

Web Security

CS 161: Computer Security

Prof. Raluca Ada Popa

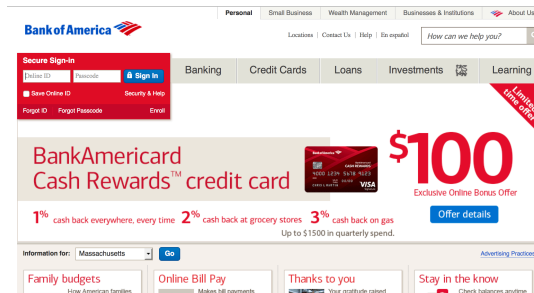
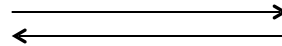
April 1, 2019

What is the Web?

A platform for deploying applications and sharing information,
portably and securely

client browser

web server

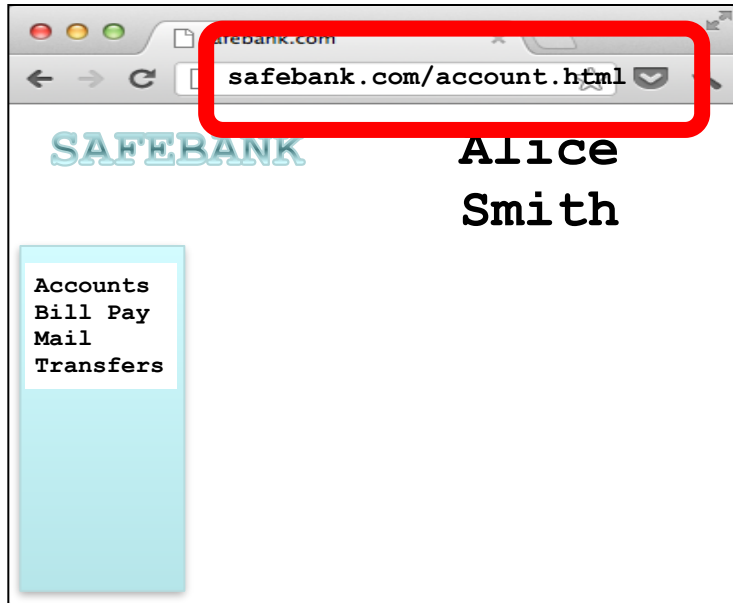


HTTP

(Hypertext Transfer Protocol)

A common data communication protocol on the web

CLIENT BROWSER



WEB SERVER

HTTP REQUEST:

```
GET /account.html HTTP/1.1  
Host: www.safebank.com
```



HTTP RESPONSE:

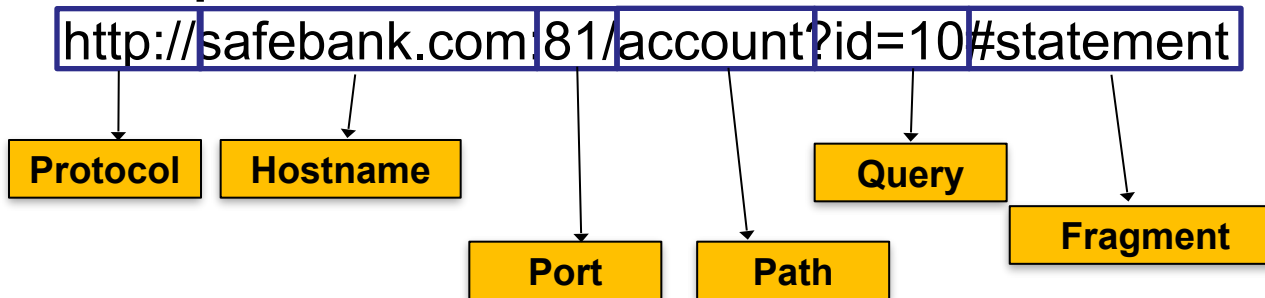
```
HTTP/1.0 200 OK  
<HTML> . . . </HTML>
```



URLs

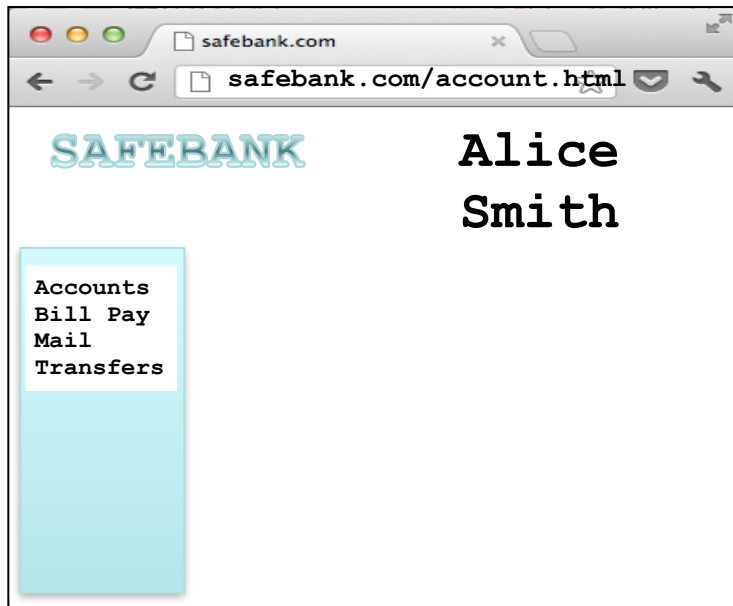
Global identifiers of network-retrievable resources

Example:



HTTP

CLIENT BROWSER



WEB SERVER

HTTP REQUEST:

```
GET /account.html HTTP/1.1  
Host: www.safebank.com
```



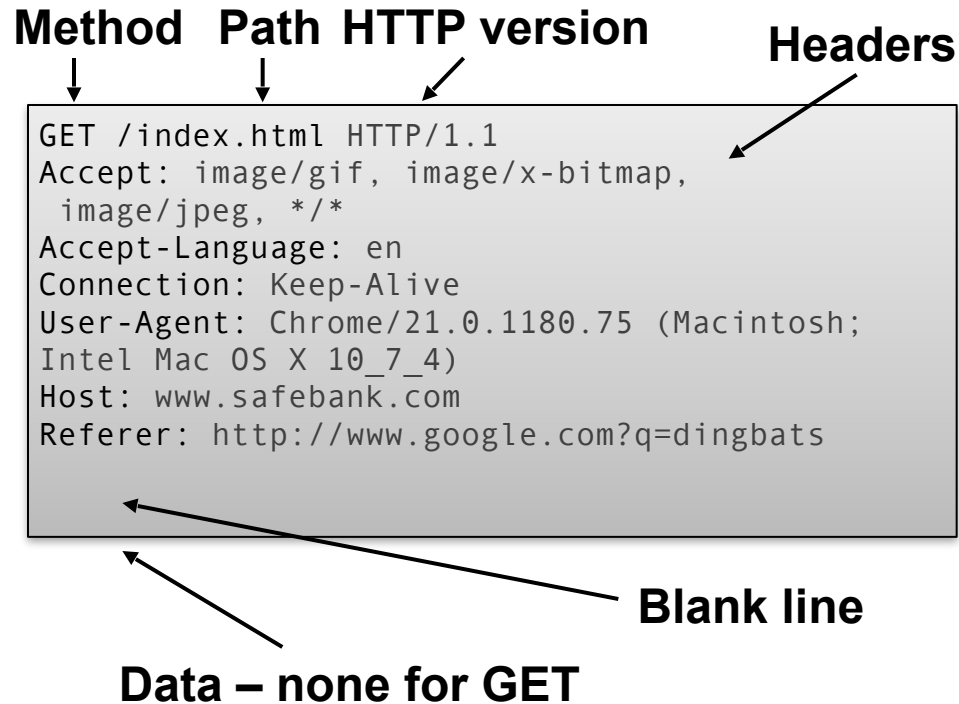
HTTP RESPONSE:

```
HTTP/1.0 200 OK  
<HTML> . . . </HTML>
```



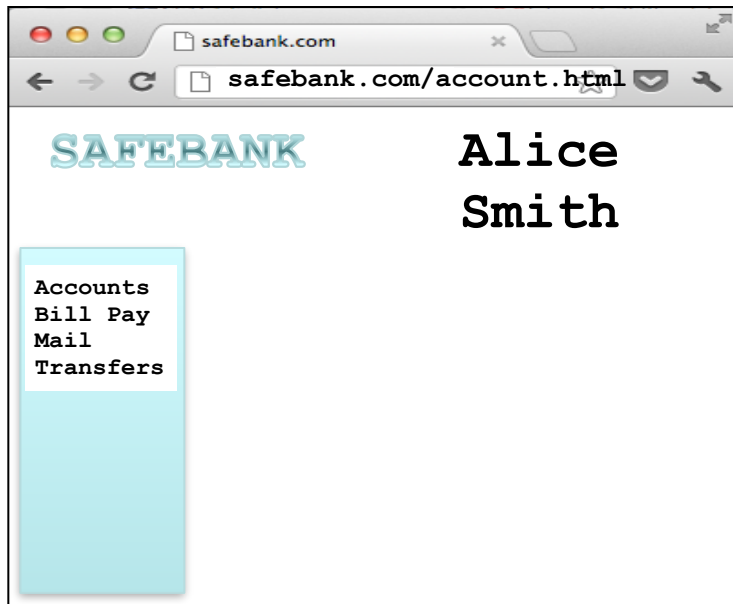
HTTP Request

GET: no side effect
POST: possible side effect



HTTP

CLIENT BROWSER



WEB SERVER

HTTP REQUEST:

GET /account.html HTTP/1.1
Host: www.safebank.com



HTTP RESPONSE:

HTTP/1.0 200 OK
<HTML> . . . </HTML>



HTTP Response

HTTP version **Status code** **Reason phrase**

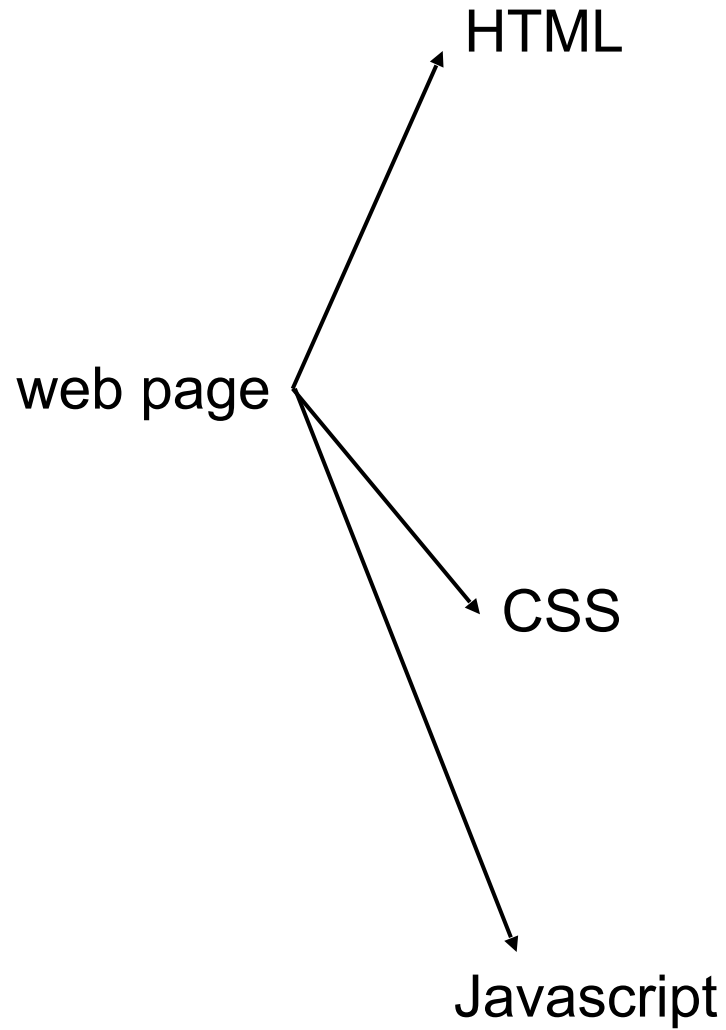
Headers

```
HTTP/1.0 200 OK
Date: Sun, 12 Aug 2012 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/
5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 9 Aug 2012 17:39:05 GMT
Set-Cookie: ...
Content-Length: 2543
<HTML> This is web content formatted using html
</HTML>
```

Data

Can be a webpage

Web page



HTML

A language to create structured documents

One can embed images, objects, or create interactive forms

```
index.html
```

```
<html>
  <body>
    <div>
      foo
      <a href="http://google.com">Go to Google!</a>
    </div>
    <form>
      <input type="text" />
      <input type="radio" />
      <input type="checkbox" />
    </form>
  </body>
</html>
```

CSS (Cascading Style Sheets)

Style sheet language used for describing the presentation of a document

index.css

```
p.serif {  
font-family: "Times New Roman", Times, serif;  
}  
p.sansserif {  
font-family: Arial, Helvetica, sans-serif;  
}
```

Javascript

Programming language used to manipulate web pages. It is a high-level, untyped and interpreted language with support for objects.

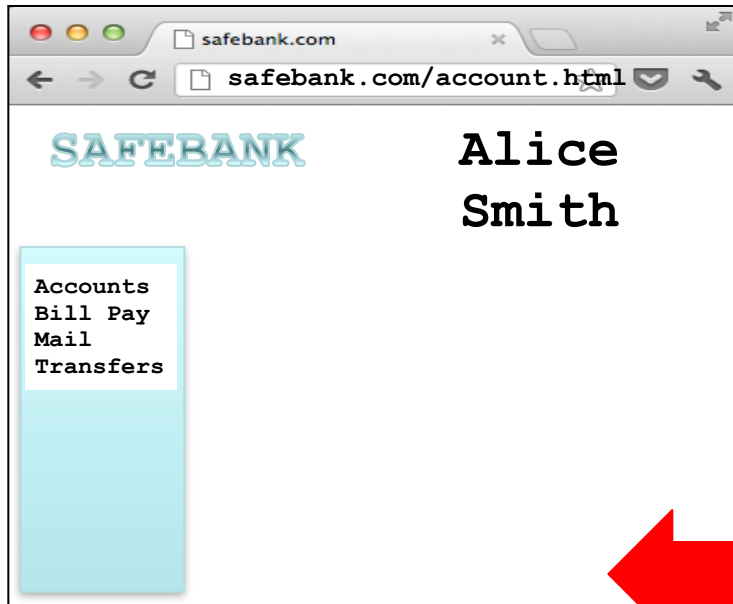
Supported by all web browsers

```
<script>
function myFunction()
{
    document.getElementById("demo").innerHTML = "Text
changed.";
}
</script>
```

Very powerful!

HTTP

CLIENT BROWSER



WEB SERVER

HTTP REQUEST:

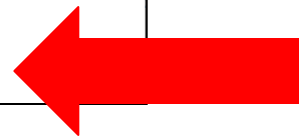
```
GET /account.html HTTP/1.1  
Host: www.safebank.com
```



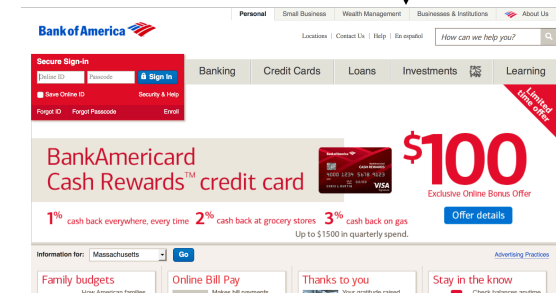
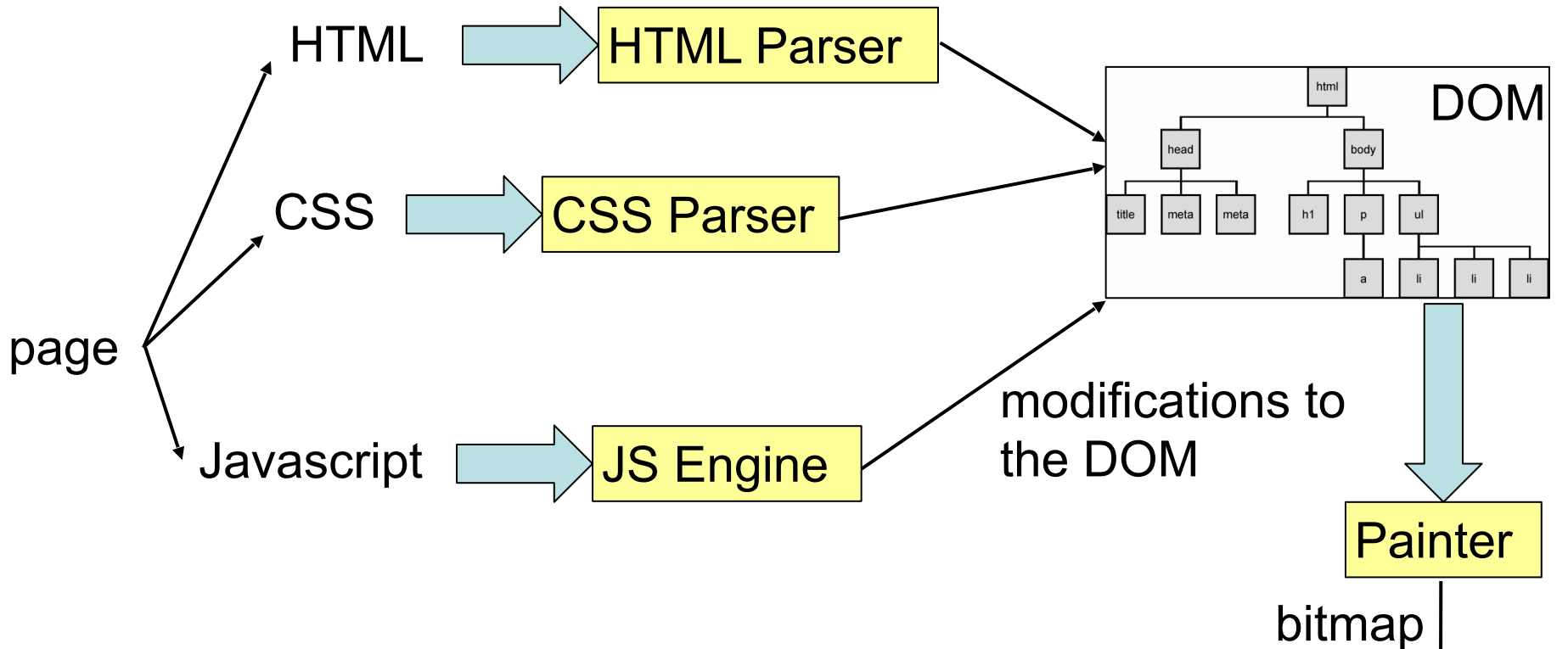
HTTP RESPONSE:

```
HTTP/1.0 200 OK  
<HTML> . . . </HTML>
```

webpage



Page rendering

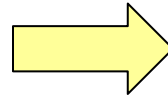


DOM (Document Object Model)

a cross-platform model for representing and interacting with objects in HTML

HTML

```
<html>
  <body>
    <div>
      foo
    </div>
    <form>
      <input type="text" />
      <input type="radio" />
      <input type="checkbox" /
    >
  </form>
</body>
</html>
```

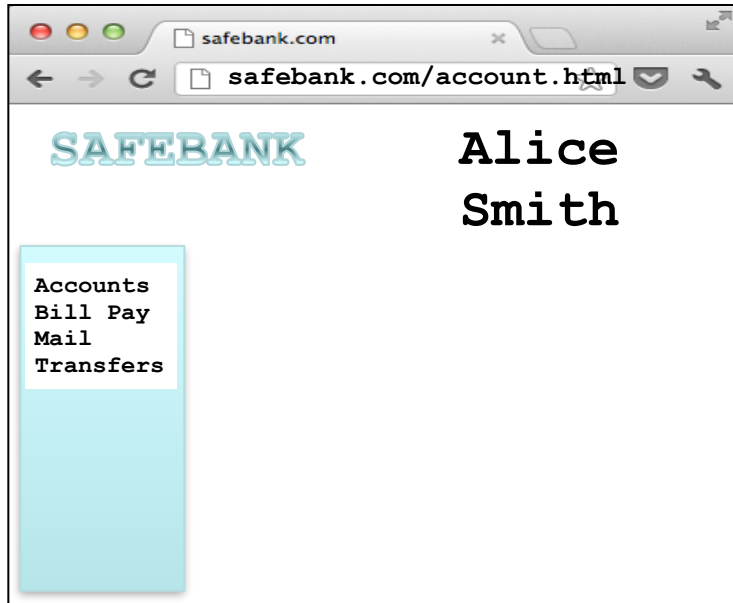


DOM Tree

```
|-> Document
  |-> Element (<html>)
    |-> Element (<body>)
      |-> Element (<div>)
        |-> text node
      |-> Form
        |-> Text-box
        |-> Radio Button
        |-> Check Box
```

Web & HTTP 101

CLIENT BROWSER



WEB SERVER

HTTP REQUEST:

```
GET /account.html HTTP/1.1  
Host: www.safebank.com
```



HTTP RESPONSE:

```
HTTP/1.0 200 OK  
<HTML> . . . </HTML>
```



The power of Javascript

Get familiarized with it so that you can think of all the attacks one can do with it

What can you do with Javascript?

Almost anything you want to the DOM!

A JS script embedded on a page can modify in almost arbitrary ways the DOM of the page. The same happens if an attacker manages to get you load a script into your page.

w3schools.com has nice interactive tutorials:

<https://www.w3schools.com/w3css/tryit.asp>

Example of what Javascript can do...

Can change HTML content:

```
<p id="demo">JavaScript can change HTML content.</p>
```

```
<button type="button"  
onclick="document.getElementById('demo').innerHTML =  
'Hello JavaScript!'">  
  Click Me!</button>
```

DEMO from w3schools.com

Other examples

Can change images

Can change style of elements

Can hide elements

Can unhide elements

Can change cursor

Other example: can access cookies

Will learn later that cookies are useful for authentication.

JS can read cookie:

```
var x = document.cookie;
```

Change cookie with JS:

```
document.cookie = "username=John Smith; expires=Thu,  
18 Dec 2013 12:00:00 UTC; path="/;
```

Frames

Frames

- Enable embedding a page within a page

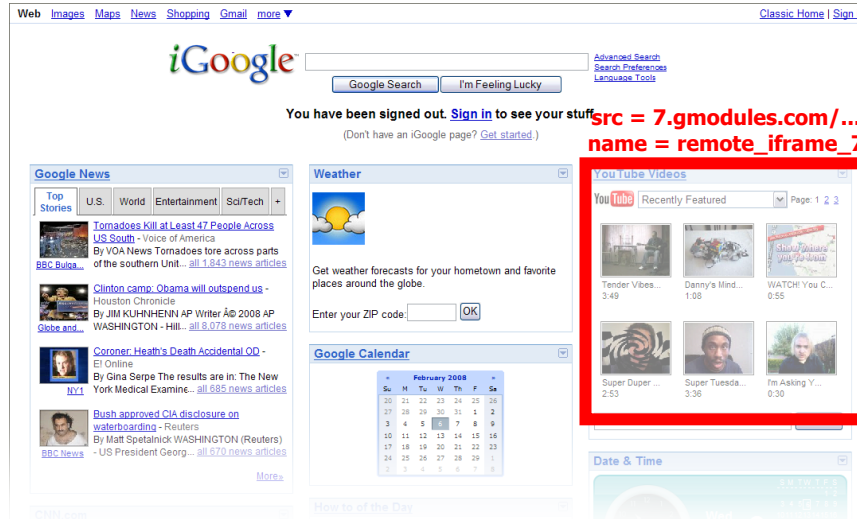
```
<iframe src="URL"></iframe>
```

The screenshot shows the Google AdSense sign-up page. At the top left is the Google AdSense logo. To the right, there is a language dropdown menu set to "English (US)" and a "Help Center" link. Below the logo, the text reads: "Earn money from relevant ads on your website. Google AdSense matches ads to your site's content, and you earn money whenever your visitors click on them." There are two main content areas: a "Sign up now »" button at the top right, and a form for "Existing AdSense users" below it. The form includes fields for "Email:" and "Password:", and a "Sign in" button. A red box highlights the "Sign up now" button and the "Existing AdSense users" form. A blue box highlights a sample advertisement for "Roses, Daisies, and more" from "www.seedsandsaplings.com". A green arrow points to the "Place ads on your site" text at the bottom. A blue arrow points from the "outer page" label to the "Sign up now" button, and another blue arrow points from the "inner page" label to the "Sign in" button.

outer page

inner page

Frames



- Modularity
 - Brings together content from multiple sources
 - Client-side aggregation
- Delegation
 - Frame can draw only on its own rectangle

Frames

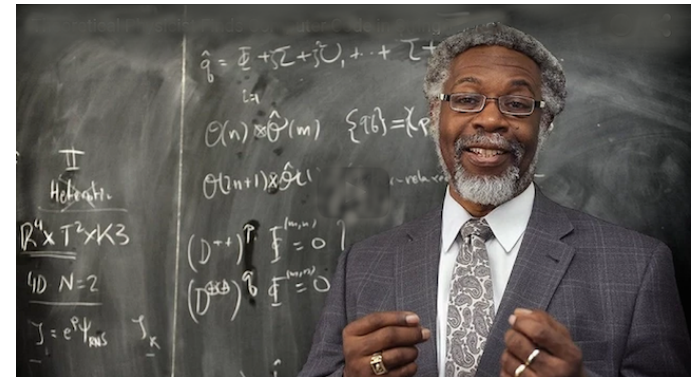
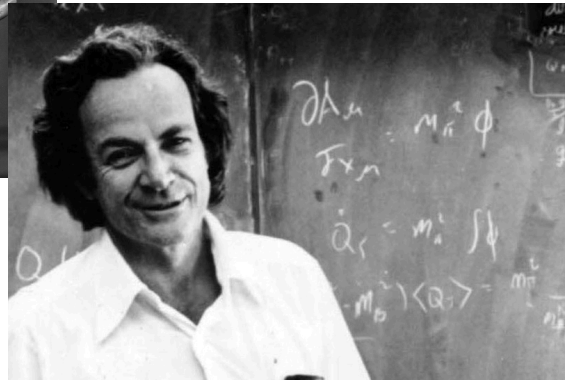
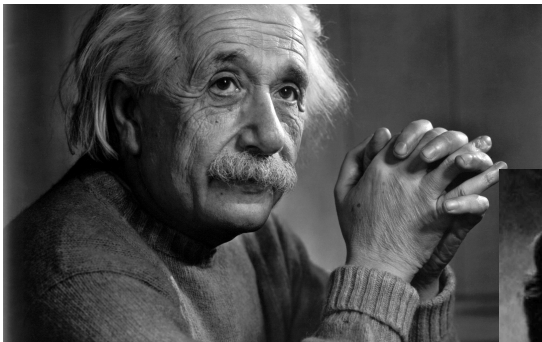
- Outer page can specify only sizing and placement of the frame in the outer page
 - demo
- Frame isolation: Our page cannot change contents of inner page, inner page cannot change contents of outer page

Web security



A historical perspective

- The web is an example of “bolt-on security”
- Originally, the web was invented to allow physicists to share their research papers
 - Only textual web pages + links to other pages;
no threat model to speak of



The web became complex and adversarial quickly

- Then we added embedded images
 - Crucial decision: a page can embed images loaded from another web server
- Then, Javascript, dynamic HTML, AJAX, CSS, frames, audio, video, ...
- Today, a web site is a distributed application
- Attackers have various motivations

Web security is a challenge!

Desirable security goals

- **Integrity:** malicious web sites should not be able to tamper with integrity of my computer or my information on other web sites
- **Confidentiality:** malicious web sites should not be able to learn confidential information from my computer or other web sites
- **Privacy:** malicious web sites should not be able to spy on me or my activities online
- **Availability:** attacker cannot make site unavailable

Security on the web

- Risk #1: we don't want a malicious site to be able to trash my files/programs on my computer
 - Browsing to `awesomevids.com` (or `evil.com`) should not infect my computer with malware, read or write files on my computer, etc.

Security on the web

- Risk #1: we don't want a malicious site to be able to trash my files/programs on my computer
 - Browsing to `awesomevids.com` (or `evil.com`) should not infect my computer with malware, read or write files on my computer, etc.
- Defense: Javascript is sandboxed; try to avoid security bugs in browser code; privilege separation; automatic updates; etc.

Security on the web

- Risk #2: we don't want a malicious site to be able to spy on or tamper with my information or interactions with other websites
 - Browsing to `evil.com` should not let `evil.com` spy on my emails in Gmail or buy stuff with my Amazon account

Security on the web

- Risk #2: we don't want a malicious site to be able to spy on or tamper with my information or interactions with other websites
 - Browsing to `evil.com` should not let `evil.com` spy on my emails in Gmail or buy stuff with my Amazon account
- Defense: **the same-origin policy**
 - A security policy grafted on after-the-fact, and enforced by web browsers

Security on the web

- Risk #3: we want data stored on a web server to be protected from unauthorized access

Security on the web

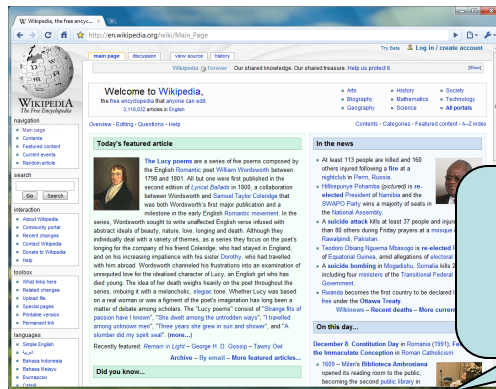
- Risk #3: we want data stored on a web server to be protected from unauthorized access
- Defense: server-side security

Same-origin policy

Same-origin policy

- Each site in the browser is isolated from all others

browser:



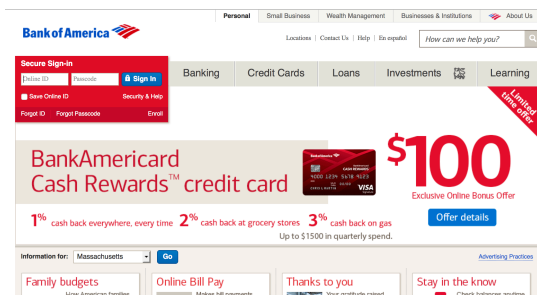
security barrier



wikipedia.org



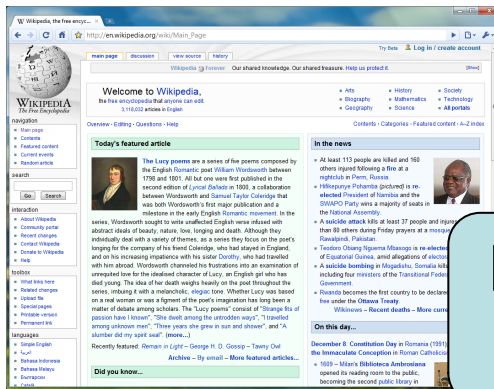
mozilla.org



Same-origin policy

- Multiple pages from the same site are not isolated

browser:



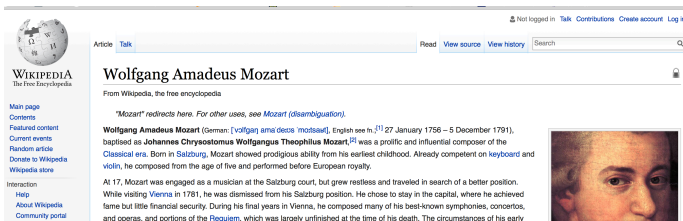
No security barrier



wikipedia.org



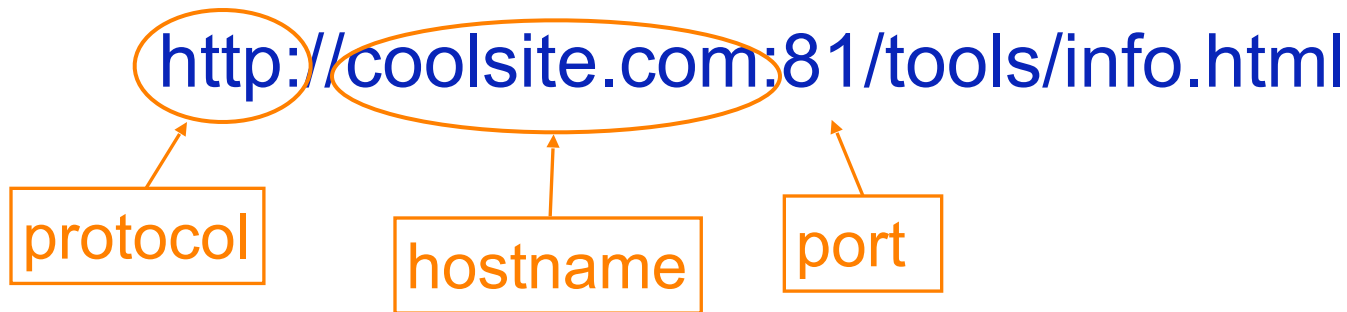
wikipedia.org



No security barrier

Origin

- Granularity of protection for same origin policy
- Origin = (protocol, hostname, port)



- It is **string matching**! If these match, it is same origin, else it is not. Even though in some cases, it is logically the same origin, if there is no match, it is not

Same-origin policy

One origin should not be able to access the resources of another origin

Javascript on one page cannot read or modify pages from different origins

Same-origin policy

- The origin of a page is derived from the URL it was loaded from

<http://en.wikipedia.org>

The screenshot shows a browser window displaying the Wikipedia main page. The address bar contains the URL `http://en.wikipedia.org/wiki/Main_Page`, which is circled in orange. A callout box above the address bar contains the text `http://en.wikipedia.org`. The page content includes the Wikipedia logo, navigation links, a search bar, and featured articles.

Wikipedia, the free encyclopedia

Wikipedia Forever Our shared knowledge. Our shared treasure. Help us protect it.

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
3,118,032 articles in English

Arts History Society
Biography Mathematics Technology
Geography Science All portals

Today's featured article

The *Lucy poems* are a series of five poems composed by the English Romantic poet William Wordsworth between 1798 and 1801. All but one were first published in the second edition of *Lyrical Ballads* in 1800, a collaboration between Wordsworth and Samuel Taylor Coleridge that was both Wordsworth's first major publication and a milestone in the early English Romantic movement. In the series, Wordsworth sought to write unaffected English verse infused with abstract ideals of beauty, nature, love, longing and death. Although they individually deal with a variety of themes, as a series they focus on the poet's longing for the company of his friend Coleridge, who had stayed in England, and on his increasing impatience with his sister Dorothy, who had travelled

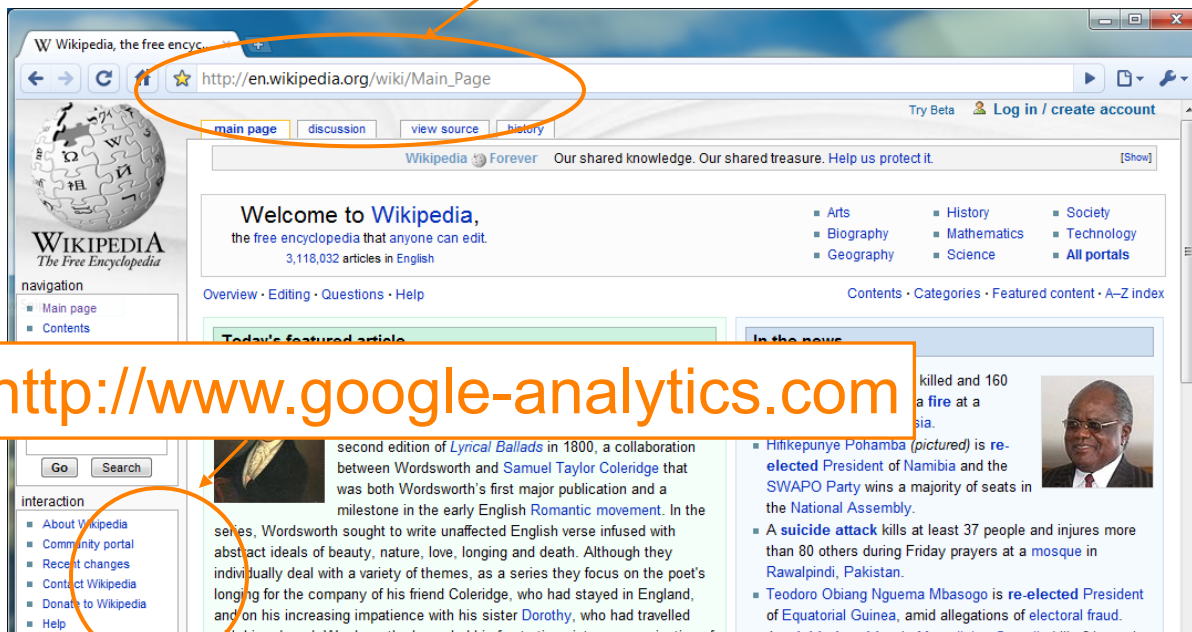
In the news

- At least 113 people are killed and 160 others injured following a fire at a nightclub in Perm, Russia.
- Hifikepunye Pohamba (pictured) is re-elected President of Namibia and the SWAPO Party wins a majority of seats in the National Assembly.
- A suicide attack kills at least 37 people and injures more than 80 others during Friday prayers at a mosque in Rawalpindi, Pakistan.
- Teodoro Obiang Nguema Mbasogo is re-elected President of Equatorial Guinea, amid allegations of electoral fraud.

Same-origin policy

- The origin of a page is derived from the URL it was loaded from
- Special case: Javascript runs with the origin of the page that loaded it

<http://en.wikipedia.org>




Origins of other components

- `` the image is “copied” from the remote server into the new page so it has the origin of the embedding page (like JS) and not of the remote origin

<http://en.wikipedia.org>

Image still has <http://en.wikipedia.org> origin



Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Main_Page

main page discussion view source history

Wikipedia Forever Our shared knowledge. Our shared treasure. Help us protect it.

Welcome to Wikipedia,
the free encyclopedia that anyone can edit.
3,118,032 articles in English

Arts
Biography
Geography

http://upload.wikimedia.org

Today's featured article

The *Lucy poems* are a series of five poems composed by the English Romantic poet William Wordsworth between 1798 and 1801. All but one were first published in the second edition of *Lyrical Ballads* in 1800, a collaboration between Wordsworth and Samuel Taylor Coleridge that was both Wordsworth's first major publication and a milestone in the early English Romantic movement. In the series, Wordsworth sought to write unaffected English verse infused with abstract ideals of beauty, nature, love, longing and death. Although they individually deal with a variety of themes, as a series they focus on the poet's longing for the company of his friend Coleridge, who had stayed in England, and on his increasing impatience with his sister Dorothy, who had travelled

In the news

- At least 113 people are killed and 160 others injured following a fire at a nightclub in Perm, Russia.
- Hifikepunye Pohamba (pictured) is re-elected President of Namibia and the SWAPO Party wins a majority of seats in the National Assembly.
- A suicide attack kills at least 37 people and injures more than 80 others during Friday prayers at a mosque in Rawalpindi, Pakistan.
- Teodoro Obiang Nguema Mbasogo is re-elected President of Equatorial Guinea, amid allegations of electoral fraud.

Origins of other components

- **iframe:** origin of the URL from which the iframe is served, and not the loading website.

Exercises: Same origin?

Originating document	Accessed document
<code>http://wikipedia.org/a/</code>	<code>http://wikipedia.org/b/</code>
<code>http://wikipedia.org/</code>	<code>http://www.wikipedia.org/</code>
<code>http://wikipedia.org/</code>	<code>https://wikipedia.org/</code>
<code>http://wikipedia.org:81/</code>	<code>http://wikipedia.org:82/</code>
<code>http://wikipedia.org:81/</code>	<code>http://wikipedia.org/</code>



except

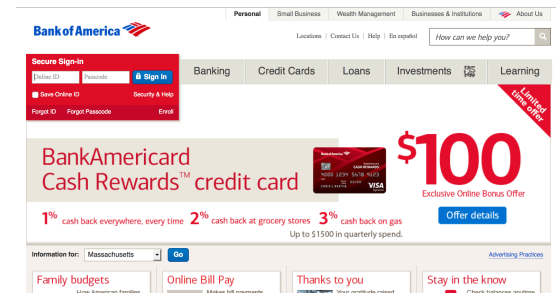


Cross-origin communication

- Allowed through a narrow API: **postMessage**
- Receiving origin decides if to accept the message based on origin (whose correctness is enforced by browser)



`postMessage`
("run this script",
script)



Check origin, and request!