Chapter One

Introduction to Internet and the World Wide Web

The Internet

- Network of networks that use the Internet protocol suite to link billions of devices worldwide
  - *The Internet is a global system of interconnected computer networks*
- Consists of millions of private, public, academic, business, government networks
- Networks linked together by electronic, wireless, & optical networking technologies
- Carries information resources and services, e.g. WWW

Brief history of Internet

- Began as a US Department of Defense network called ARPANET (1960s-70s)
  - APRPANET Main idea was to
    - support packet switching – allowing multiple users to use same communication path and
    - Use TCP (transmission control protocol) to ensure message is properly pass from sender to receiver
- Initial services: electronic mail, file transfer
- Opened to commercial interests and most universities in late 80s
- WWW created in 1989-91 by Tim Berners-Lee
- Early web browsers released: Mosaic 1992, Netscape 1994, Internet Explorer 1995
- Amazon.com opens in 1995; Google January 1996

The Web

- In 1989,Tim Berners-Lee of CERN (the European Organization for Nuclear Research) began to develop a technology for sharing information via hyperlinked text documents.
- He also wrote *communication protocols (Hypertext Transfer Protocol(HTTP))* to form the backbone of his new information system, which he called the World Wide Web.
- The **Web (World Wide Web)** consists of information organized into Web pages containing text and graphic images.
- It contains hypertext links, or highlighted keywords and images that lead to related information.
- A collection of linked Web pages that has a common theme or focus is called a **Web site**.

Web fundamentals

- The WWW comprises Web Servers and Web Browsers
  - Web Server: software that listens for Web page requests and serves up the requested pages
    - Apache - http://www.apache.org
    - Microsoft Internet Information Server (IIS) - http://www.iis.net/
    - Express - https://expressjs.com
  - Web browser: gets and renders documents from servers

Web fundamentals

- Hyperlinks: used to reference other web pages, e-mail addresses, files and more…
- URIs (Uniform Resource Identifiers)identify resources on the Internet
- URIs that start with http://are called URLs (Uniform Resource Locators)
- HTTP –hypertext transport protocol
- application-level protocol for distributed, collaborative, hypermedia information systems
- request/response protocol

HTTP

- Defines a set of commands understood by a Web server and sent from a browser
- Some HTTP commands (your browser sends these internally)
  - GET resource -- requests data from a specified resource
  - POST resource -- submits data to be processed to a specified resource
  - PUT resource -- uploads a representation of the specified URL
  - DELETE resource -- deletes the specified resource
- Readning: HTTP status codes

Internet Media Types (MIME)

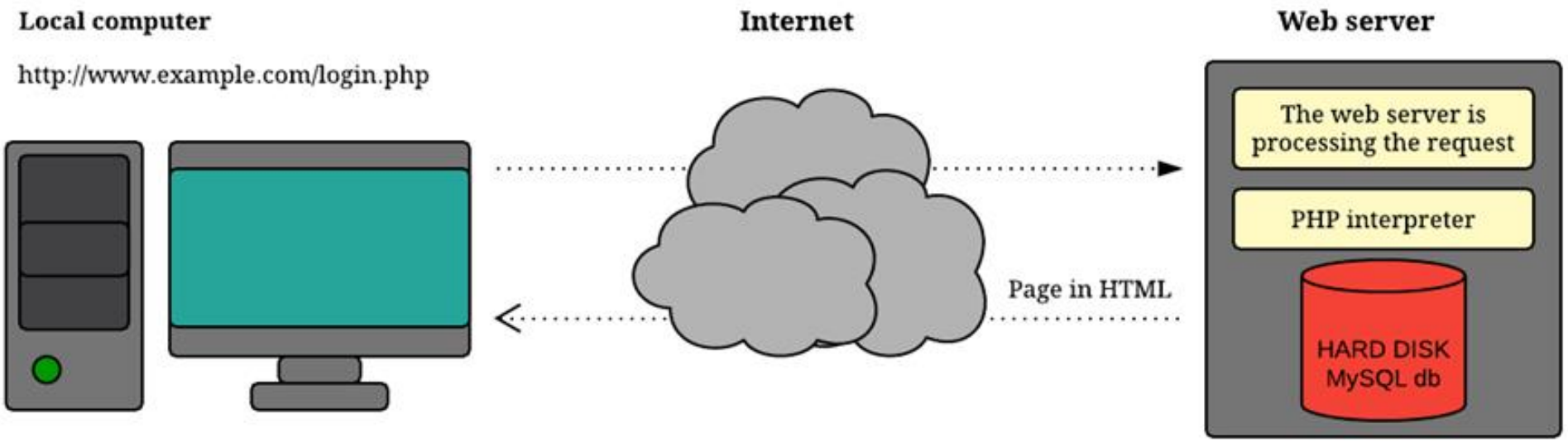
- Sometimes when including other resources in a Web page (stylesheet, image, multimedia object), we specify their type of data

MIME Type	File Extension
text/html	.html
text/plain	.txt
image/gif	.gif
image/jpeg	.jpg
videeo/quicktime	.mov
application/octec-stream	.exe

Web Client and server

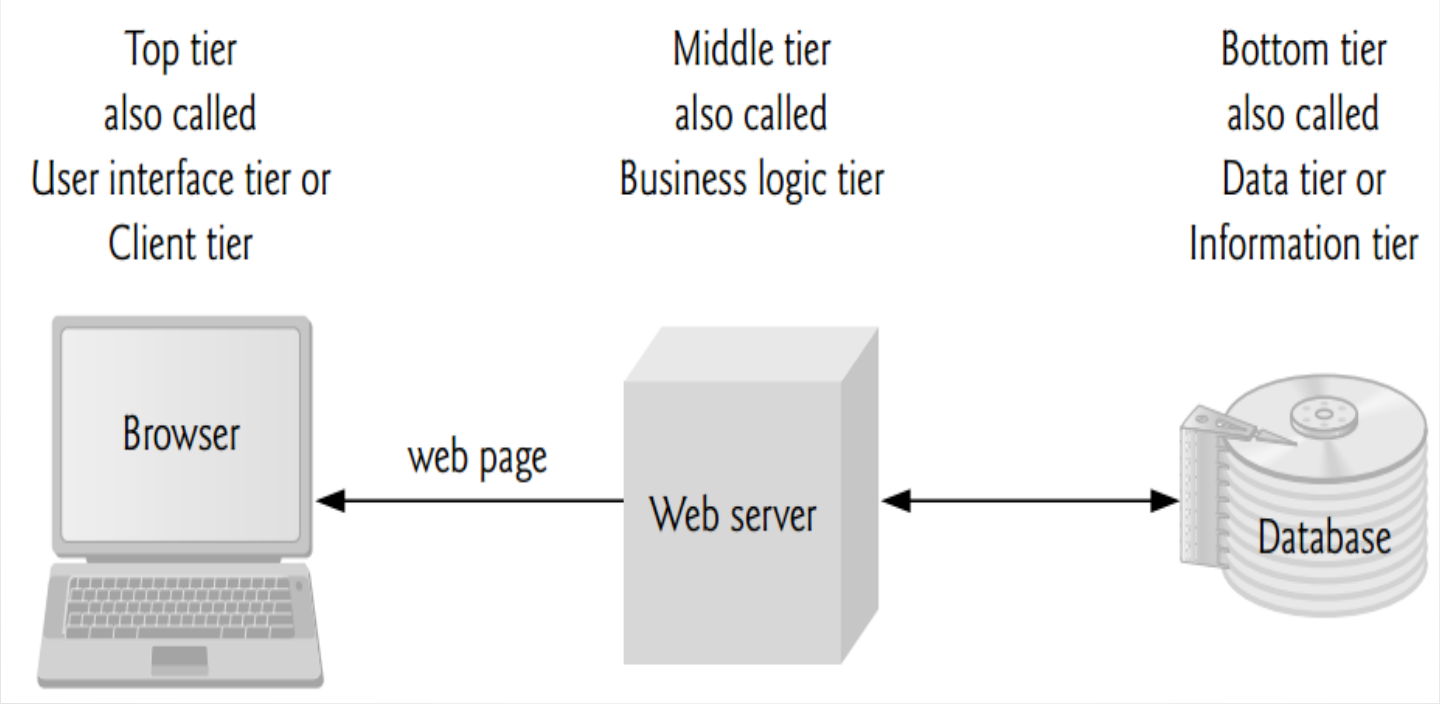
- **Client = User agent**. It is controlled by a user to operate the Web application.
- A **Web browser** is the software that you run on your computer to make it work as a web client.

Web server: A Web server is a software that supports various Web protocols like HTTP, and HTTPS, etc., to process client requests.



Multi tier application

- Web-based applications are often multi-tier applications(sometimes referred to as n-tier applications) that divide functionality into separate tiers(i.e., logical groupings of functionality)

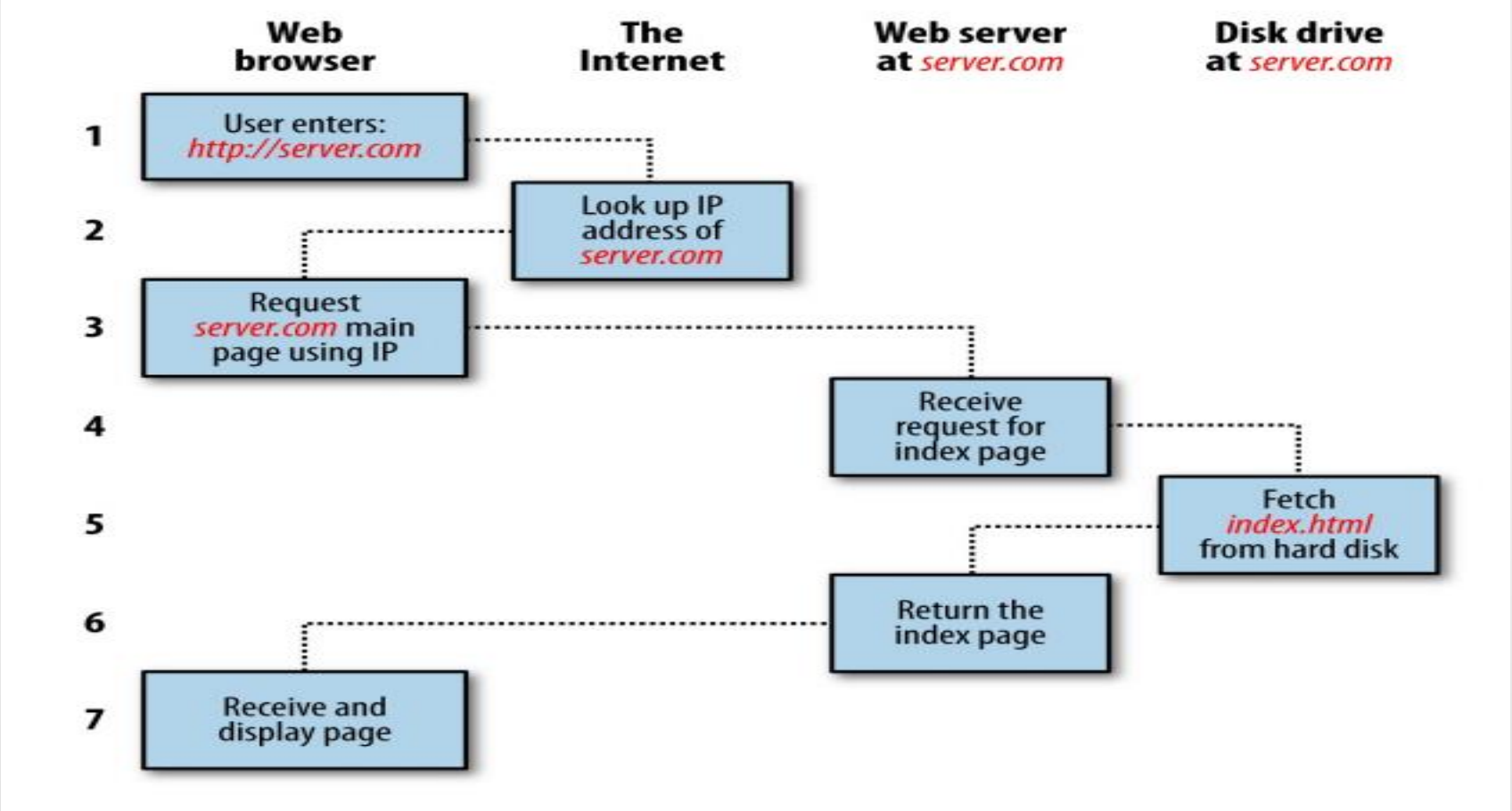


Multi tier…

- The **bottom** tier(also called the data tier or the information tier) maintains the application's data. This tier typically stores data in a relational database management system (RDBMS).
- The **middle** tier implements business logic, controller logic and presentation logic to control interactions between the application's clients and its data.
- The middle-tier controller logic processes client requests (such as requests to view a product catalogue) and retrieves data from the database.
- The middle-tier presentation logic then processes data from the information tier and presents the content to the client.
- Web applications typically present data to clients as HTML documents.
- Business logic in the middle tier enforces business rules and ensures that data is reliable before the application updates a database or presents data to users.
- Business rules dictate how clients access data and how applications process data.
- The **top** tier, or client tier, is the application's user interface, which gathers input and displays output.

Request response procedure

- The basic client/server request/response sequence

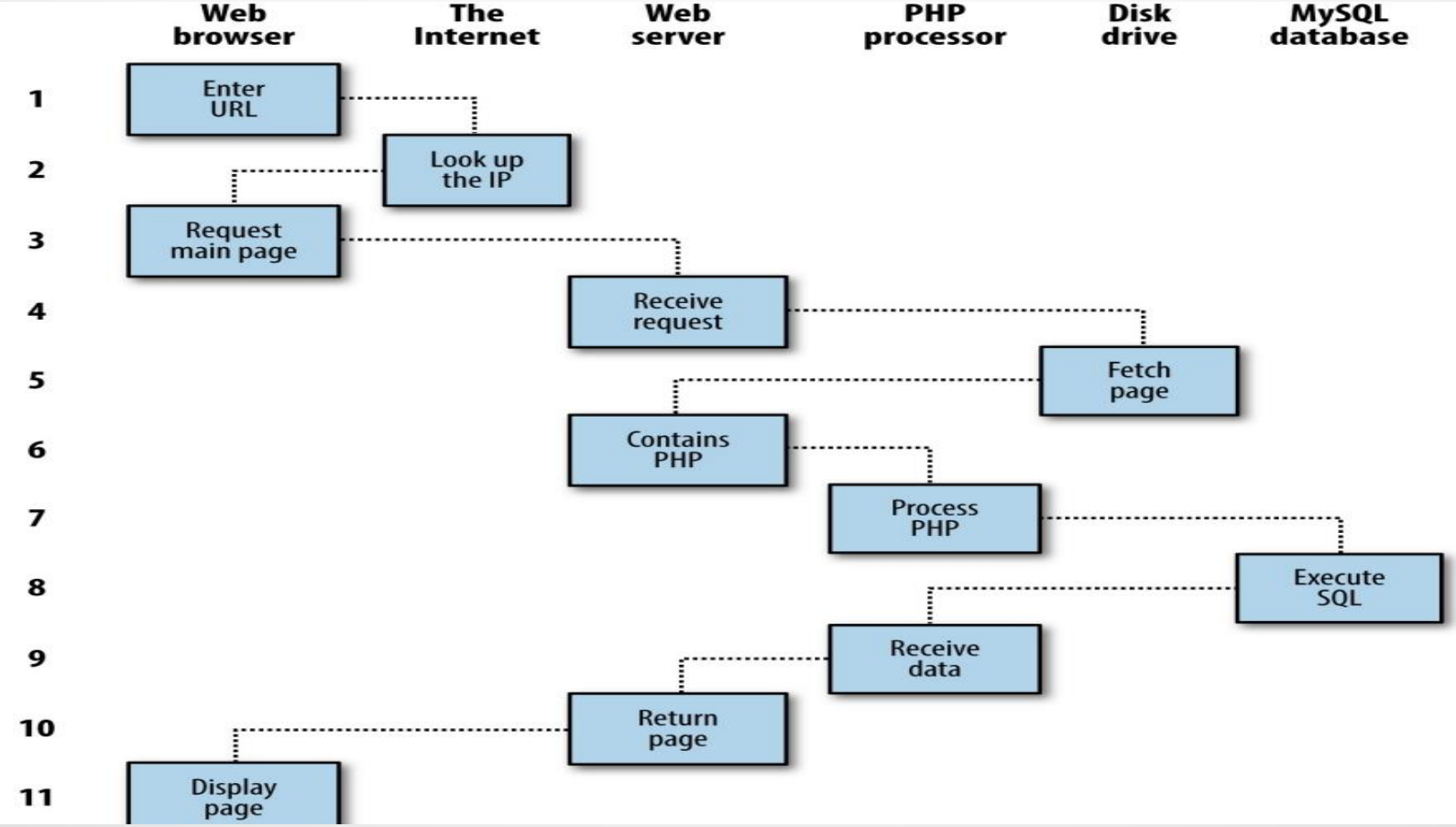


Request response…

- Each step in the request and response sequence is as follows:
-
- 1. You enter http://server.cominto your browser's address bar.
- 2. Your browser looks up the IP address for server.com.
- 3. Your browser issues a request for the home page at server.com.
- 4. The request crosses the Internet and arrives at the server.com web server.
- 5. The web server, having received the request, looks for the web page on its disk.
- 6. The web page is retrieved by the server and returned to the browser.
- 7. Your browser displays the web page.
-
- For an average web page, this process takes place once for each object within the page: a graphic, an embedded video or Flash file, and even a CSS template.

Request response…

- A dynamic client/server request/response sequence



Request response…

- You enter http://server.com into your browser's address bar.
- Your browser looks up the IP address for server.com.
- Your browser issues a request to that address for the web server's home page.
- The request crosses the Internet and arrives at the server.com web server.
- The web server, having received the request, fetches the home page from its hard disk.
- With the home page now in memory, the web server notices that it is a file incorporating
- PHP scripting and passes the page to the PHP interpreter.
- The PHP interpreter executes the PHP code.
- Some of the PHP contains SQL statements, which the PHP interpreter now passes to the MySQL database engine.
- The MySQL database returns the results of the statements to the PHP interpreter.
- The PHP interpreter returns the results of the executed PHP code, along with the results from the MySQL database, to the web server.

The HTTP Request Circle

- A typical HTTP request / response circle: [*www.w3school.com*]
- The browser requests an HTML page. The server returns an HTML file.
- The browser requests a style sheet. The server returns a CSS file.
- The browser requests an JPG image. The server returns a JPG file.
- The browser requests JavaScript code. The server returns a JS file
- The browser requests data. The server returns data (in XML or JSON).
-

Standard Web Technologies

- World Wide Web Consortium (W3C) (https://www.w3.org/)
- international consortium that develops and publishes open standards
- aims to ensure long-term growth of web
- recommends a list of standard web technologies:
- HTTP
- HTML5
- CSS3
- JavaScript
- DOM
- PHP
- MySQL
- HTML

HTML

- The public files on the web servers are ordinary text files, much like the files used by word-processing software.
- To allow Web browser software to read them, the text must be formatted according to a generally accepted standard.
- The standard used on the web is Hypertext markup language (HTML).
- HTML uses codes, or tags, to tell the Web browser software how to display the text contained in the document.
-
-

HTML…

- HTML stands for HyperText Markup Language:
- A *markup* language is a computer language that defines the structure and presentation of raw text.
- In HTML, the computer can interpret *raw text* that is wrapped in HTML elements.
- *HyperText* is text displayed on a computer or device that provides access to other text through links, also known as *hyperlinks*.

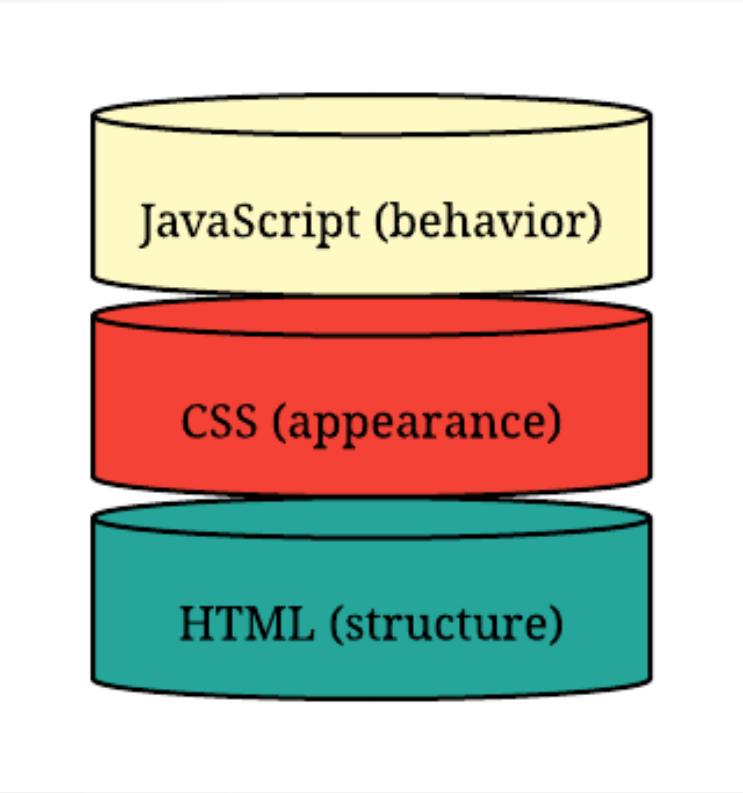
CSS

- Cascading Style Sheets
- Simple way to add style to your html and web documents
- add colour, whitespace, font, styling
- makes it possible
- to separate the content from the presentation
- share presentation rules across multiple pages
- make accessibility easier to achieve
- make responsive design easier

JavaScript

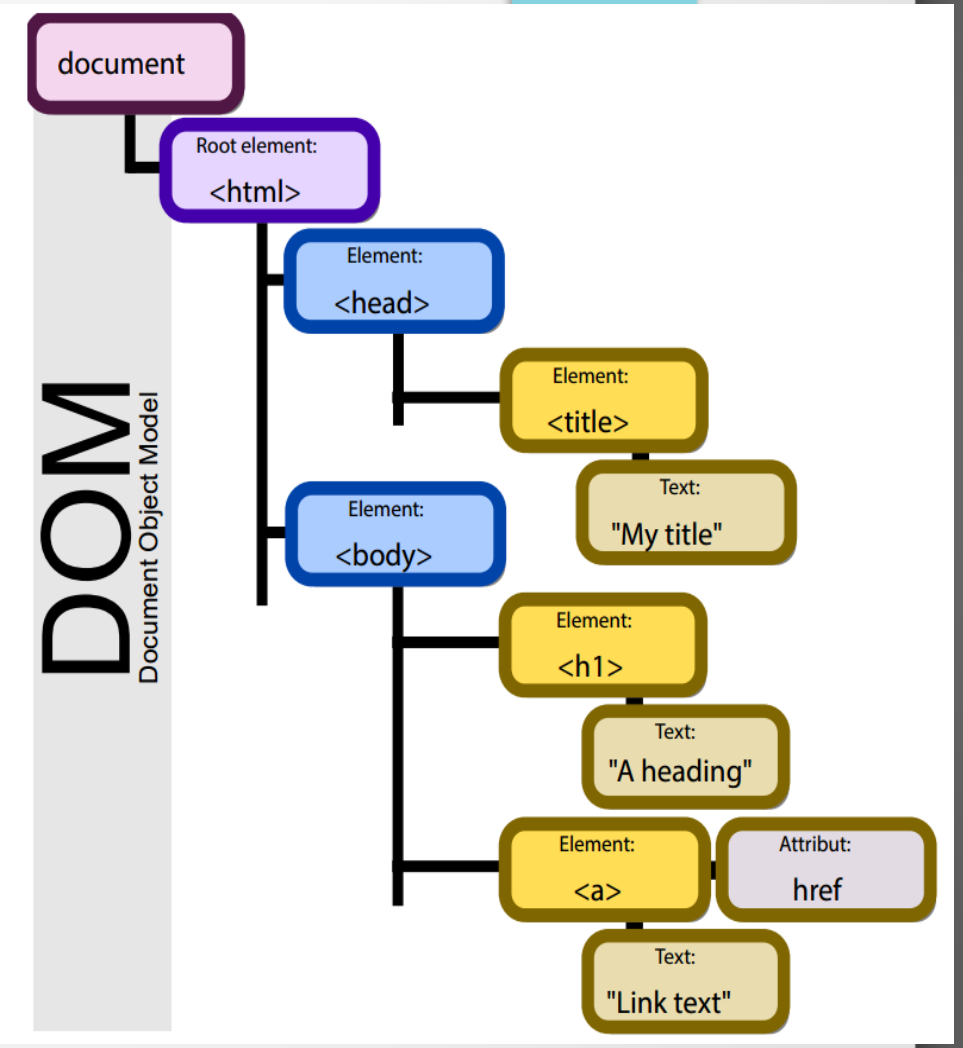
- Client-side scripting language (mainly)
- shares some syntax with Java, but is a distinct language
- Imperative, interpreted, dynamically-typed, object oriented
- JavaScript can do
- JavaScript Can Change HTML Content
- JavaScript Can Change HTML Attribute Values
- JavaScript Can Change HTML Styles CSS)
- JavaScript Can Hide HTML Elements
- JavaScript Can Show HTML Elements
-
-

How the client side works



DOM

- platform-neutral and language-neutral interface
- lets programs/scripts dynamically access:
- content
- structure
- Style
- Treats XML, HTML, XTML document like a tree



Server side scripting: PHP

- PHP: Hypertext Preprocessor
- used mainly for web development
- can be embedded in HTML
- typically interpreted
- used to dynamically generate content
- unlike JavaScript, is server-side

```
<body>
    <?php echo '<p>Hello World!</p>'; ?>
</body>
```

Web Server

- Web servers are computer systems that deliver content or services to end users over the internet.
- A Web server is also known as an Internet server.
- It consists of a physical server, server operating system (OS) and software used to facilitate HTTP communication.
- Example: Apache, IIS, Tomcat

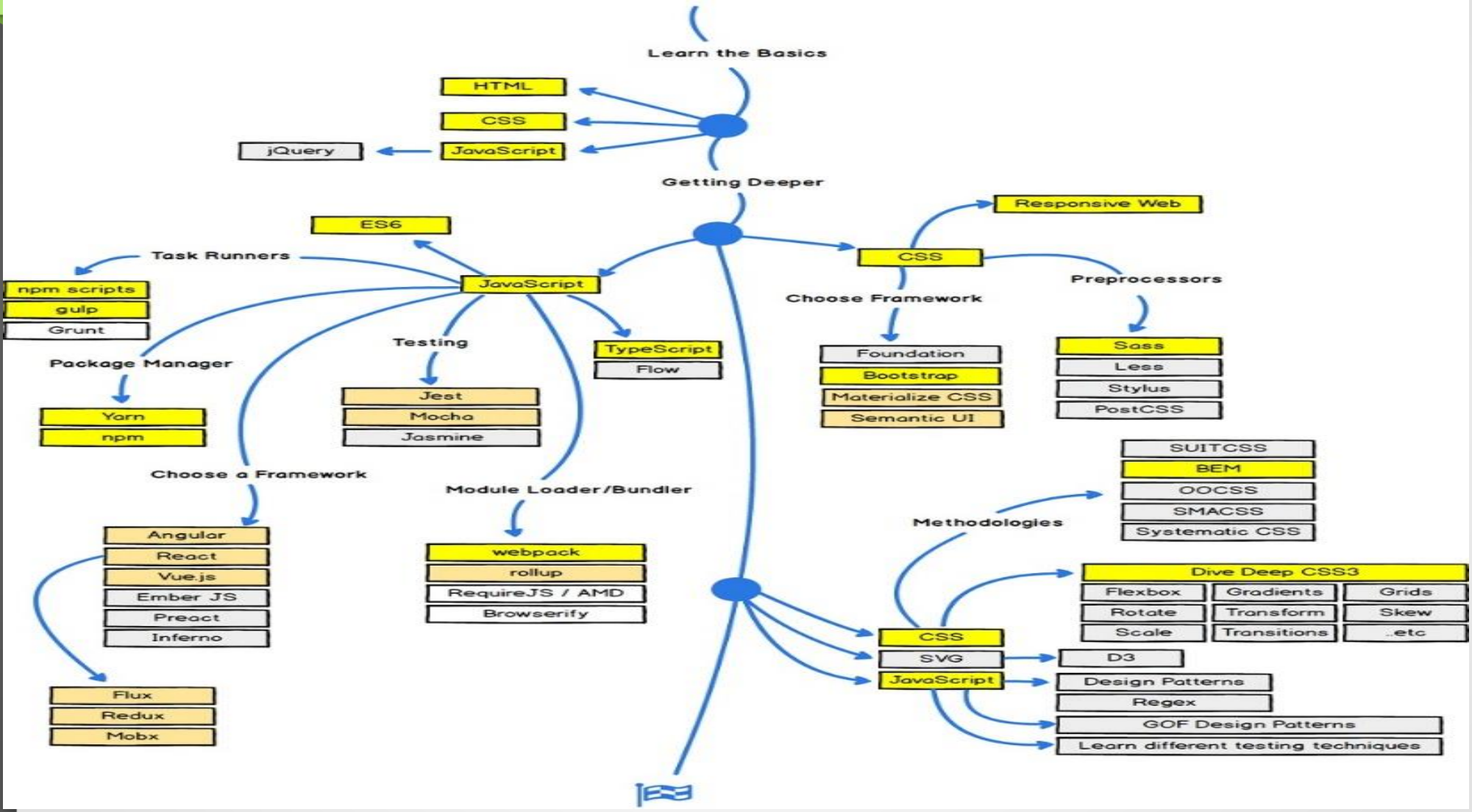
Web Development Principles

- It is important to create websites that are:
- attractive
- functional
- user-friendly
- Start with
- what is the website?
- what is the audience?
- Design navigation based on audience and purpose
- Provide multiple links from different contexts to vital partsof site

Web Development …

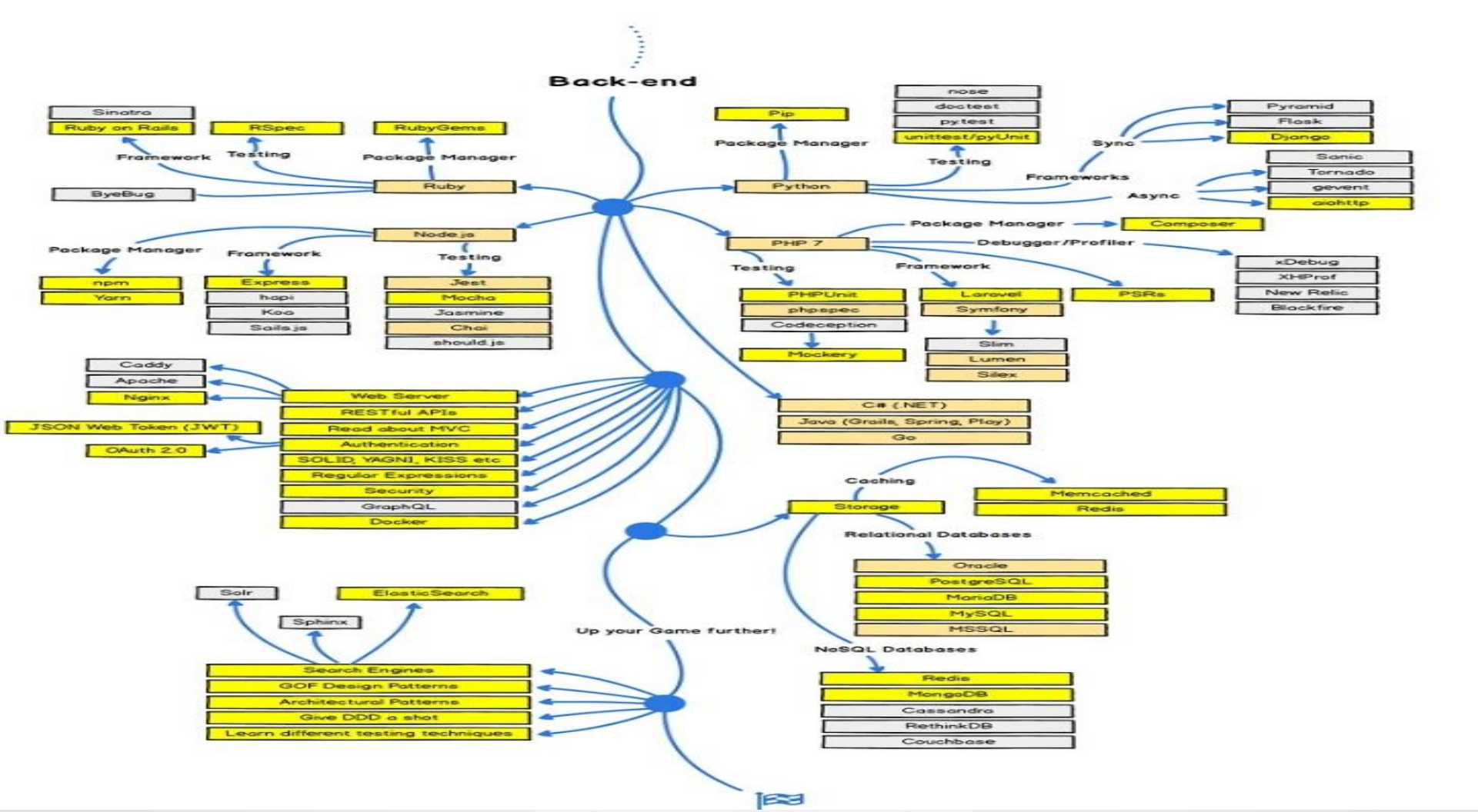
- Keep it Simple Stupid (KISS)
- avoid lots of video and audio
- choose familiar and appropriate
- fonts
- colours
- use contrast and whitespace
- ensure fast load
- use a fluid layout for different screen sizes
- use open standards
- test your site on many browsers

Front- end road map



Source: https://www.w3schools.com/whatis/default.asp

Back- end road map



Reading assignment

- IP addressing
- DNS

Chapter Two

Introduction HTML

HTML

- *HTML – HyperText Markup Language*
  - is the standard markup language used to create web pages.
- HTML describes the structure of a website semantically along with cues for presentation
  - its a markup language, rather than a programming language
- *HTML documents consist of a tree of elements and text.*
- *Each element is denoted in the source by a start tag,*
  - *such as "<body>", and an end tag, such as "</body>".*
- Web browsers can read HTML files and render them into visible or audible web pages.

Anatomy of HTML document

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
    <p>This is a web page.</p>
  </body>
</html>
```

- The <!DOCTYPE html> declaration defines this document to be HTML
- The <html> element is the root element of an HTML page
- The lang attribute defines the language of the document
- The <meta> element contains meta information about the document
- The charset attribure defines the character set used in the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

HTML Anatomy ...

- **Elements**
  - Elements define the structure and content of objects within a page.
  - Enclosed inside angel brackets < and > : E.g. <a> , , , , , <h1>…
  - meta information is placed in the head element
  - content information is placed in the body element
- **Tags**
  - The use of less-than and greater-than angle brackets surrounding an element creates what is known as a *tag*.
  - Tags most commonly occur in pairs of opening and closing tags.
  - Content of the elements always falls between the opening (e.g.)and closing (e.g.) tag: E.g. <a> …content here…
  - Tags have to be nested such that elements are all completely within each other, without overlapping:
- *Certain start tags and end tags can in certain cases be omitted and are implied by other tags.*

HTML Anatomy ...

- Attributes
  - Attributes are properties used to provide additional information about an element.
  - The most common attributes include *id, class, src, herf*
  - Attributes are defined within the opening tag, after an element's name.
  - Generally attributes include a name and a value.
  - The format for these attributes consists of the attribute name followed by an equals sign and then a quoted attribute value.
- E.g. <a *href="demo.html"*>simple

Classifying HTML elements

- **top-level elements:** html, head and body
- head elements: placed inside the head
  - Title, style, link, meta, base, script
  - do not display on the page
- **block-level elements:** flow elements that behave like paragraphs
  - occupy 100% of available width
  - stacked vertically with preceding and subsequent block elements
  - article, h1-h6, header, footer, section, p, figure, canvas,
  - pre, div, ul, ol, dl, table, form, video

Classifying HTML...

- **inline elements**
  - phrasing elements that behave like words
  - flow horizontally
  - usually placed inside block elements
  - a (anchor), audio, br, code, img, em, nav, samp
  - span, strong, sub, sup, time, var

Syntax of HTML

- All tags begin with < and end with >
- tag name is given immediately following opening <
- unrecognized tags are ignored
- attributes are given following the tag name:
- <tag attribute1=“value” attribute2=“value” ...>
- ‘ can be used instead of “
- forgetting to close a quote can result in a blank page
- names and attributes are lowercase
- values are case sensitive
- attributes that have boolean values:
- true: attribute_name=“attribute_name”
- false: attribute_name=“”

```
<textarea readonly=""></textarea>
<textarea readonly="readonly"></textarea>
```

HTML syntax ...

- unrecognized tags and attributes are ignored
- elements without closing tags are of the form <img />
  - ending / is optional for HTML5, needed for polyglot documents
- elements must be well-formed: no bad nesting
  - Stuff
- attributes can be in any order
- white space is allowed
  - between tag name and attributes
  - around the = sign
  - within attribute value (but should be avoided)

Tag	Description
h1, . . . , h6	Header tag h1 to h6
p	paragraphs (Line changes at the end)
span	No line change after span
div	make division between contents
a	hyperlink
center	Move content to center
br	Line break (no closing tag)
hr	horizontal line (no closing tag)
pre	preserve formatting
table	insert table

HTML5

- HTML5 is a cooperation between the
  - World Wide Web Consortium (W3C) and
  - Web Hypertext Application Technology Working Group (WHATWG).

- WHATWG was working with web forms and applications,
- W3C was working with XHTML 2.0.
- In 2006, they decided to cooperate and create a new version of HTML.
-

HTML5: Ground Rules

- Some rules for HTML5 were established:
  - New features should be based on HTML, CSS, DOM, and JavaScript
  - Reduce the need for external plugins
  - Better error handling
  - More markup to replace scripting
  - HTML5 should be device independent
  - Dev process should be visible to the public

New features include:

- Reduce for external plugin
- Semantic elements: `header`, `footer`, `section`, `article`, others.
- `canvas` for drawing
  - paths of rectangles, arcs, lines, images
  - mouse events
- Better support for `Local offline Storage` (variation of cookies)
- `audio` & `video` elements
  - including drawing video on canvas
- form controls, like calendar, date, time, email, url, search, color, datetime, datetime-local, month, number, range, tel, week

New API's

- HTML Geolocation
- HTML Drag and Drop
- HTML Local Storage
- HTML Application Cache
- HTML Web Workers
- HTML SSE

Cont…

- All HTML5 elements can have the following:
  - id – uniquely identifies an element in the page
  - style – gives presentation style
  - class – style class, or space-separated list of style classes
  - title – title for the element. Can be used as tool-top display
  - hidden – prevents element from being displayed when set to true
  - contenteditable, draggable, dropzone, spellcheck, and more

- HTML5 defines eight new **semantic** elements.
- All these are **block-level** elements.
- To secure correct behavior in older browsers, you can set the CSS **display** property for these HTML elements to **block**:

```
header, section, footer, aside, nav, main,
article, figure {
  display: block;
}
```

HTML 5 migration

Typical HTML4	Typical HTML5
	<header>
	<nav>
	<section>
	<article>
	<footer>

- Reading
  - Www.w3school.com
  -

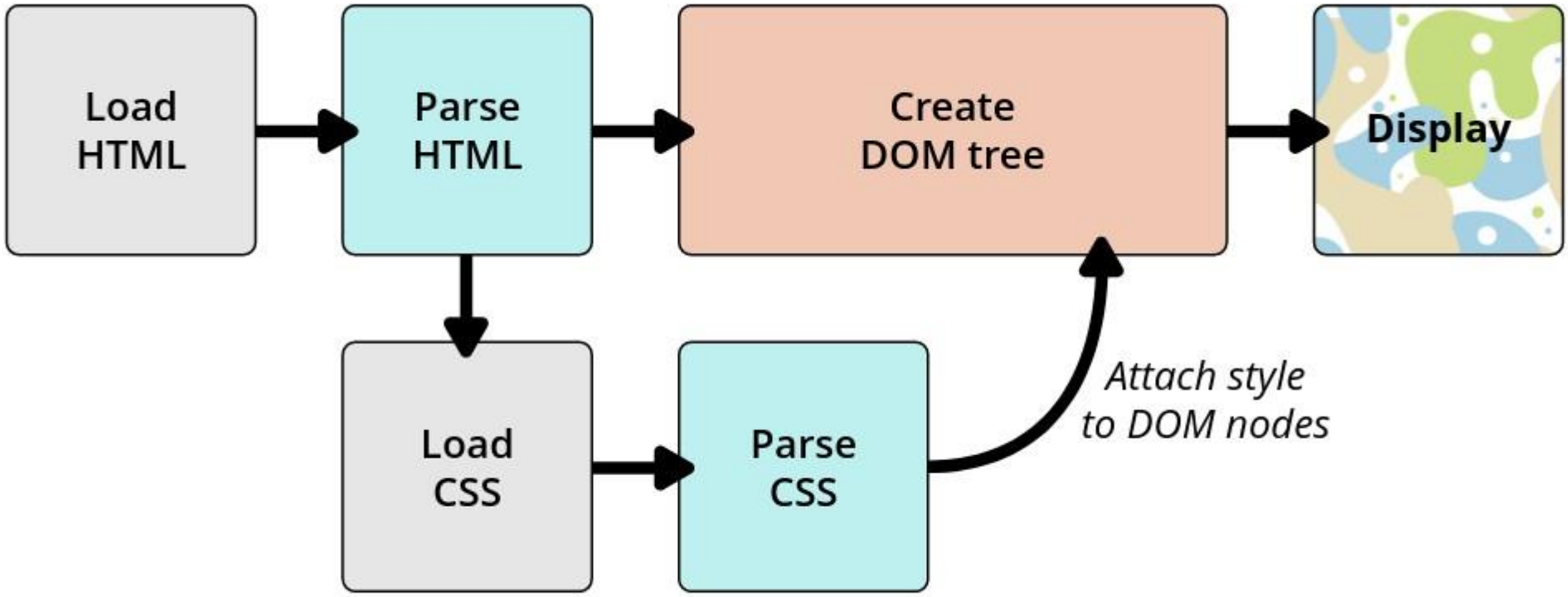
Chapter 3

Introduction to CSS

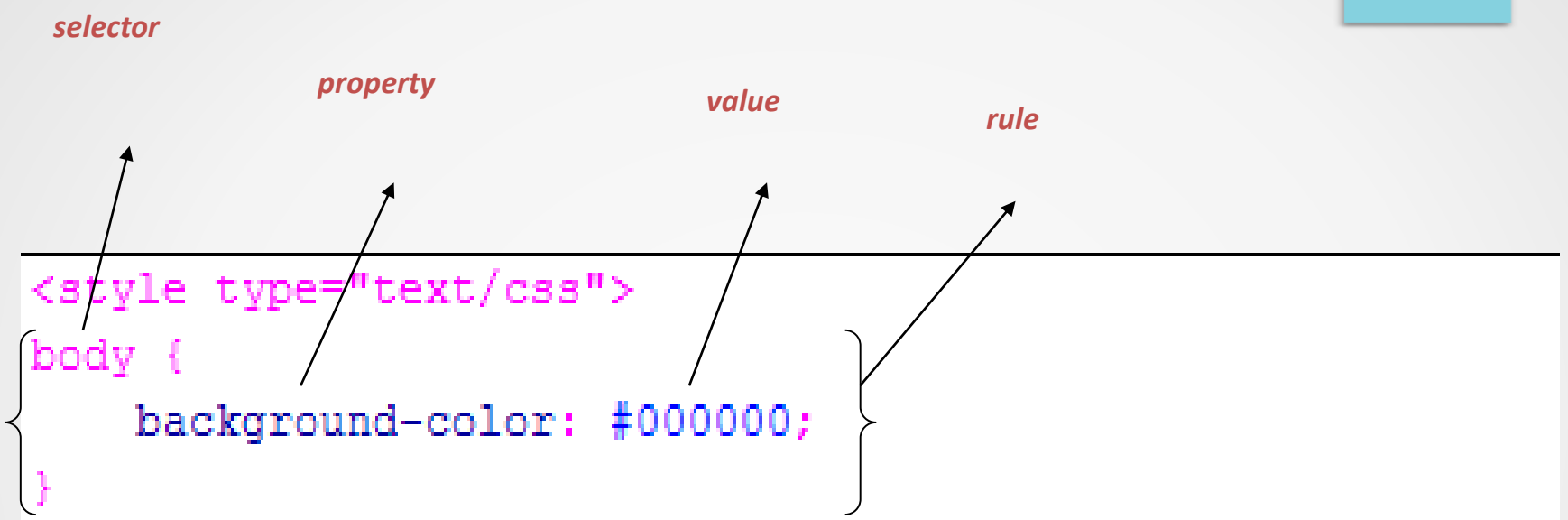
CSS

- **Cascading Style Sheets** (**CSS**): is a simple mechanism for *adding style* (e.g. fonts, colors, layouts) to Web documents.
- Styles provide *powerful control* over the **presentation** of web pages.
- A style sheet consists of a set of ***rules***.
- Each ***rule*** consists of one or more ***selectors*** and a ***declaration block***.
- A ***declaration block*** consists of a list of ***declarations*** in curly braces ({}).
- Each ***declaration*** consists of a **property**, a **colon** (**:**), a **value**, then a **semi-colon** (**;**)

How CSS works



Style Sheet Syntax Explained




```
<style type="text/css">
body {
    background-color: #000000;
}
h1 {
    font-family: Georgia, "Times New Roman", Times, serif;
    font-size: 32px;
    color: #3099D3;
    text-align: center;
}
</style>
```

Creating CSS

- Create a style sheet file and apply CSS to multiple webpages, a single page, or an individual HTML element.
- Three methods:
- External Style Sheets (preferred choice)
- Embedded Style Sheets
- Inline Styles (least desirable)

CSS scope

- Create a style sheet file and apply CSS to multiple webpages, a single page, or an individual HTML element.
- Three methods:
- External Style Sheets (preferred choice)
- Embedded Style Sheets
- Inline Styles (least desirable)
-

Example: inline

```
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
    <title>CSS Tutorial</title>

</head>
<body>

    <h3 style="color:blue"> Heading 1 </h3>
    <h3 style="color:blue"> Heading 3 </h3>
    <h3 style="color:blue"> Heading 3 </h3>

</body>
</html>
```

Example: Embedded CSS

```
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
    <title>CSS Tutorial</title>

    <style type="text/css">
        h3.h3_blue{      /*change color to blue*/
            color: blue;
        }

        h3.h3_red{       /*change color to red*/
            color:red;
        }
    </style>

</head>
<body>

    <h3 class='h3_blue'> Heading 1 </h3>
    <h3 class='h3_blue'> Heading 3 </h3>
    <h3 class='h3_blue'> Heading 3 </h3>


    <h3 class='h3_red'> Heading 1 </h3>
    <h3 class='h3_red'> Heading 3 </h3>
    <h3 class='h3_red'> Heading 3 </h3>

</body>
</html>
```

Example: External

```html
<!DOCTYPE html>
<html>
<head>
    <title>CSS Tutorial</title>
    <link rel="stylesheet" type="text/css" href="asset/css/my_css.css">
</head>
<body>

    <h3 class='h3_blue'> Heading 1 </h3>
    <h3 class='h3_blue'> Heading 3 </h3>
    <h3 class='h3_blue'> Heading 3 </h3>


    <h3 class='h3_red'> Heading 1 </h3>
    <h3 class='h3_red'> Heading 3 </h3>
    <h3 class='h3_red'> Heading 3 </h3>

</body>
</html>
```

```css
/* asset/css/my_css.css */

h3.h3_blue{
    color: blue;
}

h3.h3_red{
    color:red;
}
```

Basic CSS Selectors

- There are three types of selectors in CSS,
- **Element** : can be selected using it's name
  - e.g. 'p', 'div' and 'h1' etc.
- **Class** : can be selected using '.className' operator
- e.g. '.h3_blue'.
- **ID** : can be selected using '#idName'
  - e.g. '#my_para'.

Example: selector

```html
<!-- css.html -->

<!DOCTYPE html>
<html>
<head>
    <title>CSS Selectors</title>
    <link rel="stylesheet" type="text/css" href="asset/css/my_css.css">
</head>
<body>
    <h3>CSS Selectors</h3>

    <p class='c_head'> Paragraph with class 'c_head' </p>
    <p id='i_head'> Paragraph with id 'i_head' </p>

</body>
</html>
```

```css
/* asset/css/my_css.css */

/*element selection*/
h3 {
    color: blue;
}


/*class selection*/
.c_head{
    font-family: cursive;
    color: orange;
}


/*id selection*/
#i_head{
    font-variant: small-caps;
    color: red;
}
```

More selectors

Selectors	Description
h1, p, span etc.	element selector
.className	class selector
#idName	id selector
*	Universal selector (selects everything)
h1.className	select h1 with class 'className'
h1#idName	select h1 with id 'idName'
p span	descendant selector (select span which is inside p)
p > span	child selector ('span' which is direct descendant of 'p')
h1, h2, p	group selection (select h1, h2 and p)
span[my_id]	select 'span' with attribute 'my_id'
span[my_id=m_span]	select 'span' with attribute 'my_id=m_span'

Precedence

- Precedence
  - Local > Internal > External
- Selector Priority level :
  - ID (highest priority)
  - Class
  - Element

Chapter – 4

Server Side Programming using PHP

Part one

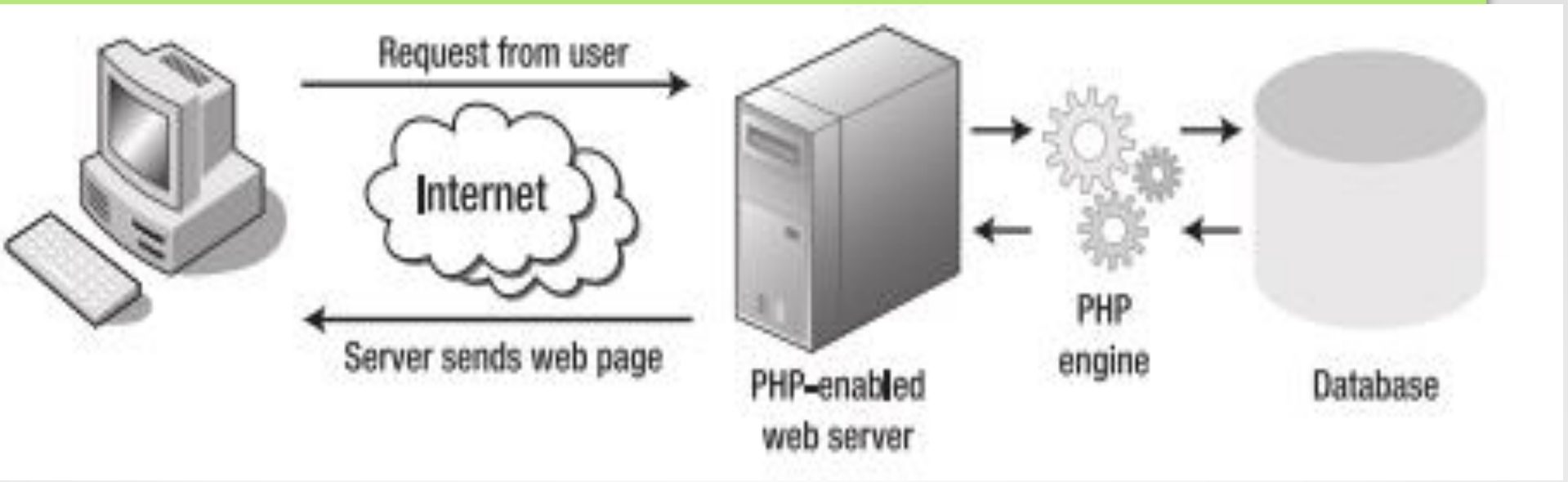
3.1 Server side scripting

- **It is** a web technology that allows custom HTML to be delivered to a client machine where the code that generates processed on the web server
- Server-side language code remains on the web server. After it has been processed, the server sends only the output of the script to the user as HTML format.
- Some of the examples of server side scripting languages are
    - CGL,
    - ASP,
    - PHP, …

Cont . . .

- Server-side language brings websites to life in the following ways:
  - ✓Uploading files through a web page
  - ✓Reading and writing to files
  - ✓Displaying and updating information dynamically
  - ✓Sending feedback from your website directly to your mailbox
  - ✓Using a database to display and store information
  - ✓Making websites searchable. Etc…

Fig. 2.1. How Server side scripting work



- When a PHP–driven website is visited, it sets in train the following sequence of events:
  1. The browser sends a request to the web server.
  2. The web server hands the request to the PHP engine, which is in the server.
  3. The PHP engine processes the code. It might also query a database
  4. The server sends the completed page back to the browser.

3.2 What is PHP and its Basics

- R. Lerdorf unleashed the first version of PHP way back in 1994.
- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML and which allow web developers to write dynamically generated pages quickly..
- It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side.
- The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

Cont…

- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP.
- PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- Create a customized user experience for visitors based on information that you have gathered from them.

What is a PHP File?

- ? PHP files may contain text, HTML tags and scripts
- ? PHP files are returned to the browser as plain HTML
- ? PHP files have a file extension of ".php", ".php3", or ".phtml"

Cont…

- PHP files should be in the web servers document root or a subfolder of the document root.
- Default location of the document root for the setups is as follows:
  - **XAMPP**: C:\xampp\htdocs
  - **WampServer**: C:\wamp\www
  - **EasyPHP**: C:\EasyPHP\www
  - **IIS**: C:\inetpub\wwwroo
  - **MAMP**: Macintosh HD:Applications:MAMP:htdocs

Why PHP?

- Runs on different platforms (Windows, Linux, Unix, etc.)
- Compatible with almost all servers used today(Apache, IIS, etc.)
- FREE to download from the official PHP resource: www.php.net
- Easy to learn and runs efficiently on the server side

Cont…

Common uses of PHP

- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, thru email you can send data, return data to the user.
- Enable you add, delete, modify elements within your database
- Access cookies variables and set cookies.
- You to restrict users to access only some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible:

- Simplicity,
- Flexibility,
- Efficiency,
- Familiarity
- Security,

Cont…

PHP Environment Setup:

- In order to develop and run PHP Web pages three vital components need to be installed on your computer system.
  - ***Web Server*** **-** PHP will work with virtually all Web Server software, including IIS but then most often used is freely available Apache Server.
  - ***Database*** **-** PHP will work with virtually all database software, including Oracle and Sybase but most commonly used is freely available MySQL database.
  - ***PHP Parser*** **-** In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser.

Cont…

Download those tools for free from the following address

- ***Apache:-*** ***http://httpd.apache.org/download.cgi***
- ***MySQL:-*** ***http://www.mysql.com/downloads/index.html***

Search for the following topics to install and configure PHP

- ***PHP Installation on Linux or Unix with Apache***
- ***PHP Installation on Mac OS X with Apache***
- ***PHP Installation on Windows NT/2000/XP with IIS***
- ***PHP Installation on Windows NT/2000/XP with Apache***

3.3 Syntax and Variables in PHP

- The PHP parsing engine needs a way to differentiate PHP code from other elements in the page. The mechanism for doing so is known as 'escaping to PHP.' There are four ways to do this:

1) Canonical PHP tags: A most universally effective PHP tag style

<?php *PHP code goes here* ?>

2) *Short-open (SGML-style) tags*:

<? *PHP code goes here* ?>

3) ASP-style tags:

<% ***PHP code goes here*** %>

4) HTML script tags:

<script language="PHP“> ***PHP code goes here*** </script>

- Mostly PHP scripting block starts with **<? *php*** and ends with **?>**

Cont…

- A PHP scripting block can be placed anywhere in the document.
- Each code line in PHP must end with a semicolon which is a separator and used to distinguish one set of instructions from another.
- If you have PHP inserted into your HTML and want the web browser to interpret it correctly, then you must save the file with a .php extension.
- PHP is case sensitive
- PHP is whitespace insensitive means that it almost never matters how many whitespace characters you have in a row
- Commenting PHP Code:
  - Single-line comments: // comments gone here
  - Multi-lines comments: /* comments gone here */

Cont…

- Outputting in PHP is using the key word echo.

```
echo” some text”;
```

- To run perfectly scripts like PHP must be embedded in HTML and stored in a web server.

Example:

```
<html>
<head>
 <title>My First PHP Page</title>
</head>
<body>
        <?php
         echo “covid-19 change our life style !";
              ?>
         </body>
</html>
```

Cont…

✓PHP – Variables

- PHP variables are like Perl, all are denoted with a leading dollar sign ($).
- The value of variable is the value of its most recent assignment.
- Variables can, but do not need, to be declared before assignment
- Do not have intrinsic types (Untyped) - a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP variables must start with a letter or underscore "_".
- PHP variables may only be comprised of alpha-numeric characters and underscores. a-z, A-Z, 0-9, or _ but you cannot use characters like + , - , % , (,) . & , etc

Cont…

- PHP doesn't require variables to be declared before being initialized
- In PHP you define a variable with the following form:

```
$variable_name = Value;
$hello = "Hello World!";
$a_number = 4;
```

- To output a string we can place either a string variable or you can use quotes,

```
$myString = "Hello!";  echo $myString;   OR
echo "<h5>I love using PHP!</h5>";
```

- Don't use quotes inside your string rather Escape your quotes that are within the string with a slash or Use single quotes

```
echo "<h5 class="specialH5">I love using PHP!</h5>";
echo "<h5 class='specialH5'>I love using PHP!</h5>";
```

Cont…

- **PHP Data Types:** has a total of 8 data types:
  - **Integers:** are whole numbers, without a decimal point, like *4195*
  - **Doubles:** are floating-point numbers, like *3.14159 or 49.1*
  - **Booleans**: have only two possible values *either true or false*.
  - **NULL**: is a special type that only has one value: *NULL*.
  - **Strings:** are sequences of characters, like *'PHP supports string operations.'*
  - **Arrays:** are named and indexed collections of other values.
  - **Objects:** are instances of programmer-defined classes, which can package up both other kinds of values and functions that are specific to the class.
  - **Resources**: are special variables that hold references to resources external to PHP (such as database connections).

3.4 PHP - Operators

- Assignment Operators (=)

```
$my_var = 4;
```

- Arithmetic Operators (+, - , / , *)

```
$my_var = 4;    $my_var2= 5;
$sum=  $my_var  +  $my_var2;
echo $sum;
```

- Comparison Operators (==, <=,>=,<,>,!=)

```
if($my_var2<= $my_var ){
}
```

- Combination Arithmetic & Assignment Operators (**.=** , /=, %=, +=,-=, etc…)

```
$my_str.="hello";
$my_str = $my_str . "hello";
```

3.5 PHP Flow Control

A) Conditional Statement

- The if, elseif ...else and switch statements are used to take decision based on the different condition.
- PHP supports following three decision making statements:
  - ***if...else statement***
    - use this statement if you want to execute a set of code when a condition is true and another if the condition is not true
  - ***elseif statement***
    - is used with the if...else statement to execute a set of code if one of several condition are true
  - ***switch statement***
    - is used if you want to select one of many blocks of code to be executed, use the Switch statement. The switch statement is used to avoid long blocks of if..elseif..else

Cont…

Example 1:

```
<html>
<body>
<?php
$d=date("D");
if ($d=="Fri")
  echo "Have a nice weekend!";
elseif ($d=="Sun")
  echo "Have a nice Sunday!";
else
  echo "Have a nice day!";
?>
</body>
</html>
```

Cont…

Example 2:

```
$destination = "Addis Ababa";
echo "Traveling to $destination <br/>";
switch ($destination){
 case "Addis Ababa":
        echo "u r 1024 KM far from Aksum";
        break;
case “Mekele":
        echo "Wow! U r 255 Km far from Aksum";
        break;
default:
        echo "which city u want to go?";
              break;
         }
```

Cont…

B) Loop statements

- Loops in PHP are used to execute the same block of code a specified number of times.
- PHP supports following four loop types.
  - ***for -*** loops through a block of code a specified number of times.
  - ***while -*** loops through a block of code if and as long as a specified condition is true.
  - ***do...while -*** loops through a block of code once, and then repeats the loop as long as a special condition is true.
  - ***foreach -*** loops through a block of code for each element in an array. **foreach**($var **as** $key=>$val).
  - **$key** refers the index of an array and **$val** indicates the value.
- PHP uses the same mechanism as other programming languages do for repetitive tasks,

Cont…

- **for – Loop Syntax**

```
for (initialization; condition; increment/decrement)
        {
        code to be executed;
        }
```

- **Example:**

```
<html> <body>
<?php
$a = 0;          $b = 0;
for( $i=0; $i<5; $i++ )
  {   $a += 10;
       $b += 5; }
echo ("At the end of the loop a=$a and b=$b" );
?>
</body>
</html>
```

Cont…

- **while – Loop Syntax**

```
while (condition){
        code to be executed;
        }
```

- **Example:**

```
<html><body>
<?php
$i = 0;      $num = 50;
while( $i < 10)
{
 $num--;    $i++;
}
echo ("Loop stopped at i = $i and num = $num"
  );
?>
</body> </html>
```

Cont…

- **do …..While – Loop Syntax**

```
do{
  code to be executed;
}while (condition);
```

- **Example:**

```
<html><body>
    <?php
    $i = 0;       $num = 0;
    do{
      $i++;
    }while( $i < 10 );
    echo ("Loop stopped at i = $i" );
    ?>
    </body></html>
```

Cont…

- **foreach– Loop Syntax**

```
foreach (array as value)
        {
code to be executed;
        }
```

- **Example 1:**

```
<html><body>
    <?php
    $array = array( 1, 2, 3, 4, 5);
    foreach( $array as $value )
    {
      echo "Value is $value <br />";
    }
    ?>
    </body></html>
```

Cont…

- **Example 2:**
- Access array elements there is a loop statement For Each loop will continue until it has gone through every item in the array.

```
<?php
$employeeAges;
$employeeAges["Lisa"] = "28";
$employeeAges["Jack"] = "16";
$employeeAges["Ryan"] = "35";
foreach( $employeeAges as $key => $value){
 echo "Name: $key, Age: $value <br />";
}
?>
```

self test questions

1. Write a Program to create following pattern using for loops:

*
**
***
****
*****
******

***2.* Write a program to count 5 to 15 using PHP loop**

3.) Write a program to print "Hello World" using echo

4.) Write a program to print "Hello PHP" using variable

5.) Write a program to print a string using echo+variable.

5.) Write a program to print two variables in single echo

6.) Write a program to check student grade based on marks

***7.)* What are the two most common ways to start and finish a PHP block of code?**

9.) Explain the difference b/w static and dynamic websites?
10.) discuss about the features of php script programming language
11.) What is the difference between "echo" and "print" in PHP?
12.) please discuss about types of error in php (we have 4 types of errors, syntax, fatal, notice and warning) and how to fix them and natures of errors)

Thank U!