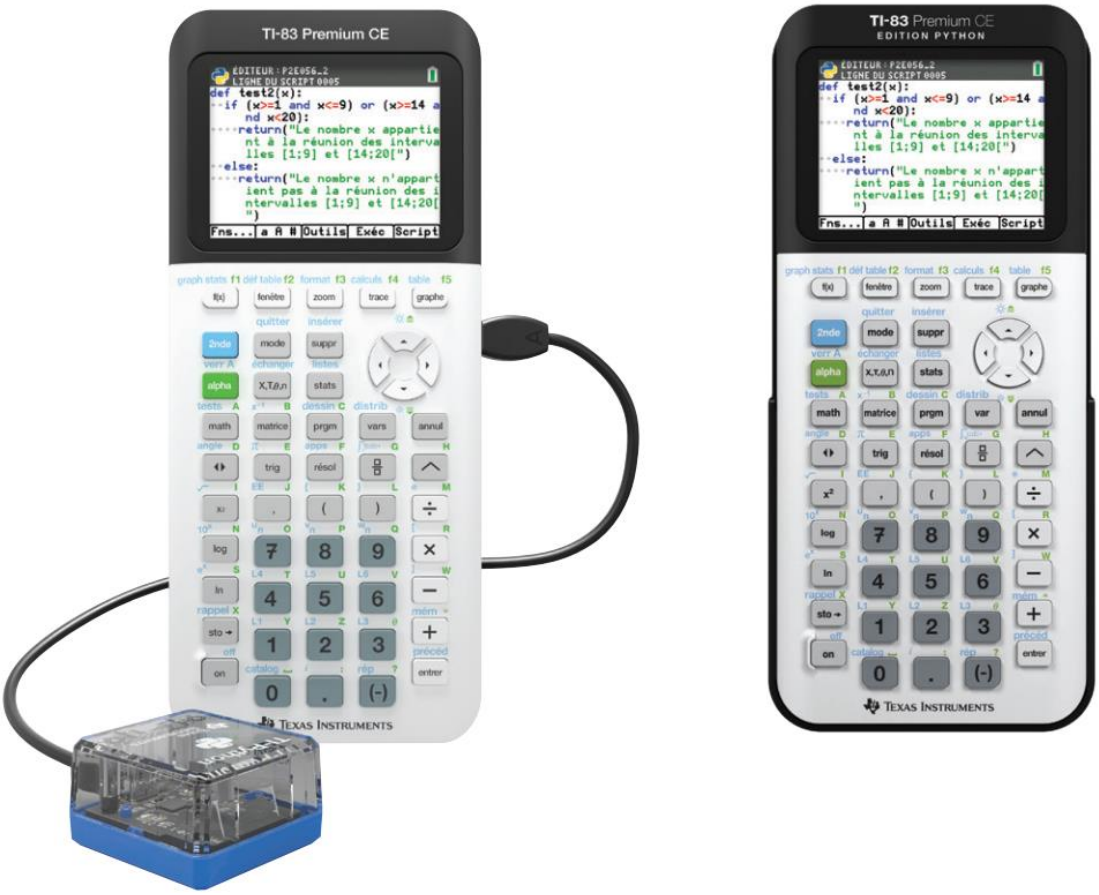
Abir MARINA
Jean-Louis BALAS
T3 FRANCE

TEXAS INSTRUMENTS

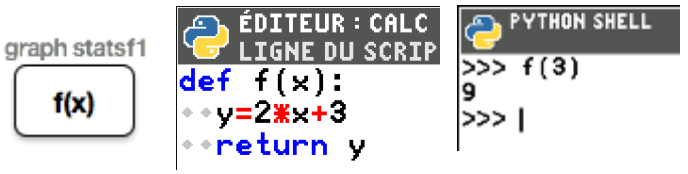
Coding with Python



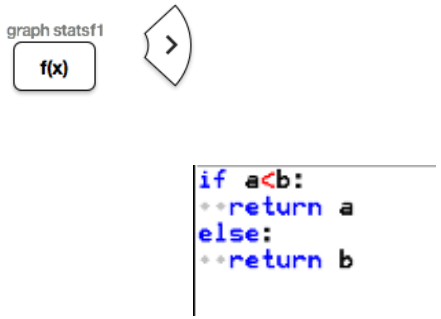
Sharing Inspiration 2019

Coding with Python

Instructions	Python Instructions	note	Example on TI 83 Premium CE
Créer une fonction			

Define a function	def name(p1,p2):
	  instruction(s)
	  return result
		Use *return* instead of *print* in order to return a result
		
		In the shell, it is possible to call a function using var
			

Instruction conditionnelle			

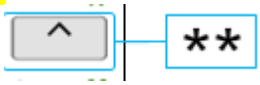
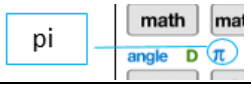
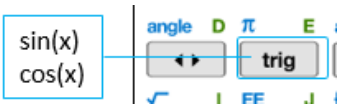
If <condition> do
  <instructions1>
else <instructions2>
End If
	if condition:
	  instructions1
	else:
	  instructions2
		« then » doesn't exist in Python.
		The indent replace it
		
		Same way,it is not needed to end the instruction, indent is enough
			

If <condition> do
  <instructions1>
Else <instructions2>
Else <instructions 3>
End If
	if condition:
	  instructions1
	elif condition2:
	  instructions2
	else:
	  instructions3		

Boucle bornée			
For i going from a to b <instructions> End For	for i in range(n): instructions	From 0 to n-1	
	for i in range(a,b): instructions	For a ≤ n < b	
	for i in range(n,m,k): instructions	For n to m-1 with a k step	
		indent end the loop	
Boucle non bornée			

While <condition> do
  <instructions>
End While
	while <condition>:
	  instructions
		indent end the loop	

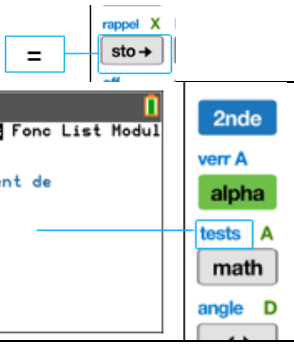
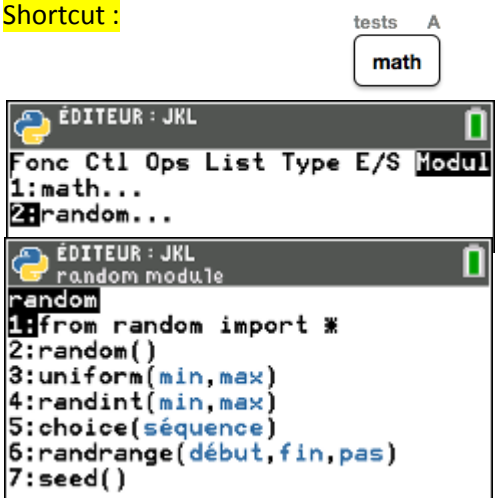
Coding with Python

Instructions	Python Instructions	Note
Opérations et fonctions mathématiques		
a^b	a**b	shortcut : 
Quotient of Euclidian division of a by b	a//b	
Rest of Euclidian division of a by b	a%b	
$\sqrt{a}$	from math import* sqrt(a)	Shorcut : 
π	from math import* pi	Shortcut : 
sin(a)	from math import* sin(a)	Shortcut : 
cos(a)	from math import* cos(a)	
round a with a b precision	round(a,b)	
Return the minimum between a et b	min(a,b)	
Return the maximum between a et b	max(a,b)	
Saisie et affectation		
enter a	a = int(input("a= ")) a = float(input("a= ")) a = input("a= ")	To avoid
Display a	print (a)	
Assign a in the variable x	x = a	natural langage : $x \leftarrow a$ Shortcut : 
Assign a in the variable x and b in the variable y	x,y = a,b	

Need to be imported from the math library

Shorcut : 

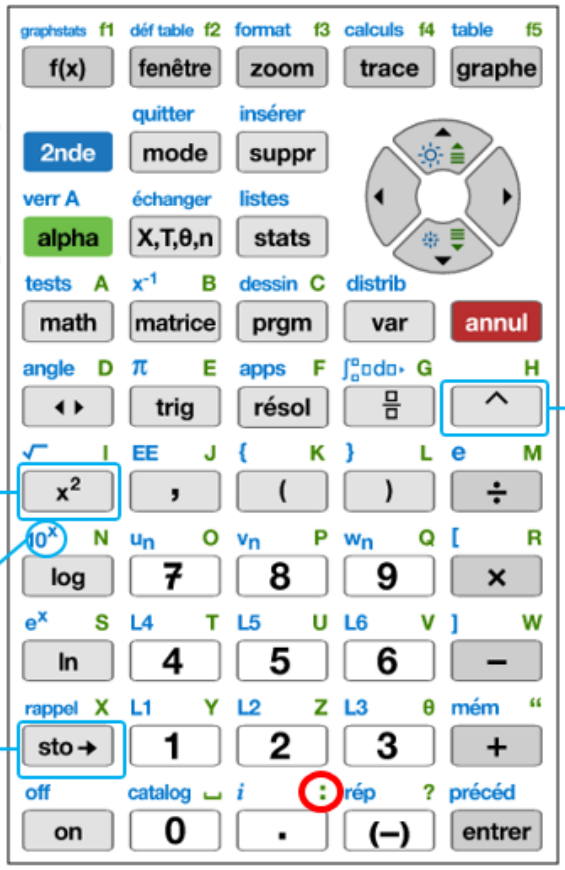
Coding with Python

<table>
<tr><th>Natural langage</th><th>Python Instructions</th><th>Note</th></tr>
<tr><td colspan="3">Tests</td></tr>
<tr><td>Test if $a = b$</td><td>a==b</td><td rowspan="3">Shortcut :
 
</td></tr>
<tr><td>Test if $a \neq b$</td><td>a!=b</td></tr>
<tr><td>Test if $a \leq b$</td><td>a<=b</td></tr>
<tr><td>Test if condition 1 AND condition 2 are verified</td><td>condition1 and condition2</td><td></td></tr>
<tr><td>Test if condition 1 OR condition 2 are verified</td><td>condition1 or condition2</td><td></td></tr>
<tr><td colspan="3">Probability</td></tr>
<tr><td>Generate an integer in $[a;b]$</td><td rowspan="3">Need to be imported from the random library</td><td>from random import*
randint(a,b)</td><td rowspan="3">Shortcut :
 
</td></tr>
<tr><td>Generate a decimal number in $[a;b]$</td><td>from random import*
uniform(a,b)</td></tr>
<tr><td>Generate a decimal number in]0;1[</td><td>from random import*
random()</td></tr>
<tr><td colspan="3">Text</td></tr>
<tr><td>Add text</td><td>“text”</td><td></td></tr>
<tr><td>Add a comment</td><td>#comment</td><td></td></tr>
<tr><td colspan="3">Characters and lists</td></tr>
<tr><td>list</td><td>L=[3,4,2]</td><td>Shortcut :
 
</td></tr>
<tr><td>word’s length</td><td>len(word)
len(list)</td><td></td></tr>
<tr><td>Extract a character</td><td>mot[k]
L[k]</td><td>Return the (k+1) object</td></tr>
<tr><td>Concatenation of two lists</td><td>word3=word1+word2
L3=L1+L2</td><td></td></tr>
<tr><td>Add a number to a list</td><td>a=[9,7,6,9]
a.append(2)</td><td>>>>a
>>>[9,7,6,9,2]</td></tr>
</table>

Coding with TI-Python

[2nde][alpha] & [alpha]

- abcde...
- ABCD...
- alpha indicator at cursor or status bar

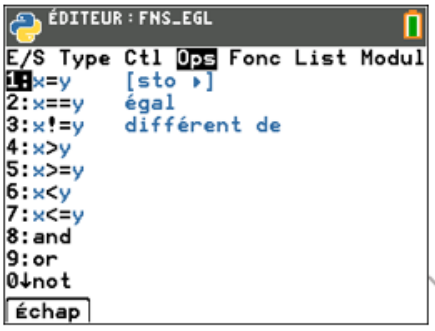


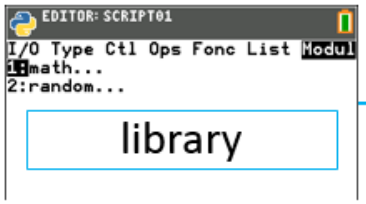
**

** 2

10 **()

=





library

pi

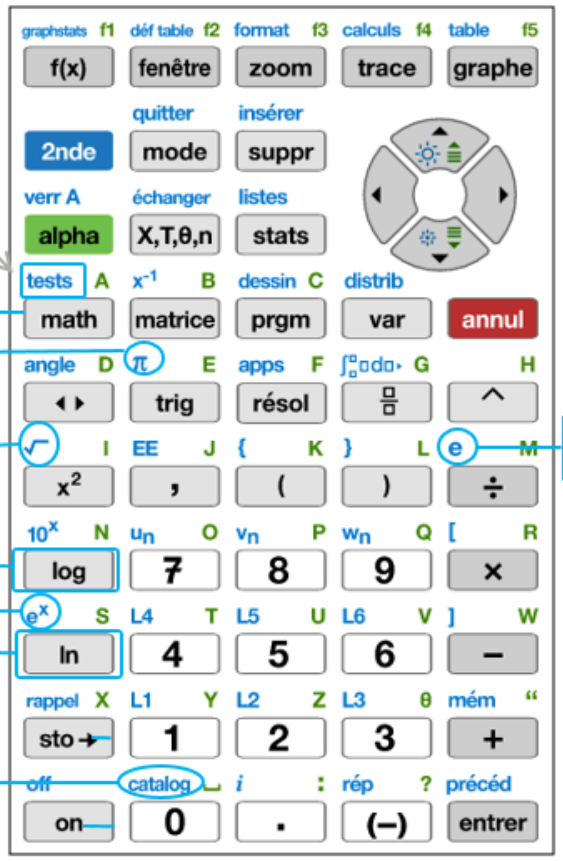
sqrt()

log(x,10)

exp()

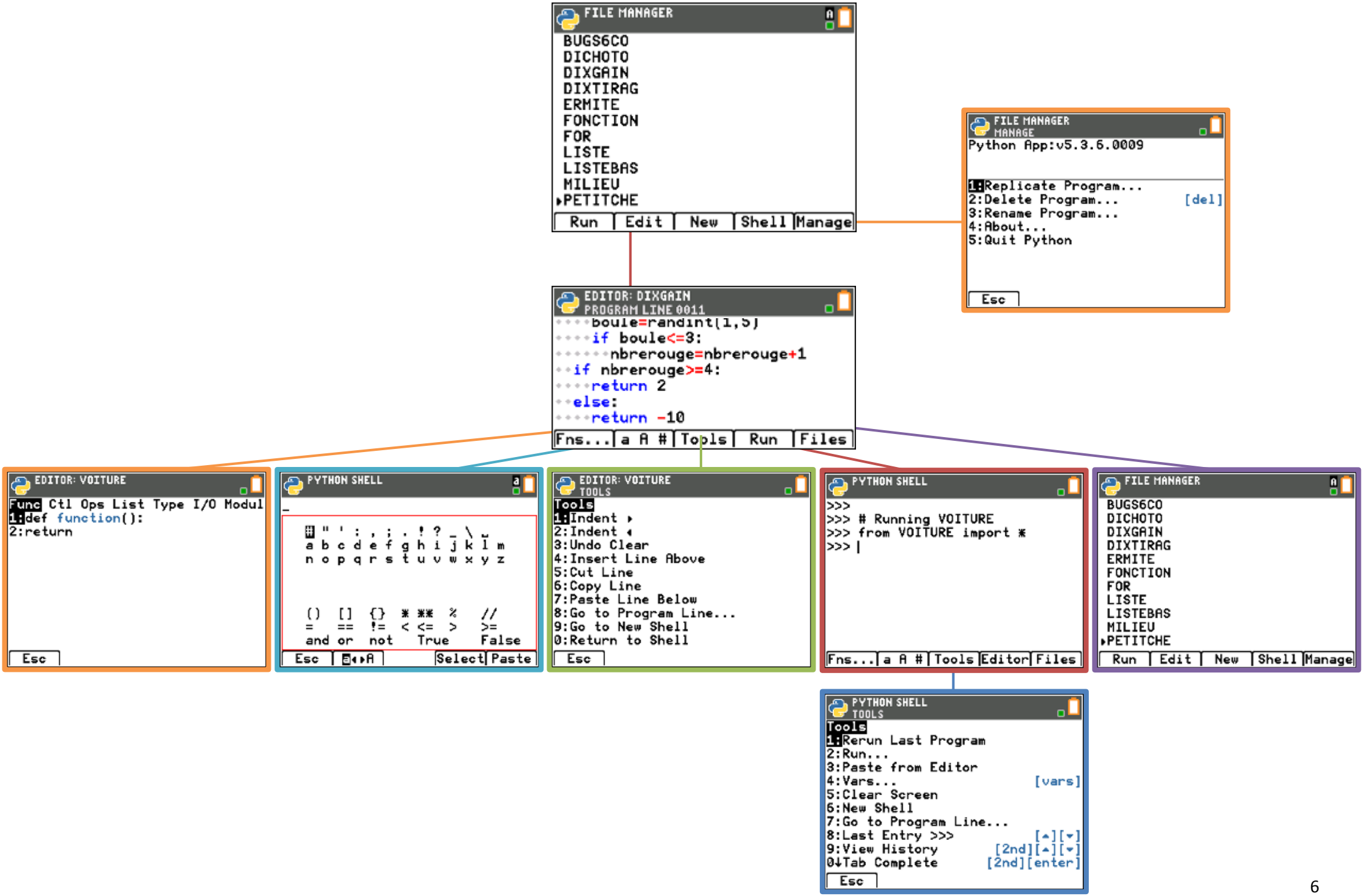
log(x)

all instructions



e

Coding with TI-Python



Coding with TI-Python

```
EDITOR: B
PROGRAM LINE 0003
def f(x):
  y=5*x+3
  return y
```

Function menu

« Fns »

Fns... | a A # | Tools | Run | Files

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:def function():
2:return
Esc
```

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(size):
5:for i in range(start,stop):
6:for i in range(strt,stp,step):
7:for i in list:
8:while condition:
9:elif :
0:else:
Esc
```

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:x=y     [sto ▸]
2:x==y    equal
3:x!=y    not equal
4:x>y
5:x>=y
6:x<y
7:x<=y
8:and
9:or
0↓not
Esc
```

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:[ ]
2:list(sequence)
3:len()
4:max()
5:min()
6:.append(x)
7:.remove(x)
8:.insert(index,x)
9:sum()
0↓sorted()
Esc
```

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:int()
2:float()
3:str()
Esc
```

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:print()
2:input()
3:eval()
Esc
```

Maths library

```
EDITOR: B
Func Ctl Ops List Type I/O Modul
1:math...
2:random...
```

Libraries

« Modul »

Esc

Random library

```
ÉDITEUR : AIREDEF
math module
Math Const Trig
1:from math import *
2:fabs()
3:sqrt()
4:exp()
5:pow(x,y)
6:log(x,base)
7:fmod(x,y)
8:ceil()
9:floor()
0↓trunc()
Modul
```

```
ÉDITEUR : AIREDEF
math module
Math Const Trig
1:e
2:pi
Modul
```

```
ÉDITEUR : AIREDEF
math module
Math Const Trig
1:radians()     degré▸radians
2:degrees()     radians▸degré
3:sin()
4:cos()
5:tan()
6:asin()
7:acos()
8:atan()
9:atan2(y,x)
Modul
```

```
ÉDITEUR : AIREDEF
random module
random
1:from random import *
2:random()
3:uniform(min,max)
4:randint(min,max)
5:choice(séquence)
6:randrange(début,fin,pas)
7:seed()
Modul
```

Activities

Exercise1 : Function

Rewrite this algorithm as concisely as possible using a function.

```
NORMAL FIXE9 AUTO RÉEL RAD MP
EDIT MENU: [alpha] [f5]
PROGRAM:ACT1
:Input "Xa ",A
:Input "Ya ",B
:Input "Xb ",C
:Input "Yb ",D
:(A+C)/2→I
:(B+D)/2→J
:Disp I,J
:
:
```

Exercise 2 : Conditional statement

A photo printing website offers prints at 0.11€ each. The price is reduced to 0.11€ each for orders of more than 200 photos.

Create an algorithm which gives the total price for a number *n* of prints.

Exercise 3 : Closed loop

The population of a village is 2300 today. As the village is growing, its population increases each year by 150 inahbitants.

Design an algorithm which gives the number of inhabitants of this village in *n* years from today.

Exercise 4 : Open loop

On the first January 2018 the price of a new car was 20 000€. Each year the value of the car diminishes by 20%.

Write an algorithm which calculates the number of years which takes the value of the car to below 2000€.

Exercise 5 : the hare and the tortoise

One part of the hare and tortoise game goes like this : The distance to run is 6 squares. The die is thrown and if a six comes up the hare advances 6 squares, otherwise the tortoise goes forward one square.

1) Programme a simulation of this game using Python.

2) Write a piece of script which returns the number of wins of the hare and the tortoise.

Exercise 6 : Primeness test

A prime number is a whole number with exactly two distinct positive divisors (which are 1 and itself). Contrary to this a number which is the non zero product of two distinct whole numbers, neither of which is 1 is said to be composite.

A test for primeness is an algorithm which reveals whether a whole number is prime. The simplest test is the following: to test N, one verifies if it is divisible by one of the whole numbers between 2 and N-1. If the response is negative then N is prime, otherwise it is composite.

Write an algorithm which tests for primeness and returns a boolean. Use the instruction `assert(n>=2)` (found in the instruction catalogue) to verify the hypothesis made in the argument.

Exercise 7 : Approximation of $\sqrt{2}$ by sweeping

Considerate the function $f : x \mapsto x^2$ define on the interval [1;2].

1) Construct the table of variations of the function f on [1;2]. Give the minimum and maximum on this interval.

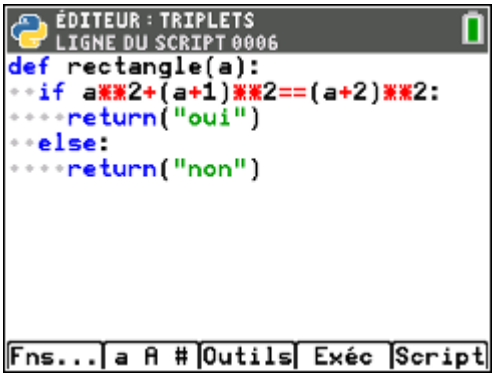
2) Is this table coherent with this sentence : the equation $f(x)=2$ has an only solution on the interval [1;2] wich is $\sqrt{2}$?

3) Write a function « *balayage(epsilon)* » wich return a couple (a,b), with a and b **such as :** $a \leqslant \sqrt{2} \leqslant b$ et $b-a$=epsilon.
For example *balayage*(0.1) must display : (1.4,1.5)

Exercise 8 : Pythagorean Triplet

1) In order to verify automatically whether or not triples of consecutive whole numbers are Pythagorean, the above code was written.

Use the code to test the triples (3,4,5) and (4,5,6).



2) a) Create in the same document and following the function " rectangle " a function "triplet". This should accept a whole number N as argument and test all the consecutive triplets from (1,2,3) up to (N,N+1,N+2) and use the function "rectangle".

b) Test the programme for N=100 then bigger values. What conjecture can you make?

3) **Proof :**

Let a be the smallest of the consecutive whole number Pythagorean Triples.

a) Construct and simplify the equation $a^2 + (a+1)^2 = (a+2)^2$
Show that validating the conjecture is the same as solving the equation $a^2 - 2a - 3 = 0$.

b) Prove that $(a-3)(a+1) = a^2 - 2a - 3$.

c) Solve the equation and write down your conclusions.

7) Write a function that search Pythagorean triplets

Corrections

Exercise 1 : Function

TI Basic

```
PROGRAM:ACT1
:Effécran
:Fixe 2
:Input "Xa ",A
:Input "Ya ",B
:Input "Xb ",C
:Input "Yb ",D
:(A+C)/2→I
:(B+D)/2→J
:Effécran
```

```
:Output(5,5," I Milieu de
[AB] :")
:Output(6,10,"(")
:Output(6,11,I)
:Output(6,17,";")
:Output(6,19,J)
:Output(6,23,")")
```

```
Fait

I Milieu de [AB] :
   (3.50  ; 4.50)
```

Python

```
def milieu(xa,ya,xb,yb):
    xi=(xa+xb)/2
    yi=(ya+yb)/2
    return xi,yi
```

```
>>> milieu(2,4,4,6)
(3.0, 5.0)
>>>
```

Exercise 2 : Conditional statement

TI Basic

```
PROGRAM:ACT2
:Effécran
:Fixe 2
:Input "Number ",N
:If N<200
:Then
:0.11*N→M
:Else
:0.08*N→M
:End
```

```
:Output(5,5,"Price to pay
:")
:Output(5,19,M)
```

```
Number 100
                    Fait

Price to pay :11.00
```

Python

```
def photo(n):
    if n<200:
        M=0.11*n
    else:
        M=0.08*n
    return M_
```

```
>>> photo(165)
18.15
>>> photo(314)
25.12
>>>
```

Exercise 3 : Closed loop

TI Basic

```
PROGRAM:ACT3
:Effécran
:Input "Population ? ",N
:2300→P
:For(I,1,N)
:P+150→P
:End
:Disp P
```

```
Population ? 1500
                 227300.00
                    Fait
```

Python

```
def population(n):
    p=2300
    for i in range(1,n+1):
        p=p+150
    return p_
```

```
>>> population(28)
6500
>>>
```

Exercise 4 : Open loop

TI Basic

```
PROGRAM:ACT4
:Effécran
:Fixe 2
:Input "Price ? ",V
:0→N
:While V≥2000
:0.8*V→V
:1+N→N
:End
:Disp N
```

```
Price ? 4000
                    4.00
                    Fait
```

Python

```
def prix(v):
    n=0
    while v>=2000:
        v=0.8*v
        n=n+1
    return n_
```

```
>>> prix(20000)
11
>>> |
```

Exercise 5 : The hare and the tortoise

TI Basic

```
PROGRAM:ACT5
:Effécran
:Fixe 0
:0→N:0→D
:While D<6 et N<6
:nbrAléatEnt(1,6)→D
:N+1→N
:Disp D
:End
:If N=6
:Then
:Disp "The Turtle wins"
:Else
:Disp "The hare wins"
:End
:Fixe 9
```

```
                    1
                    4
                    3
                    5
                    1
                    3
The Turtle wins
                    Fait
```

Python

```
from random import randint

def course():
    de=0
    case=0
    while de<6 and case<6:
        de=randint(1,6)
        case=case+1
        print (de)
    if case==6:
        return "la tortue a gagn
    é"
    else :
        return "le lièvre a gagn
    é"
```

```
>>> course()
4
3
3
2
5
4
'la tortue a gagné'
>>> |
```

Exercise 6 : Primeness test

TI Basic

```
PROGRAM:PRIMALIT
:Input "Number ",N
:For(I,2,N/2)
:If not(remainder(N,I))
:Then
:Disp "False"
:Stop
:End
:End
:Disp "True"
```

```
prgmPRIMALIT
Number 5
True
                    Fait
prgmPRIMALIT
Number 8
False
                    Fait
```

Python

```
def premier(n):
    assert n>=2
    for i in range(2,n):
        if n%i==0:
            return False
    return True_
```

```
>>> premier(9)
False
>>> premier(19)
True
>>> |
```

Exercise 7 : Approximation of $\sqrt{2}$ by sweeping

TI Basic

```
PROGRAM:ACT7
:Effécran
:Fixe 3
:1→X
:Input "EPSILON ?",E
:While X²<2
:X+E→X
:End
:Output(5,3,"[X-ε,X] :")
:Output(5,13,X-E)
```

```
PROGRAM:ACT7
:1→X
:Input "EPSILON ?",E
:While X²<2
:X+E→X
:End
:Output(5,3,"[X-ε,X] :")
:Output(5,13,X-E)
:Output(5,17,",")
:Output(5,19,X)
```

```
EPSILON ?0.1
                          Fait

  [X-ε,X] : 1.40, 1.500
```

Python

```
def balayage(epsilon):
    x=1
    while x**2<2:
        x=x+epsilon
    return (x-epsilon,x)
print(balayage(0.1))
```

```
BUGS4CO...
BUGS1CO...
BUGS2CO...
BUGS3CO...
Configuration terminée.

>>> # L'exécution de RACINE2
>>> from RACINE2 import *
(1.4, 1.5)
>>>
```

Exercise 8 : Pythagorician Triplet

TI Basic

```
PROGRAM:ACT61
:Prompt N
:For(I,1,N)
:If I²+(I+1)²=(I+2)²
:Then
:{I,I+1,I+2}→L1
:Disp L1
:Wait 2
:End
:End
```

```
PROGRAM:ACT6
:Effécran
:Prompt A
:If A²+(A+1)²=(A+2)²
:Then
:Disp "YES"
:Else
:Disp "NO"
:End
```

```
A=?3
YES
                          Fait
```

```
N=?10
                        {3 4 5}
                          Fait
```

Python

```
def tripletPythagoricien(N):
    a=1
    for loop in range(N):
        if rectangleOuNon(a)=='oui':
            print('(',a,',',a+1,',',a+2,')')
        a=a+1
```

```
def rectangleOuNon(a):
    if a**2+(a+1)**2==(a+2)**2:
        return('oui')
    else:
        return('non')
```

```
def BONUS(N):
    for a in range(1,101):
        for b in range(a,101):
            for c in range(b,101):
                if a**2+b**2==c**2:
                    print('(',a,',',b,',',c,')')
```