SEDNOVE

Sncode/Extenso

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 sncode 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

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;
b=0.1+0.2;
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
```

Comparaison operators

- < : less than
- > : greater than
- <= : less or equal
- >= : greater or uqual
- <=> : compare
- != : not equal

FPN example

```
if 1 == 2 then
     "Oh la la something is wrong here";
else
     "Ok 1 is not equal to 2";
endif
```

FPN compare with string

- String are automatically convert to double

```
a = "0.0001";
b = 0.0001;
if a == b then
    "a is equal to b";
else
    "a is not equal to b";
endif
```

- Test:

```
if "0" == 0 then "true"; else "false"; endif
```

Comparaison operator

- If we do:

```
a = 5 < 6; a;
```

- a is a boolean : true or false

```
if a then "true part"; else "false part"; endif
```

- If we do:

```
a = 1;
if a then "true part"; else "false part"; endif
```

- 1 is true and 0 is false

The art of programming... part #1

- Good indentation
- Use good meaningful variable name
- Use comment:
  - `/* ... */`
  - `/*`
  - `...`
  - `...`
  - `*/`
  - `// comment`
  - `# comment`

Comparaison operator : <=>

- Use to compare 2 numbers
- Return
  - -1 if left then
  - 0 if equal
  - 1 if greater than
- Example using a new statement : switch

```
a = 60;
b = 5;
switch a <=> b do
```

Comparaison operator : <=>

- Use to compare 2 numbers
- Return
  - -1 if left then
  - 0 if equal
  - 1 if greater than
- Example using a new statement : switch

```
a = 60;
b = 5;
switch a <=> b do
```

Comparaison operator : <=>

- Use to compare 2 numbers
- Return
  - -1 if left then
  - 0 if equal
  - 1 if greater than
- Example using a new statement : switch

```
a = 60;
b = 5;
switch a <=> b do
```

Comparaison operator : <=>

```
case -1:
    "a is lower than b";
    endc

case 0:
    "a is equal to b";
    endc

case 1:
    "a is greater than b";
    endc

default:
    "Ohh la system error";
    endc
ends
```