

APRIL 24TH 2024

FIORICONF

**SAP FIORI FOR ALL:
ENHANCING ACCESSIBILITY
AND SUSTAINABILITY**



OUR SPEAKERS



(he/him)

Nicolas leads a team of SAP developers at VINCI Energies. He is striving to incorporate aspects of modern development, accessibility principles, and clean code into their work, driven by a strong appetite for innovation and SAP new technologies.

Nicolas BOUCAUD

SAP Development
Team Lead



(he/him)

Nizare is a dedicated SAP Developer with expertise in Fiori/SAPUI5 development and SAP Cloud Technologies at VINCI Energies. Passionate about DevOps practices, he advocates for CI/CD implementation within the SAP ecosystem.

Nizare DAMOUMAT

SAP Software
Engineer



AGENDA

01

Introduction to IT
Accessibility

02

Accessibility in
Fiori Context

03

Fiori: Visual
impairments case

04

Fiori: Motor
impairments case

05

Measurements
and Green IT

06

Conclusion



01

INTRODUCTION TO IT ACCESSIBILITY

IT ACCESSIBILITY

DEFINITION

- Inclusive practice ensuring everyone can access web content.

PURPOSE

- To provide equal access to information and functionality for all users.

WHAT IS IT ACCESSIBILITY?

DESIGN/DEVELOPMENT

- Websites and applications are correctly designed, developed, and edited to be accessible to everyone.

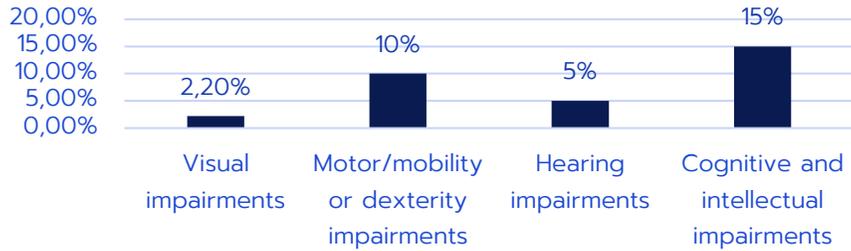
BENEFITS

- Ensures robustness to work with a variety of assistive technologies for people with impairments.

OVERVIEW OF THE ACTUAL STATE

GENERAL INDICATORS AROUND THE WORLD

% of persons with difficulty to access IT around the world



1 billion individuals worldwide!



SANCTIONS



Lawsuits



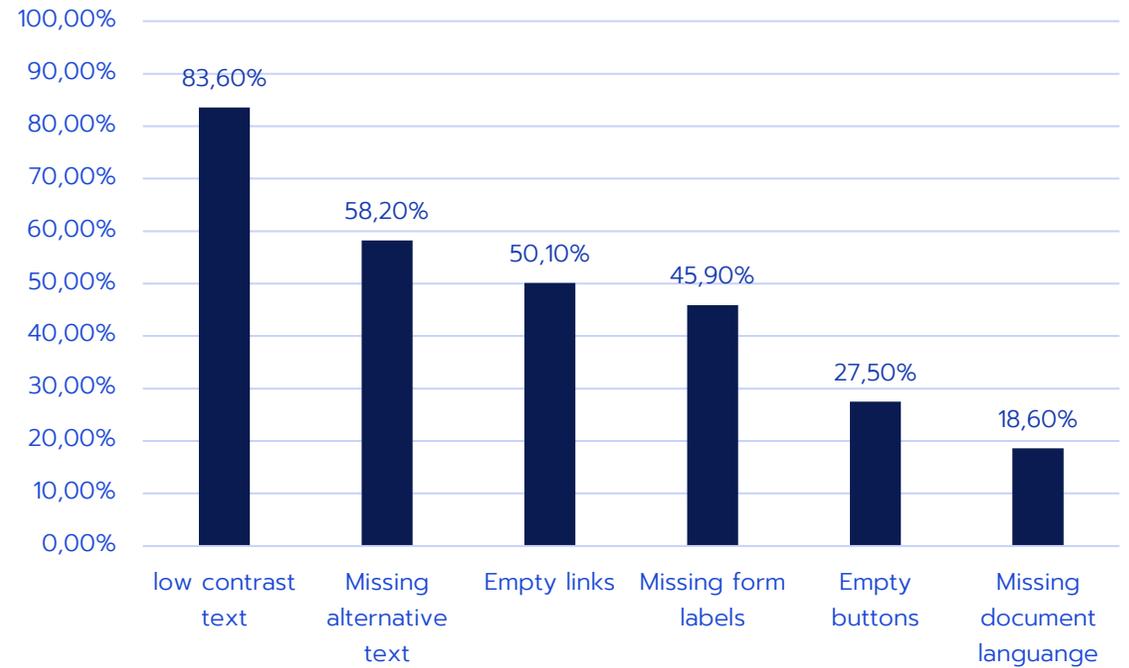
Financial liabilities



Damage to brand reputation

COMPLIANCE ON APP %

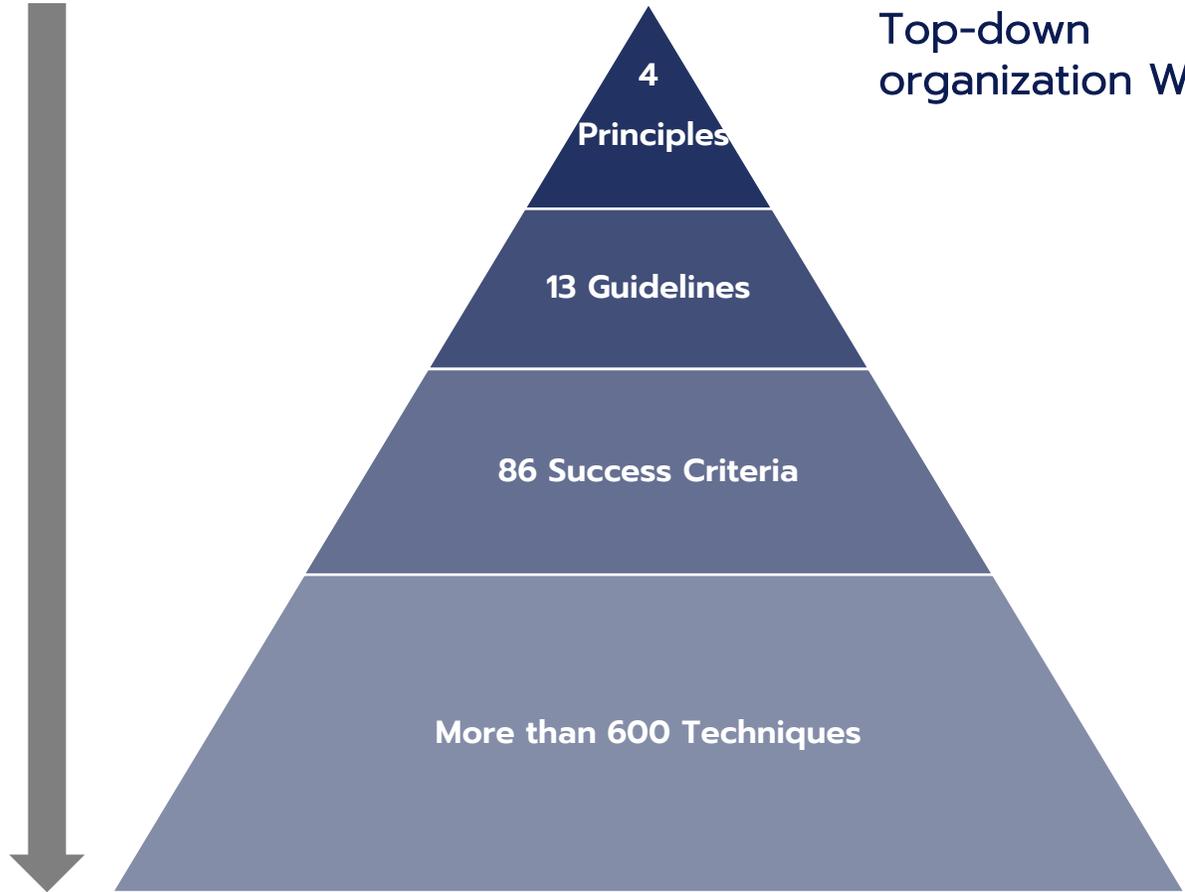
98,1% Of non-compliant sites



Source: webaim.org

INTERNATIONAL WEB ACCESSIBILITY GUIDELINES – WCAG

Top-down organization WCAG



WCAG 4 Principles



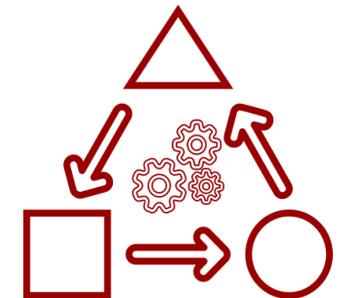
Perceivable



Operable



Understandable

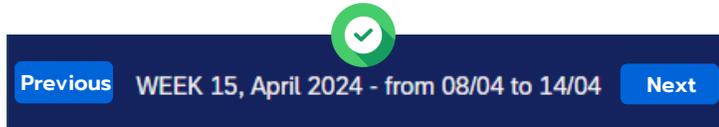


Robust

LEVELS OF COMPLIANCE WITH WCAG

LEVEL SIMPLE A

- Is the minimum level of conformance.
- Example: Guideline 1.1 Text Alternatives.



Problem

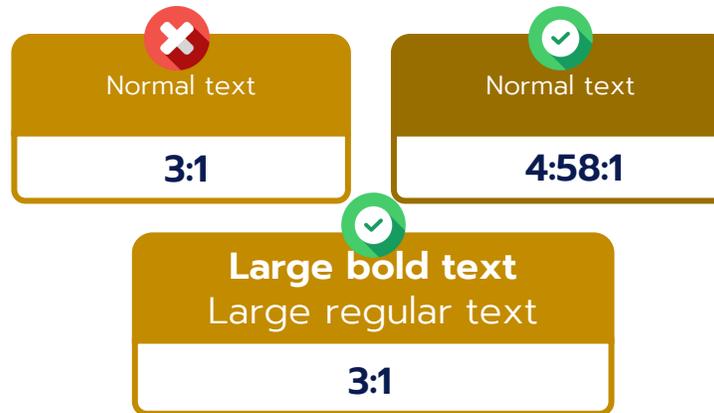
- Shapes don't help screen readers identifying the meaning.
- Some shapes can be difficult to distinguish.

Alternatives

- Words can be identified by screen readers.
- Simple words support the meaning of the shape, for example "Previous" or "CANCEL" would have different meanings.

LEVEL DOUBLE A (AA)

- Is generally the level that most sites are aiming to meet. It can often be a level that is a legal requirement.
- Example: Guideline 1.4.3 Contrast (Minimum)



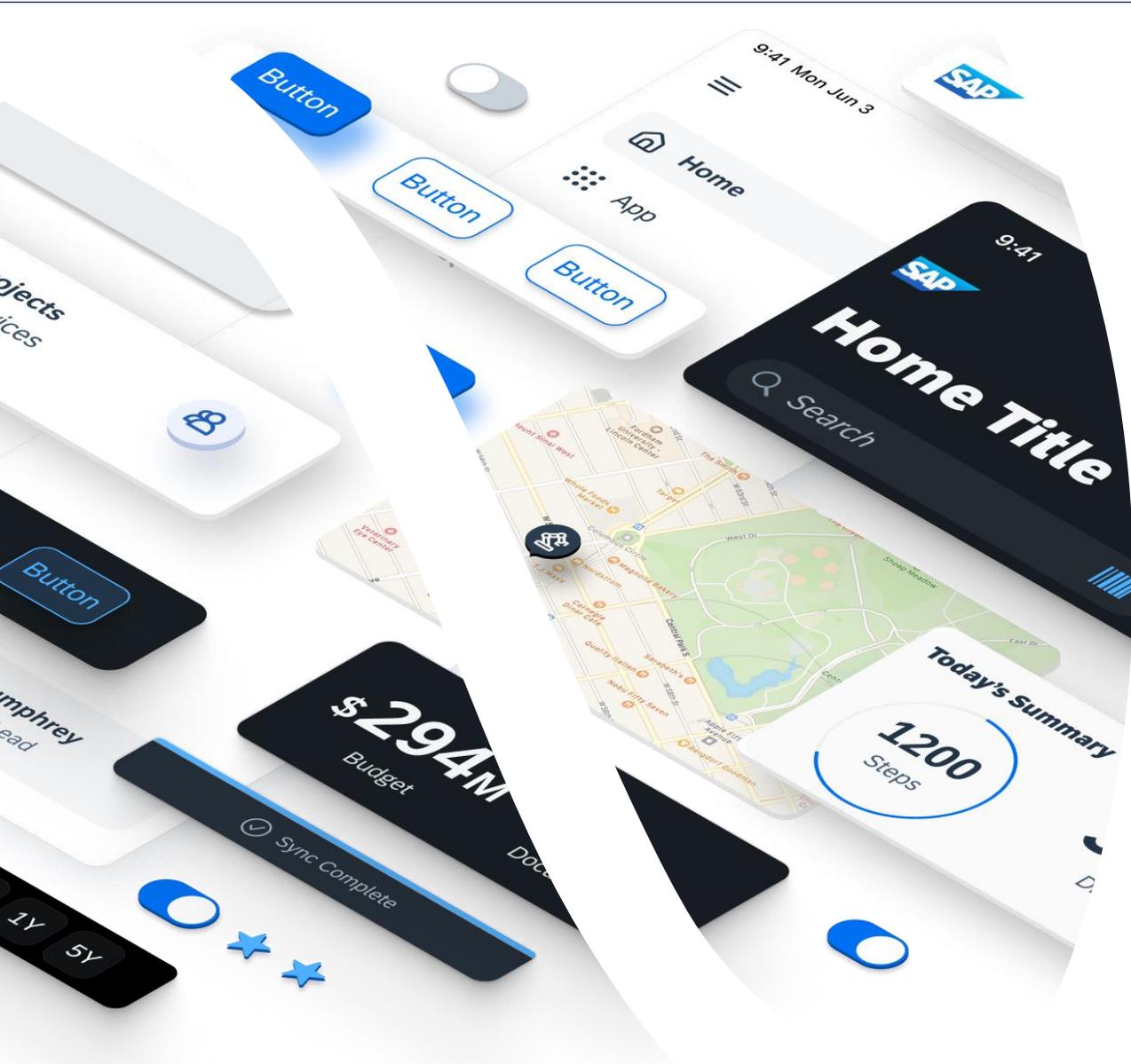
- The main success criterion is as follows:
 - The visual presentation of text and images of text has a contrast ratio of at least 4.5:1, except for the following:
 - Large Text: Large-scale text and images of large-scale text have a contrast ratio of at least 3:1.

LEVEL TRIPLE A (AAA)

- Is the strictest level and can introduce significant design challenges given the nature of the success criteria. It is considered the gold standard level of accessibility.
- Example: Guideline 2.4.13 **Focus Appearance** instead of just **Focus visible** (2.4.7 AA)



- Users should be able to see where the keyboard focus is as they navigate around a web page. The WCAG 2.2 Success Criterion adds some specifics as to how contrasted the focus element should be.
- People navigating with a keyboard need to know which element has the current focus, so they don't lose their way on the web page.

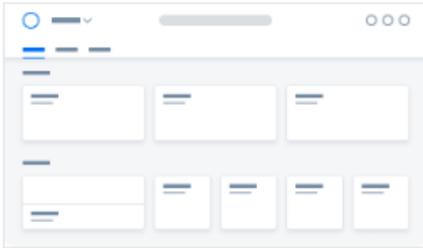


02

ACCESSIBILITY IN FIORI CONTEXT

FIORI THEMES COMPLIANCE 1/2

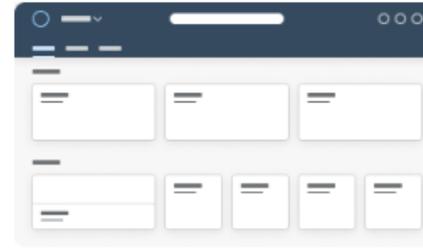
SAP Fiori Themes



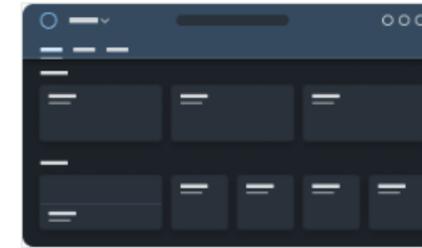
Morning Horizon



Evening Horizon

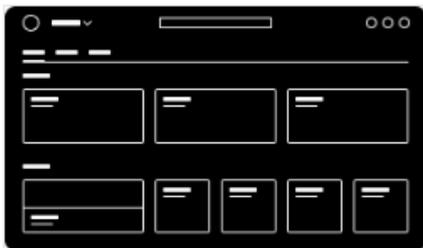


Quartz Light

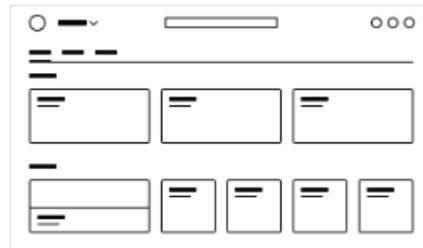


Quartz Dark

SAP Fiori High Contrast Themes



High-Contrast Black theme



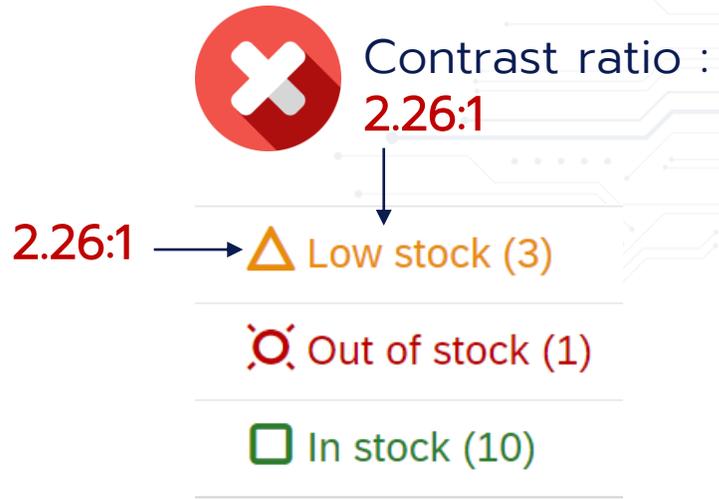
High-Contrast White theme

FIORI THEMES COMPLIANCE 2/2

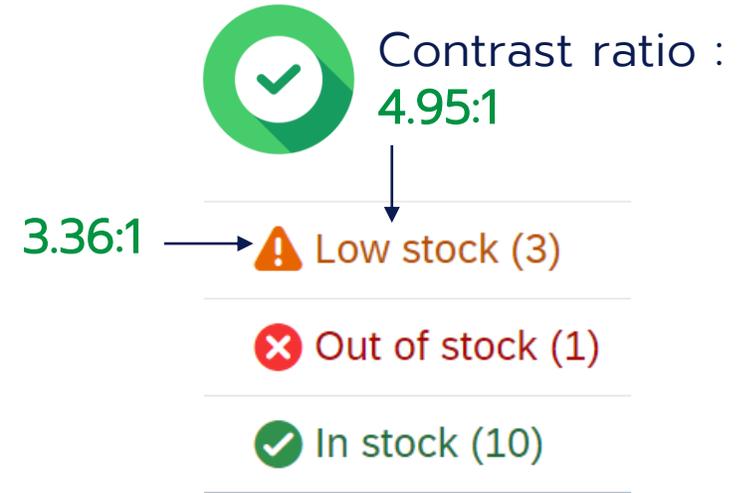
COMPLIANCE WCAG						
FIORI THEME NAME	WCAG 2.0 AA	WCAG 2.0 AAA	WCAG 2.1 AA	WCAG 2.1 AAA	WCAG 2.2 AA	WCAG 2.2 AAA
Horizon (Morning / Evening)	✓	✓	✓	✓	✓	✗
Horizon High Contrast (Black / White)	✓	✓	✓	✓	✓	✓
Quartz (Light / Dark)	✓	✓	✓	✗	✗	✗
Quartz High Contrast (Black / White)	✓	✓	✓	✓	✗	✗
Belize	✓	✗	✗	✗	✗	✗
Belize High Contrast (Black / White)	✓	✓	✗	✗	✗	✗

CONTRAST RATIO IMPROVEMENT HORIZON THEME VS BELIZE

BELIZE DEEP THEME



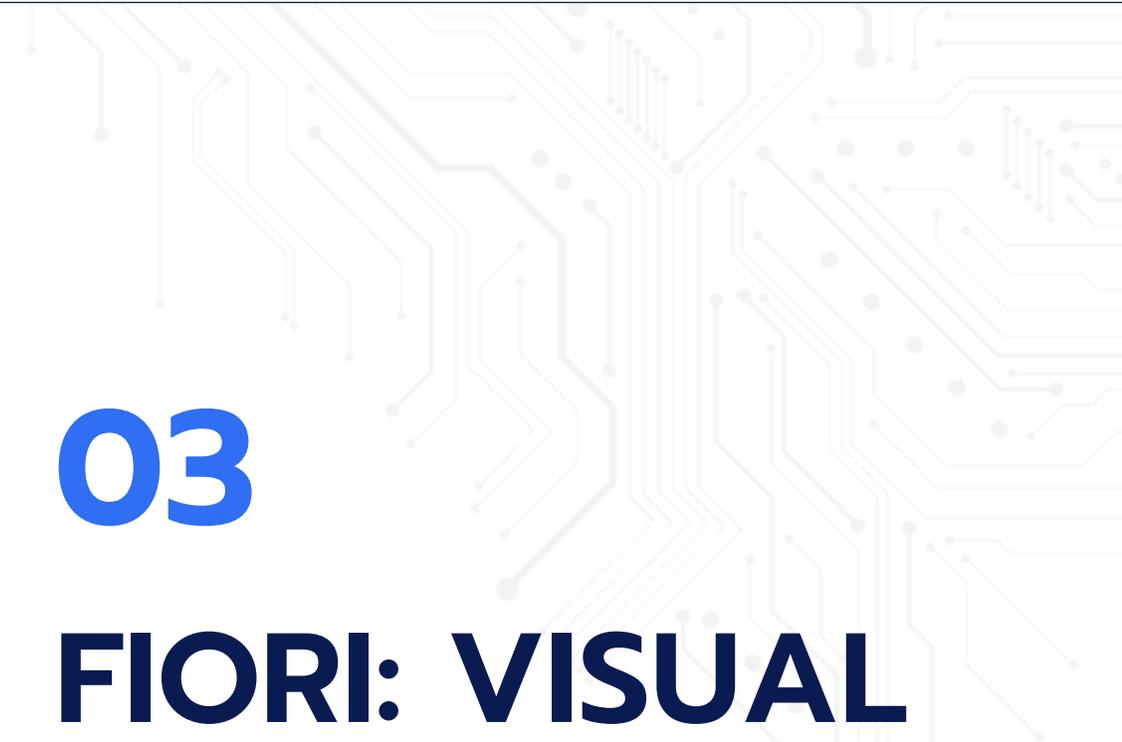
HORIZON MORNING THEME





03

FIORI: VISUAL IMPAIRMENTS CASE



VISUAL IMPAIRMENTS CASE – WHAT IT CAN BE

Visual acuity (clarity)



Field of vision



Light sensitivity



5 CATEGORIES OF VISUAL IMPAIREMENT

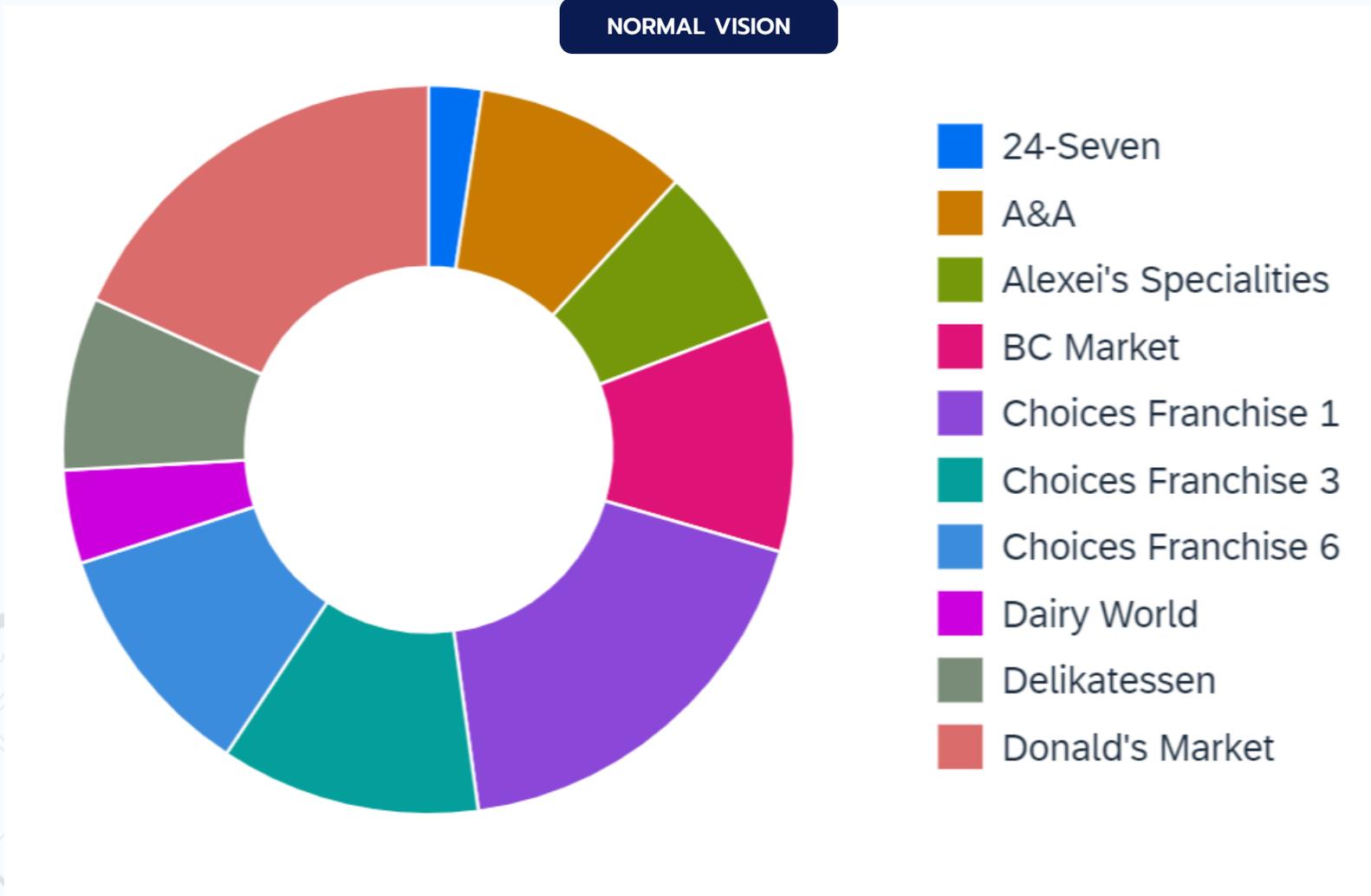
Contrast sensitivity



Color blindness

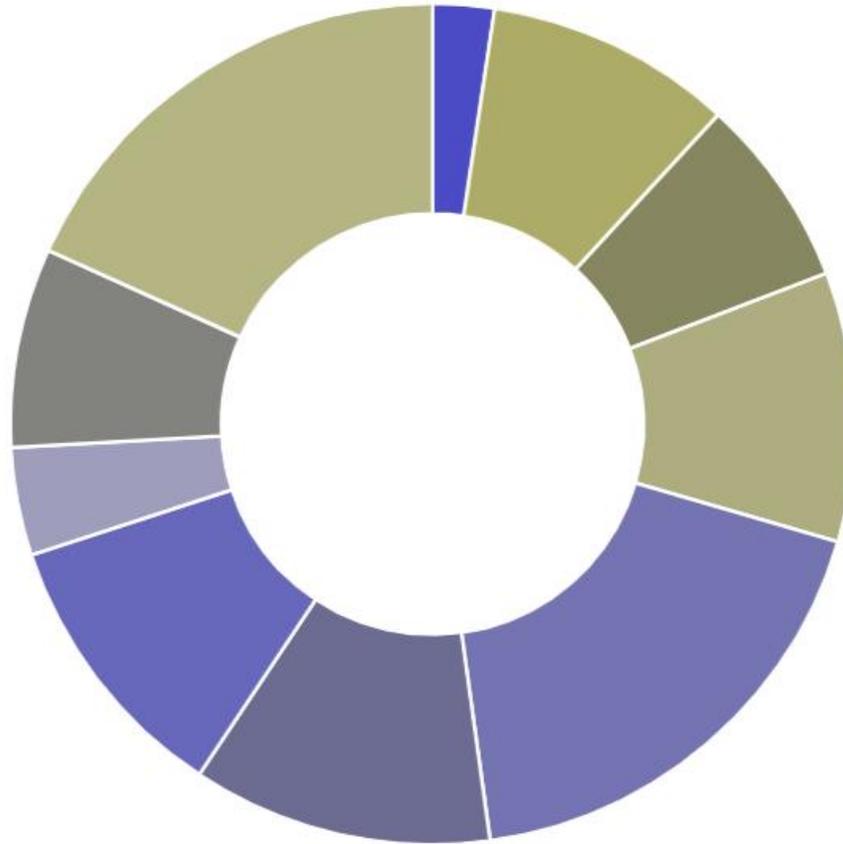


FOCUS ON COLOR BLINDNESS



FOCUS ON COLOR BLINDNESS

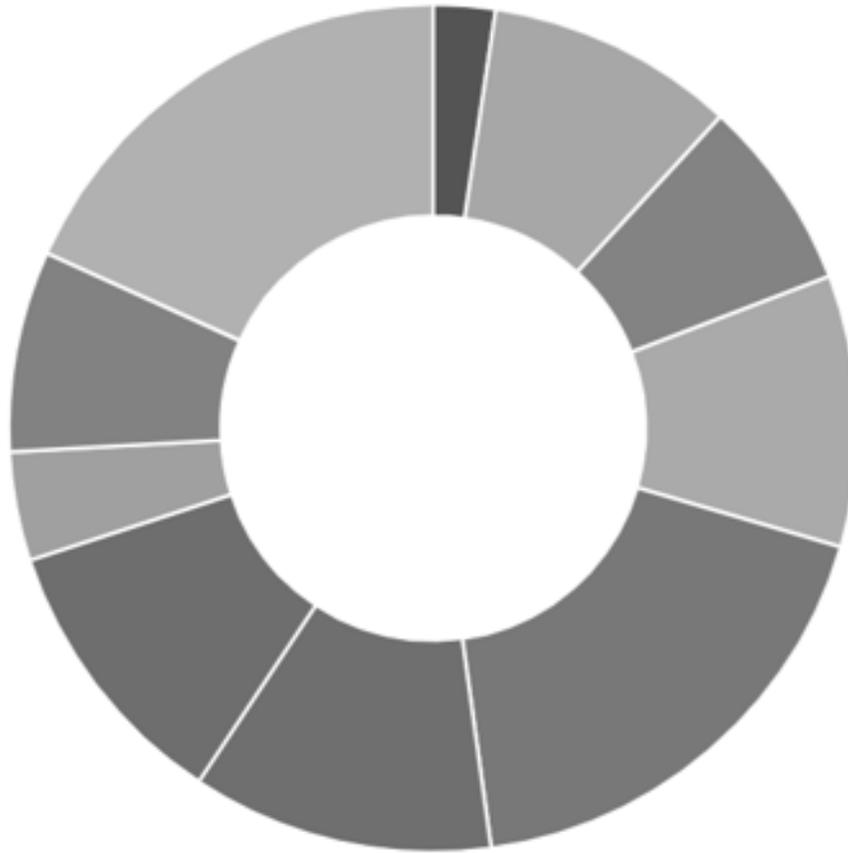
PROTANOPIA



- 24-Seven
- A&A
- Alexei's Specialities
- BC Market
- Choices Franchise 1
- Choices Franchise 3
- Choices Franchise 6
- Dairy World
- Delikatessen
- Donald's Market

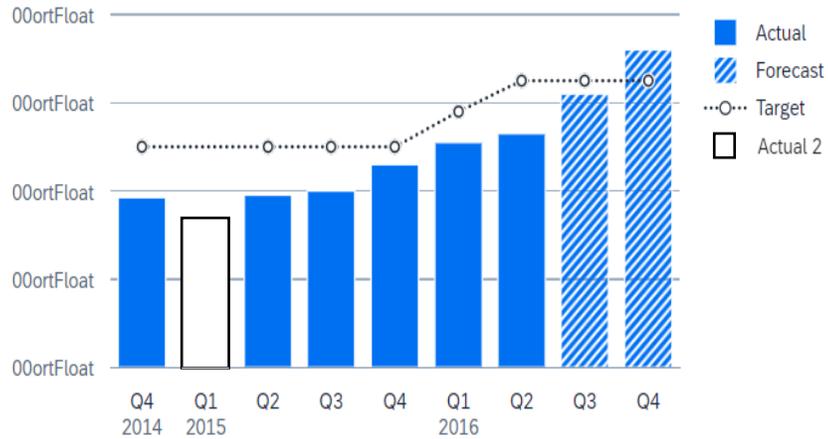
FOCUS ON COLOR BLINDNESS

ACHROMATOPSIA



- 24-Seven
- A&A
- Alexei's Specialities
- BC Market
- Choices Franchise 1
- Choices Franchise 3
- Choices Franchise 6
- Dairy World
- Delikatessen
- Donald's Market

ALTERNATIVE



We can use this type of chart, with different kinds of charts (Line, StackedBar, hatched bar, empty bar) to help users distinguish between the different indicators.

```
var vizProperties = {
  ...
  seriesStyle: {
    "rules": [{
      "dataContext": {
        "Actual2": "*"
      },
      "properties": {
        "dataPoint": {
          "pattern": "noFill"
        }
      }
    }
  ],
  dataPointStyle: {
    "rules": [{
      "dataContext": {
        "Actual": "*"
      },
      "properties": {
        "color": "sapUiChartPaletteQualitativeHue1"
      },
      "displayName": "Actual",
      "dataName": {
        "Actual": "Actual"
      }
    }, {
      "dataContext": {
        "Actual": "*"
      },
      "properties": {
        "color": "sapUiChartPaletteQualitativeHue1",
        "pattern": "diagonalLightStripe"
      },
      "displayName": "Forecast",
      "dataName": {
        "Actual": "Forecast"
      }
    }, {
      "dataContext": {
        "Target": "*"
      },
      "properties": {
        "color": "sapUiChartZeroAxisColor",
        "lineColor": "sapUiChartZeroAxisColor",
        "lineType": "dot",
        "pattern": "noFill"
      },
      "displayName": "Target",
      "dataName": {
        "Target": "Target"
      }
    }
  ]
  ...
}
```

VizFrame is a viz control that manages an initialization of a visualization layout, feeding, customization and interactions. It has various properties and features in SAPUI5.

In this example we used four of its properties:

- **noFill pattern** : to have a bar chart with no filling
- **diagonalLightStripe pattern**: to have a bar chart with hatched filling
- **Properties lineType/lineColor** to have a line chart with different types (hatched line, point line, continuous line) and colors, if we add the pattern **noFill** the bullet points will be with no color
- **Color** If we put only color in the properties without pattern we will have a colored bar chart.

SCREEN READER

A screen reader is an assistive technology that is intended to help people with **vision impairments**. It converts text, images and other screen elements into **speech** or **braille**. In the context of this presentation we will focus on **speech**.

SAPUI5 offers Screen Read support. The implementation is based on the HTML and ARIA standards.



GENERAL SCREEN READER

3 modes:

- Virtual/browse mode:
- Forms/focus mode:
- Application mode:

3 principles:

- Keyboard access
- Interactive Elements must have a role
- Elements with a role must have an accessible name

NVDA

- Open Source
- Implying a large community ready to help
- Many extensions available
- Virtual mode is called « Browse mode »



ARIA ATTRIBUTES

WHAT IS ARIA ?



Accessible Rich Internet Applications (ARIA) is a Set of attributes and roles that can be added to HTML elements to improve the accessibility of web content.

They are used by Screen readers to recognize the application structure.

Examples of usage:

- Describe the state of an element
- Describe the structure of the page
- Provide keyboard navigation

ARIA ROLES, STATES & PROPERTIES ATTRIBUTES

ARIA ROLES ATTRIBUTES

In HTML added by `role="<ROLE TYPE>"`

A **ARIA Role** attribute does not change for an element once it set.

In a SAPUI5/Fiori context for this presentation we will focus on the category « **Landmark** » Roles (there are more like Document Structures & Widget Roles).

ARIA STATE & PROPERTIES ATTRIBUTES

Used to support ARIA roles that exist on a page.

- **ARIA Properties** describe relationships with other elements and most of the time, do not change once they are set.
- **ARIA States** are more dynamic as they are intended to inform the Screen reader of the current state of a given element.

ARIA ROLE, STATE & PROPERTIES

```
<section>
  <div role="tablist">
    <button role="tab" aria-selected="true">Veg</button>
    <button role="tab" aria-selected="false">Non-Veg</button>
    <button role="tab" aria-selected="false">Vegan</button>
  </div>
  <div role="tabpanel" aria-labelledby="..."></div>
  <div role="tabpanel" aria-labelledby="..."></div>
  <div role="tabpanel" aria-labelledby="..."></div>
</section>
```

ROLE

A name to identify by assistive technologies.

STATE

State is dynamic & needs to update when the user interacts. Screen reader are notified of the changes.

PROPERTIES

Describes relationships with other elements & they do not change once they are set.

LANDMARK ARIA ROLE (1/2)

LANDMARK ARIA ROLES

Property `landmarkInfo`: used to define accessible landmarks for different regions of a `sap.m.Page`.

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m">
  <Page
    title="Titre de la Page"
    landmarkInfo="{
      headerRole: 'Banner',
      headerLabel: 'Main Header',
      subHeaderRole: 'Navigation',
      subHeaderLabel: 'Sub Header',
      contentRole: 'Main',
      contentLabel: 'Main Content',
      footerRole: 'ContentInfo',
      footerLabel: 'Additional information'
    }">
    <content>
      <!-- Page content goes here -->
    </content>
  </Page>
</mvc:View>
```

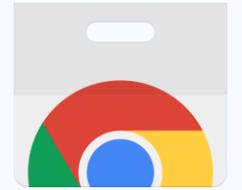
LANDMARK INFO ATTRIBUTES

landmarkInfo property	ARIA Landmarks in HTML
headerRole	role="banner"
subHeaderRole	role="navigation"
contentRole	role="main"
footerRole	role="contentinfo"

Usage of `headerLabel`, `subHeaderLabel`, `contentLabel`, `footerLabel` helps to further enhance the accessibility of these landmarks for assistive technologies such as Screen Reader.

LANDMARK ARIA ROLE (2/2)

The screenshot displays the SAP Fiori accessibility checker interface. At the top, there is a navigation bar with the SAP logo and the text 'ZCA_ACCESSIBILITY'. Below this, a header section contains the text 'Check controls and implementation for Accessibility En-tête : utiliser mode lecture du lecteur d'écran pour lire toutes les informations sur l'objet (En-tête de page d'objet)'. The main content area is titled 'Form Fields' and contains several input controls, each with a pink 'Form' label above it. These controls include: an 'Input' field with a placeholder; an 'Input Text with description' field with a placeholder and a 'Description' label; an 'Input Number' field with a placeholder; an 'Input with value help' field with a placeholder and a help icon; a 'Multi input with value help' field with a placeholder and a help icon; an 'Input with mask' field with a placeholder; a 'Select' dropdown menu with 'Comfort Easy' selected; a 'Combo box' with a placeholder; a 'Multi Combo box' with a placeholder; a 'Text area with showExceededText = true' with a placeholder and a note '10 caractères restants'; a 'Date picker' with a placeholder and a calendar icon; an 'Input group with 2 field 1' with a placeholder; an 'Input group with 2 field 2' with a placeholder; an 'Input group with 3 field 1' with a placeholder; an 'Input group with 3 field 2' with a placeholder; an 'Input group with 3 field 3' with a placeholder; a 'Radio button Group' with three radio buttons labeled 'Radio 1', 'Radio 2', and 'Radio 3'; a 'Checkbox' group with three checkboxes labeled 'Checkbox 1', 'Checkbox 2', and 'Checkbox 3'; and a 'Rating indicator' with five stars, the first of which is filled. The interface also features a 'Barre de navigation sections' at the top right and a 'Form Fields Sous-section (Region)' label. At the bottom, there is a 'Combo box date picker' and a 'Combo box date picker Sous-section (Region)' label. The overall layout is clean and professional, with a focus on accessibility testing.



LANDMARK NAVIG.
VIA KEYBOARD

ARIA STATES & PROPERTIES

ARIA STATES ATTRIBUTES

Helps the screen reader to understand when an interaction happened and what changed on the DOM.

EXAMPLE OF SAPUI5/ARIA STATES MAPPING

SAPUI5 Property	ARIA Attribute
editable	aria-readonly
enabled	aria-disabled
decorative <i>(from sap.m.Image)</i>	aria-hidden
selected	aria-selected

ARIA PROPERTIES ATTRIBUTES

Properties attributes are less likely to change throughout the application lifecycle than the values of states attributes.

EXAMPLE OF SAPUI5/ARIA PROPERTIES MAPPING

SAPUI5 Property	ARIA Attribute
required	aria-required
ariaHasPopup	aria-haspopup
ariaLabelledBy	aria-labelledby
ariaDescribedBy	aria-describedby
InvisibleMessageMode	aria-live
text	aria-label

ARIA STATES & PROPERTIES (EXAMPLE)

```
<Button text="My Button" icon="sap-icon://action" ariaHasPopup="true" class="sapUiSmallMargin"/>
```

Property « **ariaHasPopup** » is here to indicate that the button is related to a popup. It will be interpreted as such by a Screen reader.

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m">  
  <Image src="logo.png" decorative="true"/>  
</mvc:View>
```

Property « **decorative** » here will be interpreted as « **aria-hidden** » in HTML.

INVISIBLE CONTENT

INVISIBLE MESSAGING

Invisible Messaging: generates dynamically a message that is not visible to sighted users but can be read by Screen Readers. Useful to provide additional context, descriptions.

```
sap.ui.define([
  "sap/ui/core/mvc/Controller",
  "sap/ui/core/InvisibleMessage",
  "sap/ui/core/library"
], function(Controller, InvisibleMessage, coreLibrary) {
  "use strict";

  return Controller.extend("YourNamespace.ControllerName", {
    onInit: function() {
      this.oInvisibleMessage = InvisibleMessage.getInstance();
    },

    onPress: function() {
      const sMessage = "This is an example message for the screen reader.";
      this.oInvisibleMessage.announce(sMessage, coreLibrary.InvisibleMessageMode.Assertive);
    }
  });
});
```

`sap.ui.core.InvisibleMessageMode` defines the mode:

- **Polite:** screen reader is able to finish reading what is was focused on before reading the message.
- **Assertive:** screen reader interrupts what it is doing to read the message immediately.

INVISIBLE TEXT

Invisible Text: display a text that is not visible to sighted users but can be read by Screen Readers.

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m" xmlns:core="sap.ui.core">
  <core:InvisibleText id="helpText" text="Open help dialog" />
  <Button icon="sap-icon://question-mark" press="onHelpPress" ariaDescribedBy="helpText" />
</mvc:View>
```

`ariaDescribedBy` references the hidden text. `ariaLabelledBy` can also be used for short text as Label.

DEMO: INVISIBLE MESSAGING USECASE

Invisible Message

Information

Success

Error

Emphasized

A new message with text: "Button with type Default and text Infromation is pressed" was sent to the invisible messaging service.

LABELING

LABELFOR

```
<Label text="First Name" labelFor="firstNameInput" />  
<Input id="firstNameInput" placeholder="fisrt name"/>
```

« **labelFor** » attribute plays a crucial role in ensuring accessibility for users relying on screen readers. It is the the “for” attribute in plain HTML.

By associating a label with a specific UI element, such as an input field or button, users can better understand the purpose and function of each control.

ARIALABELLEDBY

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m" xmlns:core="sap.ui.core">  
  <Label text="Postal Code / City" />  
  <Input id="postalCodeInput" placeholder="Postal Code" type="Text"  
    ariaLabelledBy="postalCodeDescription"/>  
  <Input id="cityInput" placeholder="City" type="Text"  
    ariaLabelledBy="cityDescription" />  
  <core:InvisibleText id="postalCodeDescription" text="Postal Code" />  
  <core:InvisibleText id="cityDescription" text="City" />  
</mvc:View>
```

« **ariaLabelledBy** » is an ARIA attribute used to associate an element with another serving as its label, enhancing accessibility by providing context where explicit labels are absent.

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m">  
  <Title id="personalDetailsTitle" text="Personal Details"/>  
  <Label id="fullNameLabel" text="Full Name"/>  
  <Input ariaLabelledBy="personalDetailsTitle fullNameLabel"/>  
</mvc:View>
```

It offers a solution for associating multiple labels with a single UI control. Unlike the **labelFor** attribute, which associates a single label with a control, **ariaLabelledBy** allows for a whitespace-separated list of ID references to be assigned to the control, enabling it to be labeled by multiple elements.

DESCRIPTION

ARIADESCRIBEDBY

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m" xmlns:core="sap.ui.core">
  <Label text="Postal Code / City" />
  <Input id="postalCodeInput" placeholder="Postal Code" type="Text"
    ariaDescribedBy="postalCodeDescription"/>
  <Input id="cityInput" placeholder="City" type="Text"
    ariaDescribedBy="cityDescription" />
  <core:InvisibleText id="postalCodeDescription" text="Please enter your postal code" />
  <core:InvisibleText id="cityDescription" text="Please enter your city" />
</mvc:View>
```

« **ariaDescribedBy** » is an alternative to « **ariaLabelledBy** ». This ARIA attribute is used to associate an element with another serving as its description, enhancing accessibility by providing context.

DEMO: INVISIBLETEXT USECASE

SAP ZCA_ACCESSIBILITY ▾

Check controls and implementation for Accessibility

< 4 **Form Fields** ▾ Step Input Buttons ▾ Dialog and Message ▾ Header and Title 1 ▾

<p>Name: *</p> <input type="text" value="example: Rick Nova"/>	<p>Select:</p> <input style="border-bottom: 1px solid black;" type="text" value="Comfort Easy"/> ▾
<p>Postal Code / City:</p> <input style="border-bottom: 1px solid black;" type="text" value="Postal Code"/> <input style="border-bottom: 1px solid black;" type="text" value="City"/>	<p>Combo box:</p> <input style="border-bottom: 1px solid black;" type="text" value="placeholder"/> ▾
<p>Input Text with description:</p> <input style="border-bottom: 1px solid black;" type="text" value="placeholder"/> Description	<p>Multi Combo box:</p> <input style="border-bottom: 1px solid black;" type="text" value="placeholder"/> ▾
<p>Input with value help:</p> <input style="border-bottom: 1px solid black;" type="text" value="placeholder"/>	<p>Text area with showExceededText = true:</p> <div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>
<p>Multi input with value help:</p>	

LABELING IMAGE CASE

DECORATIVE

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m">
  <Image src="logo.png" decorative="true"/>
</mvc:View>
```

Setting « **decorative** » property for a `sap.m.Image` to « **true** » means you intend to mark the image as purely decorative. This configuration informs assistive technologies that the image can be ignored in the context of Screen Reading as it does not contribute essential information to the content of the page.

In pure HTML, « **aria-hidden** » will be attributed to this image.

The default value for « **decorative** » is true

NOT DECORATIVE

By setting « **decorative** » property for a `sap.m.Image` to « **false** », the Screen Reader will not ignore it.

In the example below because the image is interactive, it has to be « **decorative=false** ». It is an essential information / function.

The default value for « **decorative** » is true

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m">
  <Image src="settings_icon.png"
    decorative="false"
    alt="Open settings"
    press="onSettingsOpen"/>
</mvc:View>
```

Because the default value for « **decorative** » is true, it is primordial to manually set the decorative property to preserve your applications functionalities in the of Screen reader usage.

DEMO: DECORATIVE IMAGE

SAP ZCA_ACCESSIBILITY ▾

Check controls and implementation for Accessibility

Appearance ▾

Fiori Conf 2024 Form Fields ▾ Step Input Buttons ▾ Dialog and Message ▾ Header and Title 1 ▾ Header and Title 2 ▾ Header and Title 3 > ▾

Non decorative image with an alternative text



Decorative image used for ambiance, alternative text is ignored





04

FIORI: MOTOR IMPAIRMENTS CASE

MOTOR IMPAIRMENTS: WHO IS CONCERNED?

MOTOR IMPAIRMENTS ARE OFTEN VISIBLE DISABILITIES :

- having total loss of function of a body part
- muscle weakness, stamina loss, lack of muscle control, or paralysis

2 CATEGORIES OF MOTOR IMPAIRMENT

Traumatic Injuries



Diseases and Congenital Conditions



CHALLENGES & OBJECTIVES

CHALLENGES



1. Users may not be able to use the mouse.
2. Users may not be able to control the mouse or the keyboard well.
3. Users may be using voice-activated software.
4. Users may become fatigued when using adaptive technologies
5. Users may not be able to physically interact with their hardware device

OBJECTIVES



1. Functions are available to both mouse/touchpad and keyboard users
2. Ensure forms are error-tolerant. Create sufficiently large links and buttons that remain in a static position. Allow users to remap or disable single key shortcuts.
3. Functions are available from the keyboard
4. Provide a method for skipping over long lists of links or other lengthy content
5. Ensure content works in both horizontal or vertical orientation.

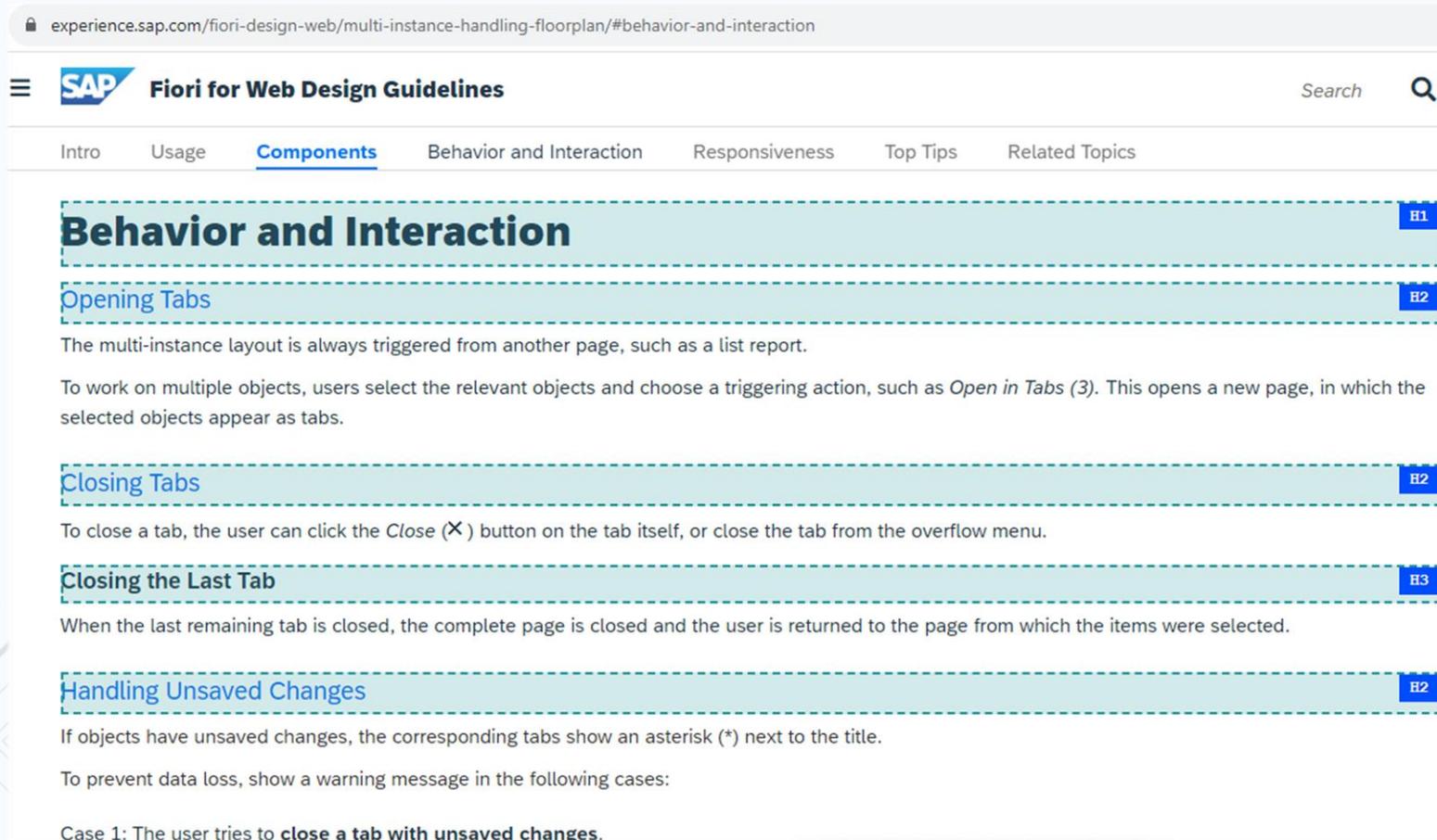
LOGICAL STRUCTURE OF HEADINGS

- Screen readers rely on heading tags to determine the content of the page.
- When heading levels are coded, assistive technologies can use them to navigate quickly jump from heading to heading.
- Text that's large, bold, or styled differently to the paragraph text is not interpreted as a heading unless you use a heading tag.

The screenshot displays a web application interface with a breadcrumb trail at the top: < > Header and Titl... > Header and Titl... > Header and Titl... >. Below the breadcrumb, the content is organized into sections with different heading levels:

- Object Header**: A light gray header bar.
- Notebook Basic 15**: A large, bold heading centered in a white box.
- Title H2**: A light gray header bar.
- Section title H2**: A large, bold heading centered in a white box.
- Title H3**: A light gray header bar.
- Sub section title H3**: A large, bold heading centered in a white box.
- Below the H3 heading is a paragraph of Lorem ipsum text: "Lorem ipsum dolor st amet, consetetur sadipscing elitr, sed diam nonumy eirmod t duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Le tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Lore dolore magna aliquyam erat".
- Title H3 with level property**: A light gray header bar.

HEADINGS LEVELS COMPLIANT CASE 1/2



The screenshot shows a web browser window with the URL `experience.sap.com/fiori-design-web/multi-instance-handling-floorplan/#behavior-and-interaction`. The page title is "SAP Fiori for Web Design Guidelines". The navigation menu includes "Intro", "Usage", "Components", "Behavior and Interaction", "Responsiveness", "Top Tips", and "Related Topics". The main content area is titled "Behavior and Interaction" (H1) and contains several sub-sections:

- Opening Tabs** (H2): The multi-instance layout is always triggered from another page, such as a list report. To work on multiple objects, users select the relevant objects and choose a triggering action, such as *Open in Tabs (3)*. This opens a new page, in which the selected objects appear as tabs.
- Closing Tabs** (H2): To close a tab, the user can click the *Close (X)* button on the tab itself, or close the tab from the overflow menu.
- Closing the Last Tab** (H3): When the last remaining tab is closed, the complete page is closed and the user is returned to the page from which the items were selected.
- Handling Unsaved Changes** (H2): If objects have unsaved changes, the corresponding tabs show an asterisk (*) next to the title. To prevent data loss, show a warning message in the following cases:
 - Case 1: The user tries to **close a tab with unsaved changes**.

Source: [experience.sap.com Fiori Web Design Guidelines](https://experience.sap.com/fiori-design-web/multi-instance-handling-floorplan/#behavior-and-interaction)

HEADINGS LEVELS COMPLIANT CASE 2/2



Behavior and Interaction H1

Opening Tabs H2

The multi-instance layout is always triggered from another page, such as a list report.

To work on multiple objects, users select the relevant objects and choose a triggering action, such as *Open in Tabs* (3). This opens a new page, in which the selected objects appear as tabs.

Closing Tabs H2

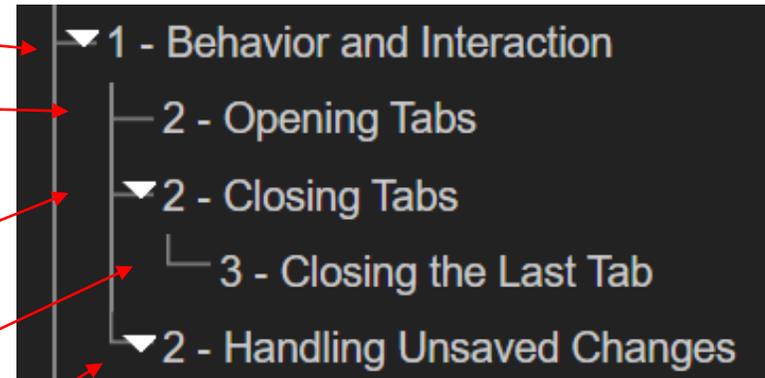
To close a tab, the user can click the *Close* (X) button on the tab itself, or close the tab from the overflow menu.

Closing the Last Tab H3

When the last remaining tab is closed, the complete page is closed and the user is returned to the page from which the items were selected.

Handling Unsaved Changes H2

If objects have unsaved changes, the corresponding tabs show an asterisk (*) next to the title.



Each heading identify its content section.

SETTING THE VISIBILITY OF THE FOCUS

BAD PRACTICE



- Removing focus visibility or light focus visibility
- Alternating focus visibility

Purchasing Group: *

Test CHARM (*01)



Storage location:



GOOD PRACTICE



- Using the default focus indicator
- Using an Author-Supplied, visible focus indicator, For example on SAPUI5 form fields

Input without value state :

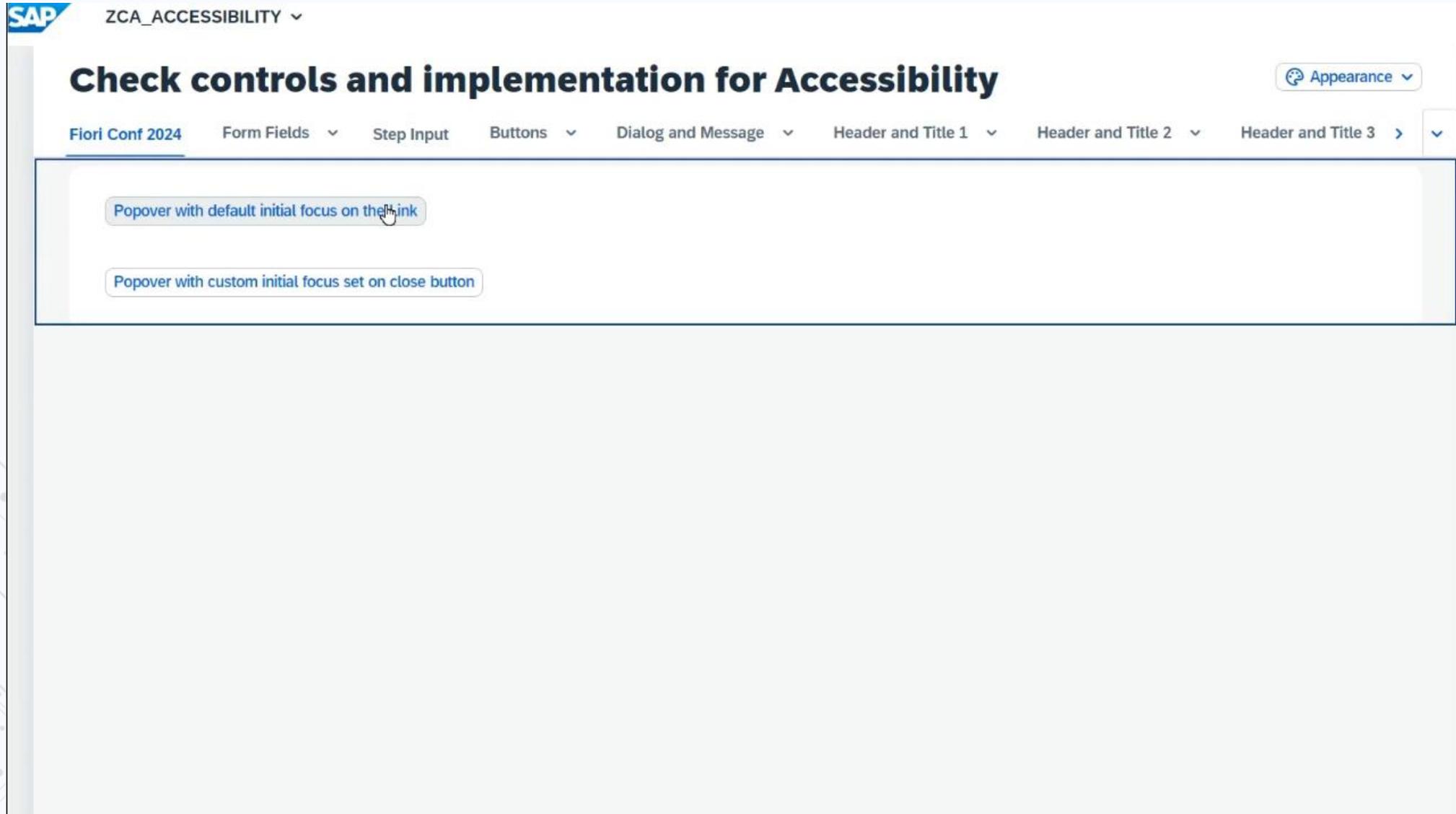
Placeholder Text

Input with value state :

Placeholder Text

WCAG 2.2 SUCCESS CRITERION 2.4.7

DEMO: SETTING THE VISIBILITY OF THE FOCUS



The screenshot shows the SAP Fiori Accessibility demo interface. At the top left, the SAP logo is visible next to the text 'ZCA_ACCESSIBILITY'. The main heading is 'Check controls and implementation for Accessibility'. In the top right corner, there is an 'Appearance' dropdown menu. Below the heading, a navigation bar contains several menu items: 'Fiori Conf 2024', 'Form Fields', 'Step Input', 'Buttons', 'Dialog and Message', 'Header and Title 1', 'Header and Title 2', and 'Header and Title 3'. The main content area is a light gray box containing two items, each with a blue link-like text and a small icon:

- Popover with default initial focus on the link
- Popover with custom initial focus set on close button

IMPLEMENTATION OF A SHORCUT

SHORTCUT

```
<VBox class="sapUiSmallMargin">
  <depends>
    <core:CommandExecution id="idOpenVh" command="commandValueHelp" enabled="true"
execute="handleValueHelp"/>
  </depends>
  <f:SimpleForm editable="false" layout="ColumnLayout" adjustLabelSpan="true"
columnsXL="2" columnsL="2" columnsM="2" singleContainerFullSize="false">
    <f:content>
      <Label text="Product"/>
      <MultiInput id="InputValueHelpCommand"
valueStateText="Please, use the alt + w short cut to open the product selection dialog"
valueHelpRequest="handleMultValueHelp">
        <core:Item key="{ProductId}" text="{Name}"/>
      </MultiInput>
    </f:content>
  </f:SimpleForm>
</VBox>
```

View.xml

sap.ui.core.CommandExecution Simplifies shortcut implementation in SAPUI5, enabling easy registration within controls and XML views. Through automatic model creation, it centralizes control over **CommandExecution** states, enhancing accessibility and user experience.

```
"sap.ui5": {
  "flexEnabled": false,
  "rootView": {
    ...
  },
  "dependencies": {
    ...
  },
  "commands": {
    "commandValueHelp": {
      "shortcut":
"Alt+w"    }
  }
}
```

Manifest.json

The shortcut information is retrieved from the owner components manifest (/sap.Ui5/commands/<command>).

MAIN KEYBOARD COMMANDS

Keyboard Handling for Basic Navigation

[Edit on GitHub](#)

The following keys and key combinations are used for navigation between controls within an application.

Standard Navigation

Navigation between controls is done using the `Tab` key. `Tab` moves the focus from one control to the next one inside the application. The tab order is defined by the placement of the control within the DOM tree, therefore apps have a large influence on it.

Note

Controls are in the tab order, if they are interactive, enabled and visible. This includes read-only controls. Disabled or hidden controls are taken out of the tab order. Non-interactive controls (for example, layout container) can never be reached with `Tab`.

Key combination	Behavior
<code>Tab</code>	Forward Navigation: On enter, move focus to the control. On leave, move focus to the next control in the application.
<code>Shift</code> + <code>Tab</code>	Backward Navigation: On enter, move focus to the control. On leave, move focus to the previous control in the tab order.

Source : [Keyboard Handling for SAPUI5 UI Elements](#)

DEMO: DEFAULT & CUSTOM SHORTCUTS

SAP ZCA_ACCESSIBILITY ▾

Check controls and implementation for Accessibility Appearance ▾

< 4
Form Fields ▾
Step Input
Buttons ▾
Dialog and Message ▾
Header and Title 1 ▾
Header and Title 2 ▾
Header and Title 3 ▾
In > ▾

Label:
Value 0

Label:
Value 1

Command

Product:

🔗

ℹ Please, use the alt + w short cut to open the product selection dialog

MEASUREMENTS AND GREEN IT



Ecosonar - Ecodesign and Accessibility Development Tool

- Recommendations to guide developers implement most impactful practices
- Raising awareness to environmental issues and eco-design practices
- Measuring the carbon impact across digital services
- Proving that web-based applications can have minimal carbon footprint
- Get an environmental & performance monitoring solution

The screenshot displays the Ecosonar dashboard with the following sections:

- ECOINDEX PROJECT GRADE**: Shows a grade of **E** with the text "EcoIndex Score for project (average from all pages) is 25".
- LIGHTHOUSE PERFORMANCE SCORE**: Shows a grade of **E** with the text "Lighthouse Performance Score for project (average from all pages) is 26".
- LIGHTHOUSE ACCESSIBILITY SCORE**: Shows a grade of **B** with the text "Lighthouse Accessibility Score for project (average from all pages) is 92".
- W3C VALIDATOR PROJECT GRADE**: Shows a grade of **B** with the text "W3C Validator Score for project is 97".
- ECOSONAR ANALYSIS**: Last analysis at Tue Jan 24 2023 - 09:13:33. It includes a table with two columns: "EcoSonar Analysis per URL" and "EcoSonar Analysis for project".

EcoSonar Analysis per URL	EcoSonar Analysis for project
	1782 Size of the Dom (average)
	76 Number of requests (average)
	6320 Size of the Page in Kb (average)
- LIGHTHOUSE METRICS**: Shows "Performance:26" with a red **E** grade indicator.
- Issues List**: Shows two issues for the file "src/.../java/com/examplehub/basics/BreakWithLabel.java". Both issues are "Code Smell", "Minor", "Open", "Not assigned", and "20min effort". The issue text is "Don't concatenate Strings in loop, use StringBuilder instead. Why is this an issue?".

MAIN COMPONENTS OF ECOSONAR





06

CONCLUSION

CONCLUSION

“Accessible design is good design.”
- Steve Ballmer

The very essence of SAP Fiori is still the same:

Provide a delightful, adaptive, simple and coherent experience...
...For everyone !

ACCESSIBILITY X SUSTAINABILITY STARTER PACK

TAKEAWAY



SAPUI5

- [Accessibility Guide](#)
- [Accessibility in Demokit](#)



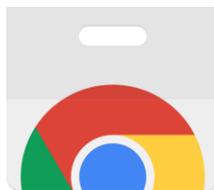
SAP Fiori

- [Accessibility in SAP Fiori](#)



WCAG 2

- [How to meet WCAG](#)



Chrome extensions

- [Landmark Navigation via Keyboard](#)
- [HeadingsMap](#)



ARIA

- [Documentation](#)
- [W3C \(WAI-ARIA\)](#)



EcoSonar

- [Main website](#)
- [GitHub Project](#)