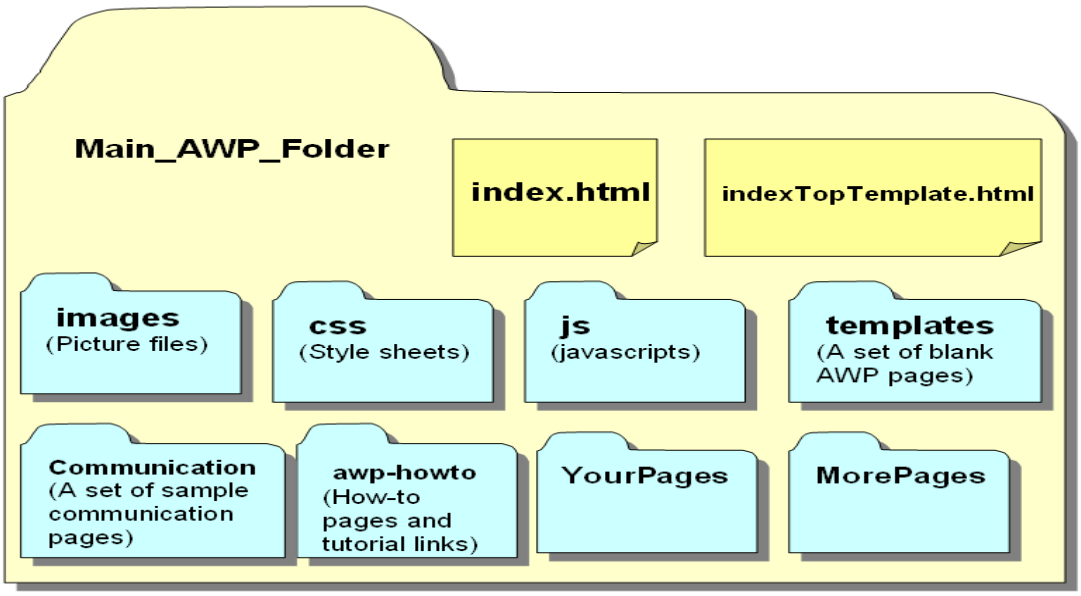
Tips for Defining Resource Addresses in AWP Pages

Almost any web page has links that tell the browser where to find resources like picture files, and to also identify the address of another web page you wish to link to. There are two basic methods that any browser understands when using a URL to reference a resource file: *absolute addressing* and *relative addressing*. You will rely on both methods to build a set of web pages using the AWP resources. When you visit a web page the absolute address always appears in the address field at the top of the browser window (except when viewing things like browser settings.)

But, before I tell you which method to use and when, we need to look at how the AWP web site is organized as a set of folders and files:



CSS files provide the browser with instructions for how a page should look
JS files provide the browser with instructions on how the page should behave

So, the only html files inside the top level of the main folder are index.html (the 'home' page) and indexTopTemplate.html (a template usable for creating a new index.html page.) All the other items in the main folder are other folders that contain either resources, or that contain other html pages. The resource folders are named so as to give you an idea as to what is in them. So the "images" folder contains pictures, and the "js" folder contains javascripts, and the "Communication" folder contains some sample communication html pages.

Now we can differentiate which resource addressing method we should use. If the resource you need to identify is inside the main AWP folder it is better to use the *relative addressing method*. If the resource you need to identify is outside the main AWP folder you should to use the *absolute addressing method*. The advantage that relative addressing offers is that you can move the main AWP folder and all the pages and links for every page inside will still work. This way you can develop a set of pages on your desktop computer and then copy them to a web hosting service, or another computer and they should work without any changes needed as long as the folders are organized the same way.

In short, an absolute address tells the browser exactly where on the Internet to find the desired resource. A relative address tells the browser where to find resources with respect to the location of the current page – usually nearby.

The absolute address method is especially useful if you are getting images on a site like Opensymbols.org. Let's say you are editing a button's picture using the gridBuilder.html built-in editor. Rather than downloading a picture from Opensymbols.org and saving it in the AWP images subfolder, you could right-click the image and choose "***Copy image address***" from the context menu, and then paste it into the "***Set the Image URL for BtnX***:" field. This method saves steps, time and space. The disadvantage is that absolute addressing like this requires an active Internet connection. It can also slow down page loading while the browser gathers resources.

To illustrate the relative method we can consider a few cases and use the diagram above as a reference.

1. **The resource file is in the same folder**:
   If the resource, let's say a page called "page02.html" is in the same folder as page you are currently editing, and you want modify a link actor to open "page02.html", you would simply type ***page02.html*** into ***Set the link for BtnX:*** field. Be very careful to pay attention to capitalization in addresses – it must be exactly the same to assure it will work everywhere and on all operating systems. Also, avoid spaces in filenames.

2. **The resource file is in a sub folder of the current folder**:
   Let's say you are editing a button on a page in the "YourPages" folder, and the image you wish to assign, an image called "oak_leaf.png", is in a folder called "leaf_pics" that is also inside the "YourPages" folder. In this case the you would type ***leaf_pics/oak_leaf.png*** into the "***Set the Image URL for BtnX***" field. Let's break the URL down:

leaf_pics	/	oak_leaf.png
This tells the browser to look inside the leaf_pics folder	The forward slash separator tells the browser that what follows it is either another folder name or file name.	The name of the resource file we want.

   You could drill deeper into a nested folder structure by adding more folder names followed by slashes.

3. **The resource file is in a neighboring folder**:
   Let's say you are editing a button on a page in the "YourPages" folder, and the image you wish to assign to it is an image called "home.png" that is in the "images" folder. In this case the you would type ***../images/home.png*** into the "***Set the Image URL for BtnX***" field. Let's look at this in detail:

..	/	images	/	home.png
The .. (two periods) tells the browser to go outside the current folder – in this case to the main_AWP_folder which contains both the YourPages and the images folders	The slash separator	Look in the images folder	The slash separator	The name of the resource file we want.

The “..” notation is another name for the parent folder – the parent folder contains child folders. You could also tell the browser to look in the parent of the parent to locate some file as in: ***../../somefile.html***.