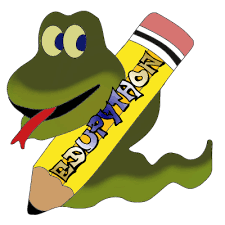
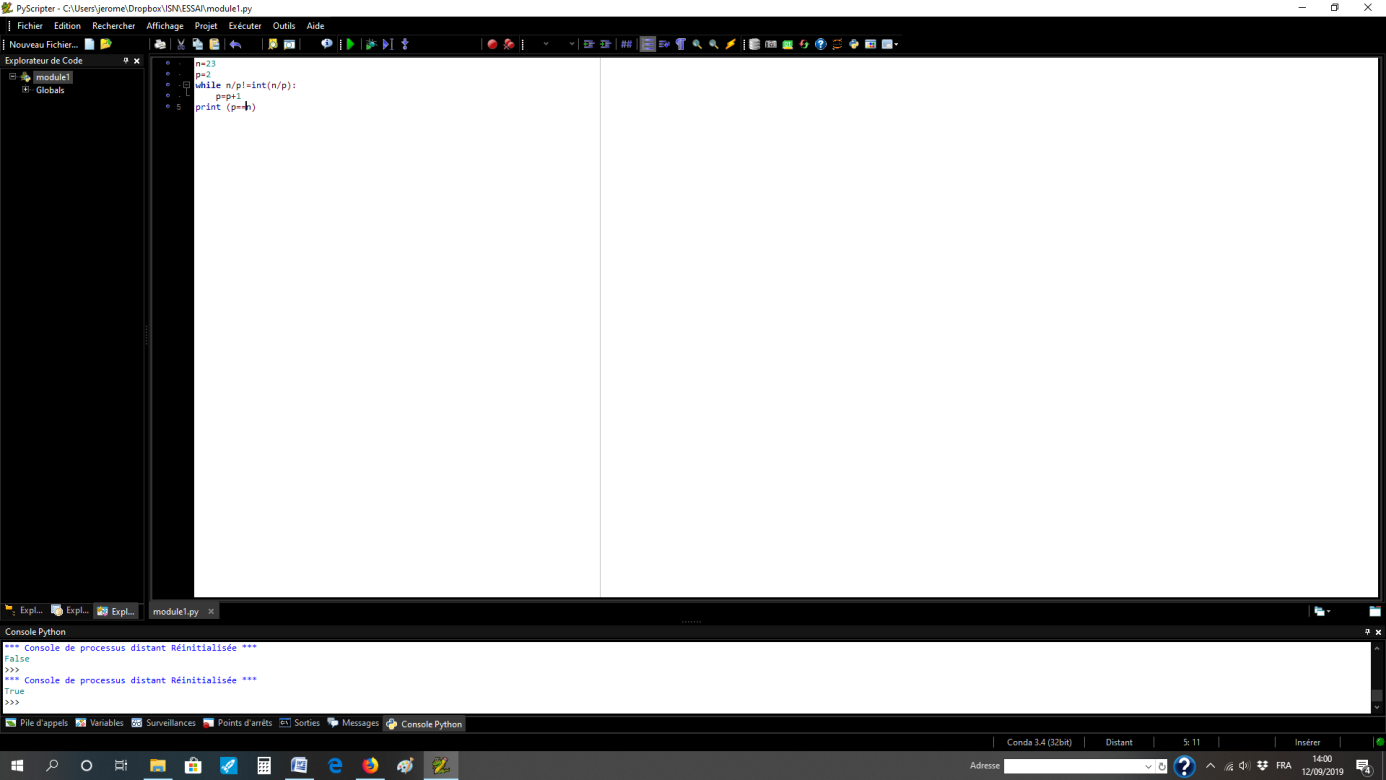
**Lancer EduPython **



La console: on peut voir le résultat du programme ou lancer des instructions simples

la flèche verte lance le programme

L'éditeur: ici on écrit le programme

**Pour commencer dans la console**

**Lancer les instructions suivantes dans la console en appuyant sur Enter à chaque fin de ligne et regarder le resultat:**

**>>>** print ( "Hello world !")

>>> x=3

>>> print(x)

>>>4+5

>>>5/2

>>>5//2

**>>> print("la valeur de x est", x)**

**La console sert à essayer des instructions simples. Il ne faut pas hésiter à l'utiliser dés que l'on a un doute sur une instruction.**

**Pour commencer un programme**

**Enregistrer IMMEDIATEMENT votre programme dans votre répertoire de travail avec le nom Mon1erProgramme.py en faisant:**

**Cliquer sur Fichier puis Nouveau puis sélectionner Nouveau Module Python.**

**Nous allons maintenant faire des programmes très simples pour comprendre la syntaxe de python.**

**Voici des algorithmes écrit en pseudo code et sa traduction en python. Ecrire le code en python dans l'éditeur et lancer le en appuyant sur la flèche verte. Mettre le résultat dans la dernière colonne. Vous pouvez écrire tous les codes à la suite dans le même fichier. Séparer les par exemple par**

**print("pseudo code i") où i est le numéro du pseudo code.**

**input** sert à insérer des chaines de caractères dans une variable de type chaine de caractère (str)

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 1** | **Python** | **Affichage** |
| **Afficher "Entrer Votre nom"**  **Entrer Nom**  **Afficher "Votre nom est ",Nom** | **Nom=input("Entrer Votre nom")**  **print("Votre nom est ",Nom)** |  |

Nom est une variable de type str (string)

Chaine de caractère

Il y a 6 valeurs quand on va de 0 à 5

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 2** | **Python** | **Affichage** |
| **Pour** N allant de 0 à 5    Afficher N **Fin** **Pour** | **for N in range(0,6):**  **print(N)** | ATTENTION les indentations sont obligatoires en python. |

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 3** | **Python** | **Affichage** |
| **Pour** N allant de 0 à 5    Afficher "N" **Fin** **Pour** | **for N in range(0,6):**  **print("N")** | Les guillemets affichent les caractères (str) |

Choisissez par exemple 7 pour la dernière valeur.

**Je voudrais afficher la plage de valeur de mon choix**

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 4** | **Python** | **Affichage** |
| **Afficher "Quelle est la dernière valeur?"**  **Entrer Der**  **Pour** N allant de 0 à Der    Afficher N Fin **Pour** | **Der=input("Quelle est la dernière valeur?"**  **Der=int(Der)**  **for N in range(0,Der+1):**  **print(N)** |  |

Il faut convertir un str en entier (int). Essayer sans cette ligne pour voir le résultat

**Je voudrais calculer a+b, a×b, a/b et a-b sachant que a=123456789 et b=987654321.**

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 5** | **Python** | **Affichage** |
| **a←123456789**  **b←987654321**  **somme←a+b**  **produit←a×b**  **quotient←a/b**  **difference←a-b**  **Afficher("a+b",somme)**  **fficher("a×b",produit)**  **Afficher("a/b",quotient)**  **Afficher("a-b", difference)** | **a=123456789**  **b=987654321**  **somme=a+b**  C'est \* et pas × ni x !!  **produit=a\*b**  **quotient=a/b**  **difference=a-b**  **print("a+b",somme)**  **print ("a×b",produit)**  **print ("a/b",quotient)**  **print ("a-b", difference)** |  |

**Remarquer que le symbole = sert à l'affectation mais il ne veut pas dire "égal" en python, il veut dire "prendre la valeur ". C'est pour cela qu'on utilise la flèche en pseudo code.**

L' algorithme suivant calcule la somme des premier entiers de 0 à N. A vous maintenant de traduire en python. Attention quand on va de 0 à N, il y a N+1 valeurs.

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 6** | **Affichage Pour N=4** | **Affichage Pour N=6** |
| **Afficher "Valeur de N ?"**  **Entrer N**  **K←0**  **Pour** P allant de 0 à N     K=K+P  **Fin** **Pour**  Afficher K |  |  |

Appeler moi quand vous avez réussi celui ci.

Boucle Tant que

On peut aussi utiliser une boucle **Tant que** si on ne sait pas le nombre d'action à effectuer. Une boucle **Pour** peut être remplacer facilement par une boucle **Tant que.**

Mettons une boucle **Tant que**  à la place d'une boucle **Pour**  dans le pseudo code 2 et testez le.

|  |  |  |  |
| --- | --- | --- | --- |
| **Pseudo-code 2 avec Pour** | **Python avec Pour** | **Pseudo-code 2.1 avec Tant que** | **Python avec Tant que** |
| **Pour** N allant de 0 à 5    Afficher N **Fin** **Pour** | **for N in range(0,6):**  **print(N)** | **N←0**  **Tant que N<6**  Afficher N  **FinTant que** | **N=0**  **while N<=5:**  **print(N)**  **N=N+1** |

L'algorithme suivant affiche toute les valeurs de la suite définie par Un+1= 5Un +3 et U0=10 qui sont inférieur à 1000:

|  |  |  |
| --- | --- | --- |
| Pseudo code 7 | python | Affichage |
| **U**←10  **Tant** **que** U<1000  **Afficher** U  U=5×U+3  **FinTantQue** | **U**=10  while U<1000:  print(U)  U=5\*U+3 |  |

Saut conditionnel

L'algorithme suivant demande un nombre décimal (float) et teste s'il est négatif ou positif:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pseudo code 7.0 | python | x=-5.2 | x=0 | x= –60.7 |
| **Afficher "**Entrer un décimal"  **Entrer** x  **Si** x>0  **Afficher** x,"est positif"  **Sinon**  **Afficher** x,"est négatif  **Finsi** | **x=input("**Entrer un décimal")  x=float(x)  if x>0:  print(x,"est positif")  else:  print(x,"est negatif") |  |  |  |

Il y a un petit problème, l'algorithme dit que 0 est négatif alors que c'est un nombre négatif et positif. Il faut un cas particulier pour 0. Voici un nouvel algorithme qui corrige ce problème.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pseudo code 7.1 | python | x=-5.2 | x=0 | x= –60.7 |
| **Afficher "**Entrer un décimal"  **Entrer** x  **Si** x=0  **Afficher** x,"est positif et negatif"  **Sinon** **si** x>0  **Afficher** x,"est positif"  **Sinon**  **Afficher** x,"est négatif  **FinSi** | **x=input("**Entrer un décimal")  x=float(x)  if x==0:  print(x,"est positif et negatif")  elif x>0:  print(x,"est positif")  else:  print(x,"est negatif") |  |  |  |

Remarquer la difference entre x=float(x), le = affecte une valeur à x

x==0 , le == vérifie une condition

L'algorithme suivant teste si un entier est un nombre premier.

* != veut dire "different de"
* int(x) c'est l'entier juste inférieur à x
* while N/P!=int(N/P) veut donc dire "Tant que le decimal N/P est different de l'entier juste inférieur à N/P" ce qui revient à "**Tant** **que** N/P n'est pas un entier".

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pseudo code 8 | python | N=2 | N=3 | N=6 | N=23 | N=1047 |
| **Entrer** N plus grand ou égal à 2  P←2  **Tant** **que** N/P n'est pas un entier  P=P+1  **FinTantQue**  **Si** P=N  **Afficher** "Vrai"  **Sinon**  **Afficher** "Faux" | **N=input("Entrer N plus grand ou egal a 2")**  **N=int(N)**  P=2  while N/P!=int(N/P):  P=P+1  if P==N:  print("Vrai")  else:  print("Faux") |  |  |  |  |  |

Exercice 1:

coder en python l'algorithme suivant qui compte le nombre de carré parfait inférieur à N:

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 9** | **Affichage Pour N=10** | **Affichage Pour N=100** |
| **Afficher "Valeur de N ?"**  **Entrer N**  **Nb=0**  **Pour i=0 à 10**  **Si** **est un entier**  **Nb=Nb+1**  **FinSi**  **FinPour**  **Afficher Nb** |  |  |

Exercice 2:

Que fait l'algorithme suivant ? Coder le en python:

|  |  |  |
| --- | --- | --- |
| **Pseudo-code 9** | **Affichage Pour N=4** | **Affichage Pour N=10** |
| **Entrer** N  P=1  **Pour** i=1 à N  P=P×i  **FinPour**  **Afficher** P |  |  |