Accessibility guidelines for HTML and CSS

(to ensure WCAG 2.0 compliance)

Date	Version	Author	Status / comments
07 February 2014	1.0	Atalan	English version.

In partnership with:
Air Liquide – Atos – BNP Paribas – Capgemini – EDF – Generali – SFR – SNCF – Société Générale – SPIE

Observers:
AbilityNet *(UK)* – Agence Entreprises & Handicap – AnySurfer (*Belgium*) – Association des Paralysés de France (APF) – Association Valentin Haüy (AVH) – CIGREF – Design For All Foundation (*Spain*) – ESSEC – Handirect – Hanploi – Sciences Po – Télécom ParisTech

Acknowledgements

Special thanks to our AcceDe Web partners for their input and commitment:

AIR LIQUIDE Atos BNP PARIBAS Capgemini
EDF GENERALI SFR SNCF
SOCIETE GENERALE SPIE

Thanks to their support, participation in the working groups, proofreading of documents, and testing of manuals, the partner companies have provided valuable input to the Accede Web project. This has made it possible to create manuals that successfully meet the needs of the different stakeholders of a web project. These companies are also contributing to making the web more accessible by authorizing the free distribution of this content.

We would also like to thank the following observers of the AcceDe Web project whose actions raise awareness of the AcceDe Web manuals and aid their distribution: AbilityNet (*UK*), Agence Entreprises & Handicap, AnySurfer (*Belgium*), Association des Paralysés de France, Association Valentin Haüys, CIGREF, Design for All foundation (*Spain*), ESSEC, Handirect, Hanploi, Sciences Po and Télécom ParisTech.

We thank all the members of the review committees for the quality and relevance of their comments:

- English version: Robin Christopherson (UK - AbilityNet), Ramon Corominas (Spain - Technosite), Deborah Edwards-Onoro (USA - Lireo Design), Thomas Frandzen (Denmark - Agency for Digitisation, Ministry of Finance), Char James-Tanny, (USA - a11yBos), Matt May (USA – Adobe), Sophie Schuermans (Belgium – Anysurfer), Jared Smith (USA - WebAIM), Christophe Strobbe (Germany), Gijs Veyfeyken (Belgium – Anysurfer)

- French version: Benjamin Ach (accessibility expert), Vincent Aniort (APF), Jean-Baptiste Audras (University of Grenoble), Claire Bizingre (accessibility consultant), Victor Brito (accessibility consultant), Anna Castilla (Provaltis), Ève Demazière (Sciences Po), Nicolas Fortin (French Ministry of Media and Culture), Marc-Étienne Vargenau (Alcatel-Lucent) and Éric Vidal (Fédération des aveugles de France), as well as all the teams of our partner companies.

Finally, we would like to thank Laurent Bracquart and Johan Ramon, Christophe Pineau and Marion Santelli of Atalan for their dedication and commitment to this initiative.

Sébastien Delorme, Sylvie Goldfain,
Atalan

Contents

Introduction

Context and objectives

This manual brings together all the accessibility specifications to consider when integrating HTML/CSS in a website or web application (HTML templates, CMS templates, etc.), whatever the target device for displaying the content (computer, mobile phone, etc.) to ensure WCAG 2.0 compliance.

This manual is part of a set of four complementary manuals that can be downloaded from the http://www.accede-web.com/en/ website:

1. Accessibility guidelines for graphic design.
2. **Accessibility guidelines for HTML and CSS (this manual).**
3. Accessibility guidelines for rich interfaces and JavaScript.
4. Accessibility guidelines for editors (template).

Warning
If the HTML/CSS development is made entirely from graphical mock-ups, the latter must respect the recommendations in the accessibility guidelines for graphic design manual. On the other hand, if the HTML/CSS development is made partially or entirely without mock-ups, it is the Front-End Web Developer who must make sure that the recommendations for graphic design are respected.

Who should read this user guide, and how should you use it?

This document should be given to the stakeholders and/or service providers who create the technical specifications, and the HTML/CSS templates, and who are responsible for the different technical developments of interfaces. It should be used in addition to the technical specifications of a project. The recommendations may be supplemented with others, or left out, according to the circumstances—the project manager is often the most appropriate person for this task.

The recommendations should be considered for the HTML/CSS development phase. Some recommendations apply when pages are produced dynamically from templates that are developed in a content management system (CMS).

There are some annotations that complete the document and should be read to understand each recommendation:

- **Note**: notes help complete the recommendations by providing additional details for specific situations.
- **Warning**: warnings highlight specific points that require attention or traps to avoid in order to guarantee good accessibility.
- **Tip**: Tips are not directly linked to accessibility, but you can improve the general quality of the interfaces by implementing them, or facilitate the integration of accessibility in subsequent steps in the project. Note that the recommendations in this project, although aimed at accessibility, are often good practice for ensuring ease of use, and improved user experience, performance, and referencing.
- **HTML5:** The HTML5 insets provide additional details or warnings to the recommendations in the light of the developments proposed in the new HTML specification.

License agreement

Contact

Please send any comments about this document to Atalan, the coordinator of the AcceDe Web project, at the following email address: accede@atalan.ca.

You can also find more information about the AcceDe Web project procedural manuals on the website http://www.accede-web.com/en/ or follow our twitter account @societe_atalan.

Credits

The icons used in the AcceDe Web guidelines are from a set of *24x24 Free Application Icons* (http://www.small-icons.com/packs/24x24-free-application-icons.htm).

The screen captures of the publishing tool illustrating good practices are taken from WordPress (http://wordpress.org).

The screen captures of content are taken from the following websites, on the date of October 10 2012.

- http://www.bnpparibas.com/en

[1] For more information on the Creative Commons BY 3.0 license, see http://creativecommons.org/licenses/by/3.0/fr/.

- http://uk.atos.net/en-uk/home.html
- http://www.generali.com/generalicom/home.do
- http://uk.voyages-sncf.com/en/
- http://www.areva.com
- http://www.societegenerale.com/en?_force=1
- http://www.rff.fr/en/

1. General structure

1.1. Regions, headings and navigation menus

1.1.1. Identify the main regions of the page with the role attribute

The main regions of the page must be identified with the ARIA attribute `role` (also referred to as a landmark).

This attribute has several different values to define the regions, including:

- `role="banner"` to define a content region that identifies the site, such as a banner with a logo, a slogan and/or a search field.
- `role="search"` to define a region for searching, such as a search form.
- `role="navigation"` to define menus or main navigation elements.
- `role="main"` to define the main content region of the page.

The `role` attribute may be added to any HTML tag such as `<div>`, `<form>`, `<nav>`, etc.

Warning
A page can only have one `role="main"`.

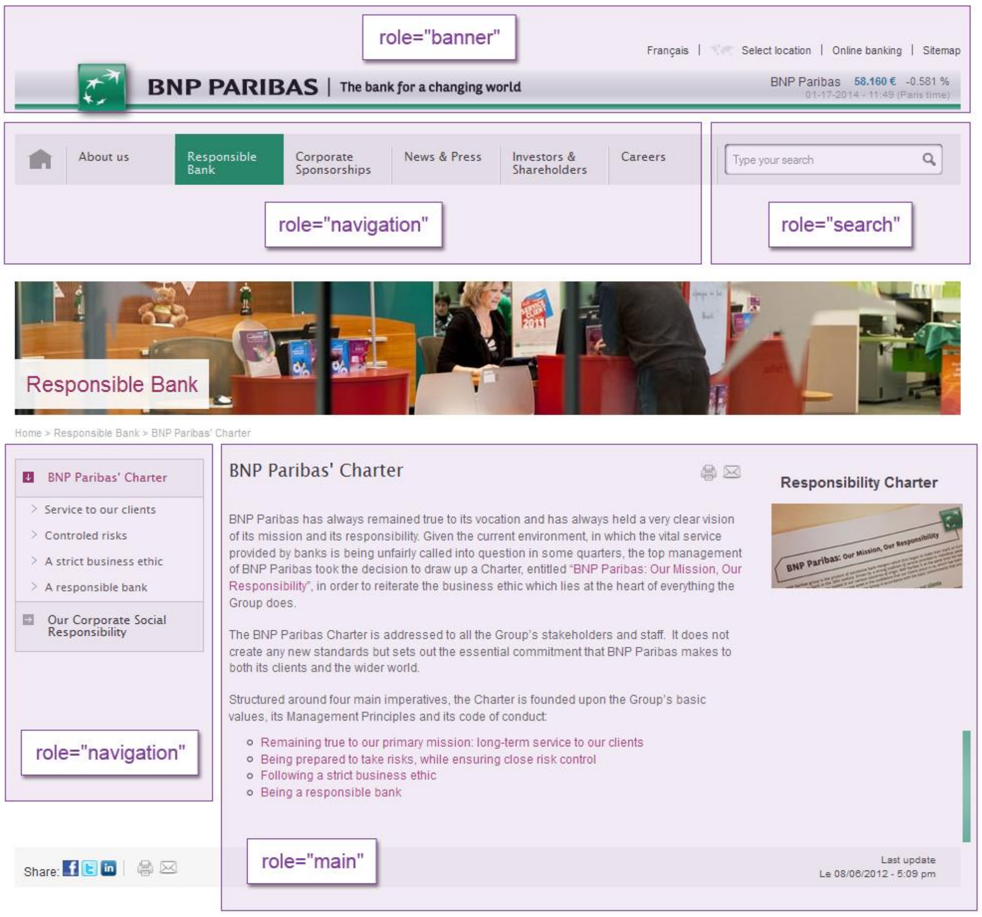
Note

- It is quite possible to nest several landmarks `ARIA: role=”search”`, one inside the other, in `role=”banner”`, for example.
- You are advised not to overload the page with `role=”navigation”` attributes. It is not necessary to identify each pagination system or each link; only the main menus must be identified.

HTML5
In HTML5, it is a good accessibility practice to systematically add `role="navigation"` to each `<nav>` tag.

```
<nav role="navigation">
    […]
</nav>
```

This temporary recommendation is valid until the development of HTML5 is finalized, and it is properly implemented by browsers and assistive devices.



BNP Paribas' Charter

BNP Paribas has always remained true to its vocation and has always held a very clear vision of its mission and its responsibility. Given the current environment, in which the vital service provided by banks is being unfairly called into question in some quarters, the top management of BNP Paribas took the decision to draw up a Charter, entitled "BNP Paribas: Our Mission, Our Responsibility", in order to reiterate the business ethic which lies at the heart of everything the Group does.

The BNP Paribas Charter is addressed to all the Group's stakeholders and staff. It does not create any new standards but sets out the essential commitment that BNP Paribas makes to both its clients and the wider world.

Structured around four main imperatives, the Charter is founded upon the Group's basic values, its Management Principles and its code of conduct:

- Remaining true to our primary mission: long-term service to our clients
- Being prepared to take risks, while ensuring close risk control
- Following a strict business ethic
- Being a responsible bank

Figure 1 : Example of ARIA role placement in a web page.

Note that the `role="navigation"` has been added to the main menu as well as the secondary menu.

1.1.2. Implement a logical and thorough hierarchy of headings using the tags <h1> to <h6>

To tag the headings on each page, use the heading tags from <h1> to <h6>, if required. The structure of the headings must be both logical and thorough.

This means that:

- There must be no "breaks" or inconsistencies in the structure of headings—no sudden change from an `<h1>` to an `<h3>` without an intervening <h2>, for example.
- All the elements that are used as headings must be tagged as headings.

Tip
To implement a thorough and logical hierarchy of headings, bear in mind that the headings form the "Table of contents" of the page. Ask yourself if the table of contents is logical and thorough.

Note

- It is perfectly acceptable to use several `<h1>` on a page if there are several headings of the same level.
- A best practice in accessibility is to avoid using hidden headings.

HTML5

Starting with HTML5, it is possible to implement a complete hierarchy of headings, on several levels, by using only the `<h1>` tag and then using the new tags designed for sections in the document, such as `<article>` or `<section>`. It is also possible to leave “breaks” in the structure of the headings.

Nevertheless, it is advisable, for the moment, to continue to structure the headings in a logical and thorough manner, as was done before the advent of HTML5. You should continue to use the headings `<h1>` to `<h6>` until the development of HTML5 is finished, and it is properly implemented in browsers and assistive devices.

Some detailed examples of heading structure are shown in the online version of this document at: http://wiki.accede-web.com/en/notices/html-css/.

1.1.3. Structure navigation menus with lists

Use unordered lists `<ul>` and `<li>` to tag the navigation menus.

With menus containing more than one level, make sure you nest the lists into each other properly.

```
<ul id="main-menu" role="navigation">
    <li><a href="page-1.html">First menu item </a></li>
    <li>
        <a href="page-2.html">Second menu item</a>
        <ul>
            <li><a href="page-3.html">First sub-menu item</a></li>
            <li><a href="page-4.html">Second sub-menu item</a></li>
        </ul>
    </li>
    <li><a href="page-5.html">Third menu item</a></li>
</ul>
```

HTML5

In HTML5, you should also wrap the main navigation menus with a `<nav>` tag, so that you have:

```
<nav id="main-menu" role="navigation">
    <ul>
       […]
    </ul>
</nav>
```

1.2. Order of content in the HTML source code

1.2.1. Write the HTML source code by following the logical reading order

The order of writing the tags in the HTML source code must follow the logical reading order of the page.

This means that if a tag immediately precedes another tag when visually browsing over the page, the first tag must also immediately precede the other in the HTML flow.

Tip
To test this criterion, you just need to deactivate the CSS and ensure that the resulting output corresponds to the reading order of the page when CSS is activated.

Warning
If content is hidden by default, make sure that it is positioned properly in the HTML flow when the styles are deactivated.

1.2.2. Ensure the consistency of the HTML order from one page to the next

The order of developing the main blocks in the HTML flow must be consistent from one page to another throughout the site.

For example, if the secondary menu is placed after the main menu, it is important to keep this order on all pages of the website.

2. Metadata

2.1. Page title

2.1.1. Enter an accurate <title> on each page

You must enter a well-defined `<title>` tag on each page.

It must at least include the name of the current page as well as the name of the website.

Note
The name of the current page must be shown first in the `<title>` tag. You could for example have the title, `<title>[Name of current page] | [website name]</title>`.

Warning
Sometimes, pages are reloaded with modified content, following a user action. This is the case:

- When using filters, such as keyword clouds.
- When using pagination.
- When an expression is searched from a search form.
- Etc.

In this case, you need to update the title of the page to reflect what has changed. For example:

```
<title>Search results for "[searched expression]" (page 3/7) | [Name of
website]</title>
```

Tip
It is an accessibility best practice to make sure the order of content in the `<title>` tags on all pages in the website is consistent.

2.2. Character encoding

2.2.1. Ensure the correct encoding of all characters

In order to obtain the correct reproduction of text content, a declaration of the character set must be included on each page. In the following example, the `<meta>` tag is used.

```
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    […]
</head>
```

This `<meta>` tag must be inserted as high as possible in the HTML page, before any content that is displayed on the screen, and preferably immediately after the opening `<head>` tag.

Warning
Remember also to check the correct coding of characters that are not automatically displayed on the screen, especially:

- Content of the `title` attributes.
- Text that is hidden by default.
- Content of the `alt` attributes of images.
- Etc.

HTML5
HTML5 simplifies the writing of the `<meta>` tag for the declaration of the character set. If you choose this type of document, you just need to use a statement like: `<meta charset="UTF-8">`.

3. Languages

3.1. Page language

3.1.1. Enter the main page language with the lang attribute in the <html> tag

To ensure the correct reproduction of text content, a declaration of the main language must be included on each page. Use the `lang` attribute in the `<html>` tag for this.

For example, for a page in French, specify `<html lang="fr">`.

Tip

To enter the `lang` attribute, you need to use the two-letter ISO 639-1 code, or if it is not available, the three-letter ISO 639-2 code.

The main codes to be used are as follows:

- `fr` for French.
- `en` for English.
- `es` for Spanish.
- `de` for German.
- `it` for Italian.

The up-to-date, exhaustive, list of codes is maintained at the following address: http://www.loc.gov/standards/iso639-2/php/English_list.php.

3.2. Change of language

3.2.1. Use the lang attribute to indicate a change in the language

If content is included in a different language to the main language, then the foreign language must be indicated with the `lang` attribute.

For example, in the case of a page in French:

```
<ul>
    <li>English version</li>
    <li><a href="http://website.fr" lang="fr">Version française</a></li>
</ul>
```

Tip

If there is no specific tag that encompasses the content in a foreign language, then use the `<span>` or `<div>` tag and specify the `lang` attribute within it.

Note

Indicating a language change is not necessary for:

- Proper names.
- Words of foreign origin, but which are included in dictionaries of the main language.
- All words of foreign origin, but which are understood and pronounced correctly when spoken with the accent of the main language (“podcast” for example, if the main language is in French).

4. HTML grammar and semantics

4.1. Conformity

4.1.1. Write valid HTML code according to the grammar rules of the DOCTYPE used

On each page, a DOCTYPE must be used, and the HTML code must be valid according to the rules of the selected DOCTYPE. The choice of DOCTYPE is open.

It is especially important to:

- Correctly nest the tags.
- Ensure the corresponding closing tag for each opening tag.
- Avoid duplicate attributes for the same tag.
- Ensure the uniqueness of each id attribute within a page.

Note

Only validation errors are to be corrected. The warnings returned by the validator do not need to be taken into account with respect to this criteria, because they have no impact on accessibility.

Tips

- The list of official DOCTYPE's is available at this address: http://www.w3.org/QA/2002/04/valid-dtd-list.html.
- Use the W3C validator, available at the following address, to test the validity of your HTML code: http://validator.w3.org/.

Warning

Including ARIA attributes in the HTML code is a real advantage for accessibility but will invalidate the source code if the page is not produced in HTML5. If errors are returned following the inclusion of these attributes, they do not need correcting.

4.2. Usage of HTML tags and attributes

4.2.1. Do not use HTML attributes or tags designed exclusively for formatting

HTML tags and attributes designed exclusively for formatting should not be used:

- This includes the `align`, `alink`, `background`, `basefont`, `bgcolor`, `border`, `color`, `link`, `text`, `vlink`, `height` and `width` attributes.
- As well as the tags `<basefont>`, `<blink>`, `<center>`, `<font>`, `<marquee>`, `<s>`, `<strike>`, `<tt>` and `<u>`.

Use CSS to obtain the required visual effect.

4.2.1. Use HTML tags for their semantic value

HTML tags must be used for their intended purpose, not for the visual effect they provide.

For example:

Tags	Incorrect usage	Correct usage
`<blockquote>`	Indent text	Tag a block quote
`<ul>` and `<li>`	Display a bullet point	Tag lists
`<fieldset>`	Display a border	Group related form items
`<h1>` to `<h6>`	Display larger or smaller text sizes	Define heading levels in hierarchy

5. Images

5.1. Illustrative and decorative images

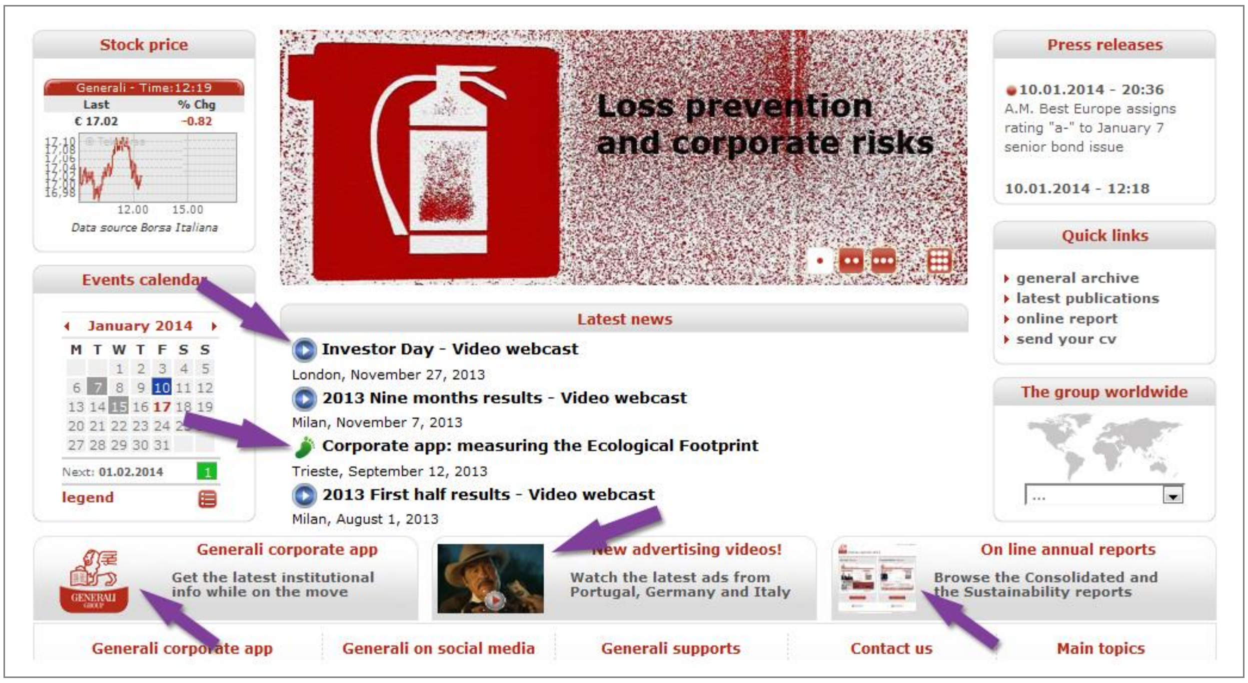
5.1.1. Use an empty alternate text (alt="") for decorative and illustrative images

When a decorative or illustrative image is included in the HTML code, the `alt` attribute must be added to the `<img/>` tag and left empty (without a space between the quotation marks of `alt=""`).

```
<img src="decorative-image.png" alt="" />
```

Tip
To check if an image is decorative/illustrative or informative, ask yourself if the absence of the image would change your understanding of the content. If the answer is "yes", then enter the alternate text; otherwise, leave the `alt` attribute empty.

Note
Whenever possible, decorative images should not be included in the HTML code but loaded into the page with CSS.



Stock price
Generali - Time:12:19
Last € 17.02 | % Chg -0.82
Data source Borsa Italiana

Loss prevention and corporate risks

Press releases
10.01.2014 - 20:36
A.M. Best Europe assigns rating "a-" to January 7 senior bond issue
10.01.2014 - 12:18

Events calendar
January 2014
Next: 01.02.2014
legend

Latest news
Investor Day - Video webcast
London, November 27, 2013
2013 Nine months results - Video webcast
Milan, November 7, 2013
Corporate app: measuring the Ecological Footprint
Trieste, September 12, 2013
2013 First half results - Video webcast
Milan, August 1, 2013

Quick links
general archive
latest publications
online report
send your cv

The group worldwide

Generali corporate app
Get the latest institutional info while on the move

New advertising videos!
Watch the latest ads from Portugal, Germany and Italy

On line annual reports
Browse the Consolidated and the Sustainability reports

Generali corporate app | Generali on social media | Generali supports | Contact us | Main topics

Figure 2: in this example, the decorative and illustrative images are indicated with the arrows. If they are included in the HTML code and are not clickable, then they should have an empty alternate text (alt="")

5.2. Informative Images

5.2.1. Do not use CSS to display informative images

CSS must **never** be used to display images that convey information.

In other words, each time there is an image that conveys information, it must be hard coded in the HTML, usually with the `<img />` tag.

Note
Using image *sprites* (images loaded in the background with CSS) is therefore not authorized for images that convey information.

Tip
All images whose absence would lead to a loss of content or functionality are considered to be informative:

- Logos.
- Text-images.
- Links and image-buttons.
- Etc.

Warning
Make sure you never use the technique which moves text outside the screen and replaces it with a background image.

When styles are not loaded in their entirety, or when they are modified by the user, the information may be lost.

```
<a href="/" id="logo">[text of my logo]</a>

a#logo
{
display: block;
background: url(images/logo.png);
text-indent: -999999px;
width: 300px;
height: 100px;
}
```

This technique should be replaced by using pure HTML. Subsequently, JavaScript can be used to dynamically modify the image on mouse-over, if required.

```
<a href="/" id="logo"><img src="images/logo.png" alt="[text of my logo]"
/></a>
```

If the image is not loaded, then the alternate text will be displayed instead, and the information will not be lost.

5.2.2. Enter the alt attribute for informative images

When an informative image is included in the HTML code:

- An `alt` attribute must be added to the `<img />` tag.
- This `alt` attribute must contain the same or equivalent information as that conveyed by the image.

Warning

You are strongly advised not to start the alternate text with the words `alt= "image […]"`. This information is implicit and may already be given to users of assistive devices.

Note: images that convey rich information

Sometimes, the image conveys a wealth of information that it is not possible to describe in simple text, just with the `alt` attribute (diagrams, complex graphs, etc.).

In this situation, it is an accessibility best practice to:

- Enter a **short alternate text** in the `alt` attribute that describes the basic purpose of the image.
- Plan for a content zone in which you can provide a **detailed description of the image**. Preferably, enter the detailed description either directly underneath the image or via a link to another page. For charts or complicated diagrams, the most appropriate detailed description is often a table of values, or a list.
- Indicate in the alternate text of the image how to access the more detailed description.

For example, in the case of a world map that shows the different countries in which a brand is present:

```
<img src="images/brand-map.png" alt="List of countries in which the
Spartacus brand is present (detailed description available by clicking on
the link underneath)" />

<a href="/zones-of-presence/">See the list of countries in which the
Spartacus brand is present</a>
```

In some guidelines, it is recommended using the `longdesc` attribute for the link to the detailed description of the image. However, the use of this attribute does not guarantee access to the information for everyone, unlike the inclusion of a hard-coded link on the page. The use of the technique above is therefore preferable.

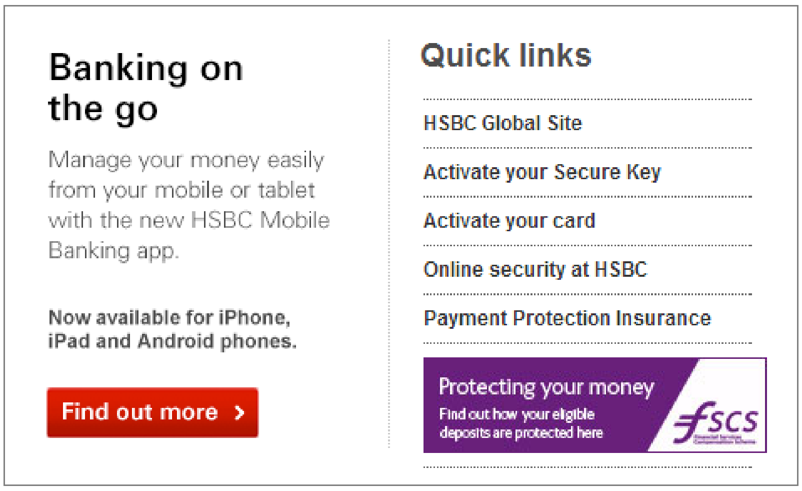

Banking on the go
Manage your money easily from your mobile or tablet with the new HSBC Mobile Banking app.
Now available for iPhone, iPad and Android phones.
Find out more
Quick links
HSBC Global Site
Activate your Secure Key
Activate your card
Online security at HSBC
Payment Protection Insurance
Protecting your money
Find out how your eligible deposits are protected here
FSCS

Figure 3: in this example, the informative image "Protecting your money" must be included in the HTML code and have an alternate text that includes the text in the image such as <img src="protecting.png" alt="Protecting your money, find out how your eligible deposits are protected here. FSCS (Financial Services Compensation Scheme)." />.

5.3. Image maps

5.3.1. Enter the alt attribute to each image map and corresponding <area /> tags

Whenever an image map is included in the HTML code:

- An `alt` attribute must be added to the `<img />` tag, as well as to each `<area />` tag.
- The `alt` attribute of the `<img />` tag must state the purpose of the image map.
- The `alt` attribute of each `<area />` tag must state the purpose of the link.

For example, imagine a map of the United States in which you can click on each state to obtain more information about the state:

```
<img src="images/us-map.png" alt="Map of the United States" usemap="#us-map" />

<map name="us-map">
    <area shape="poly" coords="[…]" href="alabama.html" alt="Alabama (AL)" />
    <area shape="poly" coords="[…]" href="alaska.html" alt="Alaska (AK)" />
    <area shape="poly" coords="[…]" href="arizona.html" alt="Arizona (AZ)" />
    […]
</map>
```

5.3.2. Include <map> and <area> tags just after each image map in the HTML code

Each time that an image map is used, the `<map>` and `<area>` tags must immediately follow the corresponding `<img />` tag in the order of the HTML code.

6. Links and buttons

6.1. Link and button texts

6.1.1. Write explicit texts for links and buttons

Make sure that link and button texts are as explicit as possible. In other words, the function of the link or button must be completely understood just by reading the text.

This text corresponds to the text contained:

- Between the `<a>` and `</a>` tags.
- Between the `<button>` and `</button>` tags.
- In the `alt` attribute for `<input type="image" />`.
- In the `value` attribute for `<input type="button" />` or `<input type="submit" />`.

So, rather than "OK" buttons or "Back" links, it is better to use:

```
<input type="submit" value="Register" />

<a href="basket.html">Return to step 1: "Your basket"</a>
```

Tip
To write explicit texts, you have to imagine that link or button texts are extracted from the page and read without context. Do they remain comprehensible? Do they give an idea of the content of the target page or the action that will be carried out?

Note
When text and images are shown together in the link or button text, the alternate text of the image should be considered as being an integral part of the text.

The following link will therefore be considered as explicit:

```
<a href="/latest-changes/">
    <img src="images/warning-pictogram.png" alt="Warning: " title="Warning!"
/>
    Make sure you take into account the latest changes to Canelis before
use.
</a>
```

Warning
Links and buttons must never be empty and must always have a text.

6.1.2. As a last resort, use the title attribute to make link and button texts explicit that cannot be made clear otherwise

Some link or button links cannot be made explicit (lack of available space on the screen, editorial choice, etc.). This is often the case for, "Read on" or "More information" type links.

In this context, the `title` attribute should be used to state the function of the link or button.

The value of the `title` attribute must both:

- Retrieve the same or equivalent text to that conveyed in the link or button text.

- Complete the information supplied by the link or button text.

For example, in the case of a link “July 2012”:

```
<a href="archives/2012/07/" title="July 2012 (read the month’s articles)">July
2012</a>
```

Warning
Using the `title` attribute to make a link or button text explicit will always be less accessible than the direct optimisation of the heading. The content of the `title` attribute is not displayed by default when navigating with the keyboard, or when using a touch screen, for example.

This attribute should therefore only be used as a last resort.

Tip
With “Read on” or “More information” type links, it is a good accessibility practice to move this information to the end of the `title` attribute content in order to make access to important information easier:

```
<a href="public-listing.html" title="Boursicota Public Listing (read
on)">Read on </a>
```

Note
The `title` attribute must also be used to differentiate links or buttons with texts that may be considered explicit, but which point to different pages or trigger different actions.

For example, in the case of two “Search” buttons on the same page, which therefore need to be differentiated:

```
<input type="submit" value="Search" title="Search news" />
[…]
<input type="submit" value="Search" title="Search for a person in the
directory" />
```

6.2. Download documents

6.2.1. Indicate the size and format of each document that can be downloaded

Each time that a link or button points to a document that can be downloaded; the following information must be included in the link text:

- Document name.
- Document format.
- Document size.

For example:

```
<a href="2011-annual-report.pdf">Download the 2011 annual report (PDF, 23 Mb)</a>
```

Tip

If symbols are used to visually indicate the format of documents that can be downloaded, make sure you include the image in the link text (hard-coded in the HTML code) and enter a value for the alternate text of the image.

For example:

```
<a href="2011-annual-report.pdf">
    Download the 2011 annual report (23 Mb)
    <img src="images/pdf-symbol.png" alt="(PDF)" title="(PDF)" />
</a>
```

Note

Sometimes, the text cannot be modified to include this information. In this case, and as a last resort, use the link or button `title` attribute to retrieve and complete the link or button text.

6.3. New windows

6.3.1. Indicate the opening of new windows

Each time that a link or button triggers the mandatory opening of a new window in the browser, the user must be forewarned:

- By adding a reference of type "(new window)", directly in the link text.
- Or by adding an image in the link text, with the alternate text duly entered.
- Or by adding a reference of type "(new window)" in the `title` attribute.

For example:

```
<a href="cgv.html" target="_blank">General terms and conditions (new window)</a>

<a href="cgv.html" target="_blank">
    General terms and conditions
    <img src="images/new-window-pictogram.png" alt="(new window)" title="(new window)"
/>
</a>

<a href="cgv.html" target="_blank" title="General terms and conditions (new
window)">General terms and conditions</a>
```

7. Forms

7.1. Labels and form controls

7.1.1. Use the <label> tag as well as the “for” and “id” attributes to associate form controls with their text labels

Each form controls (except buttons) must be associated with a label.

To do this, proceed as follows:

1. Use `<label>` to tag each text label.
2. Add a `for` attribute for each `<label>` tag as well as an `id` attribute for each form control.
3. Enter an identical value for the `id` and `for` attributes for each text label/form control pair.

```
<label for="name">Your name</label>
<input type="text" id="name" name="name" />

[…]

<label for="birth-year">Year of birth</label>
<select id="birth-year" name="birth-year">
    <option value="2012">2012</option>
    <option value="2011">2011</option>
    <option value="2010">2010</option>
    <option value="2009">2009</option>
    […]
</select>
```

Note
Sometimes, some form controls do not have a visible text label. In this context, use the `title` attribute to associate a label with a form control.

```
<input type="text" title="Your search" name="search" />
```

Note that the `title` attribute must not be used if the `for` and `id` attributes already have values to associate a text label with a form control.

Warning
It is important to use the same text labels for form controls with identical functions.

For example, if several authentication forms are present in a website, do not use the text label “Name” for one and “Login” for another.

7.1.2. Include information about expected field values directly in <label> tags

Information about the nature of expected values should be directly included in the `<label>` tags.

This is especially the case for:

- Statements of type “Mandatory field”.
- Indications that provide the format of values required; for example, “DD/MM/YYYY” for a date format.

- Indications that show the maximum size of files that can be sent.
- Etc.

```
<p>Fields marked with an asterisk (*) are mandatory.</p>

<label for="name">Your name *</label>
<input type="text" id="name" name="name" />

[…]

<label for="client-number">
    Your client number
    <input type="text" id="client-number" name="client-number" />
    <em>For example: 76432-BT-VZ</em>
</label>
```

Note

Sometimes, error messages are displayed at the level of each field. In this case, they must also be included in the associated `<label>` tags.

```
<p>Fields marked with an asterisk (*) are mandatory.</p>

[…]

<label for="email">
    Your email address *
    <input type="text" id="email" name="email" />
    <span class="error">Please enter your email address (expected format:
example@domain.us).</span>
</label>
```

7.2. Grouping and sorting information

7.2.1. Group and title fields of the same type with <fieldset> and <legend>

When a form proposes several groups of fields, some of which have identical text labels, use the `<fieldset>` and `<legend>` tags.

For example:

```
<fieldset>
    <legend>Participant 1</legend>

    <label for="first-name-1">First name</label>
    <input type="text" id="first-name-1" name="first-name-1" />

    <label for="Surname-1">Surname</label>
    <input type="text" id="surname-1" name="surname-1" />

    […]
</fieldset>

<fieldset>
    <legend>Participant 2</legend>

    <label for="first-name-2">First name</label>
    <input type="text" id="first-name-2" name="first-name-2" />

    <label for="surname-2">Surname</label>
    <input type="text" id="surname-2" name="surname-2" />
```

```
     [...]
</fieldset>
```

Warning

The `<fieldset>` and `<legend>` tags should only be used when several groups of fields have the same text labels. For example:

- A series of questions on the same page with the same possible answers, “yes” or “no”.
- A list of participants for an event with “First name” and “Surname” for each one.

In all other situations:

- Do not use the `<fieldset>` and `<legend>` tags.
- Use the `<h1>` to `<h6>` for block headings.

Note

It is a good accessibility practice to use the `<fieldset>` and `<legend>` tags when including lists of radio buttons or checkboxes on a page.

For example:

```
<fieldset>
    <legend>Title</legend>
    <ul>
        <li>
            <label for="Mrs">
                <input id="mrs" type="radio" name="title" value="Mrs." />
                Mrs.
            </label>
        </li>
        <li>
            <label for="Mr.">
                <input id ="mr" type="radio" name="title" value="Mr." />
                Mr.
            </label>
        </li>
    </ul>
</fieldset>
```

Detailed examples of using `<fieldset>` and `<legend>` with a form are shown in the online version of this document at: http://wiki.accede-web.com/en/notices/html-css/.

7.2.2. Order drop-down list options in a logical way

When drop-down lists are used, the options they contain must be ordered logically.

The logical order depends on the context, but it may be:

- Alphabetical order (a list of languages, for example).
- Numerical order (a list of rankings, for example).
- Practical order (“United Kingdom” in first place for a registration form for a British service).

Tip

It is a good accessibility practice to avoid using decorative characters (dashes, asterisks, spaces, etc.) as prefixes of the content of the `<option>` tag.

Their absence enables users to directly access the required value or group of values, by simply pressing the corresponding key on the keyboard (pressing the “I” key to go to “Italy”, for example).

Therefore:

```
<select>
    <option>--Austria</option>
    <option>--Belgium</option>
    <option>--England</option>
    <option>--France</option>
    <option>--Italy</option>
</select>
```

Should be replaced by:

```
<select>
    <option>Austria</option>
    <option>Belgium</option>
    <option>England</option>
    <option>France</option>
    <option>Italy</option>
</select>
```

7.3. Page title

7.3.1. Update the page <title> when a page returns an error or a confirmation message

Each time that a form returns an error or a confirmation message, the page `<title>` tag must be updated.

For example, in the case of success:

```
<title>Confirmation – Contact form | [Name of website]</title>
```

And in the case of error:

```
<title>Error – Contact form | [Name of website]</title>
```

Note

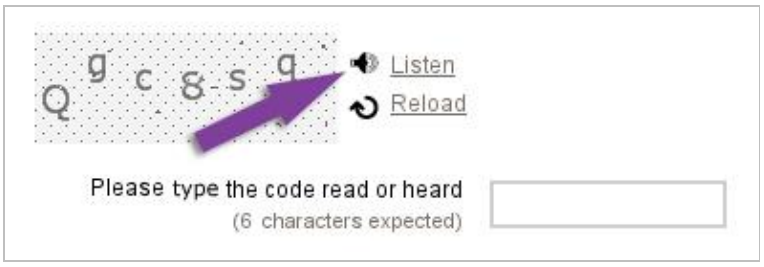
In some circumstances, it is not necessary to update the page title because the title displayed after submitting the form makes the result of the action obvious. For example:

- A connection form that sends the user to a “User profile” page.
- A “Go to the next step” button which sends the user to the next step in a form with multiple steps.
- A contact form that sends the user to a preview page.
- Etc.

7.4. CAPTCHA (anti-spam system)

7.4.1. Indicate in the alt attribute of the CAPTCHA image where the user can find the non-graphic CAPTCHA version

Whenever a CAPTCHA image is used, indicate where the non-graphic version of the CAPTCHA is found in the `alt` attribute of the image.



Often, the non-graphic version is an audio version. In this case you could have an alt text similar to this:

```
<img src="captcha.png" alt="CAPTCHA (antispam image). If you cannot see the image, use
the audio version available below." />
```

8. Lists and quotations

8.1. Lists

8.1.1. Tag unordered lists with <ul> and <li>

Use the `<ul>` and `<li>` tags to tag lists of items which can appear in any order (menus, tabs, sharing buttons, website map, etc.).

If required, make sure that lists are properly nested:

```
<ul>
    <li>First item of first level</li>
    <li>
        Second item of first level
        <ul>
            <li>First item of second level </li>
            <li>Second item of second level </li>
        </ul>
    </li>
    <li>Third item of first level </li>
</ul>
```

8.1.2. Tag ordered lists with <ol> and <li>

Use the `<ol>` and `<li>` tags to tag lists of items that must be shown in a specific order (list of steps in a procedure, ranking, etc.)—in other words when the information would not be understood if the items were entered in a different order.

8.1.3. Tag definition lists with <dl>, <dt> and <dd>

Use the `<dl>`, `<dt>` and `<dd>` tags to tag definition lists (in a glossary, for example).

You can associate several definitions with a single term by following the procedure below:

```
<dl>
    <dt>Term 1</dt>
    <dd>Definition of term 1.</dd>
    <dt>Term 2</dt>
    <dt>1. First definition of term 2.</dt>
    <dt>2. Second definition of term 2.</dt>

</dl>
```

HTML5

The HTML5 specification extends the role of the standard definition list to a broader one of a list of descriptions. With HTML5 you can use `<dl>`, `<dt>` and `<dd>` to tag any groups of key/value pairs (product sheet, publication details of an article, etc.).

```
<h2>Conference on sustainable development </h2>
<dl>
    <dt>Place</dt>
    <dd>Paris</dd>
    <dt>Date</dt>
    <dd>Saturday 29 September 2012</dd>
    <dt>Time</dt>
    <dd>From 10:00 a.m. </dd>
</dl>
```

8.2. Quotations

8.2.1. Tag quotation blocks with <blockquote>

Whenever a quotation block is to be included in an HTML page, wrap it in a `<blockquote>` tag.

Note
Any isolated quote, which is not directly included inline within the flow of surrounding text, is considered as a block quote.

9. Tables

9.1. Formatting tables

9.1.1. Ensure the correct reading order of formatting tables

When a person uses a screen reader to access a table used for formatting, the content is read line by line. This means that the table content is read cell after cell, from left to right, and row after row.

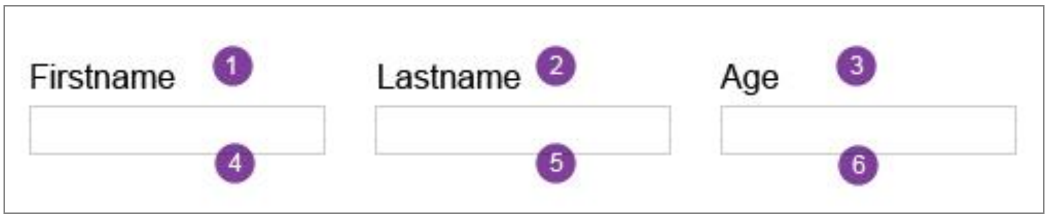
Therefore, you need to make sure that the reading order is logical for each formatting table.

For example, if the following source code is used, the reading order is:

1. First name.
2. Surname.
3. Age.
4. “First name” field.
5. “Surname” field.
6. “Age” field.

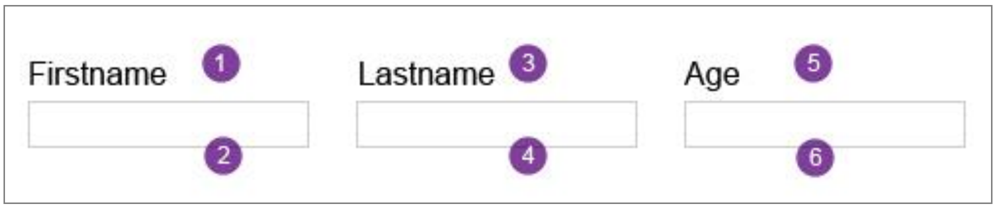
```
<table>
    <tr>
        <td>
            <label for="first-name">First name</label>
        </td>
        <td>
            <label for="surname">Surname</label>
        </td>
        <td>
            <label for="age">Age</label>
        </td>
    </tr>
    <tr>
        <td>
            <input type="text" name="first-name" id="first-name" />
        </td>
        <td>
            <input type="text" name="surname" id="surname" />
        </td>
        <td>
            <input type="text" name="age" id="age" />
        </td>
    </tr>

</table>
```



To obtain a logical reading order while still using a formatting table, it is preferable to use the following code:

```
<table>
    <tr>
        <td>
            <label for="first-name">First name</label>
            <input type="text" name="first-name" id="first-name" />
        </td>
        <td>
            <label for="name">Surname</label>
            <input type="text" name="name" id="name" />
        </td>
        <td>
            <label for="age">Age</label>
            <input type="text" name="age" id="age" />
        </td>
    </tr>
</table>
```



Warning
In order to facilitate reading by assistive devices and to ensure a consistent reading order, you are strongly advised to limit the nesting of formatting tables.

9.1.2. Do not use tags or attributes that are specific to data tables in formatting tables

A formatting table should not include tags or attributes that are specific to data tables.

In other words, the:

- `<caption>`, `<th>`, `<thead>` and `<tfoot>` tags must not be used in formatting tables.
- `scope`, `headers`, `axis`, `colgroup` and `summary` attributes must not be used in formatting tables.

9.2. Data tables

9.2.1. Provide a title for each data table with the <caption> tag

Whenever a data table is included in an HTML page, it must have a concise and clear title. The title must be tagged with `<caption>` and must introduce the content of the table.

```
<table>
    <caption>Average monthly temperatures of the 3 largest French cities in 2012.</caption>
    [...]
</table>
```

9.2.2. Tag each row header and column header cell with <th>

In each data table, tag each row and column header cell with the `<th>` tag. In other words each time that a cell is used for understanding the data shown in a table, this must be tagged with `<th>`.

9.2.3. Use the scope attribute to associate cells with the headers in simple tables

A simple data table is a table in which the header cells apply systematically to all the data cells in the corresponding row or column.

To associate the headers with the corresponding data in this type of table, use the `scope` attribute on the `<th>` tags. The value of this attribute will change according to whether the header cell concerns:

- The entirety of a column: `scope="col"`.
- The entirety of a row: `scope="row"`.

```
<table>
    <caption>Average monthly temperatures of the 3 largest French cities in
2012.</caption>
    <tr>
        <td>&nbsp;</td>
        <th scope="col">Paris</th>
        <th scope="col">Marseille</th>
        <th scope="col">Lyon</th>
    </tr>
    <tr>
        <th scope="row">June</th>
        <td>22°C</td>
        <td>28°C</td>
        <td>26°C</td>
    </tr>
    <tr>
        <th scope="row">July</th>
        <td>24°C</td>
        <td>30°C</td>
        <td>28°C</td>
    </tr>
</table>
```

9.2.4. Use the headers and id attributes to associate cells with table header cells in complex data tables

A complex data table is a table in which the header cells do not systematically apply to the entirety of row cells or column cells containing data.

To associate the headers with data in this type of table, use the `id` (identifiers) attribute on the `<th>` cells and headers attribute on the `<td>` cells.

You then need to enter the identifiers of the associated headers in the `headers` attribute of data cells. If several headers are associated with one data cell, the identifiers must be separated by a space in the `headers` attribute.

```
<table>
    <caption>Sales revenue comparison of Davis and Davison companies in the UK and in
the rest of the world</caption>
    <tr>
        <th id="header-one">In millions of pounds sterling</th>
        <th id="header-two">In the UK</th>
        <th id="header-three">ROW</th>
    </tr>
    <tr>
        <th id="header-four">Davis</th>
        <td headers="header-four header-two header-one">50.7</td>
        <td headers="header-four header-three header-one">139.3</td>
    </tr>
    <tr>
        <th id="header-five">Davison</th>
        <td headers="header-five header-two header-one">27.1</td>
        <td headers="header-five header-three header-one">476.0</td>
    </tr>

</table>
```

Warning

The `headers` and `id` attributes must not be used in combination with the `scope` attribute.

Note

It is a good accessibility practice to respect the logical order when including the `id` attributes of associated header cells in the `headers` attribute of data cells.

A screen reader will read these values in the order they are entered.

10. CSS usage

10.1. Hidden content

10.1.1. Do not use the CSS properties display or visibility to hide content that cannot be displayed by the user

There are two possible scenarios for elements that are present in the HTML code, but are not displayed on the screen by default:

Scenario 1: elements are hidden but can be displayed by a user action

This scenario includes, for example, the content of a pop-in, a drop-down menu, or the different panels in a dynamic carousel.

To find out how to use make these elements accessible, refer to the document "Accessibility guidelines for rich interfaces and JavaScript".

Scenario 2: elements are hidden but will never be displayed

Hidden elements that will never be displayed can be divided into two categories:

Category 2.1: elements that serve no purpose for users

Elements that serve no purpose should preferably be:

- Deleted from the HTML code,
  or if that is not possible,
- Hidden with the CSS properties `display: none;` or `visibility: hidden;`.

Category 2.2: elements that are useful for users of screen readers, but serve no purpose for other users

Elements that are useful to screen reader users should be made invisible or moved outside the screen with the CSS properties `opacity: 0;` or `position: absolute;` for example.

These elements include, for example, indications that are useful for navigating in pages, skip links, etc.

⚠ **Warning**
Whenever links, or more generally interactive links, are included in content that is targeted at users of screen readers, use `tabindex="-1"` on these elements in order to prevent access with the standard keyboard method.

For example:

```
<a href="alternative-interactive-map.html" tabindex="-1" class="a11y">Users
of screen readers, please follow this link to access the alternative version
of the interactive map.</a>
```

```
.a11y
{
position: absolute;
left: -999999px;
}
```

10.2. Changing the appearance and size of text

10.2.1. Use CSS to format text

Use CSS to format text. Logos and other distinctive elements that belong to a company's corporate charter can be excluded from this recommendation. Do not use text-images except as a last resort.

Note
Special attention should have been paid to the feasibility of this point from the start of the graphic design stage.

If this did not happen, refer to the section "2.1.3. Make sure that typefaces can be integrated in text format" in the Accessibility guidelines for graphic design manual and make contact with the person responsible for creating the mock-ups.

Tip
If required, do not hesitate to use the property `CSS3 @font-face`.

Warning
Technologies such as cufón and sIFR do not meet the demands of accessibility requirements, and should not be used for an accessible website or application.

10.2.2. Only use relative sizes (em, %, small, etc.) for typefaces

To define the size of character fonts, only use relative units for the `CSS font-size` property, such as `em`, `%`, `px`, `rem`, and the keywords `x-small`, `small`, etc.

Do not use absolute unit sizes, such as `pt`, `cm`, etc.

Warning
Due to its poor support in Internet Explorer, using pixels (`px`) as a unit of font size is not authorized in some guidelines such as the French guidelines AccessiWeb or RGAA. To ensure conformity with these guidelines, you must use a different unit of size for the `font-size` property.

10.2.3. Ensure the readability of content even if the text size is doubled

Make sure that the content is readable, even if the user doubles the size of the text in the browser. Therefore you need to prevent the overlapping of text, text disappearing off the screen, etc.

To ensure that this recommendation is respected as far as possible, you are advised:

- Not to use units (`px`, `pt`, `%`, `em`, etc.) with the property CSS `line-height`.
- Not to define a fixed height for elements that are likely to have text content, especially form fields.
- Not to define the width of elements in `em`.

Tip

To test this criteria:

- With **Firefox**, select the menu *View*, select *Zoom* and then select *Zoom text only*, and then double the size of the text by clicking 6 times with the keyboard shortcut CTRL and + (CTRL and 0 returns the size to the default size).
- With **Internet Explorer**, in the *View* menu, select *Text size*, then select *Largest*.

10.3. Access to information

10.3.1. Make sure content is readable when images are not displayed

When images are not displayed, the text content of the page must remain visible and readable. No information should be lost, and the contrast between the text colour and the background colour should be sufficiently high (refer to the Accessibility guidelines for graphic design).

In other words, content should be readable:

- Even when the images included in CSS are not loaded into the page.
- Even when the images included in HTML are replaced by the content of the `alt` attribute.

Tip

Whenever text is superimposed on a background image, make sure that you include a replacement colour that ensures the text is readable even if the background image is missing

```
background: black url(../images/dark-background.png) repeat-x;
```

This replacement colour can be inherited from a parent element.

Warning

Special attention should be paid to the readability of the alternate text of images integrated in the HTML code, when the images are not displayed.

An earlier recommendation in this manual advised leaving out the `height` and `width` attributes in the HTML code, which ensures that the alternate text is displayed.

11. Navigating with the keyboard

11.1. Tab order

11.1.1. Make sure the tab order follows the logical reading order

The tab order must follow the logical order of reading the page.

This means that when an interactive element immediately precedes another during a visual browsing of the page, the focus must move to the second element immediately after having left the first.

Warning
Except in very rare cases where the interface is totally stable and controlled for a long period of time, make sure you do not use the `tabindex` attribute with a value of more than `0`; otherwise, there is a risk of upsetting the tab order logic in the page.

11.1.2. Include skip links

A skip link of the type "Go to content" should systematically be included on each page in order to make navigating with the keyboard easier.

This link must be the first interactive element in the HTML code. A skip link links to an anchor on the same page and lets the user skip directly to the main content of the page.

```
<a id="skip" href="#content">Go to content</a>

[…]

<div id="content" role="main">
    […]
</div>
```

Warning
While it is strongly recommended to display this link, it can be hidden by default, if it is not planned for in the graphical mock-ups. On the other hand, it must always be made visible when the keyboard focus is set on it.

The skip link must never be hidden using the CSS properties `display: none;` and/or `visibility: hidden;` otherwise it will not be reachable with the keyboard.

Preferably, you should use another solution, such as the one shown in the code below:

```
a#skip
{
position: absolute;
left: -999999px;
}

a#skip:focus
{
position: static;
}
```

Note

Sometimes the user needs to press the TAB key many times to access the main menu and/or the search engine from the top of the page.

Consequently, it is a good accessibility practice to include a list of skip links.

```
<ul id="skip">
    <li><a href="#content">Go to content</a></li>
    <li><a href="#menu">Go to the menu</a></li>
    <li><a href="#search">Go to search</a></li>
</ul>
```

11.2. Visibility of the focus position

11.2.1. Ensure visibility of focus when set from the keyboard

To help users who navigate with the keyboard know where they are on the page, each interactive element must be visually highlighted when the focus is set on it (links, form fields, buttons, etc.).

Tip

It is a good accessibility practice to duplicate each `:hover` rule with a `:focus` rule in the CSS.

Warning

The `outline: none;` rule must never be applied by default to the focus in the CSS, except in the specific case where the visibility of the focus is then overwritten later in the style sheet.

If the styles applied to the focus position are considered to be intrusive when navigating with a mouse, it is possible to cancel them with a mouse-click, by using the pseudo-class `:active`.

For example:

```
:focus
{
outline: 3px dotted green;
}

:active
{
outline: none;
}
```

12. Embedded content

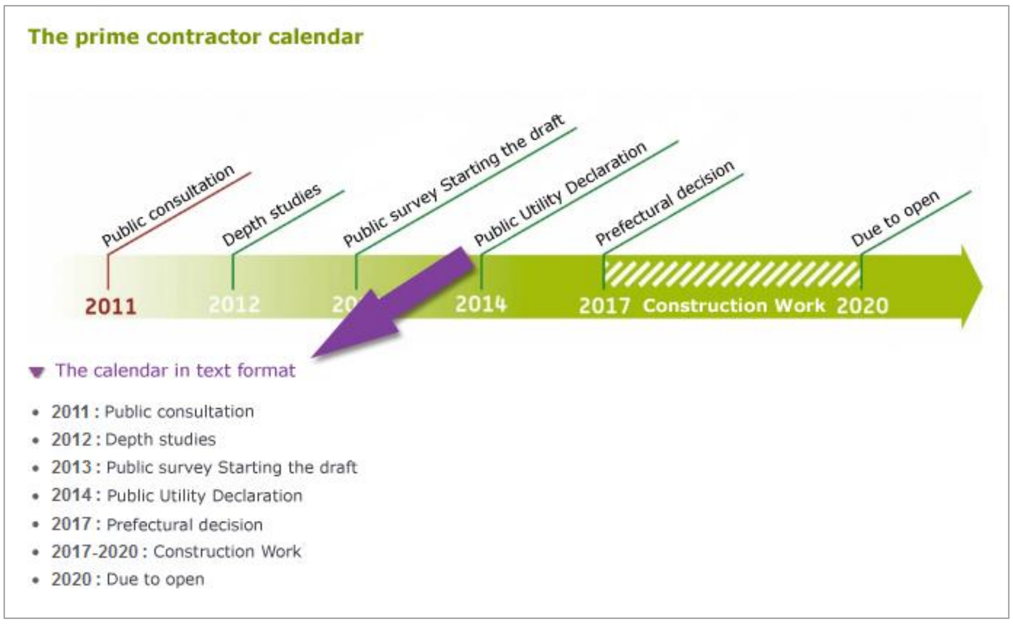
12.1. Multimedia content

12.1.1. Plan for an alternative to each multimedia content (<video>, <audio>, <object>, etc.)

Make sure there is an alternative to each multimedia content embedded in the page with tags such as `<video>`, `<audio>`, `<object>`, etc.

This alternative will provide the same level of information to users who cannot access the multimedia content (missing plugin, old browser, etc.). It may take the form of HTML content, or an image with alternative content, etc.

Tip
When a video is displayed on a page, the transcript may be considered as an alternative. The alternative could consist in providing a video that can be downloaded (in addition to the transcript).


The prime contractor calendar

Public consultation
Depth studies
Public survey Starting the draft
Public Utility Declaration
Prefectural decision
Due to open
2011
2012
20
2014
2017 Construction Work 2020

▼ The calendar in text format

- 2011 : Public consultation
- 2012 : Depth studies
- 2013 : Public survey Starting the draft
- 2014 : Public Utility Declaration
- 2017 : Prefectural decision
- 2017-2020 : Construction Work
- 2020 : Due to open

Figure 4 : The calendar above is a Flash animation that conveys information.
A "Calendar in text format" link allows the user to display an alternative in HTML format.

12.1.2. Do not use the wmode parameter with the values transparent or opaque

When an animated Flash file is embedded in the page, the `wmode="transparent"` or `wmode="opaque"` parameter must not be used, because it is likely to make the animations unusable from the keyboard.

Appendixes

Appendix 1: Additional Recommendations...

With the aim of being pragmatic and quickly operational, a few accessibility recommendations of the WCAG have not been kept in the core of this document. These criteria are rarely applicable in a web project or are of low priority. Nevertheless, they are listed below and will be described in the Accede Web project wiki[2].

...to comply with level A (WCAG 2.0)

- Tag inline quotes with `<q>`.
- Include the summary and structure of each data table with the `summary` attribute.
- Enter a value for the `title` attribute for each `<iframe>`.
- Enter a value for the `title` attribute for each `<frame>`.
- Group options of the same type with the `<optgroup>` tag in the `<select>`.
- Indicate the language of each download document written in a foreign language.
- Make sure images with an `ismap` attribute have duplicate alternative links.
- Enter the main reading order of the page with the `dir="ltr"` attribute (left to right) or `dir="rtl"` attribute (right to left) on the `<html>` tag.
- Use the `dir` attribute to indicate any changes to the reading direction in the body of the page.
- Identify the main zones of the page with the `id` attribute or a named anchor.

...to comply with level AA (WCAG 2.0)

There are no additional recommendations for level AA (WCAG).

...to comply with level AAA (WCAG 2.0)

- Include explicit link headings that are comprehensible without context and without using the `title` attribute.
- Tag definitions with `<dfn>`.
- Tag abbreviations with `<abbr>`.
- Tag acronyms with `<acronym>`.
- Limit the width of text to 80 characters, whatever the size of the text.
- Include line spacing of at least 1.5 times the text size, whatever the size of the text.
- Include paragraph spacing of at least 1.5 times the value of the line spacing, whatever the size of the text.
- Ensure the absence of a horizontal scrollbar even when the size of the text is doubled by the user.
- Ensure that the line length is less than 80 characters when the user reduces the width of the browser window.

[2] http://wiki.accede-web.com/en/notices/html-css/recommandations-additionnelles

Appendix 2: WCAG 2.0 correspondence table

The set of recommendations presented in this document is drawn from the WCAG 2.0.

In order that readers can read this document and compare the recommendations with the level of the standards (A, AA, AAA) a correspondence table between the AcceDe Web HTML/CSS recommendations and the criteria of WCAG 2.0 is maintained in the project wiki at the following address: http://wiki.accede-web.com/en/notices/html-css/grille-de-correspondances-wcag-accessiweb-rgaa.