

<b><u>COMPÉTENCES ET SAVOIR-FAIRE - PROCESSING</u></b>	😊	😐	😞
<b><u>REPRÉSENTATION BINAIRE ET HEXADÉCIMALE D'UN NOMBRE</u></b>			
<b><u>LA BASE BINAIRE :</u></b>			
Savoir ce qu'est un bit, un octet. Savoir jusqu'à quelle valeur en base 10 peut-on compter avec un octet.			
<b><u>LA BASE HEXADÉCIMALE:</u></b>			
Savoir décrire la base hexadécimale et énoncer les différents symboles qui la composent.			
Savoir que deux symboles hexadécimaux consécutifs dans une représentation de position sont nécessaires pour coder un octet. Savoir le démontrer par un petit calcul.			
<b><u>IMAGE NUMÉRIQUE</u></b>			
<b><u>IMAGE EN NIVEAU DE GRIS:</u></b>			
Savoir qu'une image en niveau de gris se représente numériquement par un tableau à deux entrées dont chaque case est un pixel auquel on attribue un octet en place mémoire, ce qui représente 256 niveaux de gris entre le noir et le blanc.			
<b><u>IMAGE EN COULEUR:</u></b>			
Savoir qu'une image en couleur représente numériquement par un tableau à trois entrées, ou une superposition de trois tableaux à deux entrées, représentant les 3 couleurs R,V,B. Chaque pixel est associé à 3 octets.			
Savoir lire les valeurs RVB, HSV, et la représentation hexadécimale d'une teinte donnée.			
<b><u>VARIABLES ET TYPES :</u></b>			
Connaître et définir la notion de variable en programmation. Connaître les principaux types primitifs et types complexes sous Processing.			

<b><u>ARCHITECTURE GÉNÉRALE D'UN PROGRAMME SOUS PROCESSING :</u></b>			
Connaître la structure générale d'un programme à toujours respecter lorsqu'on réalise un sketch sous Processing (bibliothèques, variables globales, setup(), draw()),			

classes et méthodes personnelles...)			
Savoir expliquer ce qu'est une variable globale.			
Connaître les fonctions des méthodes <i>Setup()</i> et <i>Draw()</i> ainsi que leur fonctionnement.			
<b><u>LES BOUCLES ET LES TESTS CONDITIONNELS :</u></b>			
Savoir décrire les différentes étapes et le fonctionnement d'une boucle simple, d'une boucle double et du test conditionnel <i>if</i> .			
Savoir utiliser une boucle simple, d'une boucle double et du test conditionnel <i>if</i> .			
<b><u>LA PROGRAMMATION ORIENTÉ OBJET : LES CLASSES</u></b>			
<b><u>COMPRENDRE UNE CLASSE D'OBJET :</u></b>			
Savoir décrire l'architecture générale d'une classe.			
Savoir décrire l'utilité et le fonctionnement du constructeur d'une classe			
Savoir comment on instancie un objet issu d'une classe.			
Savoir ce qu'est le polymorphisme et en donner des exemples dans Processing.			
Savoir décrire une classe d'objet en donnant ses attributs, ses méthodes et son constructeur. Savoir reconnaître une classe polymorphe.			
Savoir construire une classe donnée à partir d'un cahier des charges (c'est-à-dire la définition des attributs, et des différentes fonctionnalités des méthodes).			
Savoir faire évoluer cette classe.			
<b><u>UTILISER UNE CLASSE EXISTANTE ISSUE D'UNE LIBRAIRIE</u></b>			
Savoir décrire l'architecture générale d'une classe issue d'une librairie, c'est à dire, savoir reconnaître:			
<ul style="list-style-type: none"> <li>▪ le/les constructeur/s (<i>constructor</i>)</li> </ul>			
<ul style="list-style-type: none"> <li>▪ les attributs (<i>parameters</i>).</li> <li>▪ les méthodes (<i>methods</i>).</li> </ul>			
Savoir instancier une classe issue d'une librairie en utilisant le bon constructeur.			

Savoir reconnaître une classe polymorphe.			
Savoir utiliser les différents attributs ( <i>parameters</i> ) et méthodes ( <i>methods</i> ) d'une classe issue d'une librairie, via un objet préalablement instancié.			
Savoir trouver et lire la documentation d'une classe issue d'une librairie, c'est-à-dire, savoir trouver et comprendre la description des attributs, des méthodes et du constructeur.			