

CODAGE DES BOUCLES WHILE

<p>Codage de la boucle Tant que (While)</p>	<p>Première méthode pour gérer une note supérieure à 20 :</p> <pre>notel=-1 while (notel>20): notel = input("Quelle est votre note (entière) ? ") notel=int(notel) print("Votre note est ", notel)</pre> <p>Deuxième méthode pour gérer une note supérieure à 20 :</p> <pre>test = True while test: notel = input("Quelle est votre note (entière) ? ") notel=int(notel) if (notel>=0 and notel<=20): test=False print("Votre note est ", notel)</pre>
---	---

Le test TRY EXCEPT

<p>Gestion des erreurs lors de l'exécution pour traiter correctement les nombres.</p> <p>Gestion des exceptions dans les erreurs</p> <p>try : except *** :</p>	<pre># -*-coding:Utf-8 -* import os # Acquisition des donnees note 1 test = True while test: notel = input("Quelle est votre note (entière) ? ") try: notel = float(notel) except ValueError: notel = -1 if (notel>=0 and notel<=20): test = False print("Votre note est ", notel) os.system("pause")</pre>
---	---

SORTIR DES ACTIONS

<p>Sortir d'une boucle</p>	<p>L'instruction break permet de sortir immédiatement par le bas de la boucle en cours.</p> <p>L'instruction continue permet de revenir immédiatement en haut de la boucle pour continuer le traitement.</p>
-----------------------------------	--

CODAGE DES BOUCLES FOR NOMINATIVES

Codage de la boucle POUR (FOR)	<p>Avec une chaîne de caractère : Affichage des caractères un à un</p> <pre>y = "Hello World!" # On crée le string y # x est ici un caractère for x in y: # Pour chaque caractère x contenu dans y print(x) # Affiche x à l'écran</pre> <p>Avec une chaîne de caractère : Affichage des caractères un à un</p> <pre>y = "Hello World!" # On crée le string y nLongueur = len(y) # x est ici un integer qui correspond au numéro de case for x in range(nLongueur): # x est ici le numéro de la case print(y[x])</pre> <p>Lecture des éléments contenus dans une liste :</p> <pre>y = [1,2,9,8,"ah ah ah",45,(45,45,45)] for x in y: print(x)</pre> <p>Affichage des mots contenus dans une phrase</p> <pre>phrase = "Désormais, nous allons pouvoir gérer ..." mots = phrase.split() # mots est la liste issue du "split" for mot in mots: # pour chaque élément mot contenu dans mots print(mot) # on affiche le mot</pre>
--	---

Structure de données : l'objet LIST

La reconnaître	<p>Une liste est un objet pouvant contenir d'autres objets. Elle commence par un crochet d'ouverture et finit avec un crochet de fermeture <code>[]</code>. On sépare les différents éléments par des virgules :</p> <pre>maListe = [element0, element1, element2]</pre>
Lecture directe	<pre>maListe = [element0, element1, element2] print(maListe[0])</pre> <p>Ce code va afficher le contenu de l'élément en position 0 dans la liste.</p>
Lecture automatique	<p>On peut utiliser une boucle FOR (voir la page précédente)</p> <pre>y = [1,2,9,8,"ah ah ah",45,(45,45,45)] for x in y: print(x)</pre>
Fonction len()	<p>Renvoie le nombre d'éléments contenus dans une liste, son « nombre de cases ».</p> <pre>nombreElements = len(maListe)</pre>
Méthode append()	<p>Cette méthode permet de rajouter un élément à la fin de la liste.</p> <pre>maListe = ["a", "b", "c"] maListe.append("d") print(maListe)</pre> <p>Affiche : ["a", "b", "c", "d"]</p>

L'objet LIST et Tkinter et la méthode append

Une liste peut contenir les références de vos widgets SANS AVOIR à leur donner de noms : on peut créer des widgets dans une boucle et néanmoins parvenir à les modifier.
On peut lire le contenu d'une liste avec un FOR numérique ou un FOR nominatif.

String	<p>C'est un assemblage de caractère. Le premier a le numéro 0. On peut accéder au contenu de la case i à l'aide de la syntaxe <code>chaîne[i]</code></p> <p><u>Exemple :</u></p> <pre>chaîne = "Bonjour les gens" print(chaîne[0]) # Affiche B print(chaîne[0:10]) # Affiche Bonjour le print(chaîne[2:10]) # Affiche njour le print(chaîne[2:10:2]) # Affiche norl</pre> <p>Attention : Un string n'est pas modifiable. Pour « modifier » une chaîne de caractère, il faut en créer une nouvelle avec le même nom.</p> <p><u>Exemple :</u></p> <pre>chaîne = "Bonjour" chaîne[0] = "C" #Provoque une erreur car modification impossible.</pre>
Concaténation	<p>Il s'agit du fait d' « additionner » des chaînes de caractères entre elles.</p> <pre>y1 = "Bonjour" y2 = " à tous" y3 = y1 + " " + y2 # y3 contient "Bonjour à tous"</pre>
Fonction <code>len()</code>	<p>Renvoie le nombre de caractères contenus dans un string, son « nombre de cases ».</p> <pre>nLongueur = len(monTexte)</pre>
Fonction <code>int()</code>	<p>Convertit si possible la chaîne en interger et renvoie le résultat.</p> <pre>chaîne = "42" x = int(chaîne) # x contient le nombre 42, pas la chaîne « 42 »</pre>
Méthodes <code>lower()</code> et <code>upper()</code>	<pre>x.lower()</pre> renvoie x en minuscule si x est une lettre ou une chaîne. Pour MODIFIER x, il faudrait donc noter <code>x = x.lower()</code> . <p> </p> <pre>x.upper()</pre> renvoie x en majuscule si x est une lettre ou une chaîne. Pour MODIFIER x, il faudrait donc noter <code>x = x.lower()</code> .
Méthode <code>split()</code>	<p>Renvoie une liste de « mots » contenu dans une chaîne. Pour lire la liste, voir le FOR.</p> <pre>mots = chaîne.split() # le séparateur est l'espace mots = chaîne.split(',') # le séparateur est la virgule</pre>
Méthode <code>replace(old, new)</code>	<p>Remplace la chaîne old par la chaîne new.</p> <pre>phrase = phrase.replace(".", "") # Remplace les points par « rien » y1 = y1.replace("a", "*") # Remplace les a par des étoiles *</pre>
Méthode <code>isprintable()</code>	<p>Cette méthode renvoie True si le caractère est un vrai caractère imprimable, par oppostion à caractère de contrôle.</p>
Méthode <code>isalpha()</code>	<p>Cette méthode renvoie True si le caractère est alphabétique, False sinon. Si x vaut "abc", alors <code>x.isalpha()</code> renvoie True Si x vaut "a:c" alors <code>x.isalpha()</code> renvoie False</p>
Méthode <code>isnumeric()</code>	<p>Cette méthode renvoie True si le caractère est décimal, False sinon. Si x vaut "12", alors <code>x.isnumeric()</code> renvoie True. Si x vaut "12." alors <code>x.isnumeric()</code> renvoie False.</p>
Méthode <code>isalnum()</code>	<p>Cette méthode renvoie True si le caractère est alphabétique, False sinon. Si x vaut "abc", alors <code>x.isalnum()</code> renvoie True. Si x vaut "a5c" alors <code>x.isalnum()</code> renvoie True. Si x vaut "a5.c" alors <code>x.isalnum()</code> renvoie False.</p>