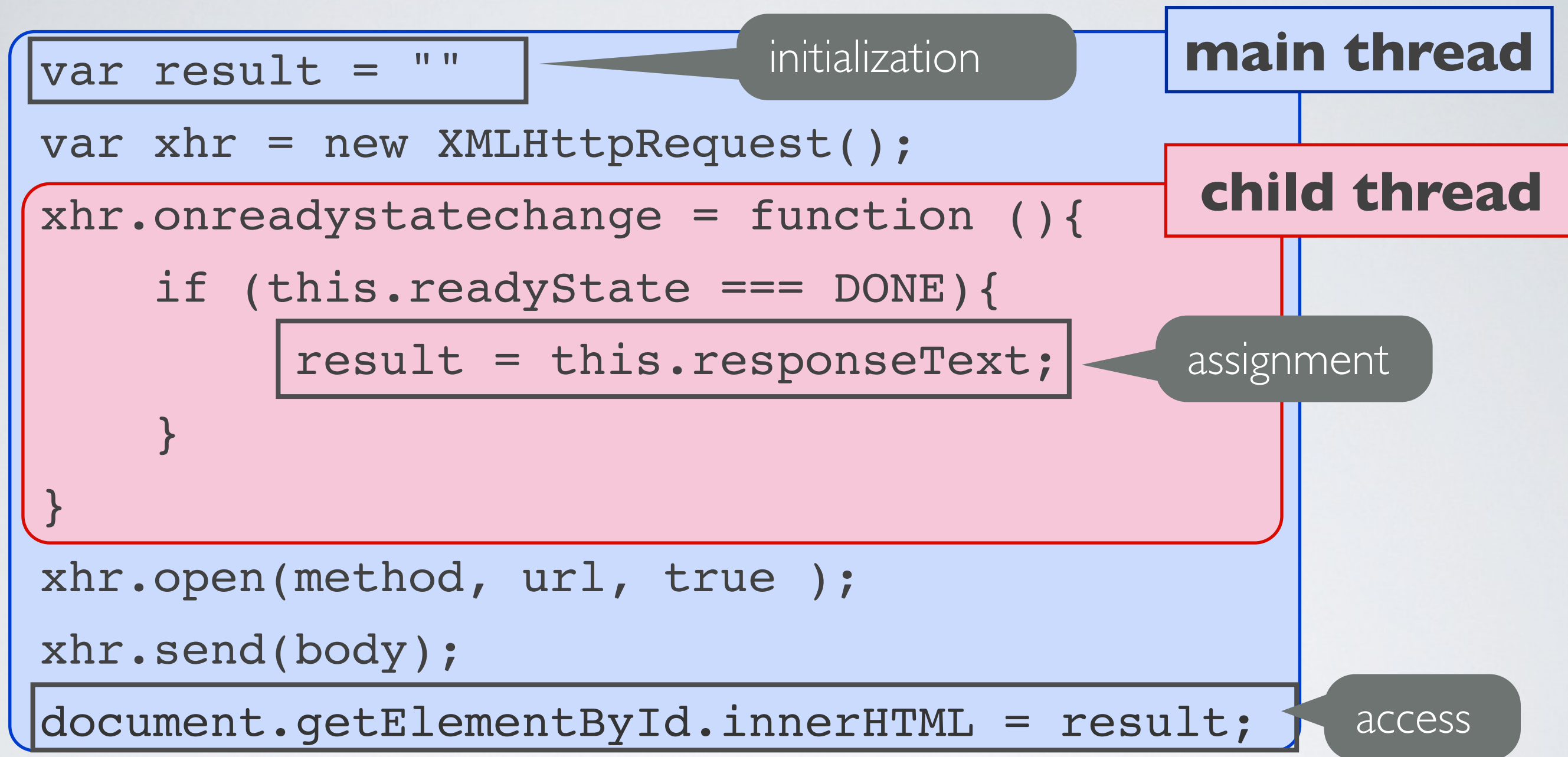
XMLHttpRequest.readyState

Value	State	Description
0	UNSENT	Client has been created open () not called yet.
1	OPENED	open () has been called.
2	HEADERS_RECEIVED	send () has been called headers and status are available.
3	LOADING	Downloading responseText holds partial data.
4	DONE	The operation is complete.

Handling the response

```
xhr.onreadystatechange = function () {  
    switch(this.readyState) {  
        case (UNSENT):  
            ...  
        case (OPENED):  
            ...  
        case (HEADERS_RECEIVED):  
            ...  
        case (LOADING):  
            ...  
        case (DONE):  
            console.log(this.status, this.responseText);  
        }  
    }  
}
```

Concurrency issue in Ajax - a typical example



`result` will either be `""` or `"Hello world"`
depending on the execution (non determinism)
➔ **Race Condition!**