# Sncode : %include ...

- This compiler directive is used to include a file at compile time.
- Example:

```
%include "/extenso/module/sed/core/package/sed_core_util.sn";
```

- After that you can use all the fonctions and packages in this file.
- This is a compiler directive.
- It's not dynamic. What does it means ?

# Sncode : %include

Consider this:

```
{{
        %include "/extenso/module/sed/core/package/sed_core_util.sn";
        res = sql("select uid,username from sed_login_user limit 5");
        sed_core_util::display_variable_in_html(res);
}}
```

- {{ %include "..." }} vs
- \{{ %include "..." }} in a widget

# Sncode : %include

- In the first version:
  - Widget is compile,  the include is run and the function is called
  - The resulting file contain the result of the function call
  - In the final file, we don't have any include or any function call
- In the second version
  - Widget is compile
  - Resulting file contain the include
  - the function is call dynamic. If sed_login_user change we see the change
  - If the include change we have to republish the widget

# Sncode : include()

- This function is use to include
  - text file
  - sncode file
  - binary sncode file
- Function in the include are not install unless you use:

```
include(load_functions:true, "/site/cours/widget/include2.sn");
```

# Sncode : GD

- gd is a graphics library.  It allows your code to quickly draw images complete with lines, arcs, text, multiple colors, cut and paste from other images, and flood fills, and write out the result as a PNG or JPEG file.  This is particularly useful in World Wide Web applications, where PNG and JPEG are two of the formats accepted for inline images by most browsers.

# Sncode : gd

- Example:

```
im=gd_new(width:1000,height:1000,bgcolor:White,truecolor:true);
....
gd_free(im);

gd_write is used to generate image:

img = gd_write(gd:im,base64: true, format:"jpg",quality:100);

<img src="{{ img }}">
```

# Sncode : class

- A Class is like an object constructor, or a "blueprint" for creating objects.
- Example of a class definition:

```
class Complex
        method Complex(real, imag)
              this.real = real;
              this.imag = imag;
          endm
```

# Method add

```
        method add(c)
                if (classtype(c) eq "Complex") then
                    x = new Complex(this.real + c.real, this.imag +
c.imag);
                else
                    x = new Complex(this.real + c, this.imag );
                endif
                return x;
            endm
```

# Method sub

```
method sub(c)
    if (classtype(c) eq "Complex") then
        x = new Complex(this.real - c.real, this.imag - c.imag);
    else
        x = new Complex(this.real - c, this.imag );
    endif
endm
```

# Method mul

```
method mul(c)
    if (classtype(c) eq "Complex") then
        x = new Complex(this.real * c.real - this.imag * c.imag,
                        this.imag*c.real + this.real*c.imag);
    else
        x = new Complex(this.real * c, this.imag * c);
    endif
    return x;
endm
```

# Method print

```
    method print()
        this.real; "+"; this.imag ; "i";
    endm


endc
```

# Class usage

```
c = new Complex(1,2);
        y = c.add(c); y.print(); " ";
        y = c.add(5); y.print(); " ";
        y = c.mul(c); y.print(); " ";
        y = c.mul(5); y.print(); " ";
        c.print();
return
2+4i 6+2i -3+4i 5+10i 1+2i
```

# Sncode : Ternary operator

- In computer programming, **?:** is a ternary operator that is part of the syntax for basic conditional expressions in several programming languages. It is commonly referred to as the **conditional operator**, **inline if (iif)**, or **ternary if**. An expression a ? b : c evaluates to b if the value of a is true, and otherwise to c. One can read it aloud as "if a then b otherwise c".

# Sncode : Ternary operator

- (expression1)  ? expression2 : expression3;
- Same as:
  - if expression1 then
    - expression2;
  - else
    - expression3;
  - endif

  - a = (x == 5) ? x +2 : x  - 2 ;

# Sncode : Ternary operator

- Try:

```
a = "Courtesy";


(b) ? b : a; "\n";


b = undefined;
"b undefined: ";
(b) ? b : a; "\n";

b = null;
"b null: ";
(b) ? b : a; "\n";
```

```
b = true;
"b true: ";
(b) ? b : a; "\n";

b = false;
"b false: ";
(b) ? b : a; "\n";


b = 1;
"b 1: ";
(b) ? b : a; "\n";


b = 0;
"b 0: ";
(b) ? b : a; "\n";
```

# Sncode : Ternary operator

- Courtesy
- b undefined: Courtesy
- b null: Courtesy
- b true: true
- b false: Courtesy
- b 1: 1
- b 0: Courtesy

# Sncode : break

## break is use to exist the a loop:

```
for i in [1,2,3] do
        if i == 2 then
            break;
        endif
        i;
endfor

This loop will display 1;
```

# Sncode : break : exercise

```
for i in [1,2,3] do
        for j in [4,5,6] do
            if i == 2 && j == 5 then
                break;
            endif
            i; j; " ";
        endfor
endfor
This will return : 14 15 16 24 34 35 36
```

# Sncode : continue

- continue skip to the next iteration of the inner loop

```
for i in [1,2,3] do
        if i == 2 then
            continue;
        endif
        i;
endfor

This loop will display 13;
```

# Sncode : continue (exercice)

```
for i in [1,2,3] do
        for j in [4,5,6] do
            if i == 2 && j == 5 then
                continue;
            endif
            i; j; " ";
        endfor
endfor
return 14 15 16 24 26 34 35 36
```

# Sncode : use

- In a generate, import a variable from the upper call
- x=5;
- generate(template:"/template.sn", file:"/staging/file.sn");
  - In the generate, use x; will get the variable x.
  - If you modified x, it will not modified the top variable

# Sncode : cast

- Use to transform a variable:
- a = (int) 2.5; "\n"; // return 2
- a = (float) 2; "\n"; // return 2 as a float
- a = (string) 2.5; "\n"; // return 2.5 as a string
- a = (bool) 1; "\n"; // return true
- a = (bool) 0; "\n"; // return false
- a = 1/2; "\n"; // return 0
- a = 1/(float) 2; // return 0.5

# Documentation....

- In git click README.md (https://daringfireball.net/projects/markdown/syntax)
- package p_extract.sn /extenso/module/sed/module_manager/package/p_extract.sn
- //!code. /*!code .... */
- function
- package
- version //!version 2022-02-14 laplante@sednove.com an
- //!anything here is the comment
- //!business-rule describe business rule in the code

# PRELOAD / Extranet / Rewrite

- In the conf file:
  - Preload : "/usr/local/website/psv1/extenso/html/secure/module/sed/login/en/preload.snc"
  - Rewrite : "/usr/local/website/psv1/extenso/html/secure/module/sed/rewrite/en/rewrite.snc"
  - Error : "/usr/local/website/psv1/extenso/html/module/sed/error/en/error.snc"