2

Web Application Essentials

Objectives

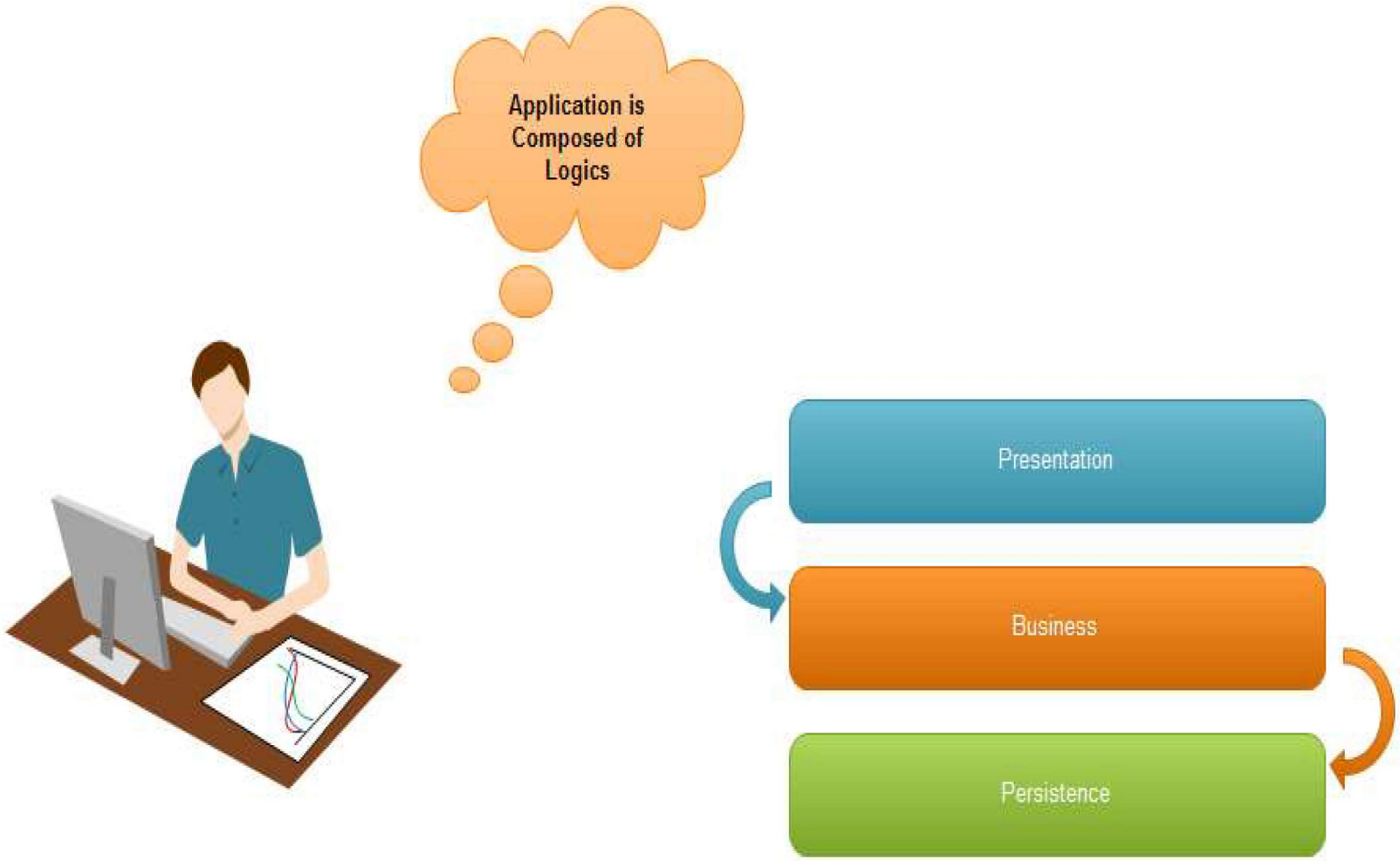
After completing this lesson, you should be able to:

- Create web applications
- Run HTML pages and analyze them by using the browser's tools
- Separate CSS and JavaScript content from HTML pages
- Run Web Applications

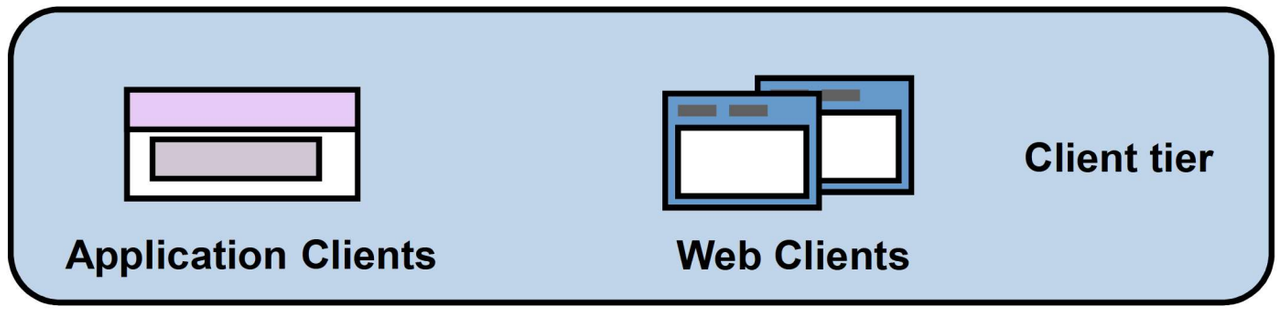
Topics

- Application Architectures
- Web servers / Middlewares
- HTTP protocol basics
- Web applications
  - Creating web applications
  - Web application components:
    - HTML files
    - Resources
    - Cascade Style Sheet files
    - JavaScript files

Application Development

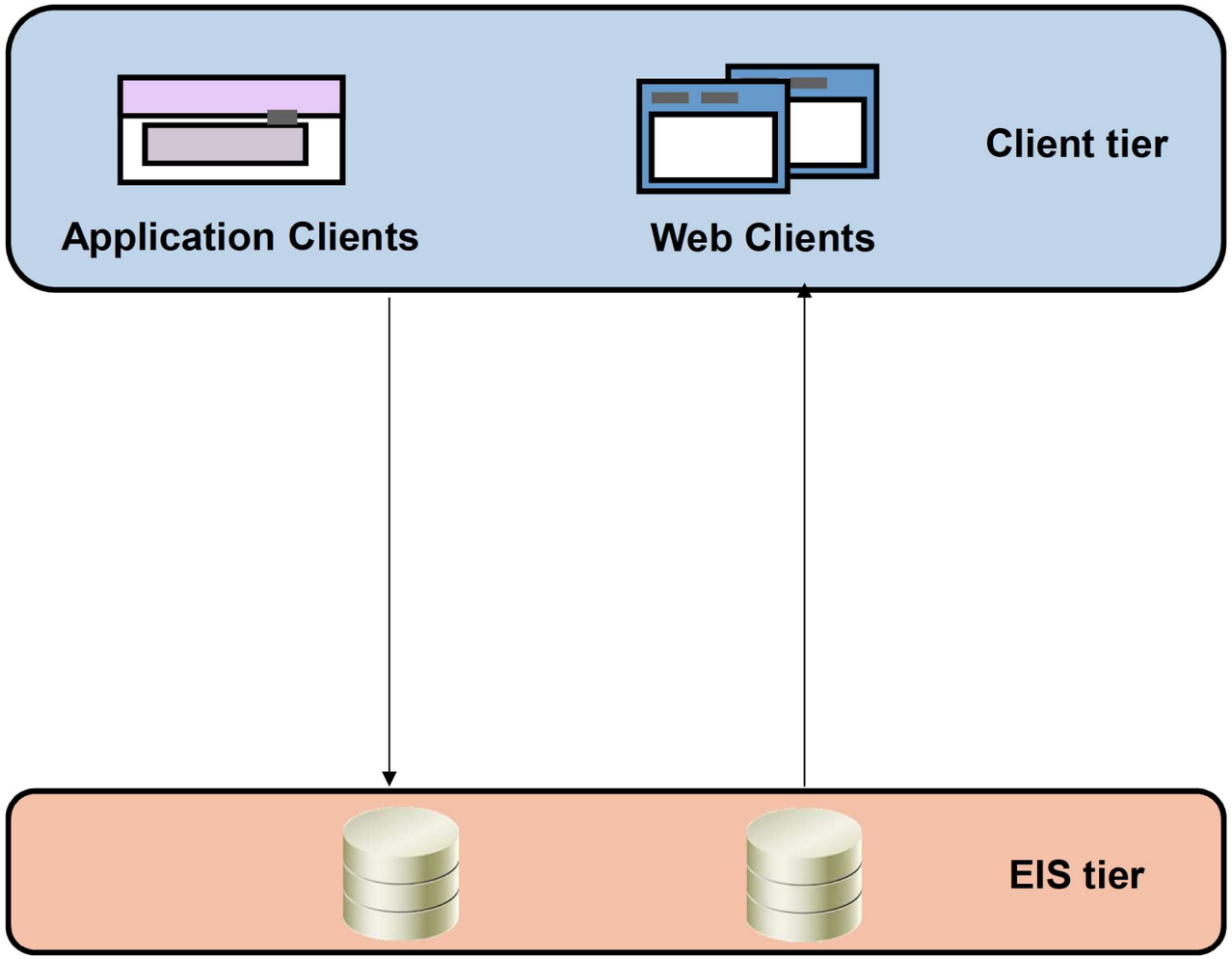


Single Tier Architecture

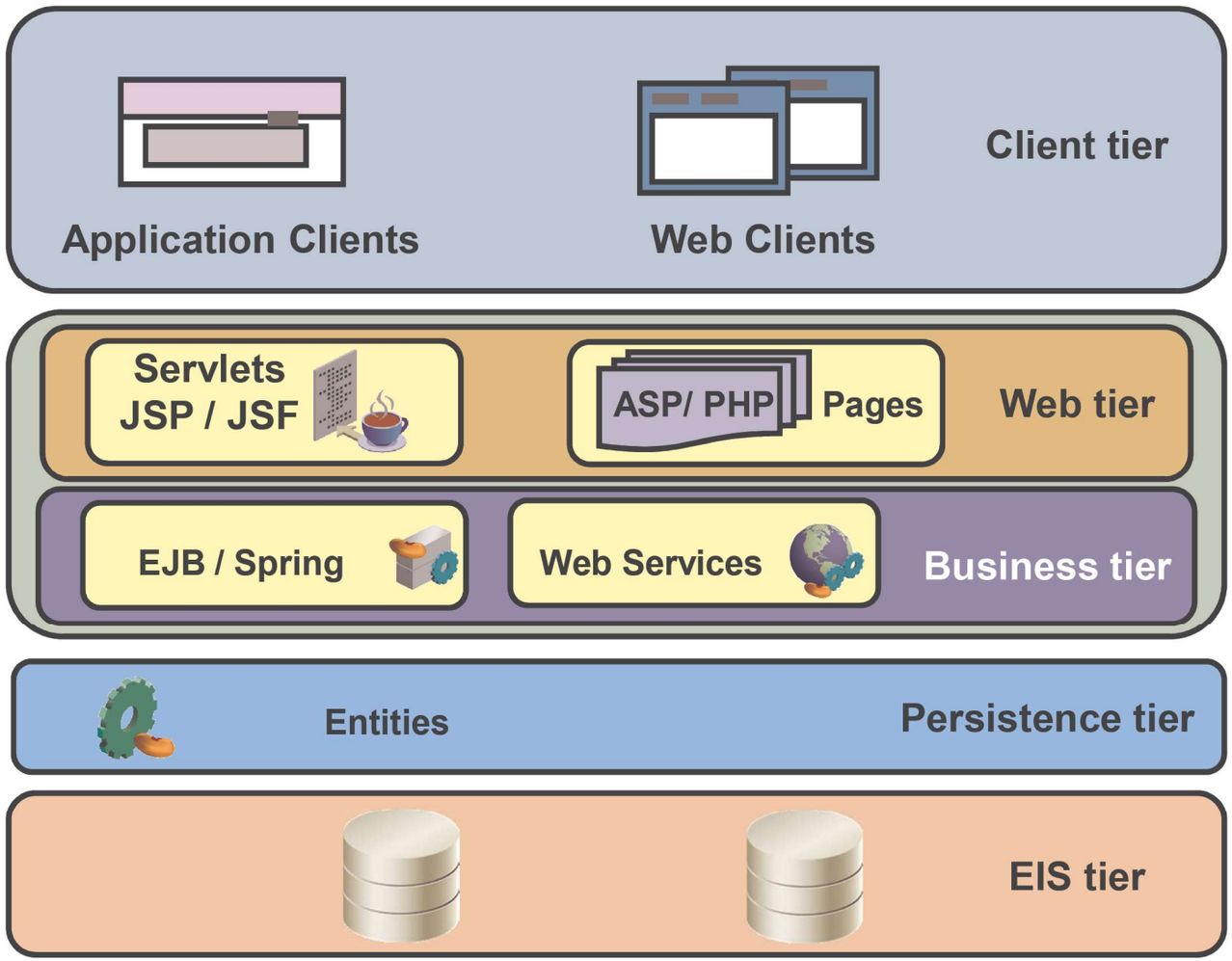


- Simple, Faster, Secured
- No N/W, Modification is Difficult, Data Lost is Lost

Two Tier Architecture



Distributed Multi Tier Architecture



Web Applications

Web applications come in many forms:

- Simple static web pages
- Single page applications
- Animated pages with JavaScript
- Interactive pages
- HTML5 games
- Forms to request user data

Examples of a web application:

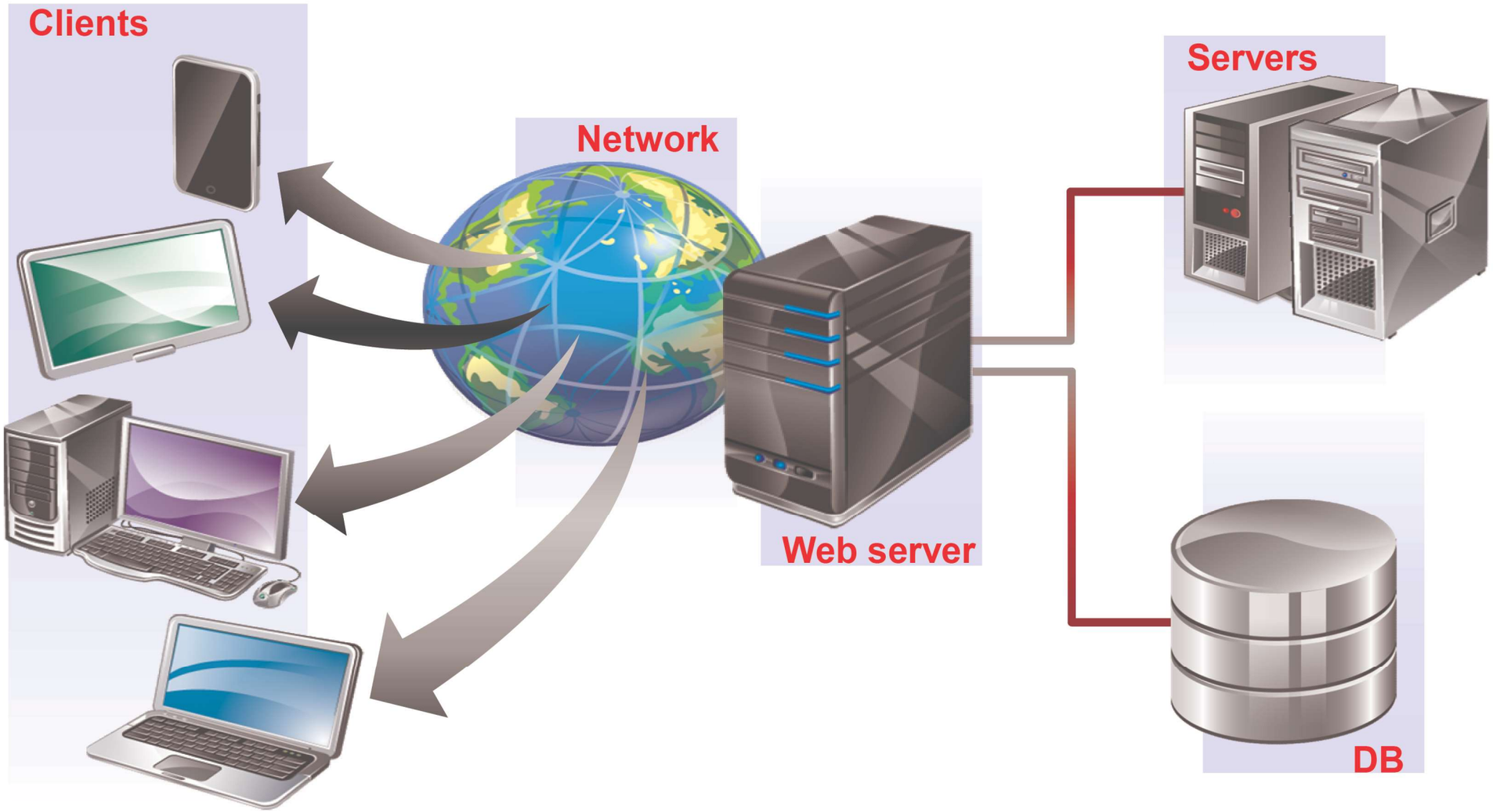
- Weather web page
- The Oracle web portal

Web Servers

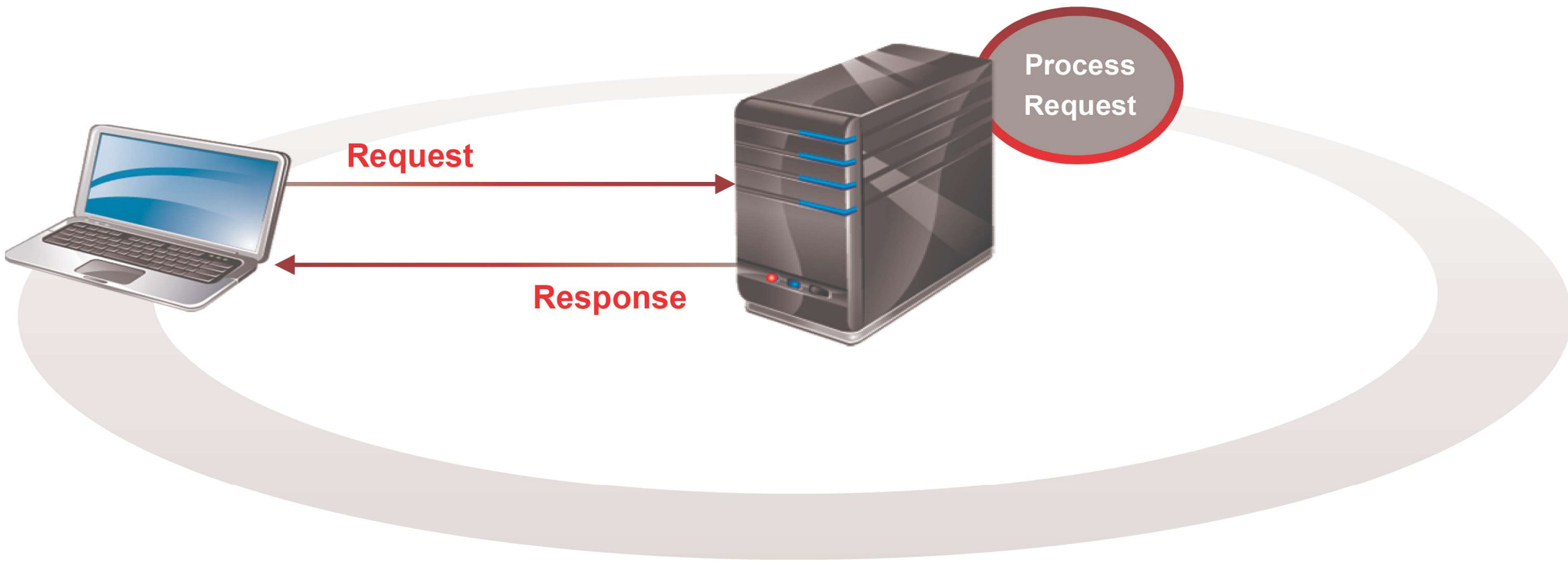
Web applications are usually stored in web servers. They:

- Handle web requests
- Store application files
- Provide access to resources

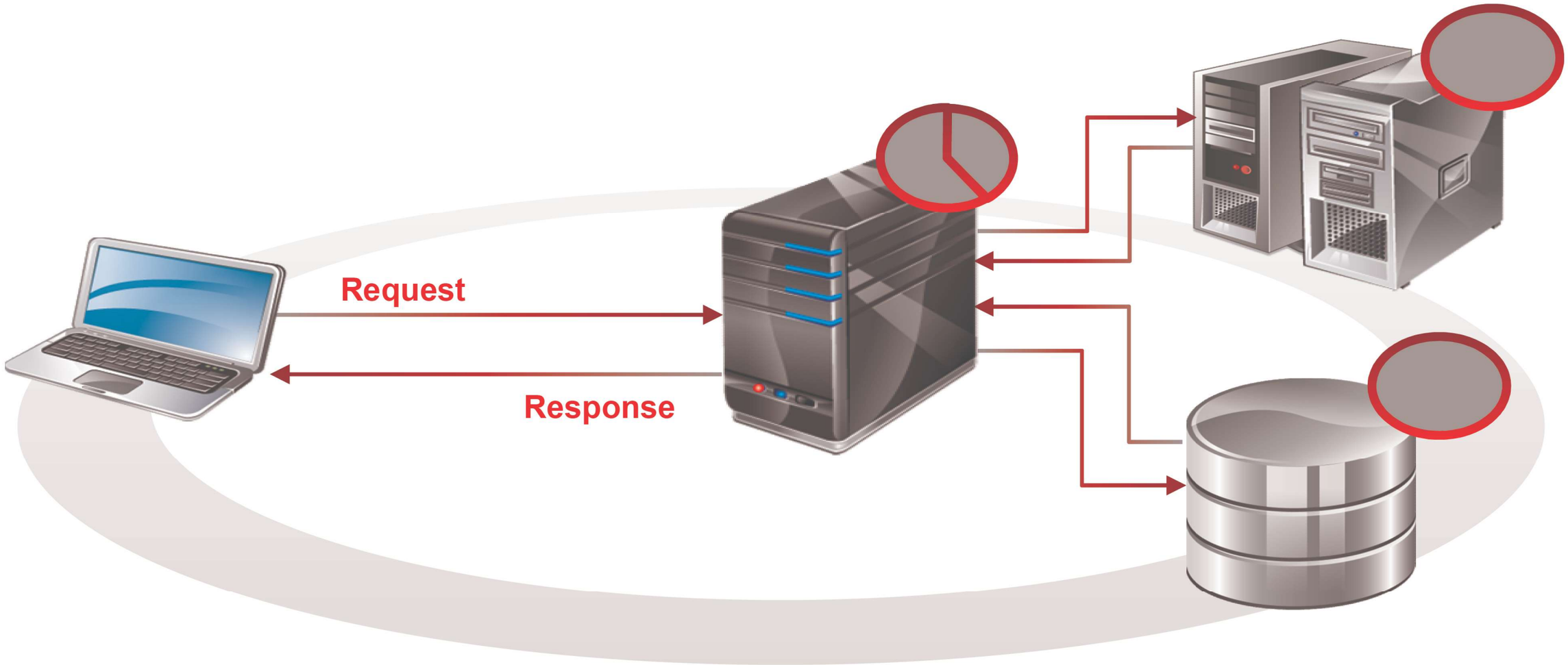
Web Architecture



How Web Servers Work



How Web Servers Work

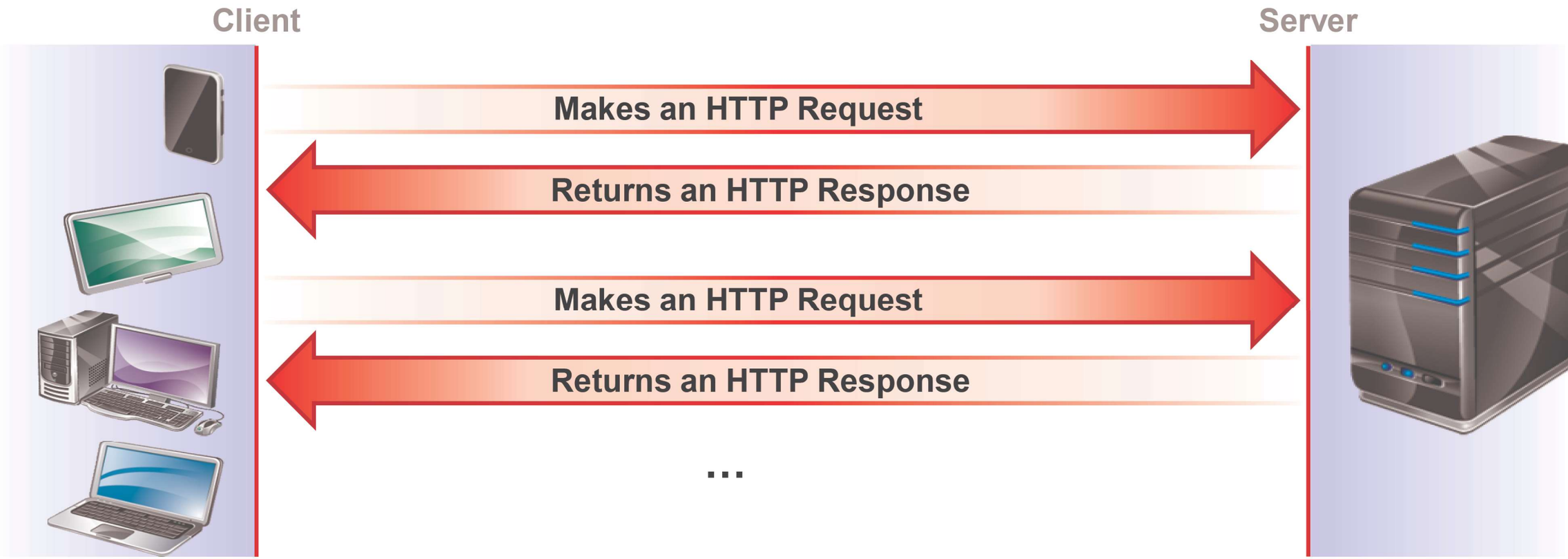


What Is a Client?

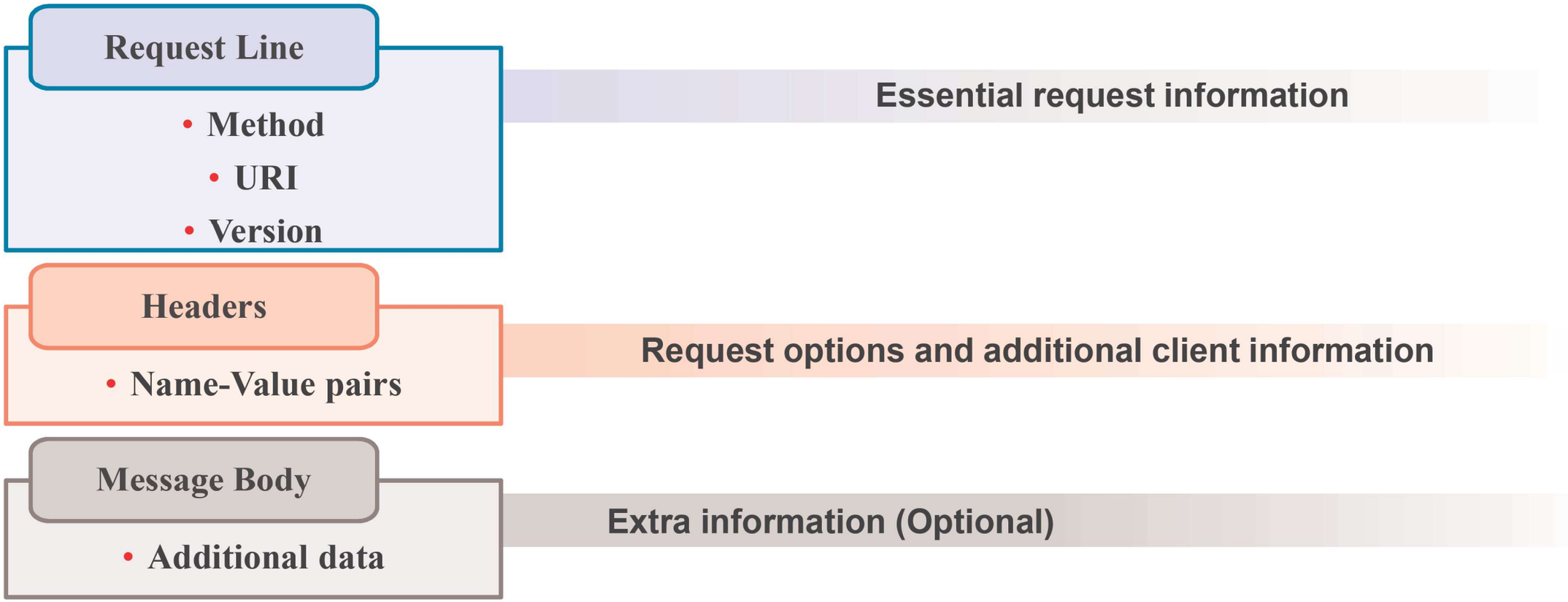
A client is an application that runs on a machine that can make HTTP requests and interpret HTTP responses.

- Web browsers
  - Firefox, Chrome, Internet Explorer, and Safari
- Other applications
  - Weather app on mobiles
  - News readers

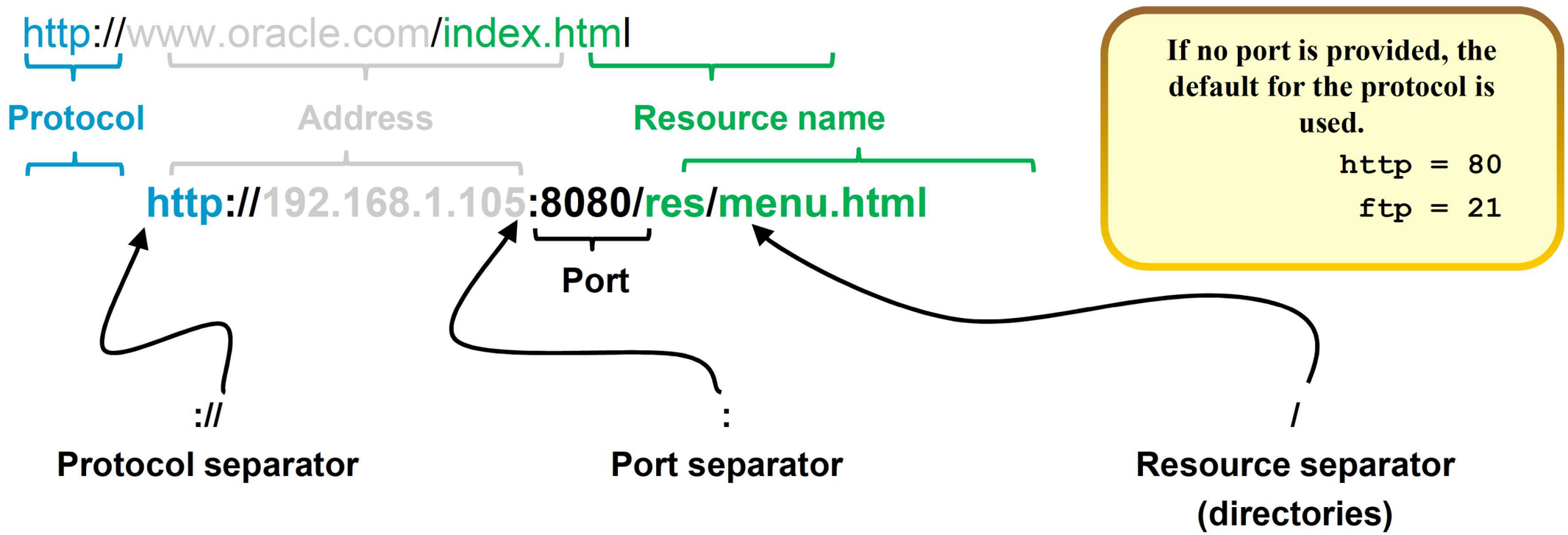
Clients communicate with the server by using the HTTP protocol.



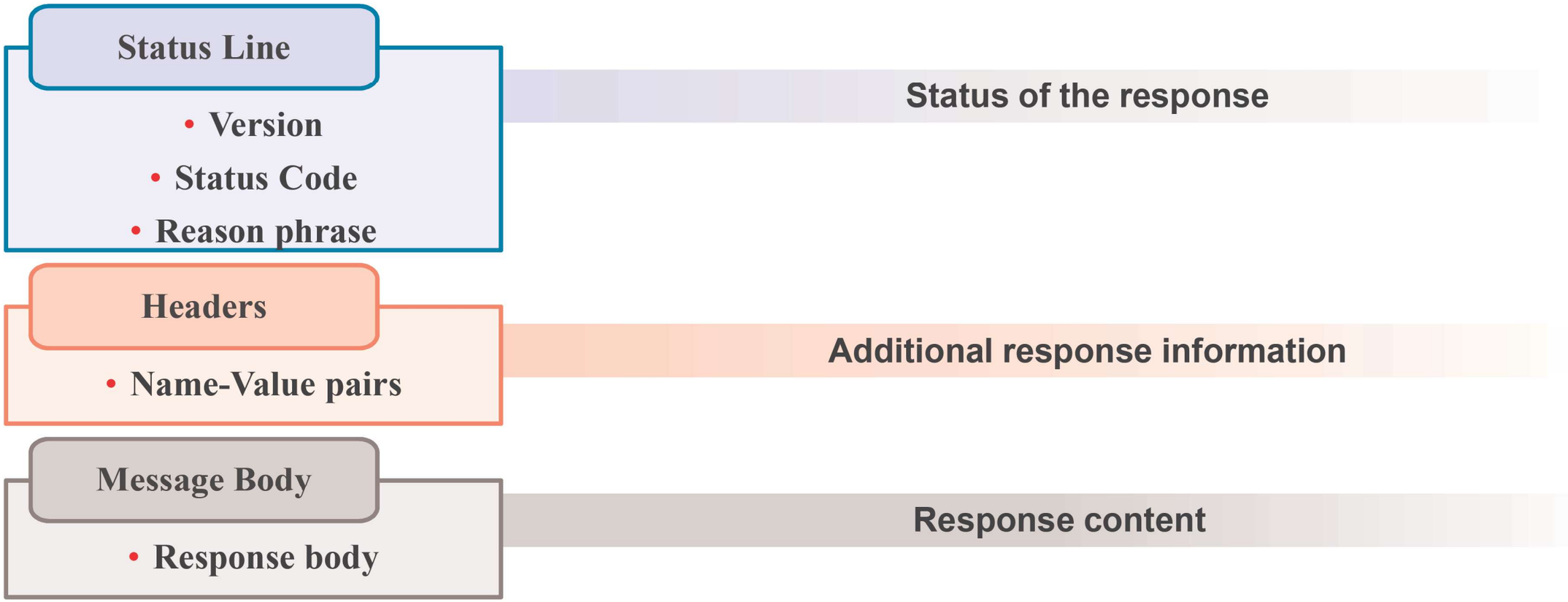
HTTP Request



HTTP Request: URL



HTTP Response



Response Bodies

A response body contains the content of a resource, including:

- Documents
- Images
- Audio
- Video
- JavaScript files

The client (web browser) usually knows how to handle or display the contents of the response.

The JavaScript code in web applications is run by the web browser on the client machine.

Web Applications

Technical Definition

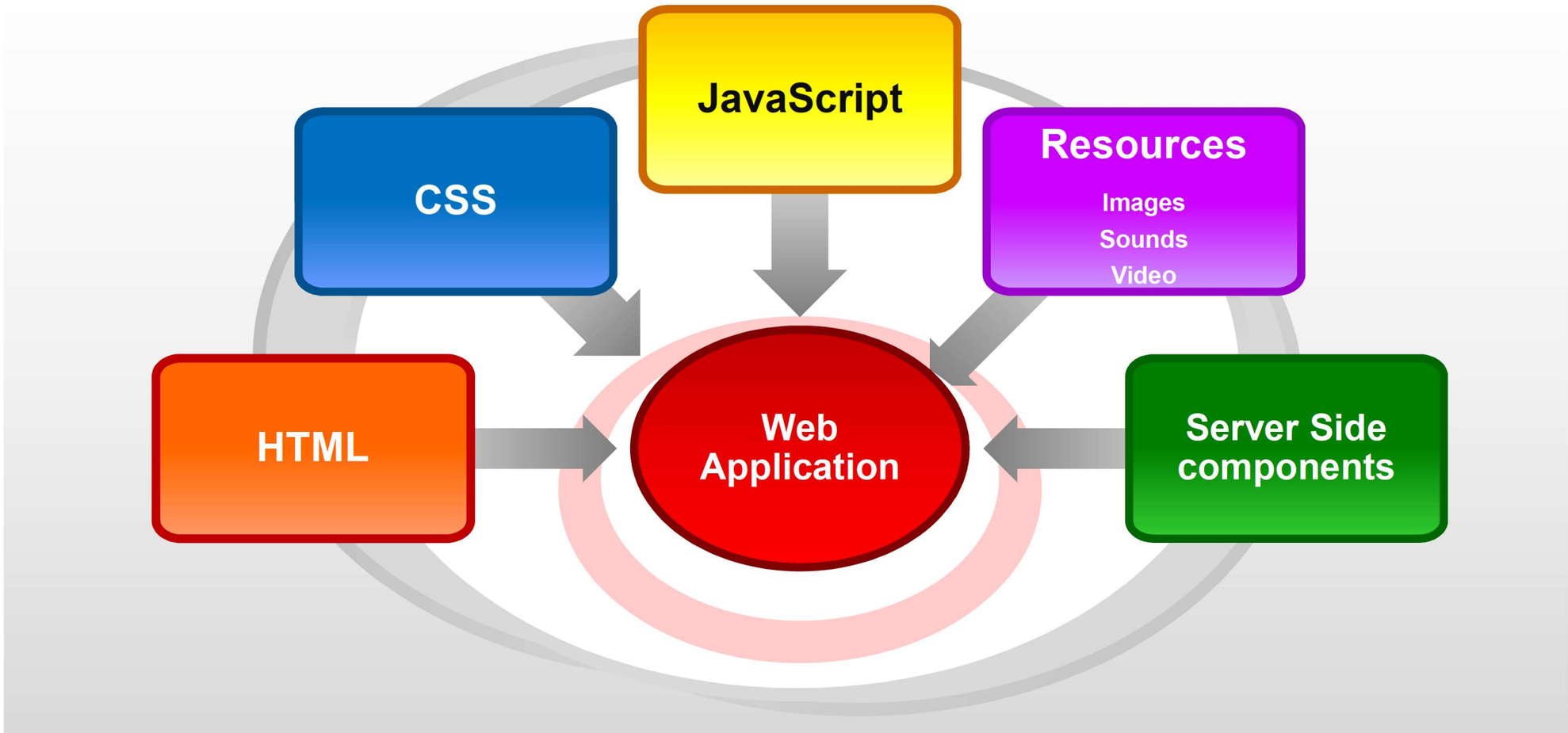
- A collection of files that are stored locally or in a web server that can run on a web browser

Conceptual Definition

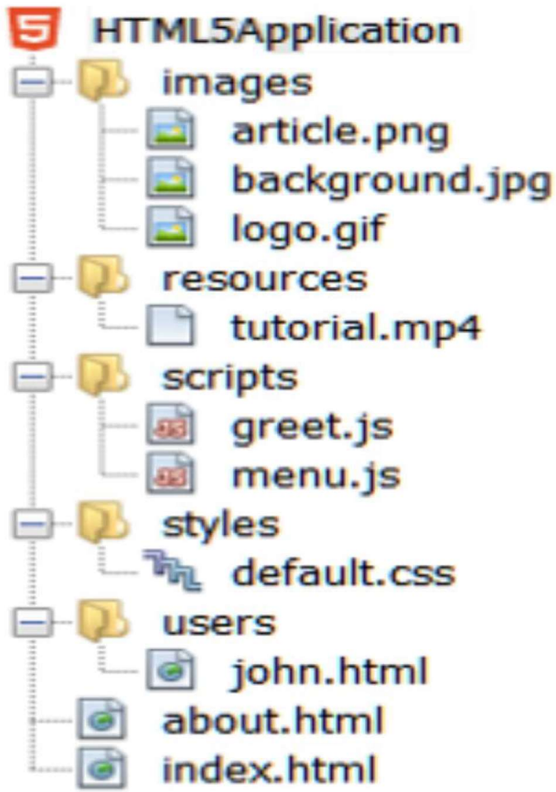
An application that runs on a web browser that:

- Provides the user a solution to a problem
- Is self-contained and focused
- Has a rich user interface
- Uses the capabilities of the user's device

Web Applications



Web Application Structure

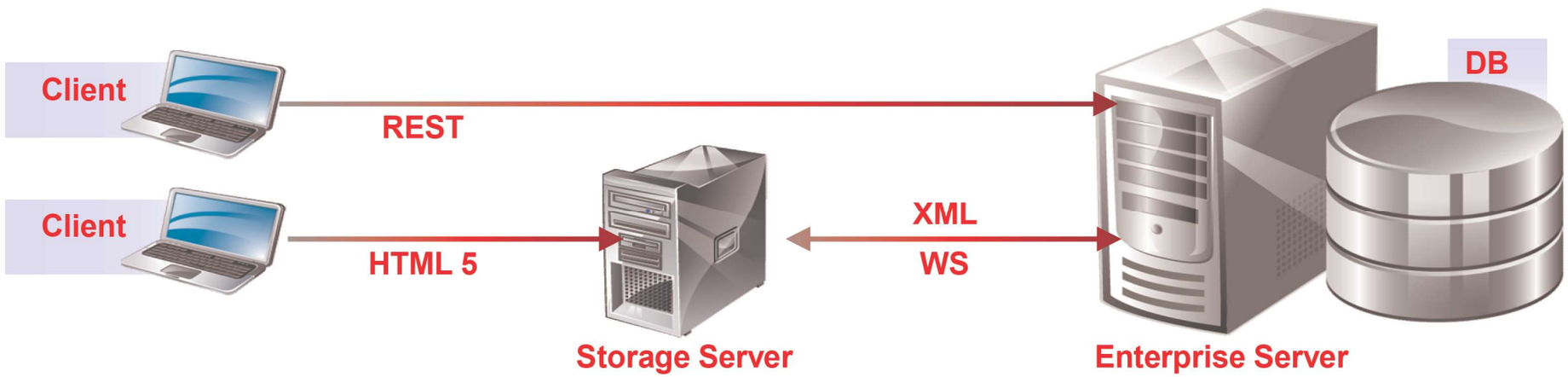


Website Versus Web Application

- Similarities:
  - Run on web browsers
  - Have HTML5, CSS3, JavaScript, and additional resources
  - Can be stored locally or on web servers
- A website
  - Is a collection of information organized in pages
  - Is navigation based
  - Does not have much interactivity
- A web application
  - Solves one problem at a time
  - Is interactive
  - Is self-contained
  - Is dynamic

The Server Side Web Application

- Generates dynamic content from data sources
- Generates dynamic pages, images, and resources
- Provides authentication, session management, and security
- Stores user and service information
- Provides XML and REST web services
- Provides AJAX and WebSocket server endpoints



Creating Web Applications

- Create a dedicated folder to store your application.
- Create the files for your application.
  - HTML
  - CSS
  - JavaScript
  - Images
- Organize the files in folders.

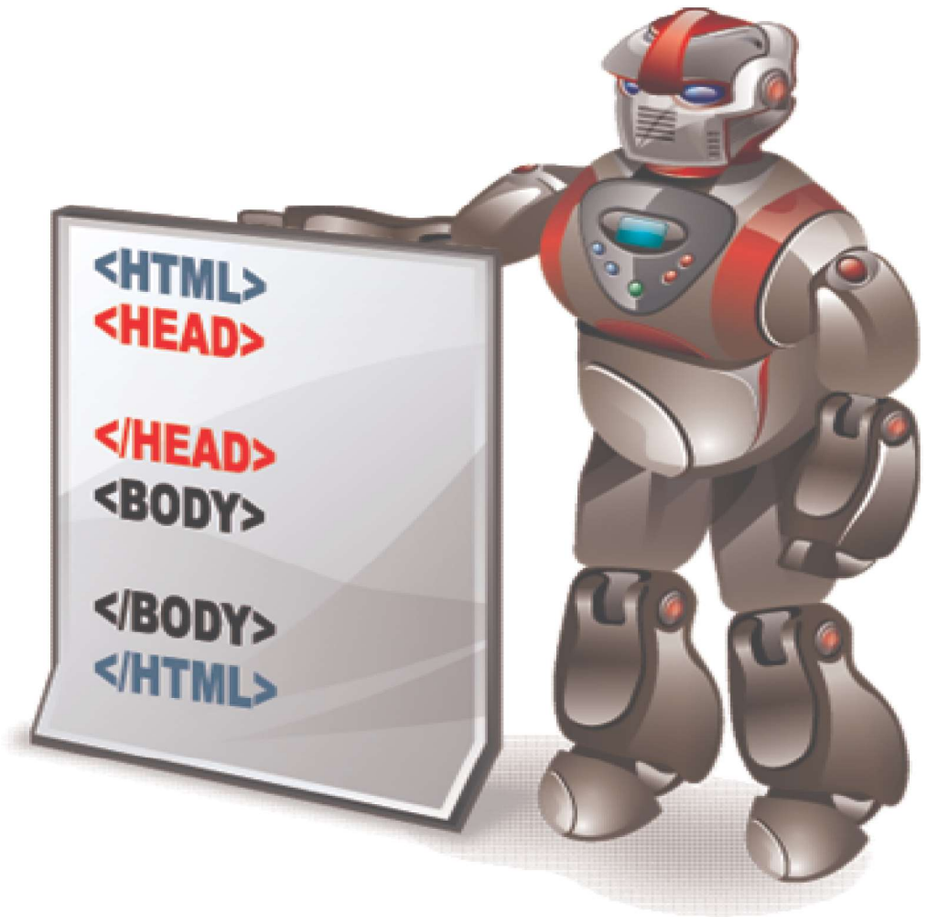
HTML Files

Hypertext Markup Language (HTML):

- It is a language used to define documents.
  - Structure, layout, and content
- Documents contain hyperlinks.
  - To navigate to other documents
  - To include resources

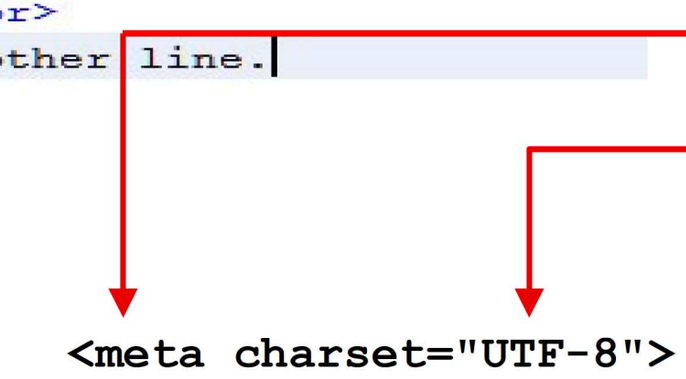
In a web application, HTML:

- Defines the page layout
- Links to other pages
- References other files
- Contains forms



HTML File Structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Title</h1>
    <p>Paragraph</p>
    Some text<br>
    Text in another line.
  </body>
</html>
```



HTML files are text files that define a document.

They contain tags. Tags are enclosed inside < >.

A tag has a Name, and might have:

- Attributes
- Body

HTML File Structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Title</h1>
    <p>Paragraph</p>
    Some text<br>
    Text in another line.
  </body>
</html>
```

HTML tags are closed by a </ > tag that matches the name.

What is inside the open and close tags is the **tag body.**

Some tags have no body, and may not be closed. You may close a tag with no body by using the `<TagName/>` notation.

HTML File Structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Title</h1>
    <p>Paragraph</p>
    Some text<br>
    Text in another line.
  </body>
</html>
```

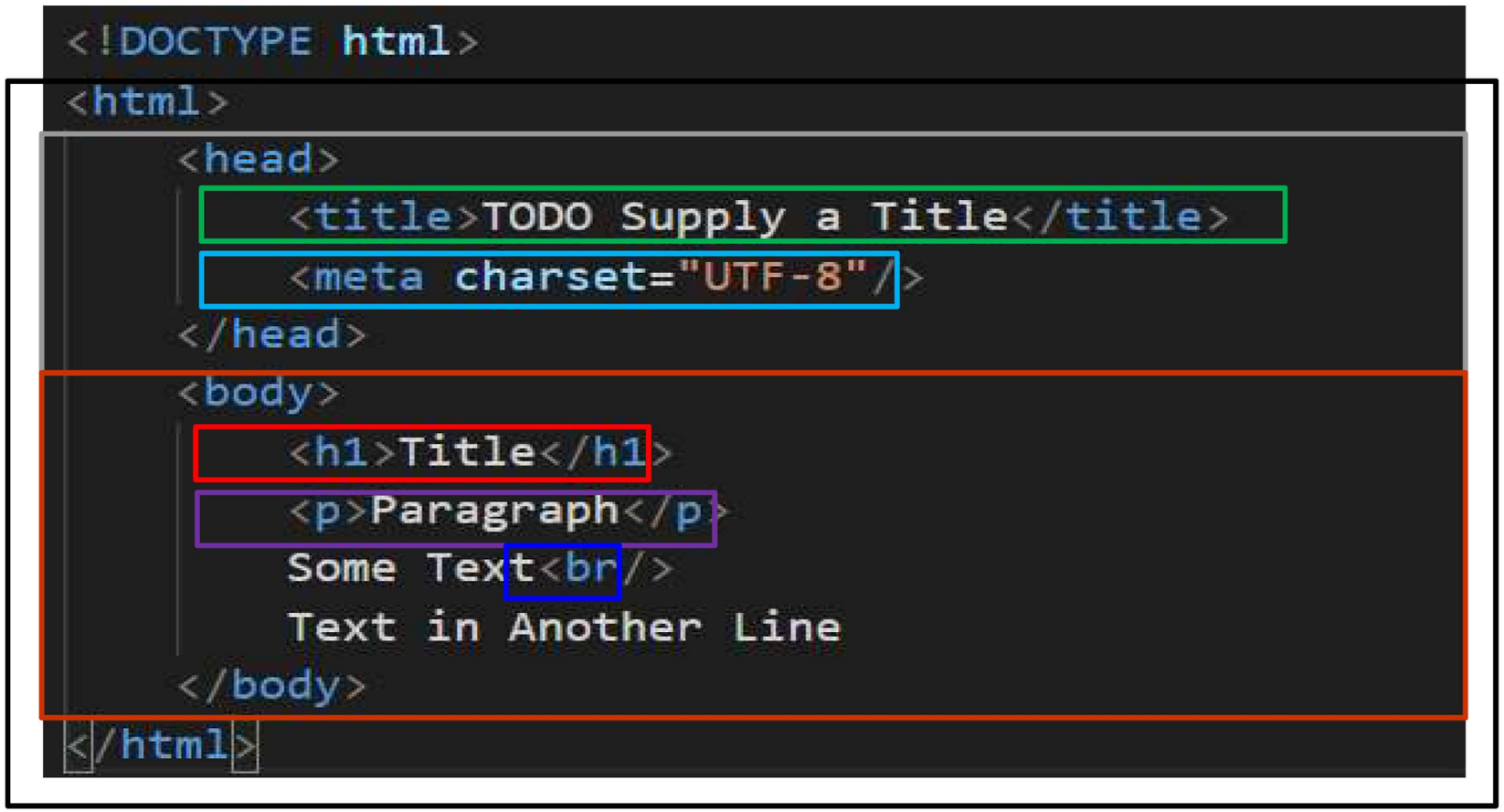
HTML5 files have a `<!DOCTYPE html>` directive.

The whole document is enclosed in an HTML tag.

An HTML document has two sections:

- The head: Contains information about the document
- The body: Contains the content of the document

HTML Tag Scope



HTML Head

Defines document properties:

- Title
- Encoding

- **Viewport size**
  - **Styles**

```
<head>
    <title>Hello Example</title>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <style>

    </style>
</head>
```

HTML Body

Contains the document content, including:

- Text
- Paragraphs
- Areas (span, div)
- Forms
- Tables
- Scripts

```
<body>
    <h1> Hello </h1>
    <p> Hi, <span id="nameSpan">You</span> ! </p>
    <script>

    </script>

</body>
```

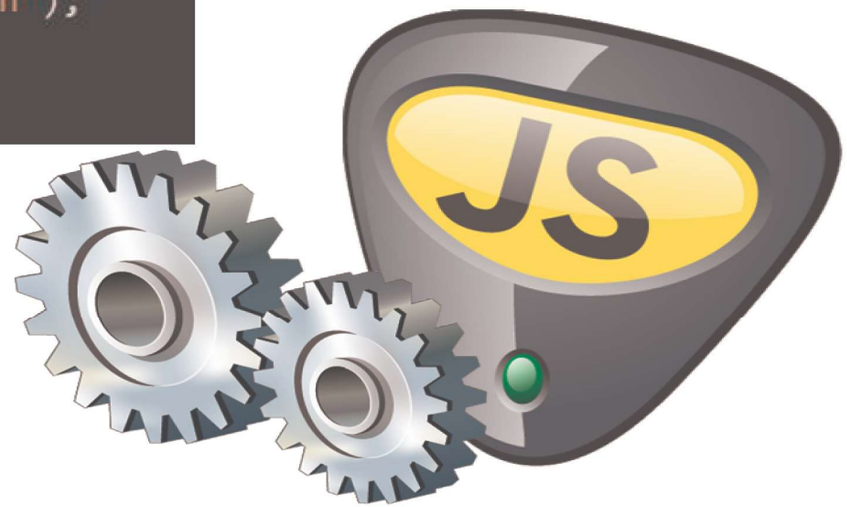
HTML Style (Embedded CSS)

- The contents of style tags form the Style Sheet for the document.
- It provides font type, colors, spacing, and display information.

```
<style>
    body {
        font-family: Arial, Helvetica, sans-serif;
        background-color: #d9e7f2;
        margin: 0px;
        border: 0px;
        padding: 0px;
    }
    h1 {
        color: #18466a;
    }
    #nameSpan {
        font-weight: bold;
    }
</style>
```

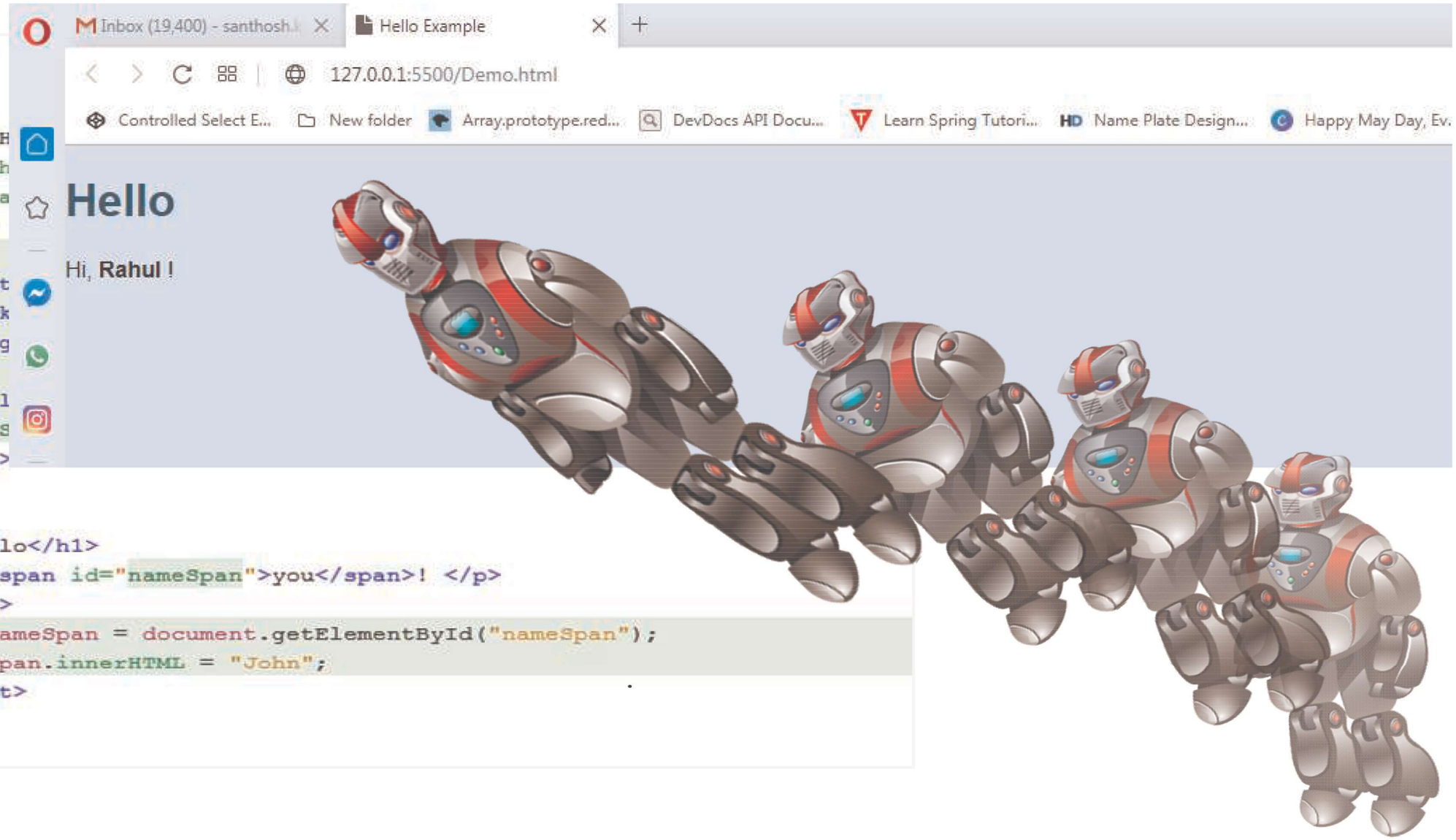
The contents of a script tag includes JavaScript code, which is run sequentially when the tag is reached.

```
<script>
    var nameSpan = document.getElementById("nameSpan");
    nameSpan.innerHTML = "Rahul";
</script>
```



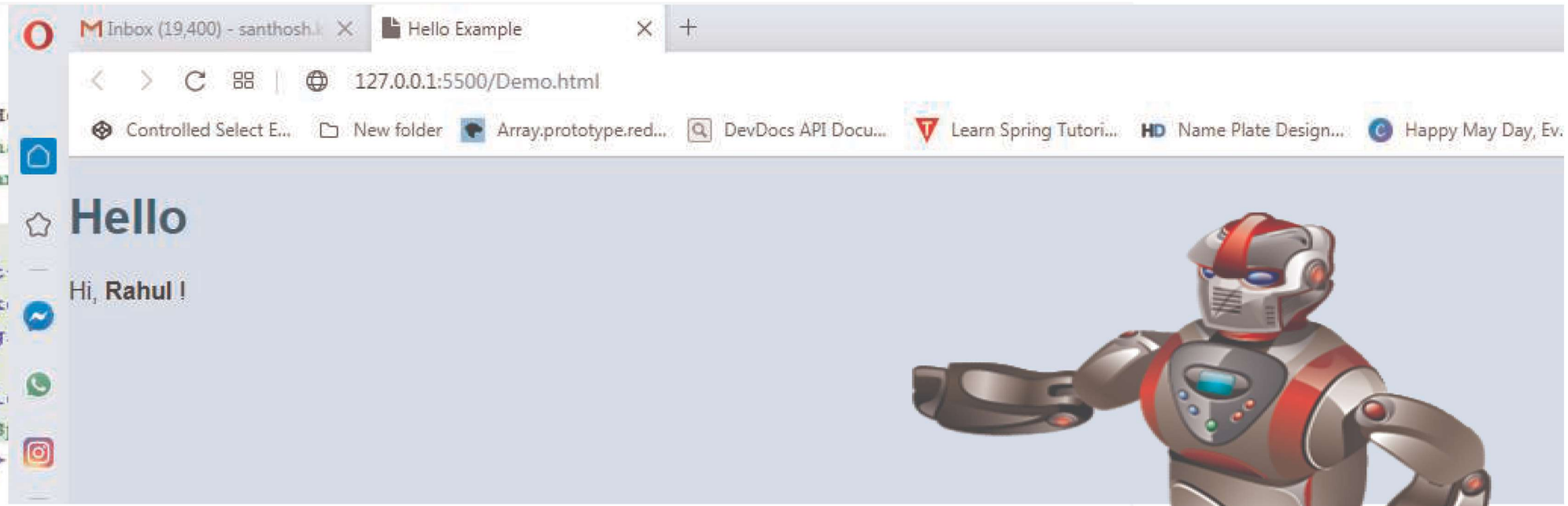
The Application in Action

```
<!DOCTYPE html>
<html>
    <head>
        <title>H
        <meta ch
        <meta na
        <style>
            body{
                font
                back
                marg
            }
            h1{col
            #names
        </style>
    </head>
    <body>
        <h1>Hello</h1>
        <p>Hi <span id="nameSpan">you</span>! </p>
        <script>
            var nameSpan = document.getElementById("nameSpan");
            nameSpan.innerHTML = "John";
        </script>
    </body>
</html>
```



The Application in Action

```
<!DOCTYPE html>
<html>
    <head>
        <title>H
        <meta ch
        <meta na
        <style>
            body{
                font
                back
                marg
            }
            h1{col
            #nameS
        </style>
    </head>
    <body>
        <h1>Hello</h1>
        <p>Hi <span id="nameSpan">you</span>! </p>
        <script>
            var nameSpan = document.getElementById("nameSpan");
            nameSpan.innerHTML = "John";
        </script>
    </body>
</html>
```



CSS Files

- A CSS file contains Cascade Style Sheet definitions that can be used inside a style tag.
- To prevent the HTML file from becoming too cluttered with styles and to be able to reuse them, use a separate file for CSS.

```
body{
   font-family: Arial;
   background-color: #d9e7f2;
   margin: 0px; border: 0px; padding: 0px;
}
h1{color: #18466a;}
#nameSpan{font-weight: bold;}
```

JS Files

- A JS file contains JavaScript code that can be inside a script tag.
- To prevent the HTML file from becoming too cluttered with scripts and to be able to reuse the code, you use a separate file for JavaScript.

```
var nameSpan = document.getElementById("nameSpan");
nameSpan.innerHTML = "John";
```

Resource Hyperlinking

An HTML file can reference resources to add them to the document.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello Example</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="styles/default.css" rel="stylesheet">
  </head>
  <body>
    <h1>Hello</h1>
    <p>Hi <span id="nameSpan">you</span>!</p>
    <script src="scripts/greet.js"></script>
  </body>
</html>
```

Absolute Paths

- To reference a resource in another server, start the path with the protocol.
  - http://www.commonstyles.com/styles/style.css
- To reference a resource in the same server, start the path with /.
  - /styles/style.css → http://localhost:8383/styles/style.css

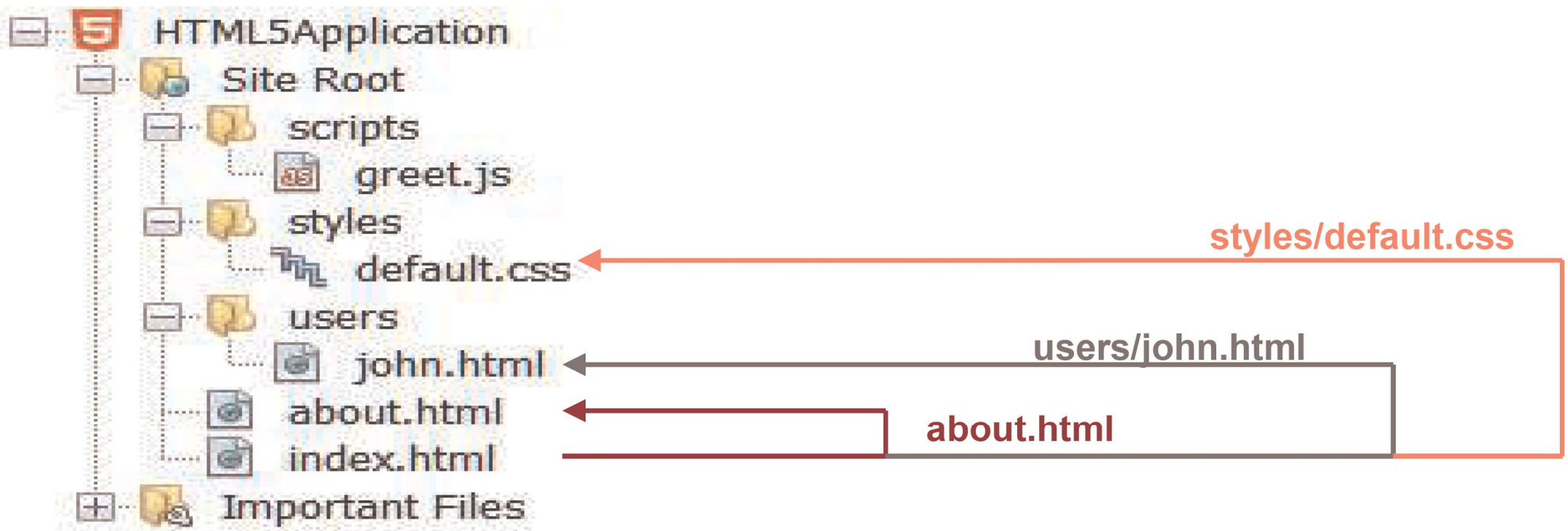
Use absolute paths to:

- Reference static resources in the current server
- Reference static resources in a different server

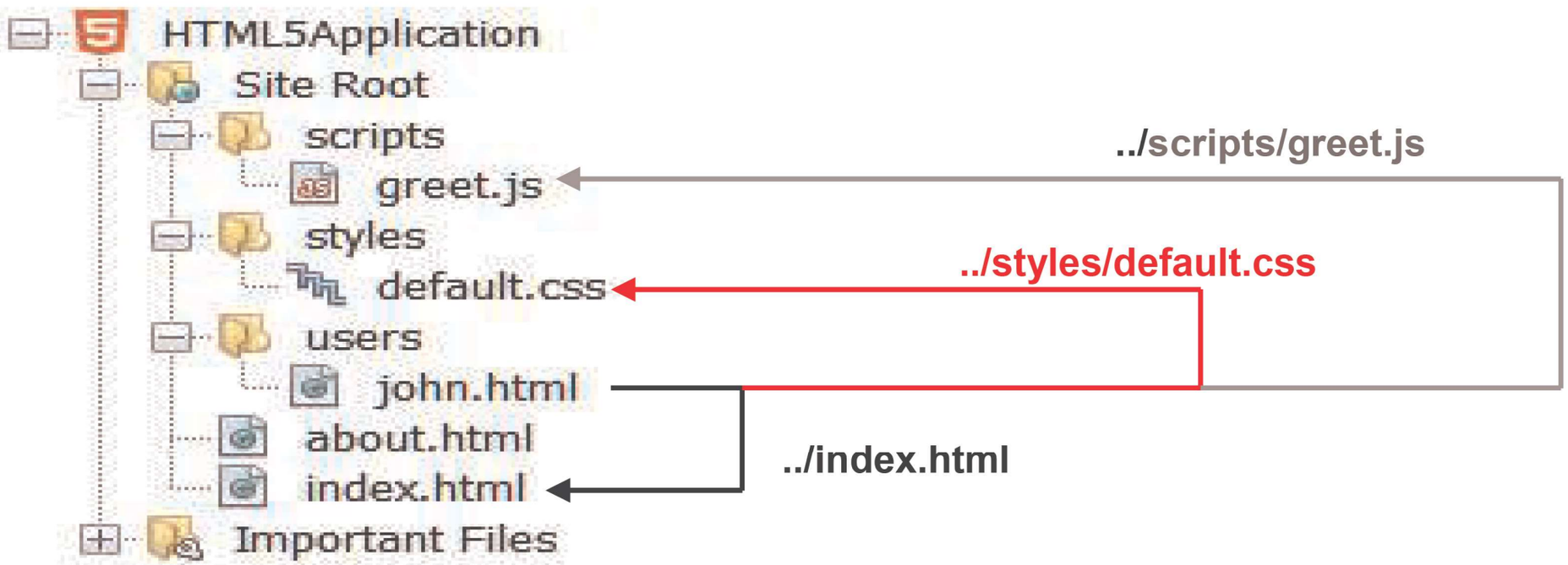
Do not use absolute paths for resources in the same application.

Relative Paths

Relative paths are used to reference resources inside the same application.



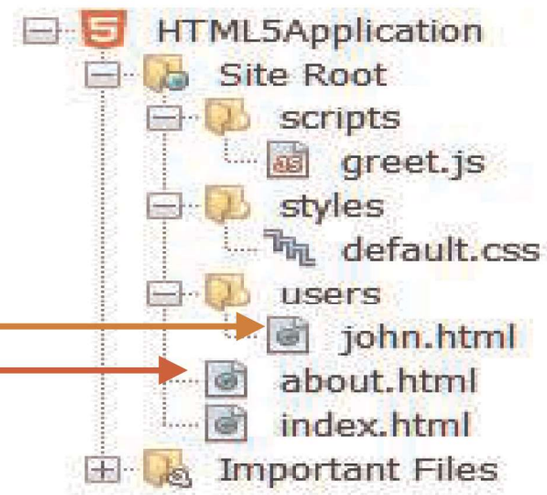
Relative Paths



Document Navigation

The anchor tab `<a href="destination">link name</a>` is used to provide links to the user to navigate to a different document.

```
<body>
   <h2>Menu</h2>
   <a href="about.html">About this Application</a><br>
   <a href="users/john.html">John's page</a><br>
 </body>
```

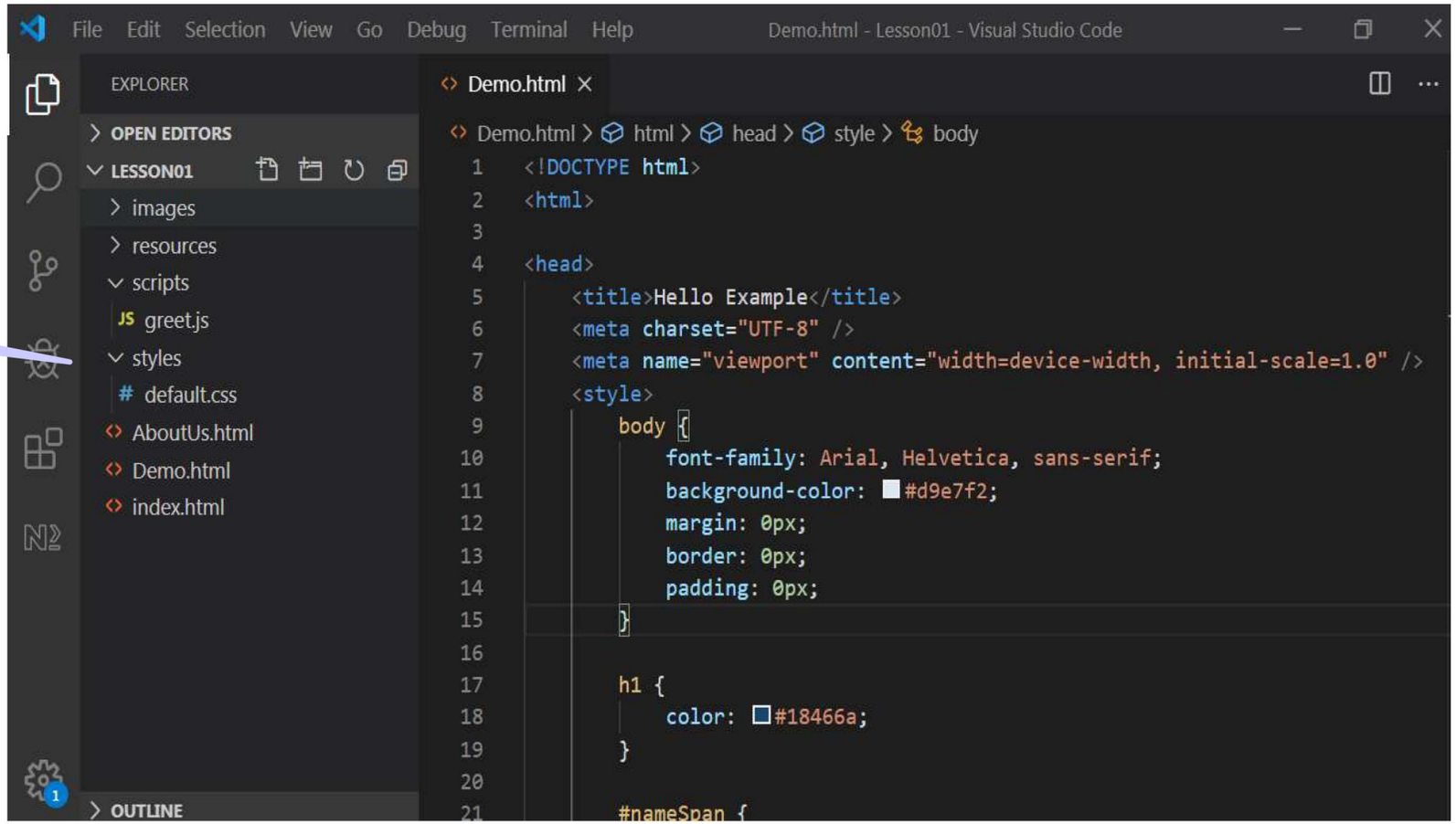


Development Tools

- Web browser
- Web browser development tools:
  - Page Inspector
  - Style inspector
  - Network Monitor
  - JavaScript Debugger
  - JavaScript Console

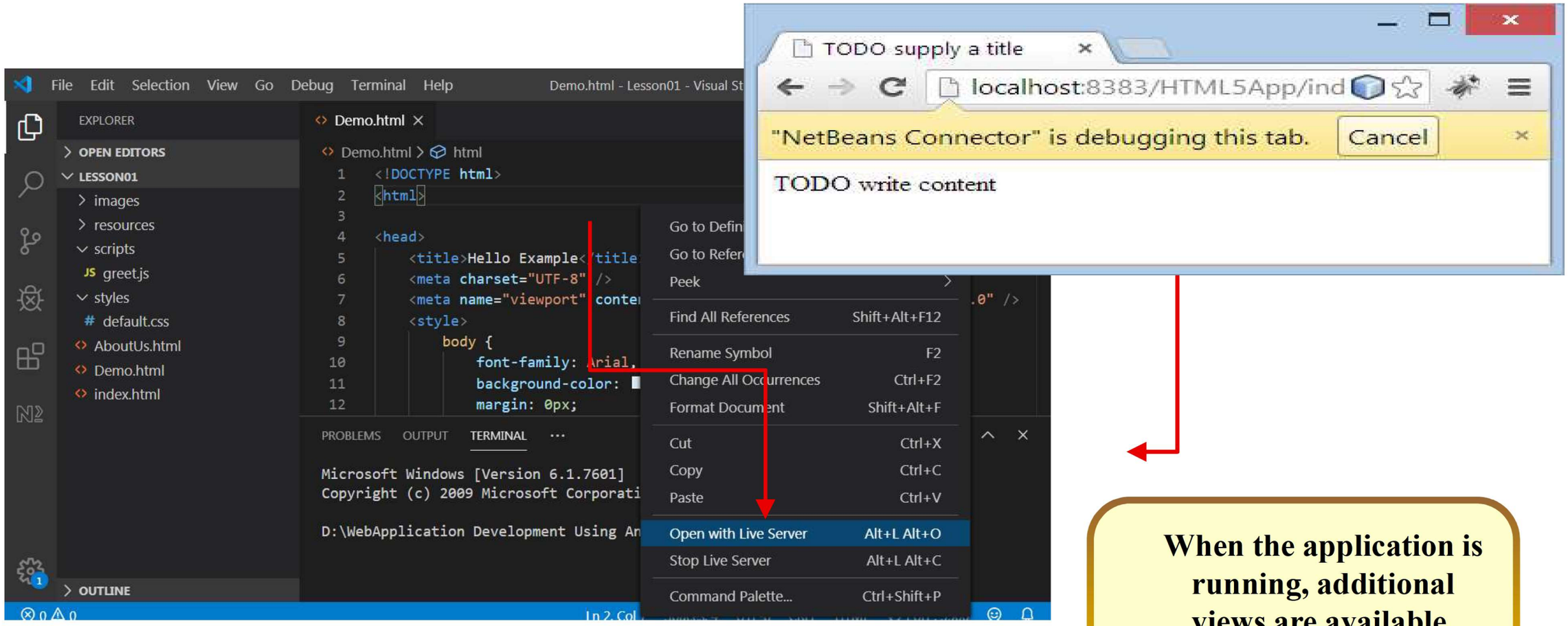
- Integrated Development Environment:
  - Organizes all your application files and settings into projects
  - Automates processes such as running, debugging, testing, and profiling applications
- An IDE also provides many useful features, such as:
  - Multiple language support
  - JavaScript syntax checking and suggestions
  - HTML5 structure check, tag completion, and attribute suggestions
  - CSS syntax check, rules match-up, and inheritance analysis

Application Files



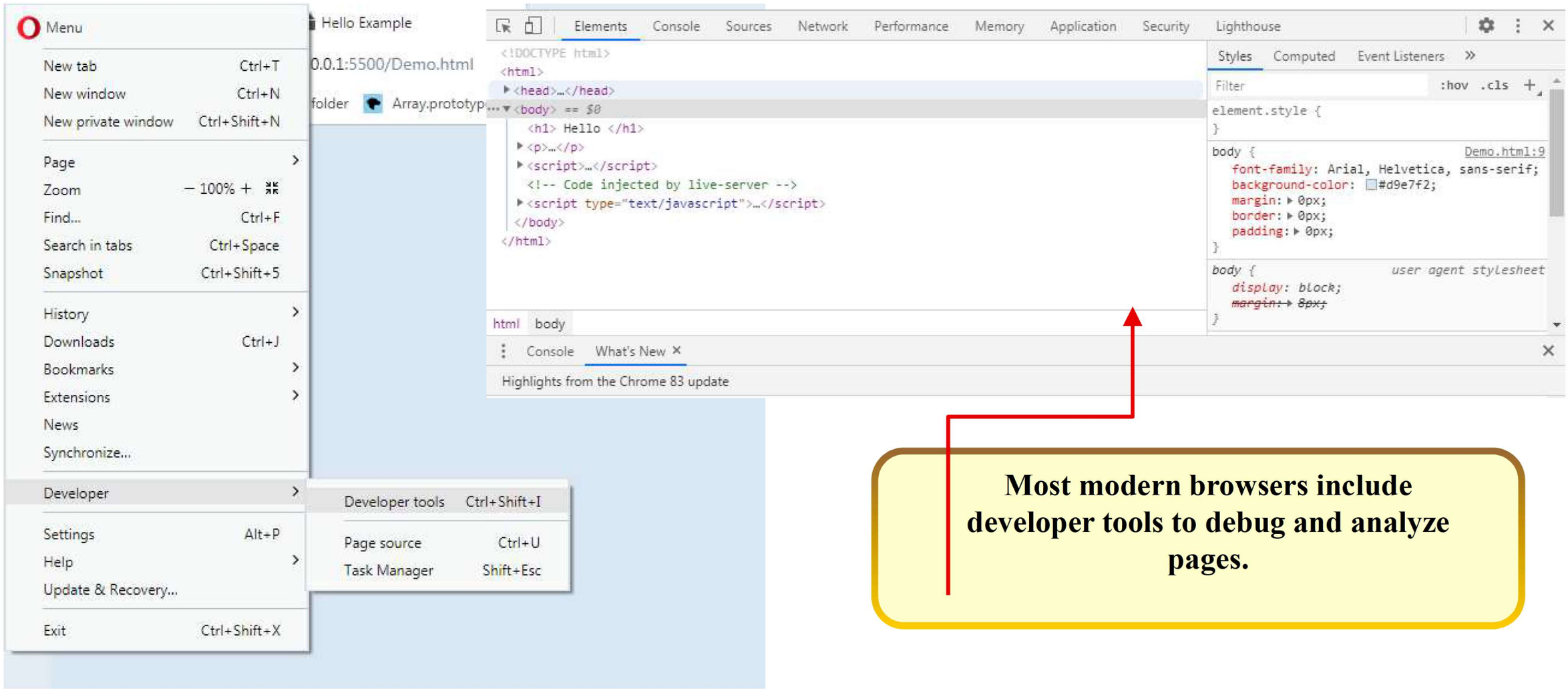
Visual Studio Elements

Running Applications



When the application is running, additional views are available.

Browser Development Tools



Most modern browsers include developer tools to debug and analyze pages.

Where Can I Learn More?

Resource	Website
HTTP Response codes	http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
HTML 5 Support in NetBeans IDE Wiki	https://netbeans.org/kb/trails/php.html
NetBeans Debugging and Testing JavaScript in HTML5 Applications	https://netbeans.org/kb/docs/webclient/html5-js-support.html
Chrome developer tools overview	https://developers.google.com/chrome-developer-tools/
Firefox developer tools	https://developer.mozilla.org/en/docs/Tools
Safari developer Tools	https://developer.apple.com/safari/tools/
Internet Explorer Dev Center	http://msdn.microsoft.com/en-US/ie/

Summary

In this lesson, you should have learned how to:

- Create and run web applications by using NetBeans
- Separate JavaScript and CSS in different files
- Link other documents and resources
- Test and debug web applications with the browser's tools