Viva Question

UNIT-1

## Part I: Web Technology Basics

### 1. What is Web Technology?

- **Answer**: Web technology refers to the tools and techniques used in the creation, development, and management of websites and web applications. This includes programming languages like HTML, CSS, JavaScript, and server-side technologies like PHP, and databases like MySQL.

### 2. What is the Internet?

- **Answer**: The Internet is a global network of computers that are interconnected and communicate with each other using standardized communication protocols. It allows users to access information and services such as the World Wide Web, emails, and file sharing.

### 3. What is the World Wide Web (WWW)?

- **Answer**: The World Wide Web is a system of interlinked hypertext documents accessed via the Internet. It is a service that enables users to browse and interact with multimedia content such as text, images, and videos.

### 4. What is the difference between the Internet and the WWW?

- **Answer**: The Internet is a global network that connects computers, while the WWW is a collection of interlinked documents and resources that are accessible over the Internet using browsers.

### 5. What is HTTP?

- **Answer**: HTTP (HyperText Transfer Protocol) is a protocol used for transferring hypertext requests and information on the web. It facilitates communication between web browsers and servers.

### 6. What are HTTP request and response messages?

- **Answer**: HTTP request messages are sent by the client (browser) to request a resource from the server. HTTP response messages are sent back from the server to provide the requested resource.

**7. What is Browser Compatibility?**

- **Answer**: Browser compatibility refers to the ability of a website or web application to function correctly across various web browsers, ensuring consistent appearance and functionality.

**8. What is Cache in Web Technology?**

- **Answer**: Cache is a temporary storage area where data, such as images and web pages, is stored to reduce load times and improve website performance by avoiding repeated retrieval of the same data.

---

**□ Part II: HTML (Hypertext Markup Language)**

**9. What is HTML?**

- **Answer**: HTML is a markup language used to structure content on the web. It consists of elements (tags) that define the structure and presentation of text, images, links, forms, etc.

**10. What are the different types of HTML elements?**

- **Answer**: HTML elements include:
  - **Structural elements**: `<html>`, `<head>`, `<body>`
  - **Text formatting elements**: `<h1>`, `<p>`, `<b>`, `<i>`
  - **Forms and input elements**: `<input>`, `<form>`, `<textarea>`

**11. What is the purpose of the `<head>` and `<body>` sections in HTML?**

- **Answer**: The `<head>` section contains metadata, links to stylesheets, scripts, and other non-visible elements. The `<body>` section contains the content that is visible to users on the web page.

**12. What is the difference between an ordered list and an unordered list?**

- **Answer**: An **ordered list** (`<ol>`) displays items in a numbered sequence, while an **unordered list** (`<ul>`) displays items with bullet points.

**13. What are frames in HTML?**

- **Answer**: Frames are used to display multiple HTML documents within the same browser window. They are created using the `<frame>` and `<frameset>` tags, but they are now considered outdated.

**14. How do you create hyperlinks in HTML?**

- **Answer**: Hyperlinks are created using the `<a>` tag with the `href` attribute. Example: `<a href="https://www.example.com">Click Here</a>`.

**15. What are the different types of form elements in HTML?**

- **Answer**: HTML forms can include text inputs (`<input>`), checkboxes (`<input type="checkbox">`), radio buttons (`<input type="radio">`), text areas (`<textarea>`), buttons (`<button>`), and select menus (`<select>`).

**16. What are the differences between HTML and HTML5?**

- **Answer**: HTML5 introduces new semantic elements like `<section>`, `<article>`, `<footer>`, and `<header>`, better support for multimedia with `<audio>` and `<video>`, and improved forms and APIs.

---

**□ Part III: CSS (Cascading Style Sheets)**

**17. What is CSS?**

- **Answer**: CSS (Cascading Style Sheets) is a style sheet language used to define the presentation and layout of HTML elements, such as colors, fonts, and spacing.

**18. What is the CSS syntax?**

- **Answer**: CSS syntax consists of a selector, followed by a declaration block. A declaration block contains one or more property-value pairs. Example: `h1 { color: red; font-size: 24px; }`

**19. What are selectors in CSS?**

- **Answer**: Selectors in CSS are used to target HTML elements. Types of selectors include:

    - **Element selectors**: `h1, p`
    - **Class selectors**: `.classname`
    - **ID selectors**: `#idname`
    - **Attribute selectors**: `[type="text"]`

**20. What is the difference between inline, document-level, and external CSS?**

- **Answer**:
    - **Inline CSS**: Applied directly within an HTML element using the `style` attribute.
    - **Document-level CSS**: Defined within a `<style>` block in the HTML `<head>`.
    - **External CSS**: Defined in an external `.css` file linked to the HTML document.

**21. What is the box model in CSS?**

- **Answer**: The CSS box model defines the rectangular boxes generated for elements, consisting of margins, borders, padding, and the content area.

**22. What are CSS pseudo-classes?**

- **Answer**: CSS pseudo-classes are used to define special states of elements, such as `:hover`, `:focus`, and `:active`.

**23. What is the importance of the `!important` rule in CSS?**

- **Answer**: The `!important` rule is used to give higher specificity to a CSS property, overriding other declarations of the same property.

---

**□ Part IV: Bootstrap**

**24. What is Bootstrap?**

- **Answer**: Bootstrap is a front-end framework that provides pre-built components like grids, buttons, navigation, and forms, helping developers create responsive and mobile-first websites quickly.

**25. What is the Grid System in Bootstrap?**

- **Answer**: The Bootstrap grid system is a 12-column layout system that allows for responsive web design. It helps structure web content using rows and columns that automatically adjust for different screen sizes.

**26. What is a container in Bootstrap?**

- **Answer**: A container in Bootstrap is a fixed-width element that holds the content and provides padding. There are two types of containers: `.container` (fixed-width) and `.container-fluid` (full-width).

**27. How do you use buttons in Bootstrap?**

- **Answer**: Buttons in Bootstrap are created using the `<button>` or `<a>` elements with specific classes like `.btn`, `.btn-primary`, `.btn-secondary`, etc.

**28. What is the difference between `class="container"` and `class="container-fluid"` in Bootstrap?**

- **Answer**:
  - **`.container`**: Provides a fixed-width layout that changes according to screen size.
  - **`.container-fluid`**: Provides a full-width layout that spans the entire width of the viewport.

Unit-2

## Part I: JavaScript

### 1. What is JavaScript?

- JavaScript is a lightweight, interpreted programming language used for adding interactivity to web pages, running on the client side (in the browser).

### 2. What is the difference between Java and JavaScript?

- Java is a **class-based, object-oriented** language used for server-side development, while JavaScript is a **lightweight, interpreted scripting** language primarily used for client-side scripting in web pages.

### 3. What is the syntax of JavaScript?

- JavaScript syntax defines rules for writing valid code, including how variables, functions, and control structures are declared.

### 4. Explain identifiers, keywords, and comments in JavaScript.

- **Identifiers** are names for variables, functions, or objects.
- **Keywords** are reserved words that cannot be used as identifiers (e.g., `var`, `if`, `function`).
- **Comments** are used to annotate code, like `// single-line comment` or `/* multi-line comment */`.

### 5. What are the data types in JavaScript?

- JavaScript supports **primitive types** like `Number`, `String`, `Boolean`, `null`, `undefined`, and `Symbol`, as well as **complex types** like `Object` and `Array`.

### 6. What is a variable in JavaScript?

- A variable stores data that can be accessed and modified later. Declared using `var`, `let`, or `const`.

### 7. Explain the different types of operators in JavaScript.

- **Arithmetic operators**: `+`, `-`, `*`, `/`, etc.

- **Comparison operators**: ==, ===, !=, etc.
- **Logical operators**: `&&`, `||`, `!`.
- **Assignment operators**: =, +=, −=, etc.

### 8. How does `document.write()` work in JavaScript?

- `document.write()` writes text or HTML directly into the web page during the page loading process.

### 9. What are the different types of loops in JavaScript?

- **`while` loop**: Executes as long as the condition is true.
- **`do...while` loop**: Executes at least once before checking the condition.
- **`for` loop**: Repeats a block of code a specified number of times.
- **`for...in` loop**: Iterates over the properties of an object.

### 10. What is a function in JavaScript?

- A function is a block of code that performs a specific task and can be called repeatedly with different arguments.

### 11. What is the scope of a variable in JavaScript?

- **Local scope**: Variables declared inside a function are local to that function.
- **Global scope**: Variables declared outside any function are accessible throughout the script.

### 12. How are arrays defined and manipulated in JavaScript?

- **Declaring an array**: `let arr = [1, 2, 3];`
- **Manipulating arrays**: Adding elements with `push()`, removing elements with `pop()`, and accessing elements by index.

### 13. What are built-in objects in JavaScript?

- JavaScript provides several built-in objects like `Math`, `Date`, `String`, `Boolean`, `Number`, and `Array`, each with its own set of methods and properties.

### 14. Explain the concept of `this` in JavaScript.

- The `this` keyword refers to the **current object** in context, often used inside methods to access the calling object.

---

## Part II: DOM (Document Object Model)

### 15. What is the Document Object Model (DOM)?

- DOM is a programming interface for web documents. It represents the structure of a document as a tree of nodes, allowing manipulation of HTML or XML documents.

### 16. How can you modify an element's style in JavaScript using DOM?

- You can modify the style using `element.style.property = value;`, e.g., `document.getElementById("myDiv").style.color = "red";`.

### 17. What is DOM event handling?

- DOM events allow JavaScript to respond to user actions, like clicks or keyboard inputs. You can attach event listeners to elements using methods like `addEventListener()`.

### 18. How would you validate a form using JavaScript?

- Form validation is typically done by checking form field values and using JavaScript to ensure they meet certain criteria before submission.

---

## Part III: jQuery

### 19. What is jQuery?

- jQuery is a fast, lightweight JavaScript library that simplifies HTML document traversal, event handling, and animation.

### 20. How do you include jQuery in your project?

- You can include jQuery by using a CDN link like: `<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>`

### 21. What are the basic jQuery syntax elements?

- **Selectors**: `$()` is used to select elements (e.g., `$(".class")`, `$("#id")`).
- **Methods**: `.css()`, `.hide()`, `.show()`, etc., to manipulate elements.

### 22. How do you manipulate the DOM using jQuery?

- jQuery allows easy manipulation of DOM elements using functions like `.text()`, `.html()`, `.css()`, `.attr()`, etc.

### 23. How do you append or remove elements using jQuery?

- **Appending**: `$("selector").append("<div>New Element</div>");`
- **Removing**: `$("selector").remove();`

---

## Part IV: AngularJS

### 24. What is AngularJS?

- AngularJS is a JavaScript framework used for building dynamic web applications. It uses **MVC (Model-View-Controller)** architecture.

### 25. What is the role of directives in AngularJS?

- Directives are special markers in AngularJS that attach behavior to DOM elements. Examples include `ng-app`, `ng-model`, `ng-repeat`.

### 26. Explain AngularJS controllers.

- Controllers in AngularJS are JavaScript functions that control the data and behavior of views (HTML). They are typically associated with the `$scope` object.

### 27. What is dependency injection in AngularJS?

- Dependency injection is a design pattern where AngularJS injects dependencies (services, factories, etc.) into controllers, making code modular and testable.

### 28. What are AngularJS filters?

- Filters are used to format data for display in views, e.g., `currency`, `date`, `uppercase`, etc.

### 29. Explain the concept of scopes in AngularJS.

- Scopes in AngularJS are objects that allow communication between the controller and the view. They hold the model data.

### 30. What is two-way data binding in AngularJS?

- Two-way data binding ensures that any changes made to the model (JavaScript data) automatically update the view, and vice versa.

### 31. How are forms handled in AngularJS?

- Forms in AngularJS are managed using directives like `ng-model` to bind form input elements to application data.

### 32. What is the `$http` service in AngularJS?

- The `$http` service allows you to make AJAX requests to the server and retrieve data asynchronously.

# UNIT-3

## Part I: Servlets

### 1. What is Server Side Programming?

- It refers to operations that are performed by the server in a client-server relationship, such as database queries, business logic, and page generation (like with servlets).

### 2. What is a Servlet?

- A Servlet is a Java class that runs on a web server, handles client requests, processes them, and returns a response.

### 3. Explain the Servlet architecture.

- Servlet Architecture follows the request-response model where the **client sends a request**, the **web server passes it to the servlet**, and the **servlet processes and sends back a response**.

### 4. What are the advantages of Servlets?

- Portable (platform independent), efficient, scalable, secure, and can handle complex tasks like database access and session tracking.

### 5. What is the life cycle of a Servlet?

- The Servlet lifecycle involves:
    1. **Loading and Instantiation**
    2. **Initialization (`init()`)**
    3. **Request handling (`service()`)**
    4. **Destruction (`destroy()`)**

### 6. Which packages are used for Servlets?

- `javax.servlet` and `javax.servlet.http`.

### 7. What are some key interfaces in `javax.servlet` package?

- `Servlet`, `ServletRequest`, `ServletResponse`, `ServletConfig`, `ServletContext`.

### 8. What is the difference between `Servlet` and `HttpServlet`?

- `Servlet` is a generic protocol-independent interface.
- `HttpServlet` is a protocol-specific (HTTP) class that extends the capabilities of Servlets for handling HTTP-specific services.

### 9. What is the use of `HttpServletRequest` and `HttpServletResponse`?

- `HttpServletRequest`: To access request information like parameters, headers, and attributes.
- `HttpServletResponse`: To send response information like setting content type, writing output, etc.

### 10. How do you retrieve parameter data from a Servlet?

- Using `request.getParameter("paramName")`.

### 11. What is Session Management?

- It's a mechanism to maintain a user's state (data) across multiple HTTP requests.

### 12. What are Cookies in Servlets?

- Cookies are small data pieces stored on the client's browser to maintain user sessions or preferences.

### 13. What is URL Rewriting?

- Adding session information directly into the URL to maintain state when cookies are disabled.

### 14. How do Servlets handle concurrency?

- Servlets create a single instance to handle multiple requests using multiple threads, so care must be taken to avoid thread-safety issues.

### 15. What is JDBC?

- JDBC (Java Database Connectivity) is an API that allows Java applications to interact with databases.

### 16. Difference between JDBC and ODBC?

- JDBC is designed for Java applications, platform-independent.
- ODBC is language-independent but platform-dependent.

### 17. What are the steps to connect a Servlet to a Database?

1. Load JDBC driver.
2. Establish a connection.
3. Create a statement.
4. Execute the query.
5. Process the results.
6. Close the connection.

---

## Part II: XML

### 18. What is XML?

- XML (eXtensible Markup Language) is a markup language for storing and transporting data.

### 19. What are XML vocabularies?

- Predefined sets of XML tags and structures used in specific domains (like MathML, SVG).

### 20. What is the purpose of an XML declaration?

- It defines XML version and character encoding used in the document (e.g., `<?xml version="1.0" encoding="UTF-8"?>`).

### 21. What are DTDs in XML?

- DTD (Document Type Definition) defines the structure and the legal elements/attributes in an XML document.

### 22. What is Schema in XML?

- XML Schema defines the structure, content, and data types of XML documents (more powerful than DTD).

### 23. Advantages of Schema over DTD?

- Schema supports data types, namespaces, is extensible, and uses XML syntax itself.

### 24. What is XML Namespace?

- Namespaces avoid element name conflicts by qualifying names with a URI.

### 25. What is DOM in XML?

- DOM (Document Object Model) represents the structure of XML documents as a tree of objects for easy processing.

### 26. What is XSLT?

- XSLT (Extensible Stylesheet Language Transformations) is used to transform XML documents into other formats (like HTML).

### 27. Difference between XML and XSLT?

- XML is for data storage.
- XSLT is for transforming that data into another format (like displaying it in a webpage).

---

## Part III: AJAX

### 28. What is AJAX?

- AJAX (Asynchronous JavaScript and XML) allows web pages to be updated asynchronously by exchanging data with the server behind the scenes.

### 29. How does AJAX work?

- It uses `XMLHttpRequest` object to send asynchronous requests to the server without refreshing the page.

### 30. What is XMLHttpRequest?

- It’s a JavaScript object that allows browser-client communication with the server in the background.

## 31. What are the advantages of AJAX?

- Faster responses, improved user experience, reduces server load.

## 32. What are the steps to create an AJAX request?

1. Create XMLHttpRequest object.
2. Open a connection using `open()` method.

# UNIT-4

## JavaServer Pages (JSP)

1. **What is JSP?**
   - JSP (JavaServer Pages) is a server-side technology that enables the creation of dynamic, platform-independent web content by embedding Java code into HTML pages.
2. **Explain the JSP life cycle.**
   - The JSP life cycle includes: Translation of JSP to Servlet, Compilation, Loading, Instantiation, Initialization (`jspInit()`), Request Processing (`_jspService()`), and Destruction (`jspDestroy()`).
3. **What are JSP scripting elements?**
   - JSP supports three scripting elements:
     - **Declarations (`<%! %>`)**: Declare variables and methods.
     - **Scriptlets (`<% %>`)**: Embed Java code within the HTML.
     - **Expressions (`<%= %>`)**: Output the value of Java expressions.
4. **What are JSP directives?**
   - JSP directives provide global information about the JSP page and include:
     - **Page Directive**: Defines page-dependent attributes.
     - **Include Directive**: Includes a file during the translation phase.
     - **Taglib Directive**: Declares a tag library, containing custom tags.
5. **What are implicit objects in JSP?**
   - Implicit objects are pre-defined variables available in JSP pages, such as `request`, `response`, `session`, `application`, `out`, `config`, `pageContext`, `page`, and `exception`.
6. **Differentiate between `<jsp:include>` and include directive.**
   - The `include` directive (`<%@ include file="..." %>`) includes content at translation time, whereas `<jsp:include>` includes content at request time, allowing for dynamic inclusion.
7. **What is Expression Language (EL) in JSP?**
   - EL simplifies the accessibility of data stored in JavaBeans components, request, session, and application scopes, using a concise syntax like `${user.name}`.
8. **How does JSP support the MVC architecture?**
   - In MVC:

- **Model**: Represents the application's data and business logic.
- **View**: JSP acts as the view, rendering the user interface.
- **Controller**: Servlets handle the request processing and control flow.

---

## 🌐 Web Services

1. **What is a Web Service?**
   - A Web Service is a standardized way of integrating web-based applications using open standards over an internet protocol backbone.
2. **Differentiate between SOAP and RESTful Web Services.**
   - **SOAP**: Protocol-based, uses XML, supports complex operations.
   - **REST**: Architectural style, uses standard HTTP methods, supports multiple formats like JSON and XML.
3. **What is WSDL?**
   - WSDL (Web Services Description Language) is an XML-based language for describing the functionalities offered by a web service.
4. **Explain the structure of a SOAP message.**
   - A SOAP message consists of:
     - **Envelope**: Defines the start and end of the message.
     - **Header**: Contains optional attributes of the message.
     - **Body**: Contains the XML data comprising the message.
     - **Fault**: Provides information about errors that occurred.
5. **What is UDDI?**
   - UDDI (Universal Description, Discovery, and Integration) is a directory service where businesses can register and discover web services.
6. **What is the purpose of using RESTful Web Services?**
   - RESTful services are lightweight, scalable, and stateless, making them suitable for web and mobile applications.
7. **What are the HTTP methods used in REST?**
   - Common HTTP methods include:
     - **GET**: Retrieve data.
     - **POST**: Create new resources.
     - **PUT**: Update existing resources.
     - **DELETE**: Remove resources.

8. **What is the role of XML in Web Services?**
    - XML is used for data representation and transmission in web services, ensuring platform-independent communication.

---

## Struts Framework

1. **What is Struts?**
    - Struts is an open-source framework for creating Java EE web applications, utilizing the Model-View-Controller (MVC) architecture.
2. **Explain the architecture of Struts.**
    - Struts architecture comprises:
        - **Model**: Business logic and data.
        - **View**: JSP pages for user interface.
        - **Controller**: `ActionServlet` that handles requests and responses.
3. **What is an ActionForm in Struts?**
    - `ActionForm` is a JavaBean that represents the form inputs from the client, facilitating data transfer between the view and the controller.
4. **What is the role of `struts-config.xml`?**
    - This configuration file defines the mapping between user requests, Action classes, and the corresponding views.
5. **What are Interceptors in Struts2?**
    - Interceptors are components that process requests before and after the execution of an action, used for cross-cutting concerns like logging and validation.
6. **How does validation work in Struts?**
    - Struts provides a validation framework where rules are defined in XML or annotations, and validations are performed before the action method is invoked.
7. **What is the purpose of the `ActionServlet`?**
    - `ActionServlet` acts as the controller in the MVC pattern, intercepting user requests and dispatching them to the appropriate Action classes.
8. **How does Struts support internationalization?**
    - Struts supports internationalization through resource bundles, allowing applications to support multiple languages and locales.

# UNIT-5

Updated saved memory

Here are some commonly asked viva (oral) questions and their answers based on the topics you've provided for PHP, ASP.NET, and Node.js. These questions are designed to help you prepare effectively for your exam.

---

## 🐘 PHP Viva Questions & Answers

### 1. What is PHP and who created it?

PHP stands for "Hypertext Preprocessor." It is an open-source, server-side scripting language designed for web development. PHP was created by Rasmus Lerdorf in 1994. Simplilearn.com

### 2. What are the scalar data types in PHP?

PHP supports four scalar data types: integer, float (double), string, and boolean.

### 3. How do you declare a variable in PHP?

Variables in PHP are declared using the `$` symbol followed by the variable name. For example:

```
php
CopyEdit
$name = "John";
```

### 4. Explain type juggling in PHP.

Type juggling refers to PHP's automatic type conversion. For instance, adding a string and an integer will result in PHP converting the string to a number, if possible.

### 5. Differentiate between == and === operators.

- `==` checks for value equality after type conversion.

- `===` checks for both value and data type equality.

### 6. What is the difference between `session_unset()` and `session_destroy()`?

- `session_unset()` clears all session variables.
- `session_destroy()` destroys the session data on the server. GeeksforGeeks

### 7. How do you connect to a MySQL database using PHP?

Using the `mysqli_connect()` function:

```
php
CopyEdit
$conn = mysqli_connect("localhost", "username",
"password", "database");
```

### 8. What are cookies and how are they set in PHP?

Cookies are small files stored on the client's computer. In PHP, you can set a cookie using the `setcookie()` function:

```
php
CopyEdit
setcookie("user", "John", time() + (86400 * 30),
"/");
```

---

## 🌐 ASP.NET Viva Questions & Answers

### 1. What is ASP.NET?

ASP.NET is a web application framework developed by Microsoft for building dynamic web pages and applications. It is part of the .NET platform. GeeksforGeeks

### 2. What are the features of ASP.NET?

- Built-in caching
- State management
- Security features
- Rich toolbox and designer in Visual Studio

- Support for multiple languages via Common Language Runtime (CLR) Expert Talent ConnectGist+2Turing+2DEV Community+2

### 3. Explain the role of the CLR in ASP.NET.

The Common Language Runtime (CLR) is the execution engine for .NET applications, providing services like memory management, type safety, exception handling, and garbage collection.

### 4. What is the difference between ASP.NET Web Forms and MVC?

- Web Forms follow an event-driven development model.
- MVC (Model-View-Controller) separates the application into three components, promoting a clear separation of concerns.AI Talent Assessment Platform - WeCP

### 5. What is the purpose of the `web.config` file?

The `web.config` file is used to configure application settings, such as database connections, session states, error handling, and security configurations.

### 6. How does ASP.NET handle state management?

ASP.NET provides several state management techniques:

- View State
- Session State
- Application State
- Cookies
- Query StringsInterviewBit

---

## ⚙ Node.js Viva Questions & Answers

### 1. What is Node.js?

Node.js is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a web browser. It is built on Chrome's V8 JavaScript engine. GeeksforGeeks+1Simplilearn.com+1

### 2. What is NPM?

NPM stands for Node Package Manager. It is the default package manager for Node.js, used to install, share, and manage dependencies in Node.js applications. GeeksforGeeks

### 3. Why is Node.js single-threaded?

Node.js uses a single-threaded event loop model to handle multiple concurrent clients. This design makes it lightweight and efficient for I/O-heavy operations. Reddit

### 4. What are the types of API functions in Node.js?

- Synchronous (blocking)
- Asynchronous (non-blocking)Expert Talent Connect+7GeeksforGeeks+7Zero To Mastery+7

### 5. How do you create a simple HTTP server in Node.js?

Using the built-in `http` module:

```
javascript
CopyEdit
const http = require('http');
const server = http.createServer((req, res) => {
  res.write('Hello World');
  res.end();
});
server.listen(3000);
```

Zero To Mastery+1GeeksforGeeks+1

### 6. What is the purpose of the `package.json` file?

The `package.json` file holds metadata relevant to the project and is used to manage the project's dependencies, scripts, version, and more.

# UNIT-6

## Part I: Ruby Programming

### 6.1 Origins & Uses of Ruby

**Q:** Who created Ruby and when?
**A:** Ruby was created by Yukihiro Matsumoto in 1995 in Japan.

**Q:** What are the main uses of Ruby?
**A:** Ruby is mainly used for web development, scripting, automation, and building web applications with frameworks like Ruby on Rails.

---

### 6.2 Scalar Types and their Operations

**Q:** What are scalar types in Ruby?
**A:** Scalar types include numeric types (integers, floats), strings, and symbols that hold single values.

---

### 6.2.1 Numeric and String Literals

**Q:** Give an example of a numeric and a string literal in Ruby.
**A:** Numeric: `42`, `3.14`; String: `"Hello"`, `'Ruby'`

---

### 6.2.2 Variable and Assignment Statements

**Q:** How do you assign a value to a variable in Ruby?
**A:** Use the = operator. Example: `name = "Ruby"`

---

### 6.2.3 Numeric Operators

**Q:** Name any two numeric operators in Ruby.
**A:** `+` (addition), `*` (multiplication)

---

### 6.2.4 String Methods

**Q:** How do you convert a string to uppercase in Ruby?
**A:** Use the `.upcase` method. Example: `"hello".upcase` gives `"HELLO"`

---

### 6.3 Simple Input and Output

**Q:** How do you take input from the user in Ruby?
**A:** Using `gets.chomp`

**Q:** How do you print output in Ruby?
**A:** Using `puts` or `print`

---

### 6.4 Control Statements

**Q:** What are control statements in Ruby?
**A:** They are used to make decisions (if, unless) and repeat actions (while, for, until).

---

### 6.4.2 Selection and Looping Statements

**Q:** What looping constructs are available in Ruby?
**A:** `while`, `until`, `for`, and `.each`

---

### 6.5 Fundamentals of Arrays

**Q:** What is an array in Ruby?
**A:** An array is an ordered collection of values like `[1, 2, 3]`

---

### 6.6 Hashes

**Q:** What is a hash in Ruby?
**A:** A hash is a collection of key-value pairs, like `{name: "Ruby", version: 3}`

---

### 6.7 Methods

**Q:** How do you define a method in Ruby?
**A:** Using the `def` keyword. Example: `def greet; puts "Hello"; end`

---

### 6.8 Classes

**Q:** How do you define a class in Ruby?
**A:** Using `class` keyword. Example:

```
ruby
CopyEdit
class Person
  def initialize(name)
    @name = name
  end
end
```

---

### 6.9 Code Blocks and Iterators

**Q:** What is a block in Ruby?
**A:** A block is a chunk of code enclosed in `{}` or `do...end`, often used with iterators.

---

### 6.10 Pattern Matching

**Q:** What is pattern matching in Ruby used for?
**A:** It's used to check data structures and extract values from them.

---

## Part II: Ruby on Rails

### 6.11 Overview of Rails

**Q:** What is Ruby on Rails?
**A:** Ruby on Rails is a web application framework written in Ruby, following the MVC architecture.

---

### 6.12 Static and Dynamic Documents

**Q:** What’s the difference between static and dynamic documents in Rails?
**A:** Static documents do not change, dynamic ones are generated based on user input or database content.

---

### 6.13 Processing Forms

**Q:** How does Rails handle form submission?
**A:** Using forms in views that submit data to controller actions which process and save it.

---

### 6.14 Rails Applications and Databases

**Q:** How do Rails apps connect to databases?
**A:** Through ActiveRecord, the ORM that maps database tables to Ruby classes.

---

## Part III: EJB (Enterprise JavaBeans)

### 6.17 Introduction to EJB

**Q:** What is EJB?
**A:** Enterprise JavaBeans is a Java EE technology for building scalable, transactional, and secure enterprise applications.

---

### 6.17.1 Types of EJB

**Q:** What are the types of EJBs?
**A:** Session Beans, Message-driven Beans, and Entity Beans.

---

### 6.17.2 Benefits of EJB

**Q:** Mention one benefit of using EJB.
**A:** It simplifies development by handling transactions, security, and scalability.

**-MADE BY SAHIL WANKHEDE**