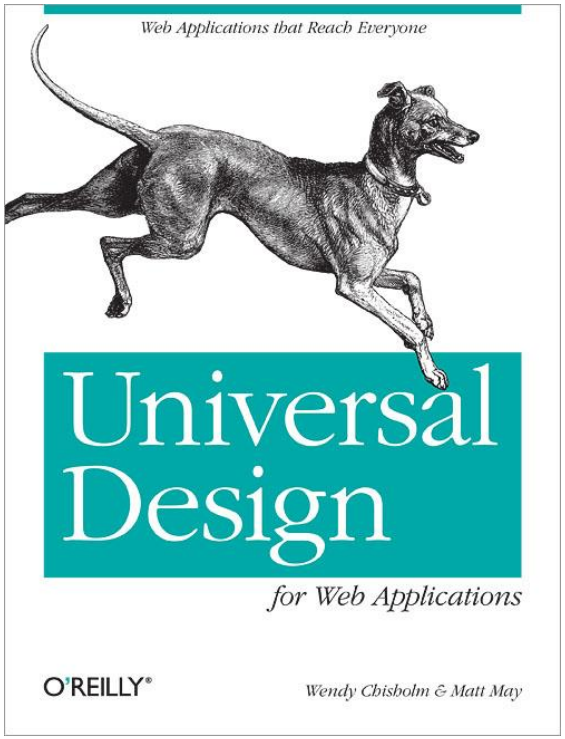
Web Engineering: Universal Design for Web Applications: Google's Approach

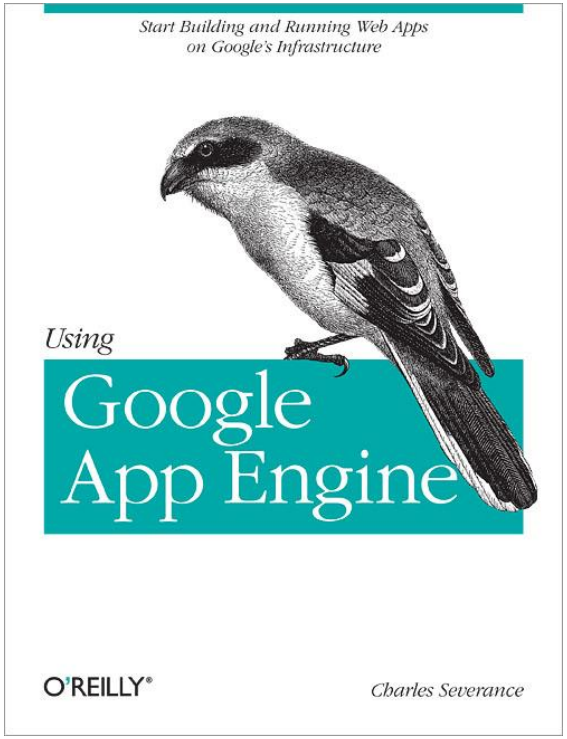
The University of Aizu
Quarter 2, AY 2018

Outline

- Introduction to Universal Design
- Google App Engine
- The App Engine Webapp Framework
- Evaluation Tools and Resourses
- Conclusion

References

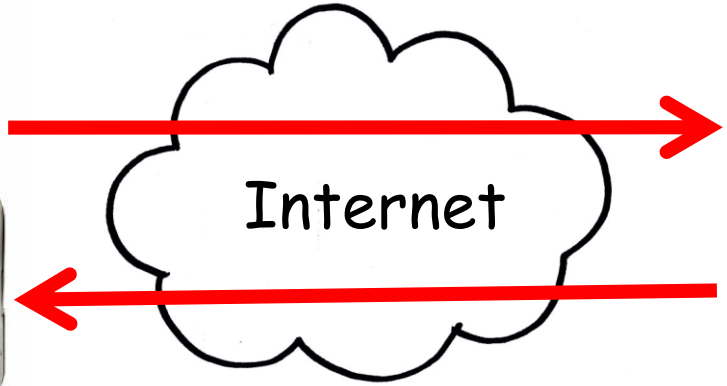

Web Applications that Reach Everyone
Universal Design
for Web Applications
O'REILLY
Wendy Chisholm & Matt May


Start Building and Running Web Apps on Google's Infrastructure
Using
Google App Engine
O'REILLY
Charles Severance

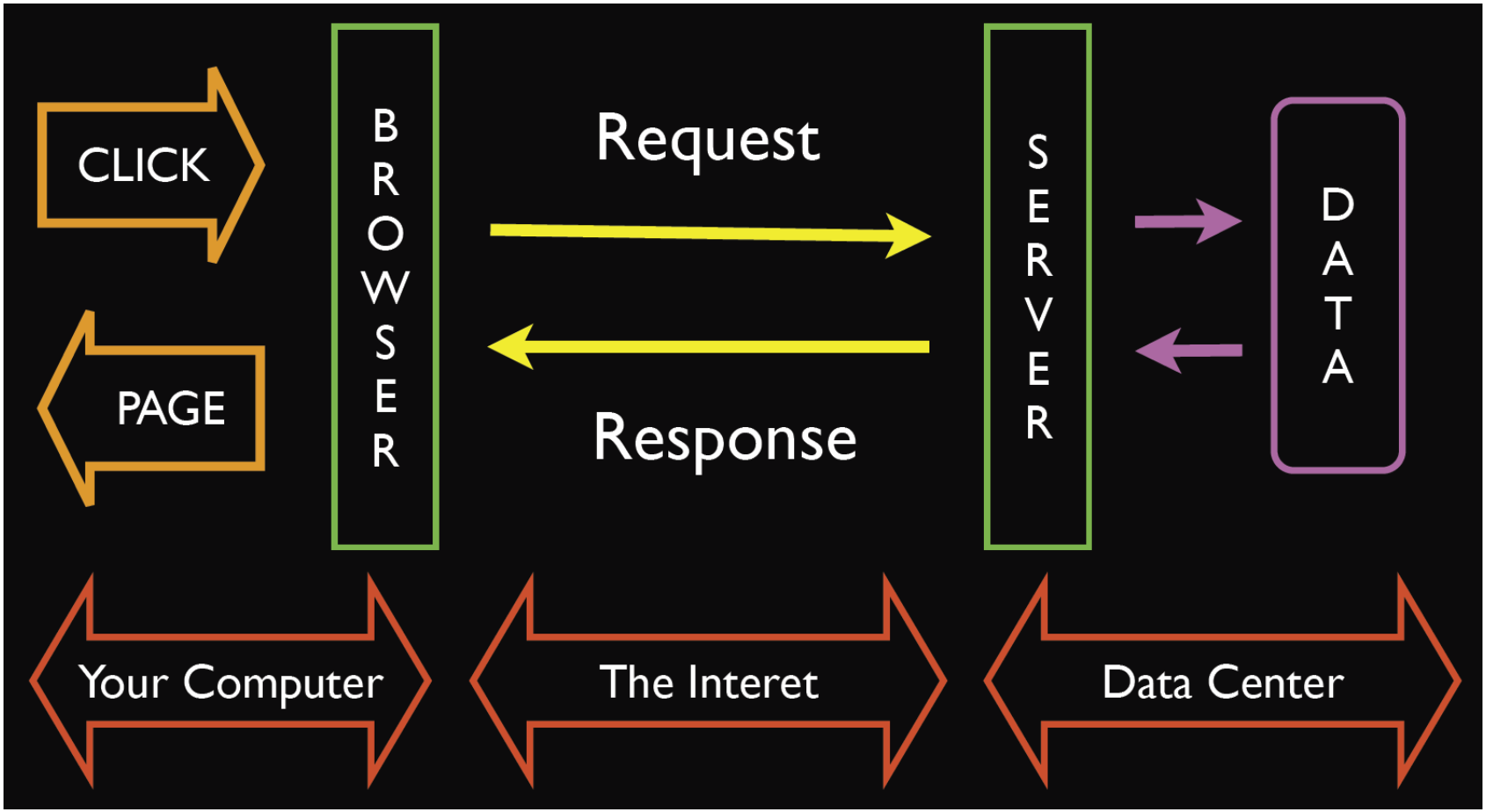
Introduction to Universal Design

- Definition:
  - Universal design is the design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design.
- Illustrations:
  - Mobile devices dominate as the method of accessing Web content.
  - The number of people with disabilities accessing Web content is growing.

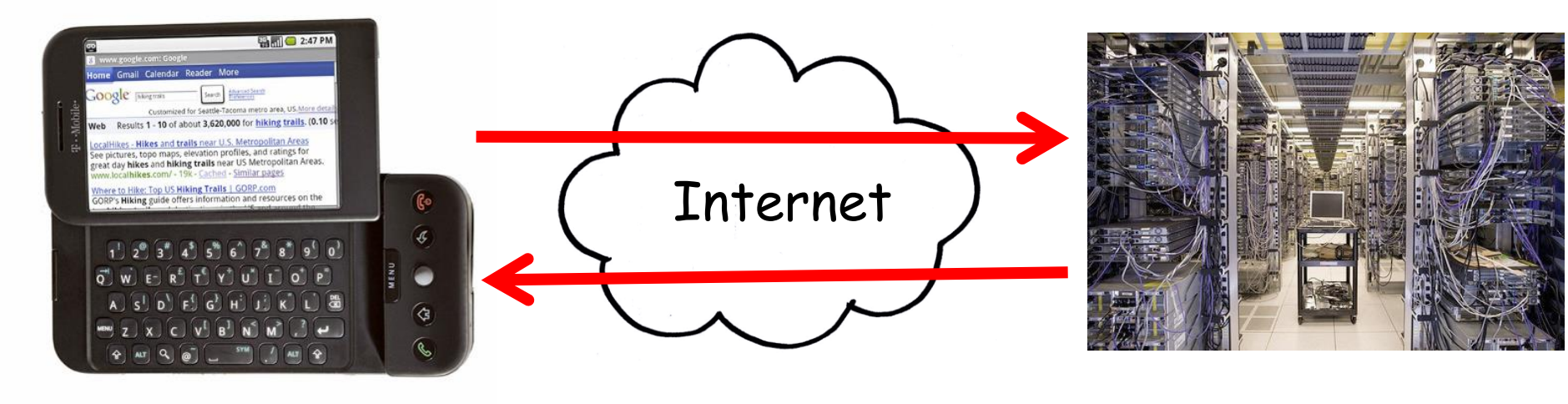
Web Application



Web Application



Web Application



JavaScript

HTML

CSS

AJAX

HTTP

Request

GET

Response

POST

Python

Data Store

Templates

memcache

Pre-cloud Era

- In a pre-cloud view, servers have a geographical location and the users use the Internet to exchange the data with those servers.

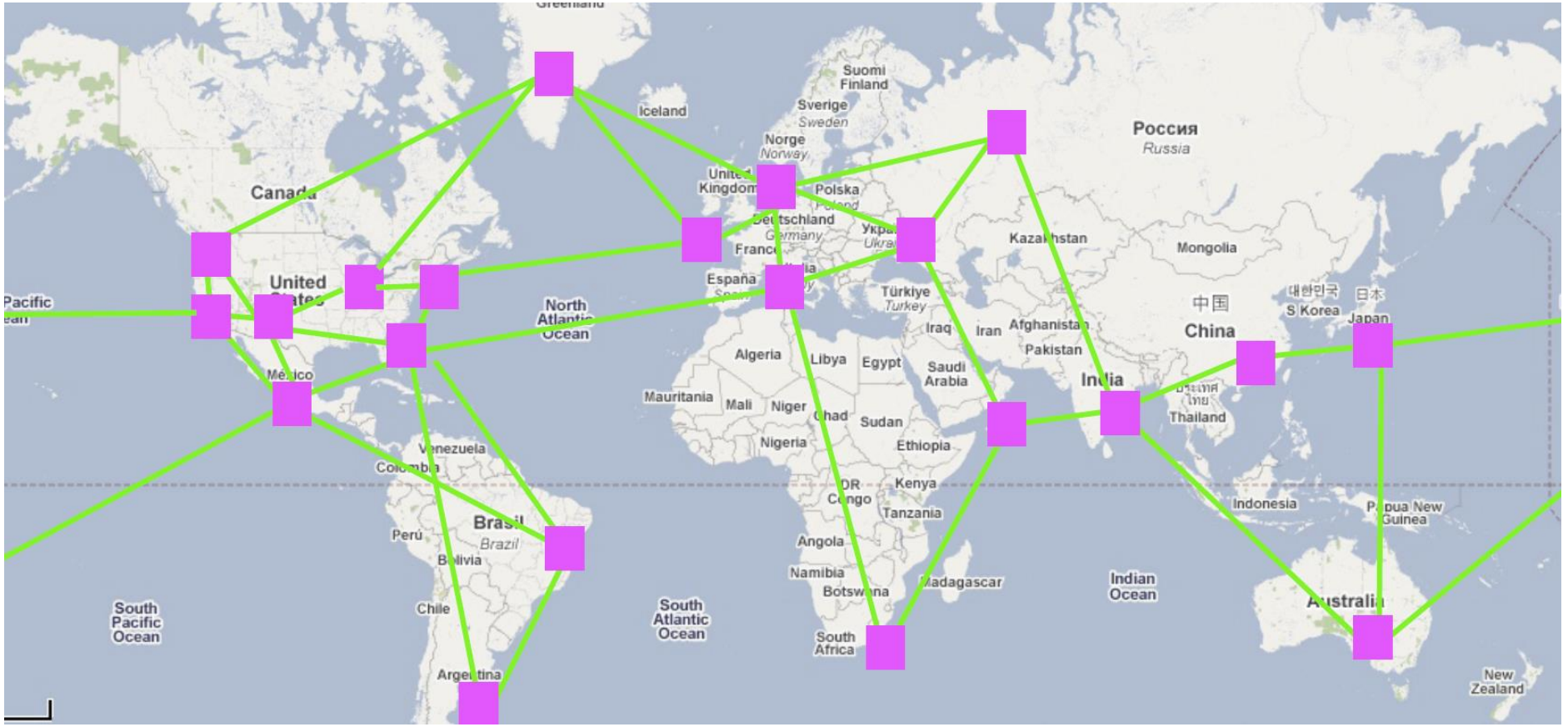
Word-Scale Applications

- For world-scale applications - the servers must be distributed around the world
- But users must see a uniform "single image" - www.google.com
- Also the programmers cannot know the structure or geography of the servers - because this always changes

Programming in the Cloud: Google's View

- Programmers operate in a controlled environment
  - Programs do their programming thing:
    - code + data
  - A complex software framework manages getting the right code and data to/from the right servers.
- Software developers are unaware of geography

Locations of Google Servers, 2008

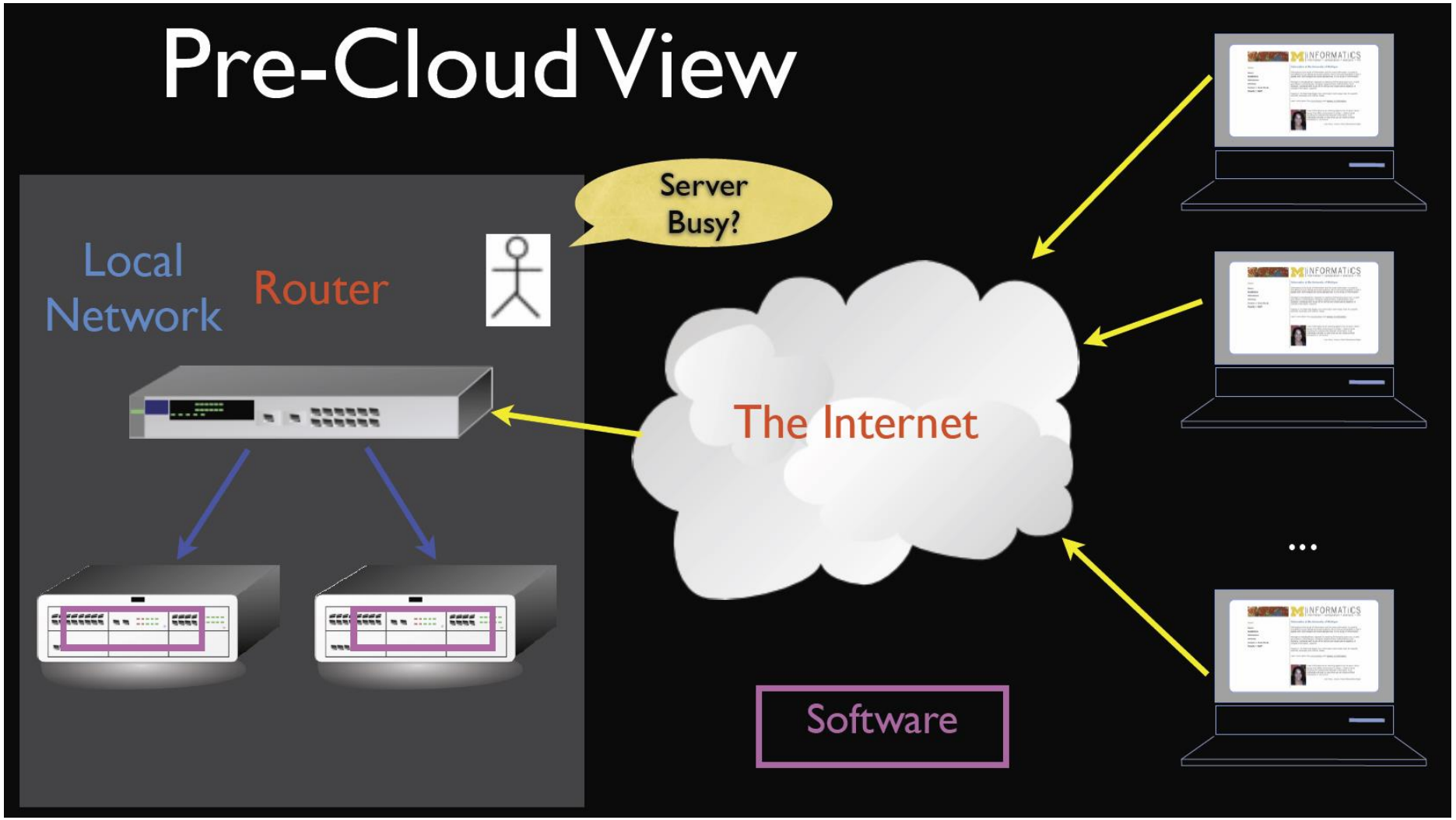


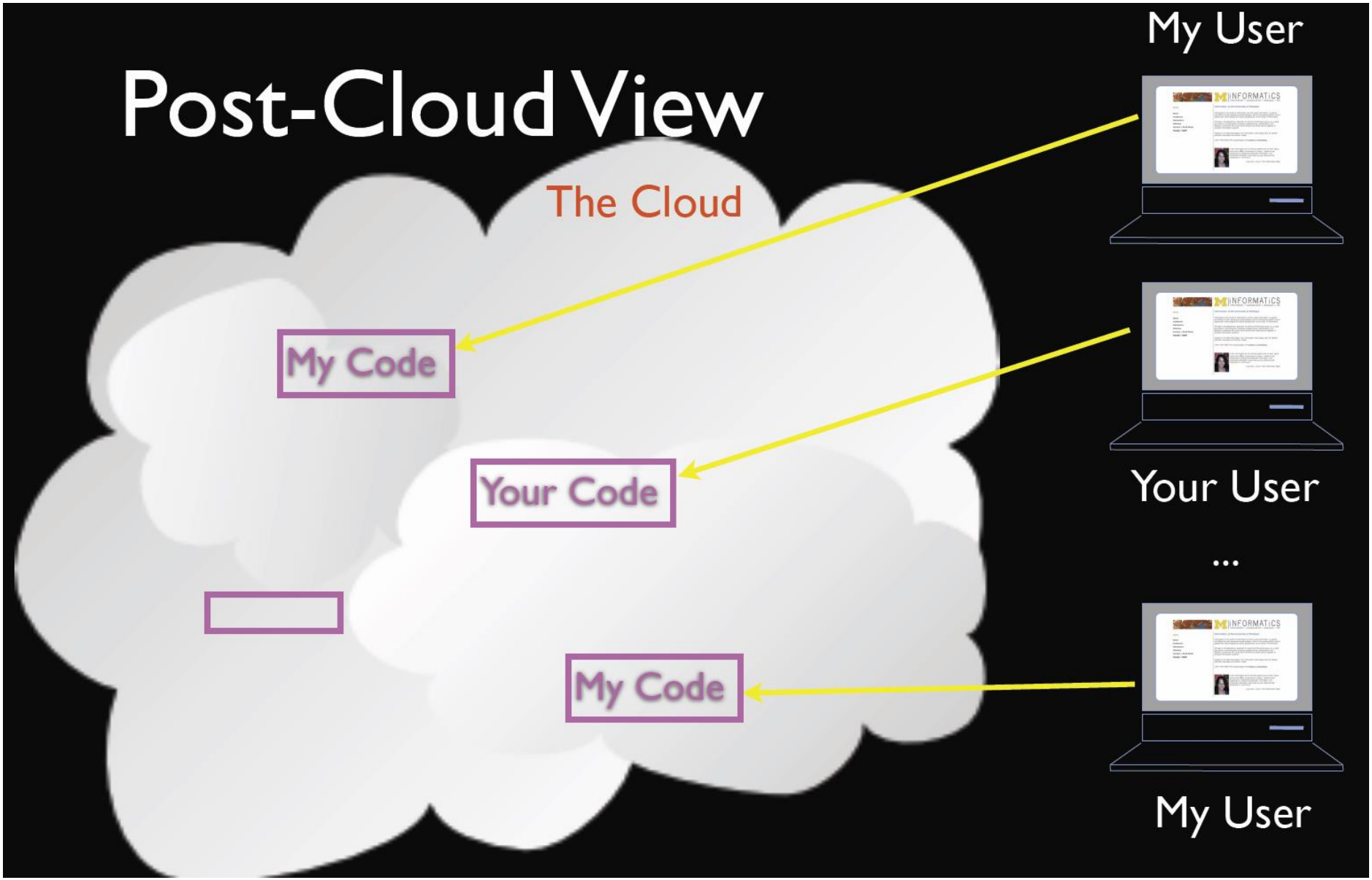
http://royal.pingdom.com/2008/04/11/map-of-all-google-data-center-locations/

http://www.google.com/about/datacenters/locations/index.html

Google's Data Center Locations, 2018

http://www.google.com/about/datacenters/inside/locations/index.html


Pre-Cloud View
Server Busy?
Local Network
Router
The Internet
...
Software


Post-Cloud View
The Cloud
My Code
Your Code
My Code
My User
Your User
...
My User

HTTP Request /Response

- The nature of the HTTP Request/Response cycle makes the cloud possible
- Since clients are not connected for very long - the cloud can be changed in between requests
- As long as the cloud "fakes" everything about the protocol – no one is the wiser.

HTTP Request / Response Cycle

Web Server

GET /index.html

HTTP Request

HTTP Response

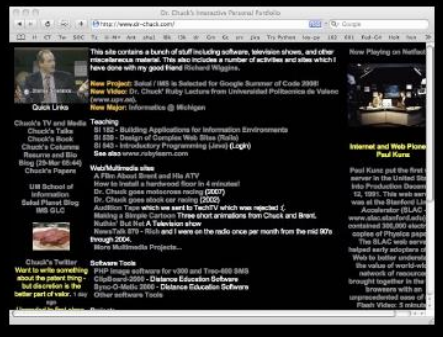
```
<head> .. </head>
<body>
<h1>Welcome to my
  application</h1>
 ....
</body>
```

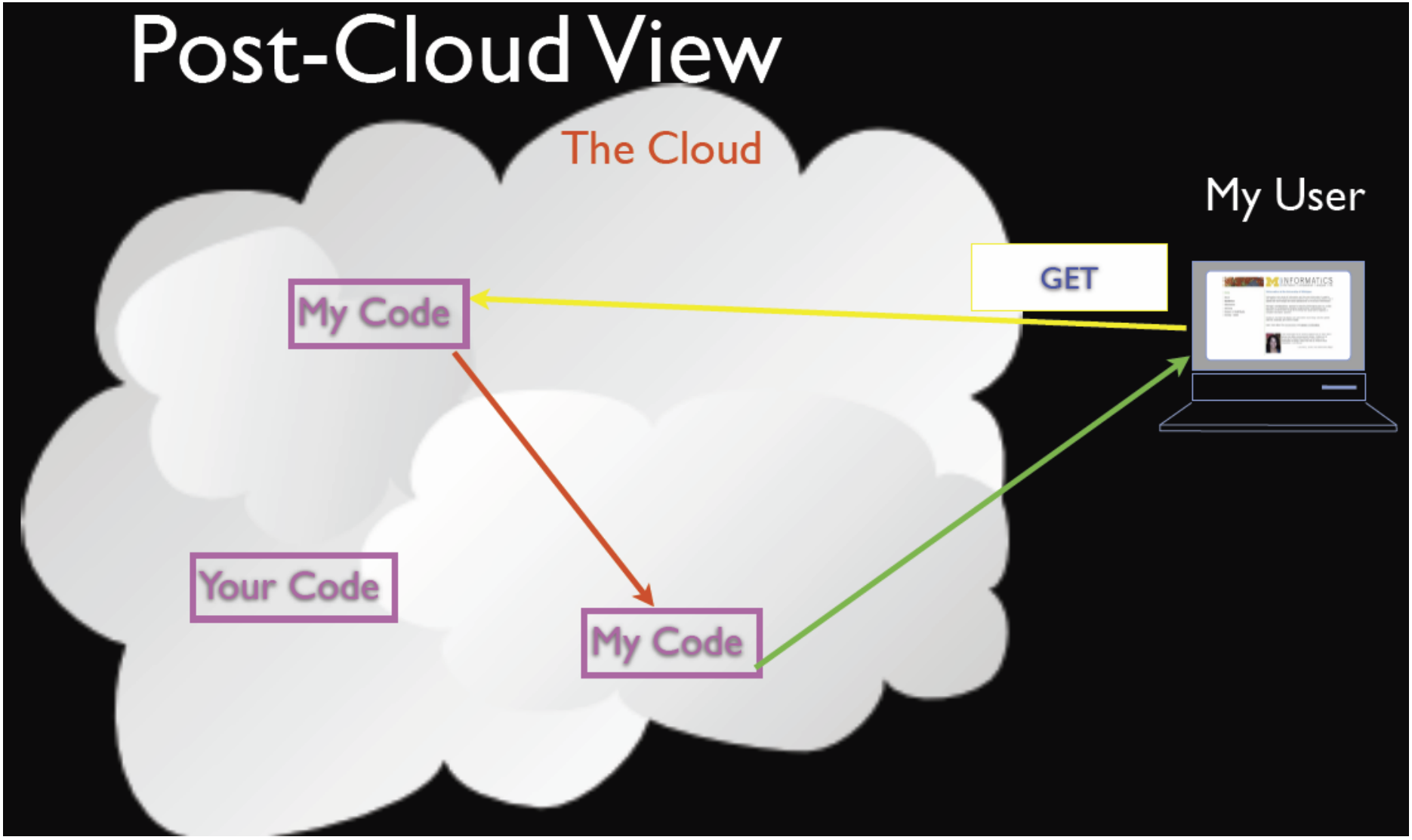
Browser

Hello there my name is Chuck.

Go ahead and click on here.

Internet Explorer, FireFox, Safari, etc.




Post-Cloud View
The Cloud
My User
GET
My Code
Your Code
My Code

Cloud: Summary

- The cloud is the Internet plus computing that is "embedded" "inside" the network
- Companies like Google, Amazon, and Yahoo put servers all over the world
- Software runs on whichever server is most appropriate and data/code is moved around and the cloud can be reconfigured dynamically

Google App Engine

- When you write a Google Application Engine Application - you are running in the Google Cloud
- Just like you were a Google Developer
- You don't know where you are running or if one copy of a thousand copies of you are running
- Google hosts small applications for *free* - larger applications pay by usage

Free Accounts

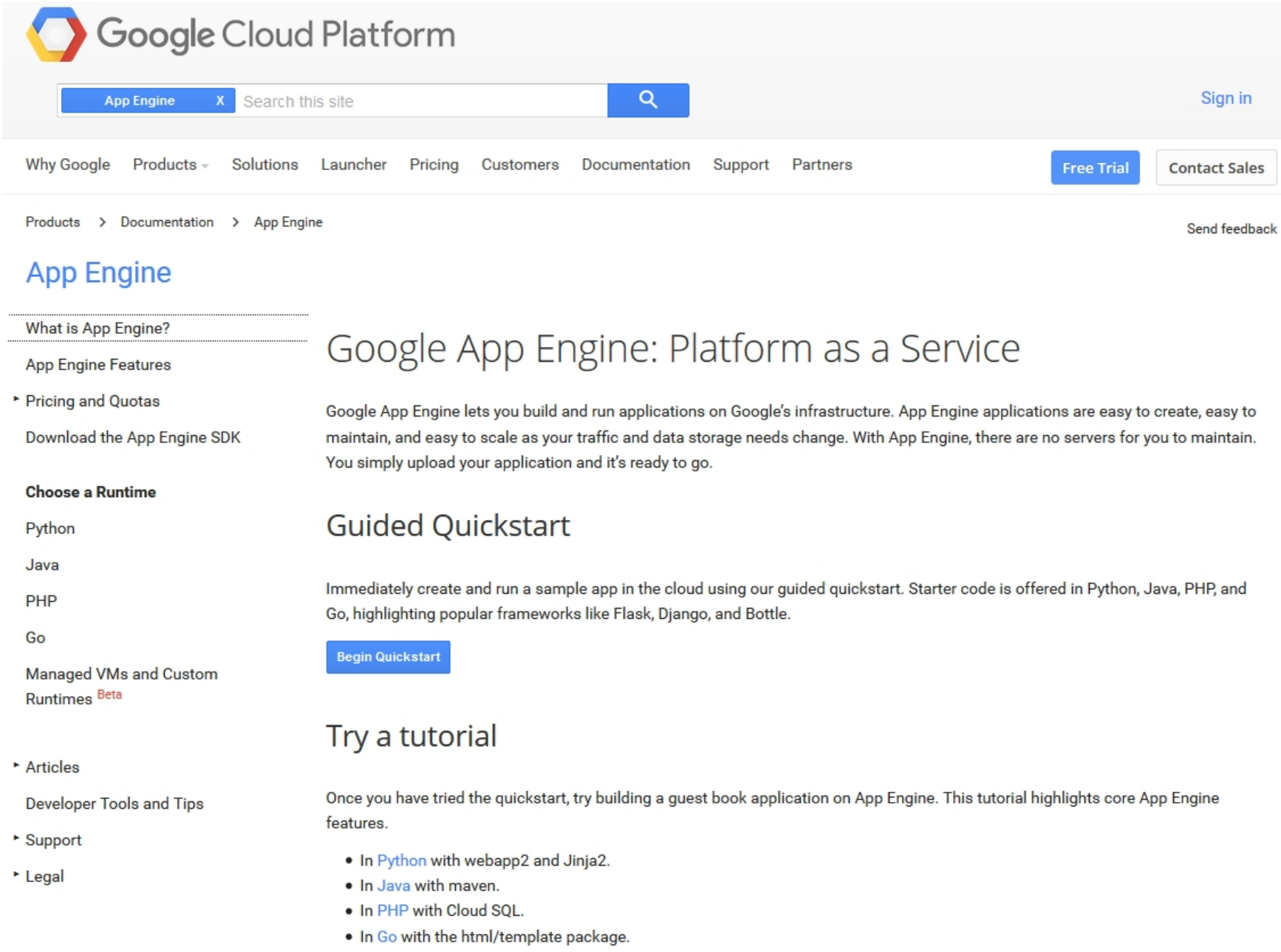
- Limits as of January 2016:

Recourse	Google's Free Limit
Google Cloud Storage	5 GB
Code & Static Data Storage	1 GB
Channel API Calls	3,000 calls/minute
Frontend Instances	28 free instance-hours per day
Mail Service	5000 mails / day

https://cloud.google.com/appengine/docs/quotas?hl=en

The Webapp Framework

- Someone has already written the common code that knows all the details of HTTP (HyperText Transport Protocol)
- We just import it and then use it.



https://cloud.google.com/appengine/docs

Technology: User View

- Only once
  - Download and install the package fr the Google App Engine SDK from: https://cloud.google.com/appengine/downloads
- For every application
  - Create a separate folder, for example
    - MyApp
  - Create at least two files:
    - app.yaml and index.py
  - Start the Google App Engine Web server and run your application
    - /usr/local/bin/dev_appserver.py MyApp

Technology: Example

File: app.yaml

```
version: 1
runtime: python27
api_version: 1
threadsafe: true

handlers:
    - url: /.*
    script: index.py
```

File: index.py

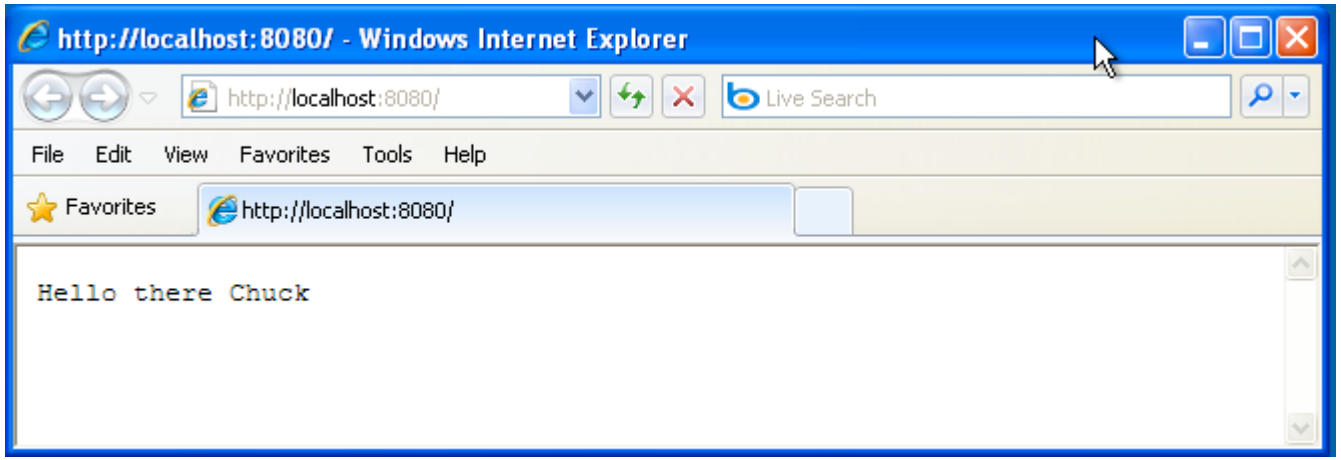
```
import webapp2

class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] =
                                            'text/plain'
        self.response.write('Hello there Chuck')

app = webapp2.WSGIApplication([
    ('/', MainPage),
], debug=True)
```

Technology: Example

Result of execution



Evaluation Tools and Resources

- Every Web application has to be checked from different angles.
- The following tools for this evaluation:
  - The Web Accessibility Toolbar (WAT) http://www.wat-c.org/tools
    - It allows you to determine if the application has issues.
  - Web Accessibility Evaluation Tool (WAVE) http://wave.webaim.org
    - It embeds icons within web page indicating potential issues

Evaluation Tools and Resources

- Firebug https://addons.mozilla.org/en-US/firefox/addon/firebug/?src=search.
  - A tool to debug JavaScripts or CSS
- W3C validation tools:
  - W3C Markup Validation service, http://validator.w3.org
  - W3C CSS Validation Service, http://jigsaw.w3.org/css-validator/
  - W3C mobileOK Checker, http://validator.w3.org/mobile/

Evaluation Tools and Resources

- Browsers: evaluation of accessibility features for your applications:
  - https://www.w3.org/WAI/ER/tools/

Evaluation Tools and Resources

- Operating systems: evaluation of accessibility features for your applications:
  - Mac OSX and iPhone, http://www.apple.com/accessibility/
  - Linux, http://www.linuxfoundation.org/collaborate/workgroups/accessibility
  - Windows, http://www.microsoft.com/enable/products/windows10/default.aspx/
  - Android, http://developer.android.com/guide/topics/ui/accessibility/index.html

20 Questions

- The aforementioned tools help you to figurate, if your application satisfies the principles of universal design
  - Q. 1: Text alternatives
    - Are the text alternatives present and sufficiently equivalent to the graphical, audio and video content?
  - Q. 2: Multimedia
    - Is multimedia captioned?

20 Questions

- Q. 3: Link and control labels
  - Are controls including links appropriately identified or labeled?
- Q. 4: Control groups
  - Are groups or controls appropriately identified?
- Q. 5: Meaningful structure
  - Can a meaningful keyboard navigation order be delivered from the application structure?
- Q. 6: Nonsensory operation
  - Can the application be operated when the color, shape, size, location or sound cannot be perceived?

20 Questions

- Q. 7: Automatic audio
  - Can audio that plays automatically be stopped, paused, or silenced?
- Q. 8: Keyboard-only operation
  - Can all functionality be operated via the keyboard alone?
- Q. 9: Bypass blocks
  - Are blocks of content identified that they can be skipped?
- Q. 10: Page titles
  - Does each page have a unique title that describes its topic or purpose?

20 Questions

- Q. 11: Language
  - Is the human language identified for each page?
- Q. 12: Predictable behavior
  - Is the application's behavior predictable in response to user input?
- Q. 13: Error identification and resolution
  - If the user makes an error, is the error clearly identified and suggestions provided for how to fix it?
- Q. 14: Syntactical and run time errors
  - Do program code in languages execute without errors?

20 Questions

- Q. 15: Change notification
  - Does the application use accessibility features that notify the changes to the user interface components?
- Q. 16: Timed response
  - If a page requires a timed response, can someone turn off or adjust the time limit?
- Q. 17: Moving, blinking, and scrolling
  - If the application has moving, blinking, or scrolling information, is there a way for a person to pause, stop, or hide the content?

20 Questions

- Q. 18: Auto-update
  - If an application auto-updates, is there a way for a person to control the frequency of updates?
- Q. 19: Flashing content
  - Does the page avoid anything that flashes more than three times per second?
- Q. 20: Field testing
  - Has the application been tested with the variety of browsers, mobile devices, and assistive technologies?

Conclusion

- We looked at Google's approach to the universal design for Web applications.
- Google App Engine is a tool to design Web applications and run them in the cloud.
  - Nowadays, this tool is gained popularity
- We looked at the Google's technology from the user and system point of view.
- We paid attention to the tools to evaluate the quality of Web application.