Concepts of Animation and Multimedia files in JavaScript

Objectives : At the end of this lesson you shall be able to
- **know animation settings in JavaScript**
- **explain multimedia in JavaScript.**

Animation

Styling the Elements

To make an animation possible, the animated element must be animated relative to a "parent container".

The container element should be created with style = "position: relative".

The animation element should be created with style = "position: absolute".

Example

```
<!Doctype html>
<html>
<style>
#myContainer {
  width: 400px;
  height: 400px;
  position: relative;
  background: pink;
}
#myAnimation {
  width: 50px;
  height: 50px;
  position: absolute;
  background: green;
}
</style>
<body>
<h1>My First JavaScript Animation</h1>
<div id="myContainer">
<div id="myAnimation"></div>
</div>
</body>
</html>
?
```

The Animation Code

JavaScript animations are done by programming gradual changes in an element's style. The changes are called by a timer. When the timer interval is small, the animation looks continuous. The basic code is:

Example

```
var id = setInterval(frame, 5);
function frame() {
if (/* test for finished */) {
clearInterval(id);
} else {
/* code to change the element style */
}
}
```

Create the Animation Using JavaScript

Example

```
<style>`
#myContainer {
  width: 400px;
  height: 400px;
  position: relative;
  background: pink;
}
#myAnimation {
  width: 50px;
  height: 50px;
  position: absolute;
  background-color: green;
}
</style>
<body>
<p>
<button onclick="myMove()">Click Me</button>
</p>
```

```
<div id ="myContainer">
<div id ="myAnimation"></div>
</div>
<script>
function myMove() {
var elem = document.getElementById("myAnimation");
var pos = 0;
var id = setInterval(frame, 10);
function frame() {
if (pos == 350) {
clearInterval(id);
} else {
pos++;
elem.style.top = pos + 'px';
elem.style.left = pos + 'px';
}
}
}
</script>
</body>
</html>
```

Multimedia files

What is Multimedia?

Multimedia comes in many different formats. It can be almost anything you can hear or see. Web pages often contain multimedia elements of different types and formats.

Examples: Images, music, sound, videos, records, films, animations and more.

Multimedia Formats

Multimedia elements (like audio or video) are stored in media files. The most common way to discover the type of a file, is to look at the file extension. Multimedia files have formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

Playing Videos in HTML

To show a video in HTML, use the <video> element:

Example

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

How it Works

The controls attribute adds video controls, like play, pause, and volume. It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads. The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format. The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.

HTML <video> Autoplay

To start a video automatically use the autoplay attribute:

Example

```
<video width="320" height="240" autoplay>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Note: Autoplay attribute does not work in mobile devices like iPad and iPhone

HTML Video - Media Types

File Format	Media Type
MP4	video/mp4
WebM	video/webm
Ogg	video/ogg

HTML Video - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <video> element. This allows you to load, play, and pause videos, as well as setting duration and volume. There are also DOM events that can notify you when a video begins to play, is paused, etc.

HTML5 Video Tags

Tag	Description
<video>	Defines a video or movie
<source>	Defines multiple media resources for media elements, such as <video> and <audio>
<track>	Defines text tracks in media players

Audio on the Web

The HTML5 <audio> element specifies a standard way to embed audio in a web page.

The HTML <audio> Element

To play an audio file in HTML, use the <audio> element:

Example

```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
Your browser does not support the audio element.
</audio>
```

HTML Audio - How It Works

The controls attribute adds audio controls, like play, pause, and volume. The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format. The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.

HTML Audio - Media Types

File Format	Media Type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

HTML Audio - Methods, Properties, and Events

HTML5 defines DOM methods, properties, and events for the <audio> element. This allows you to load, play, and pause audios, as well as set duration and volume. There are also DOM events that can notify you when an audio begins to play, is paused, etc.

HTML5 Audio Tags

Tag	Description
<audio>	Defines sound content
<source>	Defines multiple media resources for media elements, such as <video> and <audio>

Introduction to IIS and XAMPP, Dynamic Website and Hosting and FTP tool Filezilla - Projects in JavaScript

Objectives : At the end of this lesson you shall be able to

- **describe XAMPP**
- **describe what is included in XAMPP**
- **describe FTP**
- **describe fileZilla**
- **describe a web project**
- **follow SDLC.**

Introduction

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP's name is an acronym for:

- X (to be read as "cross", meaning cross-platform)
- Apache HTTP Server
- MySQL
- PHP
- Perl

What's Included in XAMPP?

XAMPP has four primary components. These are:

1 **Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.

2 **MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world's most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress. You can learn how to master PHP with this free MySQL database for beginners course.

3 **PHP:** PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.

4 **Perl:** Perl is a high-level, dynamic programming language used extensively in network programming, system admin, etc. Although less popular for web development purposes, Perl has a lot of niche applications.

Different versions of XAMPP may have additional components such as phpMyAdmin, OpenSSL, etc. to create full-fledged web servers.

XAMPP features

XAMPP requires only one zip, tar, or exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache, MySQL, PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin.

Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.

It is offered in both a full, standard version and a smaller version.

Use

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

The default FTP user is "newuser", the default FTP password is "wampp". The default MySQL user is "root" while there is no default MySQL password.

XAMPP 1.8.3-4 for Windows, includes

- Apache 2.4.9
- MySQL 5.6.16
- PHP 5.5.11
- phpMyAdmin 4.1.12
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.42 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.3.1 Portable
- XAMPP Control Panel 3.2.1 (from hackattack142)

XAMPP 1.8.3-4 for Linux, includes

- Apache 2.4.9
- MySQL 5.6.16
- PHP 5.5.11
- phpMyAdmin 4.1.12
- OpenSSL 1.0.1g

XAMPP is also available for Mac OS.

File Transfer Protocol

The **File Transfer Protocol (FTP)** is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different.

The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as Web page editors.

History of FTP server

The original specification for the File Transfer Protocol was written by Abhay Bhushan and published as RFC 114 on 16 April 1971. Until 1980, FTP ran on NCP, the predecessor of TCP/IP. The protocol was later replaced by a TCP/IP version, RFC 765 (June 1980) and RFC 959 (October 1985), the current specification. Several proposed standards amend RFC 959, for example RFC 2228 (June 1997) proposes security extensions and RFC 2428 (September 1998) adds support for IPv6 and defines a new type of passive mode.

Login

FTP login utilizes a normal username and password scheme for granting access. The username is sent to the server using the USER command, and the password is sent using the PASS command. If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence. If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.

Anonymous FTP

A host that provides an FTP service may provide anonymous FTP access. Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their email address instead of a password, no verification is actually performed on the supplied data. Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.

NAT and firewall traversal

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both NATs and firewalls, which do not allow connections from the Internet towards internal hosts. For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server. This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an application-level gatewayfor this purpose.

Differences from HTTP

When operating in its modern passive mode, FTP uses a single socket for both signalling and for actual file data, just like the HTTP protocol. But when used in its original configuration, in "active mode" with a separate socket for the download, FTP exhibits true out-of-band control which is not an option with HTTP.

Web browser support

Most common web browsers can retrieve files hosted on FTP servers, although they may not support protocol extensions such as FTPS. When an FTP-rather than an HTTP-URL is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other Web content. A full-featured FTP client can be run within Firefox in the form of an extension called FireFTP.

Syntax

FTP URL syntax is described in RFC1738, taking the form: ftp://[<user>[:<password>]@]<host>[:<port>]/<url-path> The bracketed parts are optional.

For example, the URL ftp://public.ftp-servers.example.com/mydirectory/myfile.txt represents the file myfile.txt from the directory mydirectory on the server public.ftp-servers.example.com as an FTP resource. The URL ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation, such as, for example, Firefox and Internet Explorer. By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Security

FTP was not designed to be a secure protocol, and has many security weaknesses. In May 1999, the authors of RFC 2577 listed a vulnerability to the following problems:

- Brute force attacks
- Bounce attacks
- Packet capture (sniffing)
- Port stealing
- Spoof attacks
- Username protection

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture (sniffing) on the network. This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP and IMAP) that were designed prior to the creation of encryption mechanisms such as TLS or SSL. A common solution to this problem is to use the "secure", TLS-protected versions of the insecure protocols (e.g. FTPSfor FTP, TelnetS for Telnet, etc.) or a different, more secure protocol that can handle the job, such as the SFTP/SCP tools included with most implementations of the Secure Shellprotocol.

Secure FTP

Securing FTP transfers may be accomplished by several methods.

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request that the FTP session be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in the proposed standard: RFC 4217. Implicit FTPS is a deprecated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

SFTP

The SSH file transfer protocol or secure FTP (SFTP), also transfers files and has a similar command set for users, but is built on different software technology. SFTP uses theSecure Shell protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software.

FTP over SSH (not SFTP)

FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection. Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no confidentiality or integrity protection.

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwardings for FTP data channels. Software packages that support this mode include:

- Tectia ConnectSecure (Win/Linux/Unix) of SSH Communications Security's software suite
- Tectia Server for IBM z/OS of SSH Communications Security's software suite
- FONC (the GPL licensed)
- Co:Z FTPSSH Proxy

Other methods of transferring files using SSH that are not related to FTP include SFTP and SCP; in each of these, the entire conversation (credentials and data) is always protected by the SSH protocol.

FILEZILLA

FileZilla is a free FTP solution. Both a client and a server are available. FileZilla is open source software distributed free of charge under the terms of the GNU General Public License. Using FileZilla files can be uploaded or downloaded from client to server and vice-versa. It is very user friendly and no commands are required to do upload and download operations. Files can be uploaded or downloaded by simple drag-drop operations.

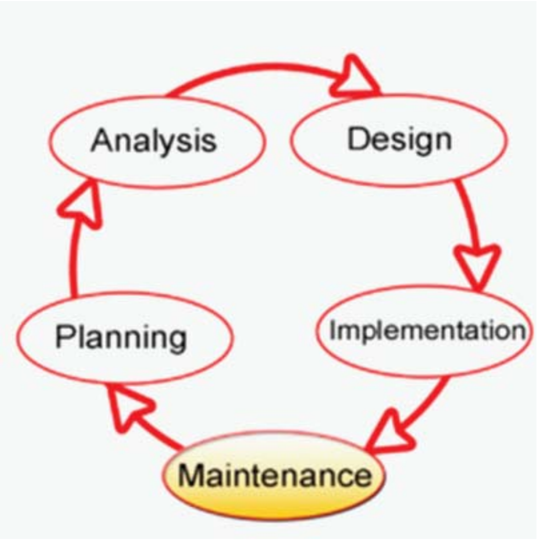
Designing a Web Project: A project in Web should be developed after comprehensive enquiry about what exactly the client/end user want. For this some meeting can be arranged with clients to find out the exact requirement of the client. This is called SRS(System Requirement Specification).

Some methods are followed for SRS, which are giving some questions abount the system to the clients and verifying the answer to gauge the requirement of the clients. Showing them some demo screen to get their response. Collecting the reports they use to understand the type of data they use.

Before development of any system or project, SRS is very important as if you cannot understand the exact user requirement, then the system/project developed by you with lot of man hours spent on it would be totally wasted and the project would be scraped.

SDLC: The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. (Fig 1)

Fig 1



The system development life cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

The SDLC adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. It includes evaluation of present system, information gathering, feasibility study and request approval. A number of SDLC models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:

- **Preliminary analysis:** The objective of phase 1 is to conduct a preliminary analysis, propose alternative solutions, describe costs and benefits and submit a preliminary plan with recommendations.
- **Conduct the preliminary analysis:** in this step, you need to find out the organization's objectives and the nature and scope of the problem under study. Even if a problem refers only to a small segment of the organization itself then you need to find out what the objectives of the organization itself are. Then you need to see how the problem being studied fits in with them.
- **Propose alternative solutions:** In digging into the organization's objectives and specific problems, you may have already covered some solutions. Alternate proposals may come from interviewing employees, clients, suppliers, and/or consultants. You can also study what competitors are doing. With this data, you will have three choices: leave the system as is, improve it, or develop a new system.

Describe the costs and benefits.

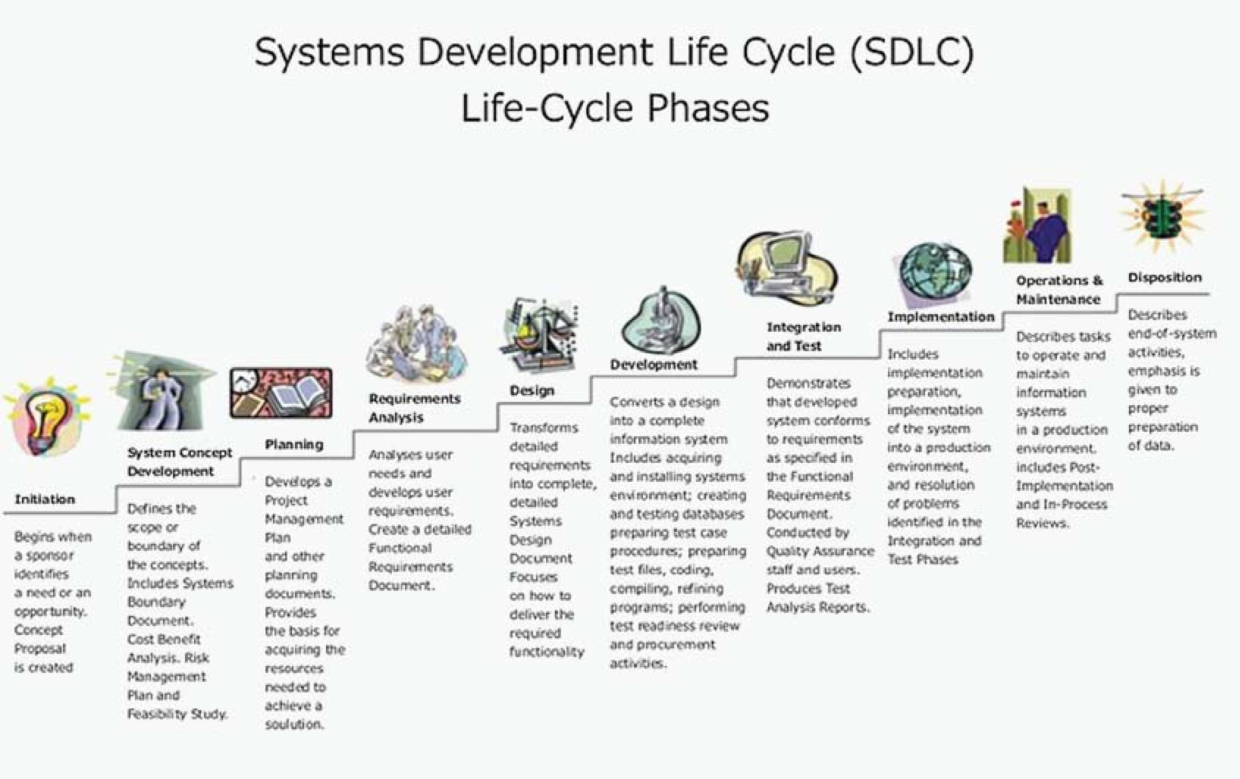
- **Systems analysis, requirements definition:** Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.
- **Development:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.
- **Maintenance:** During the maintenance stage of the SDLC, the system is assessed to ensure it does not become obsolete. This is also where changes are made to initial software. It involves continuous evaluation of the system in terms of its performance.

- **Evaluation:** Some companies do not view this as an official stage of the SDLC, but is it an important part of the life cycle. Evaluation step is an extension of the Maintenance stage, and may be referred to in some circles as Post-implementation Review. This is where the system that was developed, as well as the entire process, is evaluated. Some of the questions that need to be answered include: does the newly implemented system meet the initial business requirements and objectives? Is the system reliable and fault-tolerant? Does the system function according to the approved functional requirements? In addition to evaluating the software that was released, it is important to assess the effectiveness of the development process. If there are any aspects of the entire process, or certain stages, that management is not satisfied with, this is the time to improve. Evaluation and assessment is a difficult issue. However, the company must reflect on the process and address weaknesses.

- **Disposal:** In this phase, plans are developed for discarding system information, hardware and software in making the transition to a new system. The purpose here is to properly move, archive, discard or destroy information, hardware and software that is being replaced, in a matter that prevents any possibility of unauthorized disclosure of sensitive data. The disposal activities ensure proper migration to a new system. Particular emphasis is given to proper preservation and archival of data processed by the previous system. All of this should be done in accordance with the organization's security requirements.

In the following example (Fig 2) these stages of the systems development life cycle are divided in ten steps from definition to creation and modification of IT work products:

Fig 2



Systems Development Life Cycle (SDLC)
Life-Cycle Phases

Initiation
Begins when a sponsor identifies a need or an opportunity. Concept Proposal is created

System Concept Development
Defines the scope or boundary of the concepts. Includes Systems Boundary Document. Cost Benefit Analysis. Risk Management Plan and Feasibility Study.

Planning
Develops a Project Management Plan and other planning documents. Provides the basis for acquiring the resources needed to achieve a soulution.

Requirements Analysis
Analyses user needs and develops user requirements. Create a detailed Functional Requirements Document.

Design
Transforms detailed requirements into complete, detailed Systems Design Document Focuses on how to deliver the required functionality

Development
Converts a design into a complete information system Includes acquiring and installing systems environment; creating and testing databases preparing test case procedures; preparing test files, coding, compiling, refining programs; performing test readiness review and procurement activities.

Integration and Test
Demonstrates that developed system conforms to requirements as specified in the Functional Requirements Document. Conducted by Quality Assurance staff and users. Produces Test Analysis Reports.

Implementation
Includes implementation preparation, implementation of the system into a production environment, and resolution of problems identified in the Integration and Test Phases

Operations & Maintenance
Describes tasks to operate and maintain information systems in a production environment. includes Post-Implementation and In-Process Reviews.

Disposition
Describes end-of-system activities, emphasis is given to proper preparation of data.