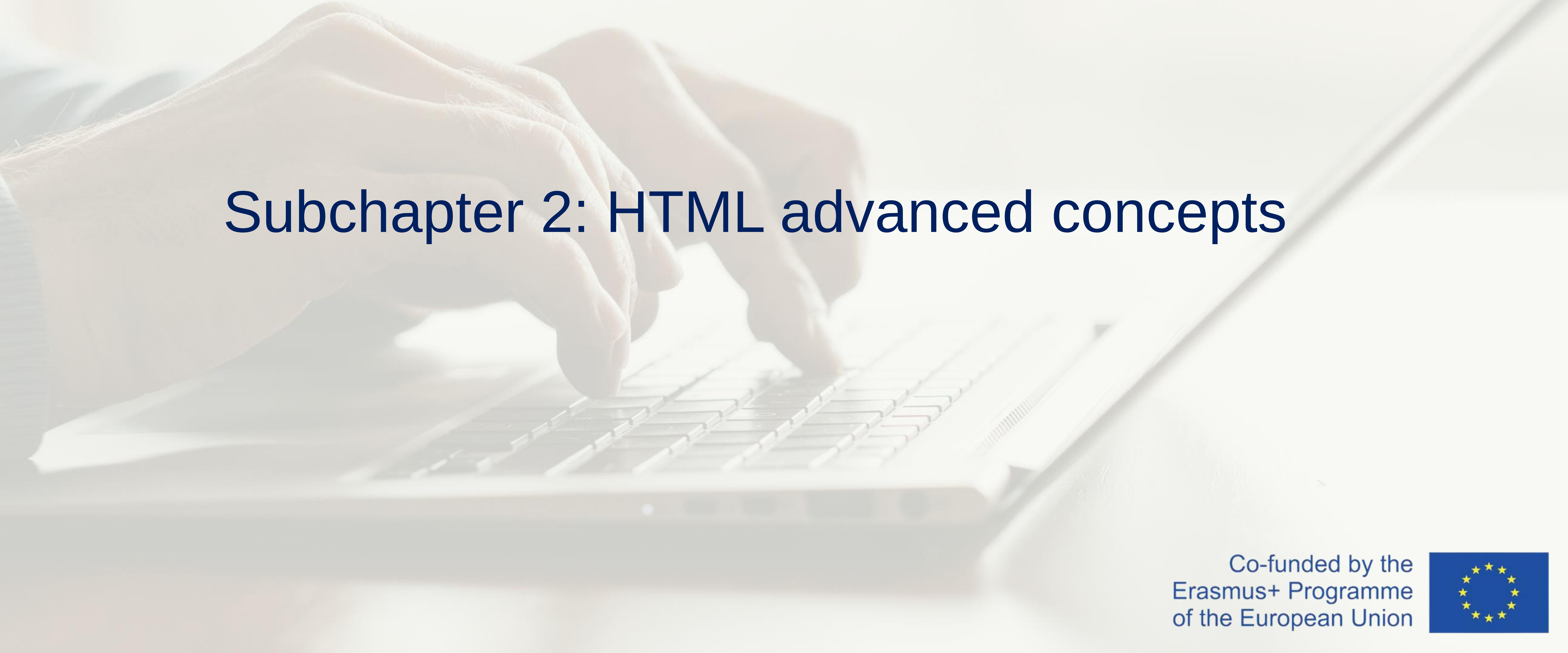
Co-funded by the Erasmus+ Programme of the European Union

code4sp
coding for social promotion

HTML Trainer Materials
Subchapter 2 – HTML advanced concepts

WP3: Code4SP Training Materials

Prepared by:



Subchapter 2: HTML advanced concepts

Co-funded by the
Erasmus+ Programme
of the European Union

HTML Doctypes

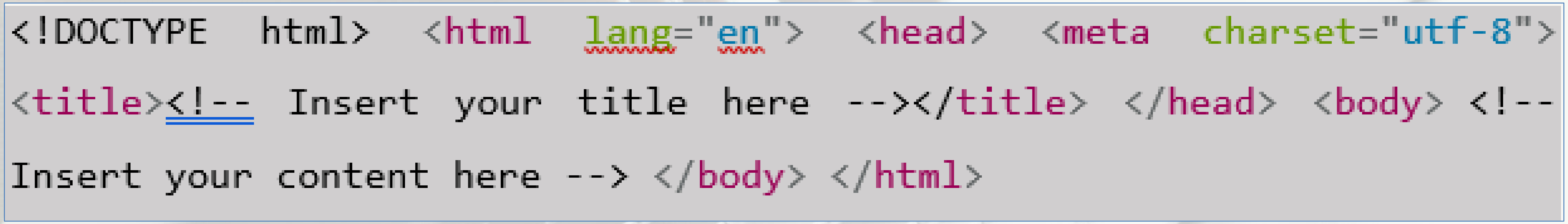
- A Document Type Declaration (DOCTYPE) is an instruction to the web browser concerning the version of markup language in which a web page is created. It appears at the top of a web page before all other elements.
- According to the HTML specification, every HTML document requires a valid document type declaration to ensure that web pages are displayed the way they are meant to be.
- The doctype declaration is frequently the first thing defined in an HTML document (even before the opening <html> tag); nevertheless, the doctype declaration itself is not an HTML tag.

Co-funded by the Erasmus+ Programme of the European Union

HTML Doctypes

The DOCTYPE for HTML5 is very short, concise, and case-insensitive: <!DOCTYPE html>.

The following markup can be used as a template to create a new HTML5 document:



```
<!DOCTYPE  html>  <html  lang="en">  <head>  <meta  charset="utf-8">
<title><!-- Insert your title here --></title> </head> <body> <!--
Insert your content here --> </body> </html>
```

Co-funded by the
Erasmus+ Programme
of the European Union

code4sp
coding for social promotion

HTML Layouts

- There are different methods of creating a web page layout, positioning the various elements that compose a web page in a well-structured manner and giving an engaging appearance to the website.

- Most websites usually display their content in multiple rows and columns, configured like a magazine to provide the users with a better reading and writing atmosphere. This can be easily attained by utilizing the HTML tags, such as <table>, <div>, <header>, <footer>, <section>, etc. and combining some CSS styles to them.

Co-funded by the Erasmus+ Programme of the European Union

HTML Layouts

- The simplest way for creating layouts in HTML is indeed obtained by designing tables. As seen in the previous sections, this usually involves the process of putting contents (text, images, etc.) into rows and columns.

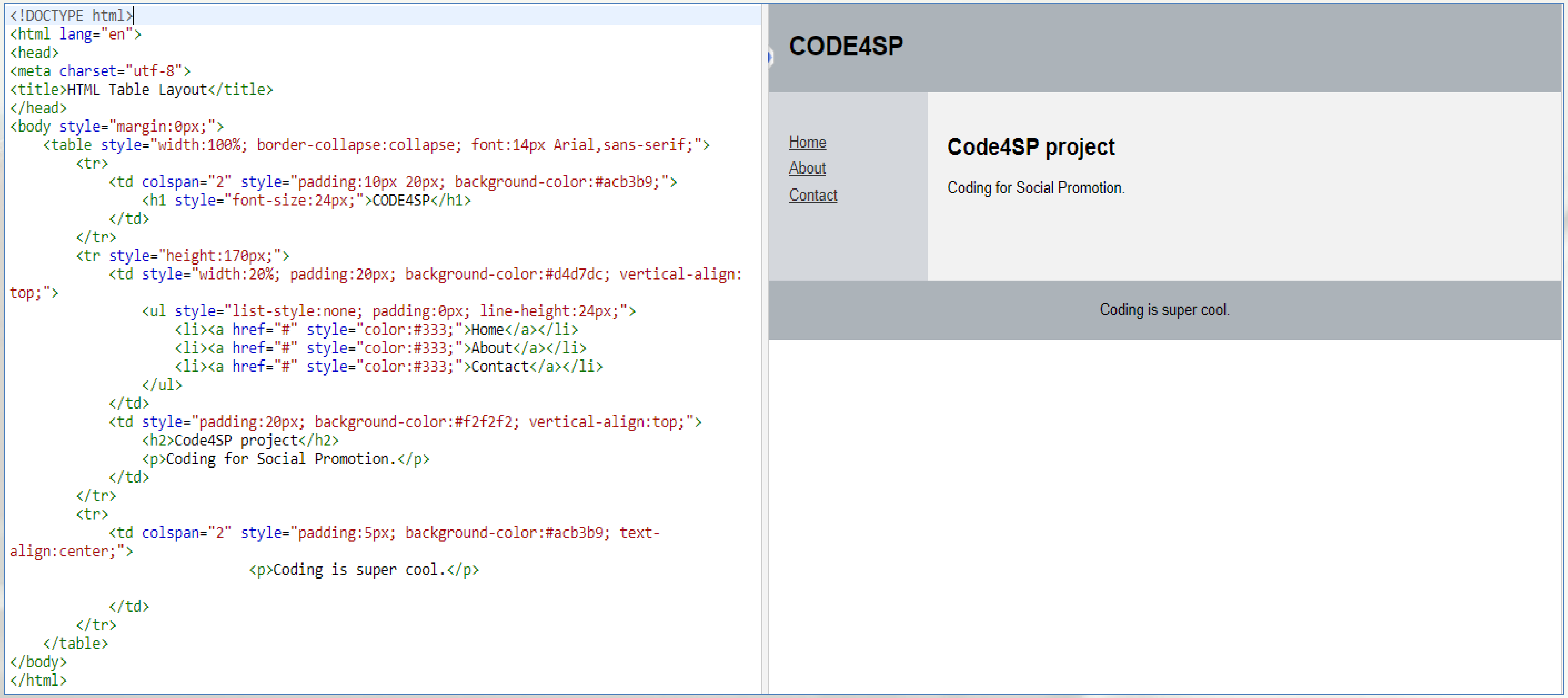
- The layout below contains an HTML table with three rows and two columns. It should be noted that the first and last row spans both uses the colspan attribute. It should be stated that the method used for creating layout in this example, although it is not wrong, it is not recommended.

- Tables and inline styles for creating layouts should be avoided. Layouts created using tables often rendered very slowly. **Tables should only be used to display tabular data.**

- For creating such layouts, **CSS float techniques are advised**. Users shall learn about this tool moreover.

Co-funded by the Erasmus+ Programme of the European Union

HTML Layouts



```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>HTML Table Layout</title>
</head>
<body style="margin:0px;">
    <table style="width:100%; border-collapse:collapse; font:14px Arial,sans-serif;">
        <tr>
            <td colspan="2" style="padding:10px 20px; background-color:#acb3b9;">
                <h1 style="font-size:24px;">CODE4SP</h1>
            </td>
        </tr>
        <tr style="height:170px;">
            <td style="width:20%; padding:20px; background-color:#d4d7dc; vertical-align:
top;">
                <ul style="list-style:none; padding:0px; line-height:24px;">
                    <li><a href="#" style="color:#333;">Home</a></li>
                    <li><a href="#" style="color:#333;">About</a></li>
                    <li><a href="#" style="color:#333;">Contact</a></li>
                </ul>
            </td>
            <td style="padding:20px; background-color:#f2f2f2; vertical-align:top;">
                <h2>Code4SP project</h2>
                <p>Coding for Social Promotion.</p>
            </td>
        </tr>
        <tr>
            <td colspan="2" style="padding:5px; background-color:#acb3b9; text-
align:center;">
                             <p>Coding is super cool.</p>

            </td>
        </tr>
    </table>
</body>
</html>
```

CODE4SP

Home
About
Contact

Code4SP project

Coding for Social Promotion.

Coding is super cool.

Co-funded by the
Erasmus+ Programme
of the European Union

HTML Layouts

- HTML5 has established new structural elements, for instance <header>, <footer>, <nav>, <section>, etc. to identify the different parts of a web page in a more semantic way.
- This example applies the new HTML5 structural elements to create the same layout presented in the previous slide.
- To know more about newly introduced tags, you should visit this source

Co-funded by the Erasmus+ Programme of the European Union

code4sp
coding for social promotion

HTML Head

- The head element is the receptacle for all head elements, which provide extra information about the document or reference to other resources, required for the document to perform properly. It illustrates the properties of the document such as title, deliver meta information like character set, tell the browser where to find the style sheets or scripts that allows to expand the HTML document interactively.

- The HTML elements that can be used inside the <head> element are: <title>, <base>, <link>, <style>, <meta>, <script> and <noscript>.

Co-funded by the Erasmus+ Programme of the European Union

HTML Head

The HTML title Element

The <title> element identifies the title of the document and only one is required in all HTML/XHTML documents to produce a valid document. It must be placed within the <head> element. The title element comprises plain text and entities and it may not include other markup tags. It may be used for different functions:

- To show a title in the browser title bar and in the task bar;
- To deliver a title for the page when it is added to favourites or bookmarked;
- To present a title for the page in search-engine results (e.g., Google search).

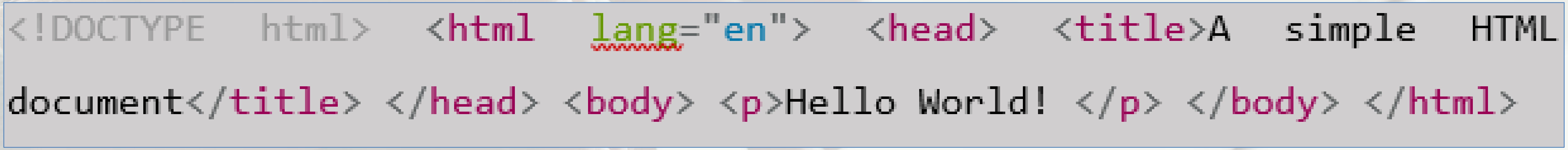
It should be short and specific to the content of the document, as search engines would pay particular attention to the words present in the title – it should have ideally 65 characters.

Co-funded by the Erasmus+ Programme of the European Union

HTML Head

The HTML title Element

A title in an HTML document should be displayed as follows:

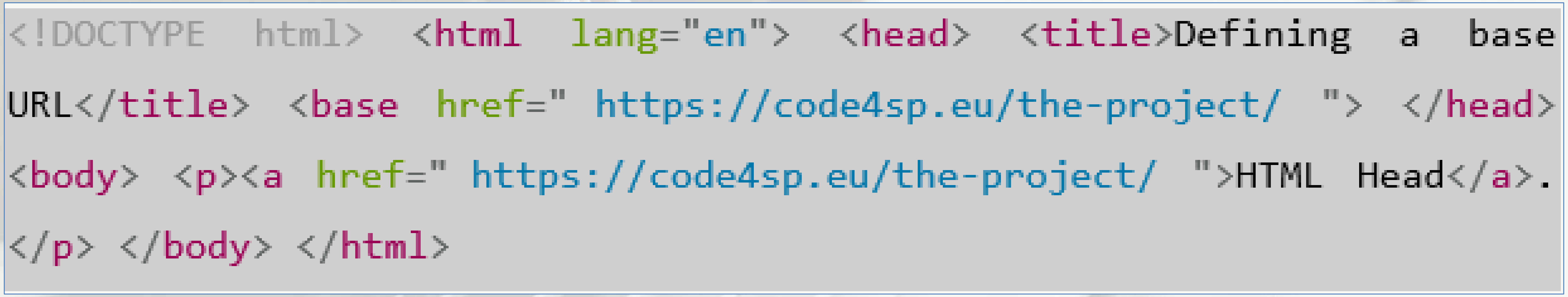


```
<!DOCTYPE html> <html lang="en"> <head> <title>A simple HTML
document</title> </head> <body> <p>Hello World! </p> </body> </html>
```

Co-funded by the
Erasmus+ Programme
of the European Union

HTML base Element

- The HTML <base> element is used to identify a base URL for all relative links contained in the document. For example, one can set the base URL once at the top of the page, and then all subsequent relative links will use that URL as a starting point, as follows:

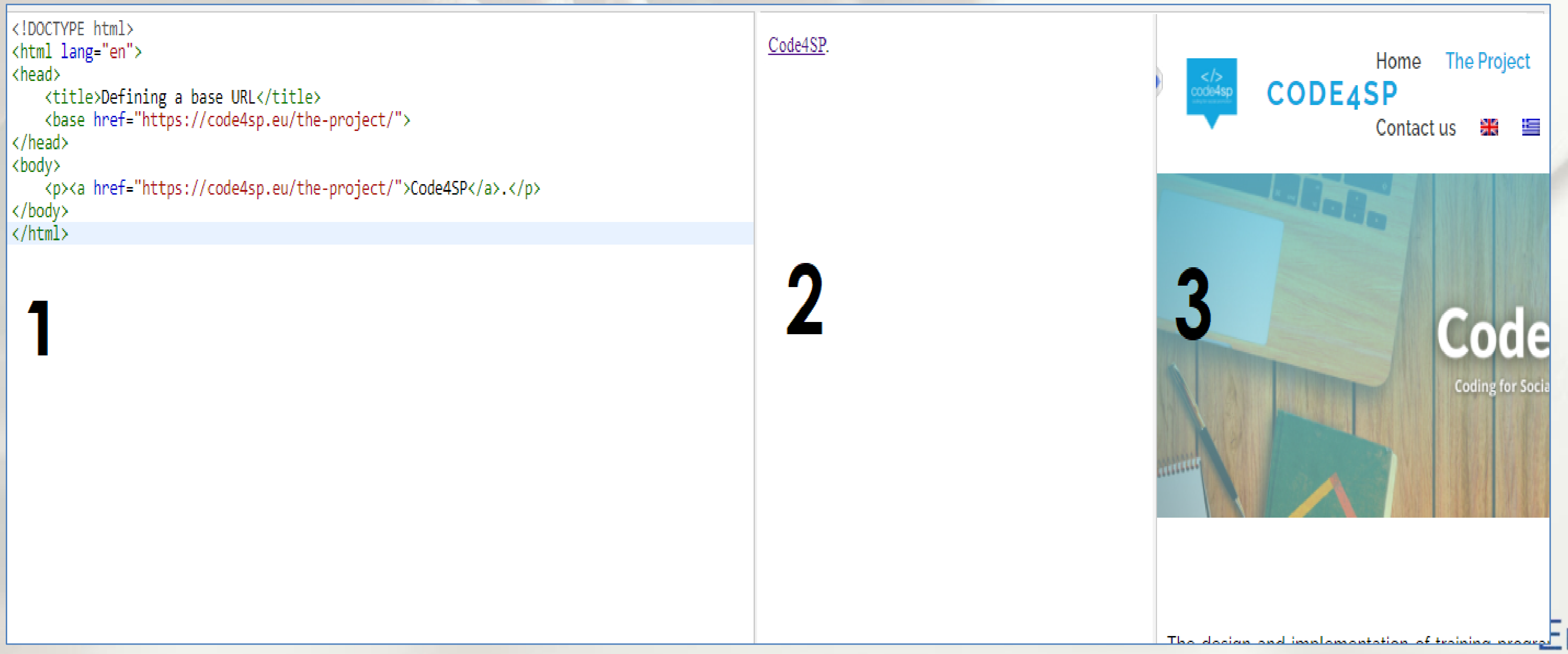


```
<!DOCTYPE html> <html lang="en"> <head> <title>Defining a base
URL</title> <base href=" https://code4sp.eu/the-project/ "> </head>
<body> <p><a href=" https://code4sp.eu/the-project/ ">HTML Head</a>.
</p> </body> </html>
```

Co-funded by the
Erasmus+ Programme
of the European Union

HTML base Element

- The HTML <base> element must be found before any element that belongs to an external resource.
- HTML permits only one base element for each document.



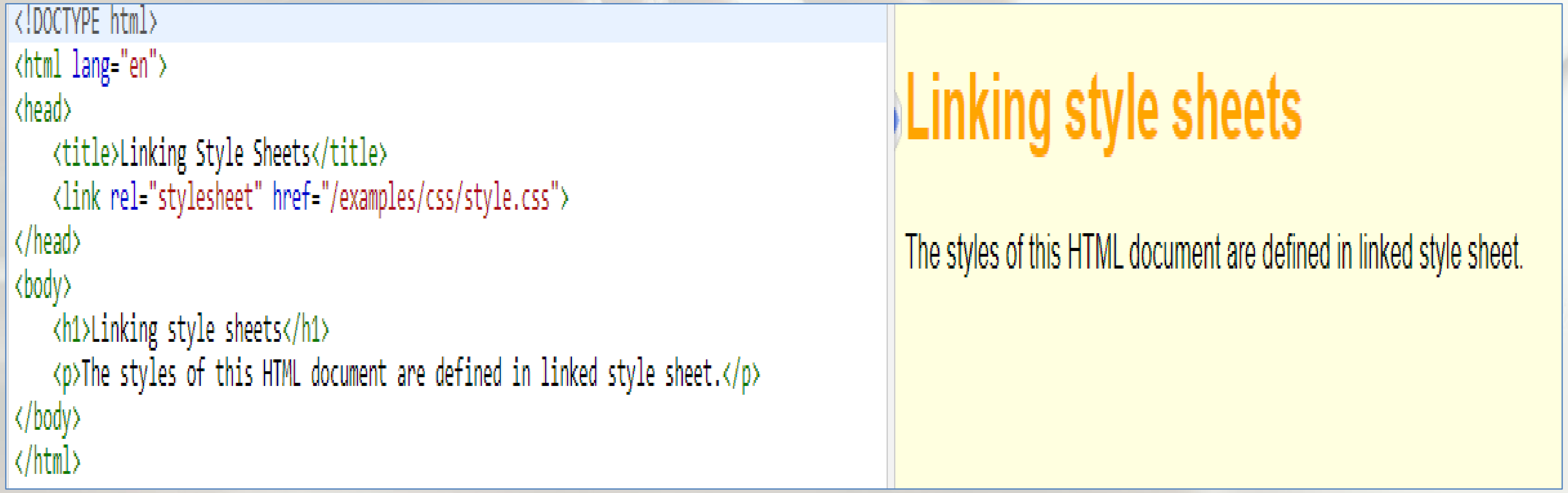
```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Defining a base URL</title>
    <base href="https://code4sp.eu/the-project/">
</head>
<body>
    <p><a href="https://code4sp.eu/the-project/">Code4SP</a>.</p>
</body>
</html>
```

Code4SP.

Co-funded by the Erasmus+ Programme of the European Union

The HTML link Element

- The <link> element describes the correlation between the current document and an external document or resource. A common use of link element is to connect to external style sheets.
- It should be stated that an HTML document's <head> element may include any number of <link> elements. The <link> element has attributes, but no contents.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Linking Style Sheets</title>
    <link rel="stylesheet" href="/examples/css/style.css">
</head>
<body>
    <h1>Linking style sheets</h1>
    <p>The styles of this HTML document are defined in linked style sheet.</p>
</body>
</html>
```

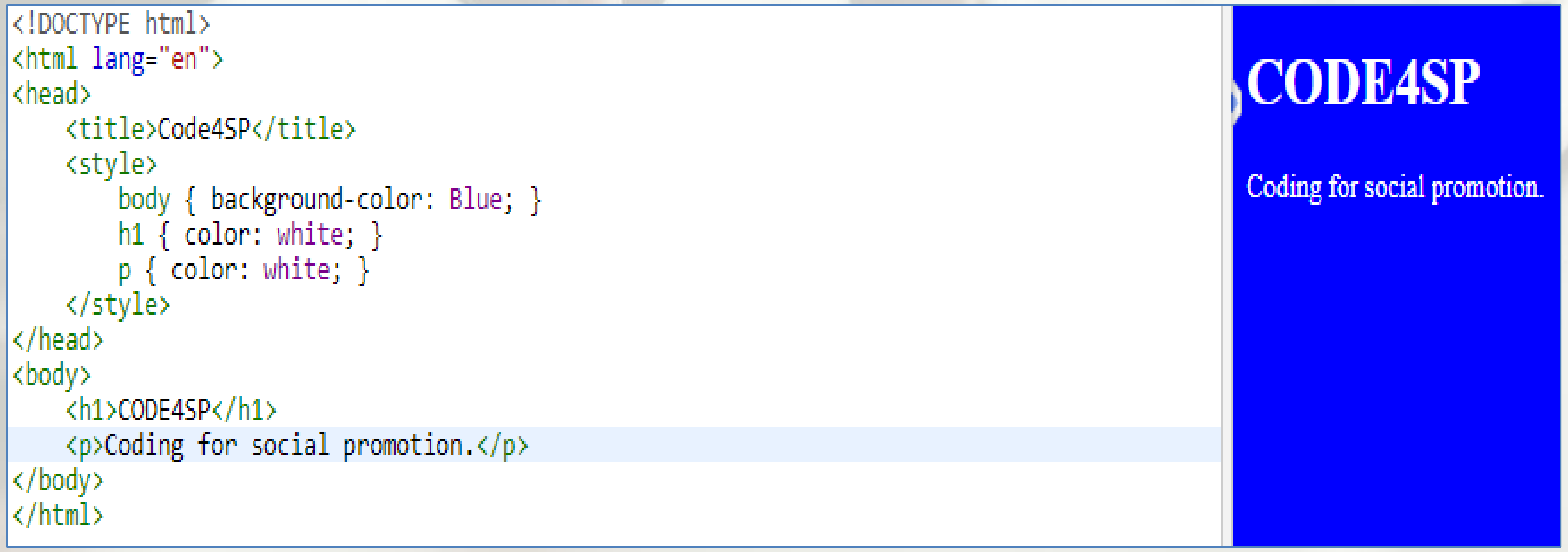
Linking style sheets

The styles of this HTML document are defined in linked style sheet.

Co-funded by the Erasmus+ Programme of the European Union

The HTML style Element

- The <style> element is applied to describe embedded style information for an HTML document. The style rules inside the <style> element indicate how HTML elements should render in a browser.
- An embedded style sheet should be used when a single document has a unique style. If the same style sheet is used in various documents, then an external style sheet would be more suitable.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Code4SP</title>
    <style>
        body { background-color: Blue; }
        h1 { color: white; }
        p { color: white; }
    </style>
</head>
<body>
    <h1>CODE4SP</h1>
    <p>Coding for social promotion.</p>
</body>
</html>
```

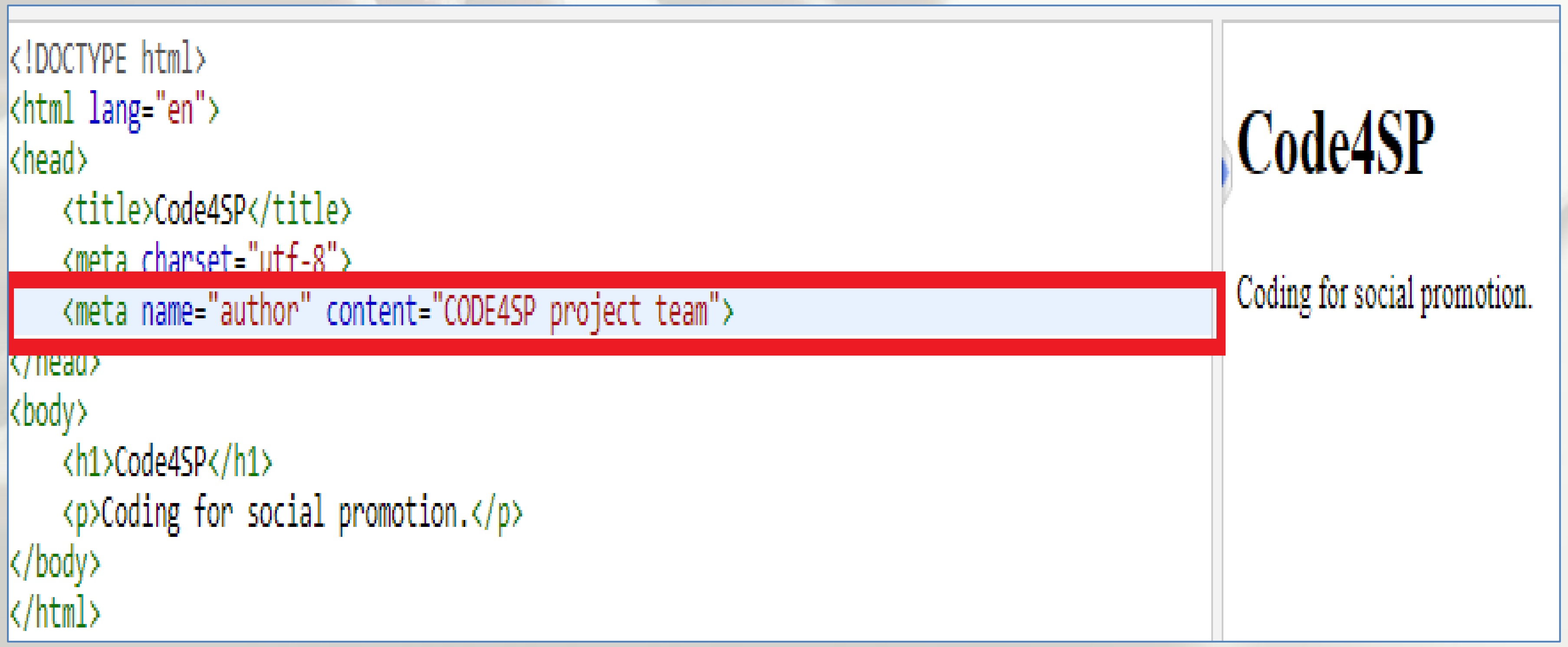
```
CODE4SP

Coding for social promotion.
```

Co-funded by the
Erasmus+ Programme
of the European Union

The HTML meta Element

- The <meta> element provides metadata about the HTML document, which is a set of data that describes and gives information about other data. <meta> tags always appear inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Code4SP</title>
    <meta charset="utf-8">
    <meta name="author" content="CODE4SP project team">
</head>
<body>
    <h1>Code4SP</h1>
    <p>Coding for social promotion.</p>
</body>
</html>
```

Co-funded by the Erasmus+ Programme of the European Union

The HTML meta Element

- As seen above, meta tags include information about a web page. It is not visible in the browser (it is machine parsable though). Metadata is utilised by browsers (how to display content or reload page), search engines (keywords), and other web services. Moreover, there is a technique to let web designers rule over the **viewport**, through the <meta> tag. The viewport is the user's visible area of a web page. It differs from device to device (it will be bigger on a computer screen than on a mobile phone).

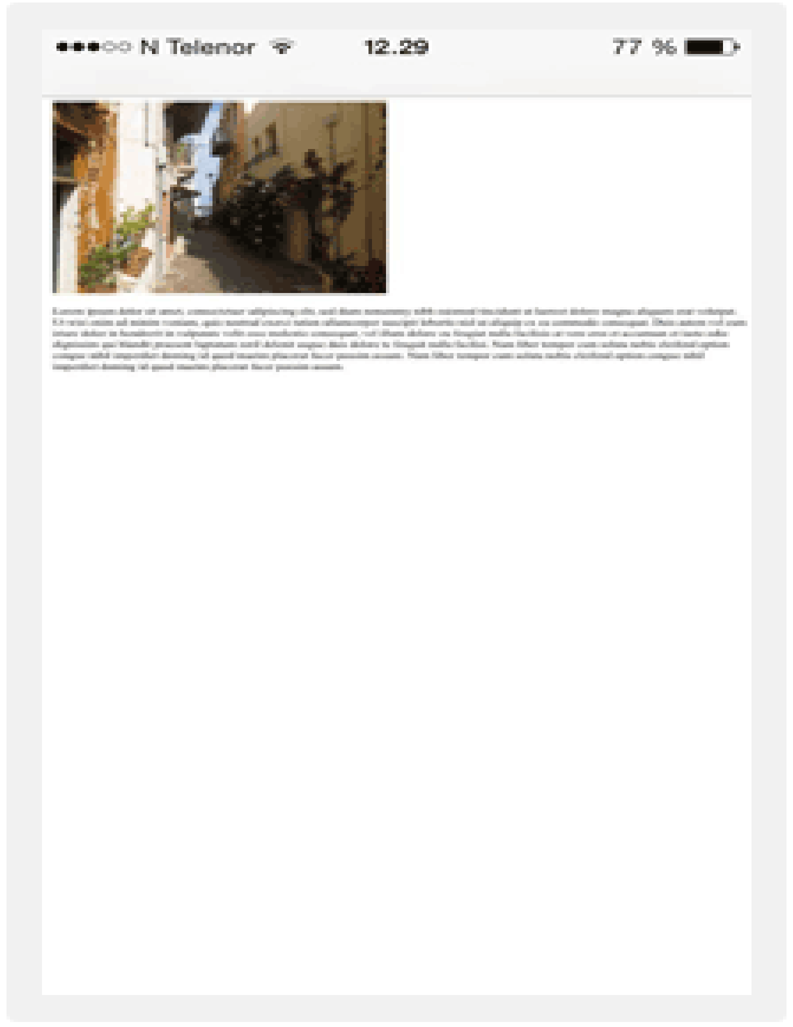
- The following <meta> element should be included in all web pages, as it will give the browser guidelines on how to assume the page dimensions and scaling: `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

- The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the screen). The initial-scale=1.0 part sets the initial zoom level when the page is initially loaded by the browser.

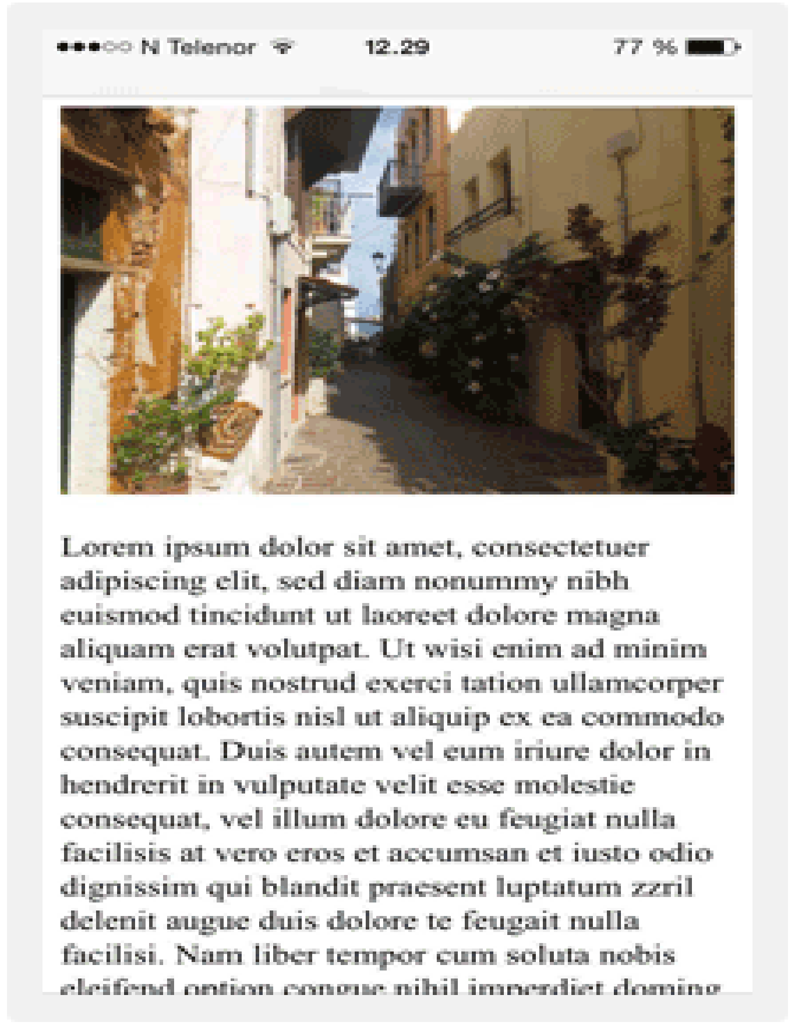
Co-funded by the Erasmus+ Programme of the European Union

The HTML meta Element

- The difference between webpages with or without viewport meta tag is quite visible in the following example:



Without the viewport meta tag



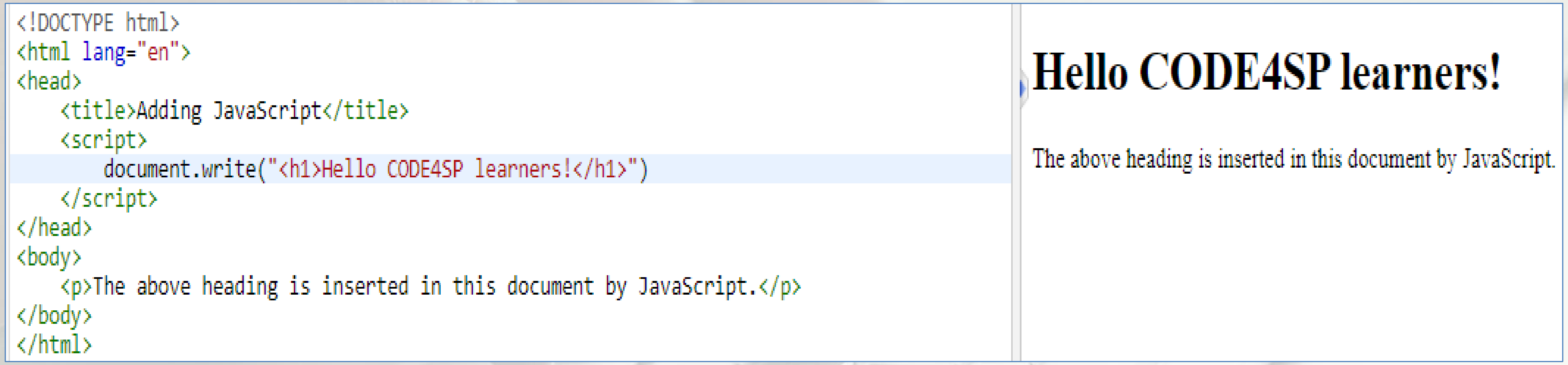
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming

With the viewport meta tag

Co-funded by the Erasmus+ Programme of the European Union

The HTML script Element

- The <script> element is used to define client-side script, such as JavaScript in HTML documents.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Adding JavaScript</title>
    <script>
        document.write("<h1>Hello CODE4SP learners!</h1>")
    </script>
</head>
<body>
    <p>The above heading is inserted in this document by JavaScript.</p>
</body>
</html>
```

Hello CODE4SP learners!

The above heading is inserted in this document by JavaScript.

Co-funded by the
Erasmus+ Programme
of the European Union

HTML base Element

Working with Client-side Script

- Client-side scripting is linked to the type of computer programs that are performed by the user's web browser. JavaScript (JS) is the most widespread client-side scripting language on the web.

- The <script> element is used to embed or reference JavaScript within an HTML document to add interactivity to web pages and to create a more user-friendly experience. Some of the most common uses of JavaScript are form validation, generating alert messages, creating image gallery, show hide content, DOM manipulation, etc.

Co-funded by the
Erasmus+ Programme
of the European Union

code4sp
coding for social promotion

HTML base Element

Working with Client-side Script

- JavaScript can be **embedded** following two ways:
  1. Directly inside the HTML page; or
  2. Placed in an external script file and referenced inside the HTML page.

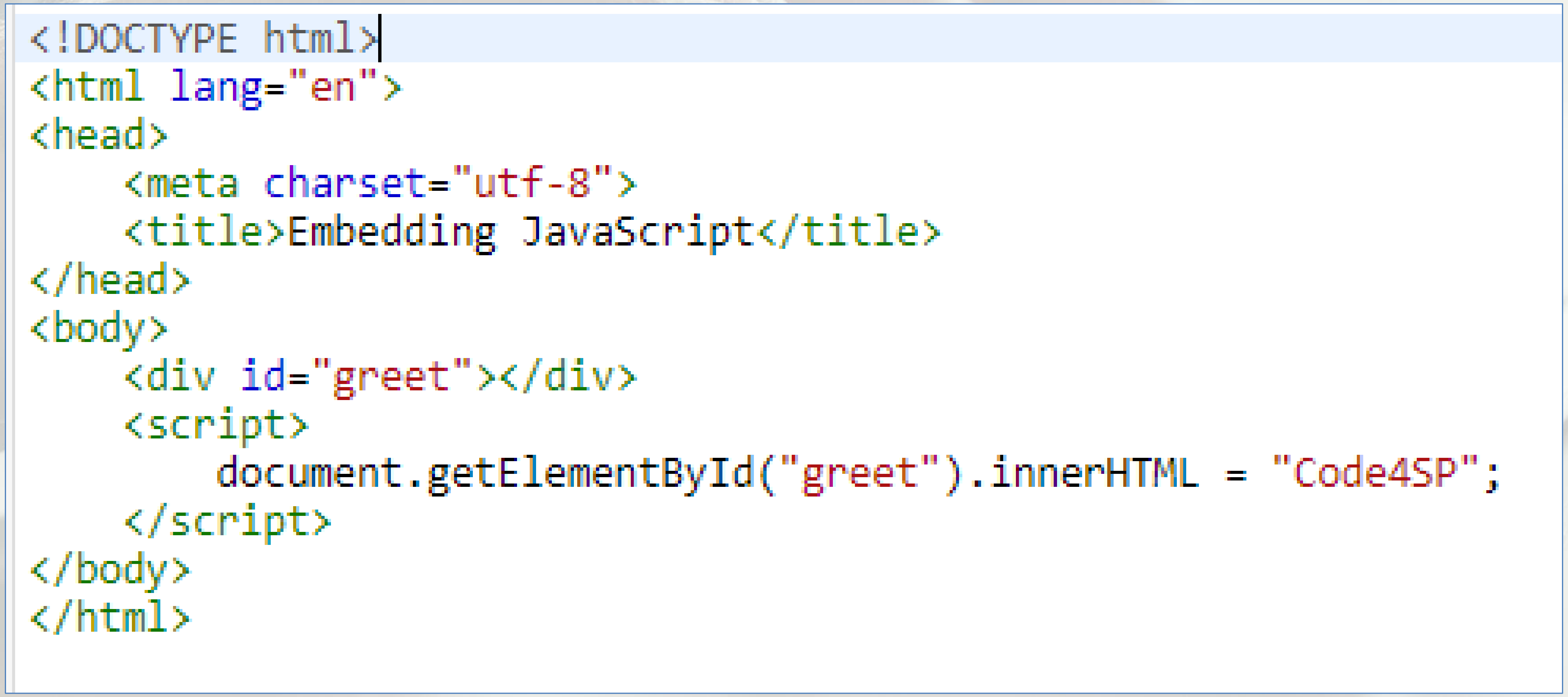
  Note: both techniques use the <script> element.

- To embed JS in an HTML file, the user must add the code as the content of the <script> element, following the example of the next slide.

- Preferably, script elements should be placed at the end of the page, before the closing body tag (</body>), because when browser encounters a script, it pauses rendering the rest of the page until it deconstructs the script that may drastically impact the website performance.

Co-funded by the
Erasmus+ Programme
of the European Union

Working with Client-side Script



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Embedding JavaScript</title>
</head>
<body>
    <div id="greet"></div>
    <script>
        document.getElementById("greet").innerHTML = "Code4SP";
    </script>
</body>
</html>
```

Co-funded by the
Erasmus+ Programme
of the European Union

HTML base Element

Working with Client-side Script

- JavaScript codes can also be placed into a separate file, calling up a .js extension, while corresponding it in the HTML document by using the src attribute of the <script> tag.

- This can be especially useful if the web designer wants to make the same script available for multiple documents, so he/she does not need to perform the same tasks repeatedly. When the src attribute is indicated, the <script> element should be empty, so the web designer cannot use the same <script> element to both embed the JavaScript and to link to an external JavaScript file in an HTML document.

- If a browser, for some reason, does not support client-side scripting, or users have disabled JS in their browser, the <no script> element can be used to provide an alternative content. This element can include any HTML elements, except <script>, as it can be included in the <body> element of a HTML page.

Co-funded by the Erasmus+ Programme of the European Union

HTML Entities

- Most learners will certainly be curious about how to display special characters and symbols in their programming processes. Hence, this subchapter is intended to explain how they can be successful in doing such thing.

- First, it is important to understand what a HTML entity is. As perceived in the previous chapters, some characters are quite reserved in HTML. For instance, one cannot use signs such as ‘<’ or ‘>’, as the browser could mistake them for a markup. Moreover, some characters are just not available in the keyboard (e.g., the copyright symbol).

Co-funded by the
Erasmus+ Programme
of the European Union

HTML Entities

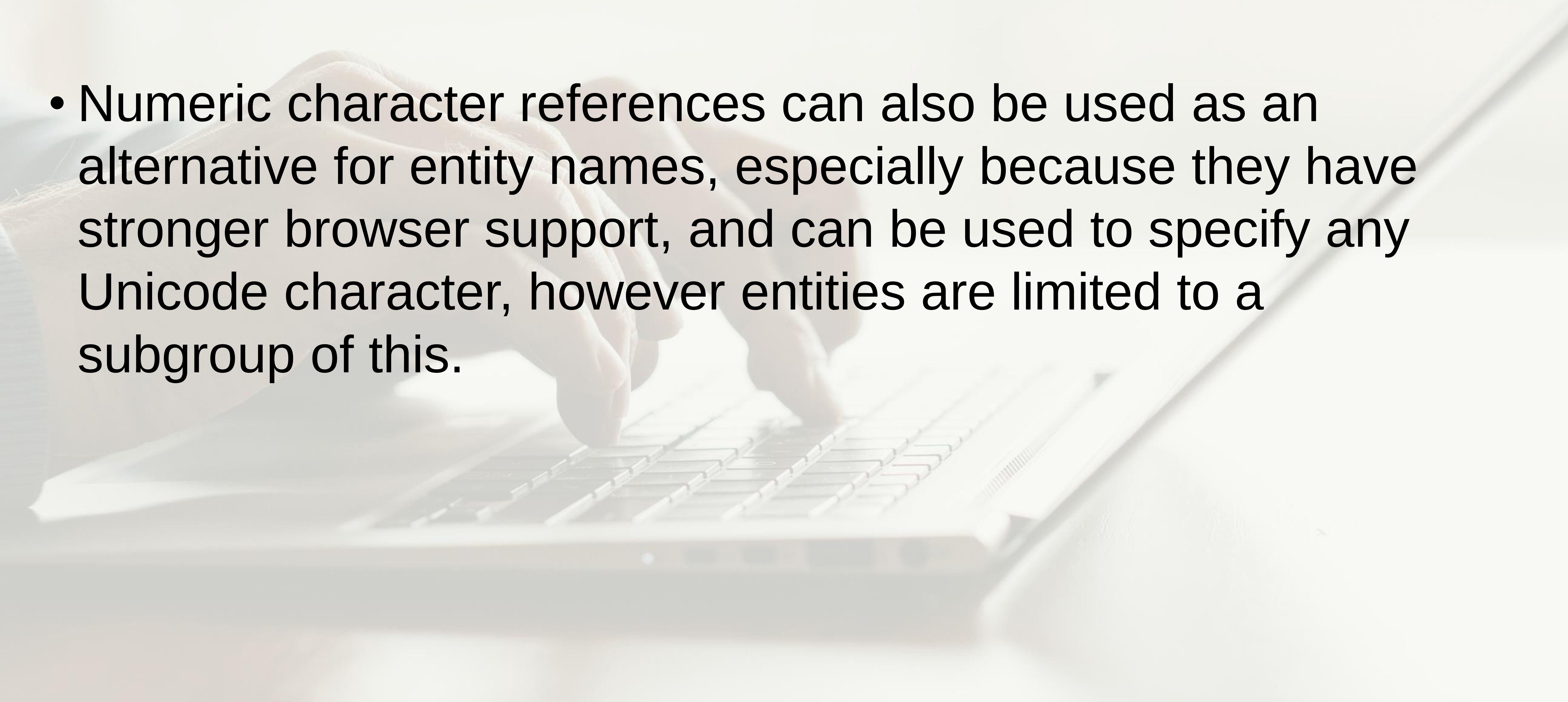
- These special characters can be displayed by simply being replaced with the character entities (or just entities), then solving the aforementioned troubles.

Co-funded by the Erasmus+ Programme of the European Union

HTML Entities

Below is a list of the most frequently used entities in HTML

Result	Description	Entity Name	Numerical reference
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
"	quotation mark	&quot;	&#34;
'	apostrophe	&apos;	&#39;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;



HTML Entities

- Numeric character references can also be used as an alternative for entity names, especially because they have stronger browser support, and can be used to specify any Unicode character, however entities are limited to a subgroup of this.

HTML URL

- URL: how many times has this abbreviation appeared in virtual environments? It stands for Uniform Resource Locator, and it works as the global address of documents and other resources on the World Wide Web.
- Its goal is to identify the location of a document and other resources available online, while specifying the mechanism for accessing it using a web browser. For instance, https://code4sp.eu/ is an URL.
- The general syntax of URLs can be described as follows: `scheme://host:port/path?query-string#fragment-id`

HTML URL

URLs have a linear structure and are composed of the following components:

- **Scheme name** — The scheme recognizes the protocol to be used to access a resource on the Internet. The scheme names are followed by the three characters :// (a colon and two slashes). The most used protocols are http://, https://, ftp://, and mailto://.
- **Host name** — identifies the host where the resource is situated. A hostname is a domain name given to a host computer. This is usually a combination of the host's local name with its parent domain's name. For example, www.code4sp.eu/ consists of host's machine name www and the domain name code4sp.eu.

HTML URL

URLs have a linear structure and are composed of the following components:

- **Port Number** — Servers often deliver more than one type of service, so they must be told what service is being requested. These requests are made by port number. Well-known port numbers for a service are normally omitted from the URL. For example, web service HTTP runs by default over port 80, HTTPS runs by default over port 443.
- **Path** — identifies the specific resource within the host that the user wants to access. For example, /html/html-url.php, /news/technology/, etc.

HTML URL

URLs have a linear structure and are composed of the following components:

- **Query String** — contains data to be passed to server-side scripts, running on the web server. For instance, parameters for a search. The query string preceded by a question mark (?), is usually a string of name and value pairs separated by ampersand (&), for example, ?first_name=John&last_name=Corner, q=mobile+phone, and so on.
- **Fragment identifier** —specifies a location within the page. Browser may scroll to display that part of the page. The fragment identifier introduced by a hash character (#) is the optional last part of a URL for a document.

HTML URL

HTML URL Encoding

- It is well-known that sometimes data is not safely transmitted over the internet. This happens mostly because URLs are not fully or accurately encoded and causes some misunderstandings among internet users.

- According to [RFC 3986](), the characters in a URL only restricted to a defined set of reserved and unreserved US-ASCII characters. Any other characters are not allowed in a URL (that is why some Latin and Cyrillic characters are not seen in URL). But URL often comprises characters outside the US-ASCII character set, so they must be converted to a valid US-ASCII format for worldwide interoperability. URL-encoding is a process of converting URL information so that it can be safely transmitted over the internet.

HTML URL

HTML URL Encoding

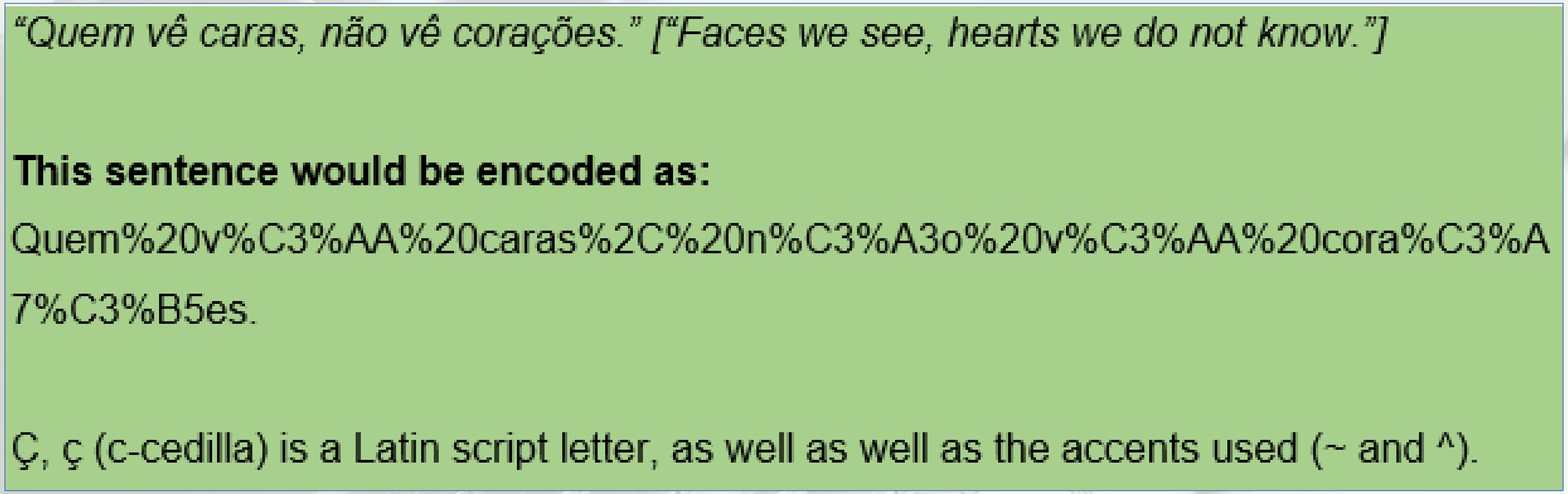
To map the large range of characters used globally, a two-step method is followed:

- Firstly, the data is encoded in relation to the UTF-8 character encoding.
- Then only those bytes that do not correspond to characters in the unreserved set should be percent-encoded like %HH, where HH is the hexadecimal value of the byte.

HTML URL

HTML URL Encoding

- Let’s take a look at this Portuguese popular saying:



“Quem vê caras, não vê corações.” [“Faces we see, hearts we do not know.”]

This sentence would be encoded as:

Quem%20v%C3%AA%20caras%2C%20n%C3%A3o%20v%C3%AA%20cora%C3%A7%C3%B5es.

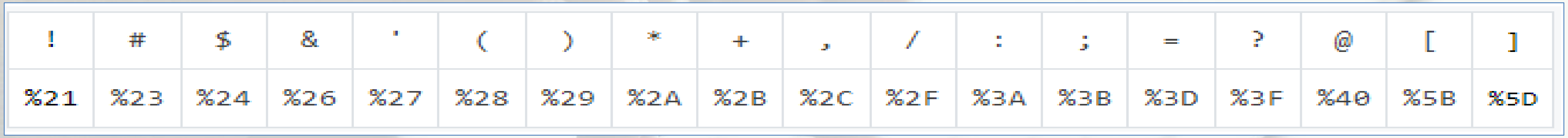
Ç, ç (c-cedilla) is a Latin script letter, as well as well as the accents used (~ and ^).

- Certain characters are restricted from use in a URL, as they may (or may not) be defined as delimiters by the generic syntax in a particular URL scheme (for instance, forward slash / characters are used to separate different parts of a URL).

HTML URL

HTML URL Encoding

- Reserved characters in a URL are as follows:

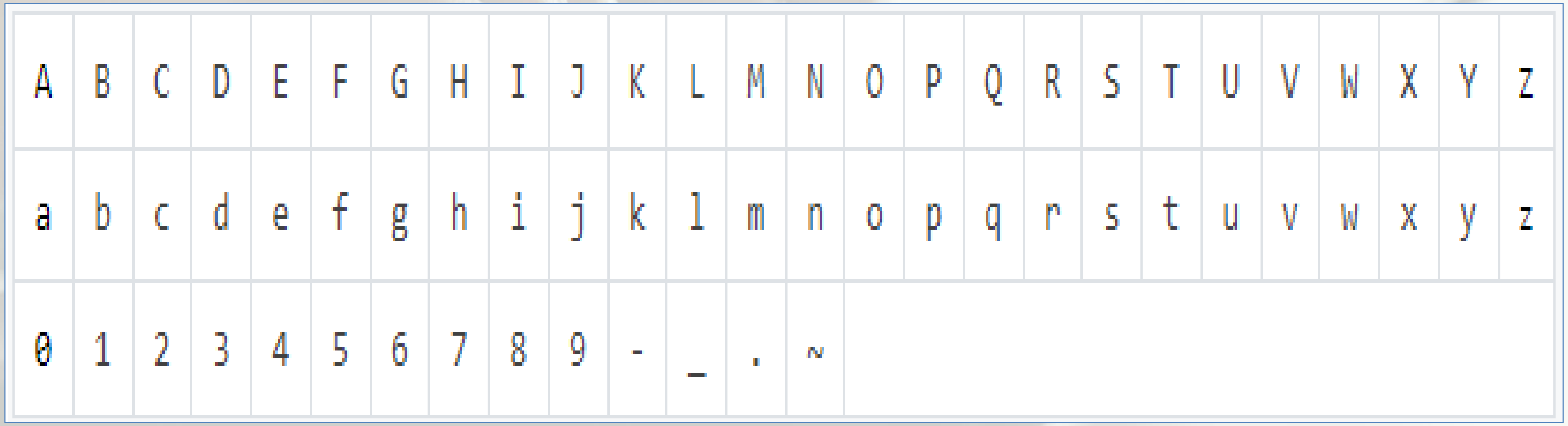


!	#	$	&	'	(	)	*	+	,	/	:	;	=	?	@	[	]
%21	%23	%24	%26	%27	%28	%29	%2A	%2B	%2C	%2F	%3A	%3B	%3D	%3F	%40	%5B	%5D

HTML URL

HTML URL Encoding

- However, there are also characters that, despite of being allowed in a URL, do not have a reserved purpose – that is why they are called “unreserved characters”.
- These include uppercase and lowercase letters, decimal digits, hyphen, period, underscore, and tilde. The following table lists all the unreserved characters in a URL



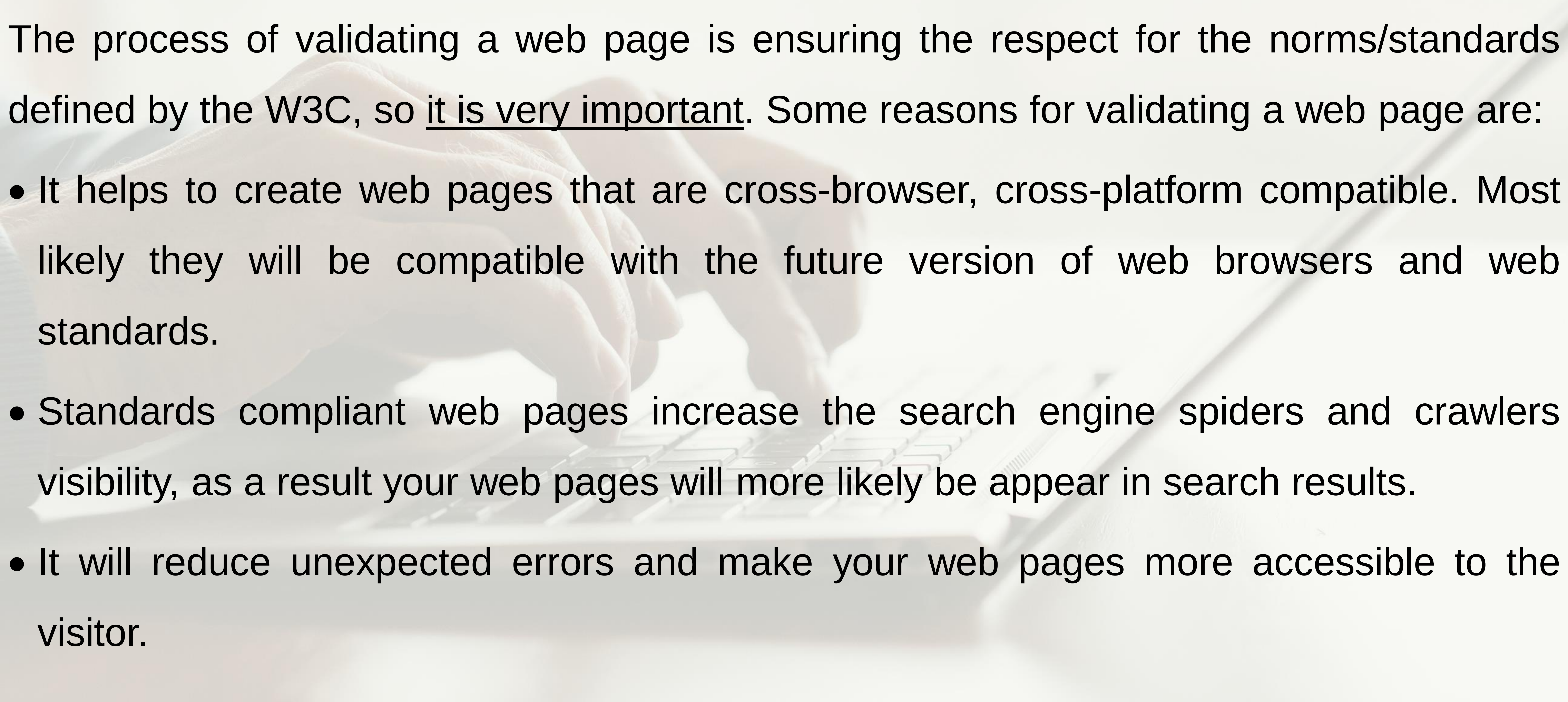
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	-	_	.	~												

- For encoding/decoding characters, you may use this converter.

HTML Validation

- To make sure a HTML code follows the current web standards, free from errors, it is of upmost importance to figure out how to validate a HTML code. Beginners often will make mistakes when writing HTML codes, and incorrect or non-standard codes will certainly cause unexpected results in how a web page is displayed in a browser.

- In order to prevent this to happen, users can test their codes within the formal guidelines and standards set by the Wide Web Consortium (W3C) for HTML/XHTML web pages. There is an online tool that automatically checks HTML codes and points out any problems/errors, like missing closing tags or missing quotes around attributes.



HTML Validation

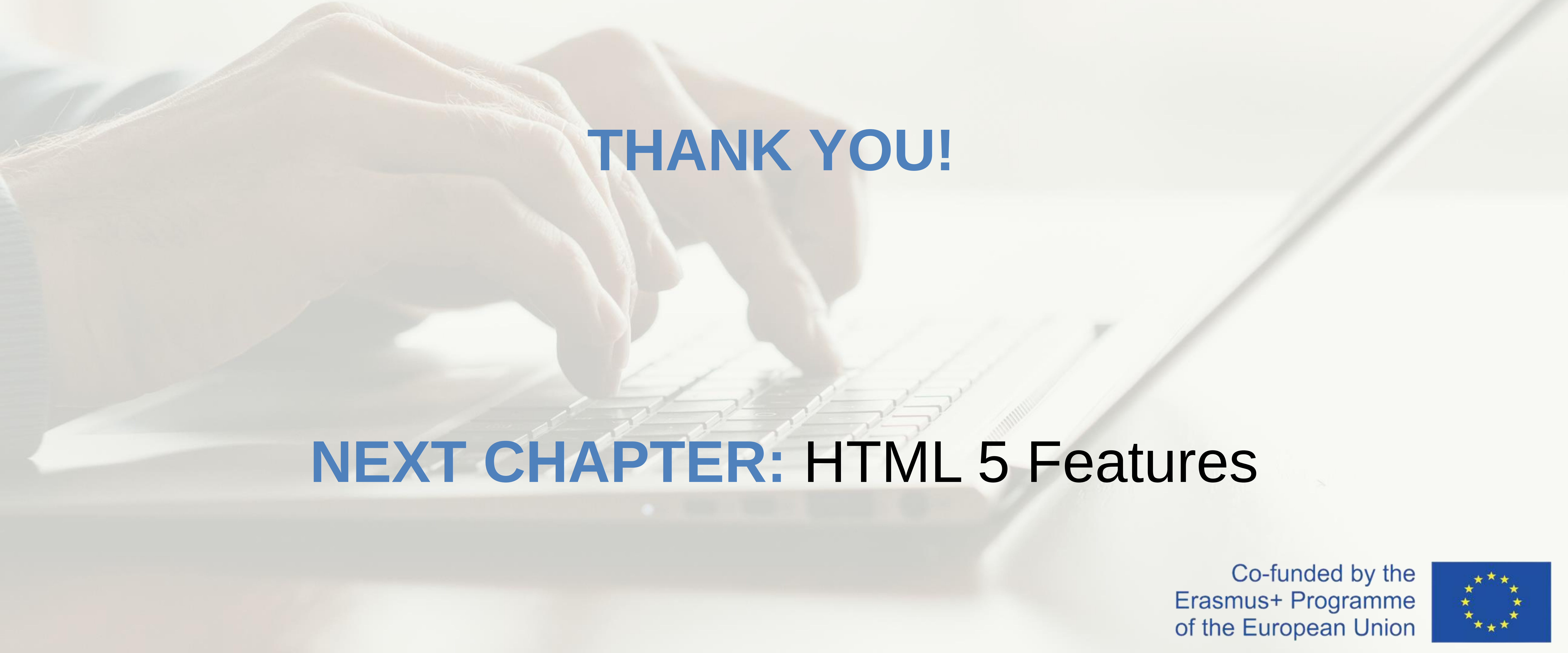
The process of validating a web page is ensuring the respect for the norms/standards defined by the W3C, so it is very important. Some reasons for validating a web page are:

- It helps to create web pages that are cross-browser, cross-platform compatible. Most likely they will be compatible with the future version of web browsers and web standards.
- Standards compliant web pages increase the search engine spiders and crawlers visibility, as a result your web pages will more likely be appear in search results.
- It will reduce unexpected errors and make your web pages more accessible to the visitor.

LET’S PRACTICE!

Open the following link in order to practice some of the concepts acquired so far:

- https://www.w3schools.com/html/exercise.asp?filename=exercise_html_attributes1



</> code4sp coding for social promotion

THANK YOU!

NEXT CHAPTER: HTML 5 Features

Co-funded by the Erasmus+ Programme of the European Union