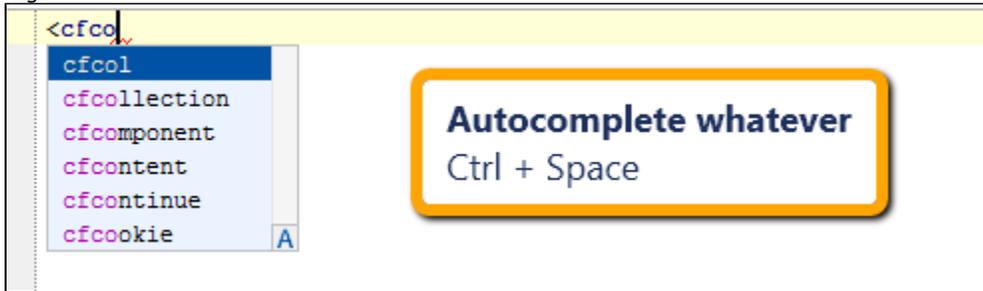# ColdFusion Plugin Features List

- Autocompletion (Ctrl+Space in IntelliJ IDEA 9 and auto popup in IntelliJ IDEA 10) works for
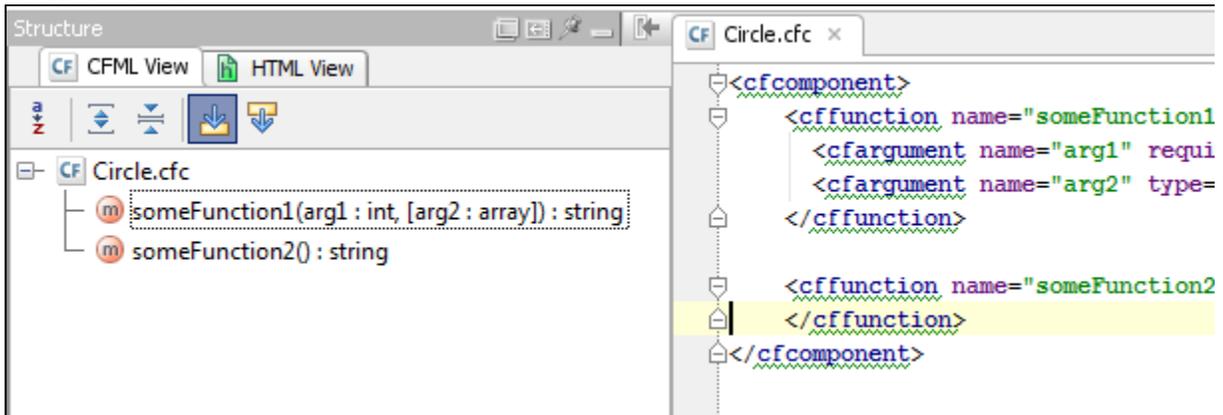   - tags

    - attributes
    - attribute values (in case of predefined constants)
    - functions and variable of the current scope
    - function arguments
    - functions of java classes and components created through 'createObject' function
    - for components' and java classes' names in 'createObject'
    - for inherited methods of a component
    - in case you have other suggestions, please scratch a feature request

- Navigation
    - open file or component with specified name (Ctrl+Shift+N)
    - go to definition of a function, component, argument or variable (Ctrl+B or Ctrl+Click on the usage)

    - go to function name inside a file (Ctrl+F12 shows a popup 'Structure View' dialog of functions defined in the file and allow to navigate to them)
    - go to the nearest opening (or closing) brace, including opening and open tags (Ctrl+[(or +]))
    - structure view tab

- Highlighting
    - usages highlighting (Ctrl+Shift+F7 for function, variable or argument inside scope)

- paired braces highlighting (when standing on left or rigth (works for [], (), {} and tags))
- parser error highlighting
- unresolved usages (when declaration can not be found) (customizable at)
- usages search (per project or scope) (Alt+F7)

- Other code helpers
  - insert closing tag in typing '>'
  - insert pared brace or quote
  - comment code block with Ctrl+Shift+'/'
  - comment line with Ctrl + '/'
  - insert predefined code construct (customize at Settings/IDE Settings/Live Templates/ColdFusion), for instance typing 'cfo' + Tab inserts cfoutput paired tags and place the cursor inside them
  - parameter info is shown for functions (Ctrl+P inside '()' of function call)
  - folding (code blocks in scripts and paired tags can be folded) - use tickets on the left or Ctr+'' and Ctrl'-' hotkeys

  - explicitly define class type of java instances

  - JavaLoader support

- MXUnit test framework support
  - to install MXUnit just unpack mxunit sources under coldfusion root
  - write a test (a component with public test methods, that extends 'mxunit.framework.TestCase')
  - create MXUnit run configuration
  - specify web path to your test-component and absolute path to test-component's file (you may also run directory of test-components and a particular method)
  - run

- benefits:
  - navigation from tests tree to source code (Double Click or F4)
  - navigation from stack trace to source code (Double Click on File Reference)
  - tests statistics
  - rerun failed option
- here is how it looks like