

La programmation

Pour faire un programme “Jeu” nous allons utiliser :

1) les conditions de type vrai/faux (booléen)

Un booléen est une variable qui peut avoir deux états possibles, généralement vrai ou faux, en langage Python True ou False. Les booléens sont très utiles pour des tests.

Pour faire les tests nous allons utiliser des **opérateurs de comparaisons**.

=	Égalité
!= (≠)	Inégalité
<	Strictement inférieur à
<= (≤)	Inférieur ou égal à
>	Strictement supérieur
>= (≥)	Supérieur ou égal à

Ces opérateurs ne semblent pas très exhaustifs. En effet, avec les opérateurs, nous ne pouvons faire que quelques tests basiques sur des nombres. Cependant, rappelez-vous : pour un ordinateur, tout n'est que nombre comme pour le stockage des données (ex : variables).

Exercice :

$\pi \approx 3,1415$ Vrai 1

$\pi \neq 3,1415$ Vrai 1

$\pi < 3,14$? Faux 0

$\pi \leq 3,141$? Faux 0

$\pi > 3,141$? Vrai 1

$\pi \geq 3,14$? Vrai 1

$\pi = 3,1415$ Faux 0

Rappel :

Code binaire naturel :

Codage décimal	Codage binaire naturel	
0	0	$0 \cdot 2^0$
1	1	$1 \cdot 2^0$
2	10	$1 \cdot 2^1 + 0 \cdot 2^0$
3	11	$1 \cdot 2^1 + 1 \cdot 2^0$
4	100	$1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$
5	101	$1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$
6	110	$1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$
7	111	$1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$
8	1000	$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$
9	1001	$_ \cdot 2^3 + _ \cdot 2^2 + _ \cdot 2^1 + _ \cdot 2^0$
10	1010	$_ \cdot 2^3 + _ \cdot 2^2 + _ \cdot 2^1 + _ \cdot 2^0$
87	101 0111	////////////////////////////////
108	110 1100	////////////////////////////////

	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	64	32	16	8	4	2	1
87	1	0	1	0	1	1	1
108	1	1	0	1	1	0	0
54	0	1	1	0	1	1	0
37	0	1	0	0	1	0	1
84	1	0	1	0	1	0	0

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	64	32	16	8	4	2	1
15	0	0	0	0	1	1	1	1
32	0	1	0	0	0	0	0	0
42	0	0	1	0	1	0	1	0
112		1	1	1	0	0	0	0
28		0	0	1	1	1	0	0
127	0	1	1	1	1	1	1	1
184	1	0	1	1	1	0	0	0

	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	512	256	128	64	32	16	8	4	2	1
768	1	1	0	0	0	0	0	0	0	0
184			1	0	1	1	1	0	0	0

2) la structure conditionnelle : Si...Alors...Sinon

SI "CONDITION" **ALORS** "ACTION A" **SINON** "ACTION B"

Le bloc **sinon** n'est pas obligatoire

If "CONDITION" **then** "ACTION A" **else** "ACTION B"

Exemple 1 :

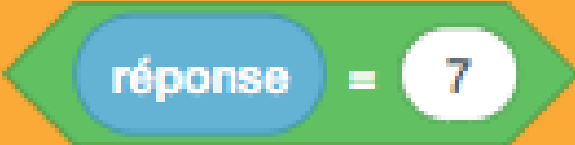
- Choisir un chiffre ?
- Si le chiffre est égal à 7 alors dire " Gagné :)" , sinon dire " Perdu :(" "

Choisir un chiffre ?

SI Chiffre = 7 ? **ALORS** Gagné :) **SINON** Perdu :(

quand  est cliqué

demander Choisir un chiffre ? et attendre

si  réponse = 7 alors

dire Gagné :)

sinon

dire Perdu :(

Si le chiffre choisi est 7 alors

Gagné :)



Si le chiffre choisi n'est pas 7 alors

Perdu :(



3) les variables informatiques

En informatique, les variables sont des éléments qui associent un nom (l'identifiant) à une valeur. La valeur peut être de nature différente : nombre, texte, etc. Les variables sont physiquement implantées dans la mémoire du système programmé (ordinateur, carte microprocesseur, etc.)

Une variable contient une valeur qui peut varier au cours de l'exécution du programme, comme la couleur des habits d'un personnage, le nombre d'activations d'un capteur, etc.

La création des variables :

On déclare les identifiants des variables. Le logiciel Scratch (mBlock, etc.) fournit la liste des opérations potentielles sur les variables (créer, mettre, ajouter, montrer, cacher).

Les variables sont des éléments qui **associent un nom (l'identifiant)** à une valeur, qui sera implantée dans la mémoire du système programmé.
Une variable contient une valeur qui peut **varier** au cours de l'exécution du programme.

Exercice :

Liste les 5 variables créées et utilisées pour faire le jeu CP.

-
- Temps du parcours
- Touches de direction
- Nombre d'essais
- Gagné
- Abscisse
-

Exemple 2 :

• Choisir un nombre non nul • Si le nombre est supérieur à 3 alors lui soustraire 3, sinon lui ajouter 5 • Ajouter 6 au résultat obtenu • Multiplier par 2 le résultat • Afficher le résultat	Choisir un nombre Si nombre > 3 ALORS (nombre -3) SINON (nombre +5) Résultat +6 Résultat *2 Afficher Résultat
On choisit comme nombre de départ 5 Alors : • Nombre >3 ? OUI alors • 5 - 3 = 2 • 2 + 6 = 8 • 8 × 2 = 16 Le résultat est 16	On choisit comme nombre de départ 1 Alors on fait : • Nombre > 3 ? NON alors • 1 + 5 = 6 • 6 + 6 = 12 • 12 × 2 = 24 Le résultat est 24

Exercice 01 :

On choisit comme nombre de départ 2

On choisit comme nombre de départ 8

On choisit comme nombre de départ 3

Alors :

- **Nombre** >3 ? --- alors
- --- - --- = ---
- --- + 6 = ---
- --- × 2 = ---

Le résultat est **28-**

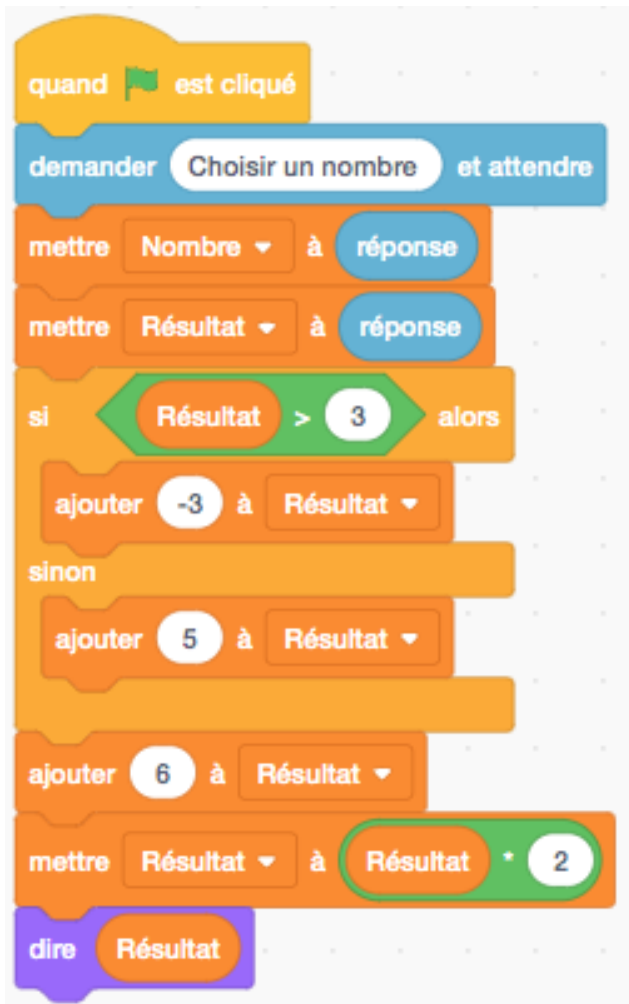
On choisit comme nombre de départ 10

Alors :

- **Nombre** >3 ? --- alors
- --- - 3 = ---
- --- + 6 = ---
- --- × 2 = ---

Le résultat est **26**

SCRATCH



On a créé 2 variables :

- Nombre
- Résultat