

Les commandes Scilab sont écrites en **violet**. Dans les exemples, toute ligne précédée de **-->** est une commande, les autres lignes sont les retours.

## Boucles

Dans le cas où la même instruction doit être répétée plusieurs fois, on utilise une structure itérative appelée boucle. Elles sont de deux types :

- **On connaît le nombre de répétitions N :**

Pour n allant de 1 à N	<b>--&gt;for n=1:N</b>
Exécuter l'instruction	<b>--&gt;Exécuter l'instruction</b>
Fin de pour	<b>--&gt;end</b>

- **On ne connaît pas le nombre de répétitions mais on a un critère d'arrêt.**

Tant que le critère n'est pas réalisé	<b>--&gt;While</b>
Exécuter l'instruction	<b>--&gt;instruction</b>
Fin de tant que	<b>--&gt;end</b>

## Tests

Lorsqu'il faut envisager plusieurs possibilités, on utilise une instruction conditionnelle appelée test.

Si le test est réalisé, alors	<b>--&gt;if test réalisé then</b>
Instruction 1	<b>--&gt;Instruction 1</b>
Sinon	<b>--&gt;else</b>
Instruction 2	<b>--&gt;Instruction 2</b>
Fin de si	<b>--&gt;end</b>

S'il y a plus de deux possibilités on utilisera **if...then...elseif...then...elseif...then...end**

## Exemples et exercices

**Exemple 1 : calcul de  $S = 1+2+3+.....+500$ .**

Algorithme	Scilab
Mettre 0 dans S Pour n allant de 1 à 500 Mettre S+n dans S Fin de pour Afficher S	<b>--&gt;S=0;</b> <b>--&gt;for n=1:500</b> <b>--&gt; S=S+n;</b> <b>--&gt;end</b> <b>--&gt;S</b> <b>S =</b> <b>125250 .</b>

**Exemple 2 : calcul du quotient q d'un nombre positif N par 11 par la méthode des soustractions successives.**

2

Algorithme	Scilab
Lire N Mettre 0 dans q Tant que N supérieur ou égal à 11 Mettre N-11 dans N Mettre q+1 dans q Fin de tant que Afficher q	<pre> --&gt;N=input("N = "); --&gt;q=0; --&gt;while N&gt;=11 --&gt;  N=N-11; --&gt;  q=q+1; --&gt;end --&gt;afficher("quotient = "+string(q)) </pre>

**Exemple 3 : calcul de la distance entre deux nombres**

Algorithme	Scilab
Soit y l'image du couple (a , b) par la fonction d qui calcule la distance. Si a supérieur ou égal à b, alors Mettre a-b dans y Sinon Mettre b-a dans y Fin de si Fin de fonction  Distance entre 3 et -5	<pre> --&gt;function y=d(a,b) --&gt;  if a&gt;=b then --&gt;    y= a-b; --&gt;  else --&gt;    y=b-a; --&gt;  end --&gt;endfunction  --&gt;d(3,-5) ans =     8. </pre>

**Exemple 4 : Bob place 5000 € à intérêts composés à 2,7 % par an en 2009.**

- 1) **Ecrire un programme pour calculer les sommes obtenues pendant les 20 prochaines années. Faire dessiner le nuage des sommes**

Algorithme

Mettre 5000 dans S(1) qui sera la somme de l'année 2008 + 1

Par la suite S(n) sera la somme de l'année 2008 + n

Pour n allant de 1 à 20

Mettre S(n) multipliée par 1,027 dans S(n+1)

Afficher l'année et la somme

Fin de pour

Dessin du nuage

Effacer l'écran graphique

Dessiner le nuage des points (n ; S(n)) avec des croix rouges

### Scilab

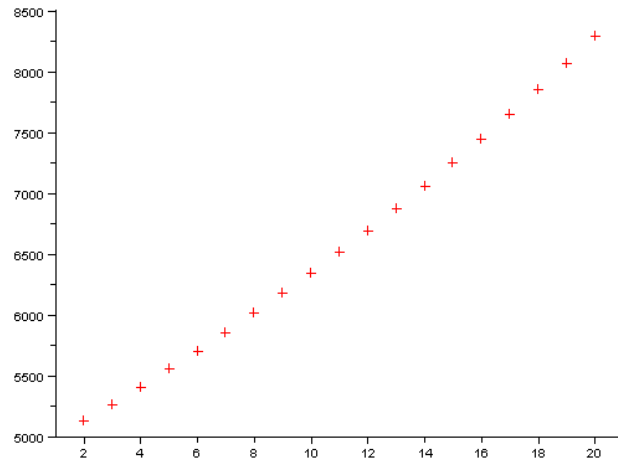
```
-->S(1)=5000;  
-->for n=1:20  
-->  S(n+1)=S(n)*1.027;  
-->  afficher ([2008+n, S(n)])  
-->end
```

```
2009.    5000.  
2010.    5135.  
2011.    5273.645
```

etc...

*Dessin du nuage avec des croix rouges*

```
-->clf  
-->plot(S, "+r")
```



## 2) Ecrire un programme lui permettant de savoir en quelle année il aura 7000 €

### Algorithme

Mettre la somme 5000 dans S

Mettre l'année 2009 dans N

Tant que la somme reste inférieure à 7000, alors

La somme est multipliée par 1,027 (elle remplace la précédente)

L'année augmente de 1 (elle remplace la précédente)

Fin de tant que

Afficher l'année

### Scilab

```
-->S=5000;  
-->N=2009;  
-->while S<7000 then  
-->  S=S*1.027;  
-->  N=N+1;  
-->end  
-->afficher("S dépasse 7000 € en : "+string(N))  
S dépasse 7000 € en : 2022
```

**Exemple 5 : Avec plus de deux choix : Virginie lance trois dés numérotés de 1 à 6. Si elle obtient une somme de 18, elle gagne 50 euros, entre 10 et 17 elle gagne 5 euros, sinon rien.**

### Algorithme

Mettre dans T trois nombres entiers tirés au hasard entre 1 et 6

Mettre dans S la somme de ces trois entiers

Si  $S < 10$  alors afficher : « Virginie ne gagne rien »

Sinon, si  $S < 18$  alors afficher « Virginie gagne 5 euros »

Sinon (*forcément*  $S=18$ ) afficher « Virginie gagne 50 euros »

Fin de si

### Scilab

```
-->T=tirage_entier(3,1,6);  
-->S=sum(T)  
-->if S<10 then  
-->  afficher("Virginie ne gagne rien")  
-->elseif S<18 then  
-->  afficher("Virginie gagne 5 euros")
```

```

-->else
-->    afficher("Virginie  gagne 50 euros")
-->end
S =
    7.
Virginie ne gagne rien

```

### Exercice 1

La suite définie par  $\begin{cases} a_1 = 2 \\ a_{n+1} = \frac{1}{2} \left( a_n + \frac{2}{a_n} \right) \end{cases}$  permet de calculer une valeur approchée de racine de 2.

- 1) Faire calculer 20 termes de la suite.
- 2) A partir de quelle valeur de n la suite donne-t-elle racine de 2 avec autant de précision que la machine ?

### Exercice 2

Etant donné les trois côtés d'un triangle, élaborer un programme qui permet de dire si ce triangle est isocèle, équilatéral, ou rien.

### Exercice 3

Dichotomie : Recherche d'une valeur approchée du nombre d'or, solution positive de l'équation  $x^2 = x + 1$ .

- 1) Définir la fonction  $f : f(x) = x^2 - x - 1$ . Faire tracer la courbe sur l'intervalle  $[-5; 5]$ , on remarque que la solution est entre 1 et 2.
- 2) Ecrire un programme permettant par dichotomie d'encadrer la solution cherchée dans un intervalle d'amplitude  $10^{-4}$ , en partant des valeurs 1 et 2.

### Exercice 4

1. Soit  $(u_n)$  la suite définie, pour tout entier n non nul, par :  $u_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{n!}$   
Calculer les 100 premiers termes de cette suite.
2. Soit la suite définie, pour tout entier n non nul, par :  $v_n = u_n + \frac{1}{n \times n!}$ . Calculer et faire afficher les 100 premiers termes de cette suite avec les 100 premiers termes de la suite  $(u_n)$
3. Représenter graphiquement les 100 premiers termes des suites  $(u_n)$ ,  $(v_n)$  et  $(u_n - v_n)$ .
4. Que peut-on conjecturer sur les suites  $(u_n)$  et  $(v_n)$  ? Démontrer cette conjecture.
5. En déduire que la suite  $(u_n)$  converge.
6. Faire afficher dans le programme précédent une valeur approchée de la limite de  $(u_n)$  à  $10^{-p}$  près. On pourra faire varier p de 1 à 10, ou définir une fonction qui retourne cette valeur en fonction de la valeur de p.