

1 Prompting

This document shows several prompts used to generate the same web pages with IA Tools.

Web of the project: <https://github.com/UACodingAI>

Article sent on CHI'2026

Date of the document: 09/09/2025

Index of the document

1.1	Step 3. RICCE prompting.....	2
1.2	Step 4. Prompt improvement.....	12
1.3	Step 5. Prompt customization.	14
1.3.1	ChatGPT.....	15
1.3.2	Claude	18
1.3.3	Copilot	20
1.3.4	Gemini	23
1.3.5	Vercel.....	25
1.3.6	Builder.io	28
1.3.7	DEEPSITE.AI	30
1.3.8	BOLT.....	32

1.1 Step 3. RICCE prompting

Role:

You are an expert web design and accessibility designer.

Objective:

Create a landing page to present an accessibility course.

Context:

The code generated must have accessibility features such as complying with ALL WCAG 2.2 guidelines, 'Making Content Usable for People with Cognitive and Learning Disabilities' guidelines, and incorporating WAI-ARIA code into the appropriate elements. In addition, considerations for easy language, keyboard navigation, and landmarks that allow for zone navigation.

List of WCAG 2.2 guidelines (the WCAG 2.2 table is included directly)

Criterio WCAG	Nivel	Versión
1.3.1: Información y relaciones A		2.0
1.3.2: Secuencia significativa	A	2.0 / 2.1
3.1.1: Idioma de la página	A	2.0
3.1.2: Idioma de partes	AA	2.0
4.1.1: Parsing (Análisis)	A	2.0
4.1.2: Nombre, función, valor	A	2.0
4.1.3: Mensajes de estado	AA	2.1
2.4.1: Omitir bloques	A	2.0 / 2.1
2.4.5: Múltiples maneras	AA	2.0 / 2.1
3.2.3: Navegación consistente	AA	2.0
3.2.4: Identificación consistente	AA	2.0
3.2.6: Ayuda consistente	A	2.2
2.4.4: Propósito del enlace (en contexto)	A	2.0 / 2.1
2.4.3: Orden del foco	A	2.0 / 2.1
2.4.7: Foco visible	AA	2.0
2.4.11: Foco no oculto (mínimo)	AA	2.2
1.4.13: Contenido en foco o al pasar el mouse	AA	2.1
2.1.1: Teclado	A	2.0 / 2.1

2.1.2: Sin trampa de teclado	A	2.0 / 2.1	
2.1.4: Atajos de teclado	A	2.1	
2.2.1: Tiempo ajustable	A	2.0 / 2.2	
2.2.2: Pausar, detener, ocultar	A	2.0	
2.4.2: Título de la página	A	2.0 / 2.1	
2.4.6: Encabezados y etiquetas	AA	2.0 / 2.1	
1.3.3: Características sensoriales	A	2.0	
1.4.1: Uso del color	A	2.0	
1.4.3: Contraste (mínimo)	AA	2.0	
1.4.11: Contraste de elementos no textuales	AA	2.1	
1.3.4: Orientación	AA	2.1	
1.4.10: Reflujo (Reflow)	AA	2.1	
1.4.12: Espaciado de texto	AA	2.1	
2.5.8: Tamaño del objetivo (mínimo)	AA	2.2	
1.4.4: Redimensionar texto	AA	2.0	
2.5.1: Gestos de puntero	A	2.1	
2.5.2: Cancelación de puntero	A	2.1	
2.5.4: Activación por movimiento	A	2.1	
2.5.7: Movimientos de arrastre	AA	2.2	
3.2.1: Al recibir el foco	A	2.0	
3.2.2: Al ingresar datos	A	2.0	
1.3.5: Identificar propósito de entrada	AA	2.1	
2.5.3: Etiqueta en nombre	A	2.1	
3.3.1: Identificación de errores	A	2.0	
3.3.2: Etiquetas o instrucciones	A	2.0	
3.3.3: Sugerencia de errores	AA	2.0	
3.3.4: Prevención de errores (legal, financiero, datos)	AA	2.0	
3.3.7: Entrada redundante	A	2.2	
3.3.8: Autenticación accesible (mínimo)	AA	2.2	
1.1.1: Contenido no textual	A	2.0	

- 1.4.5: Imágenes de texto AA 2.0
- 1.2.1: Solo audio o solo vídeo (pregrabado) A 2.0
- 1.2.2: Subtítulos (pregrabados) A 2.0
- 1.2.3: Descripción de audio o alternativa A 2.0
- 1.2.4: Subtítulos (en vivo) AA 2.0
- 1.2.5: Descripción de audio (pregrabado) AA 2.0
- 1.4.2: Control de audio A 2.0
- 2.3.1: Tres destellos o menos A 2.0

Accessibility website sources:

WCAG 2.2 guidelines: <https://www.w3.org/TR/WCAG22/>

COGA guidelines: <https://www.w3.org/TR/coga-usable/>

<https://www.w3.org/TR/wai-aria/>

<https://www.w3.org/WAI/ARIA/apg/patterns/>

It should also include usability aspects.

Heuristic List

1 1- Visibilidad y estado del sistema / Visibility and system state

1.1 La aplicación incluye de forma visible el título de la página, de la sección o del sitio?

Does the application include a visible title page, section or site?

1.2 El usuario sabe en todo momento dónde está?

Does the user always know where he is located?

1.3 El usuario sabe en todo momento qué está haciendo el sistema o aplicación?

Does the user always know what the system or application is doing?

1.4 Los enlaces están claramente definidos?

Are the links clearly defined?

1.5 Todas las acciones pueden verse directamente? (Sin requerir acciones adicionales)

Ejemplo: Si nos encontramos dentro de un proceso de compra, todas las acciones necesarias en cada uno de los pasos de este proceso se debe poder ver y entender directamente, sin necesidad de realizar acciones adicionales.

Can all actions be visualized directly? (No other actions are required)

Example: If we are within a purchase process, all the necessary actions in each of the steps of this process must be seen and understood directly, without the need for additional actions.

2 "2 - Connexión entre el sistema y el mundo real, uso de metáforas y objetos humanos /

Connection between the system and the real world, metaphor usage and human objects"

2.1 La información aparece de una manera lógica para el usuario común?

Does the information appear in a logical way for the common user?

2.2 El diseño de los iconos se corresponde con objetos cotidianos?

Does the design of the icons correspond to everyday objects?

2.3 Cada icono realiza la acción que el usuario espera?

Each icon performs the action that the user expects?

2.4 Se utilizan frases y conceptos familiares para el usuario?

Are the phrases and concepts used familiar to the user?

3 3 - Control y libertad del usuario / User control and freedom

3.1 Existe un vínculo para volver al estado inicial o a la página de inicio?

Is there a link to return to the initial state or the home page?

3.2 Existen funcionalidades para "deshacer" y "rehacer"?

Are the functions "undo" and "redo" implemented?

3.3 Es fácil volver a un estado anterior de la aplicación?

Is it easy to return to a previous state of the application?

4 4 - Consistencia y estándares / Consistency and standards

4.1 Las etiquetas de los vínculos tienen los mismos nombres que sus destinos?

Do the link labels have the same names as their destinations?

4.2 Las mismas acciones siempre conducen a los mismos resultados?

Ejemplo: Si nos encontramos en un sitio web en el cual, en una interfaz encontramos dos links con el nombre de una misma sección, cada uno de ellos debería llevar al usuario a la misma interfaz, correspondiente a esa sección.

Do the same actions always lead to the same results?

Example: If we are on a website in which, in an interface, we find two links with the name of the same section, each of them should take the user to the same interface, corresponding to that section.

4.3 Un mismo icono tiene el mismo significado en todo el sistema?

Does the same icon have the same meaning throughout the system?

4.4 La información se muestra de forma consistente en todo el sistema?

Is the information consistently displayed throughout the system?

4.5

Los colores de los enlaces son adecuados para su uso?

Are the link colors suitable for use?

4.6 Los elementos de navegación siguen los estándares? (Botones, checkbox, ...)

The navigation elements follow the standards? (Buttons, checkbox, ...)

5 "5 - Reconocimiento en lugar de memoria, aprendizaje y anticipación /

Recognition rather than memory, learning and anticipation"

5.1 Es sencillo utilizar el sistema por vez primera?

Is it easy to use the system for the first time?

5.2 Es fácil localizar información que ya ha sido buscada con anterioridad?

Is it easy to locate information that has already been searched before?

5.3 En todo momento puedes utilizar el sistema sin necesidad de recordar pantallas anteriores?

Can you use the system at any time without having to remember previous screens?

5.4 Todo el contenido necesario para la navegación o para las diferentes tareas está en la "pantalla actual"?

Is all the content needed for navigation or for the different tasks on the "current screen"?

5.5 La información está organizada según la lógica de los usuarios a los que va dirigida la interfície?

Is the information organized according to the logic of the users to whom the interface is directed?

6 6 - Flexibilidad y eficiencia de uso / Flexibility and efficiency of use

6.1 Existen atajos de teclado para las acciones frecuentes?

Are there keyboard shortcuts for frequent actions?

6.2 Si existen, ¿queda claro cómo usarlas?

If they exist, is it clear how to use them?

6.3 Al realizar cualquier acción por primera vez, queda claro cómo realizarla permanentemente?

When performing any action for the first time, is it clear how to perform it permanently?

6.4 El diseño se adapta al cambiar la resolución de la pantalla?

Does the design adapt when changing the screen resolution?

6.5 Es visible el uso de aceleradores para el usuario habitual?

Is the use of accelerators visible to the regular user?

6.6 Se mantiene siempre ocupado al usuario? (Sin tiempos de espera innecesarios)

Is the user always busy? (Without unnecessary waiting times)

7 "7 - Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errors

Help users recognize, diagnose and recover from errors"

7.1 Se muestra un mensaje antes de tomar acciones irreversibles?

Is a message displayed before taking irreversible actions?

7.2 Los errores cometidos se muestran en tiempo real?

Are the errors shown in real time?

7.3 El mensaje de error que aparece es fácilmente interpretable?

Is the error message that appears easily interpretable?

7.4 Se usa, además, algún código para referenciar el error?

Is some code also used to reference the error?

8 8 - Prevención de errores / Error prevention

8.1 Aparece un mensaje de confirmación antes de realizar las acciones?

Does a confirmation message appear before performing the actions?

8.2 Queda claro qué hay que introducir en cada campo de un formulario?

Is it clear what must be entered in each field of a form?

8.3 El motor de búsqueda tolera errores tipográficos y ortográficos?

Does the search engine tolerate typos and spelling errors?

9 9 - Diseño estético y minimalista / Aesthetic and minimalist design

9.1 Se ha usado un diseño sin redundancia de información?

Has a design without information redundancy been used?

9.2 La información textual es concisa y precisa?

Is the textual information concise and accurate?

9.3 Cada elemento de información se diferencia del resto y es inconfundible?

Each item of information differs from the rest and is unmistakable?

9.4 La estructuración del texto es correcta, con frases cortas i fácil de interpretar?

The structuring of the text is correct, with short sentences and easy to interpret?

10 10 - Ayuda y documentación / Help and documentation

10.1 Existe la opción "ayuda"?

Is there a "help" option?

10.2 En el caso de existir, es visible y de fácil acceso?

If it exists, is it visible and easily accessible?

10.3 La ayuda está orientada a la solución de problemas?

Is the help section aimed at solving problems?

10.4 Dispone de un apartado de preguntas frecuentes?

Is there a section of frequently asked questions (FAQ)?

10.5 La documentación de ayuda es clara, utiliza ejemplos?

Is the help documentation clear, uses examples?

11 11 - Guardar el estado y proteger el trabajo / Save the state and protect the work

11.1 Los usuarios pueden continuar desde un estado anterior al que quedaron en otro momento o desde otro dispositivo?

Can users continue from a previous state where they had previously been or from another device?

11.2 Se implementa la utilidad de "autoguardado"?

Is the "auto-save" utility implemented?

11.3 Tiene buena respuesta a fallos ajenos? (Cortes de corriente, de internet, ...)

Does the system have a good response to external failures? (Power cut, internet not working, ...)

12 12 - Color y legibilidad / Colour and readability

12.1 Las fuentes del texto tienen un tamaño adecuado?

Do the text fonts have an appropriate size?

12.2 Las fuentes del texto utilizan colores con suficiente contraste con el fondo?

Do the text fonts use colors with sufficient contrast with the background?

12.3 El texto con imágenes o patrones de fondo es fácil de leer?

Is the text with background images or patterns easy to read?

12.4 Se tiene en cuenta a los usuarios con visión reducida?

Are users with low vision taken into account?

Web Disability Simulator

13 13 - Autonomía / Autonomy

13.1 Se mantiene en todo momento informado al usuario del estado del sistema?

Does it keep the user informed of the system status?

13.2 Además, el estado del sistema es visible y actualizado?

Moreover, is the system status visible and updated?

13.3 El usuario puede personalizar su espacio personal o de trabajo del sistema?

Can the user customize his personal or work space of the system?

14 14 - Valores per defecto / Default values

14.1 El sistema o aparato proporciona la opción de volver a los valores de fábrica?

Does the system or device provides the option to return to factory settings?

14.2 Si es así, se indican claramente las consecuencias de dicha acción?

If so, are the consequences of such action clearly stated?

14.3 Se utiliza el término "por defecto"?

Is the term "default" used?

15 15 - Reducción de la latencia / Latency reduction

15.1 La ejecución de tareas pesadas es transparente al usuario?

Is the execution of heavy work transparent to the user?

15.2 Se muestra el tiempo restante o alguna animación de las tareas pesadas que se están ejecutando?

While heavy tasks are being executed, is it showing the remaining time or some animation?

Usability website sources: Nielsen: <https://www.nngroup.com/articles/ten-usability-heuristics/>

Structure:

Generate HTML, CSS, and JS code with accessible tags that validate the HTML, CSS, and JS code and comply with usability and accessibility aspects.

Design

Use the design system: <https://carbondesignsystem.com/>

Design colour: Include as primary colour: #1F70B8. Consider the contrast of all elements.

Legible sans-serif typography (e.g. Arial, Verdana, Open Sans).

Large buttons and sufficient separation between interactive elements.

Includes responsive features such as content organisation when the screen is small.

Change menu items to a mobile-friendly menu.

Interaction:

Include interactions with dynamic information that appears on the page.

Include content sections that are displayed dynamically.

Ensure that interactions are keyboard accessible and compatible with screen readers.

Include WAI-ARIA if necessary.

EXPECTED OUTPUT:

Submit the complete code in HTML, CSS, and JS in a single block or structured in three parts. Do not use external frameworks (such as React, Angular, etc.). Do not use external JS libraries.

FINAL INSTRUCTIONS:

Do you require further information about the course, the target audience, or the content structure to complete this task?

1.2 Step 4. Prompt improvement

OPTIMIZED PROMPT (after manual creation, review to a prompt optimization tool)

Role

You are a senior web designer specializing in accessibility and usability.

Objective

Generate the complete code (HTML, CSS, and JavaScript) for a landing page for an accessibility course, meeting the highest standards of web accessibility.

1. Accessibility requirements

1. WCAG 2.2: cover all level A and AA criteria.
2. Cognitive inclusion: apply guidelines from "Making Content Usable for People with Cognitive and Learning Disabilities."
3. WAI ARIA: label roles, states, and properties where appropriate.
4. Clear language: easy-to-read text, avoid technical jargon.
5. Keyboard navigation: full support, logical tab order, and landmarks.

2. Usability heuristics

Apply these 15 heuristics (first level), with their numbering in code comments:

1. System visibility and status
2. Connection to the real world
3. User control and freedom

4. Consistency and standards
5. Recognition vs. memory
6. Flexibility and efficiency
7. Error diagnosis and recovery
8. Error prevention
9. Aesthetic and minimalist design
10. Help and documentation
11. State saving and protection
12. Color and readability
13. Autonomy
14. Default values
15. Latency reduction

3. Visual design

- System: Carbon Design System (<https://carbondesignsystem.com/>)
- Palette:
 - o Primary: #1F70B8
 - o Secondary: choose accent color with WCAG contrast $\geq 4.5:1$
- Typography: sans serif (e.g., Open Sans), sizes defined in rem.
- Spacing: generous margins and paddings ($\geq 1\text{rem}$).
- Responsive:
 - o Desktop, tablet, and mobile (breakpoints: 1024px, 768px, 480px).
 - o Collapsible menu (hamburger) at $< 768\text{px}$.

4. Behavior and interactions

- Collapsible sections: keyboard controllable and ARIA announcement when expanding/collapsing.
- Forms: real-time validation, clear messages, and aria-live.

- Animations: smooth transitions (< 300 ms), no flickering (> 3 flashes).

5. Code delivery

HTML:

- o Semantic and validated (W3C).
- o Comments indicating section and heuristics applied.

CSS (vanilla):

- o Modular files, CSS variables for colors/spacing.
- o BEM classes or similar for organization.

JavaScript (pure):

- o ES6 modules, no external dependencies.
- o Documented accessibility events (focus, keydown).

Structure:

three root folders: /html, /css, /js.

JSON FILE WITH CONTENT (Direct JSON file)

Next step

When you're ready, send me the JSON with the content structure (sections, titles, texts, labels, CTAs, FAQs, etc.) so I can inject it into the template.

1.3 Step 5. Prompt customization.

Here you can see several prompts for each AI platform

1.3.1 ChatGPT

PROMPT:

With this content (text on Section Step 4), create an optimized prompt for ChatGPT (<https://chatgpt.com/>)

RESULT:

Objective: Generate an accessible landing page from structured JSON.

Platform: ChatGPT.

Assigned role: Senior web designer specialising in accessibility and usability.

YOUR TASK

Generate the complete code for a landing page (HTML, CSS, and pure JavaScript) for a digital accessibility course, using the content provided in JSON format.

You must comply with the following specifications for accessibility, visual design, behaviour, and interaction.

1. BASE CONTENT (JSON)

Use this JSON as content to populate the website structure:

- ◆ Navigation, modules, forms, teaching staff, etc.
- ◆ It is important to respect the key elements and their IDs for anchor navigation.

(The structured JSON, such as the one we have already generated in the last iteration, will be included here.)

2. ACCESSIBILITY CRITERIA (MANDATORY)

Comply with the following requirements when generating the code:

1. WCAG 2.2 (A and AA): ensure that you meet all current criteria.
2. Cognitive inclusion: apply guidelines from 'Making Content Usable for People with Cognitive and Learning Disabilities'.

3. WAI-ARIA: implement roles, properties, and states where necessary.
4. Clear language: easy-to-read text, without unnecessary technical terms.
5. Keyboard navigation: full support, landmarks, and logical tab order.

3. VISUAL DESIGN

- Design system: Carbon Design System
- Colours:
 - o Primary: #1F70B8
 - o Secondary: colour with WCAG contrast ratio $\geq 4.5:1$
- Typography: Sans-serif (e.g. Open Sans), in rem units
- Responsive:
 - o Breakpoints: 1024px, 768px, 480px
 - o Hamburger menu on mobiles ($< 768\text{px}$)
- Spacing: generous margins and paddings ($\geq 1\text{rem}$)

4. BEHAVIOUR AND INTERACTIONS

- Collapsible sections: accessible by keyboard, with aria-expanded and aria-controls
- Forms: real-time validation with accessible messages (aria-live)
- Animations: smooth transitions ($< 300\text{ms}$), no flickering or flashing (maximum 3)

5. USABILITY HEURISTICS

Add references to these heuristics (using their number) in code comments:

1. Visibility of system state
2. Connection to the real world

3. User control and freedom
4. Consistency and standards
5. Recognition rather than recall
6. Flexibility and efficiency
7. Error diagnosis and recovery
8. Error prevention
9. Aesthetic and minimalist design
10. Help and documentation
11. State saving and data protection
12. Colour and readability
13. Autonomy
14. Default values
15. Latency reduction

6. CODE DELIVERY (STRUCTURE)

Organise the project as follows:

sql

CopyEdit

/html --> index.html (semantic and W3C validated)

/css --> styles.css (vanilla CSS with variables and BEM)

/js --> main.js (ES6, no dependencies, accessible)

FINAL INSTRUCTIONS


- The code must be accessible, semantic, clean, and commented.
- Do not use external frameworks or libraries.
- Do not forget to include landmarks (<main>, <nav>, <header>, <footer>, etc.).
- Generate everything in a single response if possible, or in well-labelled parts if it is very long.

1.3.2 Claude

PROMPT:

With this content (text on Section Step 4), create an optimized prompt for Claude (<https://claude.ai/>)

RESULT:

 Prompt for Claude

You are a senior web designer specialising in digital accessibility and usability. Your goal is to generate the complete code (HTML, CSS, and JavaScript) for a landing page for a university course on digital accessibility, following the highest standards of accessibility and professional design.

Use this set of guidelines:

1.3.2.1 OBJECTIVE

Develop an informative and accessible landing page for a university micro-credential in digital accessibility. It should convey confidence, be clear, readable, and easy to navigate.

1.3.2.2 ACCESSIBILITY REQUIREMENTS

1. WCAG 2.2: cover all level A and AA criteria.
2. Cognition: apply guidelines from Making Content Usable for People with Cognitive and Learning Disabilities.
3. WAI-ARIA: use roles, properties, and states to improve comprehension and navigation for assistants.
4. Clear language: easy-to-read style, avoid unnecessary technical terms.
5. Keyboard navigation: logical tab order, no traps, with semantic landmarks (<header>, <main>, <nav>, <footer>, etc.).

1.3.2.3 *USABILITY HEURISTICS TO INCLUDE (as comments in the HTML):*

1. System status visibility
2. Connection to the real world
3. User control
4. Consistency
5. Recognition rather than memorisation
6. Efficiency and flexibility
7. Error diagnosis and recovery
8. Error prevention
9. Aesthetics and minimalism
10. Contextual help
11. State saving and protection
12. Colour readability
13. User autonomy
14. Default values
15. Latency reduction

1.3.2.4 *VISUAL DESIGN*

- Visual system: use Carbon Design System, complemented with a customised (modular) style sheet.
- Palette:
 - o Primary: #1F70B8
 - o Secondary: high contrast with minimum WCAG 4.5:1
- Typography: modern sans-serif ('IBM Plex Sans', 'Open Sans')
- Spacing: minimum 1rem, use generous margins.
- Responsive: breakpoints at 1024px, 768px, 480px. Hamburger menu for < 768px.

1.3.2.5 BEHAVIOUR / INTERACTIONS

- Collapsible sections accessible by keyboard (<details> or ARIA aria-expanded)
- Forms with accessible real-time validation (ARIA live, clear messages)
- Smooth transitions (< 300 ms), no flickering

1.3.2.6 CODE STRUCTURE

Generate three root folders:

bash

CopyEdit

/html → complete index.html, with heuristic comments

/css → custom styles.css compatible with Carbon

/js → main.js with accessible interactions

1.3.3 Copilot

PROMPT:


With this content (text on Section Step 4), create an optimized prompt for Copilot (<https://copilot.microsoft.com/>)


RESULT:

 PROMPT FOR COPILOT (README.md file or initial comment block)

/**

 Project: Accessible Landing Page for Digital Accessibility Course

 Objective: Generate the complete code (HTML, CSS, and JavaScript) for a landing page that meets the highest standards of accessibility and usability.

 Structure: /html, /css, /js, and a JSON content file.

1. Mandatory accessibility

- Comply with all WCAG 2.2 (level A and AA)
- Apply cognitive guidelines (Making Content Usable for People with Cognitive and Learning Disabilities)
- WAI-ARIA: roles, properties, and states where applicable
- Clear language and easy reading
- Full keyboard navigation with semantic landmarks

2. Usability heuristics (add to HTML comments):

1. System status visibility
2. Connection to the real world
3. User control
4. Consistency
5. Recognition vs. memory
6. Flexibility
7. Diagnosis and recovery
8. Error prevention
9. Aesthetics and minimalism
10. Contextual help
11. State saving
12. Colour readability
13. Autonomy
14. Default values
15. Latency reduction

3. Visual design

- Based on Carbon Design System
- Palette:

- o Primary: #1F70B8
- o Secondary: contrast $\geq 4.5:1$
- Typography: 'Open Sans', sizes in rem
- Responsive: 1024px / 768px / 480px (hamburger menu < 768px)

4. Behaviour and interactions

- Collapsible sections accessible via keyboard (aria-expanded, aria-controls)
- Forms with real-time validation and aria-live
- Smooth transitions < 300 ms
- No flickering or intense animations (> 3 flashes)

5. Structured delivery

- /html/index.html with heuristic comments
- /css/styles.css modular, with variables in :root
- /js/main.js in ES6 without frameworks
- content.json with landing page data

Copilot, please suggest:

- Semantic and accessible HTML with heuristic comments
- Vanilla CSS compatible with Carbon Design System
- Accessible JS for interactions (details, forms, focus)
- Reading the JSON file if dynamic content is desired

*/

1.3.8 Tips for using Copilot

- Open index.html, and start typing:

html

CopyEdit

```
<!-- Section 1: Header (heuristics 1, 2, 4) -->
```

```
<header class="bx--header" role="banner">
```

Copilot will complete with contextual code respecting Carbon design + accessibility.

- In CSS, start with:

css

CopyEdit

```
/* :root variables for colours and fonts according to Carbon Design  
System */
```

```
:root {
```

- In JS:

js

CopyEdit

```
// Module for accessible control of collapsible sections (heuristics 3, 6,  
8)
```

```
document.querySelectorAll("[aria-expanded]").forEach((toggle) => {
```

1.3.4 Gemini

PROMT:

With this content (text on Section Step 4), create an optimized prompt for Gemini (<https://gemini.google.com/>)

RESULT:

Role:

You are a web developer specialising in accessibility and usability. Use the following requirements to generate the ****complete and final code for an accessible landing page****, following the most demanding web standards. The structure should be divided into three folders: ``/html``, ``/css``, ``/js``.



GENERAL CONTEXT:

- Course: University Micro-credential in Digital Accessibility.
- Purpose of the page: to promote the course and provide information in an accessible manner.

- Content source: JSON file defining all sections of the course (Presentation, Academic programme, Teaching staff, Career opportunities, Entry requirements, Enrolment and price).

ACCESSIBILITY (mandatory compliance):

1. WCAG 2.2: all A and AA criteria.
2. Cognitive inclusion: apply guidelines from 'Making Content Usable for People with Cognitive and Learning Disabilities'.
3. WAI-ARIA: correctly labelled roles, properties and states.
4. Clear language: no technical terms, easy to read.
5. Keyboard navigation: logical tabulation, landmarks and visible focus.

USABILITY:

Apply these ****15 usability heuristics****, commenting in the code with their corresponding number. Example: `<!-- Heuristic 2: Connection to the real world -->`.

VISUAL DESIGN:

- System: Carbon Design System (<https://carbondesignsystem.com/>)
- Palette: primary `#1F70B8`, secondary with contrast $\geq 4.5:1$.
- Typography: sans serif (e.g. Open Sans), sizes in `rem`.
- Minimum spacing: `1rem`.
- Responsive: breakpoints 1024px, 768px, 480px.
- Collapsible hamburger menu on screens $< 768px$.

INTERACTIONS:

- Collapsible sections (keyboard + `aria-expanded`, `aria-controls`).
- Forms with real-time validation, accessible messages (`aria-live`).
- Smooth transitions ($< 300ms$), no flashes (> 3 per second).

FILE STRUCTURE:

- ``/html/index.html`` : complete semantic structure, with landmarks (``<main>``, ``<nav>``, ``<header>``, etc.).
- ``/css/styles.css`` : vanilla CSS, BEM or similar, CSS variables for colours, fonts and spacing.
- ``/js/main.js`` : modular ES6, no frameworks, accessible interactions (``keydown``, ``focus``, ``click`` events, etc.), explanatory comments.

INPUT (JSON structure of the course):

Include here the JSON file containing all the content of the page (if you have already generated it). Use the fields to dynamically generate the sections of the website with their content.

FINAL OBJECTIVE:

Generate a fully functional landing page, validated by W3C, with clean and maintainable code, following professional standards.

1.3.5 Vercel

PROMPT:


With this content (text on Section Step 4), create an optimized prompt for Vercel v0 (<https://v0.dev/>)

RESULT:

Vercel v0 Prompt for Landing Page Generator

  Role

You are a ****Senior Web Designer**** specializing in ****Accessibility**** and ****Usability****.

 Objective

Generate the **complete code** (HTML, CSS, JavaScript) for a **landing page** promoting an accessibility course. The output must be a ZIP-like project structure ready to deploy on Vercel v0.

1. Accessibility Requirements

1. **WCAG 2.2**: satisfy **ALL** Level A & AA success criteria.
2. **Cognitive Inclusion**: implement "Making Content Usable for People with Cognitive and Learning Disabilities."
3. **WAI-ARIA**: add appropriate `role`, `aria-*` attributes, and states.
4. **Clear Language**: use easy-read text; avoid jargon.
5. **Keyboard Navigation**: full support, logical `tabindex`, and semantic landmarks (`<header>`, `<main>`, `<nav>`, `<footer>`).

2. Usability Heuristics

1. Visibility & system status
2. Real-world connection & metaphors
3. User control & freedom
4. Consistency & standards
5. Recognition vs. memory
6. Flexibility & efficiency
7. Error diagnosis & recovery
8. Error prevention
9. Aesthetic & minimalist design

10. Help & documentation
11. Save state & work protection
12. Color & readability
13. Autonomy
14. Default values
15. Latency reduction

3. Visual Design

- **Design System**: Carbon Design System (<https://carbondesignsystem.com/>)
- **Color Palette**
 - Primary: `#1F70B8`
 - Accent: choose one with **WCAG $\geq 4.5:1$** contrast.
- **Typography**: sans-serif (e.g. Open Sans), sizes in `rem`.
- **Spacing**: margin/padding $\geq 1\text{rem}$.
- **Responsive Breakpoints**
 - Desktop: $\geq 1024\text{ px}$
 - Tablet: $768\text{--}1023\text{ px}$
 - Mobile: $< 768\text{ px}$ (hamburger menu)

4. Interaction & Behavior

- **Collapsible Sections**: keyboard-toggable, announce state changes with `aria-expanded`.
- **Forms**: real-time validation, clear error messages, `aria-live` regions.

5. Code Delivery

- **Project Structure** (root):

```
/html
index.html
/css
variables.css
main.css
/js
ui-interactions.js
```

- **HTML**
- Semantic, W3C-valid.
- Comments marking **sections** and **heuristic tags**.
- **CSS** (vanilla)
- Modular files, custom properties (`--color-primary`, `--spacing-base`).
- BEM or similar naming.
- **JavaScript** (plain ES6 modules)
- No external dependencies.
- Document focus, keydown handlers, and ARIA updates.

DOCUMENT JSON WITH THE CONTENT

(fichero json)

1.3.6 Builder.io

PROMT:

With this content (text on Section Step 4), create an optimized prompt for Builder.io (<https://www.builder.io/>)

RESULT:

Create a responsive, accessible landing page in Spanish for a university microcredential program.

Use semantic HTML and Tailwind CSS. Apply WCAG 2.2 accessibility practices and include ARIA where needed. Follow logical heading order, keyboard navigation, and contrast compliance.

Structure the page as follows:

1. `<html lang="es">` and set the page title as "Microcredencial en Accesibilidad Digital"

2. Header:

- Include a logo (logo.jpg)
- Course title as an `<h1>`: "Microcredencial Universitaria en Especialización en Accesibilidad Digital por la Universitat de Lleida"

3. Navigation:

- Horizontal navigation bar with links: Presentación, Profesorado, Programa, Matrícula, FAQ

4. Main section:

- Intro paragraph describing the course purpose and audience
- Three course features in cards or columns:
 - Reconocimiento oficial
 - Aprendizaje práctico con casos reales
 - Profesorado experto en accesibilidad
- Call to action button: "Inscribirse ahora"

5. Accessibility best practices:

- Use alt texts for images

- Ensure headings are sequential (h1 > h2 > h3...)
- Ensure contrast ratios comply with 1.4.3 AA
- Use focus outlines and keyboard navigation (2.4.7 AA)
- Add hover/focus feedback for links (1.4.13*)

6. Footer:

- Contact email and phone
- Logos of Universitat de Lleida
- Links: política de privacidad, política de accesibilidad
- Set role="contentinfo" on the footer

Design: clean, modern, and professional. Use accessible color contrast and readable typography.

DOCUMENT JSON WITH THE CONTENT

(fichero json)

1.3.7 DEEPSITE.AI

PROMPT:

With this content (text on Section Step 4), create an optimized prompt for DeepSite.AI (<https://deepsite.ai/>)

RESULT:

1.3.7.1 OPTIMIZED PROMPT FOR **DEEPSITE.AI**

Role:

You are a senior web designer specialized in accessibility and usability.

Goal:

Generate a complete landing page (HTML, CSS, JavaScript) for a **Digital Accessibility**

Training Course, following the **highest accessibility and usability standards**, using modern, semantic, and modular front-end practices.

1.3.7.2 Accessibility Requirements

Ensure **full compliance** with:

1. **WCAG 2.2 (Level A and AA)** – all success criteria must be addressed.
 2. **Cognitive accessibility** based on “*Making Content Usable for People with Cognitive and Learning Disabilities*”.
 3. **WAI-ARIA**: add relevant roles, states, and properties for screen reader support.
 4. **Plain language**: use clear, simple text with no technical jargon.
 5. **Keyboard navigation**: logical tab order, all elements reachable by keyboard, with semantic landmarks (<main>, <nav>, <header>, etc.).
-

1.3.7.3 Usability Heuristics (annotate in code comments)

Implement the following 15 usability heuristics, referencing each with numbered comments:

1. System status visibility
 2. Match between system and real world
 3. User control and freedom
 4. Consistency and standards
 5. Recognition rather than recall
 6. Flexibility and efficiency
 7. Error diagnosis and recovery
 8. Error prevention
 9. Aesthetic and minimalist design
 10. Help and documentation
 11. Save state and user safety
 12. Color contrast and legibility
 13. User autonomy
 14. Smart defaults
 15. Low latency
-

1.3.7.4 Visual Design Guidelines

- **Design System:** [Carbon Design System](#)
- **Color palette:**
 - Primary: #1F70B8
 - Secondary: custom accent color with **contrast ratio $\geq 4.5:1$ (WCAG compliant)**
- **Typography:** Open Sans or similar sans-serif, sizes in `rem`

- **Spacing:** padding/margin $\geq 1\text{rem}$
 - **Responsive breakpoints:**
 - >1024px (desktop)
 - 768px-1024px (tablet)
 - <768px (mobile, hamburger menu enabled)
-

1.3.7.5 Interactions & Behavior

- **Collapsible sections:**
 - Fully keyboard-accessible
 - Use `aria-expanded`, `aria-controls`, and `aria-live` as appropriate
 - **Form validation:**
 - Real-time, accessible feedback with clear instructions
 - Use `aria-invalid`, `aria-describedby`, `aria-live`
 - **Animations:**
 - Transitions must be under 300ms
 - Avoid flashing content (>3 flashes/sec)
-

1.3.7.6 Output Structure

Generate a working project with the following root folders:

- `/html`: semantic HTML5 with **W3C validation**
 - `/css`: vanilla CSS, modular, with CSS variables (colors, spacing) and BEM-style class names
 - `/js`: native JavaScript (ES6 modules), no external dependencies, accessibility-focused events documented
-

DOCUMENT JSON WITH THE CONTENT

(fichero json)

1.3.8 BOLT

PROMPT:

With this content (text on Section Step 4), create an optimized prompt for Bolt.new (<https://bolt.new/>)

RESULT:

⚡ OPTIMIZED PROMPT FOR [BOLT.NEW](#)

Role:

You are a senior web designer specializing in **web accessibility** and **usability**.

Goal:

Generate a complete and accessible **landing page** for a digital accessibility course. The page must follow **WCAG 2.2 (Level A and AA)**, support **cognitive accessibility**, follow **WAI-ARIA best practices**, and apply **usability heuristics** within modern responsive design.

✅ Accessibility Requirements

Implement:

1. **WCAG 2.2 (Level A & AA)**: all success criteria must be respected
 2. **Cognitive accessibility**: apply *Making Content Usable for People with Cognitive and Learning Disabilities*
 3. **WAI-ARIA** roles, states, and properties where appropriate
 4. **Plain language**: write in clear, easy-to-read language with no technical jargon
 5. **Full keyboard support** with logical tab order and semantic landmarks (`<header>`, `<nav>`, `<main>`, etc.)
-

💡 Usability Heuristics

Apply the following **15 usability heuristics** and annotate them in code comments using numbered references (e.g. `// Heuristic 3: User control and freedom`):

1. System status visibility
2. Match between system and real world
3. User control and freedom
4. Consistency and standards
5. Recognition rather than recall

6. Flexibility and efficiency
 7. Error diagnosis and recovery
 8. Error prevention
 9. Aesthetic and minimalist design
 10. Help and documentation
 11. Save state and user protection
 12. Color and readability
 13. User autonomy
 14. Useful defaults
 15. Low latency interaction
-

Visual Design

- **Design system:** [Carbon Design System](#)
 - **Colors:**
 - Primary: #1F70B8
 - Secondary: any accent color with contrast ratio $\geq 4.5:1$ (WCAG compliant)
 - **Typography:** Open Sans or similar sans-serif font, sizes in `rem`
 - **Spacing:** generous spacing ($\geq 1\text{rem}$)
 - **Responsive design:**
 - Breakpoints: 1024px, 768px, 480px
 - Collapsible hamburger menu below 768px
-

Interactions & Behavior

- **Collapsible sections:** keyboard operable, `aria-expanded`, `aria-controls`
 - **Forms:**
 - Real-time validation with `aria-live` announcements
 - Clear labeling and error messages
 - **Animations:**
 - Smooth transitions under 300ms
 - No flashing or strobe effects (avoid >3 flashes/sec)
-

Code Delivery Guidelines

Project structure (three folders):

- `/html`: semantic HTML, W3C validated, with comments for heuristics
- `/css`: vanilla modular CSS with BEM-style class names and CSS variables
- `/js`: pure ES6 JavaScript, modular, no external dependencies, with documented accessibility event handlers (`focus`, `keydown`, etc.)

DOCUMENT JSON WITH THE CONTENT

(fichero json)