- Hello, my name is Victor, I'm an Application Developer for Atos and I am going to show you how to create tables in HTML.

- Through this video I will cover some things, we are starting with an overview, then we are going to talk about the correct use of tables in HTML, table tags and attributes, semantics, captions, cell spanning and column grouping.

a) What is a table?

1. What is a table.

- Letts’s start with the basics, what is a table? Well, we all have seen a table before, a table is an element used for displaying data in the form of rows and columns.

2. Table examples from the web.

- In HTML, we can create our own tables to display data.

- On this example we see a simple HTML table, it has rows and columns containing the data, and a header which indicates the data element each column holds.

b) Correct use.

- Now, before we go ahead, we need to talk about the correct use of tables in HTML, it may look pretty straight forward but tables are often misused in HTML.

1. Correct use of tables (tabular data).

- As tables are defined, HTML tables are supposed to be used to display data, tabular data to be precise...

2. What is tabular data.

- And what is tabular data? well, it is essentially data which can be shown in a table.

- So, let's say we have 3 fruits: mango, pineapple and kiwi, in order to present this data, we could use a table, but is it really necessary? it depends on what you are doing, but probably it isn't, we could just use a list.

- In the other hand, if we had a bigger piece of data like this JSON file which contains customers information, maybe a table would be useful.

3. Incorrect use of tables (layout).

- So that is the correct use of tables in HTML, but what’s is the incorrect way? well, let's say I have to create a web page that needs to be portrayed in a "3 column way" and I use a table for it, that would be an example of an incorrect use, because tables are not supposed to be used for layout purposes, just for tabular data.

- In order to achieve a 3-column page, you should use a CSS layout like the grid layout.

- Always remember, using a table for layout is not a good practice.

[Code editor section]

- Before we go jump into some code, let me show you what you need and what I'm gonna be using for coding during the video.

- First thing you need is a code editor, if you don't know what that is don't worry, I'll explain: you can understand a code editor as an app which allows you to create code files like some programing languages and of course HTML pages.

- My personal choice is Visual Studio Code, if you have a bit more experience and you have your own choice feel free to use it.

- So, if you are following me, just go to 'code.visualstudio.com', click on download and follow the installer's instructions.

- Also, something else you will see is that I'm not reloading the page after each change I make, that's because I'm using an extension. If you go to the left side you are going to find 5 sections, explorer, search, source control, run & debug, and extensions, if you are following me just look for live... server, this one right here, it's already installed for me, you click on it, then you click on install and that’s it.

[]

c) Table tags.

- Ok, now we can go create a table.

- What do we need to do it? We need the TABLE tag, which represents the main container for the table, the TR tag which stands for table rows, the TD tag which can be understanded as table data or table cells and the TH tag for table headers.

1. Table tag.

- Let’s see this in code.

[VS Code overview, creating & opening folder]

- Open your VS Code, if you go to the left side, you can see 5 sections: explorer, search, source control, run and debug, and extensions.

- Create a folder on your desktop, then go to the explorer section and open the folder.

[]

[Creating web page & EMET]

- So, first let's create a simple web page.

- Just click on new file button, type index.html and press enter.

- Ok, VS Code comes with something called EMMET, which helps by providing shortcuts to write in HTML, if you type an exclamation sign and then you press TAB, an HTML structure will be generated, very simple.

[]

- Now that we have our page, let’s just add the table tag to create a table, and we are doing that inside the body.

2. Row tag.

- Next, we need to add some rows to it, for that we have the TR and TD tags.

- Let’s type a TR tag inside the table, now we have a row, just that it’s empty as you can see, we need some cells.

3. Data tag.

- To do that, we need to add some TD tags inside the row.

- Using EMMET you can simply type the tag name and press enter.

- If the TR tag stands for a row, each TD tag we add inside will mean a column or a cell.

- Now we can see that the 4 fields we add as TD's inside our row are showing as 4 separate columns.

- But right now, our table only has 1 row, let's add more, if we copy the TR and paste it 4 times below, we end up having 5 rows.

- So, we have a couple of rows, but something is missing, we need a header.

4. Header.

- In order to add a header, we need to add another row right after the table tag, before the other rows.

4.1. Heading tag.

- But for it to be a header, we are going to use the TH tag instead of the TD.

- And now we can see that our new row appears with bold text, that’s because the TH tag defines a cell as a header.

d) Table attributes.

- There are a couple of attributes we can add to a table...

1. Deprecated attributes.

- But actually, these attributes are now deprecated...

2. Style for layout (recommendation).

- Because you should use CSS styling instead to apply those properties.

- Besides these attributes, there are others you can use like 'scope'.

3. Scopes.

- The 'scope' attribute is related to headers, in our example we have our headers on top, but we can also add headers to the side, and for this we can add some scoped headers.

- If we go to the table's header, I can add the 'scope' attribute to each TH tag and set it as 'column'.

- Ok, now to create some side headers, I'm going to add a TH tag inside each row, and it’s going to be for an ID.

- But this time I'm setting the scope attribute as 'row’.

- Also, I’m adding another column header for the ID.

- And there we go; you can see now we have some side headers in our table.

- But there is something I want to show you, I'm going to remove the 'scope' attribute from all TH tags, you can see that nothing changes in our page, it still the same, why? Well, the 'scope' tag doesn't precisely reflect on the page, it's more used as an accessibility resource, for example, it can be used by screen readers, so you see, setting the 'scope' attribute for the headers is a good practice.

e) Semantic tags.

- Ok, now it's time to talk about semantic tags.

1. What & why?

- Within HTML, there is a group of tags called semantic tags, which are meant to add a better content organization to both developers and web browsers. HTML comes with 3 semantic tags for tables: THEAD, TBODY and TFOOT.

2. Head tag.

- The THEAD tag stands for the header part of the table, so I'm adding a THEAD tag right after the TABLE tag, and now I'm coping the first row with our column headers and pasting it inside the THEAD tag. Now we have defined the header part of our table.

3. Body tag.

- Just as the THEAD, the TBODY tag is meant to define the body of the table, so, I'm adding a TBODY tag after the THEAD, and pasting the rest of the rows inside it.

4. Footer tag.

- At the end, we can also add a TFOOT tag in order to define the footer of our table, let's add it at the end.

- Sometimes it could be not necessary, but let's say you have a large table and you want to repeat the column names at the end, so I'm adding a row inside the tfoot tag.

- And there we go; we have defined the table's content. Don't forget, these tags are only meant for structuring and content organization.

f) Captions.

- There is another tag we can include inside the table, the 'caption' tag.

1. What for?

- You can take the caption tag as a name or description for the table, so, I'm adding the tag before anything else inside the table.

2. Caption tag.

- And now our table is shown with a text on top.

- It's also a good practice to place the caption tag as the first one inside the table.

g) Table span's.

- Now, what if you want to expand a table cell to another row o column? Well, you can use the 'rowspan' and 'colspan’ attributes.

1. Column span.

- Let's say I want this cell to expand to the next column, so let's add the 'colspan' attribute and set it as 2, let me change the header name to ‘full name’, and I’m also commenting the next cell.

- With this, you can tell the cell is now using 2 column spaces beginning on its current position.

2. Row span.

- And the same happens for rows, if we need to expand a cell to the next row, we have to add the 'rowspan' attribute and set it also as 2.

- That's how we span cells along the table.

**h) Grouping.**

- So, there is one last thing I want to cover, and it is called column grouping.

1. What for?

- Grouping allows us to create some column groups, so instead of applying some properties to each column, we can assign them to the group.

2. Colgroup & col tags.

- Let's see, I added a COLGROUP tag after the caption tag, by the way that’s the recommended order for the tags. Inside the COLGROUP you can see I added 5 'col' tags, which represents each column in the table.

- I have prepared some style for this example, I'm going to add a class 'blue' to each col, you can see that the background of each column has turned blue, but now I want to set a different color for the columns, so, I'm adding a class 'red' to the last 3 col's, you can see I'm leaving only 2 col tags instead of 5, and I'm adding another attribute called 'span', which is the same as ‘colspan’ but for the ‘col’ tag, setting it as 2 for the first col and 3 for the second one, now the columns are shown with different background colors.

- You can also separate the red and blue columns into 2 separated groups, and it will work just fine.

- And with this we just finished HTML tables, feel free to ask me any questions and thank you for your time, see you next time.