

A little bit more advanced Scilab

ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเชียงใหม่

More on control statement

- **If** : เลือกทำงานสองอย่าง ขึ้นอยู่กับ เงื่อนไข
 if เงื่อนไข **then**
 งานที่ 1
 else
 งานที่ 2
 end
- **For** : ทำงานแบบวนซ้ำเท่ากับจำนวนรอบที่ต้องการ
- **Select** : คล้าย **if** แต่มีงานให้เลือกทำมากกว่า 2
- **While**: คล้าย **for** แต่การวนซ้ำจะขึ้นอยู่กับเงื่อนไข

The *Select* Statement

- คือการเลือกทำงานตามเคสที่เข้ามา มี **syntax** ดังนี้

select ตัวแปร

case ตัวเลข หรือ ตัวอักขระ

process 1

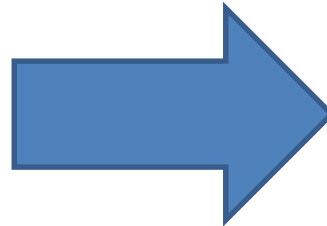
case ตัวเลข หรือ ตัวอักขระ

process 2

else

process 3

end



```
i = 2
select i
case 1
    disp("One")
case 2
    disp("Two")
case 3
    disp("Three")
else
    disp("Other")
end
```

The *While* statement

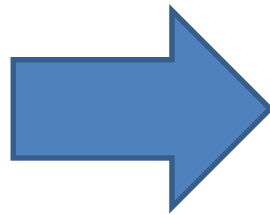
- เป็นการทำงานแบบวนซ้ำ จนกว่าเงื่อนไขที่ให้ทำงานจะไม่เป็นจริง
- แปลเป็นภาษาไทย *ในขณะที่ ให้ทำ*

while เงื่อนไขเป็นจริง

process 1

process 2

end

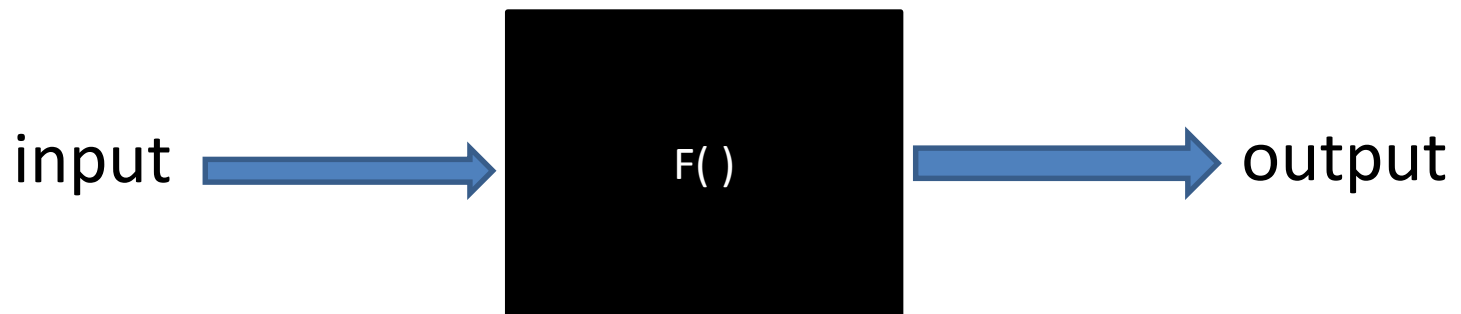


```
s = 0
i = 1
while ( i <= 10 )
    s = s + i
    i = i + 1
end
```

ถ้าเงื่อนไขเป็น 1 while จะกลายเป็น infinite loop

Functions in Scilab

- ฟังก์ชันคือ ส่วนของโปรแกรมที่ทำหน้าที่เฉพาะ มีการรับข้อมูลเข้า (**input**) เข้าไปประมวลผล แล้วส่งกลับข้อมูลออก (**output**) กลับมาให้ผู้ใช้
- ฟังก์ชันมีประโยชน์ในการรวมการประมวลผลที่ใช้บ่อยๆ เข้าไว้ด้วยกัน เพื่อง่ายต่อการเรียกใช้



Calling a function

- การเรียกใช้ฟังก์ชันที่ถูกกำหนดไว้แล้วทำได้โดย

`outvar = function_name(invar)`

`outvar` คือ ผลลัพธ์จากการประมวล

`invar` คือ ข้อมูลที่จะส่งเข้าไปประมวลผล

เช่น `x = dice()`

`m = mean([1 2 3])` , `m` คือ `outvar`

`[1 2 3]` คือ `invar`

Defining a function

- การประกาศฟังก์ชันทำได้โดย

```
function output = function_name(input)
```

```
endfunction
```

- เช่น

```
function [r] = dice()
```

```
    r = 1 + floor(rand(1)*6);
```

```
endfunction
```

Why using function?

Using function

```
function [r] = dice()  
    r = 1 + floor(rand(1)*6);  
endfunction
```

```
d1 = dice();  
d2 = dice();  
d3 = dice();
```

```
if d1+d2+d3 > 9 then...
```

Sequential programming

```
d1 = 1 + floor(rand(1)*6);  
d2 = 1 + floor(rand(1)*6);  
d3 = 1 + floor(rand(1)*6);
```

```
if d1+d2+d3 > 9 then...
```


Input & Output

- การอ่านหรือบันทึก ตัวแปร ด้วยมาตรฐาน **scilab** ไว้ในไฟล์ ใช้คำสั่ง
 - load, save
- การอ่านหรือบันทึก ข้อมูลที่เป็นมาตรฐาน **CSV** ไว้ในไฟล์ ใช้คำสั่ง
 - variable = csvRead(filename)
 - csvWrite(variable, filename)

CSV = Comma Separated Values คือข้อตกลงหนึ่งในการเก็บ
ข้อมูลแบบตาราง / เมทริกซ์

CSV File

Scilab code

Writing

```
M = eye(4,4);  
csvWrite(M,'test.dat');
```

Reading

```
N = csvRead('test.dat');
```

Content of test.dat

1,0,0,0

0,1,0,0

0,0,1,0

0,0,0,1

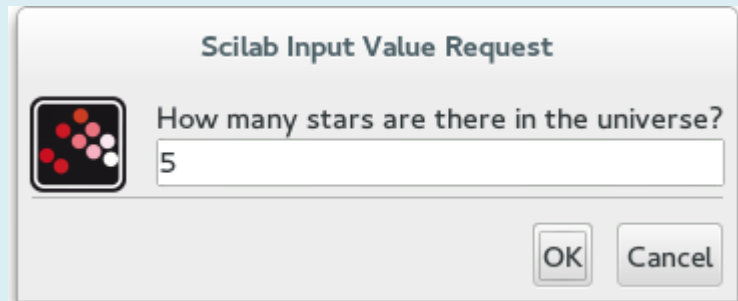
User Interface: Input Dialog

- Syntax

`m = x_dialog('message to user', 'default value');`

m ที่ได้จะเป็นข้อความ (**string**) ต้องใช้ฟังก์ชัน **evstr()** เพื่อแปลงค่าเป็นตัวเลข หากต้องการใช้ค่าตัวเลขในการคำนวณต่อ

```
x = x_dialog('How many stars are there in the universe?','5')
```

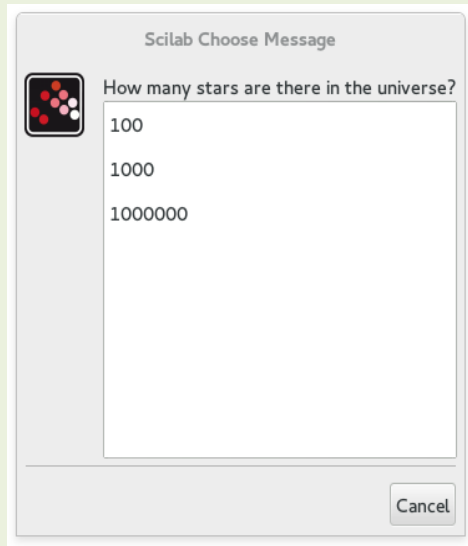


User Interface: Selection box

- บังคับให้เลือกจากตัวเลือกที่ถูกกำหนดไว้แล้ว
- Syntax

```
n = x_choose(['item1', 'item2', 'item 3'], ['message']);
```

```
x = x_choose(['100', '1000', '1000000'] 'How many stars are there in the universe?')
```



Double click to
select one option

User Interface: Message box

- เอาไว้แสดงข้อความแก่ผู้ใช้ โดยไม่ต้องการคำตอบ
- Syntax
`messagebox('message');`

`Messagebox('hello there');`

