

Introduction to HTTP

Jonathan Sillito

If you interested in working with a professor next Summer 2011 apply for an **NSERC Undergraduate Student Award**.

- Students must have a GPA of 3.0 or higher to be eligible. (Foreign students are not eligible)
- Value of the NSERC USRA is \$4,500 + \$1,125 (minimum)
- Students complete part I and supervisors complete part II, forms must be completed on line
- Deadline: completed applications (part I & II and sealed official transcripts) to Susan Lucas by the end of the day on February 8, 2011.
- Application forms and details are at **http://www.nserc-crsng.gc.ca/index_eng.asp**
- Questions are to be directed to Susan Lucas by email at susan@cpsc.ucalgary.ca or by phone at 220-6092

Hypertext Transfer Protocol

HTTP is the foundation of data communication for the World Wide Web

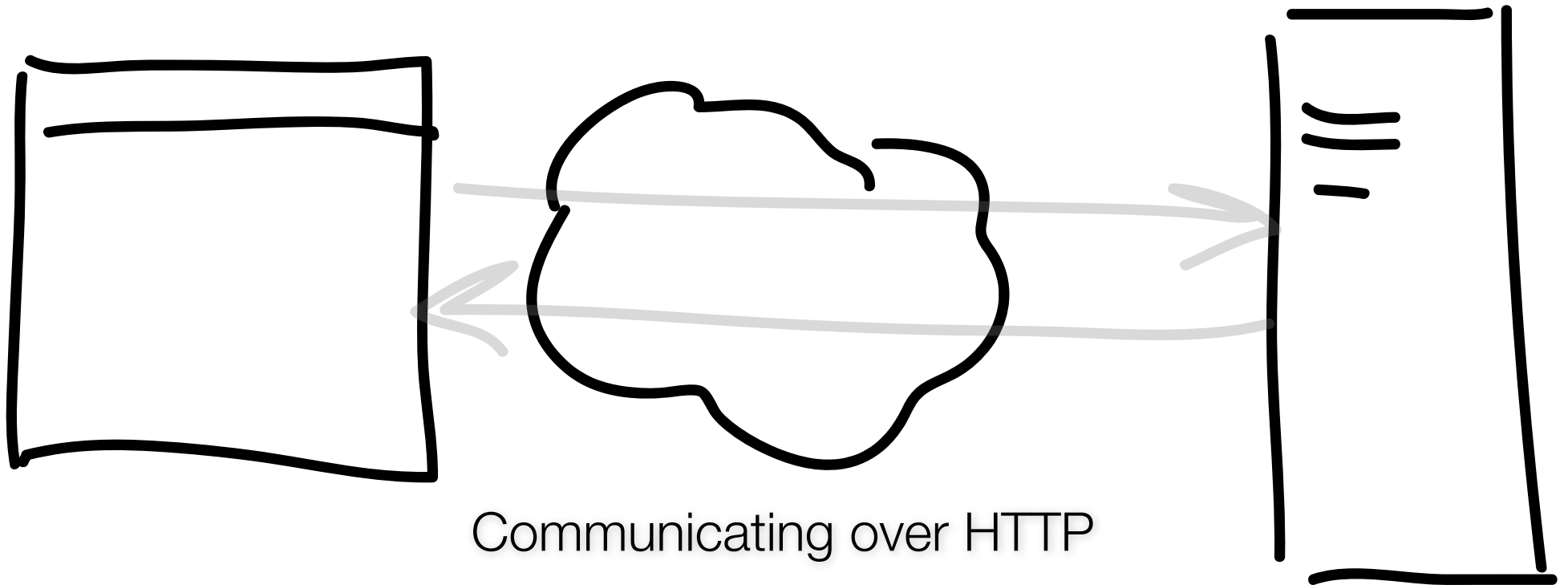
Standardized by the Internet Engineering Task Force and the World Wide Web Consortium

HTTP/1.1 is the version of HTTP in common use today

A **request-response** protocol

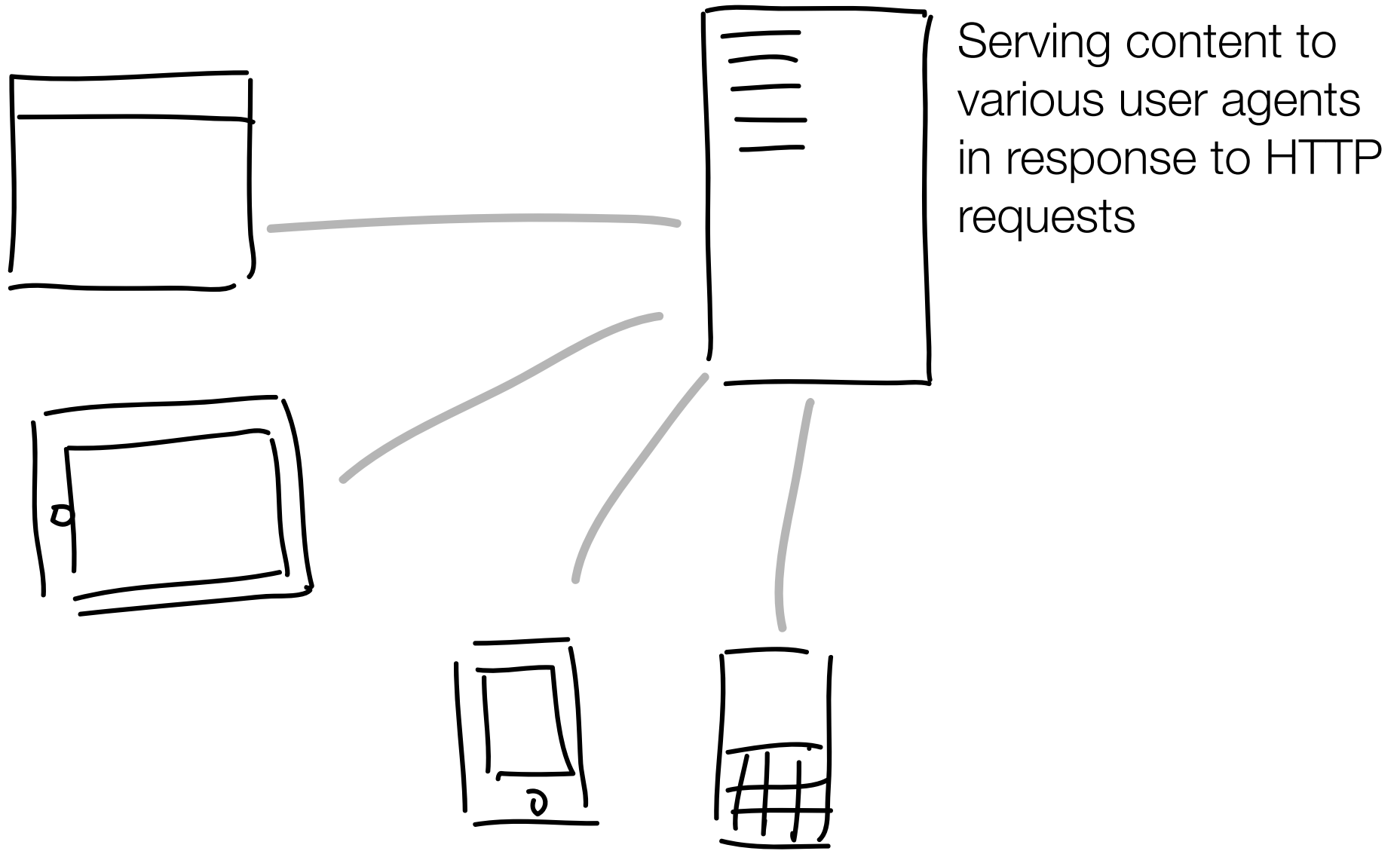
Browser rendering HTML (etc)

Web server serving HTML content



Here the web browser is a client in a **client-server** sense

Other kinds of clients and content



A **request**-response protocol

HTTP requests are sent to a **server** from a client (**user agent**)

- E.g., when a link being clicked in the browser
- A web cache may handle a request in place of the **origin server**, but we'll ignore caching for now.

A **Uniform Resource Locator** (URL) conceptually identifies a resource located on the network

- <http://www.google.com/>

A **request** consists of ...

A request line which includes the *method**, the *URL* and an *HTTP version number*

- GET /pub/WWW/TheProject.html HTTP/1.1
- Most common methods: GET, POST, & DELETE. As the names imply some methods are intended to be side-effect free while others may modify a resource on the server

Some number of headers followed by an empty line

- HOST: www.w3.org

An optional message body (e.g., when there is a post)

* called verbs

A request-**response** protocol

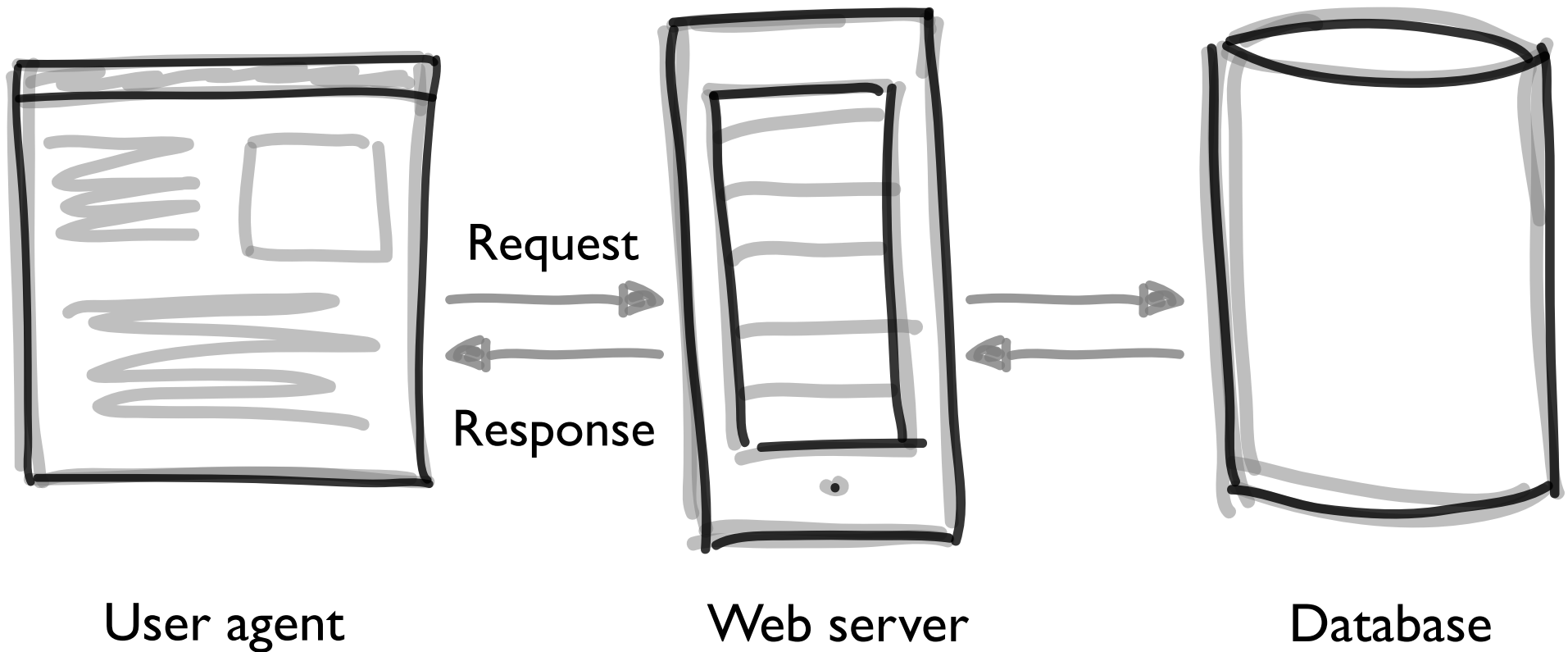
In response to a request from a client a server takes some action on behalf of the client and returns a response

- Returning a static resource (e.g., an image)
- Storing some information in a database
- Dynamically generating a webpage

Note that HTTP is stateless meaning the protocol does not require the server to retain information across multiple requests

- Cookies or hidden variables can be used as a way around this (more later)

Aside: A **3 tier** architecture



Simple versions of this kind of system are sometimes called **CRUD** applications(Create, Read, Update and Delete)

A **response** consists of ...

A status line including a numeric status code and the textual reason. Here are some example status codes and their text

- 404 Not Found
- 200 OK
- 301 Moved Permanently

Headers followed by a blank line

- Content-Type: text/html; charset=UTF-8

The body of the request (e.g., the HTML of a generated page)

In Node servers are single threaded,
event-based and support non-blocking IO

```
var http = require('http'),
    util = require('util');

http.createServer(function (req, res) {
  util.log(req.method + ' ' + req.url);
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World');

}).listen(8124);

util.log('Server running at http://127.0.0.1:8124/');
```

Cookies for saving state between requests

A cookie is:

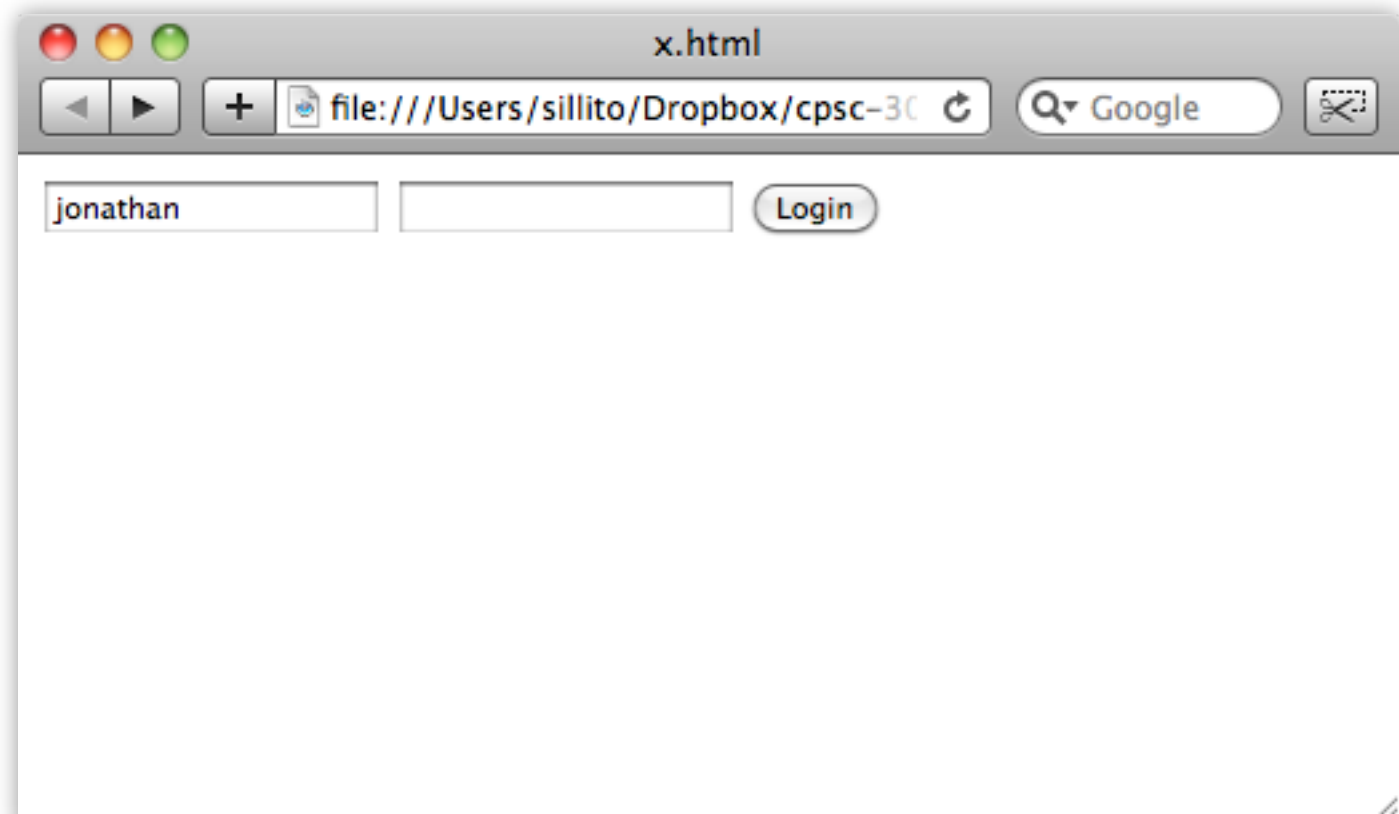
- A piece of text stored on by a user agent (i.e., on the user's computer).
- One or more name-value pairs containing bits of information
- Set as part of a response from a web server (in the header of the response)
 - Set-Cookie: name=value
- Returned by the browser with each subsequent request to that server (in the header of the response)
 - Cookie: name=value

Sample GET request

```
GET / 1.1
host: localhost:8124
connection: keep-alive
accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
user-agent: Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_6_6; en-US) AppleWebKit/534.10 (KHTML, like Gecko) Chrome/8.0.552.237 Safari/534.10,
accept-encoding: gzip,deflate,sdch,
accept-language: en-US,en;q=0.8,
accept-charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
```

Simple HTML login form

```
<html>
<body>
  <form action="http://localhost:8124/" method="post">
    <input type="text" name="login"/>
    <input type="password" name="password"/>
    <input type="submit" name="submit" value="Login"/>
  </form>
</body>
</html>
```



Sample POST request

```
POST / 1.1
host: localhost:8124
origin: file://
user-agent: 'Mozilla/5.0 (Macintosh; U; Intel Mac OS X
10_6_6; en-us) AppleWebKit/533.19.4 (KHTML, like Gecko)
Version/5.0.3 Safari/533.19.4
content-type: application/x-www-form-urlencoded
accept: application/xml,application/xhtml+xml,text/
html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
accept-language: en-us
accept-encoding: gzip, deflate
content-length: 44
connection: keep-alive

login=jonathan&password=cpsc301&submit=Login
```

Sample Redirect response

```
302
location: http://www.google.ca/
cache-control: private
content-type: text/html; charset=UTF-8
set-cookie: some complicated cookie data ...
date: Wed, 19 Jan 2011 19:48:03 GMT
server: gws
content-length: 218
x-xss-protection: 1; mode=block
connection: close

<HTML><HEAD><meta http-equiv="content-type" content="text/
html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.ca/">here</A>.
</BODY></HTML>
```