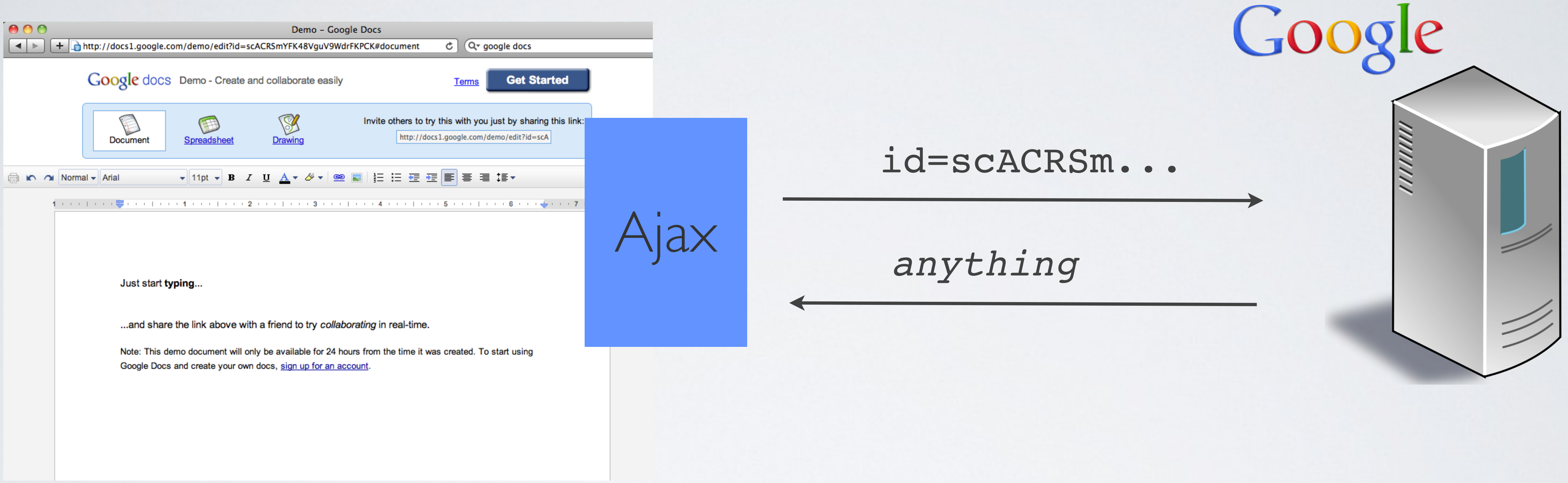
Ajax

Thierry Sans

Ajax - fetching data without refreshing the page



Demo - Create and collaborate easily

Invite others to try this with you just by sharing this link:

http://docs1.google.com/demo/edit?id=scA

Just start **typing**...

...and share the link above with a friend to try *collaborating* in real-time.

Note: This demo document will only be available for 24 hours from the time it was created. To start using Google Docs and create your own docs, sign up for an account.

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)
  - XMLHTTP 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.onload = function(){
  if (xhr.status !== 200)
     console.error("[" + xhr.status + "]" + xhr.responseText);
  else
     console.log(xhr.responseText);
};
xhr.setRequestHeader(key, value);
xhr.open(method, url, true);
xhr.send(body);
```

(always) asynchronous

Concurrency issue in Ajax - a typical example

```
var result = ""
var xhr = new XMLHttpRequest();
xhr.onload = function (){
            result = xhr.responseText;
}
xhr.open(method, url, true );
xhr.send(body);
document.getElementById.innerHTML = result;
```

initialization

asynchronous

assignment

access

`result` will either be `""` or `“Hello world”` depending on the program and the execution context

➡ **Race Condition!**