

tmssoftware.com

Moderne Web-Entwicklung mit TMS WEB Core for Visual Studio Code

develop • faster



 facebook.com/tmssoftware
 [@tmssoftwarenews](https://twitter.com/tmssoftwarenews)
 youtube.com/tmssoftwareTV

Overview

Part 1: Classification, Introduction, Setup, and the first App with TMS WEB Core for Visual Studio Code

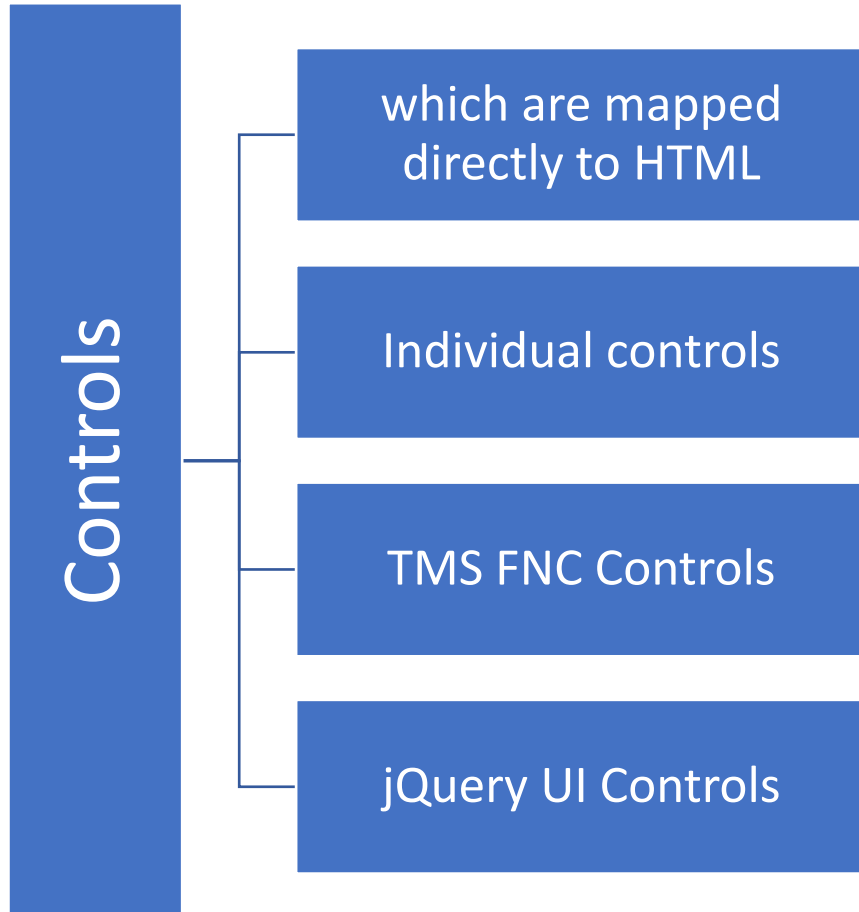
Part 2: Application Development and Features of TMS WEB Core for Visual Studio Code

- User Interface Components
- Form concept for the design of websites
- Design of forms
- Aligning, Grouping, Anchoring
- Interaction between components

Part 3: Web, Mobile, and Desktop Applications with TMS WEB Core for Visual Studio Code

Note: This set of slides is intended for users of the Visual Studio Code cross-platform editor on Windows, macOS, or Linux. Alternatively, if you would like to use the integrated development environment Delphi, download the corresponding set of slides.

User Interface Components



Employee ID	Name	Joined	Status	Progress	Target
001DK	David Kleermakers	18/03/2008	Office	<div style="width: 100%;"></div>	€ 3219,00
002LS	Laura Schneider	17/06/2011	Flight	<div style="width: 75%;"></div>	€ -6692,00
003IT	Isaac Tailor	11/08/2009	On Leave	<div style="width: 50%;"></div>	€ -8271,00
004LS	Leon Sastre	2/01/1996	Flight	<div style="width: 100%;"></div>	€ 10652,00
005MT	Marie Tailleur	16/07/2014	Flight	<div style="width: 100%;"></div>	€ 7891,00
006VS	Valerie Skredder	7/01/2015	Office	<div style="width: 75%;"></div>	€ -3137,00
007CA	Carol Alfaiate	11/08/1998	On Leave	<div style="width: 50%;"></div>	€ -9080,00
008KT	Kristoff Terzi	19/12/2004	Abroad	<div style="width: 50%;"></div>	€ -9685,00
009RS	Arthur Skraddare	24/04/2007	Abroad	<div style="width: 25%;"></div>	€ -3184,00

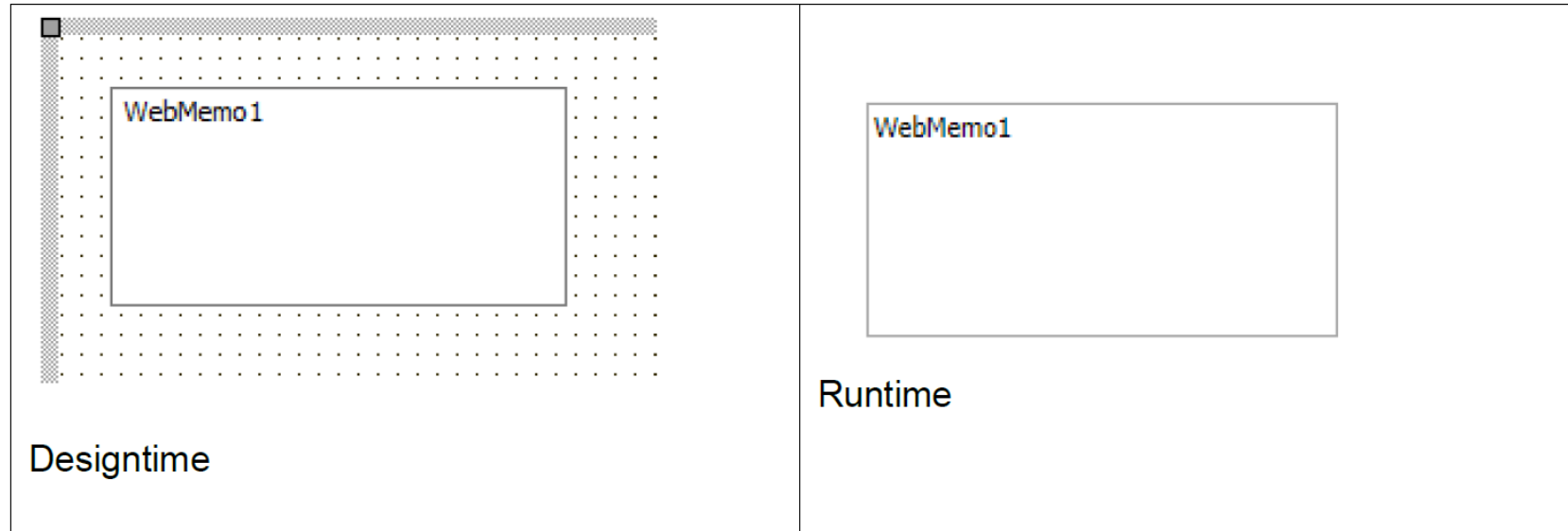
Controls that are mapped directly to HTML

- the visual component consists of one or more HTML elements (mapping the control in TMS WEB Core for Visual Studio Code to the native HTML element)
- Examples:

TMS WEB Core Control	HTML-Element
TWebDateTimePicker	<INPUT TYPE="DATE">
TWebButton	<BUTTON>
TWebListBox	<SELECT>
TWebCheckBox	<INPUT TYPE="CHECKBOX">
TWebMemo	<TEXTAREA>

- the component's properties and events then determine the attributes of the HTML element

Example: TWebMemo



Property	Meaning
AutoSize	if true, the control will automatically resize to match the text in the memo.
BorderStyle	Defines the style of the border boundary
Lines	Access the contents of the memo via a TStringList property
...	...

Custom Controls

- If there are no suitable HTML elements, the components can also be drawn on a canvas like in the VCL (Visual Component Library).
- these components are drawn independently by the TMS framework
- they derive from the *TCustomControl* class
- the class provides an object instance on *TCanvas*
- an override of the *Paint()* method can be used to paint the control
- for UI interaction, the *KeyPress/ KeyDown/ KeyUp/ MouseDown/ MouseMove/ MouseUp* methods are provided for overriding
- the canvas element can be drawn using the usual drawing methods, for example *MoveTo(), LineTo(), Rectangle()*

TMS FNC-Controls

- TMS FNC controls can also be used in TMS WEB Core
- TMS FNC controls can be used in several frameworks (framework neutral), these are:
 - VCL (Windows)
 - FireMonkey (Windows, macOS, Linux, iOS, Android)
 - Lazarus LCL Framework (Windows, macOS, Linux, iOS, Android)
 - TMS Web Core (Web)

Compatibility

Frameworks



Operating systems/browsers










IDEs



More Information:

<https://www.tmssoftware.com/site/fnc-products.asp>

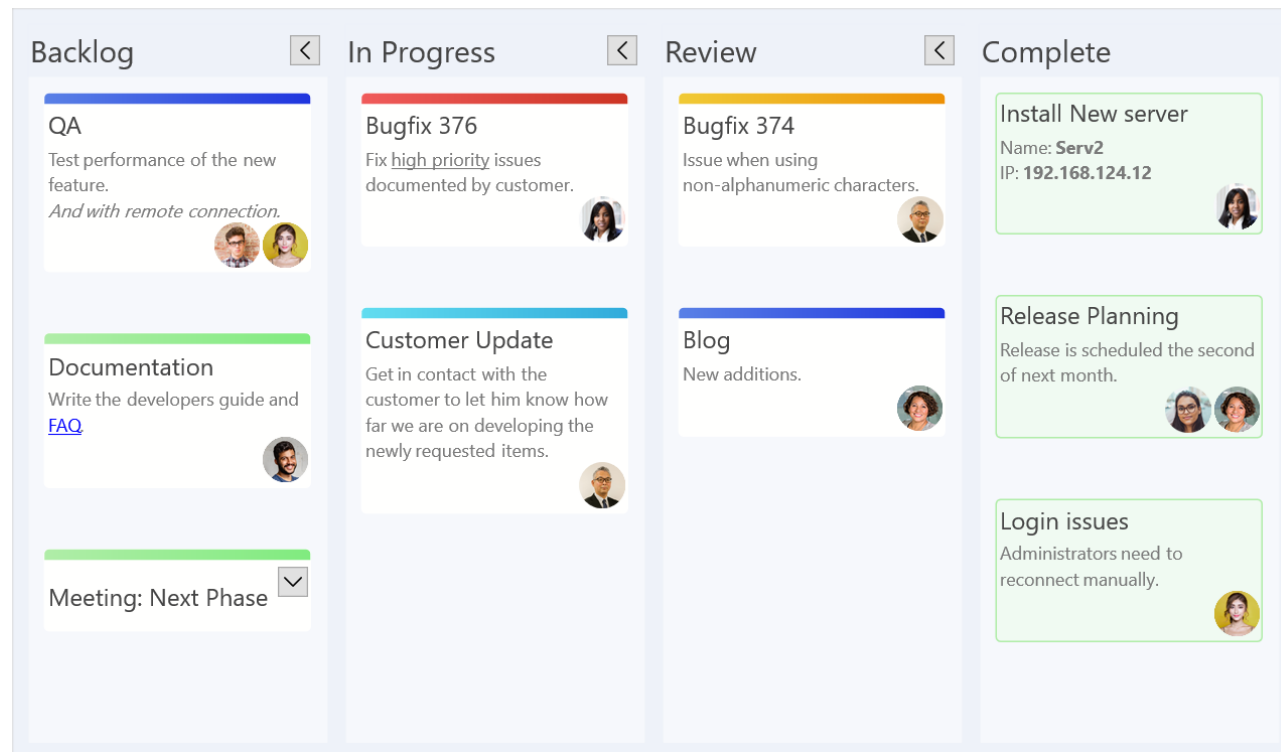
TMS FNC-Controls: Example

Employee ID	Name	Joined	Status	Progress	Target
001DK	 David Kleermakers	18/03/2008	Office	<div style="width: 50%;"></div>	€ 3219,00
002LS	 Laura Schneider	17/06/2011	Flight	<div style="width: 75%;"></div>	€ -6692,00
003IT	 Isaac Tailor	11/08/2009	On Leave	<div style="width: 20%;"></div>	€ -8271,00
004LS	 Leon Sastre	2/01/1996	Flight	<div style="width: 90%;"></div>	€ 10652,00
005MT	 Marie Tailleur	16/07/2014	Flight	<div style="width: 85%;"></div>	€ 7891,00
006VS	 Valerie Skredder	7/01/2015	Office	<div style="width: 60%;"></div>	€ -3137,00
007CA	 Carol Alfaiate	11/08/1998	On Leave	<div style="width: 40%;"></div>	€ -9080,00
008KT	 Kristoff Terzi	19/12/2004	Abroad	<div style="width: 55%;"></div>	€ -9685,00
009RS	 Arthur Skraddare	24/04/2007	Abroad	<div style="width: 15%;"></div>	€ -3184,00

DataGrid

- attractive visualization
- Data binding
- Functions: filter, sort, group
- Export to PDF, HTML, XLS formats

TMS FNC-Control: Example

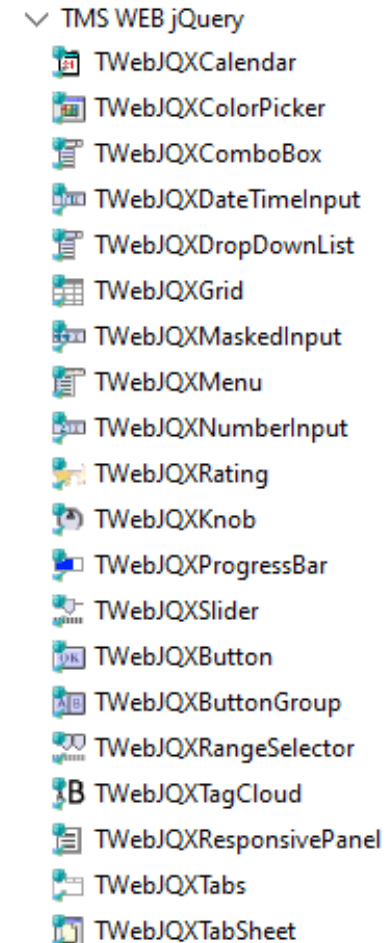


Kanban

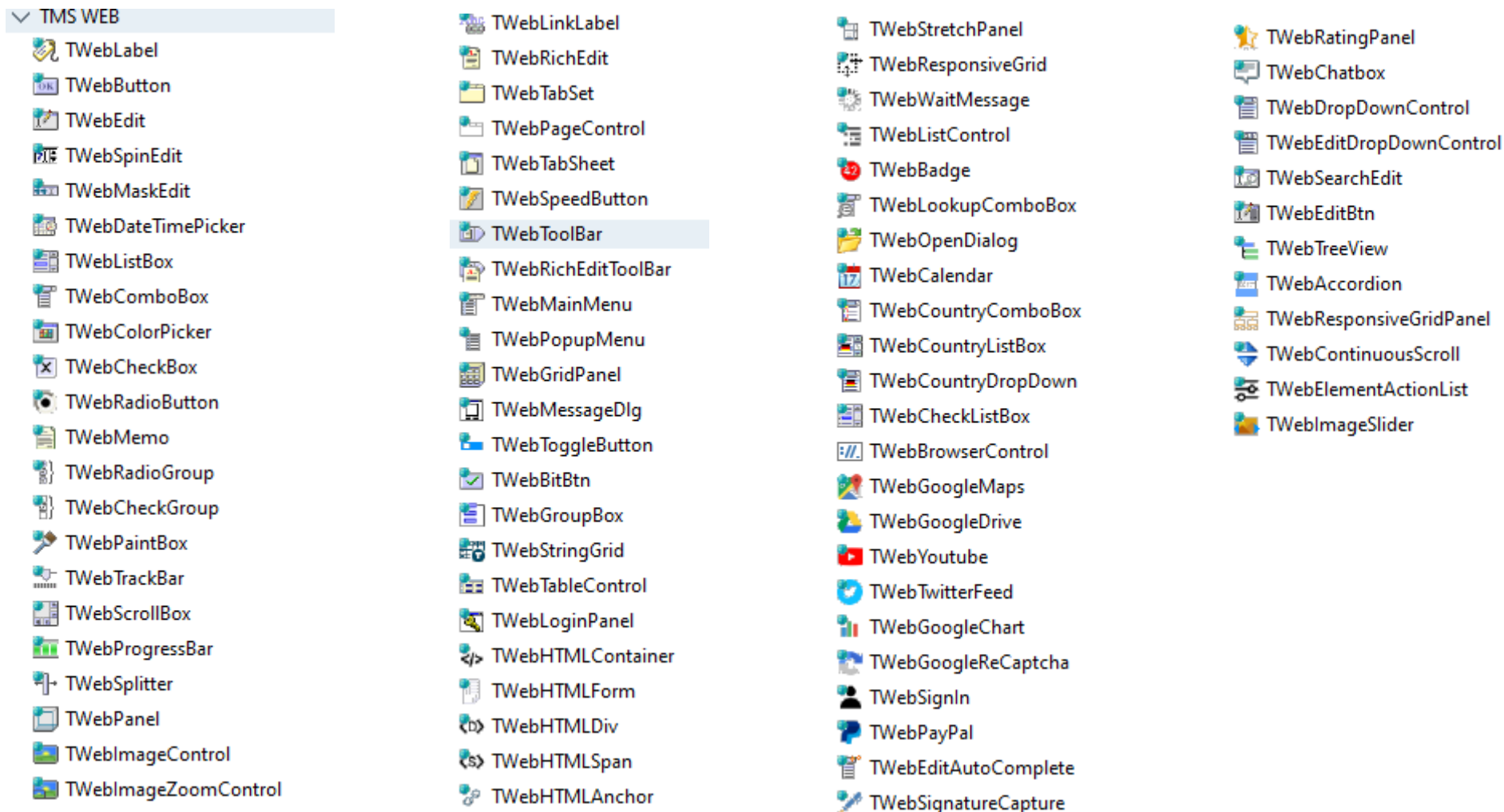
- visually appealing design
- fully configurable
- Presentation of different data in column form

jQuery UI Components

- Wrapper around the graphical components of the jQuery framework
- in the development environment, the components will be displayed as a rectangle (placeholder)
- the drawing routines are implemented in JavaScript and are also displayed in the Visual Studio Designer



Controls appear in the Components palette



Use Controls

- powerful and comprehensive components significantly reduce the effort required to design the user interface
- complex requirements can be implemented with little effort:
 - selecting the component from the palette
 - placement on the form
 - customization of layout
 - customization of properties
 - connecting to program logic (Delphi)

Example: Login Form (Draft)

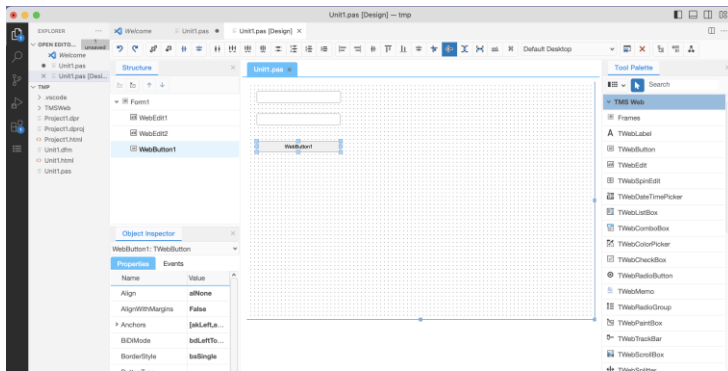
The screenshot shows the Delphi IDE in Design mode for a web application. The main canvas displays a grid with a yellow rectangular form containing a login interface. The form includes labels for 'Login', 'Username:', and 'Password:', two text input fields, and a 'Login' button. A blue circle with the number '2' is placed over the form, indicating its placement on the grid. To the right, the 'Tool Palette' is open, showing a list of web components. A blue circle with the number '1' is placed over the 'TWebLabel' component, indicating its selection. Below the main canvas, the 'Object Inspector' shows the properties of the selected 'Form1: TForm1' component. A blue circle with the number '3' is placed over the 'Color' property, which is set to 'clWhite'. At the bottom, a browser window displays the rendered login form. A blue circle with the number '4' is placed over the browser window, indicating the final step of running the application.

1. Selection from the palette.
2. Placement on the form.
3. Configuration of the properties, for example the background color.
4. Run the web app.

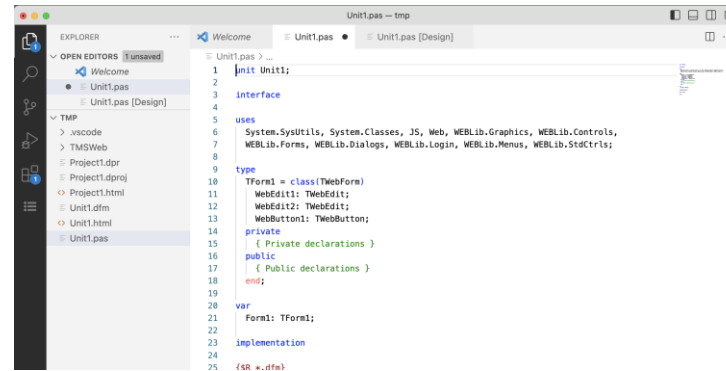
Form concept for designing websites

- in TMS WEB Core, a form is a web page or part of a web page
- Forms are derived from the *TWebForm* class
- Visual and non-visual components can be placed on a form
- a *TWebForm* represents a complete page in the web browser
- the contents of the form are inserted into the `<body>` tag of the web page (HTML)
- additional content can be inserted here using HTML if necessary

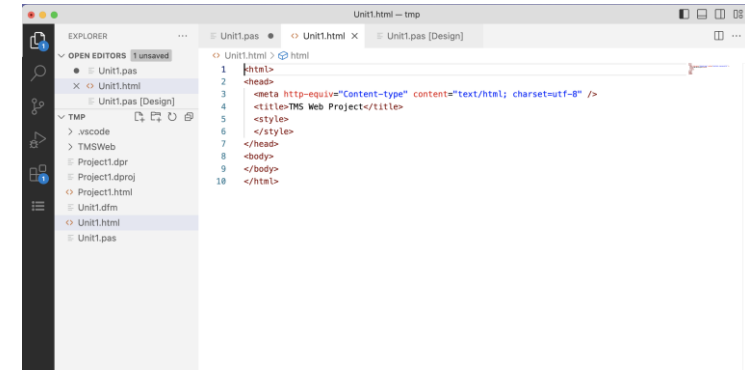
Views of a Form in Visual Studio Code



Designer



Source code



HTML

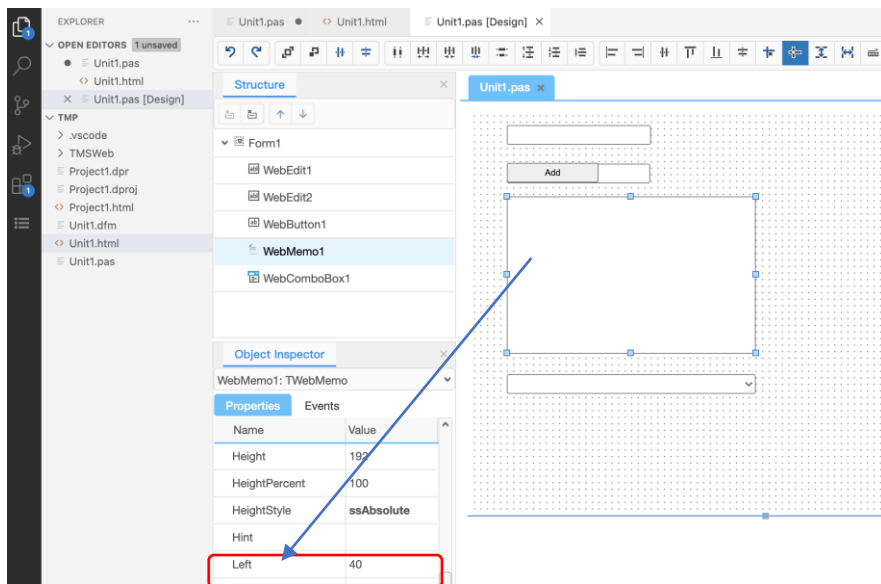
<body>-Tag is replaced by elements of the form

Design of formulars

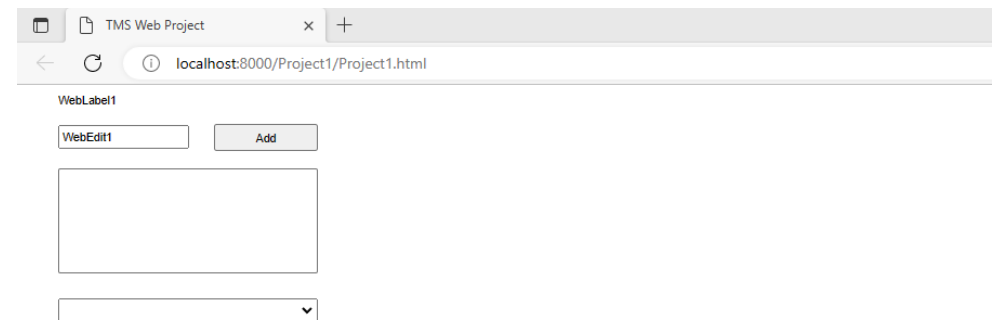
- in web applications, forms (web pages) are designed using HTML (structure) and CSS (design, layout).
- with TMS WEB Core there are several approaches to design
 - exclusive use of the form designer, i.e. HTML and CSS are generated automatically
 - combined use of the form designer and customization via CSS
 - only the containers are defined via form designers and the layout and design are done by manual definition in HTML and CSS
- methods can be combined with each other
- approach for professional and large websites:
 - define containers in the designer and use them for data binding and programming of business logic
 - Definition of the layout and design in HTML and CSS by a designer

Absolute Positioning

- in form designer each component has an absolute point for positioning
- Example:



Designer



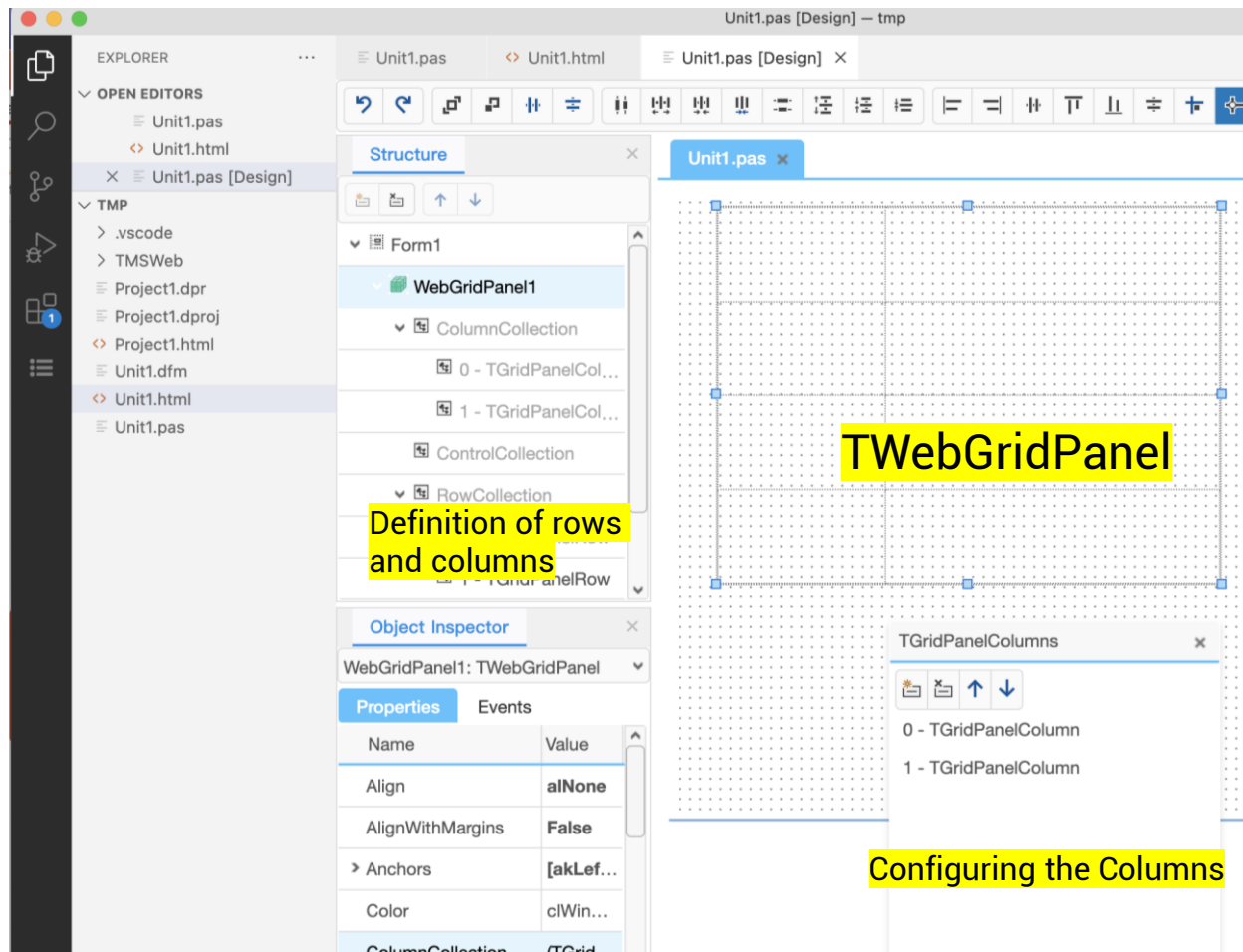
Runtime (Browser)

Layout with containers

- the typical containers such as *TWebPanel*, *TWebGroupBox*, *TWebScrollBar* and *TWebGridPanel* are available for aligning the components
- the owner and parent principle applies
- example:
 - Generate Form
 - layout container: *TWebGridPanel*
 - definition of columns and rows (relative or absolute)
 - assignment of the controls to the cells of the *TWebGridPanel*
 - customization of the properties of the controls.

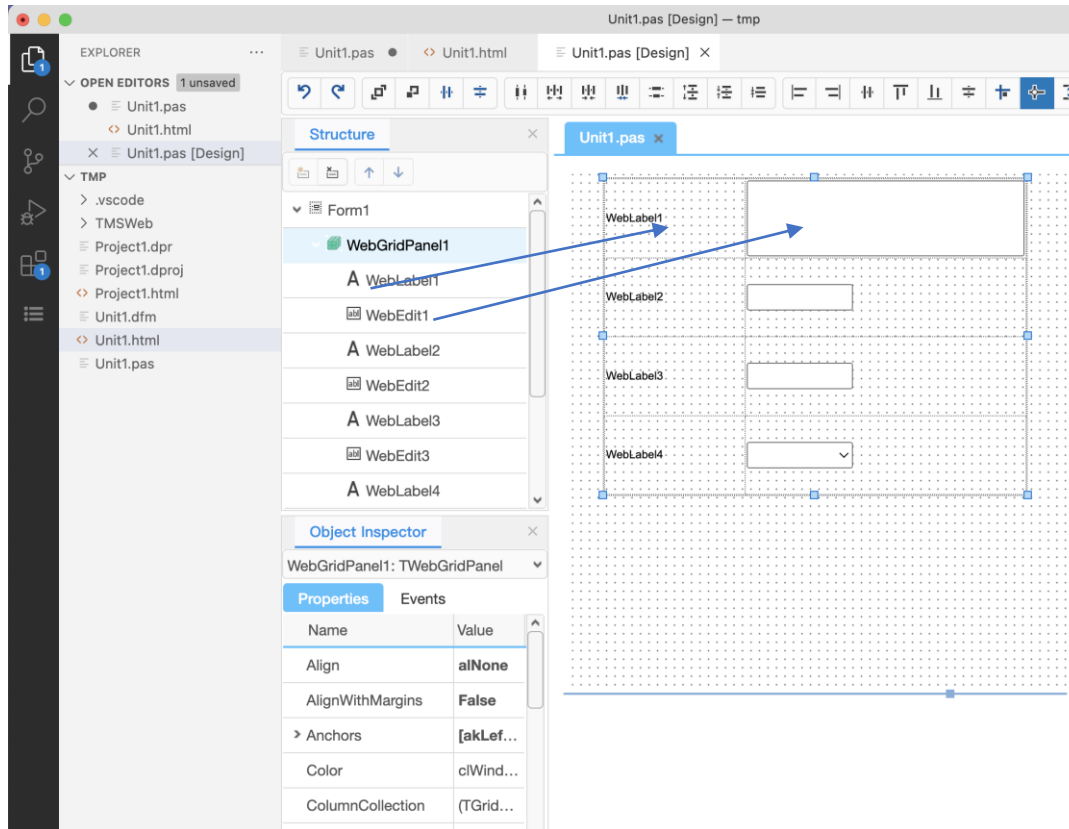
Layout with Containers: Example

Definition of the layout



- *TWebGridPanel as a layout container*
- *Definition of columns and rows*
- *Specifying Height and Width*

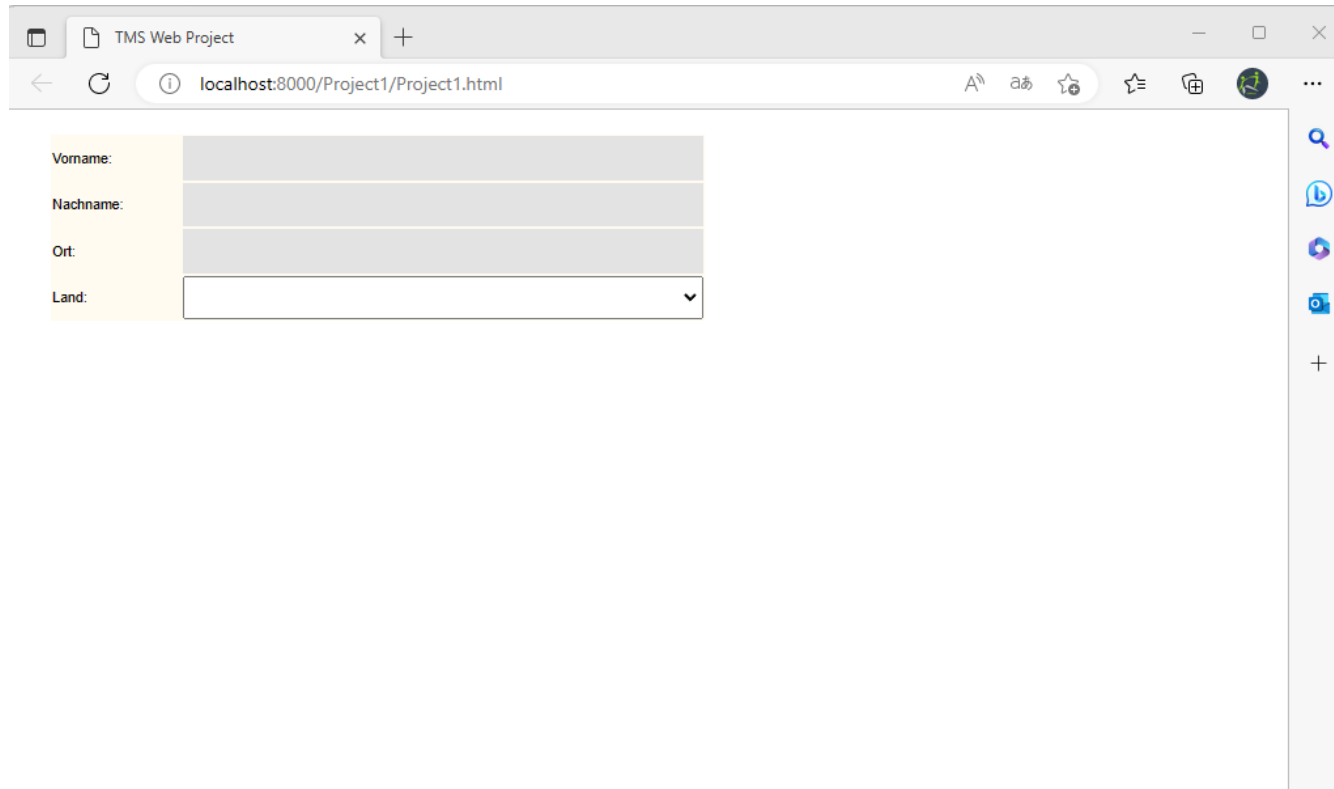
Layout with Containers: Assignment of the components



- assignment of the components to the Cells in the *WebGridPanel*
- Alignment of components in Panel, for example *Align:=clClient*
- set the properties of the controls

Layout with Containers

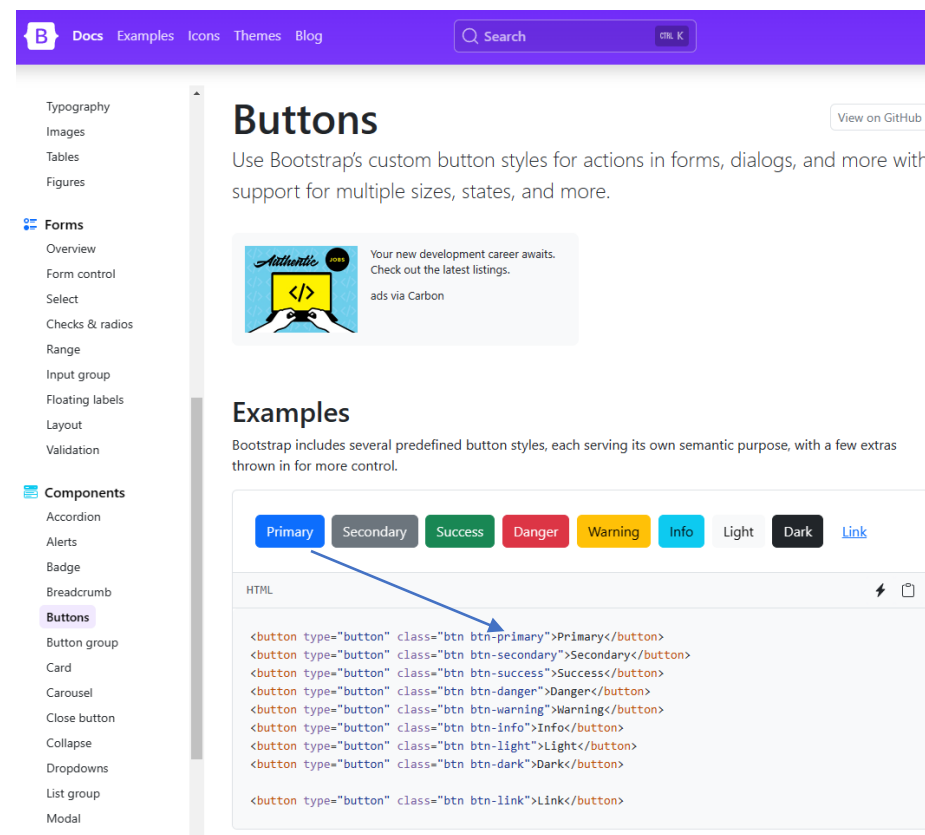
Run Time



- the layout container ensures correct placement of the controls in the browser
- this approach does not require writing HTML and CSS

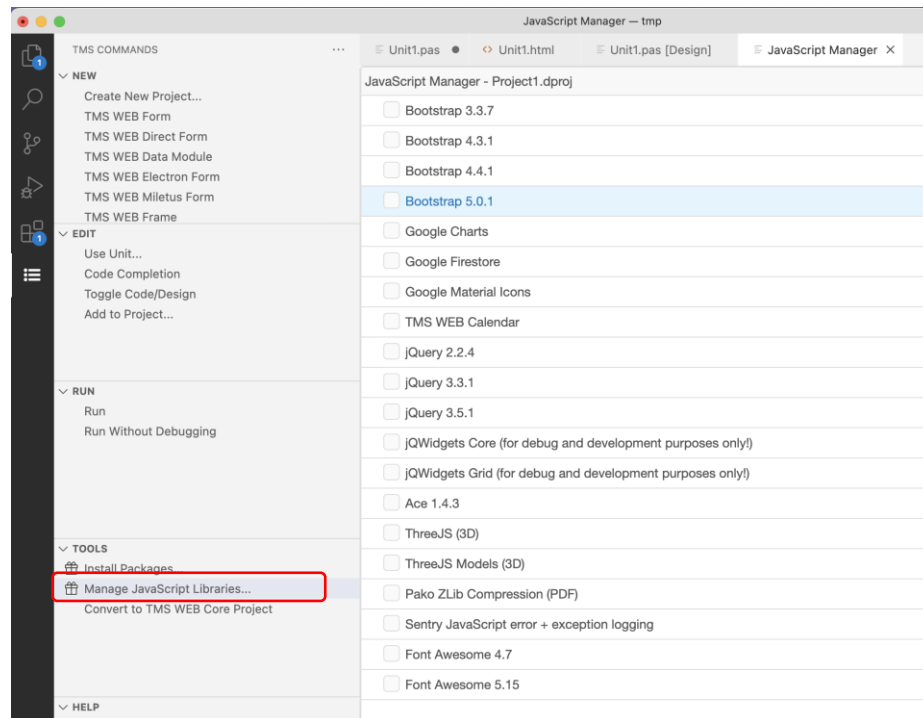
Layout with CSS and Bootstrap

- many web applications are designed with the Bootstrap library
- Bootstrap has many classes for designing the layout, defining components and designing the application
- more information: <https://getbootstrap.com/>
- in TMS WEB Core you can use this library



The screenshot shows the Bootstrap documentation page for Buttons. The page has a purple header with navigation links: Docs, Examples, Icons, Themes, Blog, and a search bar. A sidebar on the left lists various components like Typography, Images, Tables, Figures, Forms, and Components. The main content area is titled "Buttons" and includes a "View on GitHub" link. Below the title is a paragraph explaining that Bootstrap's custom button styles are used for actions in forms, dialogs, and more, with support for multiple sizes, states, and more. There is an example image showing a button with a code icon and text: "Your new development career awaits. Check out the latest listings. ads via Carbon". Below this is an "Examples" section with a row of buttons: Primary (blue), Secondary (grey), Success (green), Danger (red), Warning (yellow), Info (cyan), Light (light grey), Dark (dark grey), and Link (blue). Below the buttons is a code editor showing the HTML for each button style, with a blue arrow pointing from the "Primary" button to the first line of code: `<button type="button" class="btn btn-primary">Primary</button>`.

Layout with CSS and Bootstrap Example



- the bootstrap library needs to be added to the project
- this can be done directly in the Delphi development environment
- for example, search for Bootstrap and add the latest version to the project
- other libraries can also be added to the project in this way, for example jQuery
- alternatively, it would be possible to edit the HTML files manually
- the order in which the libraries are added is crucial; this determines which library is loaded first at runtime
- if the jQuery library is to be added, then this must be the first library

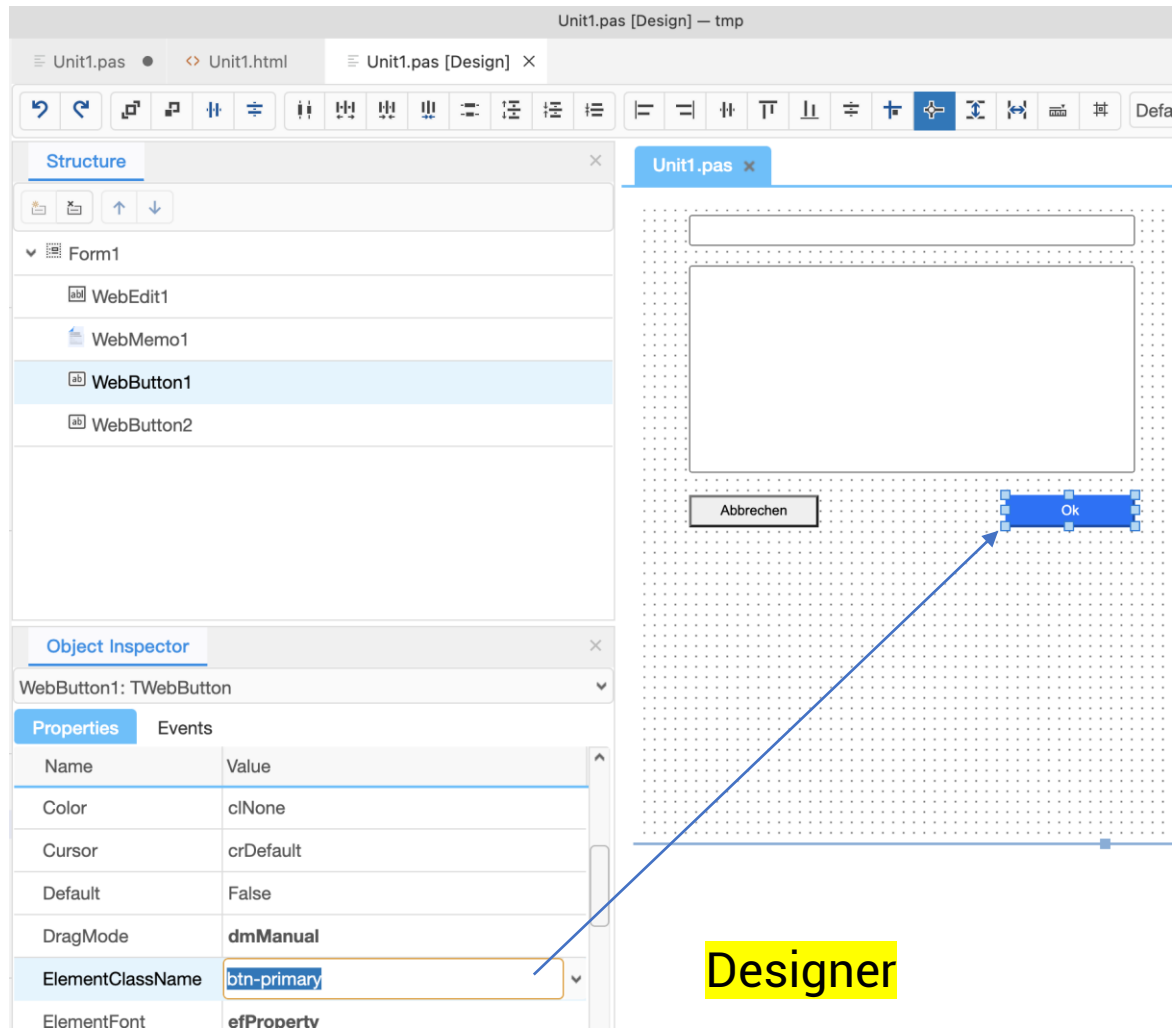
Layout with CSS and Bootstrap Example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta content="width=device-width, initial-scale=1" name="viewport"/>
    <meta $(ThemeColor)/>
    <noscript>Your browser does not support JavaScript!</noscript>
    <link href="data:;base64,=" rel="icon"/>
    <meta $(Manifest)/>
    <title>TMS Web Project</title>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js" type="text/javascript"></script>
    <link crossorigin="anonymous" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet"/>
    <script src="$(ProjectName).js" type="text/javascript"></script>
    <style>
    </style>
  </head>
  <body>
<meta $(BodyParameters)/>
    </body>
    <script type="text/javascript">rtl.run();</script>
</html>
```

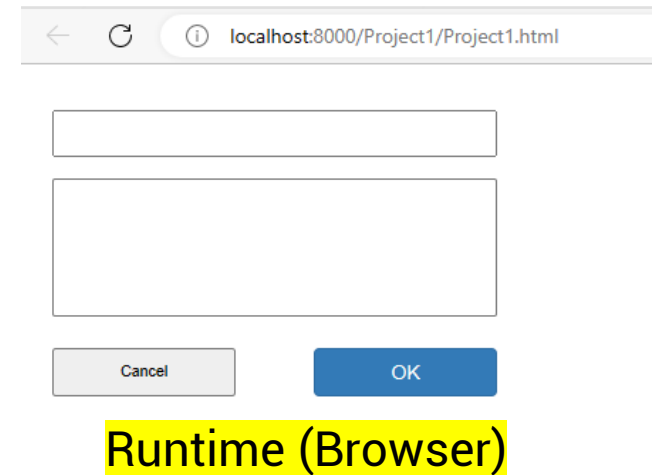
- the references to the library(s) are inserted into the file *Project1.html*
- the Bootstrap library classes are then available for use in Delphi's Object Inspector

Layout with CSS and Bootstrap

Example



- Designing the form in the designer
- assigning Bootstrap classes to controls in object Inspector
- in the designer, the styling is displayed directly



Layout with CSS and Bootstrap

Example 2

The screenshot shows the Delphi IDE Designer for a form named 'Form1'. The design view displays a form with a 'Benutzername:' label, a text input field, a 'Passwort:' label, a password input field, and an 'Ok' button. The Structure panel on the left lists the components: Form1, WebEdit1, WebButton1, WebLabel1, WebLabel2, and WebEdit2. The Object Inspector for 'Form1: TForm1' shows the 'CSSLibrary' property set to 'cssBootstrap', which is highlighted with a red box.

Designer

This screenshot shows the Object Inspector for a 'WebButton1' component. The 'ElementClassName' property is set to 'btn-success', which is highlighted with a red box. A blue arrow points from this property to the 'Ok' button in the design view.

This screenshot shows the runtime view of the form in a browser. The browser address bar shows 'localhost:8000/Project1/Project1.html'. The form displays the 'Benutzername:' and 'Passwort:' labels, input fields, and an 'OK' button. The Structure panel on the left shows the component hierarchy, and the Object Inspector for 'WebEdit1: TWebEdit' shows the 'ElementClassName' property set to 'form-text', which is highlighted with a red box. A blue arrow points from this property to the text input field in the browser view.

Runtime (Browser)

Align, group, anchor

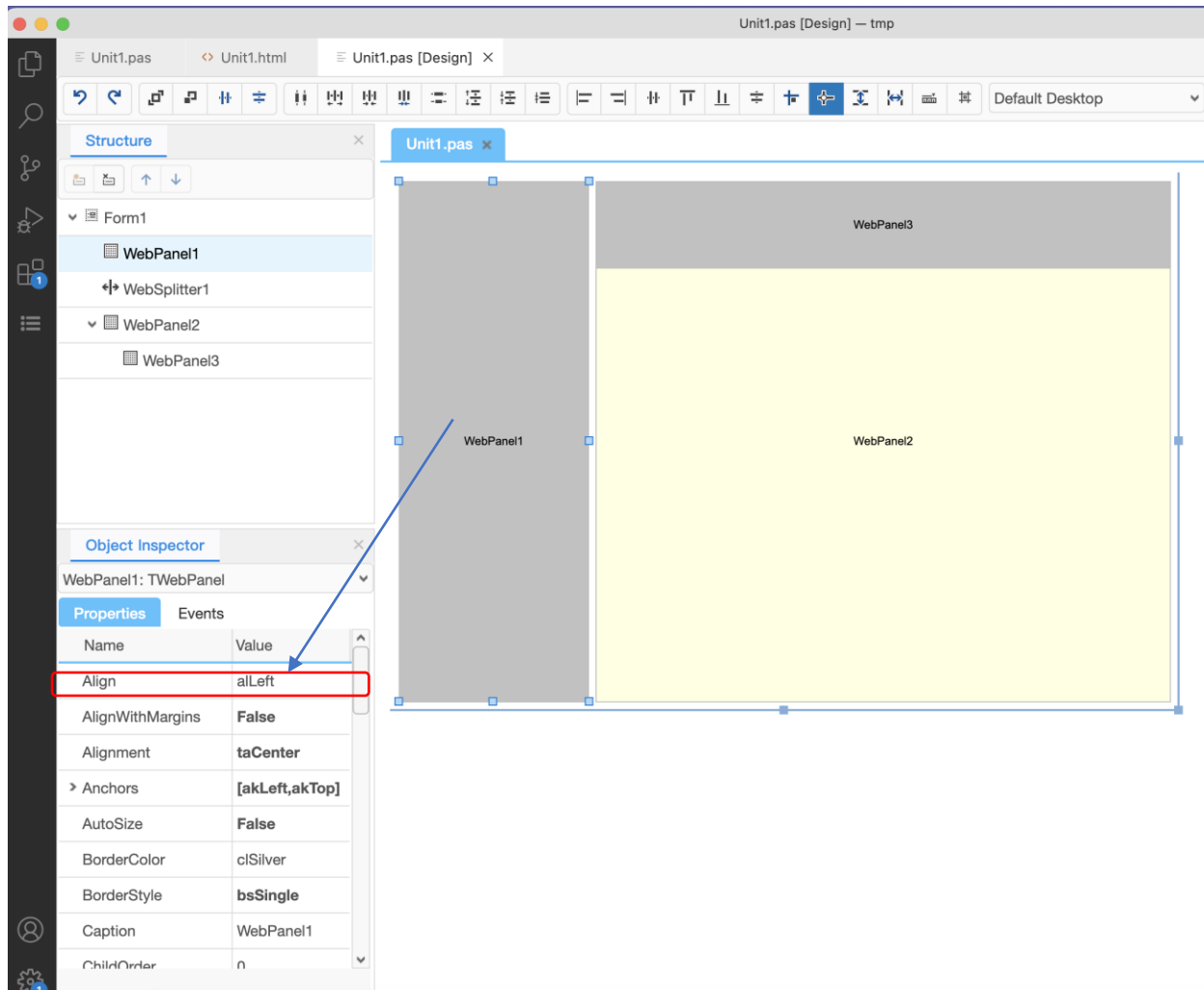
- the components for the user interface can be aligned, anchored and grouped directly in Delphi
- this makes basic responsive design possible directly via the designer without having to explicitly write HTML and CSS
- overview
 - Align: *Align* property
 - Grouping: using panel components
 - Anchor: *Anchor* property

Align

- controls can be aligned on the form
- big advantage over most web development frameworks
- in combination with the panel components, a powerful tool for designing the surface is created
- the procedure is identical to the user interface frameworks for the desktop, i.e. VCL and FireMonkey
- almost the completely layout of the app can be designed this way
- ideal for designing the layout of business applications

Aligning the controls

Example – Designer

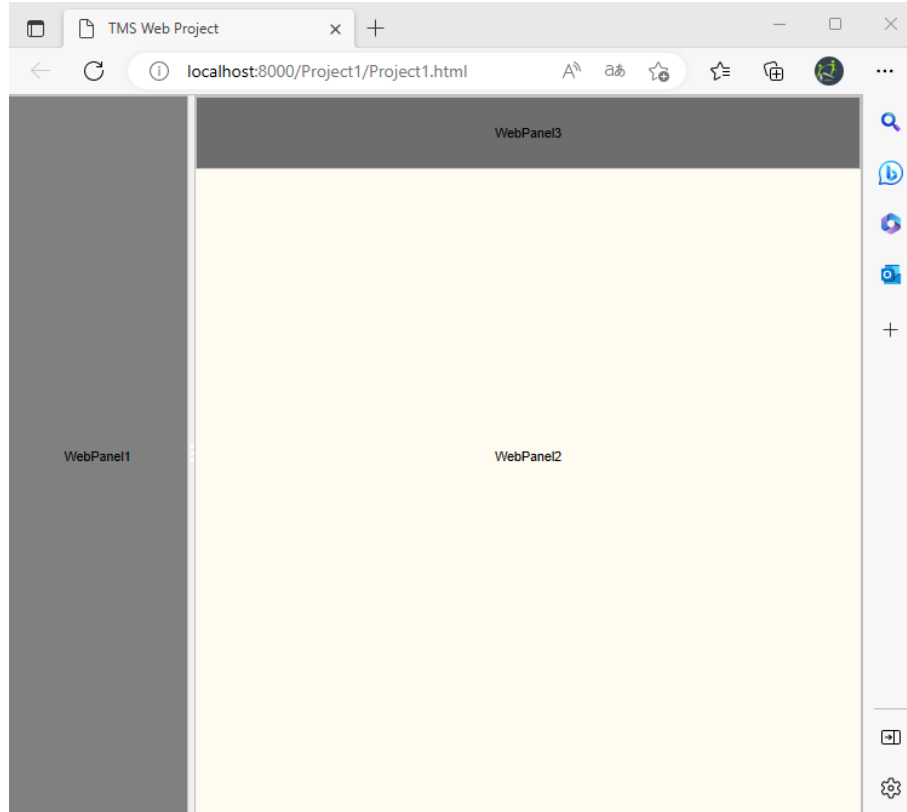


Control	Align
WebPanel (left)	alLeft
WebPanel (top)	alTop
WebSplitter	alLeft
WebPanel (Content)	alClient

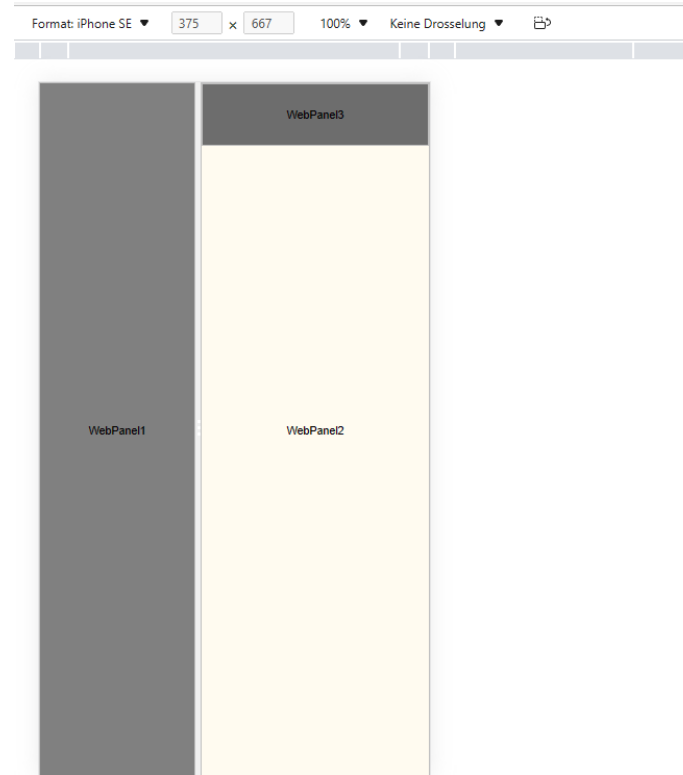
- design the layout completely in the designer through alignment
- easily usable for business web applications with a similar interface

Aligning the Controls

Example – Runtime



Desktop



Mobile

- fully responsive layout
- panel containers can accommodate additional controls

Grouping the controls

- grouping controls is done using panels
- TWebPanel:
 - accommodation of multiple components with free placement
 - a component is assigned via the property's *Owner* and *Parent*
- TWebGridPanel:
 - arrangement of control elements in rows and columns
 - during placement, the components are taken from the toolbox and placed on the grid
 - these are assigned in order, i.e. from left to right and from top to bottom
 - cell size can be specified in pixels or in percent
 - the components can be nested within the cell

1	2
3	4
5	6

Grouping the Controls

Example – Designer and Runtime

The image displays a web application in two states: the design phase and the runtime phase.

Designer View: The left pane shows the 'Structure' tree with a grid component 'WebGridPanel1' containing three rows and two columns. The right pane shows the visual design of the grid on a dotted background. The first row contains a label 'Vorname:' and a text box with 'Max'. The second row contains a label 'Nachname:' and a text box with 'Mustermann'. The third row contains a label 'Ort:' and a text box with 'München'. A yellow box labeled 'Designer' is positioned below the structure tree.

Runtime View: The bottom pane shows the rendered HTML output of the grid. The browser address bar indicates the URL 'localhost:8000/Project2/Project2.html'. The rendered output is a table with three rows and two columns:

Vorname:	Max
Nachname:	Mustermann
Ort:	München

A yellow box labeled 'Runtime' is positioned below the rendered output.

- Grid with 3 rows and 2 columns
- Arrangement of Controls in the Grid
- within a cell, the control elements can be aligned, for example *withAlign:=clClient*

Aligning the controls

- in addition to the property `Align`, most graphical components of TMS WEB Core offer the option of anchoring
- this is done via the property *Anchors*

Property	Meaning
<code>akLeft</code>	anchor left
<code>akTop</code>	anchor top
<code>akRight</code>	anchor right
<code>akBottom</code>	anchor bottom

Aligning the Controls

Example - Designer

Structure

Unit1.pas

Form1

- WebLabel1
- WebLabel2
- WebEdit1
- WebEdit2
- WebMemo1

Object Inspector

WebMemo1: TWebMemo

Properties

Name	Value
Align	aNone
AlignWithMargins	False
▼ Anchors	[akLeft,akTop,a...
akLeft	True
akTop	True
akRight	True
akBottom	True

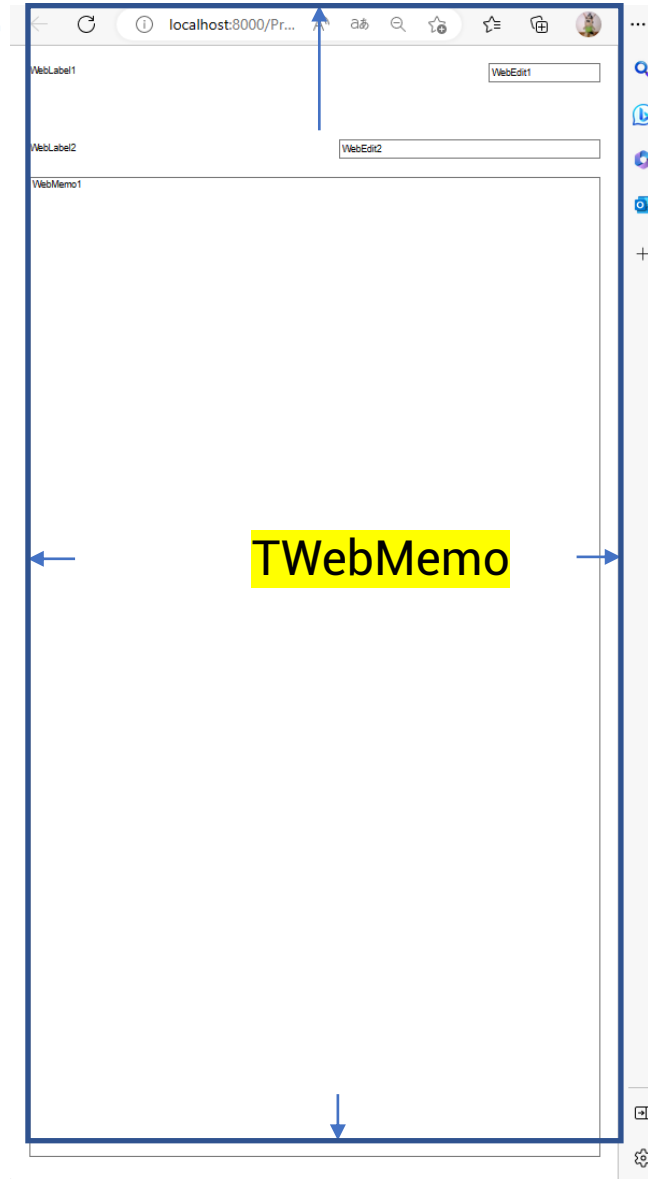
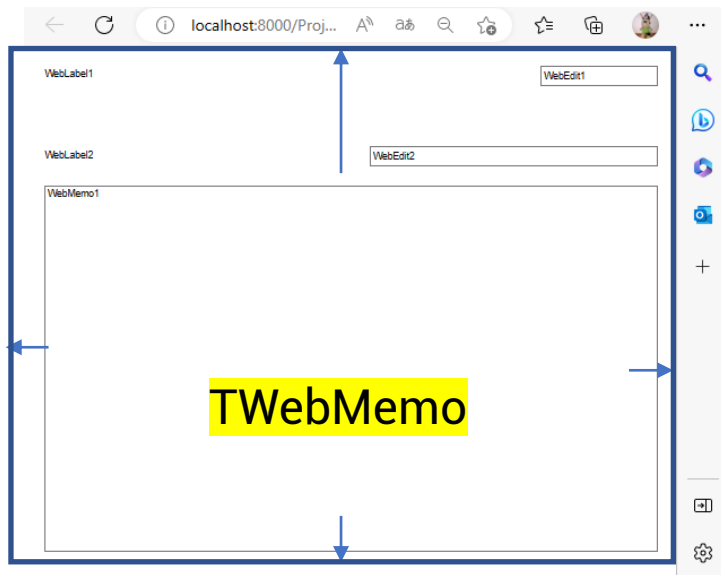
Anchor: Left

Anchor: Right

Anchor: Left, Right, Top, Bottom

Aligning the controls

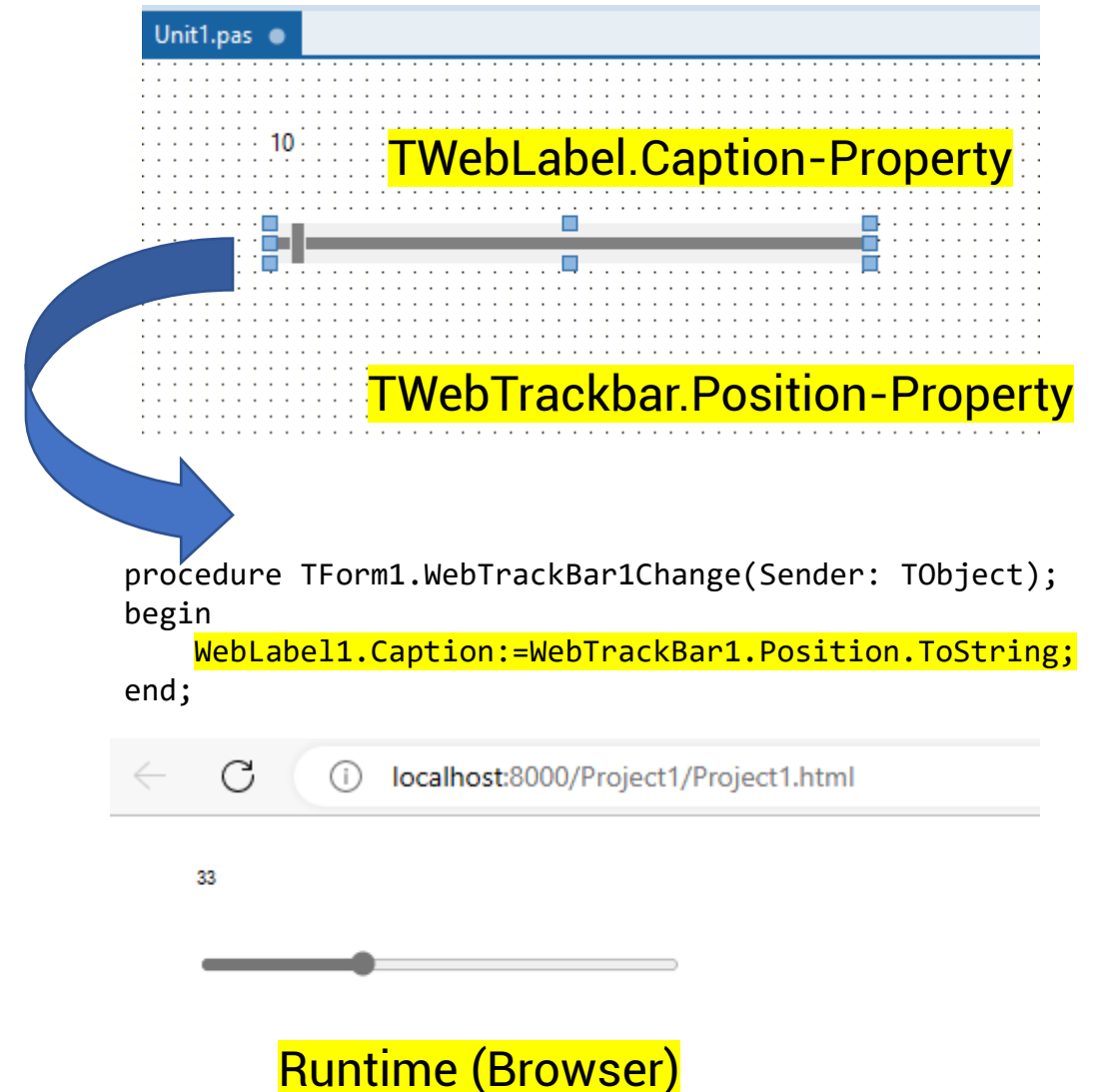
Example - Runtime



- the alignment of the components on the website (form) remains identical when the size is changed
- for example, the *TWebMemo* component is anchored on all four sides, i.e.:
Anchor:=[akLeft, akTop, akRight, akBottom]

Interaction between controls

- TMS WEB Core ensures that components are automatically updated when the values linked to the properties change
- the principle of event-oriented programming used in desktop programming is transferred to the development of web applications
- the concepts commonly used in web programming are based on manipulation of the DOM of the website in single page applications (SPA).
- TMS WEB Core takes care of these processes automatically, i.e. the translation takes place in JavaScript (no reload of the page necessary)



Overview

Part 1: Classification, introduction, setup and the first app with TMS WEB Core

Part 2: Application development and features of TMS WEB Core

Part 3: Web, Mobile and Desktop Applications with TMS WEB Core

Resources

- TMS WEB Core:

<https://www.tmssoftware.com>

- Visual Studio Code

<https://code.visualstudio.com/>

- TMS Academic License:

<https://www.tmssoftware.com/site/academic.asp>

- Flick, Holger: TMS WEB Core: Developing Web Applications with Delphi,
Independently published, 2020

tmssoftware.com

develop • web

Framework for creating modern web applications



More info:

<http://web.tmssoftware.com>