

# Demo

Créée le mardi 01 octobre 2019

## Headers

Appliqué sur : zim, htmlexport, print

Les headers sont décalés pour une meilleure visibilité du niveau de section où on se trouve

### Header H1

#### Header H2

##### *Header H3*

##### Header H4

##### Header H5

## Souligné

Appliqué sur : zim, htmlexport, print

Les textes soulignés sont soulignés et non surlignés.

## Verbatim

Appliqué sur : zim, htmlexport, print

Pour bien le mettre en évidence, le verbatim `inline` est coloré en darkmagenta.

Tout comme  
le bloc verbatim

A noter que le bloc verbatim est indenté, zim fait la différence par le fait de sélectionner plusieurs lignes avant d'appliquer le style (<Ctrl>T).

# Tableaux

Appliqué sur : htmlexport, print

Les tableaux dans zim ne sont vraiment pas terribles, mais on peut avoir quelque chose de sympa une fois exportés.

Prénom	Nom	email	Couleur préférée
Joe	Smith	<a href="mailto:joe.smith@here.com">joe.smith@here.com</a>	Orange
Marcel	Dupont	<a href="mailto:marcel.dupont@nothere.com">marcel.dupont@nothere.com</a>	Bleu
Nathalie	Charpentier	<a href="mailto:nathalie.charpentier@lasbas.com">nathalie.charpentier@lasbas.com</a>	Vert
Electra	Cigales	<a href="mailto:electra.cigales@nowhere.com">electra.cigales@nowhere.com</a>	Rouge

# Images

Appliqué sur : htmlexport



Lors de l'export html, les images sont centrées, c'est plus beau. Elles ne le sont pas lors du print parcequ'en effet secondaire, les images peuvent être coupées à l'impression.

# Coloration syntaxique

Appliqué sur : htmlexport, print

Il s'agit des bloc de code insérés dans zim avec le plugin adéquat.

Un exemple en C++

```
#include <iostream>

int main(int argc, char *argv[]) {

    /* An annoying "Hello World" example */
    for (auto i = 0; i < 0xFFFF; i++)
```

```

    cout << "Hello, World!" << endl;

    char c = '\n';
    unordered_map<string, vector<string> > m;
    m["key"] = "\\\\"; // this is an error

    return -2e3 + 12l;
}

```

## Un exemple en Python

```

@requires_authorization
def somefunc(param1='', param2=0):
    r'''A docstring'''
    if param1 > param2: # interesting
        print 'Gre\`ater'
    return (param2 - param1 + 1 + 0b10l) or None

class SomeClass:
    pass

>>> message = '''interpreter
... prompt'''

```

## Un exemple en Ruby

```

# The Greeter class
class Greeter
  def initialize(name)
    @name = name.capitalize
  end

  def salute
    puts "Hello #{@name}!"
  end
end

g = Greeter.new("world")
g.salute

```

la coloration syntaxique est appliquée avec un thème sombre lors de l'export et avec thème clair lors du print.

## Police

Appliqué sur : `htmllexport`, `print`

Comme il est recommandé d'avoir une police sans empattement sur un écran et inversement avec empattement sur papier, lors de l'export html, la police est "sans serif" et lors du print, la police est "serif".

# Navigation

Appliqué sur : htmlexport



Les textes de navigation sont remplacés par des icones représentant des flêches, le texte apparait néanmoins lors d'un survol du pointeur de la souris.