

# Code-moji

Par: Cassandra Lenters

Duration: 1 heure

LEÇONS	SUJETS	PROVINCES / TERRITORIES	PROGRAMME
7 <sup>e</sup> et 8 <sup>e</sup> année	Arts plastiques, mathématiques	À l'échelle du Canada	Processing

## Résumé

Créez des emojis avec le code. Dans le cadre de cette activité, les apprenants remixeront une création Processing pour créer leurs propres emojis interactifs.

## Préparation

- Créez un compte gratuit pour la classe à l'adresse suivante : [openprocessing.org/home/join](https://openprocessing.org/home/join).
- Consultez l'exemple de projet : <https://www.openprocessing.org/sketch/559871>.
- Imprimez le corrigé de l'activité principale : <http://bit.ly/codemoji-corrige>
- Cette leçon nécessite des ordinateurs et une connexion à Internet.

## Concepts de programmation

- ✓ Algorithmes
- ✓ Expressions
- ✓ Conditionnelles
- ✓ Fonctions
- ✓ Séquences

## Liens avec le programme de formation

Couleur, modèles de couleurs (RVB), théorie des couleurs, forme, symboles, alignement, mesure, aire, graphiques, coordonnées x,y, géométrie, formes 2D

## Leçon

### Introduction

Visionnez cette vidéo sur [l'histoire des emojis](#) (environ 5 minutes).

Réflexion rapide : « Quels sont les éléments qui t'ont surpris? » « Qu'as-tu appris de nouveau? »

Nous allons créer des emojis interactifs avec un langage de programmation appelé Processing.

### Programmation en groupe

- Demandez aux apprenants d'aller à l'adresse [openprocessing.org](http://openprocessing.org) et de se connecter en utilisant le compte que vous avez créé pour la classe.
- Dites aux apprenants d'aller voir le projet « Getting Started » dans OpenProcessing : <http://bit.ly/processing-code-along>
- Demandez-leur de créer un projet dérivé (« Create a fork »), puis de modifier le projet créé (« Edit ») et de le renommer (voir l'étape 1).
- Faites-leur remarquer ces aspects du projet « Getting Started » :
  1. **Void setup** : Cette fonction met en place notre canevas. Il s'agit du premier événement qui a lieu lorsque nous exécutons notre projet. Il n'a lieu qu'une fois.
  2. **Void draw** : L'emplacement où nous mettons notre code pour dessiner et créer des animations. Cette fonction est exécutée après la première, mais elle s'exécute en boucle dans le même ordre (la même séquence) que dans le code.
  3. **Points-virgules** : Nous devons les mettre à la fin de chaque ligne de code à l'intérieur des fonctions void et draw.
  4. **Coordonnées X et Y** : Veuillez noter que notre canevas utilise des coordonnées x,y. Toutefois, (0,0) n'est pas à l'endroit que l'on pense. Plutôt que d'être dans le milieu, (0,0) se trouve en haut à gauche. Lorsque la valeur de X augmente, le point se déplace vers la droite. Lorsque la valeur du Y augmente, le point se déplace vers le bas.
  5. **Commentaires**: Le texte avec des barres obliques // représente des commentaires. Nous les voyons, mais ils ne font pas partie du projet.
- Présentez le groupe à leur nouvelle amie : la référence de Processing! Ouvrez-la dans un autre onglet : <https://processing.org/reference/>.
- Faites deux ou trois exercices :
  - Faites bouger le cercle vers le milieu.
  - Dessinez un rectangle derrière le cercle (faites une montre!).
  - Changez la couleur de l'arrière-plan (indice : cet événement ne doit qu'arriver une fois).

- Jetez un coup d'œil à cet exemple :  
<https://www.openprocessing.org/sketch/561200>

## Activité

Montrez **l'exemple de projet** aux apprenants. Nous allons créer des emojis avec le code.

Demandez aux apprenants d'ouvrir le **projet de départ** (<http://bit.ly/emoji-starter>) et de cliquer sur l'icône « FORK » (la fourchette) pour créer un projet dérivé (« Create a fork »).

Utilisez le **corrigé** pour réaliser les étapes suivantes avec les apprenants :

- Associer des formes aux coordonnées.
- Dessiner d'autres formes.
- Ajouter de l'interactivité.
- Ajouter de la couleur avec des valeurs RVB (RGB).

Accordez-leur une période pour qu'ils puissent travailler sur leur projet. Si le temps vous le permet, faites l'exercice supplémentaire « Personnaliser son emoji ».

## Évaluation

### Objectifs d'apprentissage

**Je peux** créer des algorithmes dans Processing.

**Je peux** dessiner des formes à l'aide des coordonnées x,y.

**Je peux** utiliser une séquence pour superposer des formes sur mon canevas.

**Je peux** utiliser des expressions conditionnelles pour ajouter de l'interactivité à mon projet.

**Je peux** mélanger des couleurs (numériques) à l'aide de valeurs RVB.

### Critères de réussite

J'ai remplacé le titre de ma copie du projet par mon prénom.

J'ai mis toutes les formes de départ sur le visage de mon emoji.

J'ai ajouté au moins une nouvelle forme à mon dessin.

Mon projet est interactif; l'utilisateur peut faire une action en cliquant, en appuyant ou en déplaçant quelque chose.

J'ai changé la couleur d'au moins un élément dans mon dessin.

### Méthodes d'évaluation

Demandez aux apprenants d'écrire un énoncé d'artiste expliquant leurs décisions de conception qu'ils ajouteront comme commentaire dans le code de leur dessin. « Cette œuvre s'appelle [nom de l'œuvre]. C'est un [style artistique] créé avec [matériel]. J'ai décidé de [décision de conception] parce que [raison]. » Par exemple : « Cette œuvre s'appelle Amusant. C'est un projet d'art génératif créé avec Processing. J'ai décidé de mettre l'arrière-plan bleu parce que le bleu est une couleur complémentaire à l'orange. »

Demandez aux apprenants d'ajouter des commentaires à leur code chaque fois qu'ils modifient le projet de départ. Ainsi, ils pourront comprendre ce que signifie chaque ligne de code.

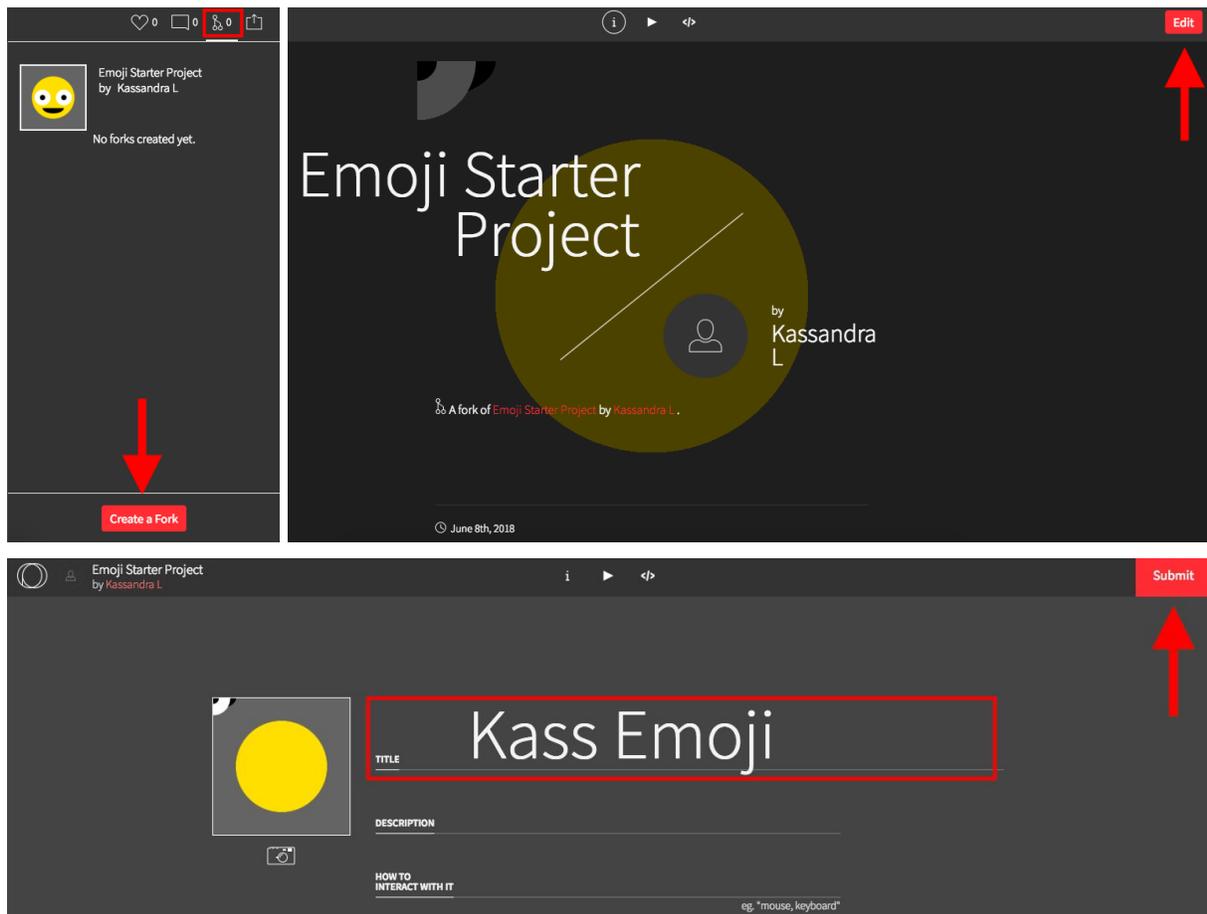
## Prolongements

- Visionnez la vidéo [Step into the Page](#) sur Glen Keane, animateur chez Disney. Exercez-vous à dessiner des émotions de la même façon que Glen Keane et son père (01:30) en traçant des cercles sur une feuille de papier en écrivant une liste d'expressions sous chaque dessin. Demandez aux apprenants de choisir l'un de ces visages et de s'en inspirer pour leur emoji :
- Faites un lien avec une leçon de maths sur l'aire. Les apprenants noteront l'aire de chaque cercle et rectangle de leur dessin en pixels. Lancez-leur le défi d'additionner l'aire totale de tous les cercles et les rectangles de leur dessin ou de calculer l'aire du dessin Processing d'un autre partenaire.
- Voyez des notions sur la théorie des couleurs. Discutez de RVB, qui est un modèle de synthèse additive de couleurs. Utilisez Adobe Colour pour apprendre à propos des agencements de couleur ou discutez des émotions associées aux couleurs.
- Profitez de l'occasion pour lancer une discussion sur l'utilisation des emojis à l'école (p. ex. ne pas utiliser des emojis dans des travaux écrits, mais les emojis sont permis dans un document de remue-méninges dans Google Docs).

# Code-moji

## ÉTAPE 1 : Ouvrir le projet initial

1. Allez sur le site [openprocessing.org](http://openprocessing.org) et connectez-vous.
2. Ouvrez le projet de départ : <http://bit.ly/emoji-starter>.
3. Faites un projet dérivé (Create a fork) et renommez le projet (icône « i »).



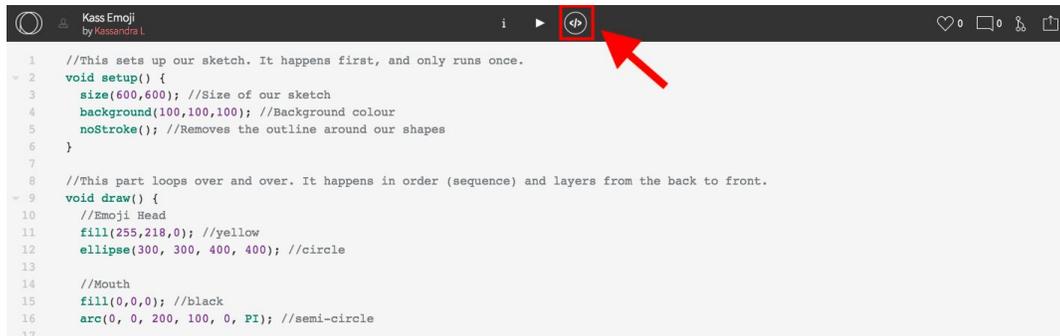
## ÉTAPE 2 : Associer des coordonnées aux formes

1. Cliquez sur l'icône de code `</>`.
2. Regardez le code déjà présent dans le projet. Il y a une tête, un œil (un globe oculaire et une pupille) et une bouche. Nous avons déjà quelques éléments de notre emoji, mais nous devons les mettre au bon endroit.
3. Voyez les notions sur les coordonnées x,y avec l'activité hors ligne suivante : <http://teacherslearningcode.com/en/lessons/stretch-and-scuttle>

Corrigé créé pour [Canada en programmation](http://Canada en programmation)  
 À propos de Processing : <https://processing.org>

4. Déplacez la bouche : changez les valeurs x,y de l'arc.
5. Déplacez l'œil : changez les valeurs x,y de l'ellipse.
6. Déplacez la pupille (sur le globe oculaire blanc) : changez les valeurs x,y de l'ellipse.
7. N'oubliez pas d'enregistrer votre travail (en haut à droite).

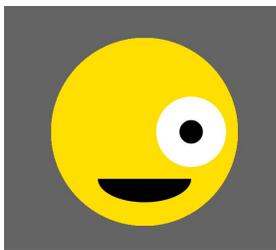
**Remarque :** Après avoir modifié le code, démontrez aux apprenants l'importance de la mise à l'essai du code en cliquant constamment l'icône de code et l'icône d'exécution (Run).



```

1 //This sets up our sketch. It happens first, and only runs once.
2 void setup() {
3   size(600,600); //Size of our sketch
4   background(100,100,100); //Background colour
5   noStroke(); //Removes the outline around our shapes
6 }
7
8 //This part loops over and over. It happens in order (sequence) and layers from the back to front.
9 void draw() {
10  //Emoji Head
11  fill(255,218,0); //yellow
12  ellipse(300, 300, 400, 400); //circle
13
14  //Mouth
15  fill(0,0,0); //black
16  arc(0, 0, 200, 100, 0, PI); //semi-circle
17

```



```

14  //Mouth
15  fill(0,0,0); //black
16  arc(300, 400, 200, 100, 0, PI); //semi-circle
17
18  //Right Eye
19  fill(255,255,255); //white
20  ellipse(400, 300, 150, 150); //circle
21
22  //Right Pupil
23  fill(0,0,0); //black
24  ellipse(400, 300, 50, 50); //circle
25

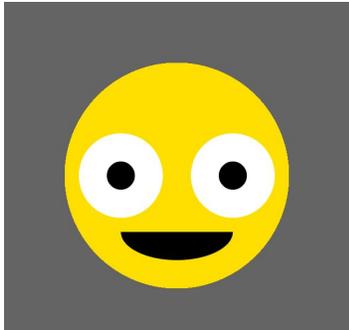
```

### ÉTAPE 3 : Dessiner d'autres formes

Complétons notre emoji. Nous avons le choix entre a) copier le code de l'œil et le déplacer ou b) dessiner une nouvelle forme à l'aide de la référence.

(A) Pour copier le code :

1. Sélectionnez le code de l'œil et de la pupille, puis copiez (ctrl+c) et collez (ctrl+v) le code à l'intérieur de « draw ».
2. Changez les valeurs du x afin qu'il y ait deux yeux parallèles.



```

26 //Left Eye
27 fill(255,255,255); //white
28 ellipse(200, 300, 150, 150); //circle
29
30 //Left Pupil
31 fill(0,0,0); //black
32 ellipse(200, 300, 50, 50); //circle
33

```

(B) Pour dessiner une nouvelle forme :

1. Trouvez une nouvelle forme à dessiner à l'aide de la référence. Essayez d'utiliser l'arc pour dessiner un emoji qui fait un clin d'œil.
2. Tapez ou collez le code de l'arc à l'intérieur de « draw ».
3. Entrez les valeurs de la forme. Pour le clin d'œil, vous devrez commencer à PI et finir à TWO\_PI (plutôt que de commencer à 0 et de finir à PI comme pour la bouche).
4. N'oubliez pas d'enregistrer votre travail.



```

29 //Left eye - winking
30 //also black - don't need to update fill()
31 arc(200, 300, 100, 50, PI, TWO_PI); //semi-circle
32

```

## ÉTAPE 4 : Ajouter de l'interactivité

Pour rendre notre projet **interactif**, nous devons permettre à l'utilisateur de modifier le dessin et d'interagir avec ce dernier. Par exemple, créons du code qui effectuera une action SI l'utilisateur appuie sur une touche sur le clavier.

**Remarque :** SI (IF) est une expression conditionnelle. Nous vérifions SI quelque chose est vrai. Si cette chose est vraie, le programme effectuera une action.

1. Ouvrez la page **keyPressed** de la référence pour en comprendre le fonctionnement (sans les parenthèses). <https://processing.org/reference/keyPressed.html>
2. Nous avons besoin de taper ou de copier l'expression IF (voir ci-dessous) à l'intérieur de la fonction « draw ».
3. Ajoutez quelque chose à l'intérieur de l'expression conditionnelle. Cette action (dessin) n'arrivera que SI une touche est pressée. Dans ce cas-ci, nous dessinerons une langue.
4. N'oubliez pas d'enregistrer votre travail.

```

Name      keyPressed

Examples  // Click on the image to give it focus,
          // and then press any key.

          // Note: The rectangle in this example may
          // flicker as the operating system may
          // register a long key press as a repetition
          // of key presses.

void draw() {
  if (keyPressed == true) {
    fill(0);
  } else {
    fill(255);
  }
  rect(25, 25, 50, 50);
}

```



```

30      //Sticking out tongue - IF a key is pressed
31      if(keyPressed == true){
32          fill(255,78,83); //red
33          arc(300, 400, 100, 150, 0, PI); //semi-circle
34      }

```

## ÉTAPE 5 : Ajouter de la couleur à l'aide des valeurs RGB

1. Ouvrez la page sur les arrière-plans dans la référence pour en comprendre le fonctionnement. Remarque : Les trois valeurs dans les parenthèses sont RVB (rouge, vert, bleu).
2. Demandez aux apprenants d'aller sur [color.adobe.com](https://color.adobe.com).
3. Sélectionnez une couleur d'arrière-plan.
4. Modifiez les valeurs RVB de l'arrière-plan. Par exemple, l'image ci-dessous affiche les valeurs du bleu.
5. Faites la même chose avec les valeurs de remplissage fill() pour changer la couleur de l'emoji.
6. N'oubliez pas d'enregistrer votre travail.



```
4 void setup() {  
5   size(600,600); //Size of our sketch  
6   background(126,157,255); //Background colour  
7   noStroke(); //Removes the outline around our shapes  
8 }
```

## BONUS : Personnaliser son emoji

1. Accordez du temps aux apprenants afin qu'ils puissent personnaliser leur emoji. Donnez-leur deux options : programmer l'un des emojis ci-dessous ou programmer un emoji avec une autre expression.
2. Remarque : Certains emojis sont plus difficiles à programmer que d'autres. Rien n'est impossible à programmer, mais il faut s'interroger sur la démarche. Pour trouver une solution, il suffit de consulter la référence, de travailler avec ses collègues et de chercher des informations sur Google.



Inquiet



Neutre



Lever les yeux



Content / rougir