CS 50 Web Design

APCSP Module 2: Internet

Unit 1: HTML (Chapter 1-4)

LECTURE 1: BASIC HTML PAGE DESIGN

DR. ERIC CHOU

IEEE SENIOR MEMBER

What is Website Design?

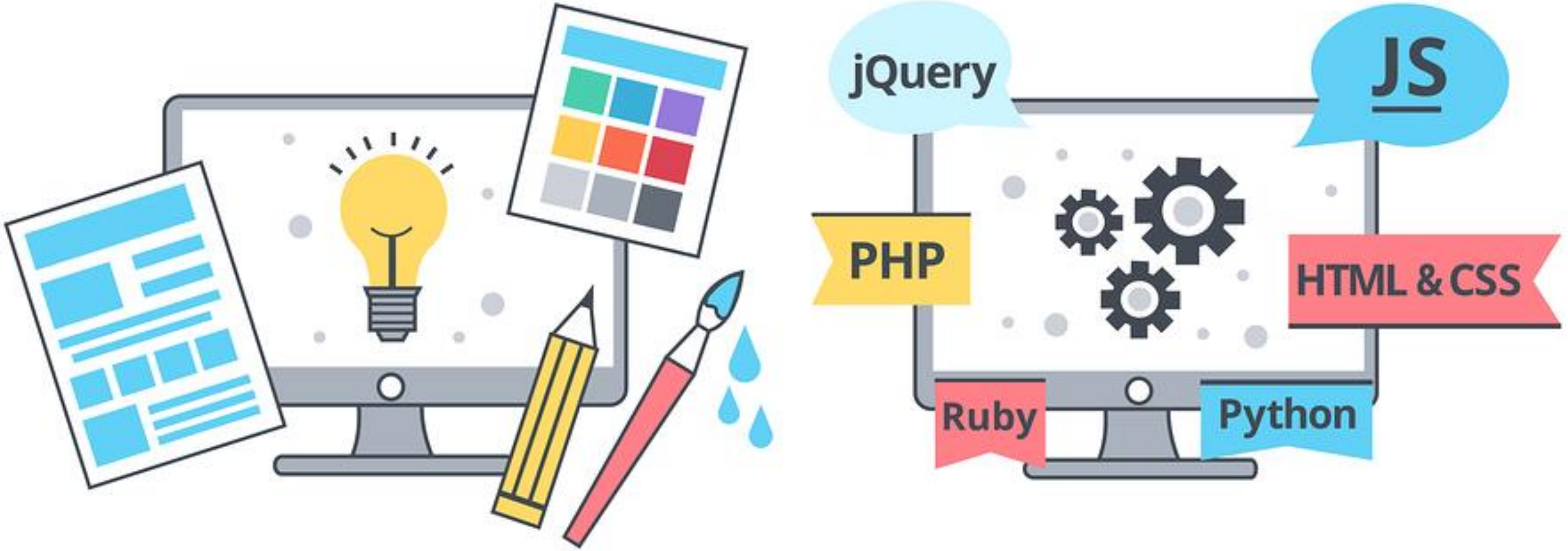
SECTION 1


Content
Homepage
Webhost
WHAT IS
WEBSITE
Design

What are view ports?

Mega Byte

WHAT IS WEB DESIGN?



WEB DESIGN VS DEVELOPMENT

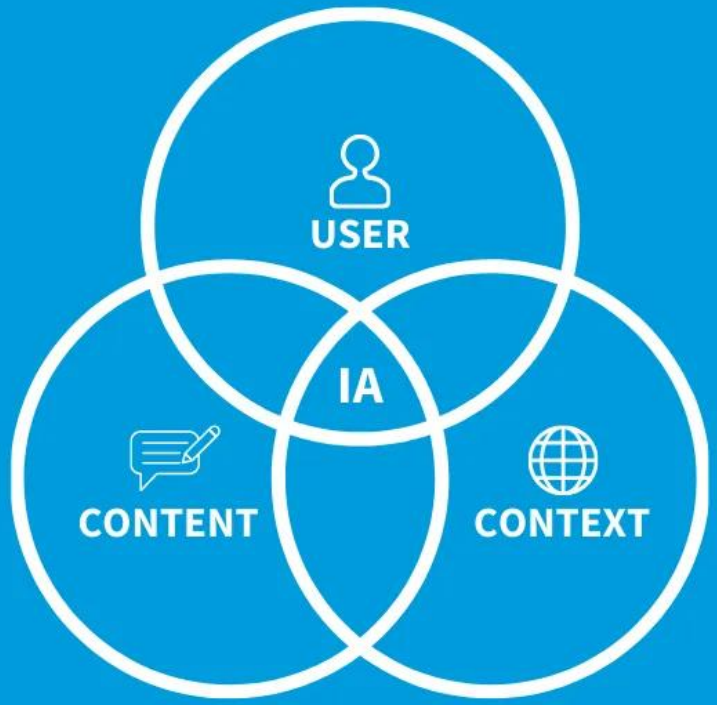
It Takes a Village

Website Creation Roles

Content Wrangling

- Information Architecture
- Content Strategy

Information Architecture



Content

- What kind of information is available?
- What relevance does it have to the user?

Context

- Where is the user seeking out the content?
- When, why and how is the user engaging with the content?

User

- Who is consuming the content?
- What value does it provide?
- What preexisting expectations do they have?


home
Blog
About
Categories
Contact
Year
Month
Category
Post
Team
Staff Profiles
By Year
By Month
By Day
Contact Methods

Responsibilities of a Content Marketing Strategist:

Identify business goals and the marketing means to those ends.

Coordinate with editorial teams on content creation.

Target audiences, keywords and best-performing content forms.

Consult on search engine optimization or deliver analytics.

All Manner of Design

SECTION 2

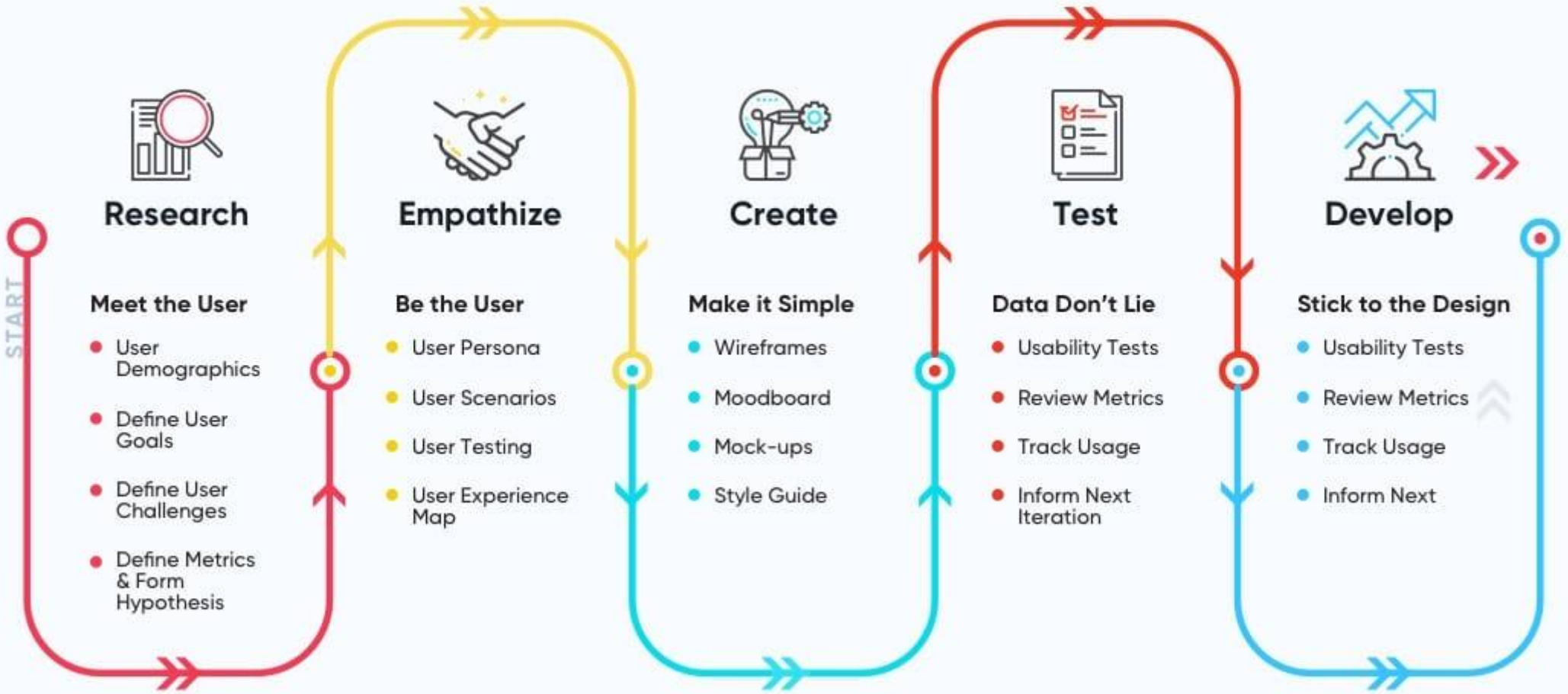
All Manner of Design

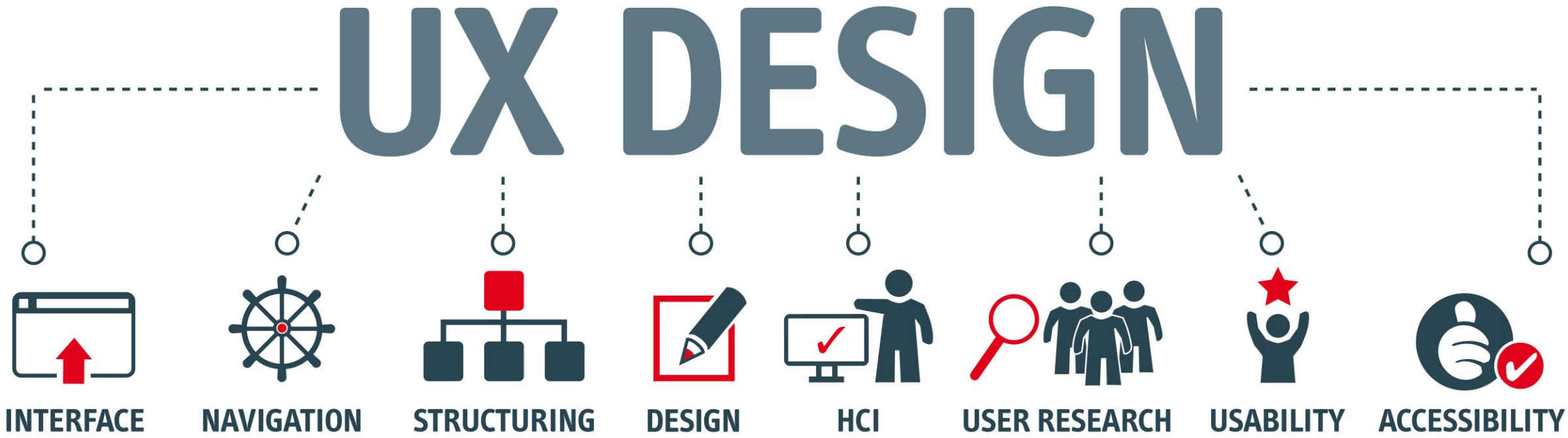
- User Experience (UX) design,
- Interaction Design (IxD), and
- User Interface (UI) design

User Experience (UX) Designer

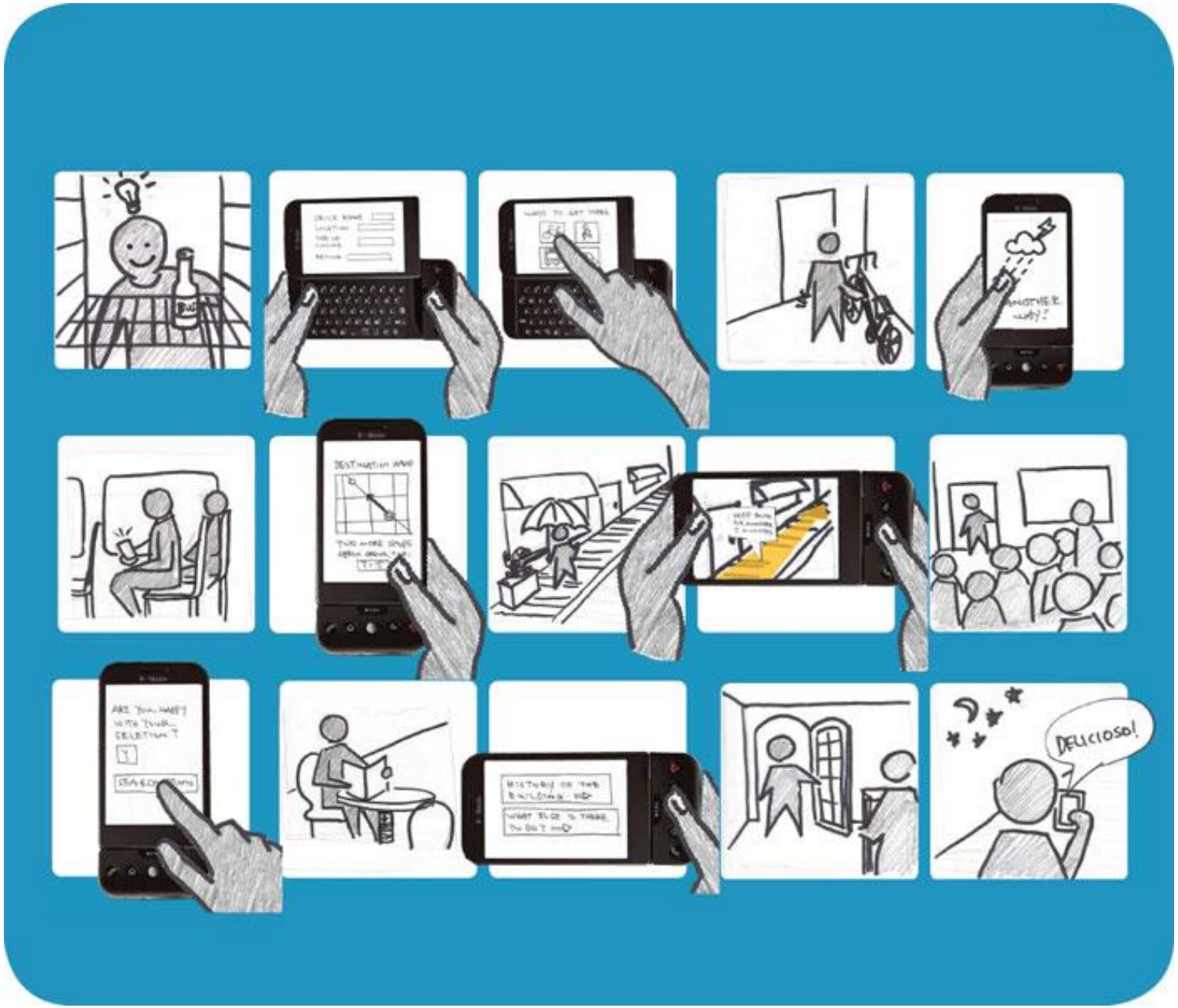
- The User Experience designer takes a holistic view of the design process—ensuring the entire experience with the site is favorable. **UX design is based on a solid understanding of users and their needs based on observations and interviews**.
- According to Donald Norman (who coined the term), UX design includes "all aspects of the user's interaction with the product: how it is perceived, learned, and used." For a website or application, that includes the visual design, the user interface, the quality and message of the content, and even the overall site performance. The experience must be in line with the organization's brand and business goals in order to be successful.

USER EXPERIENCE DESIGN PROCESS



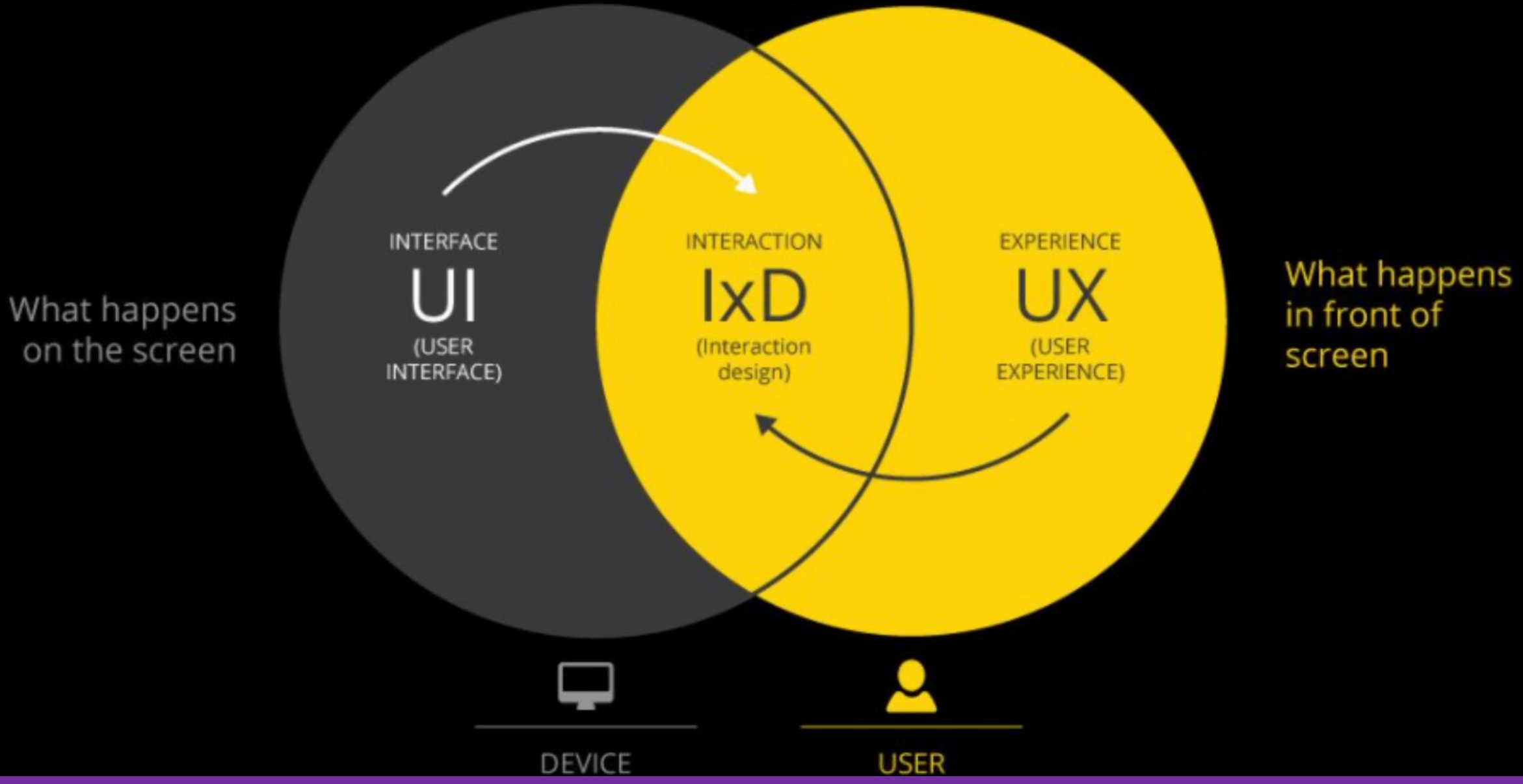

UX DESIGN
INTERFACE
NAVIGATION
STRUCTURING
DESIGN
HCI
USER RESEARCH
USABILITY
ACCESSIBILITY

User Stories & Scenarios


DESTINATION ADDRESS
DELICIOSO!

Interaction Design (IxD)

- The goal of the Interaction Designer is to make the site as easy, efficient, and delightful to use as possible. Closely related to interaction design is User Interface design, which tends to be more narrowly focused on the functional organization of the page as well as the specific tools (buttons, links, menus, and so on) that users use to navigate content or accomplish tasks.


What happens on the screen
INTERFACE
UI
(USER INTERFACE)
INTERACTION
IxD
(Interaction design)
EXPERIENCE
UX
(USER EXPERIENCE)
What happens in front of screen
DEVICE
USER

User Interface Design (UI)

- User Interface (UI) Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. UI brings together concepts from interaction design, visual design, and information architecture.

MOBILE UI DESIGN

Methods for the Future

- GESTURE-BASED INTERFACE
- FACIAL RECOGNITION
- AI ASSISTANT
- AUGMENTED REALITY

Deliverables

- User research and testing reports
- Wireframe diagrams
- Site diagram
- Storyboards and user flow charts

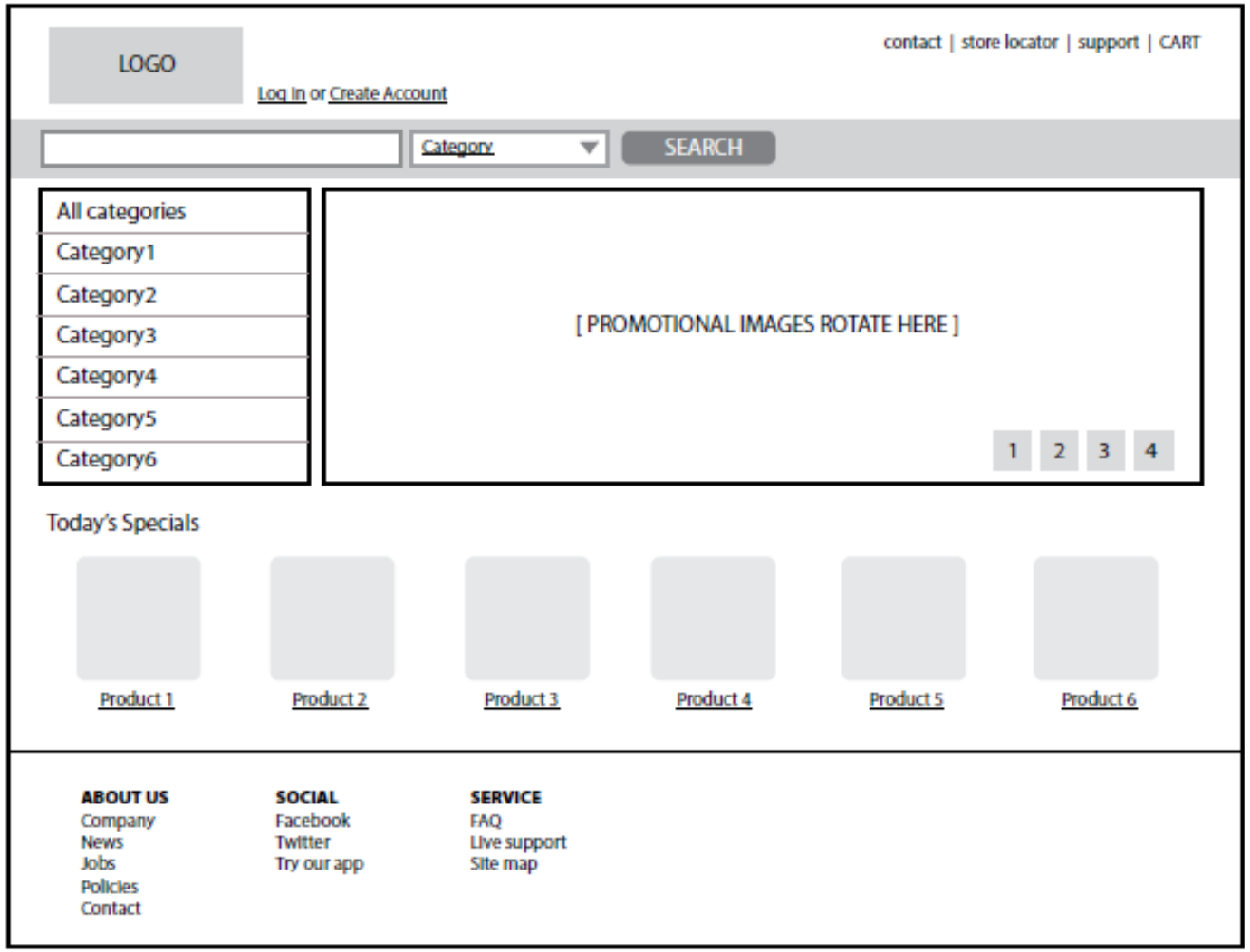


FIGURE 1-1. Wireframe diagram.

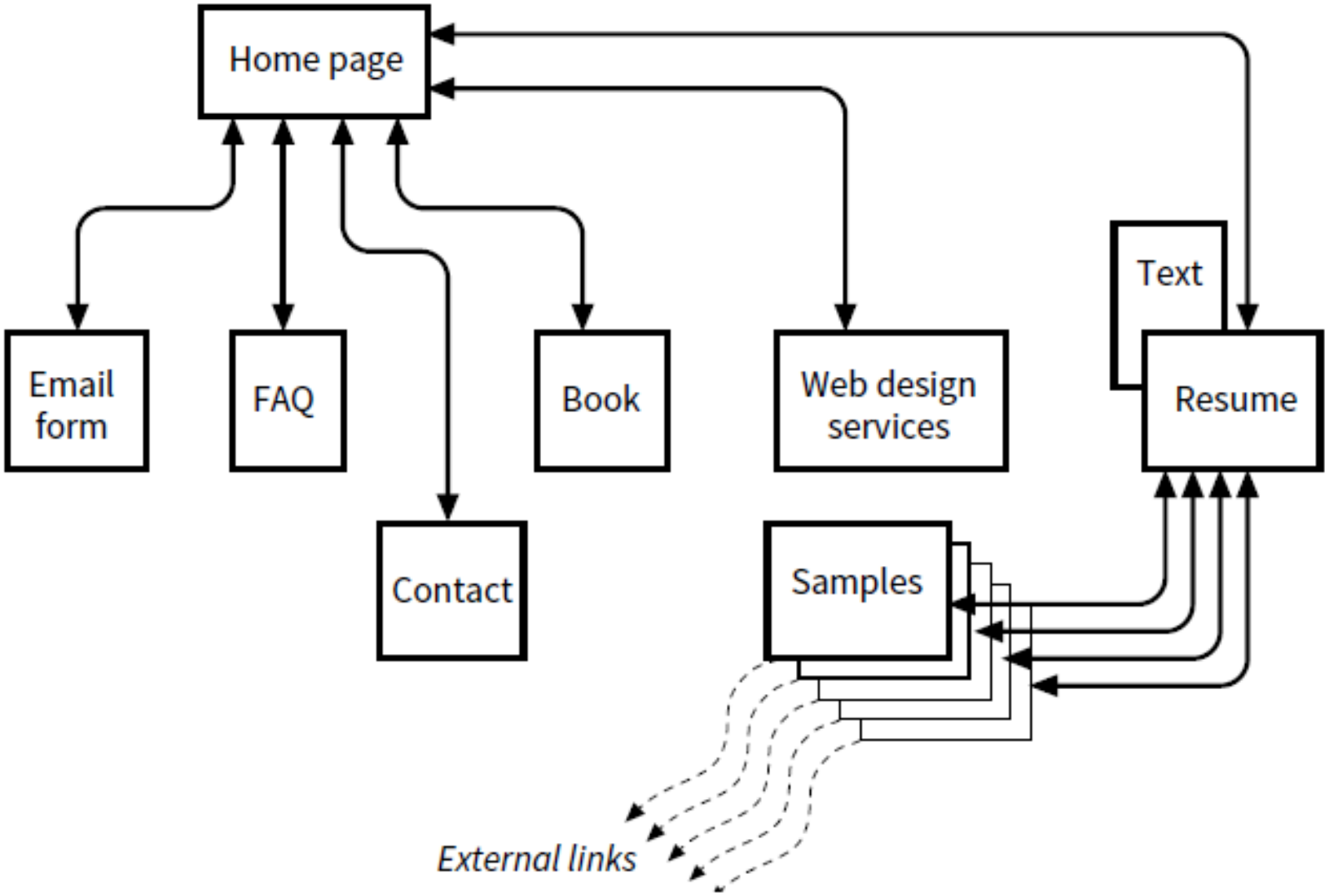


FIGURE 1-2. A simple site diagram.

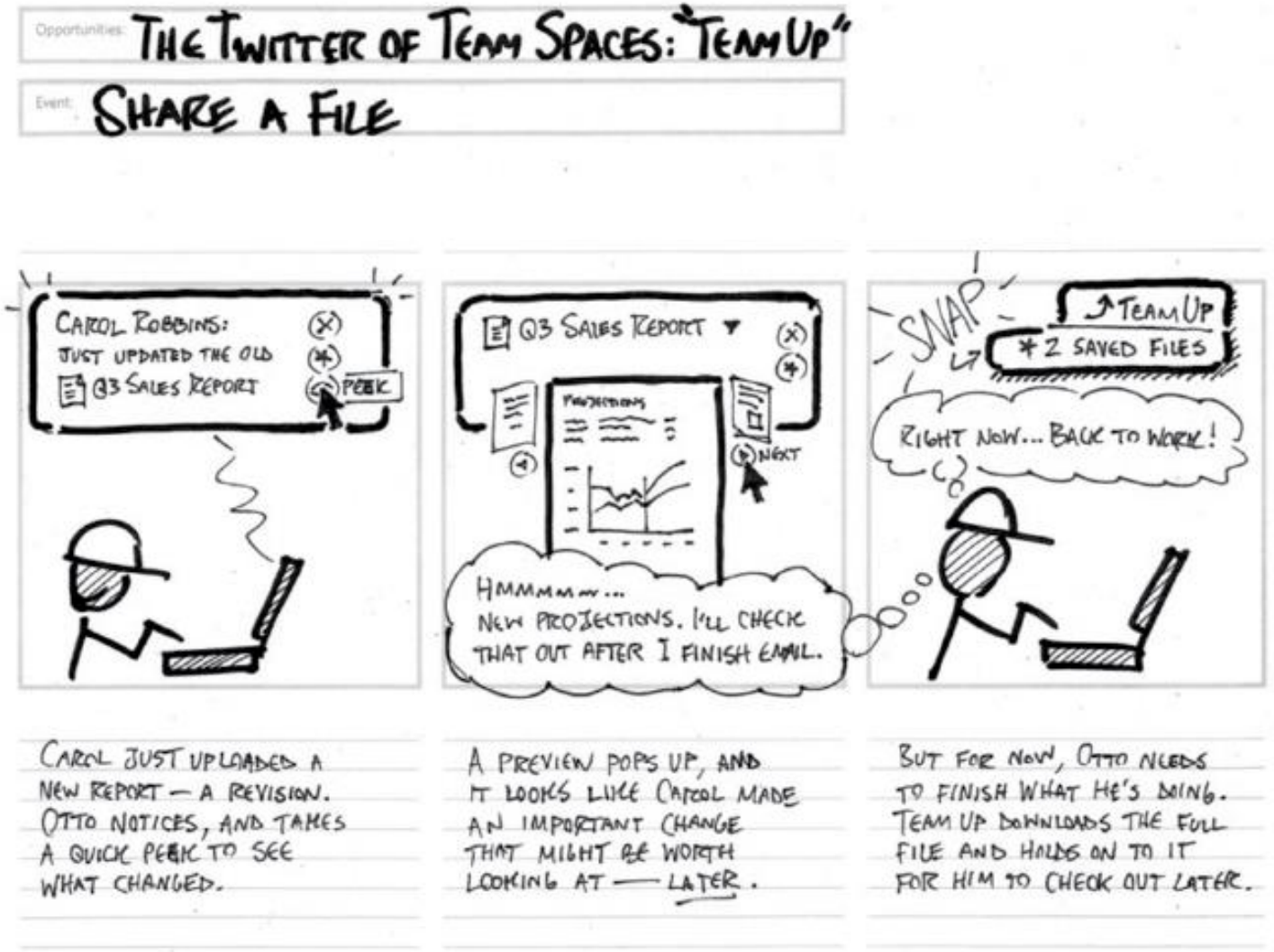


FIGURE 1-3. A typical storyboard (courtesy of Adaptive Path and Brandon Schauer).

Demo 1-1 First Page

1. Title
2. Favicon
3. Unicode

Front-End Design

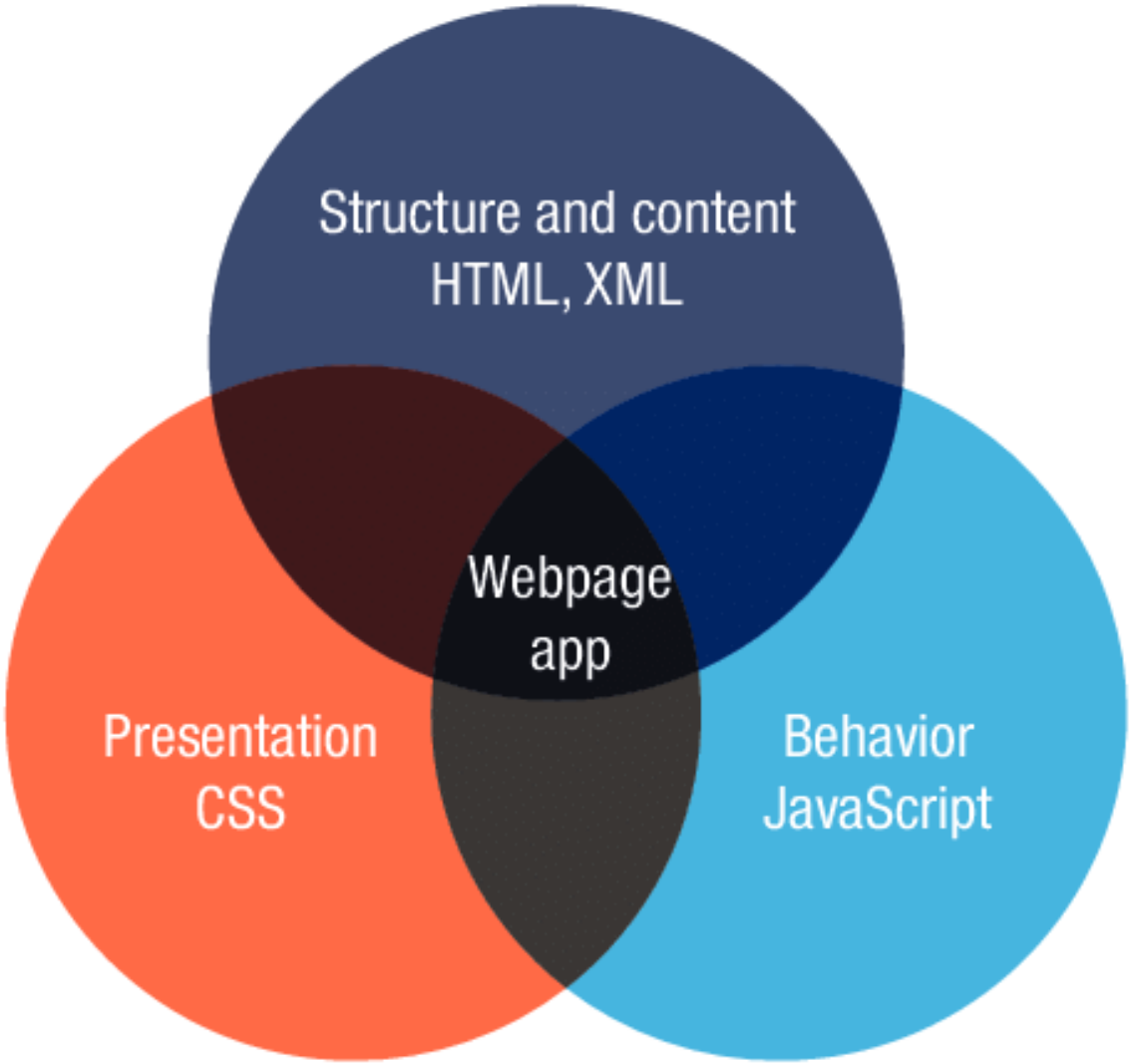
SECTION 3

Frontend development

- Frontend refers to any aspect of the design process that appears in or relates directly to the browser. That includes HTML, CSS, and JavaScript, all of which you will need to have intricate knowledge of if you want a job as a web developer. Let’s take a quick look at each.

Front-End Design

- Authoring/markup (HTML)
- Styling (CSS)
- JavaScript and DOM scripting


Structure and content
HTML, XML
Webpage
app
Presentation
CSS
Behavior
JavaScript

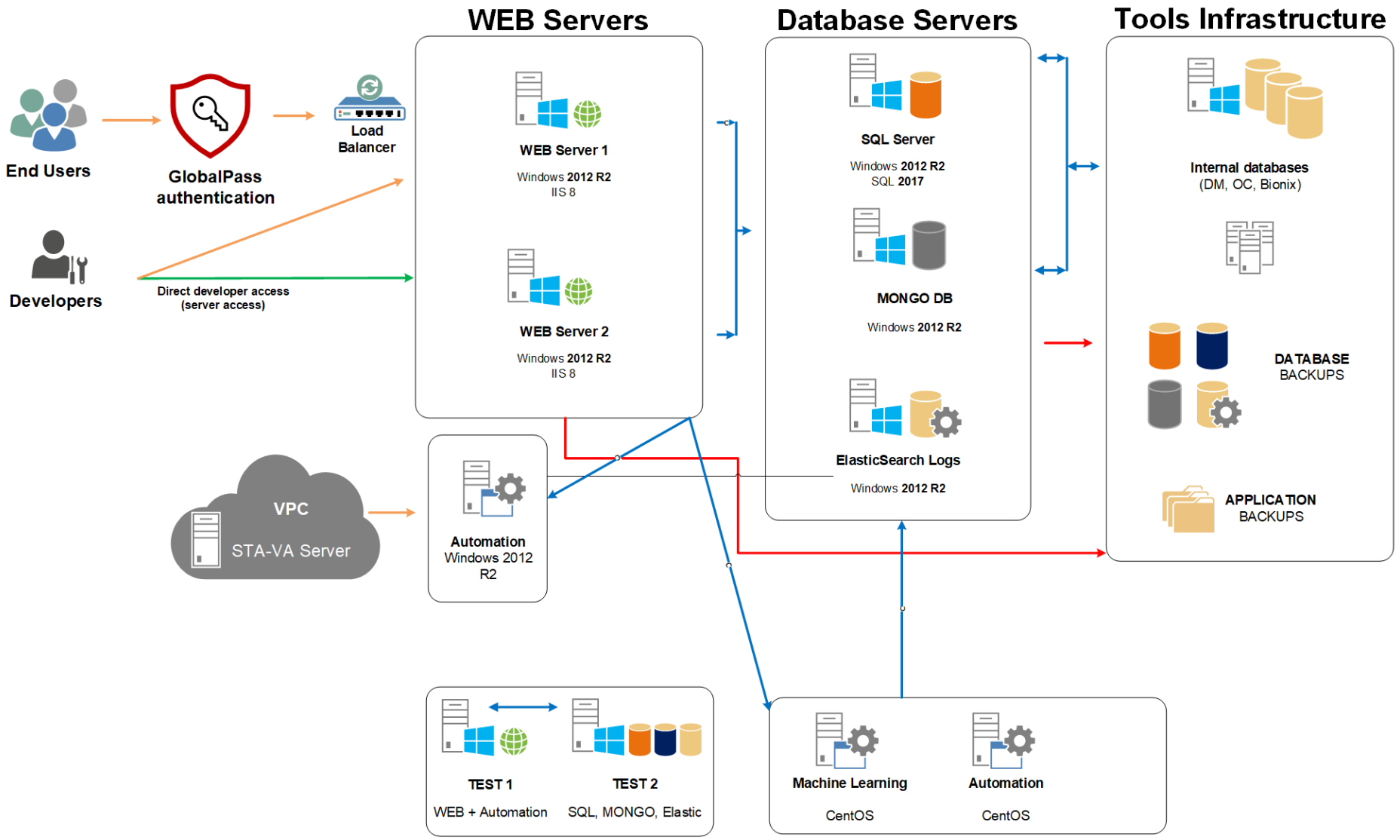
Backend Design

SECTION 4

Backend Design

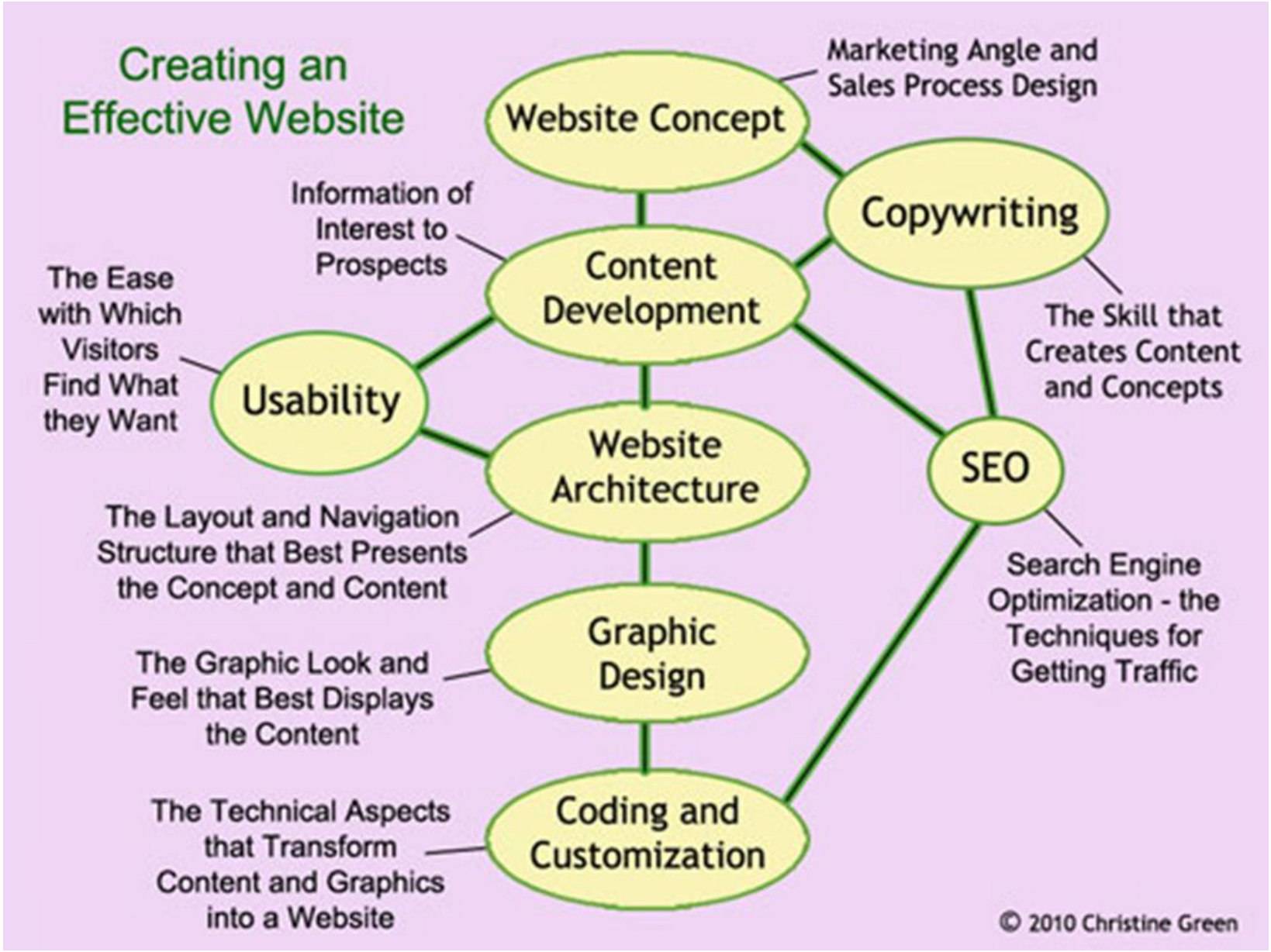
- Backend developers focus on the server, including the applications and databases that run on it. They may be responsible for installing and configuring the server software (we'll be looking more at servers in Chapter 2, How the Web Works). They will certainly be required to know at least one, and probably more, server-side programming languages, such as PHP, Ruby, .NET (or ASP.NET), Python, or JSP, in order to create applications that provide the functionality required by the site. Applications handle tasks and features like forms processing, content management systems (CMSs), and online shopping, just to name a few.
- Additionally, backend developers need to be familiar with configuring and maintaining databases that store all of the data for a site, such as the content that gets poured into templates, user accounts, product inventories, and more. Some common database languages include MySQL, Oracle, and SQL Server.

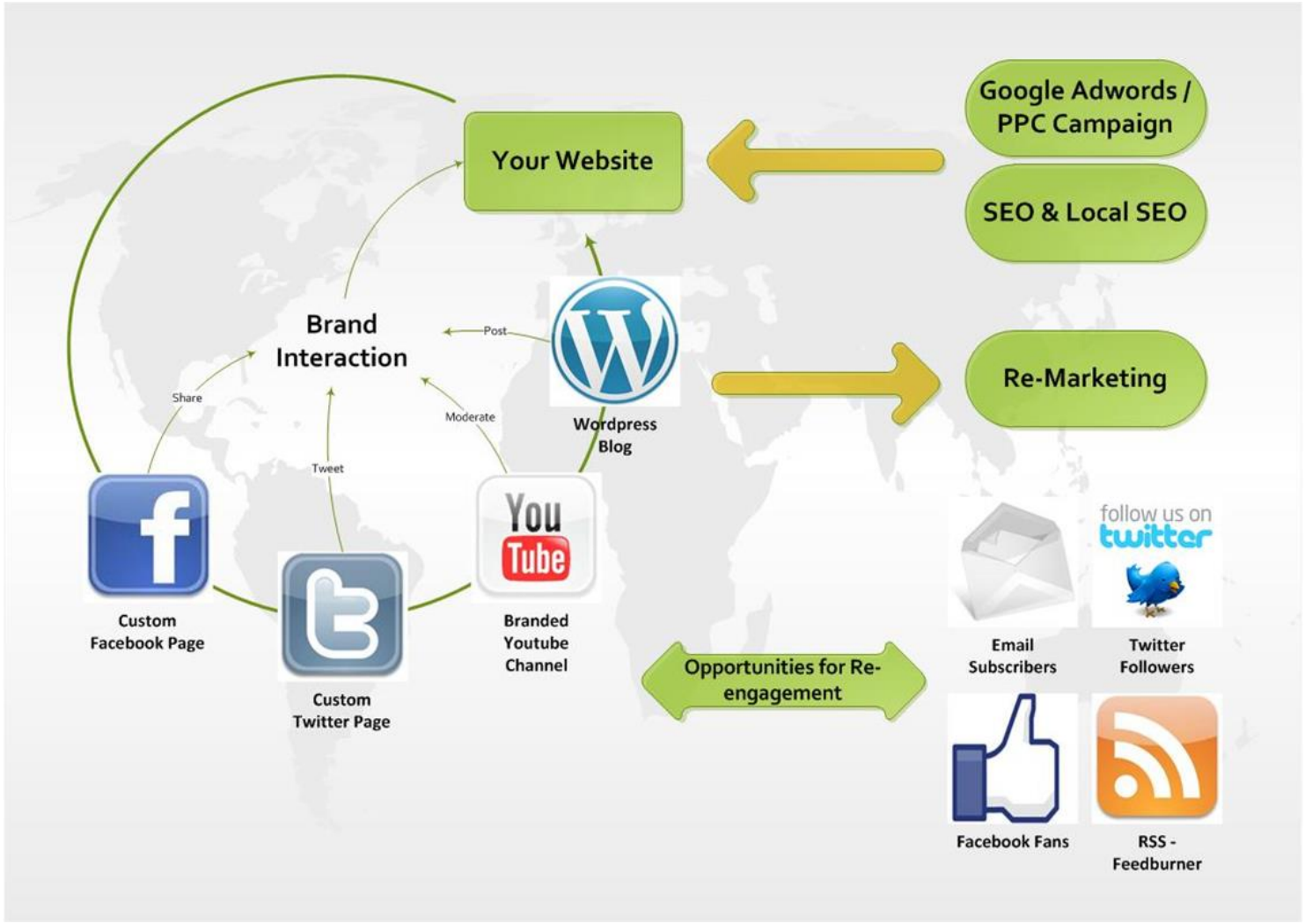
STA Bionix Lean backend infrastructure design



Backend Design is not the Main Focus of This Course

- Backend design requires the knowledge of database, web-server, system architecture, machine learning, and AI. They are beyond the scope of Web Design


Creating an Effective Website
Website Concept
Marketing Angle and Sales Process Design
Copywriting
The Skill that Creates Content and Concepts
Content Development
Information of Interest to Prospects
Usability
The Ease with Which Visitors Find What they Want
Website Architecture
The Layout and Navigation Structure that Best Presents the Concept and Content
SEO
Search Engine Optimization - the Techniques for Getting Traffic
Graphic Design
The Graphic Look and Feel that Best Displays the Content
Coding and Customization
The Technical Aspects that Transform Content and Graphics into a Website
© 2010 Christine Green



Web-site Promotion


THE WEBSITE DESIGN PROCESS
1
DISCOVERY MEETING
2
DESIGN PROCESS
3
WEBSITE SHELL BUILD
4
CONTENT CREATION
5
NAVIGATION FLOW
6
PLUGIN INTEGRATION
7
FORM CREATION
8
WALK-THROUGH TESTING


Website Production
Marketing
SMM
SEo
Target Group
Emails
SEM
Affiliates
Back End
Hosting
Database
CMS
Editors
Admin
Content
Design
Layout
Colors
Photos
Graphics
Fonts
Front End
Technologies
User Interface
Code
Responsible Design
Navigation

Implementation

- Authoring/markup (Structure) – HTML
- Text Design
- Graphic Design
- Animation Design
- Digital Imaging
- Video Creation
- Styling (Presentation) - CSS
- Scripting and Programming (Functional/behaviors) – Javascript

Front-end V.S. Backend Development

Front-End

- Graphic design and image production
- Interface Design
- Information design as it pertains to the user's experience of the site
- HTML document and style sheet development
- Javascript

Back-End

- Graphic design and image production
- Forms Processing
- Database Programming
- Content Management
- Other Server-side Web applications using PHP, JSP, Ruby, ASP.NET, Java, and other programming languages

Programming Language for Web-site Construction

Client Side: Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) Javascript and Document Object Model (DOM scripting

Server-side programming and database management: Java, Python, Ruby, PHP, C++, ASP.NET, Javascript, SQL (MySQL, PostgreSQL)

Equipment for Web-Design

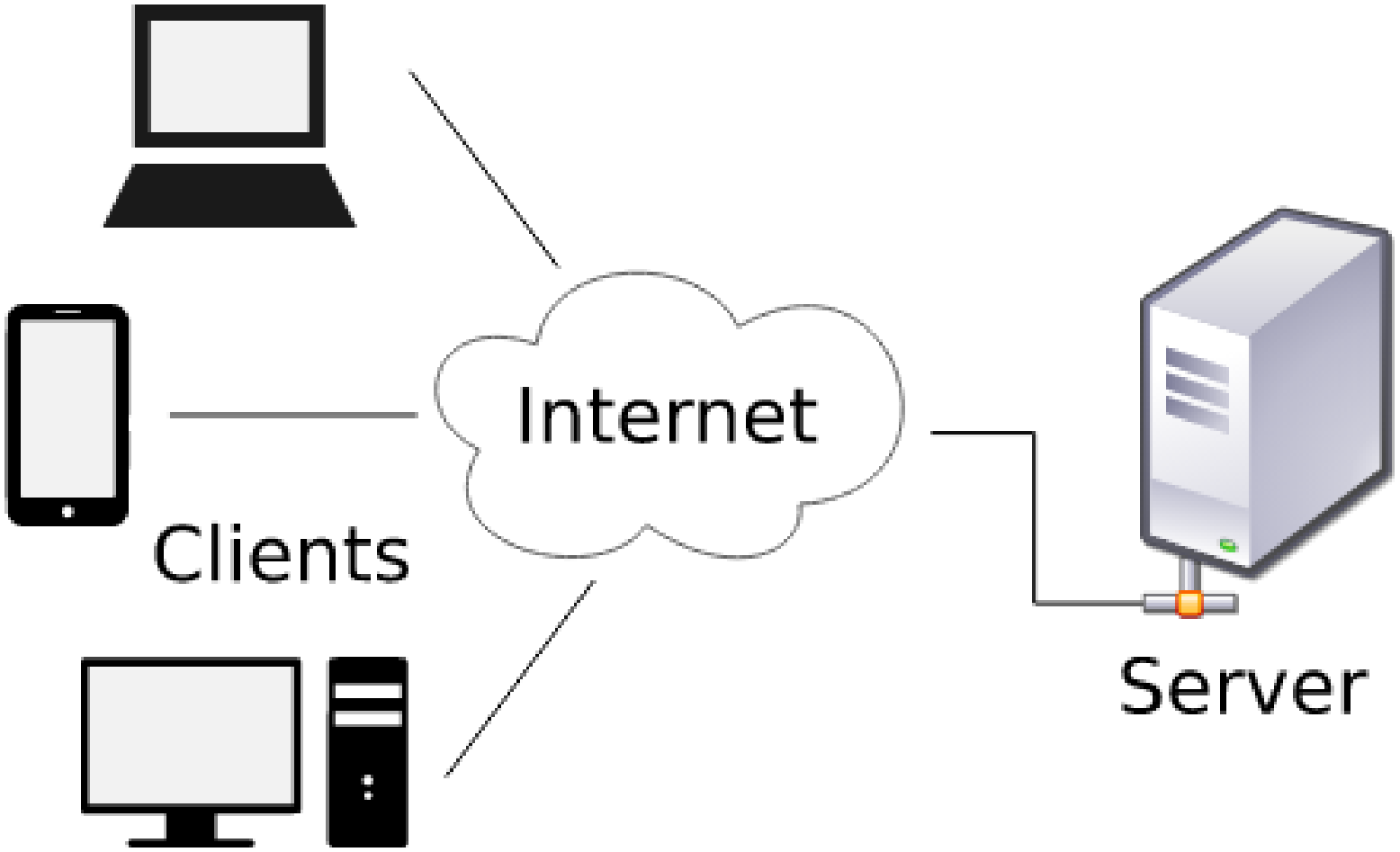
- A solid, up-to-date computer
- Extra memory
- A large monitor
- A scanner, graphic tablet, and/or digital camera
- A second computer
- Mobile Devices

Software Requirement for Web-Design

- Web Page Authoring: Adobe Dreamweaver, MS Expression Web 4, Nvu
- HTML Editor: Visual Studio Code, TextPad, Subline text, Coda, TextMate, BBEdit
- Image Editing and Drawing Software: Adobe Photoshop, Adobe Illustrator, Adobe Fireworks, Corel Paint Shop Pro Photo, GIMP, Inkscape
- Internet Browser: Chrome, Firefox, IE, Opera, Safari
- FTP tool: WS_FTP, Cute_FTP, AceFTP, Filezilla, Transmit, Cyberduck, Fetch
- Video Maker: MS movie maker, Cyberlink Power Director,
- Audio: Microsoft Expression Encoder/screen capture
- Screen Capture: Screenhunter

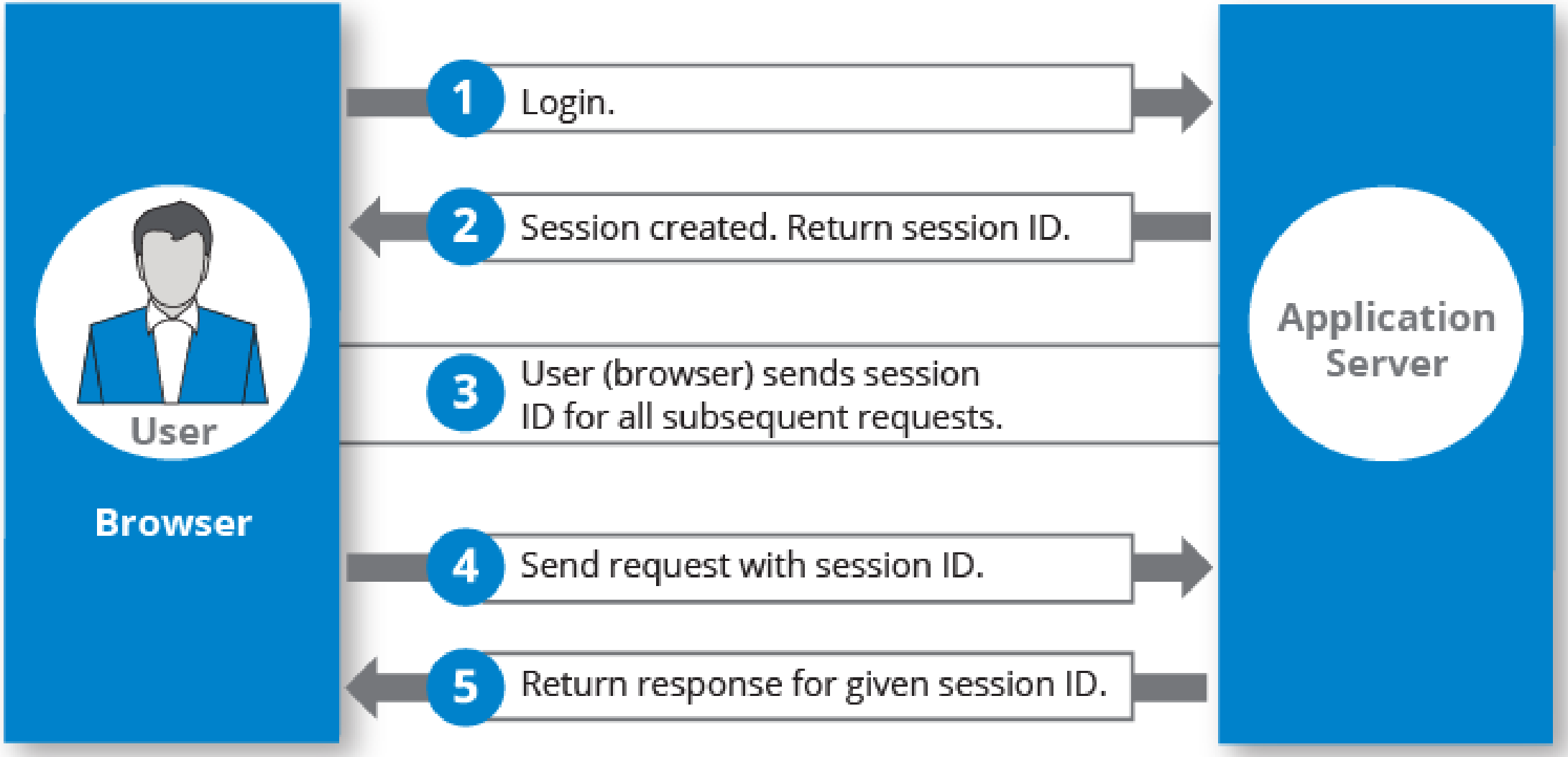
How Web Works?

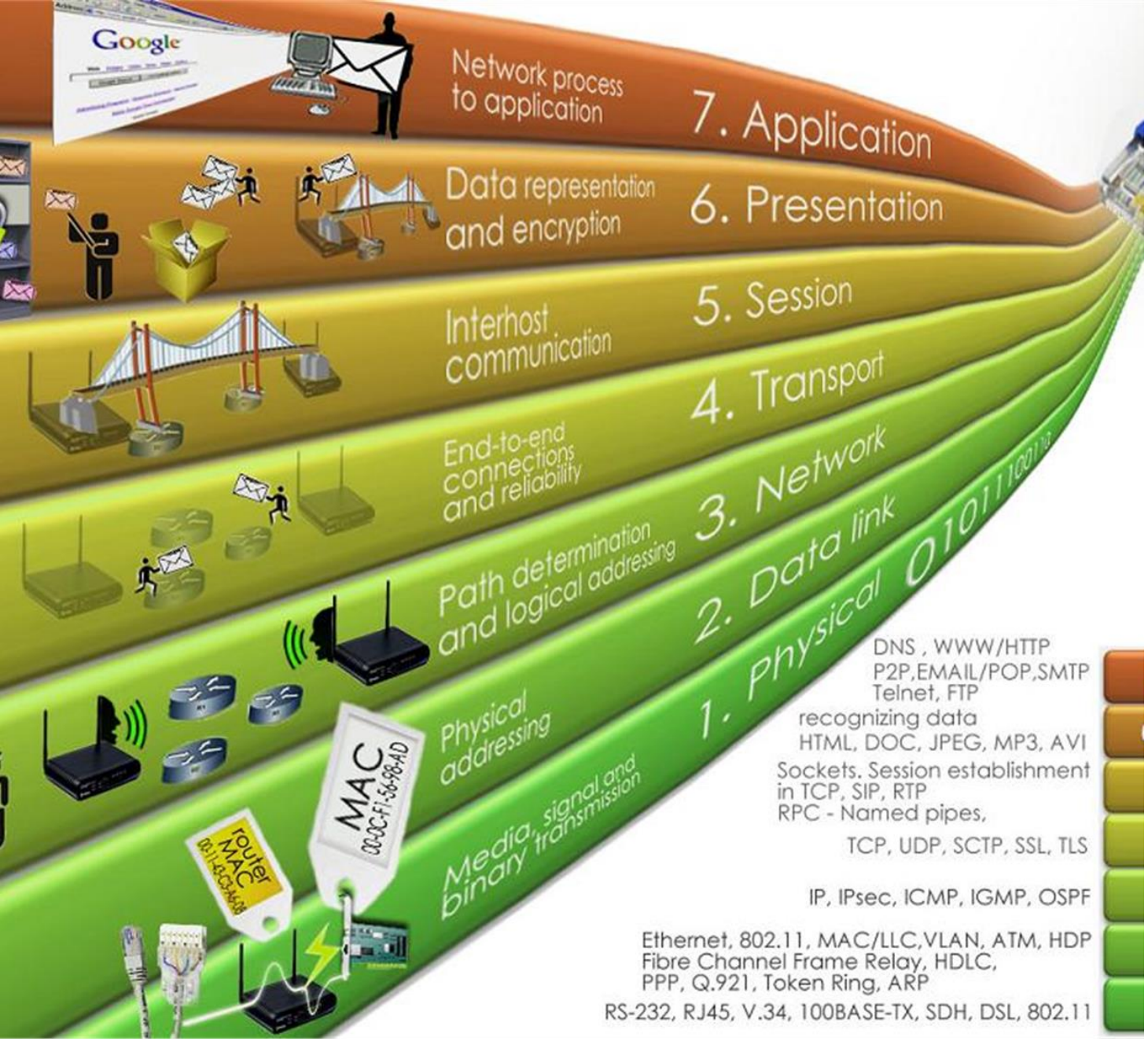
SECTION 5

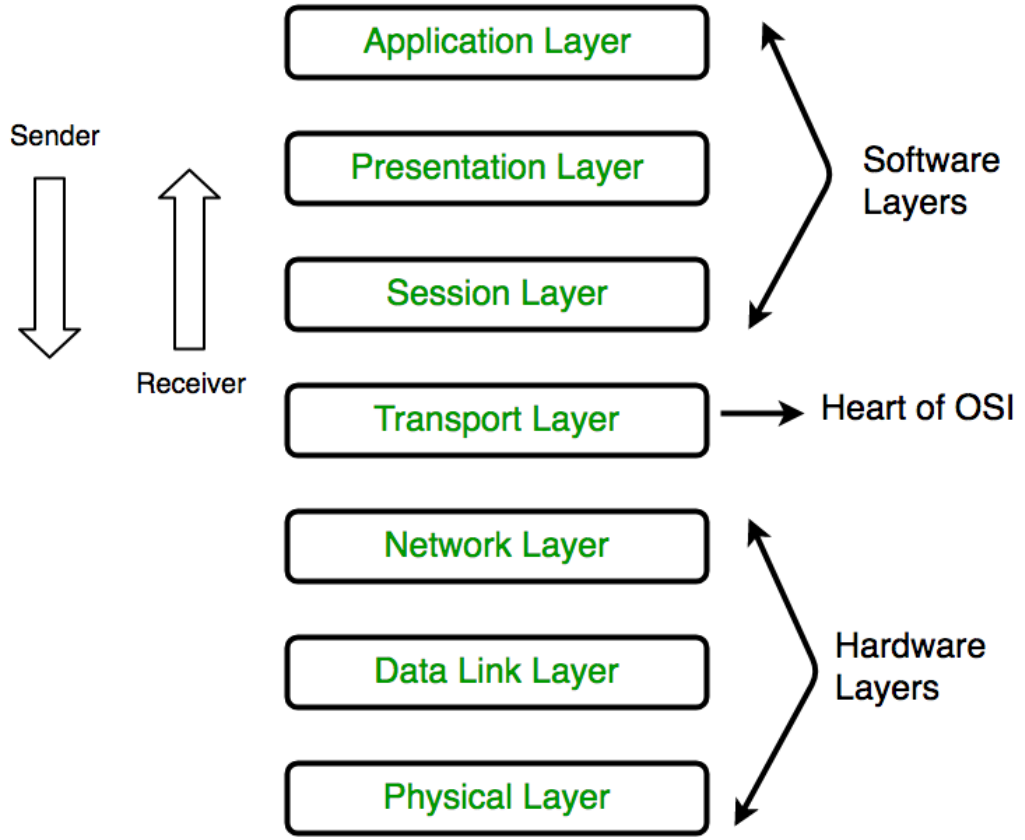

Internet
Clients
Server

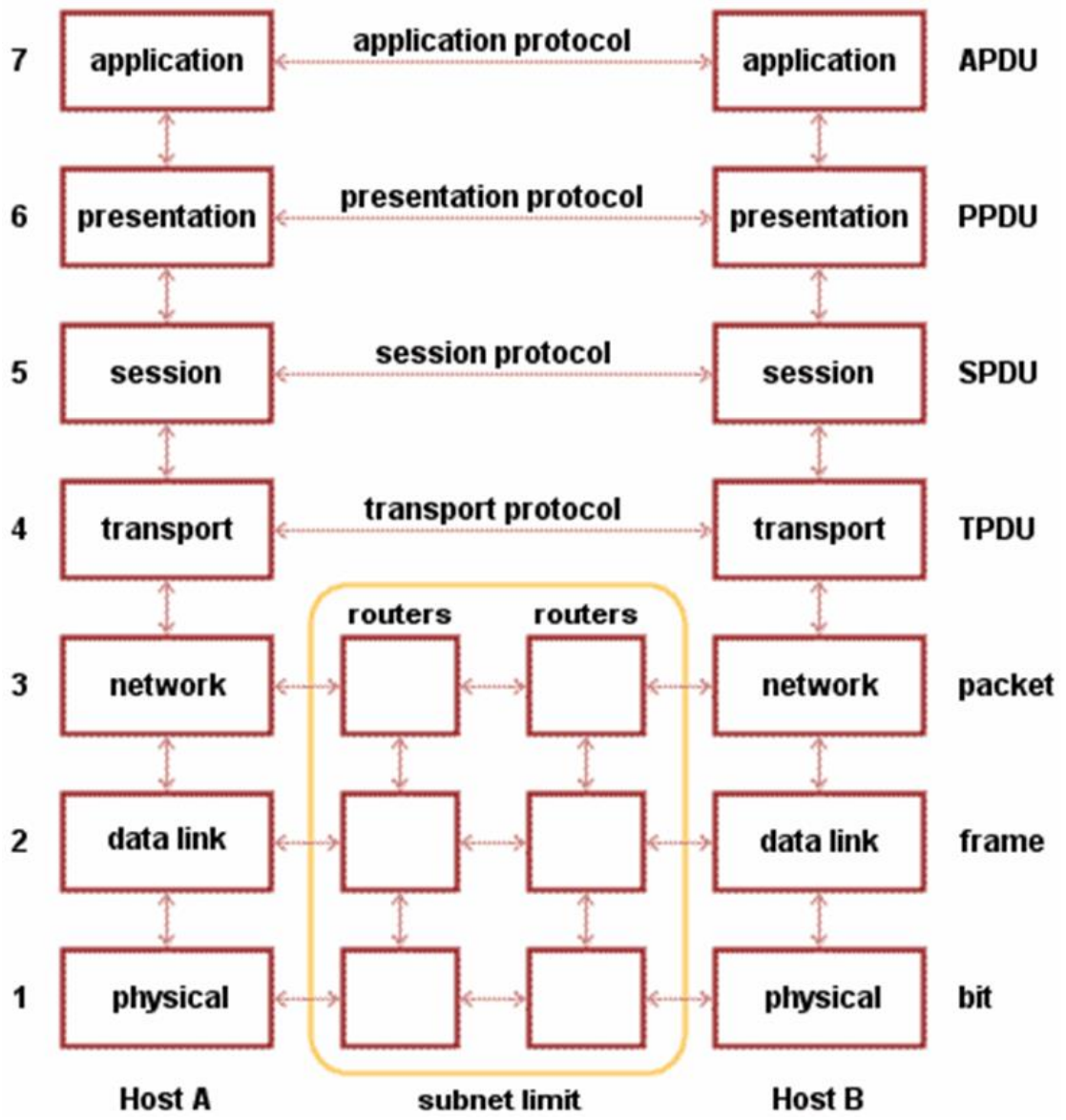
Web Browsers

- It is common to think of a browser as a window on a computer monitor with a web page displayed in it. These are known as graphical browsers or desktop browsers and for a long time, they were the only web-viewing game in town. The most popular desktop browsers as of this writing include Edge and Internet Explorer for Windows, Chrome, Firefox, and Safari, with Opera and Vivaldi bringing up the rear.


User
Browser
1
Login.
2
Session created. Return session ID.
3
User (browser) sends session
ID for all subsequent requests.
4
Send request with session ID.
5
Return response for given session ID.
Application
Server


Google
Network process to application
7. Application
Data representation and encryption
6. Presentation
Interhost communication
5. Session
End-to-end connections and reliability
4. Transport
Path determination and logical addressing
3. Network
Physical addressing
2. Data link
Media, signal and binary transmission
1. Physical
MAC
00-0C-F1-56-98-AD
router MAC
DNS , WWW/HTTP
P2P,EMAIL/POP,SMTP
Telnet, FTP
recognizing data
HTML, DOC, JPEG, MP3, AVI
Sockets. Session establishment
in TCP, SIP, RTP
RPC - Named pipes,
TCP, UDP, SCTP, SSL, TLS
IP, IPsec, ICMP, IGMP, OSPF
Ethernet, 802.11, MAC/LLC,VLAN, ATM, HDP
Fibre Channel Frame Relay, HDLC,
PPP, Q.921, Token Ring, ARP
RS-232, RJ45, V.34, 100BASE-TX, SDH, DSL, 802.11


Sender
Receiver
Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer
Software Layers
Heart of OSI
Hardware Layers



Logical Internet Model

Application Layer Protocols

- File Transfer Protocol (FTP)
- Hypertext Transfer Protocol (HTTP)
- Simple Mail Transfer Protocol (SMTP)
- Post-Office Protocol (POP3)
- Domain Name System (DNS)

Application Layer Protocols

- **Uniform Resource Locator:**

  <scheme>:<hierarchical part>[? query][# fragment]

- **Important Schemes:**

  ftp, http, mailto, file, app, dns, https, im, info, mms

- **URL Sample:** type http in google
  https://**www.google.com**/search?**q**=Application+Layer+Protocols&oq=Application+Layer+Protocols&aqs=chrome.0.69i59.6246j0j8&sourceid=chrome&es_sm=122&ie=UTF-8#q=http

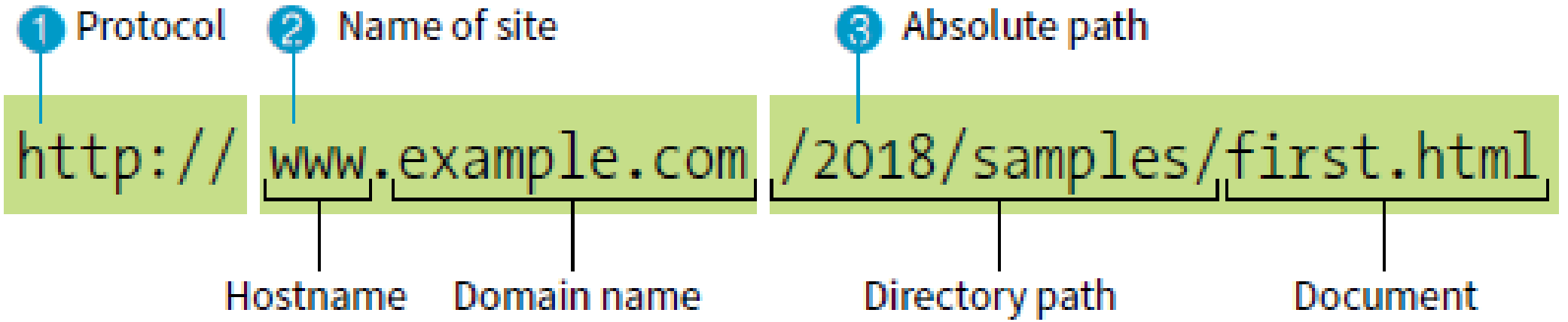
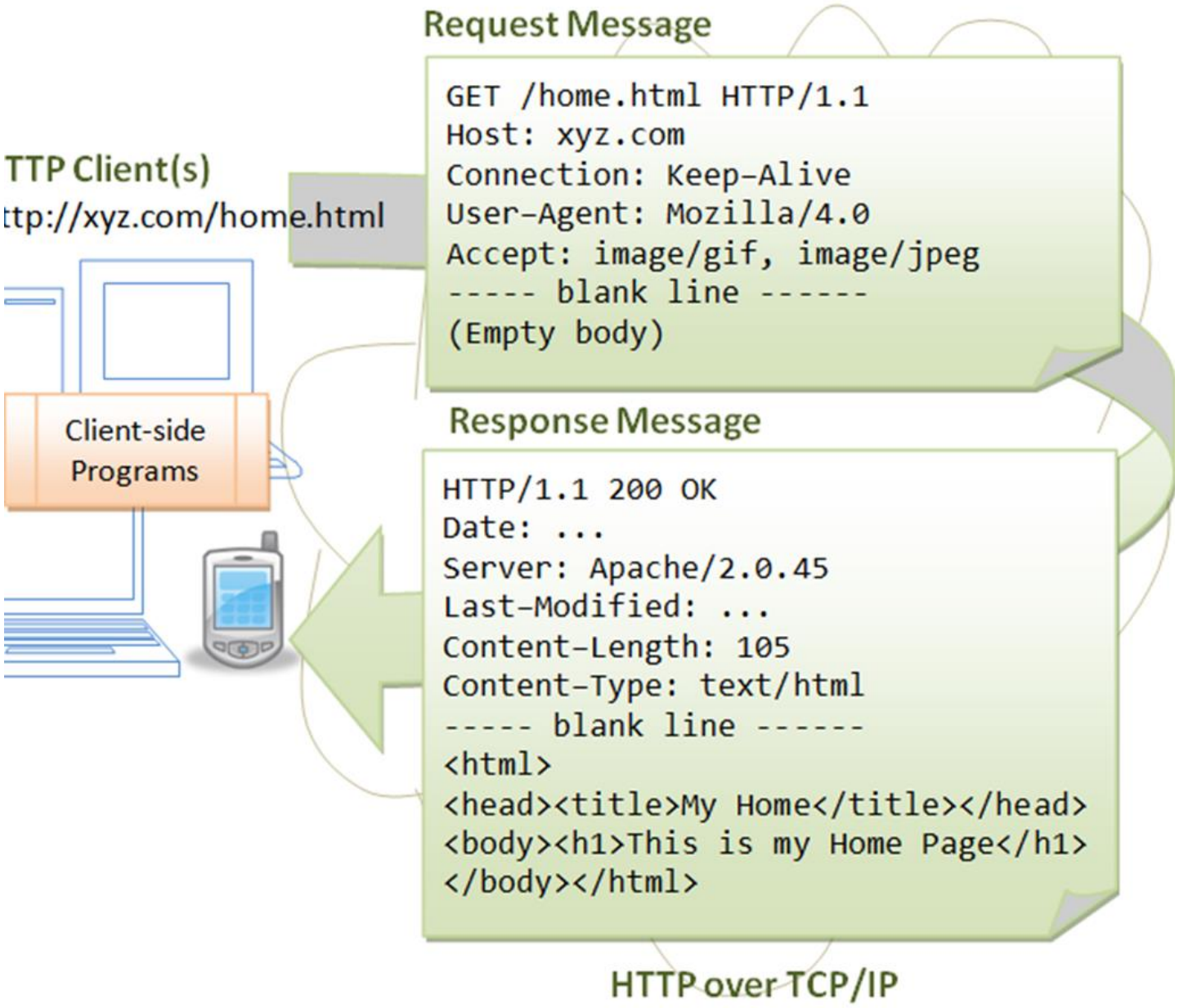


FIGURE 2-1. The parts of a URL.


Request Message
GET /home.html HTTP/1.1
Host: xyz.com
Connection: Keep-Alive
User-Agent: Mozilla/4.0
Accept: image/gif, image/jpeg
----- blank line ------
(Empty body)
TTP Client(s)
ttp://xyz.com/home.html
Client-side Programs
Response Message
HTTP/1.1 200 OK
Date: ...
Server: Apache/2.0.45
Last-Modified: ...
Content-Length: 105
Content-Type: text/html
----- blank line ------
<html>
<head><title>My Home</title></head>
<body><h1>This is my Home Page</h1>
</body></html>
HTTP over TCP/IP

The Anatomy of a Web Page

The web page shown in this browser window consists of four separate files:

- An HTML text document
- A style sheet
- Two images

Tags in the HTML source document give the browsers instructions for how the text is structured and where the images should be placed.

Jen's Kitchen

If you love to read about **cooking and eating**, would like to learn about some of the best restaurants in the world, or just want a few choice recipes to add to your collection, *this is the site for you!*

Your pal, Jen at Jen's Kitchen

---

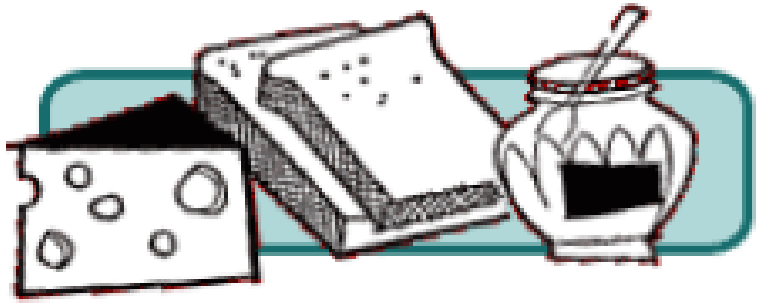
index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Jen's Kitchen</title>
  <link rel="stylesheet" href="kitchen.css" type="text/css">
</head>

<body>
<h1><img src="foods.png" alt="food illustration"> Jen's Kitchen</h1>

<p>If you love to read about <strong>cooking and eating</strong>, would like to learn about some of the best
restaurants in the world, or just want a few choice recipes to add to your collection, <em>this is the site
for you!</em></p>

<p><img src="spoon.png" alt="spoon illustration"> Your pal, Jen at Jen's Kitchen</p>

<hr>
<small>Copyright 2018, Jennifer Robbins</small>
</body>
</html>
```

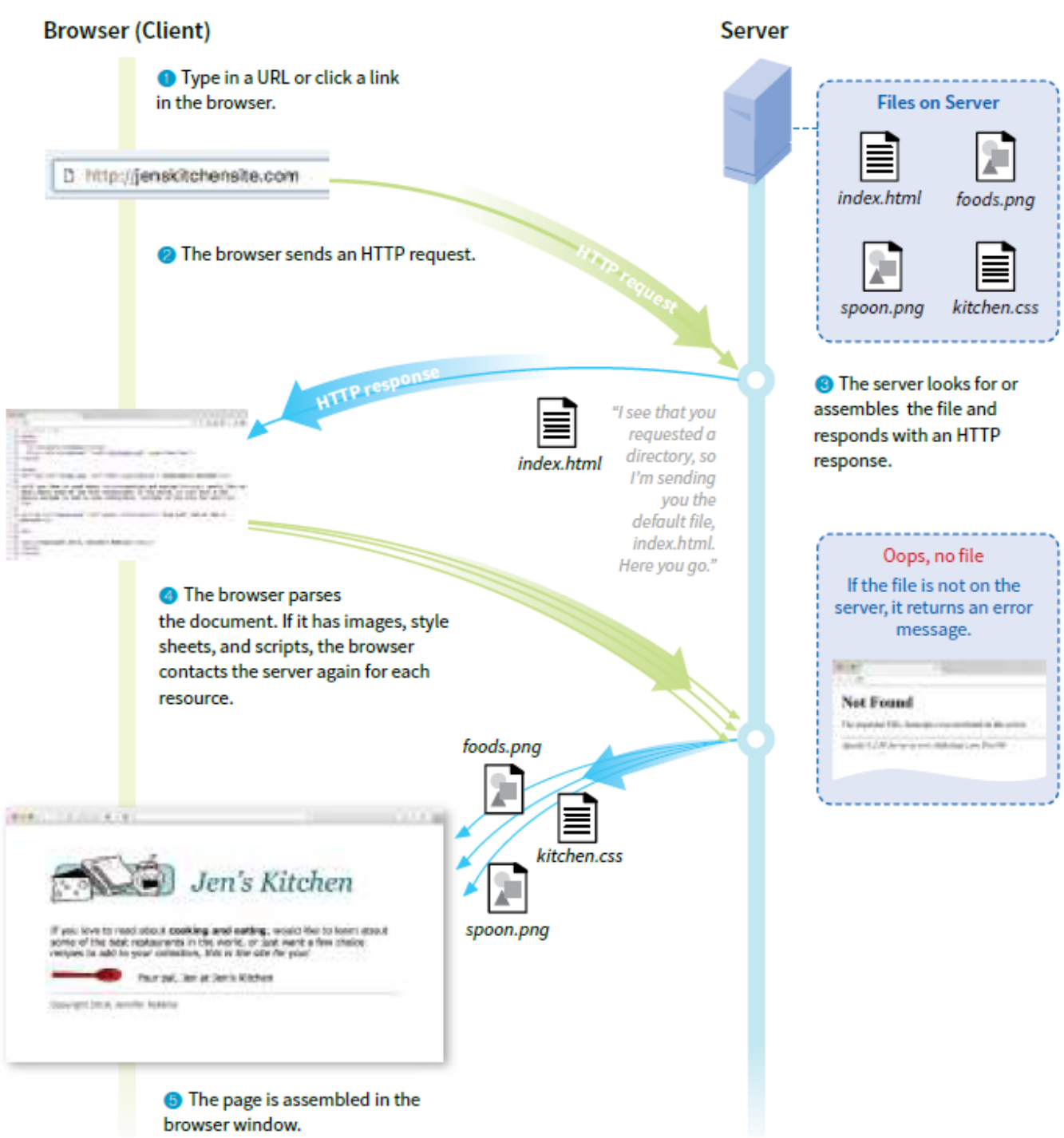
kitchen.css

```
body { font: normal 1em Verdana; width: 80%; margin: 1em auto; }

h1 { font: italic 3em Georgia; color: rgb(23, 109, 109);
     margin: 1em 0 1em; }

img { margin: 0 20px 0 0; }

h1 img { margin-bottom: -20px; }

small { color: #666666; }
```

foods.png



spoon.png


Browser (Client)
Server
1 Type in a URL or click a link in the browser.
http://jenskitchensite.com
2 The browser sends an HTTP request.
HTTP request
Files on Server
index.html
foods.png
spoon.png
kitchen.css
3 The server looks for or assembles the file and responds with an HTTP response.
HTTP response
index.html
"I see that you requested a directory, so I'm sending you the default file, index.html. Here you go."
Oops, no file
If the file is not on the server, it returns an error message.
Not Found
4 The browser parses the document. If it has images, style sheets, and scripts, the browser contacts the server again for each resource.
foods.png
kitchen.css
spoon.png
Jen's Kitchen
5 The page is assembled in the browser window.

Demo 1-2 First Web-Page

1. Server
2. URL
3. IMG
4. h2/p tags
5. Style

Big Concepts That You Need to Know

SECTION 6

FIGURE 3-1. Brad Frost sums up the reality of device diversity nicely (*bradfrostweb.com*).

Sticking with the Standards

- A good start is to follow the standards documented by the World Wide Web Consortium (W3C). Sticking with web standards is your primary tool for ensuring your site is consistent on all standards-compliant browsers (that's approximately 99% of browsers in current use). It also helps make your content forward-compatible as web technologies and browser capabilities evolve. Another benefit is that you can tell your clients that you create "standards-compliant" sites, and they will like you more.

Sticking with the Standards

- The notion of standards compliance may seem like a no-brainer, but it used to be that everyone, including the browser makers, played fast and loose with HTML and scripting. The price we paid was incompatible browser implementations and the need to create sites twice to make them work for everyone. I talk more about web standards throughout this book, so I won't go into too much detail here. Suffice it to say that the web standards are your friends. Everything you learn in this book will start you off on the right foot.

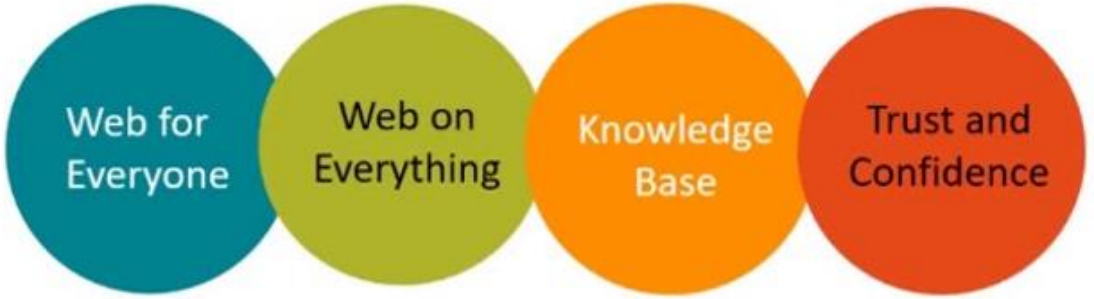
WEB STANDARDS > WORLD WIDE WEB CONSORTIUM (W3C)

Mission:
"to lead the Web to its full potential"

In order to achieve the mission and goals W3C sets and monitors **WEB STANDARDS.**

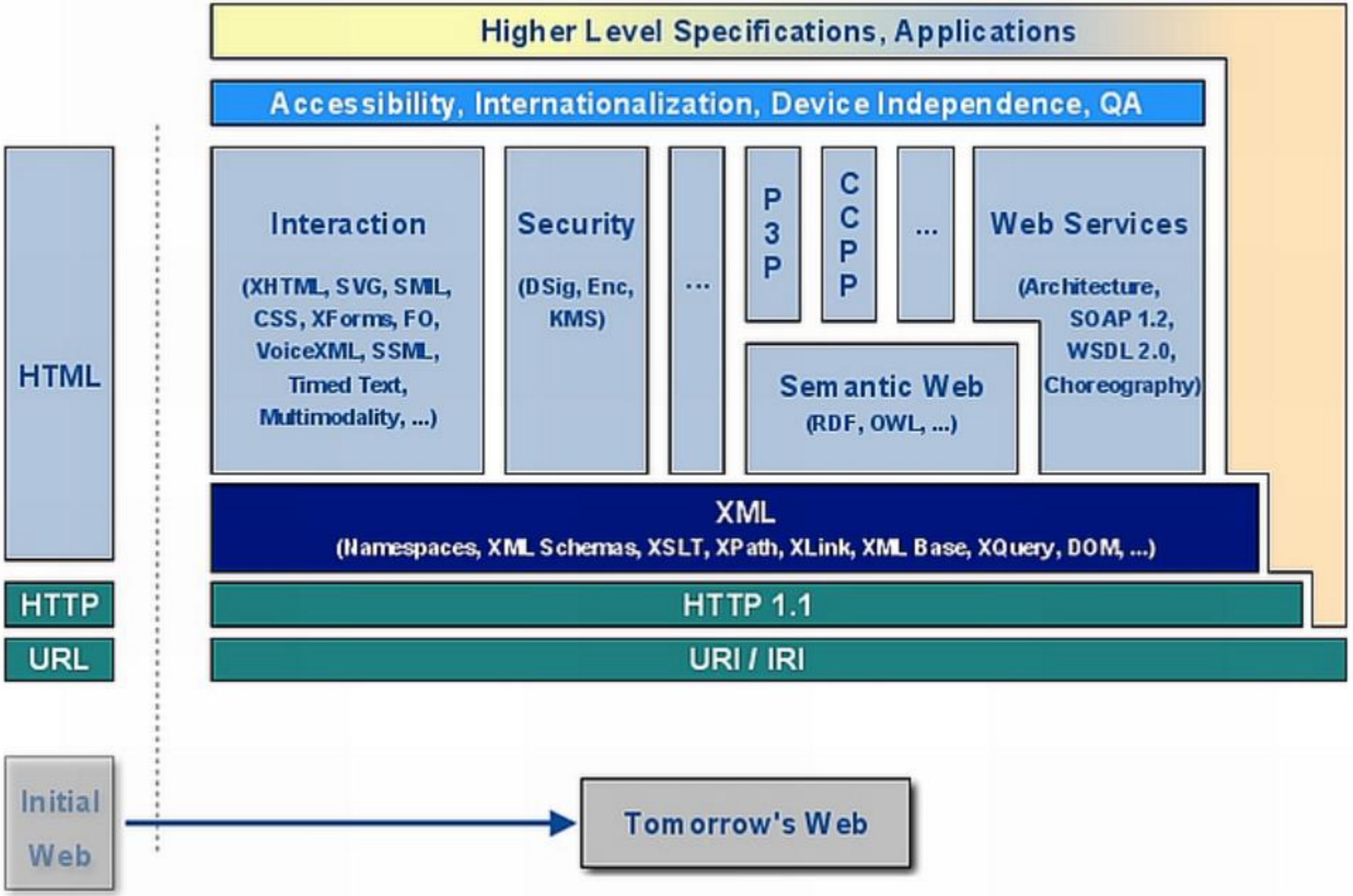
And W3C also creates a **TECHNOLOGY STACK** that meets the **STANDARDS.**

General Goals of W3C



Goals of W3C (under the hood)

- Prevent the Web from breaking apart
- Grant interoperability
- Make sure that the web is a creative space
- Maintain extensibility
- Lead the web to its full potential


Higher Level Specifications, Applications
Accessibility, Internationalization, Device Independence, QA
HTML
Interaction
(XHTML, SVG, SMIL, CSS, XForms, FO, VoiceXML, SSML, Timed Text, Multimodality, ...)
Security
(DSig, Enc, KMS)
...
P3P
CCPP
...
Web Services
(Architecture, SOAP 1.2, WSDL 2.0, Choreography)
Semantic Web
(RDF, OWL, ...)
XML
(Namespaces, XML Schemas, XSLT, XPath, XLink, XML Base, XQuery, DOM, ...)
HTTP
HTTP 1.1
URL
URI / IRI
Initial Web
Tomorrow's Web

WEB 3.0 & new marketing world

Web 1.0

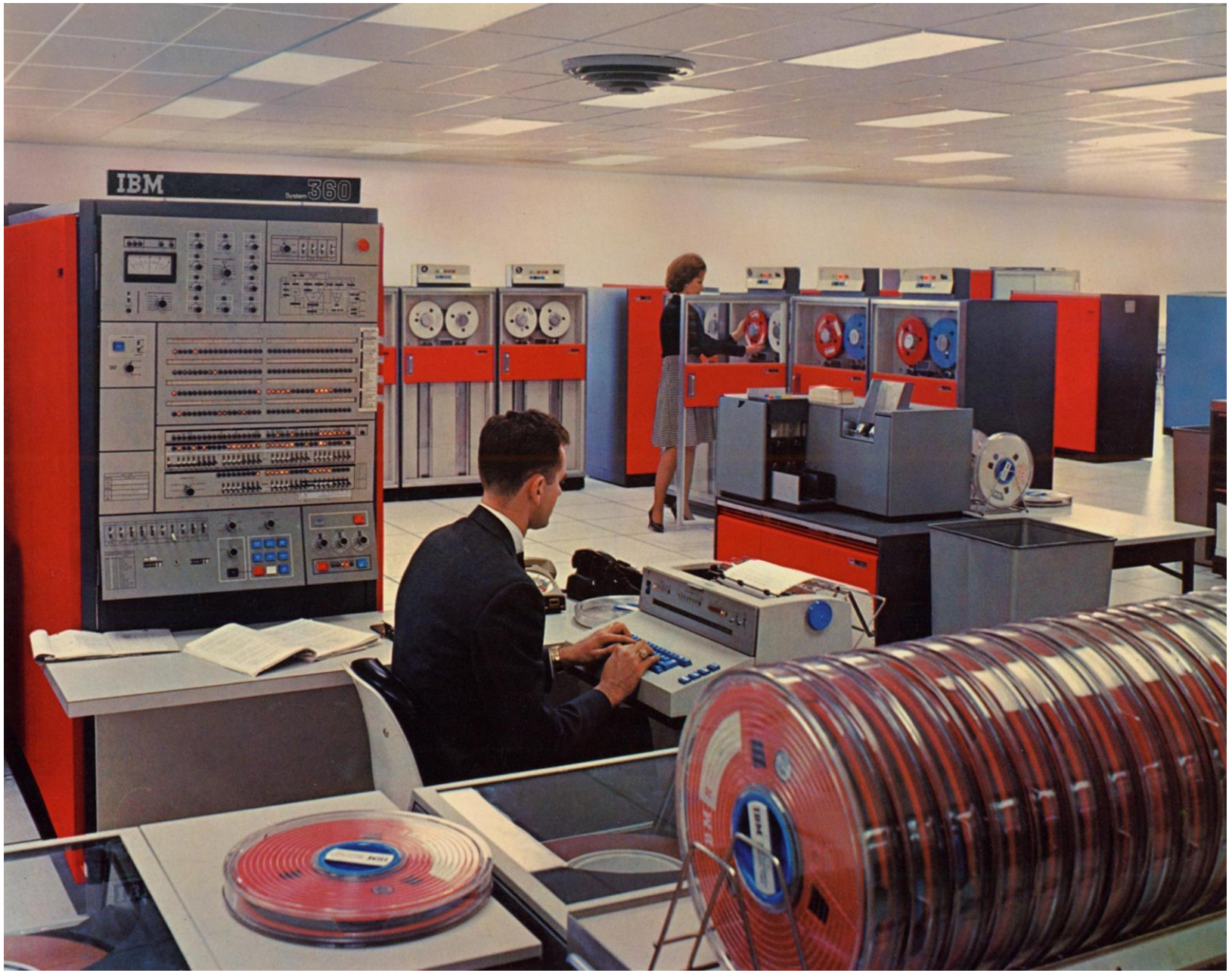
"Read Only",
Decentralized

Web 2.0

Participatory,
Centralized

Web 3

No Intermediaries,
Decentralized

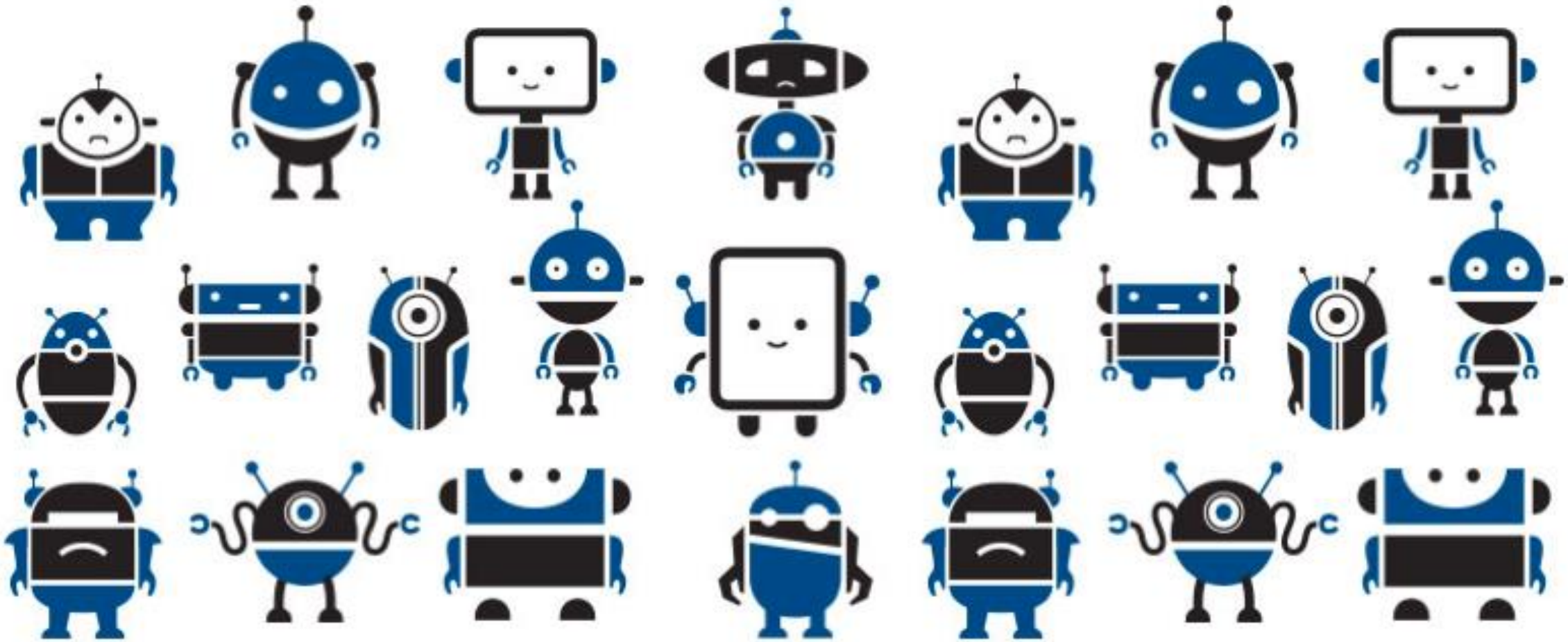


IBM


GIGABYTE


SMS

amazon




WELCOME TO
METAVERSE


Services
Projects
Volunteering
About
Contact
Search The Fed...
The Fed's Accessible City Guide
Read More
What is the Fed? The Fed is a user-led organisation that promotes independent living for all. It works towards equality by inspiring disabled people to identify barriers and define solutions.
How can we help you?
We offer a range of services to help you live independently. Our services include:


Desktop
Tablet
Mobile
Laptop

Web Design Criteria

Scalability

Progressive Enhancement

Responsive Design

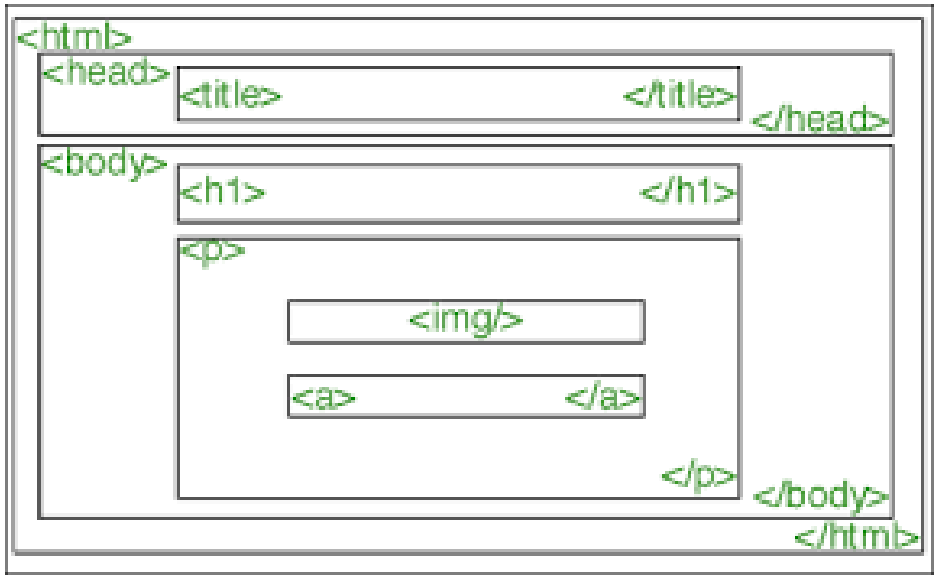
Accessibility (Vision, Mobility, Auditory, or Cognitive impairment)

Site Optimization (Speed Optimization)

Smallest Image File, Minimize HTML/CSs, Minimum Javascript, load script in parallel, Reduce Requests, Don’t Download Unnecessary Files

Creating First Page

SECTION 7



HTML Page Structure

- That's all you need for a basic web page!

```
<!DOCTYPE HTML PUBLIC
   "-//W3C//DTD HTML 4.01//EN"
     "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>My Title</title>
  </head>
  <body>
    <p>This is my first web page.</p>
  </body>
<footer>
 <p>This is footer section of page.</p>
</footer>
</html>
```

HTML <tag> and Elements

What is a tag?

- A tag is a boundary mark. It is used to represent some features, as markup for contents for an element or as an element.
- Tag for a feature (also called empty element):
 for 1 carriage return character as break of a line, <wbr> for 1 word break,. <hr> thematic break
- Tag for an element (also called empty element) definition: <img>, <meta>
- Tag as a markup for elements: <html></html>, <body></body>,<head></head>,<title></title>,<h1></h1>, ,
- An element is some text content in the HTML page
- <!-- put your comments here --> is a comment which is hidden from the html page view for users.

A Web Page Step by Step

Step 1	start with content
Step 2	give the document structure
Step 3	Identify text elements
Step 4	Add an image
Step 5	Change the page appearance with a style sheet

Step 1: Starts with Contents

EXERCISE 4-1. Entering content

1. Type the home page content below into the new document in your text editor. Copy it exactly as you see it here, keeping the line breaks the same for the sake of playing along. The raw text for this exercise is also available online at *learningwebdesign.com/5e/materials/*.

```
Black Goose Bistro

The Restaurant
The Black Goose Bistro offers casual lunch and dinner fare in a relaxed
atmosphere. The menu changes regularly to highlight the freshest local
ingredients.

Catering
You have fun. We'll handle the cooking. Black Goose Catering can handle
events from snacks for a meetup to elegant corporate fundraisers.

Location and Hours
Seekonk, Massachusetts;
Monday through Thursday 11am to 9pm; Friday and Saturday, 11am to
midnight
```

2. Select "Save" or "Save as" from the File menu to get the Save As dialog box (FIGURE 4-4). The first thing you need to do is create a new folder (click the New Folder button on both Windows and Mac) that will contain all of the files for the site. The technical name for the folder that contains everything is the local root directory.

Step 2: give the document structure

EXERCISE 4-2. Adding minimal structure

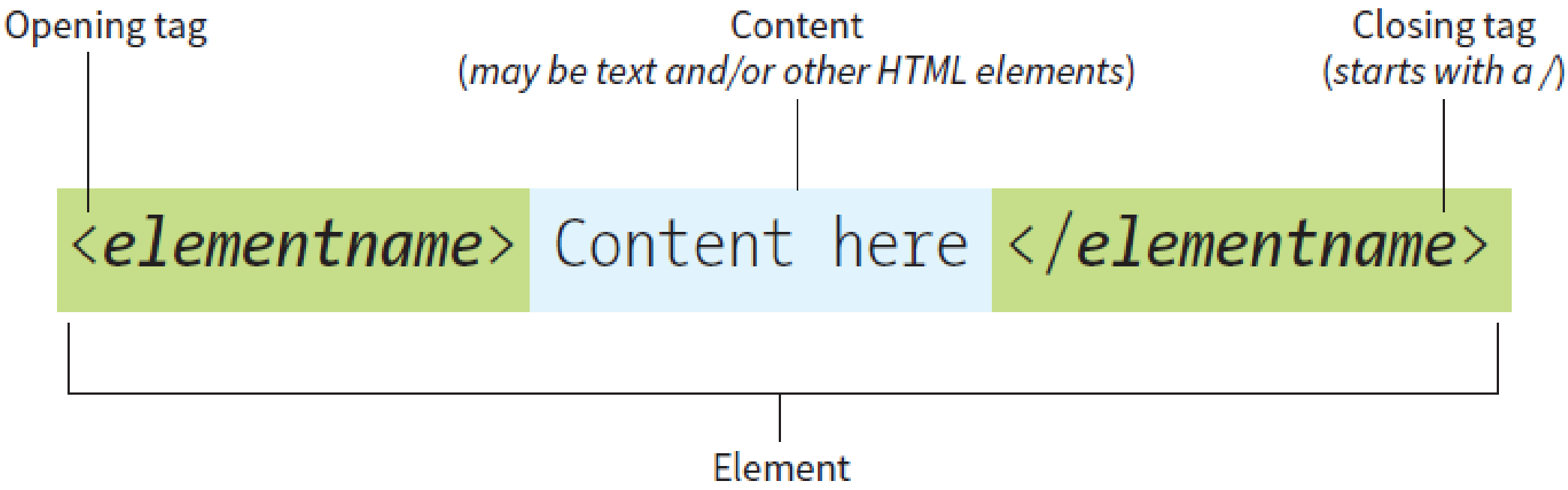
1. Open the new *index.html* document if it isn't open already and add the DOCTYPE declaration:

```
<!DOCTYPE html>
```

2. Put the entire document in an HTML root element by adding an **<html>** start tag after the DOCTYPE and an **</html>** end tag at the very end of the text.
3. Next, create the document head that contains the title for the page. Insert **<head>** and **</head>** tags before the content. Within the **head** element, add information about the character encoding **<meta charset="utf-8">**, and the title, "Black Goose Bistro", surrounded by opening and closing **<title>** tags.
4. Finally, define the body of the document by wrapping the text content in **<body>** and **</body>** tags. When you are done, the source document should look like this (the markup is shown in color to make it stand out):

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>Black Goose Bistro</title>
</head>

<body>
Black Goose Bistro

The Restaurant
The Black Goose Bistro offers casual lunch and
dinner fare in a relaxed atmosphere. The menu
changes regularly to highlight the freshest local
ingredients.

Catering
You have fun. We'll handle the cooking. Black
Goose Catering can handle events from snacks for a
meetup to elegant corporate fundraisers.

Location and Hours
Seekonk, Massachusetts;
Monday through Thursday 11am to 9pm; Friday and
Saturday, 11am to midnight
</body>
</html>
```

5. Save the document in the *bistro* directory, so that it overwrites the old version. Open the file in the browser or hit Refresh or Reload if it is open already. FIGURE 4-9 shows how it should look now.



Example: `<h1>Black Goose Bistro</h1>`

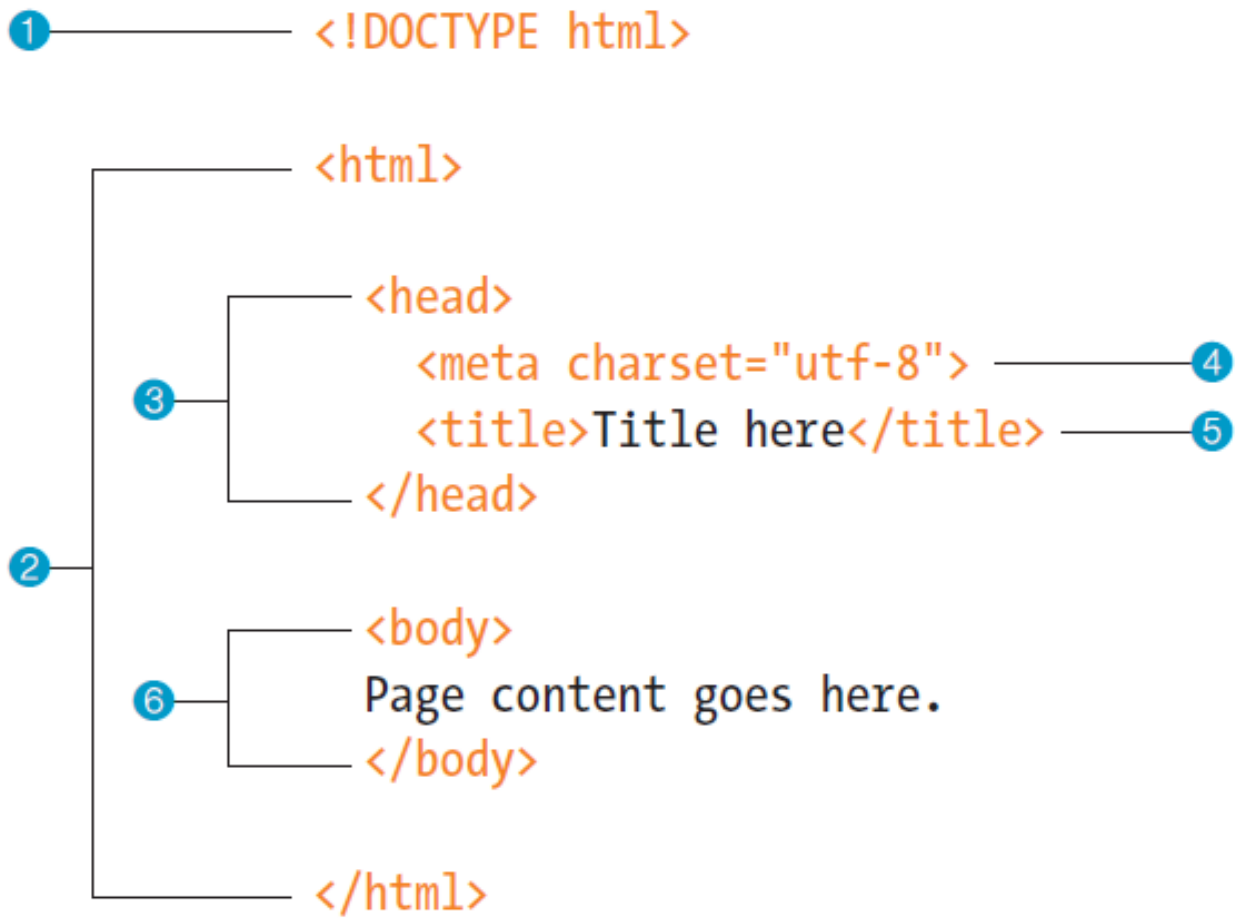


FIGURE 4-8. The minimal structure of an HTML document includes **head** and **body** contained within the **html** root element.

Step 3: Identify text elements

EXERCISE 4-3. Defining text elements

1. Open the document *index.html* in your text editor, if it isn't open already.
2. The first line of text, "Black Goose Bistro," is the main heading for the page, so we'll mark it up as a Heading Level 1 (**h1**) element. Put the opening tag, **<h1>**, at the beginning of the line and the closing tag, **</h1>**, after it, like this:

```
<h1>Black Goose Bistro</h1>
```

3. Our page also has three subheads. Mark them up as Heading Level 2 (**h2**) elements in a similar manner. I'll do the first one here; you do the same for "Catering" and "Location and Hours."

```
<h2>The Restaurant</h2>
```

4. Each **h2** element is followed by a brief paragraph of text, so let's mark those up as paragraph (**p**) elements in a similar manner. Here's the first one; you do the rest:

```
<p>The Black Goose Bistro offers casual lunch and dinner fare in
a relaxed atmosphere. The menu changes regularly to highlight the
freshest local ingredients.</p>
```

5. Finally, in the Catering section, I want to emphasize that visitors should just leave the cooking to us. To make text emphasized, mark it up in an emphasis element (**em**) element, as shown here:

```
<p>You have fun. <em>We'll handle the cooking.</em> Black Goose
Catering can handle events from snacks for a meetup to elegant
corporate fundraisers.</p>
```

6. Now that we've marked up the document, let's save it as we did before, and open (or reload) the page in the browser. You should see a page that looks much like the one in FIGURE 4-10. If it doesn't, check your markup to be sure that you aren't missing any angle brackets or a slash in a closing tag.

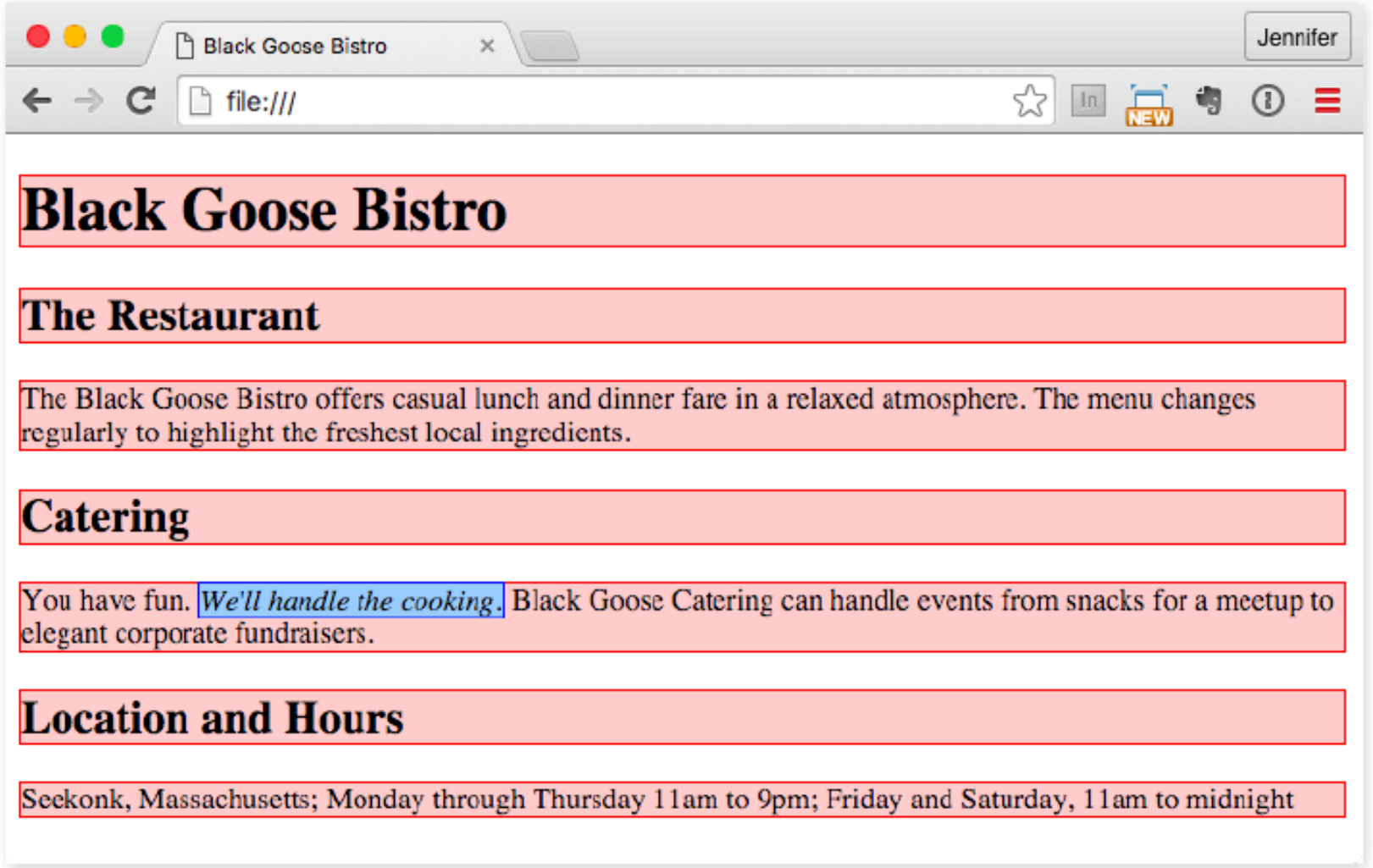


FIGURE 4-11. The outlines show the structure of the elements in the home page.

Step 4: Add an image

EXERCISE 4-4. Adding an image

1. If you're working along, the first thing you'll need to do is get a copy of the image file on your hard drive so you can see it in place when you open the file locally. The image file is provided in the materials for this chapter (*learningwebdesign.com/5e/materials*). You can also get the image file by saving it right from the sample web page online at *learningwebdesign.com/5e/materials/ch04/bistro*. Right-click (or Control-click on a Mac) the goose image and select "Save to disk" (or similar) from the pop-up menu, as shown in FIGURE 4-14. Name the file *blackgoose.png*. Be sure to save it in the *bistro* folder with *index.html*.
2. Once you have the image, insert it at the beginning of the first-level heading by typing in the **img** element and its attributes as shown here:

```
<h1><img src="blackgoose.png" alt="logo">Black Goose Bistro</h1>
```

   The **src** attribute provides the name of the image file that should be inserted, and the **alt** attribute provides text that should be displayed if the image is not available. Both of these attributes are required in every **img** element.
3. I'd like the image to appear above the title, so add a line break (**br**) after the **img** element to start the headline text on a new line.

```
<h1><img src="blackgoose.png" alt="logo"><br>Black Goose Bistro</h1>
```

4. Let's break up the last paragraph into three lines for better clarity. Drop a **
** tag at the spots you'd like the line breaks to occur. Try to match the screenshot in FIGURE 4-15.
5. Now save *index.html* and open or refresh it in the browser window. The page should look like the one shown in FIGURE 4-15. If it doesn't, check to make sure that the image file, *blackgoose.png*, is in the same directory as *index.html*. If it is, then check to make sure that you aren't missing any characters, such as a closing quote or bracket, in the **img** element markup.

Step 4: Add an image

<img src="bird.jpg" alt="photo of bird">

- img is the tag for image elements
- src is the source file URL location
- alt is the alternative text shown if image file is somehow not available (server down, file damaged, file removed, …)
- src and alt are attribute (or property) of an element. An element can have many properties (attribute), in DOM format, they can also be viewed as a variable or property for an element object.
- Do Ex 4.4

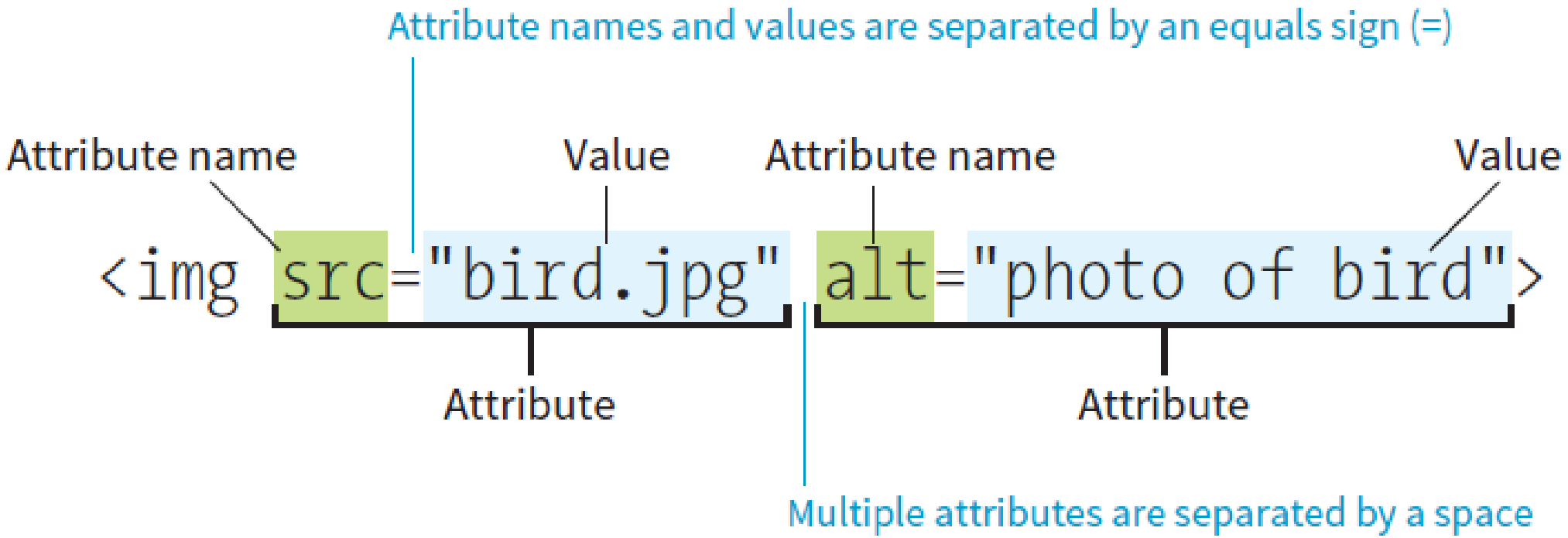


FIGURE 4-13. An `img` element with two attributes.

Step 5: Change the page appearance with a style sheet

EXERCISE 4-5. Adding a style sheet

1. Open *index.html* if it isn't open already. We're going to use the **style** element to apply a very simple embedded style sheet to the page. This is just one of the ways to add a style sheet; the others are covered in **Chapter 11, Introducing Cascading Style Sheets**.
2. The **style** element is placed inside the document **head**. Start by adding the **style** element to the document as shown here:

```
<head>
  <meta charset="utf-8">
  <title>Black Goose Bistro</title>
  <style>

  </style>
</head>
```

3. Next, type the following style rules within the **style** element just as you see them here. Don't worry if you don't know exactly what's going on (although it's fairly intuitive). You'll learn all about style rules in **Part III**.

```
<style>
body {
  background-color: #faf2e4;
  margin: 0 10%;
  font-family: sans-serif;
  }
h1 {
  text-align: center;
  font-family: serif;
  font-weight: normal;
  text-transform: uppercase;
  border-bottom: 1px solid #57b1dc;
  margin-top: 30px;
}
h2 {
  color: #d1633c;
  font-size: 1em;
}
</style>
```

4. Now it's time to save the file and take a look at it in the browser. It should look like the page in FIGURE 4-16. If it doesn't, go over the style sheet to make sure you didn't miss a semicolon or a curly bracket. Look at the way the page looks with our styles compared to the browser's default styles (FIGURE 4-15).

FIGURE 4-16. The Black Goose Bistro page after CSS style rules have been applied.

Step 5: Change the page appearance with a style sheet

```
body {
    background-color: #faf2e4;   /* in RGB format, R is #fa, G is #f2, B is #e4 */
    margin: 0 15%;   /* no top and bottom margin,  right and left margin 15% */
  }
h1 {
    text-align: center;  /* text aligned to center */
    font-family: serif;   /* font style serif */
    font-weight: normal;   /* not bold, not italics */
    text-transform: uppercase;  /* make all letters uppercase */
  }
```

Do Ex 4.5 and validate your page by loading to a browser !!!

Favicon and Unicode

SECTION 8

Adding Favicon

```
<!DOCTYPE html>
<html>
<head>
  <title>My Page Title</title>
  <link rel="icon" type="image/x-icon" href="/images/favicon.ico">
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Unicode

<meta charset="ISO-8859-1">

Character-set	Description
UTF-8	A character in UTF8 can be from 1 to 4 bytes long. UTF-8 can represent any character in the Unicode standard. UTF-8 is backwards compatible with ASCII. UTF-8 is the preferred encoding for e-mail and web pages
UTF-16	16-bit Unicode Transformation Format is a variable-length character encoding for Unicode, capable of encoding the entire Unicode repertoire. UTF-16 is used in major operating systems and environments, like Microsoft Windows, Java and .NET.