

jeu de devinette



Projet 4:

Un jeu de devinette consiste à générer un nombre x (Aléatoire entre 1 et 100) puis essayer de le deviner .



- Générer un entier x au hasard entre 1 et 100.
- Afficher un message contenant la question « Donner un entier entre 1 et 100 = ? » et demander de l'utilisateur une réponse.
- On veut afficher un message suivant la valeur entrée : « Bravo » si la réponse est juste et « plus grand » ou « plus petit » sinon .
- On veut **répéter** la saisie d'un nombre et l'affichage du message correspondant **jusqu'à** ce que l'utilisateur trouve la solution ($x=n$)
- Ajouter le nombre d'essai nb , et le jeu se termine si le joueur trouve le nombre caché et un message « Bravo » est affiché ou si le joueur fait 10 essais sans trouver le nombre et un message « Perdu ! » et affiché, ainsi que le nombre à trouver.



Objectif 1 : Générer un entier x au hasard entre 1 et 100.

Point d'information

Aléa Permet de générer un entier au hasard.

Algorithme	Python
$x \leftarrow \text{Aléa}(0,10)$	from random import * x=randint(0,10)

Remarque : Pour utiliser **randint** en Python, il faut commencer par importer la bibliothèque **random**

Point d'information

Pour affecter une valeur à une variable, on utilise le symbole d'affectation.

Algorithme	Python
$x \leftarrow 10$	x=10

Attention : ne pas confondre avec le test d'égalité

Algorithme	Python
Si $x=10$ alors	if $x==10$:



On va avoir la solution suivante :

Algorithme	python
Algorithme devinette Début $x \leftarrow \text{Aléa}(1,100)$	from random import * #ou from random import randint x=randint(1,100)



Objectif 2 : Afficher un message contenant la question « Donner un entier entre 1 et 100 = ? » et demander de l'utilisateur une réponse.

Point d'information

Pour afficher une variable ou un message :

Algorithme	Python
Ecrire("Message",x)	print('Message',x)

- En Python on peut délimiter un texte avec ' ' ou " "
- Exemple: print("C'est un texte")
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : print('___',end='')

Point d'information

Pour saisir une variable :

Algorithme	Python
lire(n)	n=input()

- En Python : saisir une chaîne (str) : n=input()
saisir un entier (int): n=int(input())
saisir un réel (float) : n=float(input())
- Pour lire une variable en affichant un message:
n=input('Message=')



On va avoir la solution suivante :



Algorithme	python™
Algorithme devinette Début $x \leftarrow \text{Aléa}(1,100)$ Ecrire("Donner un entier entre 1 et 100:"),Lire(n)	<pre>from random import randint x=randint(1,100) n=int(input('Donner un entier (entre 1-100) = '))</pre>



Objectif 3 : On veut afficher un message suivant la valeur entrée : « Bravo » si la réponse est juste et « plus grand » ou « plus petit » sinon .

Algorithme	python™
Algorithme multiplication Début $x \leftarrow \text{Aléa}(1,100)$ Ecrire("Donner un entier entre 1 et 100:") Lire(y) Si $x=y$ alors Ecrire("Bravo") Sinon si $y>x$ alors Ecrire("Plus petit") Sinon Ecrire("Plus grand") Fin	<pre>from random import * x=randint(1,100) y=int(input('Donner un entier (entre 1-100) = ')) if y==x : print('Bravo') elif y>x: print('plus petit !') else: print('Plus grand :')</pre>



Point d'information

Structure conditionnelle si

Algorithme	Python
Si condition alors Traitement1	if condition : ↔ traitement1
Sinon Traitement 2	else :
FinSi	↔ traitement2

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : `print('___',end=' ')`



Objectif 4 : On veut **répéter** la saisie d'un nombre et l'affichage du message correspondant **jusqu'à** ce que l'utilisateur trouve la solution ($x=n$) .

Il y a un traitement qui se répète, quelle est la boucle à choisir ?



Point d'information

La boucle **Pour**

Algorithme	Python
Pour i de 0 à n-1 faire Traitement FinPour	<pre>for i in range(n) : ↔ Traitement</pre>

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- `range(4)` donne 0 1 2 3 `range(1,4)` donne 1 2 3



Point d'information

La boucle **Répéter jusqu'à**

Algorithme	Python
Répéter Traitement Jusqu'à condition(s)	<pre>valide=False while valide==False: Traitement valide=(cond(s))</pre>

La boucle **Tant que faire**

Algorithme	Python
Tant que Cond(s) Faire Traitement Fin Tantque	<pre>while condition : Traitement</pre>

On va avoir la solution suivante :



Algorithme	python™
Algorithme nombre Début $X \leftarrow \text{Aléa}(1,100)$ Répéter Ecrire("Donner un entiere entre 1 et 100:"),Lire(y) Si $x=y$ alors Ecrire("Bravo") Sinon si $y>x$ alors Ecrire("Plus petit") Sinon Ecrire("Plus grand") Finsi Jusqu'à ($x=y$) Fin	<pre>from random import * x=randint(1,100) valide=False while valide==False: y=int(input('Donner un entier (entre 1-100) = ')) if y==x : print('Bravo') elif y>x: print('plus petit !') else: print('Plus grand :') valide= (x==y)</pre>

On pourra aussi implémenter l'algorithme en utilisant While :

```

python
from random import *
x=randint(1,100)
y=-1
while x !=y:
    y=int(input('Donner un entier (entre 1-100) = '))
    if y==x :
        print('Bravo')
    elif y>x:
        print('plus petit !')
    else:
        print('Plus grand :')

```



Objectif 5 : Ajouter le nombre d'essai nb, et le jeu se termine si le joueur trouve le nombre caché et un message « Bravo » est affiché ou si le joueur fait 10 essais sans trouver le nombre et un message « Perdu ! » et affiché, ainsi que le nombre à trouver.

On va avoir la solution suivante :



Algorithme	python™				
<p>Algorithme nombre</p> <p>Début</p> <p>X ← Aléa(1,100)</p> <p>Nb ← 0</p> <p>Répéter</p> <p>Nb ← nb+1</p> <p>Ecrire("Donner un entiere entre 1 et 100:"),Lire(y)</p> <p>Si x=y alors Ecrire("Bravo")</p> <p>Sinon si y>x alors Ecrire("Plus petit")</p> <p style="padding-left: 40px;">Sinon Ecrire("Plus grand")</p> <p>Finsi</p> <p>Jusqu'à (x=y) ou (nb=10)</p> <p>Si x≠y alors Ecrire("Perdu !, le nombre à trouver=", x)</p> <p>Finsi</p> <p>Fin Déclaration des objets</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>x, y, nb</td> <td>entier</td> </tr> </tbody> </table>	Objets	Types/Nature	x, y, nb	entier	<pre> from random import * x=randint(1,100) nb=0 valide=False while valide==False: nb=nb+1 y=int(input('Donner un entier (entre 1-100) = ')) if y==x : print('Bravo') elif y>x: print('plus petit !') else: print('Plus grand :') valide= (x==y)or (nb==10) if x!=y: print('perdu !, le nombre à trouver=',x) </pre>
Objets	Types/Nature				
x, y, nb	entier				