

Algorithmique	Langage C	Python 3.6
Structure générale		
Début du programme	<pre>int main() { return 0 ; }</pre>	Aucune structure
Fin du programme		
Déclaration des variables		
i : entier	<pre>int i ;</pre>	Aucune déclaration : la déclaration d'une variable se fait simplement en lui affectant une valeur
rayon : réel	<pre>float rayon ;</pre>	
nom : chaîne de caractère	<pre>char nom[10]; // 10 caractères maxi</pre>	
ma_lettre : caractère	<pre>char ma_lettre; // 1 caractère</pre>	
OuiNon : booléen		
Affectation d'une variable		
i <- 0	<pre>i=0;</pre>	<pre>i = 0</pre>
rayon <- 8.20	<pre>rayon=8.2;</pre>	<pre>rayon = 8.2</pre>
nom <- "Nestor"	<pre>strcpy(nom, "Nestor");</pre>	<pre>nom = "Nestor" # ou nom='Nestor'</pre>
ma_lettre <- 'd'	<pre>ma_lettre = 'd';</pre>	N'existe pas
OuiNon <- VRAI		<pre>OuiNon = True</pre>
Afficher un message		
Afficher "Bonjour "	<pre>printf("Bonjour\n");//avec saut de ligne printf("Bonjour");// sans</pre>	<pre>print('Bonjour') # avec un saut de ligne print('Bonjour',end="");// sans</pre>
Afficher "i=",i	<pre>printf("i=%d\n",i);</pre>	<pre>print('i=',i)</pre>
Afficher "rayon=",rayon	<pre>printf("rayon=%f\n", rayon);</pre>	<pre>print("rayon=", rayon)</pre>
Afficher "nom=", nom	<pre>printf("nom=%s", nom)</pre>	<pre>print("nom=", nom)</pre>
Afficher " ma_lettre =", ma_lettre	<pre>printf("ma_lettre=%c\n", ma_lettre);</pre>	
Afficher OuiNon	<pre>printf("OuiNon=%d\n",OuiNon); // =1</pre>	<pre>print('OuiNon=',OuiNon)</pre>
Saisir au clavier et affecter		
Demander nom	<pre>scanf("%s", nom);</pre>	<pre>nom=input()</pre>
Demander i	<pre>scanf("%d",&i);</pre>	<pre>i=int(input())</pre>
Demander rayon	<pre>scanf("%f",&rayon);</pre>	<pre>rayon=float(input())</pre>
Demander ma_lettre	<pre>scanf("%c",&ma_lettre);</pre>	N'existe pas

Commentaires		
Remarque	// Sur une ligne /* Sur plusieurs lignes */	# Sur une ligne """ Sur plusieurs lignes """
Actions conditionnelles		
Si (note>10) alors afficher « reçu » afficher « bravo » sinon afficher « recalé»	<pre>if (note>10) { printf("reçu"); printf("bravo"); } else { printf("recalé"); }</pre>	<pre>if note>10: print("reçu") #une tabulation avant print("bravo") else: print("recalé")</pre>
Saisir jour Selon jour : Cas 1 : afficher "lundi" Cas 2 : alors afficher "mardi" ... Cas 7 : alors afficher "dimanche" Sinon : ("Erreur : le jour doit être compris entre 1 et 7") FinSelon	<pre>int jour; printf("Entrez un jour de la semaine :"); scanf("%d", &jour); printf("\n"); switch (jour) { case 1 : printf("lundi\n"); break; case 2 : printf("mardi\n"); break; // et ainsi de suite case 7 : printf("dimanche\n"); break; default : printf("Erreur : le jour doit être compris entre 1 et 7\n"); }</pre>	<pre>#le switch case n'existe pas jour =int(input("Entrez un jour de la semaine :")) if (jour==1): print("Lundi") elif (jour==2): #mélange de else if print("Mardi") #et ainsi de suite else: print("Erreur : le jour doit être compris entre 1 et 7")</pre>
Conditions booléennes		
Si (i=0) ET (rayon ≠0)	<pre>if ((i==0) && (rayon!=0))</pre>	<pre>if (i==0) and (rayon!=0) :</pre>
Si (aire≤10) OU (aire ≥20)	<pre>if ((aire <=10.0) (aire >=20.0))</pre>	<pre>if (aire <=10.0) or (aire >=20.0) :</pre>
Si nom= "Nestor"	<pre>if (strcmp(nom, "Nestor")==0)¹</pre>	<pre>if nom=="Nestor" :</pre>
Boucles		
Pour i variant de 1 à 10 afficher(i) Fp	<pre>for(i = 1; i <=10; i=i+1) { printf("i=%d\n", i); }</pre>	<pre>for i in range(1,11): print("i=", i)</pre>
Tant que somme<100 faire somme=2*somme+1	<pre>while (somme<100) { somme=2*somme+1; } printf("somme=%d\n", somme);</pre>	

¹ #include <string.h>

Reponse : caractère répéter lire reponse tant que reponse ≠ 0	<pre>do { printf("Voulez-vous sortir de la boucle (O/N) ?"); scanf("%c", &reponse); printf("\n"); } while (reponse != '0');</pre>	<pre>while True: reponse = input("Voulez-vous sortir de la boucle (O/N) ? ") if reponse=="0": break</pre>
Tableaux		
tab : tableau de 3 entiers tab[0]←-1 tab[1]←-10 tab[2]←-100 Afficher tab tab[0]←-2 tab[1]←-20 tab[2]←-200	<pre>int tab[3]={1,10,100}; for (int i = 0; i < 3; i++) printf("tab[%d]=%d ", i, tab[i]); // affiche tab[0]=1 tab[1]=10 tab[2]=100 tab[0] = 2; // 1er élément tab[1] = 20; // 1er élément tab[2] = 200; //dernier élément for (int i = 0; i < 3; i++) printf("tab[%d]=%d ", i, tab[i]); // affiche tab[0]=2 tab[1]=20 tab[2]=200</pre>	<pre>tab=[1,10,100] ² print(tab) #affiche [1, 10, 100] tab[0]=2 tab[1]=20 tab[2]=200 print(tab) #affiche [2, 20, 200]</pre>
tableau2D : tableau de 4 sur 3 entiers Afficher tableau2D	<pre>int tab2D[4][3]={{0,0,0},{1,1,1},{2,2,2},{3,3,3 }};// 4lignes*3colonnes for (int ligne = 0; ligne < 4; ligne++) for (int col = 0; col < 3; col++) printf("tab2D[%d][%d]=%d\n ", ligne,col, tab2D[ligne][col]); // affiche tab2D[0][0]=0 tab2D[0][1]=0 tab2D[0][2]=0 tab2D[1][0]=1 ...</pre>	<pre>tab2D=[[0,0,0],[1,1,1],[2,2,2],[3,3,3]] print(tab2D) #affiche [[0, 0, 0], [1, 1, 1], [2, 2, 2], [3, 3, 3]]</pre>
noms : tableau de chaînes contenant "Alice", "Bobbii", "Samia", et "Zheng-You" Afficher noms[1]	<pre>char noms[][16] = {"Alice", "Bobbie", "Samia", "Zheng-You"}; printf("%s",noms[1]); // Affiche Bobbie</pre>	<pre>noms = ["Alice", "Bobbie", "Samia", "Zheng-You"] print(noms[1]) #affiche Bobbie</pre>
Fonctions		
Les procédures (fonction sans retour)		
Définir procédure tirerUnTrait Aller à la ligne Afficher "-----" Aller à la ligne	<pre>void tirerUnTrait()³{ println("\n-----\n"); } L'appel se fait dans le programme principal : tirerUnTrait() ;</pre>	<pre>def tirerUnTrait():⁴ print("-----") tirerUnTrait()</pre>

² Utilisation des listes concept bien plus puissant (avec des fonctions comme tab.append pour ajouter à la fin, len(tab) longueur, del(tab[1]) supprimer ...)

³ La déclaration se fait avant le main()

⁴ La déclaration peut se faire au milieu du code, il est vivement conseillé de le mettre en tout début

Fonctions avec arguments et valeur de retour

<p>Fonction abs Entrée: x: entier Sortie: entier Début Si (x > 0) Alors Retourner x Sinon Retourner -x Fin_abs</p> <p>Afficher("Entrer y: ") Saisir(y) Afficher abs(y)</p>	<pre>int abs(int x) { if (x > 0) { return x; } else { return -x; } } int y; printf("Entrer y: "); scanf("%d", &y); printf("%d\n", abs(y));</pre>	<pre>def abs(x): if (x>0): return x else: return -x y=int(input("Entrer y: ")) print(abs(y)) <i>#affiche par ex 2 si y=-2</i></pre>
Opérations diverses		
<p>Afficher ("13 modulo 5 = " 13 modulo 5) Le module est le reste de la division entière</p>	<pre>printf("13 modulo 5 = %d",13%5); // affiche 3</pre>	<pre>print("13 modulo 5 = ",13%5)</pre>
<p>Afficher ("un nombre aléatoire entre 0 et 10") Afficher ("un nombre aléatoire entre 1 et 10")</p>	<pre>srand (time(NULL)); // indispensable⁵ printf("nombre aléatoire entre 0 et 10 : %d ", rand() % 11); printf("nombre aléatoire entre 1 et 10 : %d ", rand() % 10 + 1);</pre>	<pre>import random print("nombre aleatoire entre 0 et 10 :" , random.randint(0, 10)) print("nombre aleatoire entre 1 et 10 :" , random.randint(1, 10))</pre>

⁵ L'initialisation du générateur de nombres aléatoires, obligatoire sinon on aura toujours le même nombre aléatoire à chaque exécution du programme