

# CONCEPTS DE PROGRAMMATION ET COMPÉTENCES DÉVELOPPÉS AVEC SCRATCH

Dans le processus de création d'histoires interactives, de jeux, et d'animations avec Scratch, les jeunes peuvent acquérir des compétences et apprendre d'importants concepts informatiques.


## CAPACITÉS À LA RÉOLUTION DE PROBLÈMES ET À LA CONCEPTION DE PROJETS

- raisonner logiquement
- rechercher des bugs
- développer des idées depuis la conception initiale jusqu'à la réalisation de projet complet
- favoriser la capacité de concentration et la persévérance

## IDÉES FONDAMENTALES SUR LES ORDINATEURS ET LA PROGRAMMATION

- Écrire un programme informatique impose de dire à l'ordinateur ce qu'il doit faire de façon précise et ceci pas à pas
- Écrire un programme informatique ne nécessite pas d'être un expert mais d'avoir une pensée claire et attentive

## CONCEPTS SPÉCIFIQUES DE PROGRAMMATION

Concept	Explication	Exemple
séquence	Pour créer un programme avec Scratch, vous devez systématiquement penser à l'ordre des étapes.	
itération (boucle)	<b>répéter indéfiniment</b> et <b>répéter</b> peuvent être utilisés pour une itération (répétition d'une série d'instructions)	
instructions conditionnelles	<b>si</b> et <b>si - sinon</b> permettent d'engager une action suivant qu'une condition est réalisée ou non.	
variables	Les blocs <b>variable</b> permettent de créer des variables et de les utiliser dans un programme. Les variables peuvent contenir des nombres ou des chaînes de caractères. Scratch permet d'utiliser aussi bien des variables globales que des variables spécifiques à un objet.	
listes (tableaux)	Les blocs <b>listes</b> permettent de stocker et d'avoir accès à une liste de nombres et de chaînes de caractères. Cette sorte de structure de données peut être considérée comme un tableau dynamique.	

# CONCEPTS DE PROGRAMMATION ET COMPÉTENCES DÉVELOPPÉS AVEC SCRATCH

Concept	Explication	Exemple
gestion d'évènements	L'utilisation du bloc <b>quand ... est pressé</b> ou un clic sur un lutin sont des exemples de gestion d'évènements déclenchés par l'utilisateur ou par une autre partie du programme.	
processus (exécution parallèle)	Lancer 2 piles en même temps crée 2 processus indépendants qui s'exécutent en parallèle.	
coordination et synchronisation	<b>envoyer à tous ...</b> et <b>quand je reçois ...</b> peuvent coordonner les actions de plusieurs lutins. En utilisant <b>envoie et attends</b> il est possible de synchroniser plusieurs actions.	<p>Par exemple, Lutin 1 envoie le message "gagnant" quand cette condition est remplie :</p> <p>Le script du Lutin 2 est activé quand le message "gagnant" est reçu :</p>
entrée au clavier	<b>demande ... et attends</b> invite l'utilisateur à saisir sa <b>réponse</b> . <b>réponse</b> enregistre l'entrée au clavier après sa validation.	
nombres au hasard	<b>nombre aléatoire entre ... et ...</b> choisit un nombre entier au hasard dans un intervalle donné.	
opérateur logique booléen	<b>et, ou, non</b> sont des exemples d'opérateurs logiques booléens.	
interaction dynamique	Pour une interaction dynamique en temps réel <b>souris x, souris y</b> , et <b>volume</b> peuvent être utilisés.	
conception d'une interface utilisateur	Dans Scratch, vous pouvez concevoir une interface utilisateur : par exemple, en déclenchant une action par un clic sur un lutin.	

## CONCEPTS DE PROGRAMMATION NON INTRODUITS ACTUELLEMENT DANS SCRATCH

- procédures et fonctions
- passage de paramètres et retour de valeurs
- récursion
- définition de classes d'objets
- héritage
- gestion d'exceptions
- fichier d'entrée sortie