

Web Sites/Applications



theme · content · navigation
· structure · templates ·
style · interaction · busine...

Building a Web Site

- Step 1: Determine Theme + Content
- Step 2: Devise Navigation Structure
- Step 3: Create Page Structure
- Step 4: Factor out Page Structure in (reusable) Templates
- Step 5: Apply a Style
- Step 6: Build, Test & Deploy



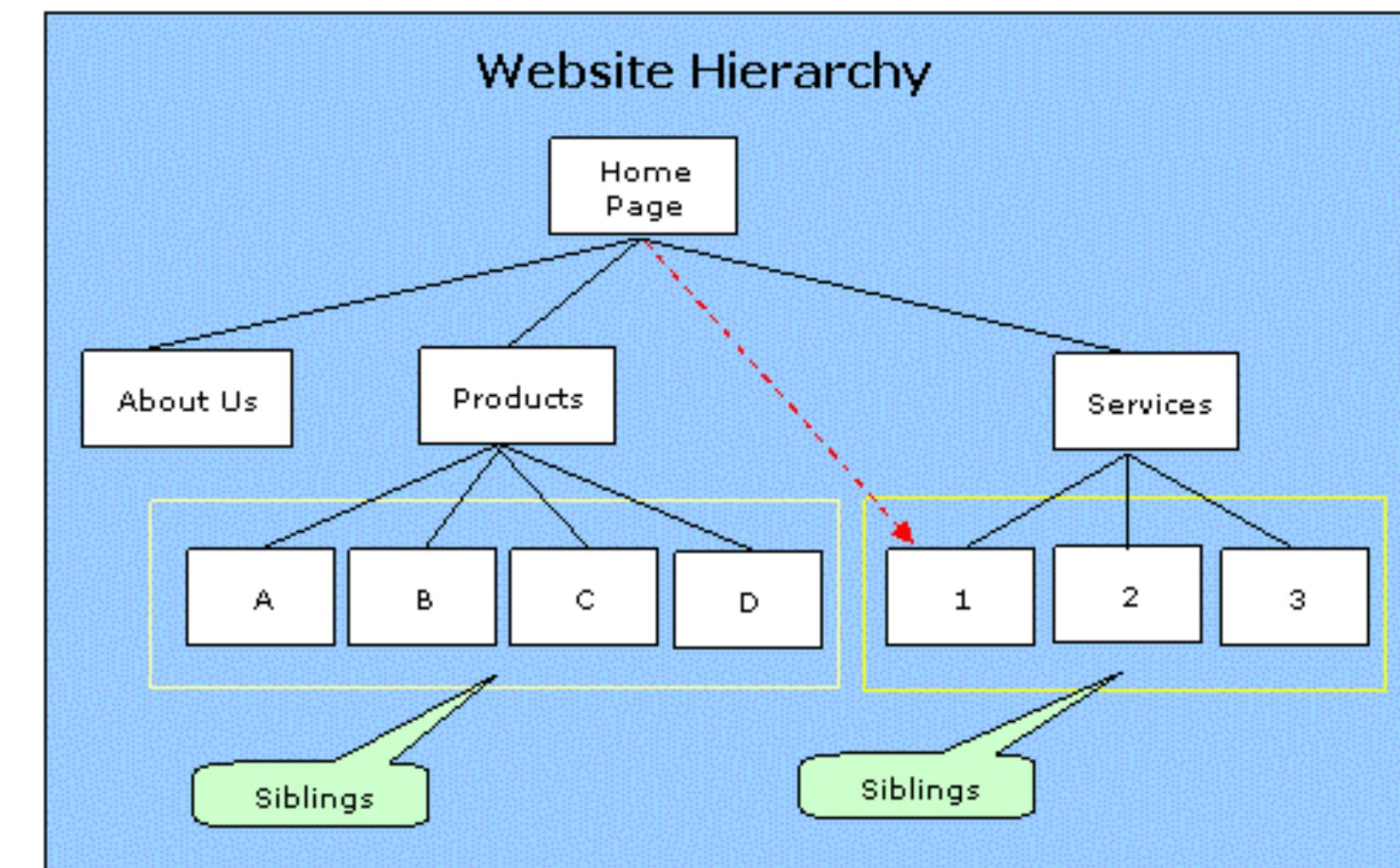
Web Site: Step 1: Determine Theme + Content

- Agree a ‘theme’ and ‘look and feel’ for site with customer
- Acquire or develop the core ‘content’ of the site
 - Text
 - Images
 - Media (video/audio)



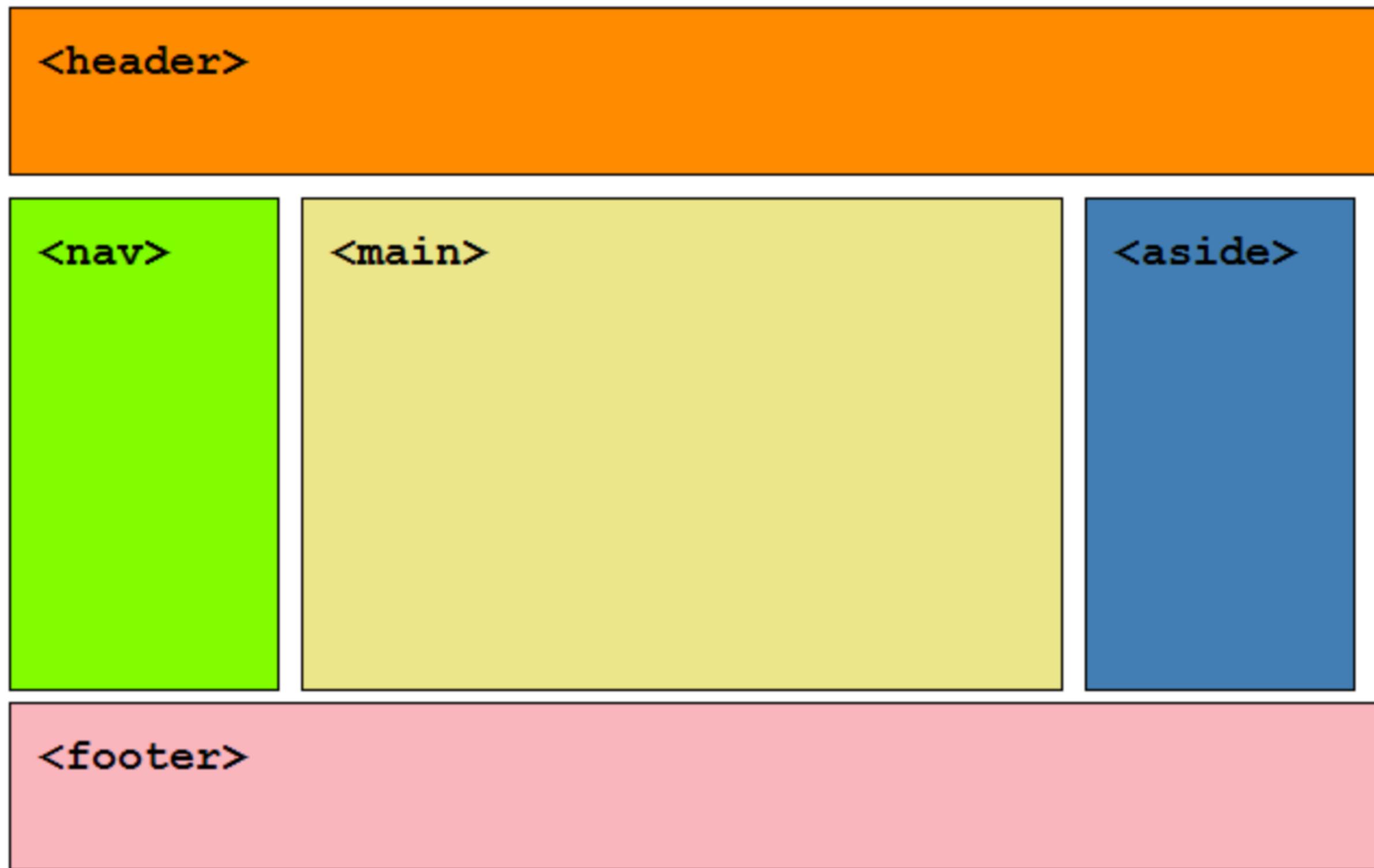
Web Site: Step 2: Determine Navigation Structure

- Determine number of pages in the site
- Decide on navigation ‘metaphor’
 - ‘Tabs’
 - Sidebar
 - Menubar



Web Site: Step 3: Create Page Structure

- Typical Sections:
 - Header
 - Footer
 - Navigation
 - Main Content
 - Primary
 - Secondary



index.html

```
<div id="navigation">
  <ul id="menu" ...>
</div>
<div id="maincontent">
  <div id="primary" ...>
  <div id="secondary" ...>
</div>
```

apps.html

```
<div id="navigation">
  <ul id="menu" ...>
</div>
<div id="maincontent">
  <div id="primary" ...>
</div>
```

directions.html

```
<div id="navigation">
  <ul id="menu" ...>
</div>
<div id="maincontent">
  <div id="primary" ...>
</div>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Bundle APP Store</title>
    <link rel="stylesheet" href=".css/home.css" />
  </head>
  <body>
    
    <div class="container">
      <% include('./partials/_header.ejs') %>
      <% yield %>
      <% include('./partials/_footer.ejs') %>
    </div>
  </body>
</html>
```



_header.html

```
<div id="header">
  <h1>Welcome to the App Bundle Store</h1>
</div>
```

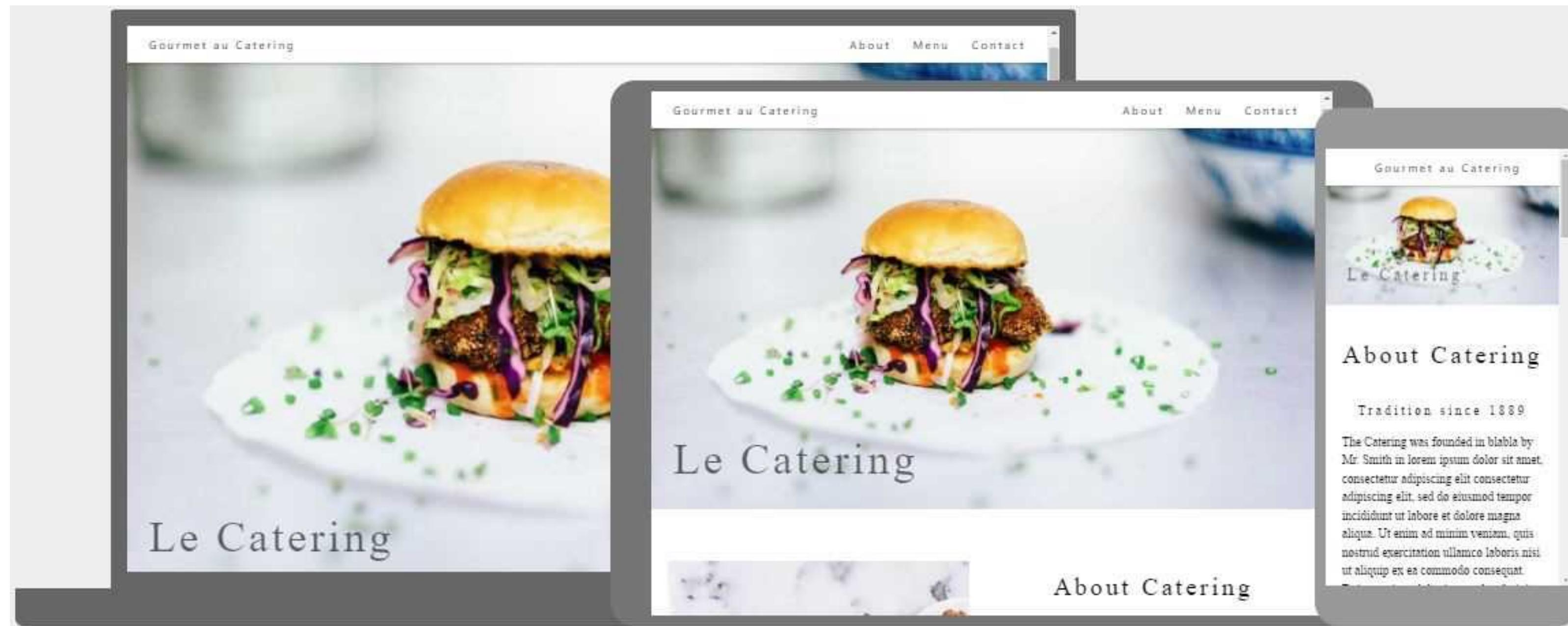
_footer.html

```
<div id="footer">
  <p> Contact us at : bundle@store.co</p>
</div>
```

Step 4: Factor out Page Structure in (reusable) Templates

Web Site: Step 5: Apply a Style

- Compose CSS to capture
 - Navigation
 - Layout : structure, layout, number of columns, positioning
 - Look and Feel (theme)



Web Site: Step 6: Build, Test & Deploy

- Build the site itself
- Verify that all links work as expected
- “Push” the site to an external server.

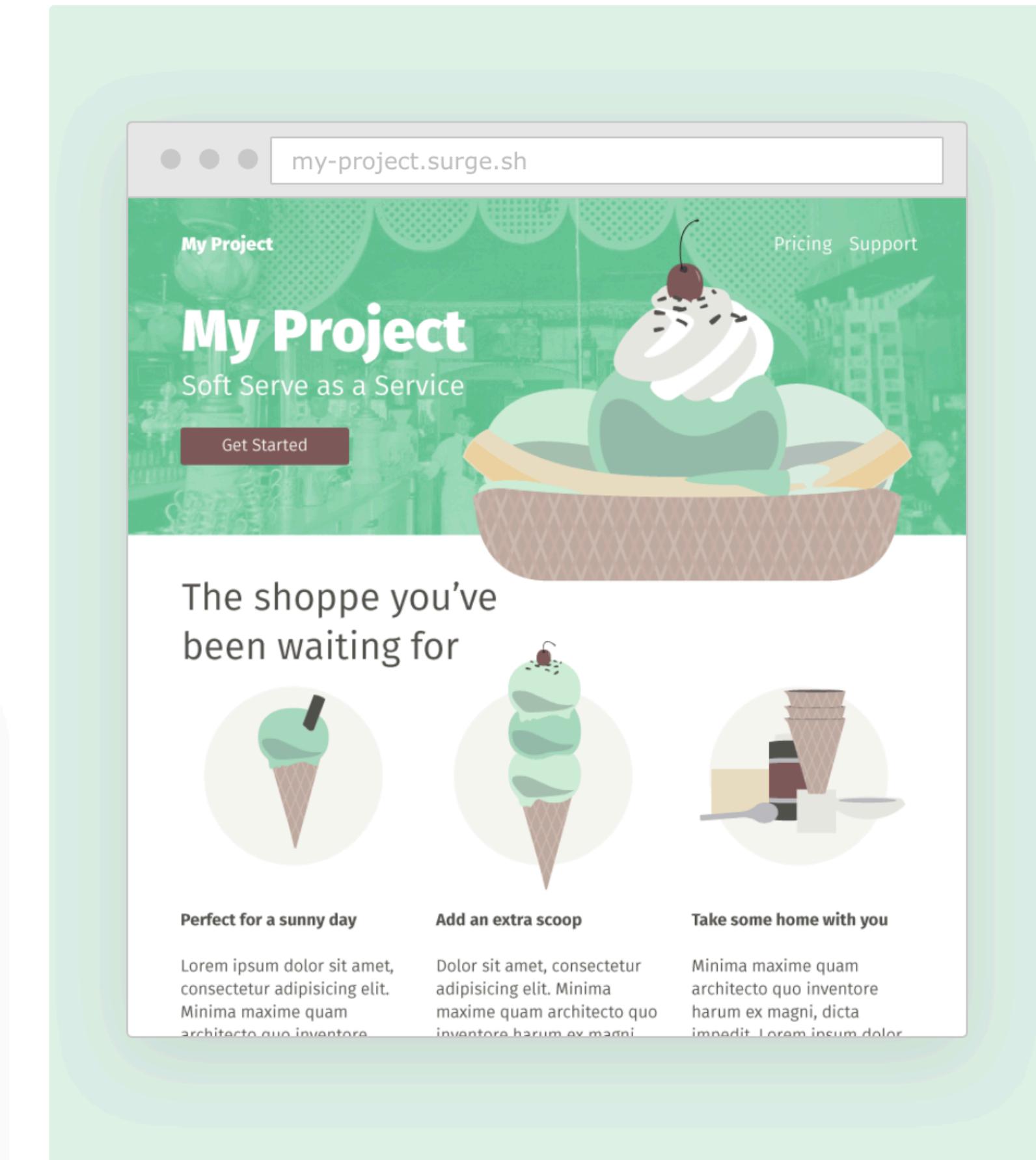
Publish any folder, right from the CLI

Front-end developers like you are better with the command line than ever—take advantage of it. Publish any directory on your machine with a single command: `surge`.

Install the Surge CLI globally through npm, and get any static assets or client-side app online, **for free**.

```
$ npm install --global surge  
$ surge  
  
project: ~/Jane/Desktop/my-project/  
domain: my-project.surge.sh  
upload: [=====]
```

Success! Published and running at `my-project.surge.sh`

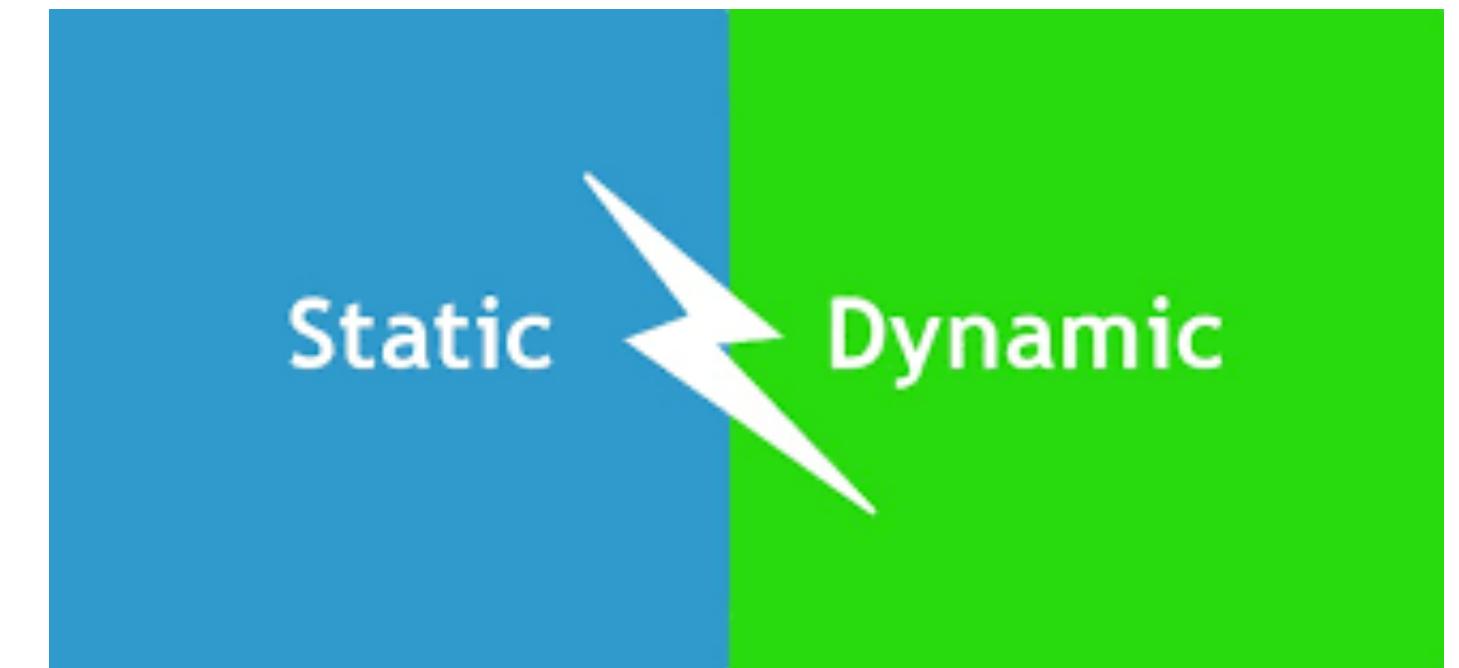


What if...?

- A user is to “Log in” to a site?
- A user needs to supply information to the site?
- The content of some of the pages is not known until the site is ‘live’?
- The content of some pages is very specific to the identity of the current user?
- The site is to implement a ‘business process’ such as
 - shopping cart
 - payment for a good or service
 - communication with other users - such as messaging
- Such features require a ***Dynamic Web Site*** or a ***Web Application***



Static vs Dynamic



- A knowledge of HTML, CSS + simple web deployment is necessary in order to build a *Static Web Site*
- However, these skills are ***not sufficient*** to build a ***Web Application***
- A Web Application is capable of:
 - Responding to user interaction
 - Generating new information based on context
 - Allowing a user to provide information
 - Implement core business processes
- A ***Static Web Site*** is not capable of these features.

Web App Development

- Solid understanding of HTML & CSS, including page structure, layout, styling and approaches to navigation

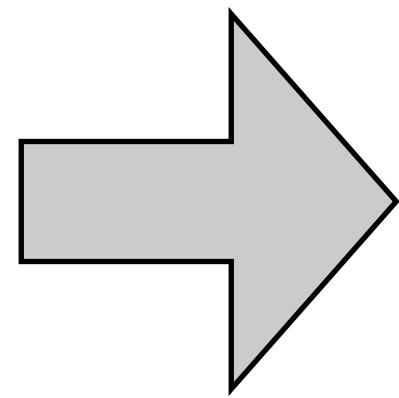
+ knowledge of:

- Structure of the Internet, including role of HTTP, DNS & how URLs are structured
- Detailed understanding of the nature of the HTTP protocol
- Client / Server Architecture
- How pages can be composed of templates
- Databases
- ***How to Programme Application Features (in Java)***



!!!

- Structure of the Internet, including role of DNS & **URLs**
- Nature of the **HTTP protocol**
- **Client / Server** Architecture
- Pages decomposed using **templates**
- **Databases**
- How to **Programme Java** Application Features



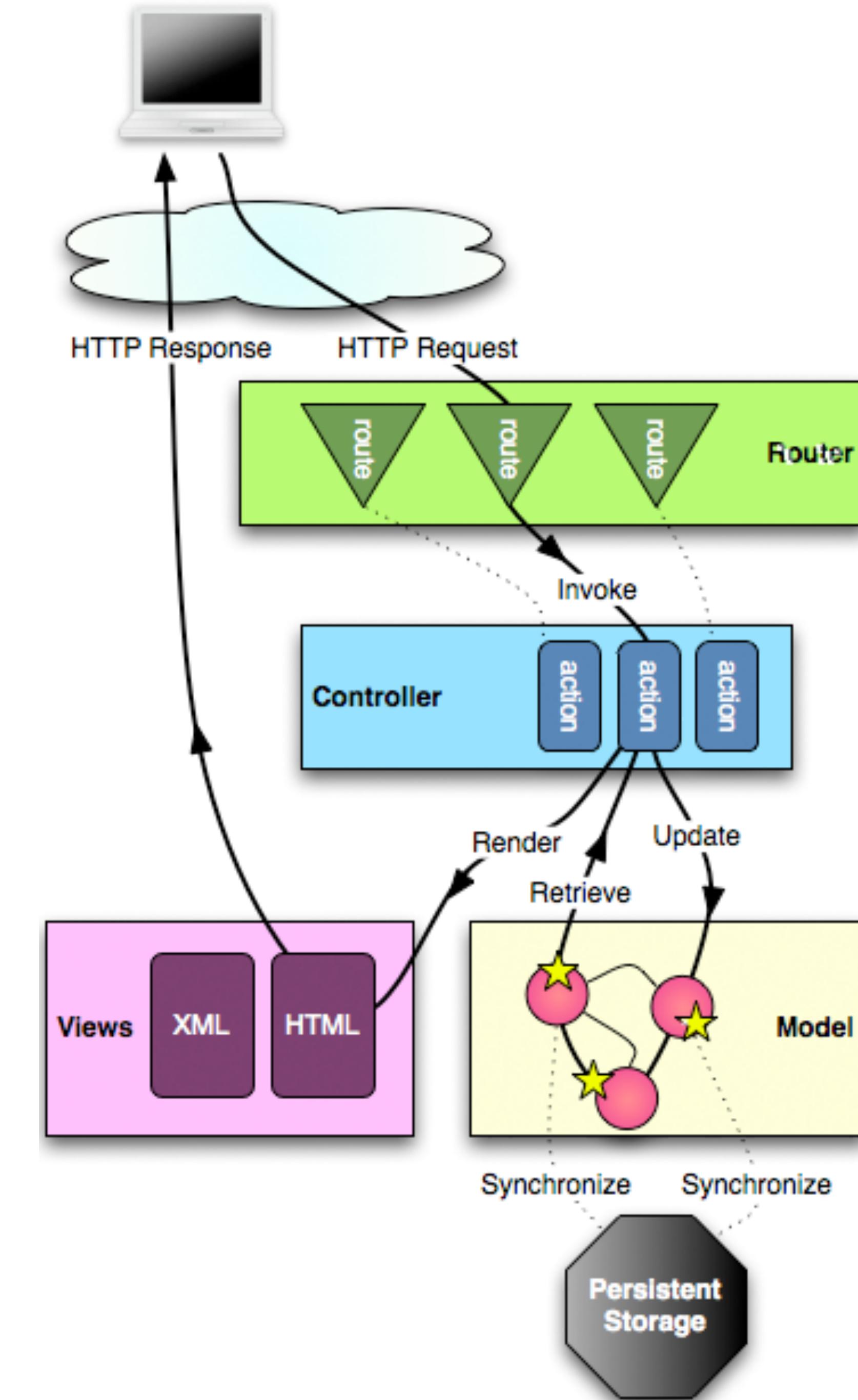
- Expanded understanding of the nature of the Internet

However, modern tools & frameworks are starting to dramatically simplify the process.



Play Framework

- A toolkit to enable the construction of **Web Applications** in the Java Programming language
- Does not replace the use of HTML + CSS - the toolkit is for building Web Applications, which is built on these technologies
- However, HTML + CSS constructs are restructured to enable them to interoperate with **Programs** written in **Java**
- Play is a **Web Application Development Framework**



Web Sites/Applications



theme · content · navigation
· structure · templates ·
style · interaction · busine...