

Human Computer-Interaction

HTML

Web design

- When designing a website, we distinguish between:
 - Content: Everything that represents **information** within a website, as well as its structure and organisation.
 - Presentation: Everything related to the visual aspect of the website.

HTML

- HiperText Markup Language.
 - Set of rules for writing a web page that can be understood by any web browser
 - Language used to represent the **CONTENT** of Web pages.
- An International recognised standard which rules are defined by an organisation called W3C (*World Wide Web Consortium*).
 - W3C standards are followed by companies, publishing organisations, browser developers..



HTML

- Started as the Web language in 1991:
 - It was useful in academic settings
 - Little importance to the graphic appearance, it was mainly about presenting some information
- The popularisation of the Internet created the need to introduce more presentation elements in HTML
 - Mixing presentation elements and content in the same document
 - Source code files that were too long
 - Websites were difficult to maintain

HTML

- CSS (Cascade Style Sheets) is born in 1996 to make code separation possible
 - This separation is now mandatory
 - Presentation-related HTML tags are maintained for backwards compatibility only and must NEVER be used.
- Current HTML version
 - HTML 5.3 (January 2021) is the latest version

HTML 5

- Current version of HTML
 - It was first launched in 2014 but is still under continuous development.
 - It gives name to the current web standard, which includes HTML 5, CSS3 and JavaScript.
 - W3C decided to exclude from this version external technologies that were in great demand in the 1990s (Flash, Silverlight, etc.).
 - As alternatives new CSS and JavaScript functionalities were proposed.

CSS

- *Cascade Style Sheet*
 - It is a complement to HTML
- CSS defines the **PRESENTATION**, that is, the format and appearance of a website
- Versions
 - CSS I
 - CSS II
 - **CCS3** (adapted to HTML 5.0)

W3C Standards

- Both HTML and CSS are standards, defined by the W3C
 - <http://www.w3.org>
 - Sets web guidelines to be followed by
 - Web browsers and ...
 - ...other Web tools.
 - As these are standards, the interpretation may be different depending on the browser used
 - Always test websites with several different browsers and devices.

Web development tools

- Web editors
 - An .html document is a simple text file that is interpreted by a web browser.
 - It can be opened and edited from any simple editor, such as Notepad++
 - If we want to structure and organise projects
 - Aptana Studio, BlueGriffon, Aloha Editor, Maquette...
 - WebStorm: <https://www.jetbrains.com/webstorm/>
- W3Schools Tutorials
 - <http://www.w3schools.com/>

Tools

- Web browsers
 - Chrome, Firefox, Opera, etc...
- To verificate if your browser supports HTML 5.0
<http://html5test.com/>
- Check tool:
 - [HTML 5 Test](#) | [Find Me By IP](#) | [Can I Use](#)

HTML elements

- An HTML element consists of:
 - **Opening tag + Contents + Closing tag**
`<p>Content</p>`
 - **One standalone tag**
 - `<hr>` : Separates two conceptually different texts (a horizontal line will be displayed)
- Note: In versions prior to HTML5, "self-closing" was required in these tags.: <hr/>*
- Some tags have **attributes**, whose values are included between **double quotes**
 - ``

Attributes in HTML tags

- Most **attribute** values are included between **double quotes**
 -
- However, boolean attributes are just included or not
 - If they are present → true
 - If they are not part of the tag → false

<input type="checkbox" **checked**>

Note: In versions prior to HTML5, boolean parameters were given their own name as a value between double quotes to define them as true:

<input type="checkbox" checked="checked">

HTML page structure

```
<!DOCTYPE html>
```

Specifies the type of document and rules

```
<html lang="es">
```

Start of the web page

```
  <head>
```

```
    <meta charset = "utf-8">
```

Character
encoding

Heading

Title, authors,
styles... (hidden information)

```
    <title>Página simple</title>
```

```
  </head>
```

```
<body>
```

```
  <h1>Página simple</h1>
```

Body

```
  <p>Esto es un
```

Page content
(visible information)

```
    <a href="demo.html">ejemplo</a> simple.</p>
```

```
    <!-- this is a comment -->
```

```
  </body>
```

```
</html>
```

End of the web page

Default visualisation

- Without CSS, the browser applies very basic styling to the HTML.
 - When we create HTML, we should not think about presentation

Página simple

Esto es un ejemplo simple.

HTML Semantic elements

- HTML elements are used to mark parts of a website: Headings, paragraphs, lists...
- With the exception of two specific elements (DIV and SPAN), all other HTML tags are used to create a semantic structure.
- We should **use elements for its semantic meaning and not for the visual appearance.**
- Example: Typically, a heading 1 <h1> will be bigger than a heading 2 <h2>, but we should never use h2 before h1, as headings represent a tree information structure

Basic HTML elements

- Paragraphs
 - `<p> </p>`
- Headings
 - `<h1> </h1>, ..., <h6> </h6>`
 - **Numeric order must be followed**
- Content separators
 - `
`: Line break
 - `<hr>`: Horizontal line
- HTML comments
 - `<!-- Comment on an HTML web page -->`

HTML Head

HTML Head

- 01 `<title>`
- 02 `<style>`
- 03 `<script>`



- 04 `<link>`
- 05 `<meta>`
- 06 `<base>`

HTML Head <head></head> (I)

Content inside the header is not displayed, but it provides useful information, such as <**title**> </**title**>

- Defines the title of the page (required in HTML 5.0)
- Displayed in the title bar of the browser (tab), also when bookmarking the page and for Internet search engines..
- <**meta**> </**meta**>

- Describes metadata

```
<head>
```

```
    <title> My First Website </title>
```

```
    <meta charset="UTF-8"> <!-- character encoding -->
```

```
</head>
```

HTML Head <head></head> (II)

- Examples of optional metadata with attributes **name** and **content**:
 - <meta **name**= "author" **content**= "Otfried Preußler" >
 - <meta **name**= "description" **content**= "The Robber Hotzenplotz" >
 - <meta **name**= "keywords" **content**= "book, children, german" >
 - <link>
 - Links to external resources, usually style sheets
- ```
<head>
 <title> My First Website </title>
 <meta charset="UTF-8">
 <meta name="author" content="Victor Alvarez">
 <link rel="stylesheet" href="styles.css">
</head>
```

# HTML Head <head></head> (III)

- **<style> </style>**

- Alternative to **<link>**. It describes specific styles inside the header:

```
<head>
 <style>
 h1 {color:red;}
 p {color:blue;}
 </style>
</head>
```

- **<link>** and **<style>** will be used in next lab session:  
Cascade Style Sheets **CSS**

# HTML Head <head></head> (IV)

- **<base>**

- Specify a default URL and a default target (\_blank, \_self) for all links on a page:

```
<head>
```

```
 base href="http://www.mywebsite.es/" target="_self">
```

```
</head>
```

This means urls in the body of the page will refer to www.mywebsite.es and they will open on the same tab

```
<body>
```

```
 Main Page in English
```

```
</body>
```

This link <a href=> opens the URL www.mywebsite.es/en\_GB/index.html

# HTML Head <head></head> (V)

- **<script> </script>**
  - Contains instructions written in a programming language, such as JavaScript:

```
<head>
 <script src="myscripts.js"></script>
</head>
```

This means the webpage will use the JavaScript code included in the external file myscripts.js

# HTML Body

# <body>

# HTML <body></body> structure (I)

- HTML 5 includes several tags that allow us to define the structure of the document.
- These tags do not have a default visualisation in the browser.
- They enable us to:
  - Define blocks
  - Assign roles to blocks
    - Each tag gives meaning to its block

# HTML <body></body> structure (II)

All these tags have **opening** and **closing**

- **<header></header>**
  - Define headers (page headers, article headers, section headers...)
- **<footer></footer>**
  - Define footers (page, article, section...)
- **<nav></nav>**
  - Define areas with navigation links between pages of our website
- **<section></section>**
  - Define large sections, sub-sections, etc..

# HTML <body></body> structure (III)

All these tags have **opening** and **closing**

- **<article></article>**
  - Define a block with an article format (a news item, a blog post, etc.)
- **<aside></aside>**
  - Defines content that is related to the page, but is separate from the main content.
- **<mark></mark>**
  - Highlights parts of a text
  - By default, yellow background and black characters

# HTML <body></body> structure (IV)

All these tags have **opening** and **closing**

- **<figure></figure>**
  - Defines multimedia content (images, video, audio)
  - Inside you can use **<figcaption></figcaption>** to specify the figure caption, and thus being linked
- **<time></time>**
  - Assigns the actual date we refer to in the content, using the parameter **datetime**.

# HTML Tags

## Tags in HTML



01

**Head tags**

02

**Text-formatting  
tags**

03

**Link tags**

04

**List tags**

05

**Table tags**

06

**Form tags**

07

**Scripting tags**

08

**Image & Object  
tags**



# Generic block markup:

## The `<div></div>` tag

If you wish to define a block that does not match the previous semantic tags, use `<div></div>`

- The attributes ***id*** and ***class*** allow to identify them in the CSS
    - These attributes are common to all elements of the body.

```
<div id="Chapter13" class="Book">Chapter 13 on a book</div>
```
  - Before HTML5, `<div>` was the common section separator, used for almost anything
- `<div id="header"> → <header>`

# Text tags (I) : Emphasising information

## Highlight and *emphasise*

- **<strong> </strong>**
  - Highlight (Default style shows it in bold)
- **<em> </em>**
  - Emphasise (Default style shows it in italics)

Note: Early versions of HTML included the following formatting tags:

- **<b> (bold) </b>**
- **<i> (italic) </i>**

These tags should not be used under any circumstances as they refer to the design and not to the type of information.

# Text tags (II):

## Types of information

All these tags have opening and closing

- **<cite></cite>**
  - title of a work,  
play..
- **<code></code>**
  - programming code
- **<kbd></kbd>**
  - keyboard input
- **<var></var>**
  - variable
- **<samp></samp>**
  - examples

# Text tags (III): Additional information

All these tags have opening and closing

- **<dfn></dfn>** definition.
  - **<abbr></abbr>** abbreviation.
  - **<acronym></acronym>** acronym.
- 
- Any of these tags uses the **title** parameter to complete the information.  
*<acronym title=“University of Oviedo”> UniOvi </acronym>*

# CSS Style external – inline - embedded

There are three ways to include CSS in HTML:

- Define styles in an **external** file
  - Enable different pages to access the same CSS
  - There are linked in the HTML head
- **Inline** styles in the head of a web page
  - CSS styles specific for the web page
  - CSS rules specified in the HTML head
- Embedded styles as attributes of HTML elements
  - Own style for a single item
  - Using the attribute *style*

# *Style* attribute

- This attribute is common to all HTML elements and it is used to apply CSS styles to the element
- Mainly for testing purposes. It is really no recommended using it in the final website code
  - Each rule is separated with ;
  - Between attribute and value :
  - Example:

```
<p style="color: rgb(200, 20, 20); font-family: Arial,sans-serif; font-style: italic; text-align: center;">
```

This is a paragraph with style

```
</p>
```

*This is a paragraph with style*

# Generic text mark-up:

## <span> tag

- <span> allows marking any item that cannot be marked with the other defined tags.
- This tag allows us to apply CSS styling to text fragments that do not semantically match any other HTML tag.
- As with the HTML elements <div>, attributes ***id*** and ***class*** are of particular relevance.

# HTML links

- `<a></a>` using the attribute ***href*** to specify the URL
- Example:
  - `<p> This is a link to <a href=“http://www.google.es”> Google </a></p>`

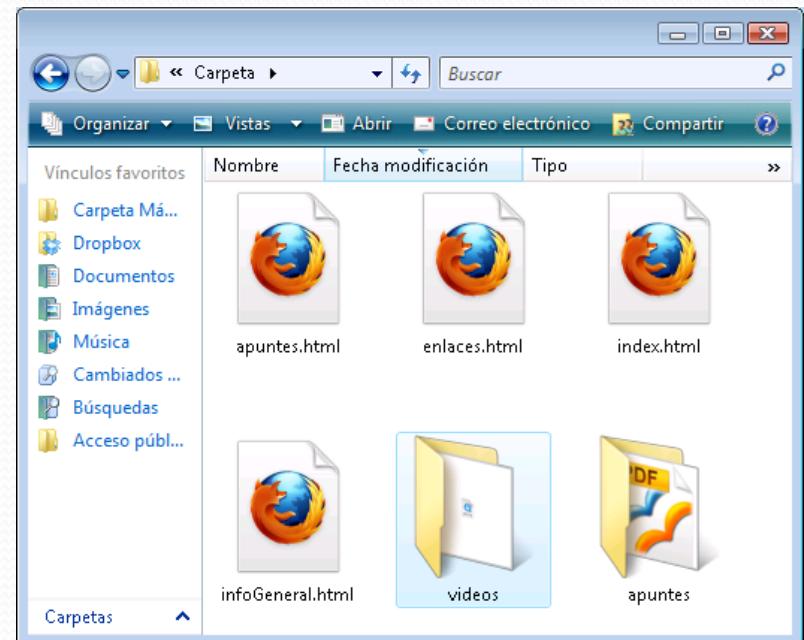
This is a link to [Google](http://www.google.es)

# Link types

- Depending on the page location:
  - **External** links: They point to pages located on other servers.
  - **Internal** links: They point to a page located on the same server.
- According to the link format:
  - **Absolute** link: They contain a full URL, which includes all the information needed to find a particular resource, typically on a different server.
  - **Relative** link: They only include the part of the information needed to access a resource, commonly on the same website.

# Link examples

- Absolute and external links
  - <a href=<http://www.google.es>> Google </a>
- Relative and internal links
  - <a href= [mynotes.html](#) >  
Notes </a>
  - <a href= [videos/video.avi](#) >  
Video </a>
  - <a href= [../index.html](#)>  
Home Page </a>



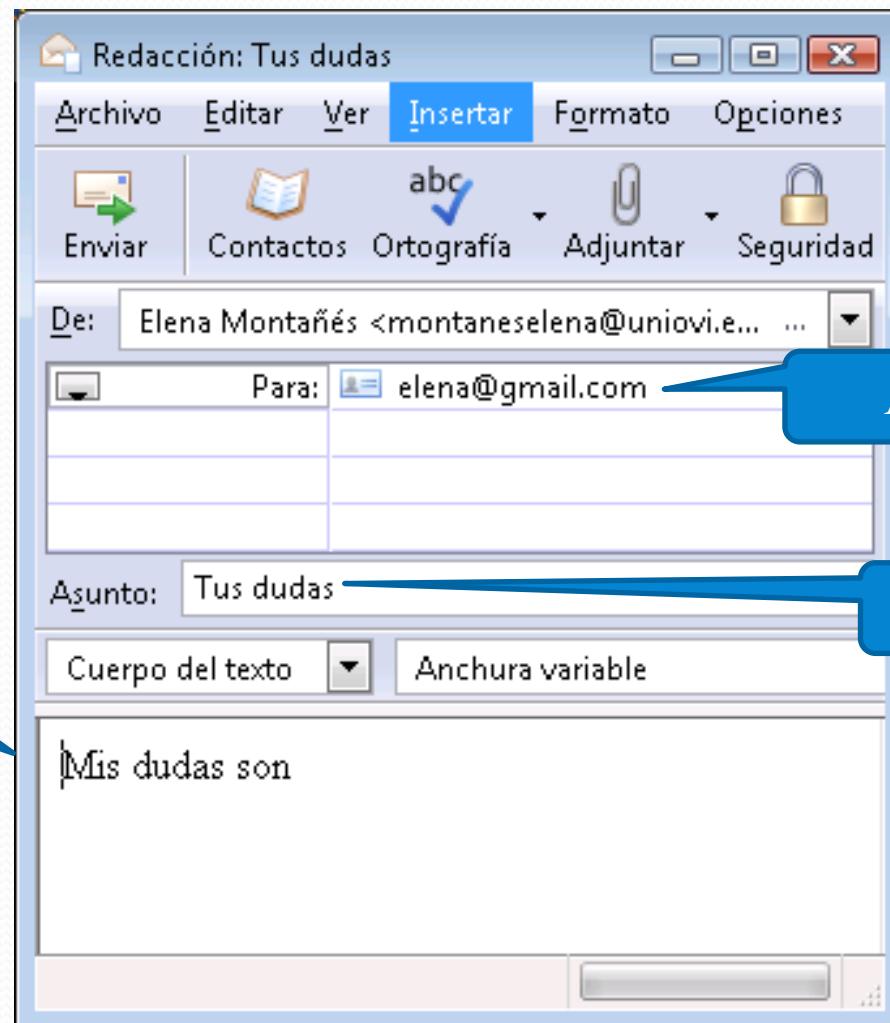
# Link target value

- Attribute ***target***

```
<a href=http://www.google.es target="_blank"Google
```

- ***Target values***
  - **\_blank** (open in new tab or new window)
  - **\_self** (open in the same window, default value)
  - **\_parent** y **\_top** (open in the parent frame or in the main space of the page)

# Email (I)



Body (Text)

Addressee (To:)

Subject

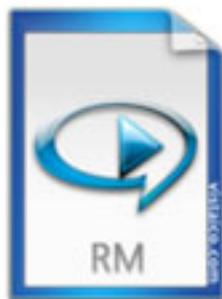
# Email (II)

- We use a call to a service **mailto**
- We can fill in attributes, including *cc* and *bcc*

```
 Send email
```

- **?** Separates the addressee from other email attributes
- **&** separates each attribute
- **%20** unicode for blank space
- **%0A** unicode for line break

# HTML Multimedia



# Pictures (I)

- Images that are part of the information
  - Decorative pictures → CSS
- Typically in the form of **bitmaps** (pixels)
  - **png** (for simple images, backgrounds)
  - **jpg** (more complicated photos and images)
  - **gif** (as a png, but they can have movement)
- **<img>** tag, attributes: **src** and **alt**
  - **<img src=“mypictures.png” alt=“my picture”/>**

**There is no closing tag!**

The value of src can be a link to an image on a web page

# Pictures (II)

- Attributes: **width** and **height**
  - Number of pixels (default value)
    - ``
  - Percentage (from original size)
    - ``  
**Use only one to keep aspect ratio!**
- When possible, do not change the original size
  - Bigger size → Quality lost
  - Smaller size → Storage takes more than needed
  - Solution: Use an editing tool to rescale image

# Videos (I)

- Different formats
  - MPEG4 (.mp4)
  - Flash (.flv o .f4v)
  - Avi (.avi)
  - WebM (.webm)
  - Ogg Vorbis (.ogg)
- **<video>** tag, attribute: **src**
  - **<video src="videos/myvideo.webm"></video>**
- Also, attributes: **width** and **height**

# Videos (II)

- If any format fails (very common given the heterogeneity), alternatives can be put in place:
  - <video>
  - <**source** src="videos/myvideo.mp4" type="video/mp4" />
  - <**source** src="video/myvideo.ogg" type="video/ogg" />
  - <**source** src="video/myvideo.webm" type="video/webm" />

Sorry, your browser doesn't support embedded videos.

- </video>

# Videos (III)

- Additional attributes:
  - **Controls** (controls for starting and stopping)
  - **Autoplay** (starts as soon as it is loaded)
  - **Preload** (loads when the page is loaded)
  - **Loop**
  - **Muted**

*Simply indicate the parameter, no values are required*

# Audio

- Different formats
  - .mp3
  - .wav
  - .ogg
- `<audio>` tag, attribute: `src`
  - `<audio src="audio/myaudio.mp3"></audio>`
- Also, `controls` and `autoplay`

# HTML Lists

## HTML Lists

### Unordered List

We have books on a variety of languages.

- Oracle
- Java
- HTML
- CSS

### Ordered List

Follow these steps to install Java:

1. Download the software.
2. Extract to a folder.
3. Double click the executable to run.

# Lists (I)

- Each list item <li></li>
- Bullet points <ul></ul>
  - <ul>  
<li>HCl</li>  
<li>INM</li>  
</ul>
  - Numbered list <ol></ol>
    - <ol>  
<li>HCl</li>  
<li>INM</li>  
</ol>
    - CPM
    - MIN

# Lists (II)

- Definition list

- **<dl>**

- <dt>HCI</dt>**

- <dd>Human-Computer Interaction</dd>**

- <dt>MIN</dt>**

- <dd>Multimodal Interfaces</dd>**

- </dl>**

HCI

Human-Computer Interaction

TPP

Multimodal Interfaces

# Lists (III)

- **style** attributes obtained from CSS ***list-style-type***, ***list-style-position*** and ***list-style-image***
  - <li style="**list-style-type**: square; **list-style-position**: inside;">HCI</li>
  - <li style="**list-style-image**: url(myimagen.gif); **list-style-position**: outside;">TPP</li>

**Style attributes can also be applied to <ul> y <ol>**

# HTML Tables



## Create Tables in HTML

| Column 1     | Column 2     | Column 3     |
|--------------|--------------|--------------|
| Row 1 Cell 1 | Row 1 Cell 2 | Row 1 Cell 3 |
|              | Row 2 Cell 2 | Row 2 Cell 3 |
| Row 3 Cell 1 |              |              |

# Tables (I): Neither header nor footer

- `<table></table>, <tr></tr> and <td></td>`

`<table border="2">`

`<tr>`

`<td>HCl</td>`

`<td>MIN</td>`

`</tr>`

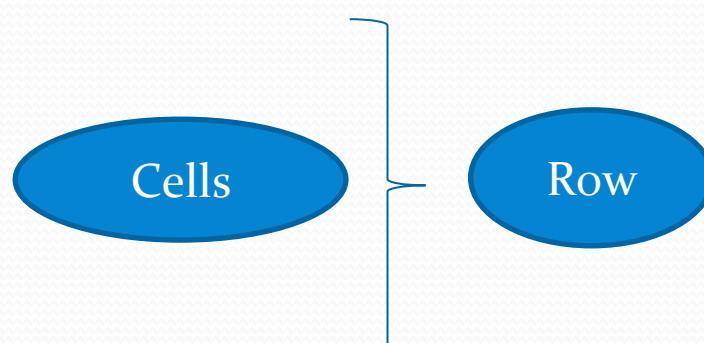
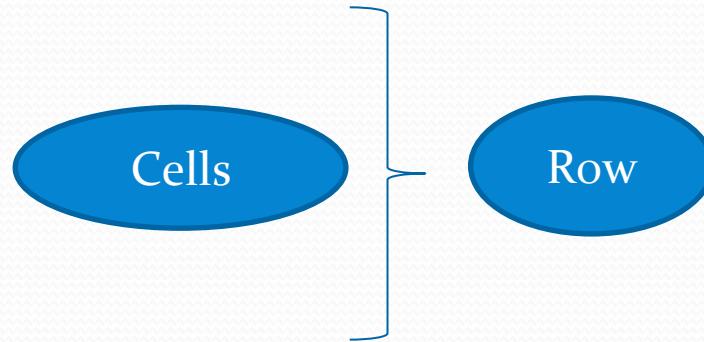
`<tr>`

`<td>VR</td>`

`<td>AR</td>`

`</tr>`

`</table>`



# Tables (II): Title and column format

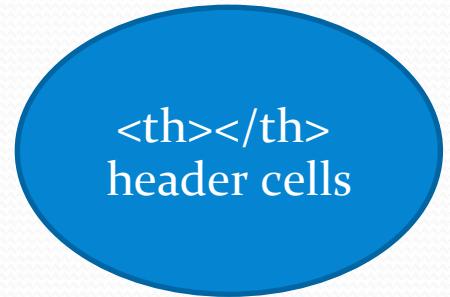
- **<caption></caption>** for the title
  - <table border="2"><caption>Courses</caption>  
...  
</table>
- **<colgroup></colgroup>** to format a number of columns (**span** indicates the number)
  - <table border="2">  
**<colgroup style="background-color: rgb(20, 20, 20);"**  
**span="1"></colgroup >**  
...  
</table>

# Tables (III): With header and footer

- Header **<thead></thead>**, body **<tbody></tbody>** and footer **<tfoot></tfoot>**

```
<table border="2">
<thead>
<tr><th>1st semester</th><th>2nd semester <--
 </th></tr></thead>

<tbody>
<tr><td>HCI</td><td>MIN</td></tr>
<tr><td>VR</td><td>AR</td></tr>
</tbody>
<tfoot><tr><td>2</td><td>2</td></tr></tfoot>
</table>
```



<th></th>  
header cells

# Tables (IV): Combine columns

- **colspan** to combine (group) columns

- ```
<table border="1">
  <tbody>
    <tr> <td colspan="3" style="text-align: center;">Courses</td> </tr>
    <tr><td> HCI </td> <td> MIN </td> <td> VR </td> </tr>
  </tbody>
</table>
```

| Courses | | |
|---------|-----|----|
| HCI | MIN | VR |

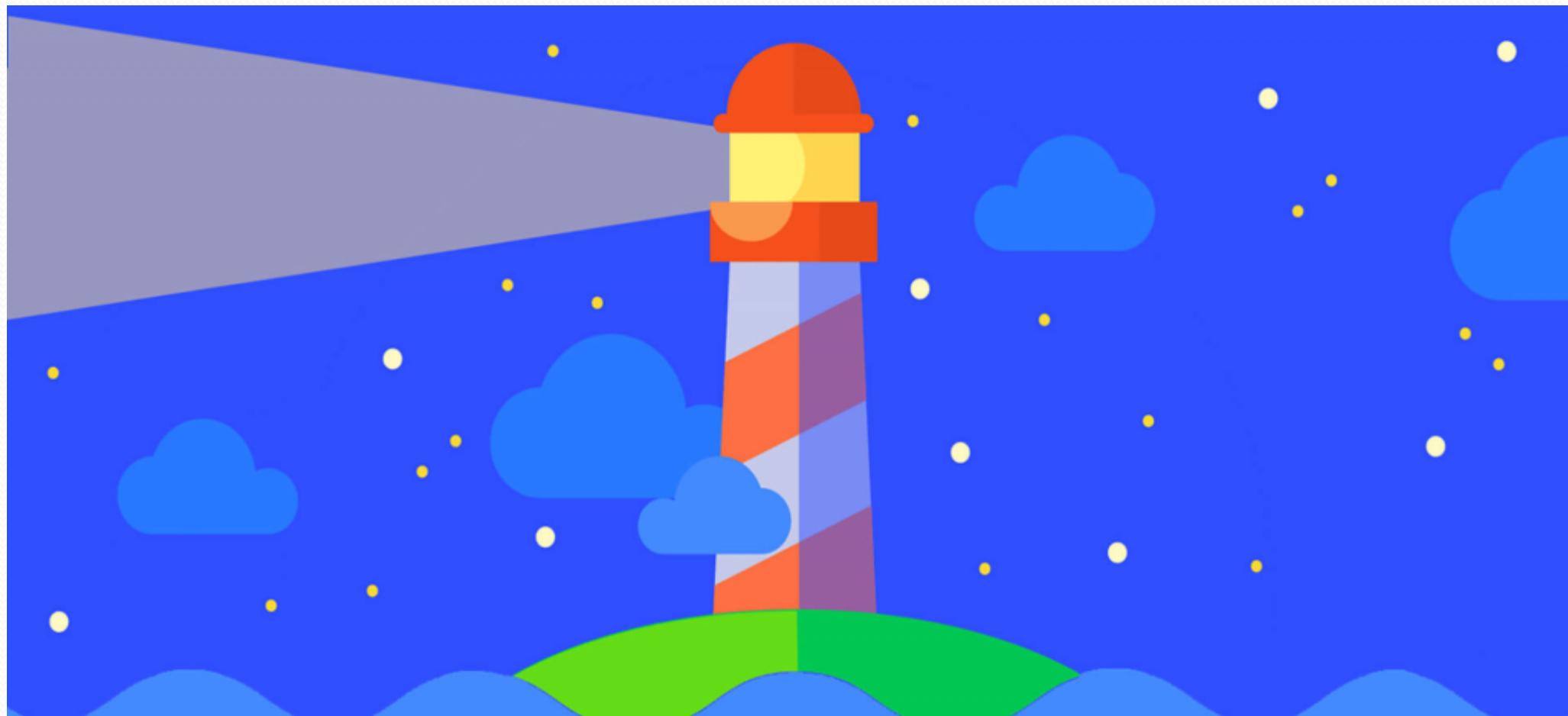
Tables (V): Combine rows

- **rowspan** to combine rows

- ```
<table border="2">
<tbody>
<tr> <td rowspan="3"> Courses </td>
<td> HCI </td> </tr>
<tr><td> MIN </td> </tr>
<tr><td> VR </td> </tr>
</tbody>
</table>
```

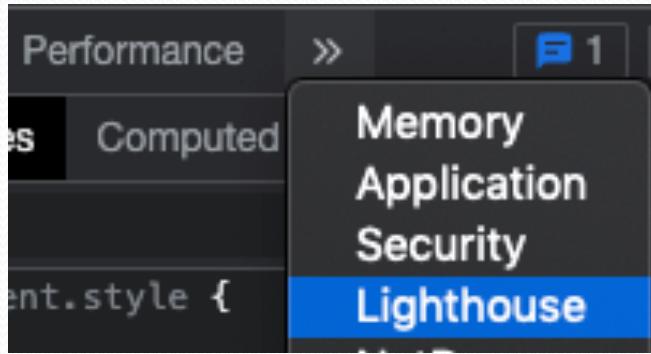
|         |     |
|---------|-----|
| Courses | HCI |
|         | MIN |
|         | VR  |

# Website accessibility



# Accessibility report with lighthouse

- **Google Chrome** right click and choose Inspect  
Expand the arrow to choose the lighthouse tool:



Generate accessibility report for Desktop or mobile:

The Lighthouse interface has a dark theme. On the left, there's a lighthouse icon and a button labeled "Generate report". Below that, a paragraph encourages users to identify common problems like performance, accessibility, and user experience. To the right, there are two columns of checkboxes under "Categories":

- Performance (unchecked)
- Progressive Web App (unchecked)
- Best practices (unchecked)
- Accessibility** (checked)
- SEO (unchecked)

Below these categories, there are sections for "Community Plugins (beta)" and "Publisher Ads", each with an unchecked checkbox. At the bottom, there's a "Learn more" link.

Categories

- Performance
- Progressive Web App
- Best practices
- Accessibility
- SEO

Device

- Mobile
- Desktop

Community Plugins (beta)

Publisher Ads

[Learn more](#)

The report shows a score of 84. The "Accessibility" section includes a note about opportunities to improve accessibility. Below it, the "NAMES AND LABELS" section lists an issue: "Form elements do not have associated labels". Further down, the "CONTRAST" section lists another issue: "Background and foreground colors do not have a sufficient contrast ratio".

84

Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

NAMES AND LABELS

Form elements do not have associated labels

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

Background and foreground colors do not have a sufficient contrast ratio.