# Global attributes: examples

# Global attributes: examples

## Global attribute: 'id'

Like we saw in the previous unit, here is a list of all the global attributes and the values they accept. To understand attributes, we considered an example of usage `id` and `class`. We are going to look at it in depth here and discuss another global attribute `title`.

The `id` attribute gives your element a unique identifier. In your HTML document, that ID value can only be used in one element.

Naming rules for `id` attribute:

- Must be of at least one character
- Should not contain any spaces
- Values are case-sensitive. This means 'QuestioN' and 'question' are NOT the same. That does not mean you can use two different IDs that only differ by case, i.e. "myid" and "MyId". They are different and so legal but extremely confusing!

```
<p id="question-about-html">How many times can a particular 'id' value be used in a HTML document?</p>
<p id="html-answer">Once</p>
```

`id` is primarily used for:

1. Styling your element. You can specify the style you want for the element in your style sheet by referencing the 'id'.

Using CSS, you can specify code that will give different styles to "question-about-html" (eg: black text, center-aligned) and "html-answer" (eg: green text)

2. Specifying a link target. We will be learning about hyperlinks later in this section. You can link to a section of your HTML page using the 'id' of the section. You should reference the 'id' value with a number sign preceding it - '#id-value'

```html
<a href="#introduction">1.1 Introduction</a> <!-- This is a hyperlink element which we will learn about later in this week -->
<p id="introduction">This paragraph is the Introduction to the Web page</p>
```

3. In JavaScript, 'id' can be used to manipulate an html element. Using the 'id' of the element, you can write JavaScript code to make it perform an action, i.e. change the text within paragraph tags.

## Global attribute: 'class'

The `class` attribute, while similar to `id`, groups a set of elements in the same class. It's name-value pair is `class="classname"`. Unlike `id`, which is unique to the element, the same class name can be assigned to more than one element.

For example:

```html
<p class="question">What is your name?</p>
<p class="question">Do you like HTML5?</p>
```

Both paragraphs above are grouped under the class named 'question'. An element can have one or more class names. If we also want the second question to be under the 'html' class because it is a html related question, you can add two class names by separating them by space:

```html
<p class="question html">Do you like HTML5?</p>
```

Naming rules for `class` attribute:

- Must begin with a letter (a-z or A-Z)
- First letter can be followed by a letter, digit, hyphen or underscore
- Values are case-sensitive

`class` is primarily used for:

1. Styling your elements. You can specify the style you want for all the elements that belong to the class in your stylesheet.

```
<p class="question">Who are you?</p>
<p class="answer">I am the author</p>
<p class="question html">Do you like HTML5?</p>
<p class="answer">Yes</p>
```

In your CSS, you can include code to style your classes like this:

- 'question' class: text color is black and text is bold
- 'answer' class: text color is green
- 'html' class: text color is red

The code above will look like this:

**Who are you?**

I am the author

**Do you like HTML5?**

Yes

The 'Do you like HTML5?' question has styles for both classes 'question' and 'html' applied to it.

2. In JavaScript, `class` can also be used to manipulate html elements of the same class.

## Global attribute: 'lang'

The `lang` attribute indicates the language of the text in the element to which it is attached.  Identifying the language of content is increasingly important, as browsers adapt styling and other aspects of the user's experience according to the language of the content.

For example, if you create a page in Japanese, the browser will automatically apply a font that produces Japanese shapes for the characters, rather than Chinese shapes – but only if you have told it that your content is in Japanese. Various presentational aspects also require a knowledge of the language of the content: for example, CSS will style content differently for line-breaking, hyphenation, and text-transforms depending on the declared language. Other features, such as spell-checking and voice-browsers for visually-challenged people, also work differently according to which language the content is in.  For more details see Why use the language attribute?

The value of a `lang` attribute must be a language tag that is composed of one or more subtags defined in the IANA Language Subtag Registry. Multiple subtags are separated by hyphens.  (Do **not** use the ISO lists of languages and countries! Those lists are already subsets of the IANA registry.) You may find it easier to look up subtags using the unofficial Language Subtag Lookup tool.

You should **always declare the language of your page in the `<html>` tag**.  You can also declare the language of content within the page by attaching a `lang` attribute to an element that surrounds it.

For example:

```
<html lang="en-GB">...</html>
<p>In French you'd say <span lang="fr">On voit souvent des
chats sur le Web.</span></p>
```

The first example above shows how you can qualify the language (English) with a region subtag (GB) to specify British English.  This distinction can be useful for spellchecking your source. You can also add other subtags, such as scripts and variant labels to further refine the language. However, the golden rule is to always keep the `lang` value as short as possible, and only use additional subtags when you have a good reason (ie. use just ja for Japanese, not ja-JP). For more information, see the article Declaring language in HTML.

The second example shows how you could specify a change of language within the document. This would help a voice browser pronounce the French word 'chats' correctly, meaning 'cats' and not 'chats' in English.

## Global attribute: 'title'

Try this: Place your cursor on the word and then on the picture below. Don't click on it, just rest your cursor there.

**NASA**

Did you see the two secret messages? A message that appears when you point your cursor at something is called a tooltip. Be it a paragraph, header, image or any element, the title attribute is used to provide additional information about it. It is very useful to elaborate abbreviations or add some context. For images, you must use an `alt` attribute as there is no guarantee that the `title` attribute is presented to assistive technology users. The title can be of any text value.

```
<abbr title="National Aeronautics and Space
Administration">NASA</abbr>
```

## Knowledge check 2.2.3 (not graded)

0 points possible (ungraded)

```
<p id="greeting" class="hello world">This is me greeting the world</p>
```

Which of the following is the correct behavior with respect to the code above?

○ Both the classname and paragraph text cannot contain "world"

● Two different classes "hello" and "world" will be applied to the paragraph ✓

○ The code is invalid because space is not allowed in class attribute's value

○ One class "hello world" will be applied to the paragraph

Submit