



[Course](#) > [Modul...](#) > [2.2 Attr...](#) > [Introdu...](#)

Introduction to attributes

Introduction to attributes

We learned a little bit about what attributes are in the previous module. Let's look into it in more depth, by using examples.

Here is an ordered list:

```
1. <ol>
2.   <li>Lights</li>
3.   <li>Camera</li>
4.   <li>Action</li>
5. </ol>
```

Output:

1. Lights
2. Camera
3. Action

If i want an ordered list to start with the number 5 instead of 1 (as it does by default), let's code like this:

```
1. <ol start="5">
2.   <li>Lights</li>
3.   <li>Camera</li>
4.   <li>Action</li>
5. </ol>
```

Output:

5. Lights
6. Camera
7. Action

Here, using the `start` attribute, we made our list start with 5 instead of 1.

Like `start`, we have many useful attributes we will see in this section that can affect your element. Attributes are a significant part of HTML. Tags and attributes make up the language.

Syntax:

Attributes are used in tags to further define the tag:

- It is used inside the opening tag it is applied to and should be added after a space from the tag name: `<ol start="5">`. The `start` attribute is used inside the `` tag.
- `start="5"`
Attribute name, equal sign, opening quote, attribute value, closing quote
- Attributes are a name-value pair: `start="5"`
name: `start`
value: any positive integer
- The only exception to the name-value pair is if the attribute is a 'boolean attribute'. These attributes have only two types of values - true or false. But instead of writing "true" or "false" for its value, you add the attribute name to indicate true and omit it to indicate false. An example is the 'reversed' attribute in an ordered list ``. Adding this attribute is an indication that the list order should be reversed (in descending order).
`<ol reversed>`
- A tag can have multiple attributes:
`<ol start="5">`
`<ol id="cinema" class="attribute-list" start="5">`
`<ol start="5" class="attribute-list">`

Example 1: the 'id' attribute

Imagine you have two paragraphs in your HTML page:

1. `<p>I am paragraph 1 and I want to be in red</p>`
2. `<p>I am paragraph 2 and I want to be in blue</p>`

Your task is to make the text color of the first paragraph red and the other blue. How do we do that? You add styling to your HTML document through CSS. CSS is a style sheet language where you add any presentation related information for your HTML document. You will learn about this in the next chapter. In this case, you will have to write code in your style sheet to inform that it needs to change the text colors respectively.

But to identify each paragraph, we need to give them each a name first so we can instruct our style sheet to make X red and Y blue. This unique name we give each element is called an 'ID'. This is very similar to your school or corporate ID that is unique to you. No one else in your company will have the same ID as you. `id` is an attribute. It should be unique to the element because we know that two people having the same ID will just cause a lot of confusion.

1. `<p id="para1">I am paragraph 1 and I want to be in red</p>`
2. `<p id="para2">I am paragraph 2 and I want to be in blue</p>`

Here, we can style `para1` and `para2` separately using CSS. The `id` attribute helps us do this by letting us give each paragraph an ID.

Example 2: the 'class' attribute

A similar attribute is `class`. `class` like `id` is a very useful attribute and one you will be using very frequently. Let's assume you are an author of a book. You like poems and you want to include at least 20 of them in your new book. You add IDs for them: `'poem1'`, `'poem2'`, `'poem3'`.

You want your poems to look different from your other text. Grey text color, italic and bold. Like this:

***To move, to breathe, to fly, to float,
To gain all while you give,
To roam the roads of lands remote,
To travel is to live.***

- Hans Christian Andersen

So all poems have the same requirements.

If you use `id` attribute, you can instruct the stylesheet to style each poem in a particular way. It will look something like (we will learn how to write proper styles in the next chapter, so for now we will just phrase it in English) -

'Make poem1's text color grey, italic and bold'

'Make poem2's text color grey, italic and bold'

'Make poem3's text color grey, italic and bold'

Can you imagine how repetitive your style sheet will look if you have to instruct it to do the same thing 20 times for different poem IDs? HTML makes it easier. We use the `class` attribute. Let's name this class of poems 'poetry'.

```
1. <p id="poem1" class="poetry">To move, to breathe, to fly,  
   to float...</p>  
2. <p id="poem2" class="poetry">Roses are red, violets are  
   blue...</p>
```

So now, all you have to do in your style sheet, is to instruct it to make all elements belonging to the 'poetry' class grey, italic and bold.

Knowledge check 2.2.1 (not graded)

0 points possible (ungraded)

```
<ol id="student-male" class="primary" start="10"></ol>  
<ol start="10" class="primary" id="student-male"></ol>
```

True or False? Both tags will give the same output. The order in which the attributes are specified in the opening tag does not matter.

☐ True

☐ False

Submit

© All Rights Reserved