A diagram of a course

Description automatically generated with medium confidenceSAP

**A diagram of a diagram

Description automatically generated**

What is SAP UI5

SAP UI5 is the new UI technology from SAP to build **responsive web** applications and is the successor of SAP GUI, Webdynpro and BSP. It is a framework which is based on libraries to quick design user experience.

Framework - collection of libraries

Library - collection of classes

Class - collection of properties, events, methods and aggregations

So basically SAP provide the facility to build our user experience, which is going to help the companies to build web applications which can run on mobile devices any platform and any browser.

**Responsive Web Application** - An application which adopt itself according the device, platform and browser. so that is the definition of responsive application.

So when we will build responsive web application using UI5 then make sure that application works on any platform and any device.

Follow Below Steps in order to build UI5 application

1. We are going to launch our development tool call SAP Business Application Studio. (It is a web-based development tool available in free.)
2. Create a basic skeleton of our project (Which is nothing but a folder inside the best tool). The Folder should have webapp folder (web-application folder)
3. *A screenshot of a computer

   Description automatically generated*Add a index.html file to the webapp folder. *[when we start building applications, it starts with this html file called* ***index.html****]*.

Basically html file consists of **2 parts** -> Header Body.

A screenshot of a computer screen

Description automatically generatedHeader is brain and

Body is what the content we want to show to the user.

1. Now we use **ui5 framework** in html file, to use ui5 framework we need to call them. So basically we use **script tag** in html to load or call 3rd party libraries/framework. We use this tag in **header part**

|  |  |  |
| --- | --- | --- |
| Factors | SAP Business Application Studio (BAS) | VS Code |
| Usage : Online/Offline | The tool can only be used in online mood | The tool can be used in offline mood |
| Owns this tool | SAP is offering this tool in SAP BTP(Business Technology Platform) | Offered by microsoft, which is for download |
| Cost | Trial mode for 90 days it is free. But after, it will cost 2-6 EUR per month per user. | Lifetime free |
| Speed | Sometimes the speed and performance is not good. | It depends on local machine, but still its performance is very good. |
| Responsible for code backup | All code which we write are store on internet, so anytime if device is lost or damaged, we can continue with other device as well. | As it is offline, code is not store on internet. Here you are responsible for backup. |
| Tech | It is a baby of VS code, infact SAP downloaded the source code of vs code and published as BAS | It is more supreme and it is original. |
| Limitations | It is a multi-cloud development tool provides all development capabilities for UI5, Native HANA | As per UI5 concerned except adaptation project, it supports everything |

**Development tools for Ui5 Applications (video-02)**

# BTP (Business Technology Platform)

BTP stands for Business Technology Platform which is platform as a Service (PaaS) to allow development teams to design, develop, deliver, test, manage end to end applications in the cloud.

For learning purpose for 90 days SAP is offering free BTP account – we can register through gmail.

Runtime

Tools

Virtualization

OS

BTP

Hardware

Networking

So before develop/build any software we need to have setup for all this things **hardware, software, networking, OS, Virtualization, Tools, Runtime** for each employee. So basically it is a time consuming and expensive. So to remove this problem BTP is providing a platform, where we no need to setup all these things, just we need to take subscription for BTP and we can continue to building the app.

# BAS (Business Application Studio)

A screenshot of a computer

Description automatically generatedBAS is a multi-cloud development tool which is a development tool service inside BTP. SAP recommend this tool for development applications for SAP UI5 and Fiori technology. This tool can be used only when you are connected to internet.

# Dev Space

A dev space is private area for developer which confine the requirements to develop a particular application. SAP installs all the recommended tools needed to build Fiori apps in the dev space for us. It is a virtual machine running in the cloud.

* It isolate developers work from each other.
* When a developer does something wrong, which cause a system corruption, it won’t affect to the other developer.
* In trial account we can run max 2 dev space. And one will be running at a time.
* After starting the dev space, click on the **open folder** and choose Projects
* In the Project folder create a new project.
* Add index.html file with some content.
* Go to extension and install Live server extension (one time)
* Right click on the index.html file and choose run with five server.

# VS Code – Tool in local machine

Pre-requisites

1. Download and install node js. [Click here and Get Link](https://nodejs.org/en/download) (Only **v16.19.0**)
2. Download and install vs code. [Click here and Get Link](https://code.visualstudio.com/download)
3. You can check node js is ready or not by command : node -v
4. VS codes needs a workspace, it is a folder on your computer where you keep all the projects and files
5. Once VS code is installed, we also need to download the live server (Like we did in BAS)

# Client Server Architecture**(video-03)**

Client-server architecture is a type of computer network where multiple clients request and receive files and services from a centralized server over a local or internet connection. A client uses an application as an interface to connect to the server.

# Request Response Architecture

Server

Browser(Client)

[Mobile, Workstation, Laptop]

Request- ***header, body(opt)***

Internet

**port**

response – ***header, body (html, text, pdf, image, Json, xml, word)***

* GET – read the data from server
* POST – Insert data from server
* PUT – Update the data
* DELETE – Delete the data

types of request

**Port** – it is like a door, once request goes first it comes to port and port is listen which type of request it is then it process in the server and from the data -base it validate the request and then it gives the response. This is how any web application works, and this is how also our Fiori application will work.

So when we want to build applications and run from client side, then we need to understand the Web Technologies

**Web Technologies**

* **HTML**
* **CSS**
* **Java Script**
* **J Query**

HTML And HTML5

HTML stands from hyper text markup language. It is used to create static web page. We can not do any dynamic operation in web page with help of html. HTML introduced in 1991.

* HTML is markup language, not programming language.
* HTML is purely tag based
* These tags are released by the company named W3 School.
* All browser vendors are part of it.
* Browser directly understand html.
* An HTML page consists of head (brain) and body (content part user will see)

|  |  |
| --- | --- |
| YEAR | VERSION |
| 1991 | HTML |
| 1995 | HTML 2.0 |
| 1997 | XHTML |
| 1999 | HTML 4.0 |
| 2014 | HTML 5 |

# Difference between HTML5 and UI5

HTML5 is a markup language and it doesn't have programming capabilities but SAP UI5 is a framework which is based on libraries approach to build web applications.

 HTML5 is mainly for the purpose of creating a simple webpage without formatting and logic but UI5 provides standard style and components to build rich UIs

# Funda Fox

* An HTML page is also known as HTML document.
* All HTML elements will follow below syntax.

<tagName property=””>CONTENT</tagName>

* There are some exceptions where you can skip writing end-tag.
* HTML is case sensitive, but it is good practice to write tags in small letters.
* We can use CTRL + SPACE to show code completion, in any tools.
* HTML at runtime produced a tree data structure, where the nodes of the tree are html element,
* This data structure is known as ***Document Object Model (DOM)***
* We can see this data structure by clicking **F12** key

# What is DOM

A screenshot of a computer code

Description automatically generatedDom is nothing but a tree data structure, which is produced by html, when you execute html. where the nodes of the tree are html element,

# HTML <!DOCTYPE> Declaration

All HTML documents must start with a <!DOCTYPE> declaration.

The declaration is not an HTML tag. It is an "information" to the browser about what document type to expect.

A screen shot of a computer

Description automatically generatedIn HTML 5, the declaration is simple: <!DOCTYPE html>

# HTML <meta> Tag

The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.

<meta> tags always go inside the <head> element, are typically used to specify character set, page description, keywords, author of the document, and viewport settings. Metadata is used by browsers (how to display content or reload page), search engines (keywords), and other web services.

A screen shot of a computer code

Description automatically generated

The charset attribute specifies the character encoding for the HTML document.

The description attribute specifies the description of my web page.

The author specifies the author of the web page

The keyword specifies the keyword of the web page, this keyword used by the search engine.

# HTML Headings

HTML headings are defined with the <h1> to <h6> tags.

<h1> defines the most important heading. <h6> defines the least important heading.

A screenshot of a computer code

Description automatically generatedA white background with black text

Description automatically generated

# HTML <em> Tag

The <em> tag is used to define emphasized text. The content inside is typically displayed in **italic.**

# Comments in html

To comment the html line enter the Command **CTRL + /**

# HTML <p> Tag

The <p> tag defines a paragraph in html.

# HTML <a> Tag

The <a> tag defines a hyperlink, which is used to link from one page to another.

The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

# HTML <img> Tag

The <img> tag is used to embed an image in an HTML page.

The <img> tag has two required attributes:

**src** - Specifies the path to the image

**alt** - Specifies an alternate text for the image, if the image for some reason cannot be displayed

**Note:** Also, always specify the width and height of an image

# HTML Unordered Lists

The HTML <ul> tag defines an unordered (bulleted) list.

An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.

The list items will be marked with bullets (small black circles) by default:

# HTML Ordered Lists

The HTML <ol> tag defines an ordered list. An ordered list can be numerical or alphabetical.

An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.

The type attribute of the <ol> tag, defines the type of the list item marker:

|  |  |
| --- | --- |
| type="1" | The list items will be numbered with numbers (default) |
| type="A" | The list items will be numbered with uppercase letters |
| type="a" | The list items will be numbered with lowercase letters |
| type="I" | The list items will be numbered with uppercase roman numbers |
| type="i" | The list items will be numbered with lowercase roman numbers |

A screenshot of a computer code

Description automatically generated

# HTML <label> Tag

In the label tag we define a attribute **for** that binds or connect to the input element such as text, email, password, text area etc. So that when ever user click on the label it directly redirects to the user to the input.

**Note :** The <label> tag needs a ***for*** attribute whose value is the same as input **id.**

# HTML <input> Tag

The <input> tag specifies an input field where the user can enter data.

The <input> element is the most important form element.

A screenshot of a computer program

Description automatically generatedThe <input> element can be displayed in several ways, depending on the **type** attribute.

# HTML <form> Tag

An **HTML form** is *a section of a document* which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing. HTML forms are required if you want to collect some data from of the site visitor.

**There are two attributes of the form tag that we should be familiar with:**

* There is a attribute called **action** in the form tag, where we pass the server URL, so when user submit the form it redirects to that URL.
* The **method** attribute is used to upload the data. The most commonly used attributes are the GET and POST methods. Default is **GET**

# HTML – Blocks (Video-4)

All the HTML elements can be categorized into two categories **(a)** Block Level Elements **(b)**Inline Elements.

# Block Elements

Block elements are the type of elements who is having line break before and after them. example, the <p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6> etc are the block level element.

# Inline Elements

Inline elements, on the other hand, can appear within sentences and do not have to appear on a new line of their own. The <b>, <i>, <u>, <em> etc are all inline elements.

# The <div> tag

This is the very important block level tag/element which plays a big role in grouping various other HTML elements and applying CSS on group of elements.

# The <span> tag

The HTML <span> is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

# HTML Audio

The HTML <**audio**> element is used to play an audio file on a web page. The **controls** attribute adds audio controls, like play, pause, and volume. **Src** It specifies the source URL of the audio file.

# HTML Video

The HTML <**video**> element is used to show a video on a web page. The **controls** attribute adds audio controls, like play, pause, and volume. **Src** It specifies the source URL of the video file.

# HTML Tables

HTML **table** tag allow web developers to arrange data into rows and columns.

In consists of **thead** and **tbody.**

The <thead> tag is used to group header content in an HTML table.

The <th> tag defines a header cell in an HTML table.

The <tbody> tag is used to group the body content in an HTML table.

The <td> tag defines a standard data cell in an HTML table.

The <tr> tag defines a row in an HTML table.

# HTML Iframe

An HTML iframe is used to display a web page within a web page. The HTML <iframe> tag specifies an inline frame.

# Syntax

<iframe src="*url*" title="description"></iframe>

Use the height and width attributes to specify the size of the iframe.

# CSS

CSS stands for cascading style sheet. We can create HTML content but we can not format is using html. If we use html formatting, it becomes a nightmare for developers because of high efforts required.

To solve this problem CSS came into picture, It is used to format and beautify html content without much effort.

***There are 3 ways we can apply CSS on html content :***

* Inline css
* Internal css
* External css

# **Inline CSS**

An inline style may be used to apply a unique style for a single element. To use inline styles, add the **style** attribute to the particular element. The style attribute can contain any CSS property.

**Syntax : *Style=”prop:val;prop:val”***

***Disadvantage :*** *Will invite more and more maintenance for the developer because it is not re-useable.*

# **Internal CSS**

An internal style sheet may be used to apply unique style for an html page. The internal style is defined inside the <**style**> element, inside the **head** section.

**Syntax : selector {**

**Selector is an identifier to target the html element(s)**

* **tagName**
* **.className**
* **#id**

**Prop : val,**

**Prop : val,**

**…..**

**}**

# **External CSS**

With an external style sheet, you can change the look of an entire website by changing just one file!

For use the external CSS file it uses a tag called **link** tag and this tag should be put inside the head section. CSS file extension is **.css** file

# Pixel or px in css

px. Pixels, or px , are one of the most common length units in CSS. In CSS, 1 pixel is formally defined as 1/96 of an inch.

# CSS Borders

The CSS border properties allow you to specify the style, width, and colour of an element's border.

A text on a white background

Description automatically generatedThe border-style property specifies what kind of border to display.

The border-width property specifies the width of the four borders.

The border-color property is used to set the color of the four borders.

*Note : For specify the border we need theses 3 property border-style, border-width, border-color  ( border-width, border-color optional) and border-style is mandatory but instead of specifying these 3 property we can specify it at once*

**Syntax :** border : border-width border-style border-color;

# Float in CSS

The CSS float property specifies how an element should float.

The float property can have one of the following values:

left - The element floats to the left of its container.

right - The element floats to the right of its container

none - The element does not float (will be displayed just where it occurs in the text). This is default

inherit - The element inherits the float value of its parent.

# Clear in CSS

The clear property controls the flow next to floated elements.

The clear property specifies what should happen with the element that is next to a floating element.

# What is Responsive Web design

A web design which adapt itself to the different screen sizes. You no need to code separately for the mobile and tablet.

# CSS Box Model

The **CSS box model** is a container that contains multiple properties including borders, margins, padding, and the content itself. It is used to develop the design and structure of a web page

Explanation of the different parts:

* **Content** - The content of the box, where text and images appear
* **Padding** - Clears an area around the content. The padding is transparent
* **Border** - A border that goes around the padding and content
* **Margin** - Clears an area outside the border. The margin is transparent

# Font Weight & Font Family & Font Family

The font-weight property sets how thick or thin characters in text should be displayed.

A font family is a collection of fonts that share similar design elements.

The font-style CSS property sets whether a font should be styled with a normal, italic, or oblique face from its font-family

# Opacity Property

A bridge with railings and trees

Description automatically generatedOpacity Property is used to specify of the clarity of an image.

# What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

* Style an element when a user mouses over it
* Style an element when it gets focus

# Syntax

The syntax of pseudo-classes:

selector:pseudo-class {  
  property: value;  
}

# What are the styling order for different selector

Tag Based Style🡪Class Based Style🡪Id Based Style.

Id based style always takes higher priority than the Class based style. Class based style always takes higher priority than the Tag based style

# Border Collapse

The border-collapse property sets whether **table** borders should collapse into a single border or be separated as in standard HTML. Only with the **table** tag we can add border-collapse property

**CSS Syntax**

border-collapse: separate|collapse|initial|inherit;

# nth-child(odd/even)

The :**nth-child**selector allows you to select one or more elements based on their source order,

Example : here we can select the child based o odd, even.

*/\* Select the first list item \*/*

li:nth-child(1) { }

*/\* Select the 5th list item \*/*

li:nth-child(5) { }

*/\* Select every other list item starting with first \*/*

li:nth-child(odd) { }

*/\* Select every other list item starting with first \*/*

li:nth-child(even) { }

# What is Java Script

**JavaScript** is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language. It is also known as the scripting language for webpages.  JavaScript can be used for [Client-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments as well as [Server-side](https://www.geeksforgeeks.org/server-side-client-side-programming/) developments.

# What JavaScript Do

JS can show the output to the user

It can validate user input

It is used to code processing logic

We can change the CSS at runtime

Apply/make dynamic changes to our app

# What are the output functions in JavaScript

JavaScript can "display" data in different ways:

* **The innerText property sets or returns the text of an element :innerText**.
* **Write some text directly to the HTML output:** **document.write().**
* **Writing into an alert box, using** **window.alert().**
* **Writing into the browser console, using** **console.log().**

# How to apply JavaScript

**Inline JavaScript –** When the JS code is applied at html element level using **event=”code”**. This is not reusable, hence causing heavy maintenance.

**Internal JavaScript –**The JS code is written at page level inside a **<script>** tag. This **script** tag will be inside the **head** tag. of We will create a function to modularize and reuse the code.

**External JavaScript –** The JS code is referred from an external JS file which is marked with extension **.JS** We use **<script src = “reference”>**

# Funda Fox

JavaScript is a case-sensitive programming language **e.g. I or i** both are different variables.

Every statement in JS code ends with semi-colon (;)

All browser directly understand JS so we can try the code in the console directly.

**JS has 2 important system variables which offered.**

**Document –** It is a object of the DOM, using this we can manipulate and get access to the HTML elements of our page. It contains APIs (function) like

**document.getElementById,**

**document.getElementByClassName,**

**document.getElementByTagName.**

**Window** – It is a object of the browser window.  It represents the browser's window.

In JavaScript we follow the camel case naming convention for variables, functions etc. First letter of every word is small and next consecutive words first letter is capital e.g. 🡪 **iLoveIndia, weLearnJavaScript.**

In JavaScript we can declare variables using let and var keywords

The JavaScript always written w.r.t events of an html element.

In JavaScript for a variable we no need to declare the data-type, it automatically detects the data type based on the value.

In JavaScript **=** is assignment operator and **== & ===** used to compare

**==** is used to just compare value

**===** is used to compare value and data type

# Difference Between var and let

|  |  |  |
| --- | --- | --- |
| **SN** | **var** | **let** |
| 1. | It has global scope. | It is limited to block scope. |
| 2. | It can be declared globally and can be accessed globally. | It can be declared globally but cannot be accessed globally. |
| 4. | Variable declared with var keyword can be re-declared and updated in the same scope. **Example:**  function varGreeter(){  var a = 10;  var a = 20; //a is replaced  console.log(a);  }  varGreeter(); | Variable declared with let keyword can be updated but not re-declared. **Example:**  function varGreeter(){  let a = 10;  let a = 20; //SyntaxError:  //Identifier 'a' has already been declared  console.log(a);  }  varGreeter(); |

# Types of variable in JS

|  |  |  |
| --- | --- | --- |
| Scalar Variables | Array | Objects |
| Hold single value. The value will get overwritten if we assign another value  var variableName = value  Or  let variableName = value | **Arrays are used to store multiple values of same Datatype.**  **They are declare using []**  **The element are accessed using the index of the element.**  **The index of the first element starts from 0**  **Var arrFruit = [“Apple”, ”Banana”]** | **The Objects are used to store multiple values of different Datatype in structure format.**  **They are created using curly braces. It contains key value pairs.**  **{ “key” : value,**  **“key” : value,**  **“key” : value,**  **………..**  **}** |

# JavaScript Practice

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generated

# Introduction to JavaScript anonymous functions

An anonymous function is a function without a name.

**The following shows how to define an anonymous function:**

**(function () {**

**//...**

**});**  Note that if you don’t place the anonymous function inside the (), you’ll get a syntax error.

# The For in Loop

The JavaScript for in statement loops through the properties of an Object:

### **Syntax**

for (key in object) {  
  // *code block to be executed*  
}

A screenshot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

# JavaScript Events

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event.

# JavaScript Event Handlers

Event handlers can be used to handle and verify user input, user actions, and browser actions: Things that should be done every time a page loads. Things that should be done when the page is closed. Action that should be performed when a user clicks a button.

# Function in JavaScript

A JavaScript function is a block of code designed to perform a particular task.

**Anonymous Function**

An anonymous function is a function without a name. The following shows how to define an anonymous function:

**function(p1, p2) {**

**console.log('Anonymous function');**

**};**

**Named Function**

An named function is a function, where we give the function name.

**function name(parameter1, parameter2, parameter3) {**

**// code to be executed**

**}**

# Debugger Shortcut

We can debug **js** code directly in the browser by clicking F12 (which is browser developer toolbar). Here we can see the function under the sources tab.

The debugger is set by clicking on the line or using debugger statement.

Once the debugger is hit the following shortcuts can be used.

**ABAP JS**

**Step Over F6 F10**

**Step in F5 F11**

**Step Out F7 Shift + F11**

**Continue F8 F8**

# Validate User Input using JavaScript

When User gives username and password and we validate the credentials using JavaScript if successful then will redirect to home page otherwise will throw error. Using JavaScript we do this validation.

# What Is Method Chaining?

Method chaining, or simply chaining, in JavaScript can be defined as when one or more sequential methods get invoked from an object without the introduction of unnecessary variables. The sole purpose of chaining is to make our code more readable and reduce the redundancy within.

# Asynchronous Non-Blocking IO

A screen shot of a diagram

Description automatically generated

When in the application user try to save data, then until save the data in the backend, user need to wait. Basically users are locked, they can not perform any action on UI, until the processing getting completed. This is call Synchronous programming. But it is wasting time to user, because user can not perform action.

On the other hand, in the Asynchronous programming user no need to wait until the process complete. User locked for very short time almost we can say non-blocking programming. In this case one as user is un-blocked, once the process of saving the data is finished then user get notified. This JavaScript allow to developer to send the notification to the end-user once process is finished this thing can be implemented with the help of **callback function / promise** technique.