css

using css... bla bla bla

Using IDs

When you use custom IDs in your template for assigning styles, be sure that the ID does not conflict with an existing ID. In particular, be sure that it won't conflict with the IDs automatically assigned to section headers. The easiest way to ensure this is to use two adjacent underscores (__) in your ID. Because section IDs are always valid pagenames, they will never contain adjacent underscores.

In plugins use `<pluginname>__<id>`. For example '' 'searchindexplugin__buttonfield' ''

Styling based on page properties

Most often people only need to modify a style of some element on all pages. For example one may want to change the font style for h1 heading. But from time to time, it is desirable to modify a style only on certain page. You could want a special background for any page in the `attic` namespace so that readers instantly realize that they are reading outdated documentation left as a reference. One could also want to add a specific style to some elements if visitor is a connected user, or only on edit page but not when viewing the page. This is possible because most template use an internal DokuWiki function called tpl_classes(). All you need to know is that this function is used by templates to add classes based on page properties to one of the top div element. For example, when a connected visitor is reading the page :wiki:syntax the default template produces this div element :

```
<div id="dokuwiki__top" class="site dokuwiki mode_show tpl_dokuwiki loggedIn
showSidebar hasSidebar lv_1 ns__wiki ns_wiki_ pg_syntax pg_wiki_syntax">
```

This offers quite a lot of possibilities. you can create a style for

- this page only

  conf/userstyle.css.snippet

  ```
  div.dokuwiki.pg_wiki_syntax {
    background: red;
    h1 {
      text-transform: uppercase;
    }
  }
  ```

- any page viewed by a connected user in a namespace just below the root

conf/userstyle.css.snippet

```
div.dokuwiki.loggedin.lv_1.mode_show {
  background: red;
  h1 {
    text-transform: uppercase;
  }
}
```

If your particular template does not use the tpl_classes() function, you will have different classes available. In fhis case, you may want to contact the developer of your theme and ask that the template makes use of this function.

Enough examples, here are the details :

- `dokuwiki`
  This class is always present
- `mode_<ACT>`
  the action that created the page. Most of the time, this will be `mode_view`. But see action_modes for a list of action modes. Note that plugins may create more actions.
- `tpl_<name_of_template>`
  By default, this will be `tpl_dokuwiki`. If you use the sprintdoc, this will be `tpl_sprintdoc`. and if you use the Art for water template, this will be `tpl_artforwater`.
- `loggedIn`
  This class is present only for connected user. There is no «notloggedin» or «loggedout» class. To target page for not loggedIn users, use the `:not()` CSS pseudo class.
- `notFound`
  This class is added for non existent page. Follow this link :this:page:does:not:exist:on:purpose ; the rendered page will have this `notFound` class in its main DW div.
- `home`
  This class is only present on the main DokuWiki page.
- ~~`showSidebar`~~ and ~~`hasSidebar`~~
  These 2 elements shown are **not** added by the tpl_classes() function. They are specific to the default template and not described here (although their meaning is quite obvious)
- `lv_<X>`
  This is the tree depth level of a page. for :dokuwiki page, this level is 0 (zero), so the class will be `lv_0` ; for the page devel:css this will be 1, `lv_1` and for fr:plugin:struct:aggregation, this will be 3, `lv_3`.
- `ns__<namespace-chain>`
  These class are built on the page namespace. Each namespace component concatenated with _ separator. If a namespace has an _ in its name, then this result in a __ in the class name.
  examples :
  - page `:start` ⇒ class `ns__`
  - page `fr:plugin:struct:aggregation` ⇒ 3 classes are added :
    - class `ns__fr`
    - class `ns__fr_plugin`
    - class `ns__fr_plugin_struct`
  - page `matching_food_and_drinks:big_juicy_steak:pan_galactic_gargle_blaste`

r ⇒ 2 classes added :
    - class `ns__matching__food__and__drinks`
    - class `ns__matching__food__and__drinks_big__juicy__steak`
- `ns_<last_ns_component>_`
  This class is based on the last component of the page namespace. Please note that for all pages in the root namespace, there is nothing since the empty root namespace would produce `ns__` which is already added by the previous rule. Examples :
    - page `:start` ⇒ nothing
    - page `fr:plugin:struct:aggregation` ⇒ class `ns_aggregation_`
    - page `matching_food_and_drinks:big_juicy_steak:pan_galactic_gargle_blaster` ⇒ class `ns_big__juicy__steak_`
- `pg_<full-page-id>`
  Same thing as the namespace one, but for the page name. _ in page name are also doubled in the class name. examples :
    - page `:start` ⇒ `pg_start`
    - page `fr:plugin:struct:aggregation` ⇒ class `pg_fr_plugin_struct_aggregation`
    - page `matching_food_and_drinks:big_juicy_steak:pan_galactic_gargle_blaster` ⇒ class `pg_matching__food__and__drinks_big__juicy__steak_pan__galactic__gargle__blaster`
- `pg_<rel-page-id>`
  Same thing as the full-page-id one but don't use the namespace components. For pages in the root namespace, this would be the same as <full-page-id>, so it is not present
    - page `:start` ⇒ nothing
    - page `fr:plugin:struct:aggregation` ⇒ class `pg_aggregation`
    - page `matching_food_and_drinks:big_juicy_steak:pan_galactic_gargle_blaster` ⇒ class `pg_pan__galactic__gargle__blaster`

example selectors

To target all start page in any namespace use this

```
div.dokuwiki.pg_start
```

To target the main page

```
div.dokuwiki.home
```

To target a page exactly in the fr namespace

```
div.dokuwiki.ns__fr.lv_1
```

To target a page anywhere in the fr namespace (meaning fr:syntax or fr:plugin:struct:aggregation)

```
div.dokuwiki.ns__fr
```

To target “this page does not yet exist” in any namespace when using the boozurk template and only for anonymous user and when user is viewing the page but not editing it

```
div:not(.loggedIn).dokuwiki.notFound.mode_show.tpl_boozurk
```

A note for non english page and namespace

CSS accepts fnencode settings

From:
https://schplurtz.lflinkup.net/ - **c**

Permanent link:
https://schplurtz.lflinkup.net/wiki:dokuwiki-css-doc

Last update: **25/08/2021 07:21**