



SEDNOVE

Sncode/Extenso

pierre.Laplante@sednove.com

Version 1.2 : 2021-12-31

Course #1

- What you will see in this course:
 - Introduction about Extenso
 - Sednove
 - Extenso – Sncode
 - Architecture
 - Sncode programming

About this course

- Introduction to programming in Sncode
- Goal: Enable non-programmers to learn how to program, in particular with Sncode and Extenso
- What you will learn:
 - Sncode
 - Extenso
 - HTML
 - CSS
 - Javascript
 - JQuery
 - Ajax
 - Websocket
 - WebRTC

What you will need for this course

- HTML

- <https://www.youtube.com/watch?v=BvJYXI2ywUE>
- <https://www.youtube.com/watch?v=PypMN-yui4Y>
- https://www.youtube.com/watch?v=1rbo_HHt5nw
- <https://www.youtube.com/watch?v=bFvjE4ZRtSE>

- CSS / Bootstrap

- <https://www.youtube.com/watch?v=-qfEOE4vtxE>
- https://www.youtube.com/watch?v=1PnVor36_40

- Javascript

- <https://www.youtube.com/watch?v=cmlkfezTnBE&list=PL9dbBb7MI9bXwgPTH5STNGEQNQNeCDXdu>

What you will need for this course

- jQuery
 - <https://www.youtube.com/watch?v=hMxGhHNOKCU>
- JSON
 - <https://www.youtube.com/watch?v=iiADhChRriM>
- SQL MariaDB/Mysql
 - https://www.youtube.com/watch?v=p3qvj9hO_Bo
- REGEX : Regular Expression
 - <https://www.youtube.com/watch?v=rhzKDrUjJVk>
- Git
 - <https://www.youtube.com/watch?v=IHaTbJPdB-s>
- And programming experience in a language...

What you **may** need for this course

- Linux (CentOS). Basic Command line interface (CLI)
 - <https://www.youtube.com/watch?v=5jIIOkA0Npl>
- Apache configuration
 - <https://www.youtube.com/watch?v=rCr3-YIL5S8>
- C programming
 - cmake / make / gcc
- Websockets

About this course

- keep your personal question for later with me directly
- this is an introduction course not an advanced course.
- the course is recorded
- if you have a question, please raise your hand first
- you will need a headphones to speak

Sednove

- Founded in 1997 by Chantal Bilodeau and Pierre Laplante
- Web and mobile applications development
- Technological development
- Branding
- Design

Platinum

- Founded in 1995.
- Software developed in FoxPro
- Front Desk component only until 2000 (schedule, patient file, transactions and reports)
- EHR module in 2000 (doctor notes, patient flow with check in and calling patients to the room)
- in USA and around the world since 2002
- Need to move to the cloud to integrate new tools

Extenso / Sncode

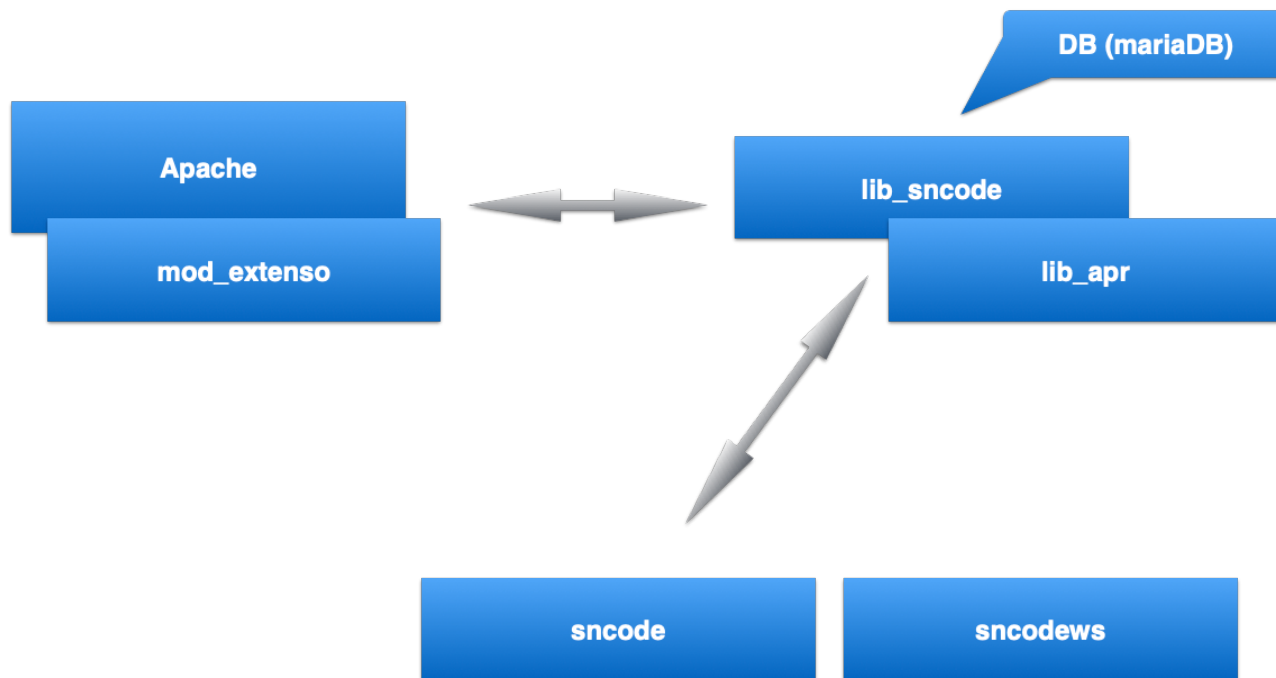
- Extenso : Clients and programmers interface
- Sncode : Programming language



Sednove's tools

- slack
- clickup
- uxpın
- gecko
- email
 - laplante@sednove.com
- Phone
 - 514-945-1779 (also whatsapp)

Architecture of Extenso



What is different about Extenso ?

- Dynamic / static
- Compile language
- Security with virtual machine
- Grid manager
- Style manager
- Modules manager
- Database manager
- Git/Gitlab management of modules

Sncode

- Key concepts:
 - Sncode is a compiled language
 - Uses a virtual machine to execute code
 - Rich and extensible library
 - External modules
 - Simple syntax for non programmer
 - Power-full for professional programmer

compile, virtual machine etc.

- Interpreted language, PHP, Ruby
- Compile language : C
- Virtual machine : Java, C#, Sncode
- JIT : Just in Time
- Assembler
- Machine code

Sncode concepts

- File with extension .sn are compile and execute
- File with extension .snc are file already compiled in binary
- Convention use in the naming of site:

<https://ssnode.sednove.com> for the staging version

<https://sncode.sednove.com> for the production version

Sncode concepts #2

- staging require a login to modify the site
- Production deployment (or publishing) is the process of copying the file from staging to production
- Directory for staging is /staging
- Directory for html is /html

Sncode

- In a page everything that is not between {{ and }} is print as-is
- {{ Start Sncode
- }} End Sncode
- All text outside of Sncode is returned to the browser without being parsed

IDE

- Introduction to IDE in Extenso
- How to execute a program:
 - In staging : <https://ssncode.sednove.com/ex.sn>
 - In html : <https://sncode.sednove.com/ex.sn>

Simple example

- ```
<html>
 <body>
 <h1>{{ "Date is "; datetime(); }}</h1>
 </body>
</html>
```

# Exercices

- Exercice #1 : try to reproduce the previous example.
  - Create a file in /staging/ex1.sn
  - Execute it with the URL <https://ssncode.sednove.com/ex1>
- Exercice #2 Only display the time not the date

PS : <https://extenso.live>

- PS #2 <https://getbootstrap.com/docs/4.5/getting-started/introduction/>

# Sncode's documentation

- All documentation under <https://extenso.live>
- <https://module.sednove.com>
- Man pages under Linux:

`man sql`

## Sncode's types

- Integer : `{{ a = 5 ; }}`
- Double : `{{ a = 3.1415 ; }}`
- Array : `{{ a = [ 1, 2.0, "3.1415" ] ; }}`
- Boolean : `{{ a = true ; }}`
- Null : `{{ a = null ; }}`
- Undefined : `{{ a = undefined ; }}`
- Associative Array / Hash array / Context :  
`{{ a = { "x" : 1.5, "y" : 2.0 } ; }}`
- `a.type()` will return the type of variable a

# Integer

- Try This program:

```
{ {
 a = 5;
 b = a / 2;
 "b = "; b;
 type(b);
} }
```



# Operators

- By order of priority:

- +, -,
- \*, /,
- \*\* (power), % (modulo)

- Try:

$$2 + 3 * 4 ** 2 \Rightarrow 50$$

- Use () to modify the order:

$$((2 + 3) * 4) ** 2 \Rightarrow 400$$



SEDNOVE

Sncode/Extenso

[pierre.Laplante@sednove.com](mailto:pierre.Laplante@sednove.com)

Version 1.4 : 2020-07-03

# Course #2

- What we have seen so far:
  - Extenso presentation
  - How to use IDE
  - Sncode's type
  - Structure of directory in Extenso

# Integer / float

- Try this program:

```
{ {
 a = 5;
 b = a / 2.0;
 "b = "; b;
 type(b);
} }
```

# Function printf

- `printf` : print formatted
- `printf("Number = %05d", 10);`
- `d` : use to print integer
- Try
  - `printf("%7d", 10); a=printf("%x", 10); a;`
  - `printf("%-10s", a); printf("%10.4f", 10.2);`
  - `printf("%010.4f", 10.2); printf("%+010.4f", 10.2);`
  - `printf("%+10.4f", 10.2); printf("%10.1f", 5.17);`
- `f` : float, `s` : space, `x` : hexadecimal

# Floating point number

- Example:

```
a = 1.123456789;
a;
```

- By default snprintf use: `printf("%.8f", number)`
- According to Wikipedia:

"In [computing](#), **floating-point arithmetic (FP)** is arithmetic using formulaic representation of [real numbers](#) as an approximation to support a [trade-off](#) between range and [precision](#)."

# Floating point number

- Floating point number are represented as double in C
- Try:

```
{ { printf("%.20f", 0.1+0.2); } }
```

- Check:

<https://docs.python.org/3/tutorial/floatingpoint.html>

[https://doc.lagout.org/science/0\\_Computer%20Science/3\\_Theory/Handbook%20of%20Floating%20Point%20Arithmetic.pdf](https://doc.lagout.org/science/0_Computer%20Science/3_Theory/Handbook%20of%20Floating%20Point%20Arithmetic.pdf)

# Floating point number

- Try this program:

```
a = 1/5;
b = 1/5.0;
c = 1.0/5;
"a="; a; ", b="; b; ", c ="; c;
```



# Floating point number

- Try:

```
a = 48.0 * atan(1.0/49.0) +128.0 *
 atan(1.0/57.0) -
 20.0 * atan(1.0/239.0) + 48.0 *
 atan(1.0/110443.0);

printf("%.25f", a);
```

# Floting point number comparaison

- Floating point comparaison : operator ==
- try:

```
a=0.15+0.15; // 0.3000000000000155
b=0.1+0.2; // 0.3000000000000144
a==b;
```

- return !

false

# Floating point number

```
function compare(a,b)
 //!code Minimal function to compare FPN to 0.0001
 if a==b then
 return true;
 endif

 if abs(abs(a) - abs(b)) < 0.001 then
 return true;
 endif
 return false;
endf

a = 0.15+0.15; b = 0.1 + 0.2; a==b; compare(a,b);
```

# Comparison operators

- $<$  : less than
- $>$  : greater than
- $\leq$  : less or equal
- $\geq$  : greater or uqual
- $\Leftrightarrow$  : compare  $1 \Leftrightarrow 2$ ;  $2 \Leftrightarrow 1$ ;
- $\neq$  : not equal