



JavaScript

L'ARTISTE EN JAVASCRIPT

15/10/2016

Objectifs


- Apprendre à programmer en Javascript en s'amusant!
- Comprendre la structure d'un programme.
- Pour être plus confiant afin d'apprendre seul des choses plus complexes en JavaScript.
- Exploiter votre créativité.
- Revoir un peu les mathématiques!
- Pour vous encourager à faire une présentation.

C'est quoi Javascript?



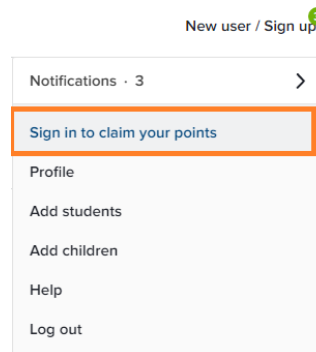
- Une langue que tous les navigateurs comprennent.
- Comme toutes autres langues (ex. français, anglais, néerlandais) on doit respecter des règles pour la grammaire / syntaxe en écrivant un programme.
- Sans Javascript, les pages ne seraient pas aussi cool! Facebook, Gmail, Twitter utilisent tous abondamment JavaScript pour rendre leur site plus facile à utiliser.

Qu'est qu'on peut faire avec Javascript?

- Jouer de la musique
- Interaction sur des pages web (exemple zoomer sur une photo, glisser un email dans la poubelle etc..)
- Dessiner 
- Développer des jeux
- Et beaucoup beaucoup d'autres choses!
- Regardez par vous même:
 - <http://lights.helloenjoy.com/>
 - <http://www.patatap.com/>
 - <https://www.cubeslam.com/>

Comment écrire un programme?

- Nous allons utiliser un éditeur en ligne très sympa.
- Naviguez vers <https://www.khanacademy.org/computer-programming/new/pjs>
- Loggez-vous en utilisant votre compte Gmail ou Facebook, pour sauvegarder vos programmes plutard.



L'éditeur

< COMPUTER PROGRAMMING

New Program [Edit Title](#)

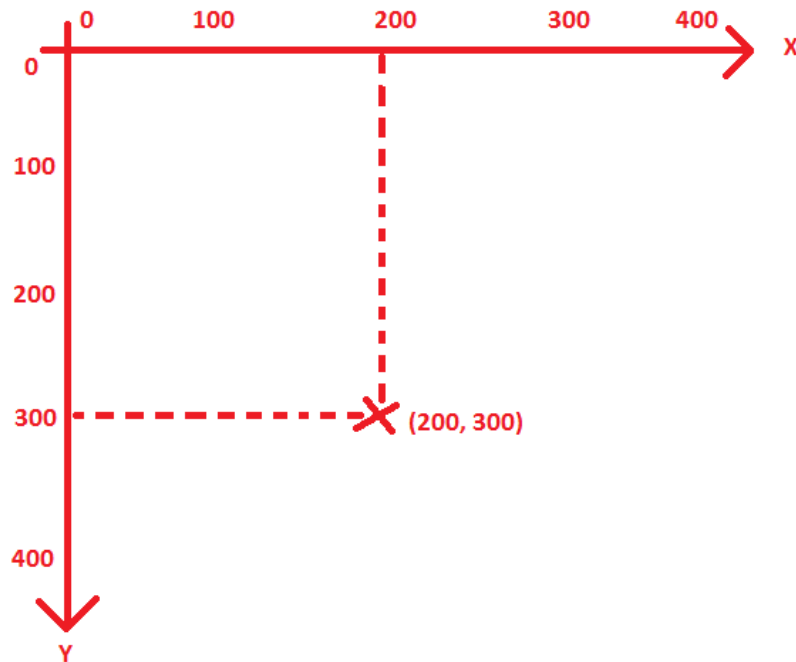
```
1 line(50, 50, 350, 350);  
2 |
```

Éditeur pour
écrire notre
programme



Save

~~Canvas~~



Le Canvas

- L'endroit pour nos dessins
- Taille : 400 x 400
- Suit le (x, y) système de coordonnée
- X – de gauche à droite
- Y – de haut en bas
- Comme exercice, pouvez-vous identifier:
 - (300, 100)
 - (50, 400)
 - (0,0)
 - (400,400)
 - (399, 17)

Notre tout premier programme

line(50, 50, 350, 350);



termine par un point virgule



fonction *Paramètres*

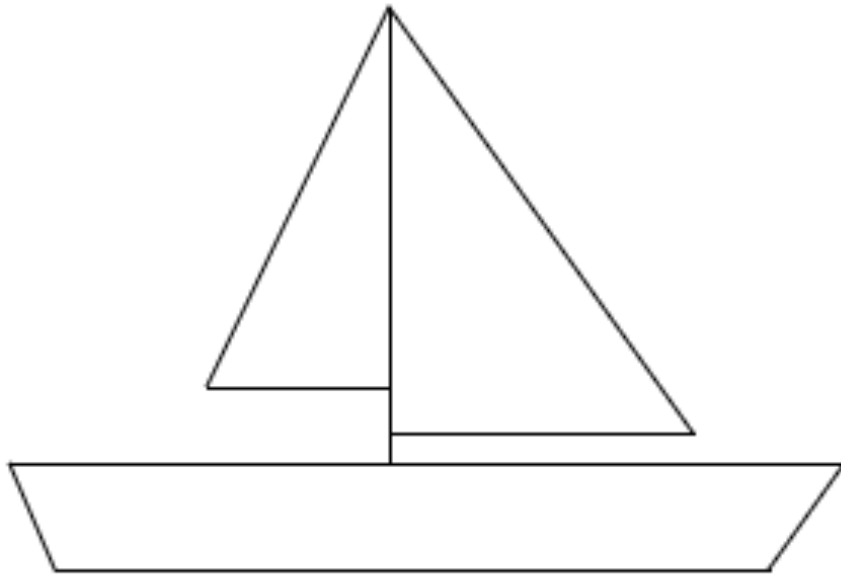
- Qu'est-ce-que vous voyez sur le canvas?
- Changez les paramètres 50, 350 etc.
- Pouvez-vous dessiner un carré?

line(100, 100, 300, 100);

line(300, 100, 300, 300);

line(100, 100, 100, 300);

line(100, 300, 300, 300);

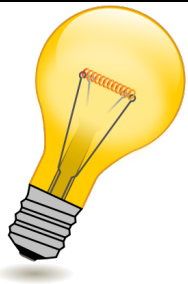


Un petit exercice

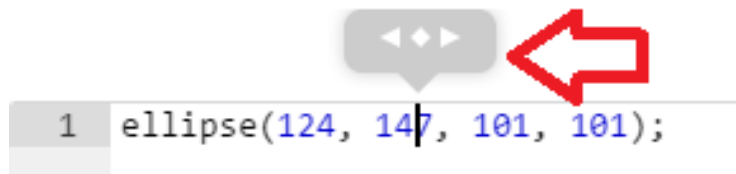
- Pouvez-vous dessiner ce bateau avec ce que vous avez appris jusqu'à maintenant?
- N'oubliez pas de sauvegarder votre programme!

Des cercles..

- `ellipse(120, 150, 100, 150);`
- Cette commande dessine un ellipse.
- En changeant les paramètres, pouvez-vous dessiner un cercle?



En cliquant sur les paramètres, un pointeur est affiché.
C'est pour vous faciliter à choisir correctement la valeur de votre paramètre. C'est plus facile à visualiser les changements que vous voulez faire.



Rectangles

- `rect(100, 100, 50, 50);`



X



Y



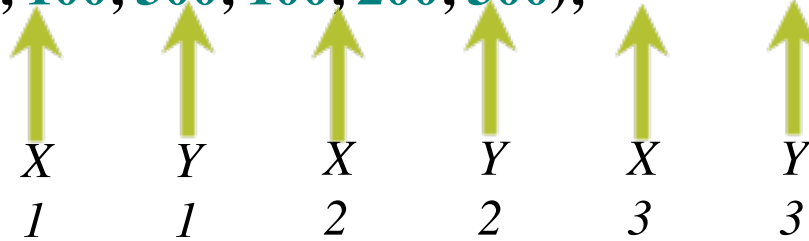
Largeur



Hauteur

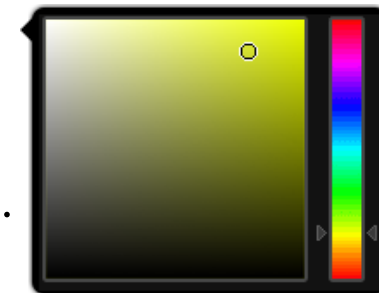
Triangle

- Un triangle a 3 points
- **triangle(100, 100, 300, 100, 200, 300);**



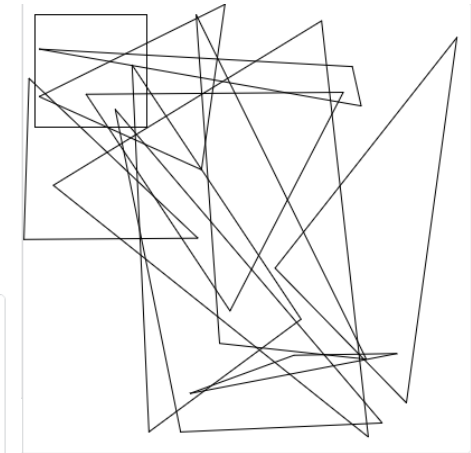
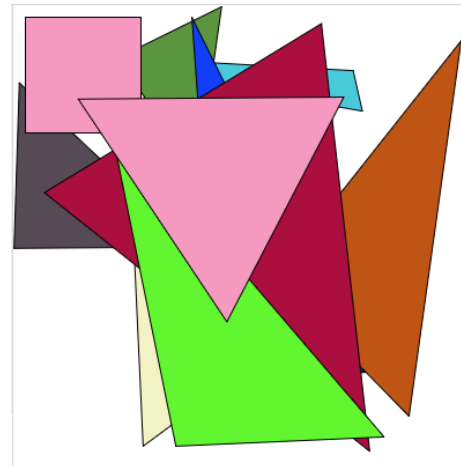
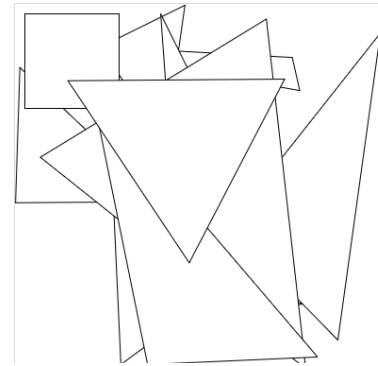
Des couleurs

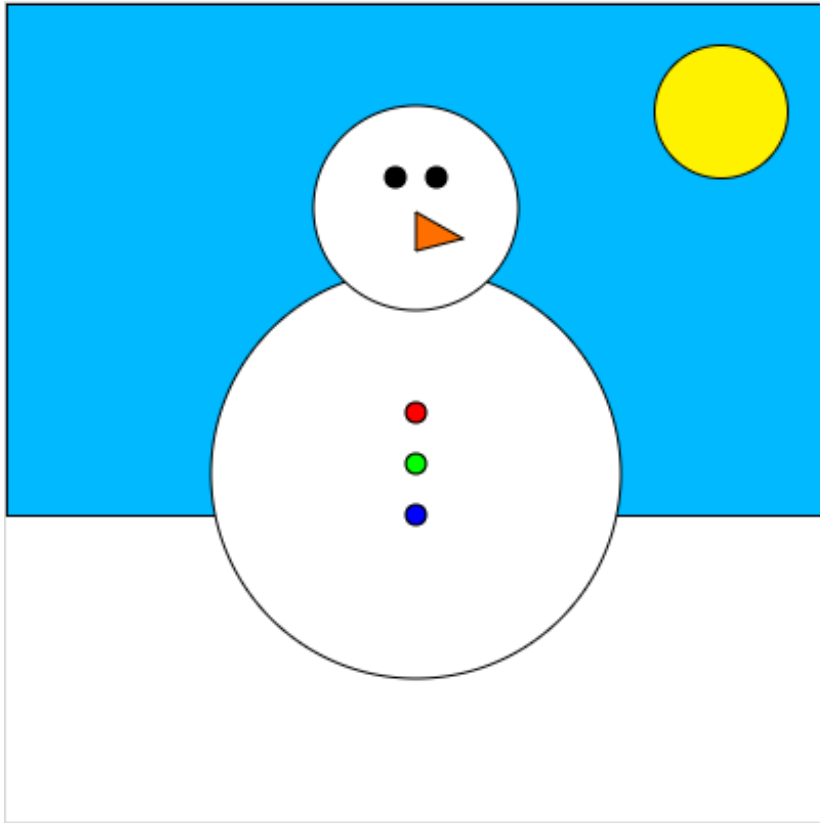
- Jusqu'à maintenant, on a utilisé des lignes noires pour nos dessins!
- En JS, vous pouvez mélanger le ROUGE, VERT et BLEUE pour faire d'autres couleurs. ROUGE, VERT et BLEUE = RGB.
- Chaque quantité de rouge, bleue et vert peuvent être spécifiés par un chiffre entre 0 et 255.
- Un rectangle rouge
`stroke(255, 0, 0);`
`rect(10, 10, 100, 100);` ← $R = 255, G = 0, B = 0$
- Vous pouvez aussi choisir la couleur par le menu qui s'affiche.



Des couleurs

- Par défaut, les figures géométriques sont replis en blanc. C'est pour cela quand vous placez plusieurs figures un sur l'autre, vous voyez partiellement ceux qui sont derrière.
- Pour les rendre transparent, avant de dessiner une figure géométrique, utilisez la commande `noFill()`;
- Pour remplir une figure avec une couleur spécifique, utilisez la commande `fill(r, g, b)`;
r, g, b prendront bien sûr les valeurs de 0 à 255.





Un petit exercice

- Pouvez-vous dessiner le bonhomme de neige sous le soleil?
- Utilisez ce que vous avez appris jusqu'à maintenant.
- Soyez créatifs..

Générer un nombre aléatoire et les variables

- Pour générer un nombre aléatoire, utilisez cette commande:

```
var x = random(0, 400);
```



- Variable le plus petit et le plus grand pour stocker des valeurs. On peut les réutiliser plus tard et même les modifier.
- Ici, le nom du variable c'est x, qui stockera une valeur aléatoire de 0 à 400.
- La variable x peut avoir des valeurs comme 0, 1, 1.5, 114.56, 200.99, 400, 399.99 etc..
- Nommez différemment les variables que vous utilisez: y, z, myVariables, toto, .. c'est mieux que le nom commence par une lettre de l'alphabet.
- Pouvez-vous générer un rectangle rempli d'une couleur aléatoire?

Décision avec if.. Else if

- La bloque « if » nous permet d'inclure des décision dans notre programme. C'est similaire quand on décide qu'est-ce-qu'on va porter s'il pleut ou s'il y a du soleil.
- Ce petit programme génère un nombre aléatoire entre 0 et 100 et affiche où le nombre se situe.

```
var x = random(0, 100);  
fill(255, 0, 0);  
text(x, 5, 10);  
if (x < 10) {  
    text(" is less than 10!", 50, 10);  
}  
else if (x >= 10 && x <= 50) {  
    text(" is between 10 and 50!", 50, 10);  
}  
else {  
    text(" is greater than 50!", 50, 10);  
}
```

Des boucles

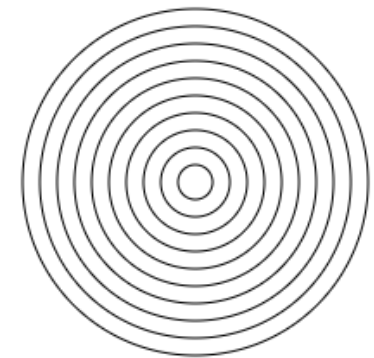
- On utilise des boucles pour exécuter plusieurs fois un morceau de code de notre programme.
- Générez 10 cercles.

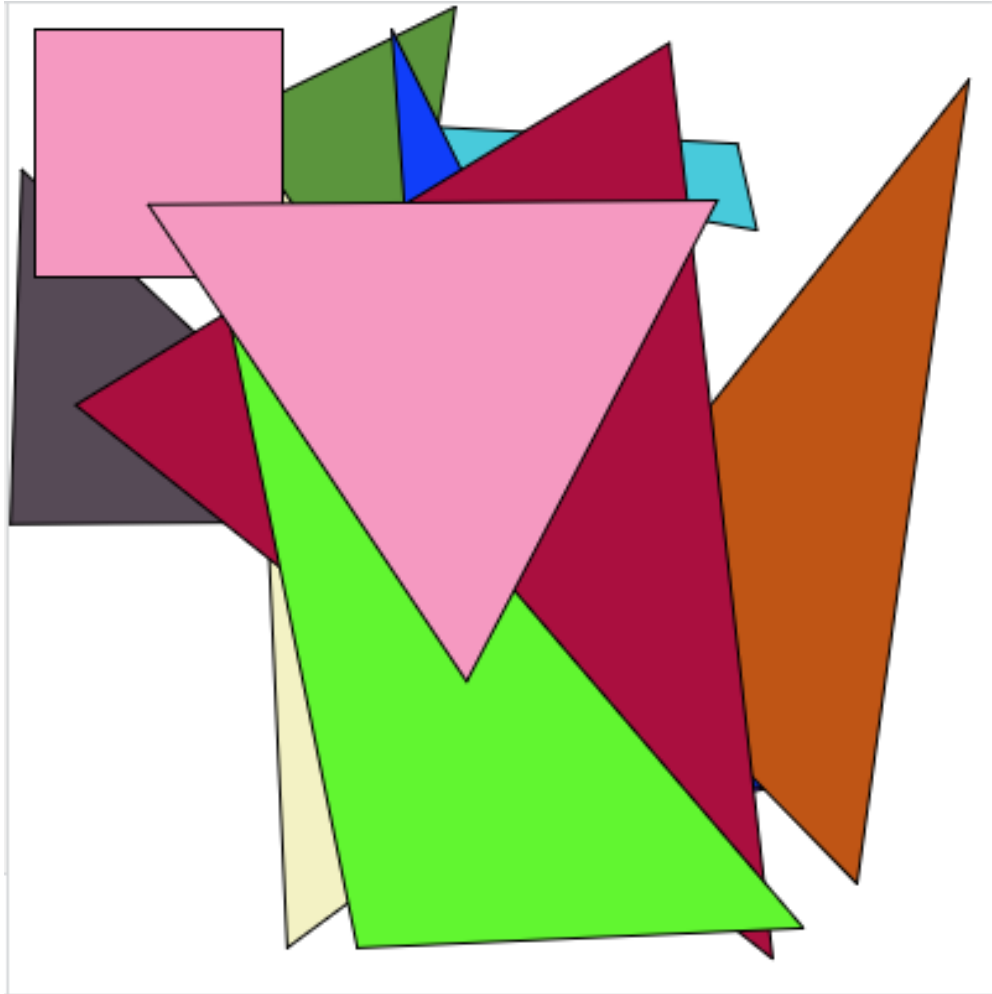
```
noFill();  
var i = 1;  
while (i <= 10) {  
  ellipse(200, 200, i*20, i*20);  
  i = i + 1;  
}
```

Initialiser la variable $i = 1$

Tant que i est moins que ou égale à 10:

- augmentez la valeur de i par 1
- Dessinez un cercle





Un petit exercice

- Générez 10 figures aléatoire et remplissez les avec des couleurs aléatoires.
- Placez les au hasard sur le canvas.

Images

- On peut aussi inclure des images sur le canvas.
- On a à notre disposition une librairie d'images sympas.
- Cette commande dessine une image sur le canvas..

```
var mrPink = getImage("avatars/mr-pink");
```

```
image(mrPink, 10, 1
```



Animations basiques

- Une fonction spéciale « draw »

```
draw = function() {
```

```
•
```

```
•
```

```
};
```

- Cette fonction est appelée 30 fois par seconde.
- Nous allons exploiter le fait qu'elle est appelée plusieurs fois par seconde pour créer des animations.
- C'est comme si tout le canvas est effacé et les figures sont dessinées de nouveaux 30 fois par secondes.

Animation en utilisant la fonction “draw”

```
var x = 0;

var dx = 10;

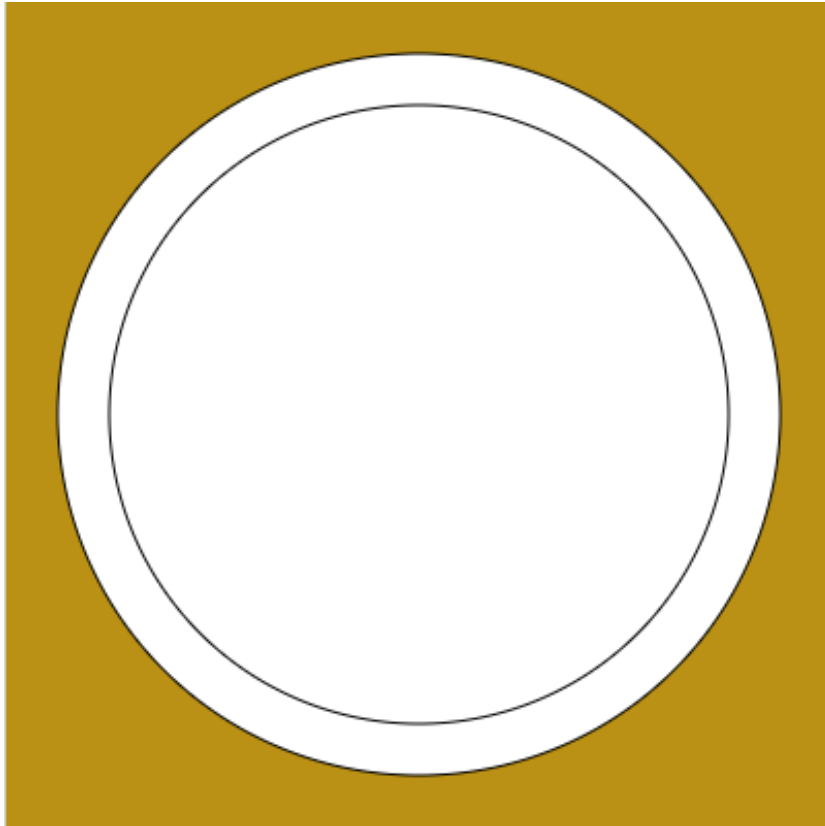
draw = function() {

  background(255, 255, 255); // le fond est de couleur blanche
  fill(255, 0, 0);
  ellipse(x, 200, 10, 10);
  x = x + dx;
  if (x >= 400) {
    dx = -10;
  }
  else if (x < 0) {
    dx = 10;
  }
};
```

Ce code dessine une petite balle rouge qui bouge sur le canvas.

- 1) Essayez d’ajouter 2 ou 3 balles qui bougent plus rapidement.
- 2) Dessinez une batte sur le canvas et quand la balle la touche, elle rebondit

CHALLENGES



Challenge #1

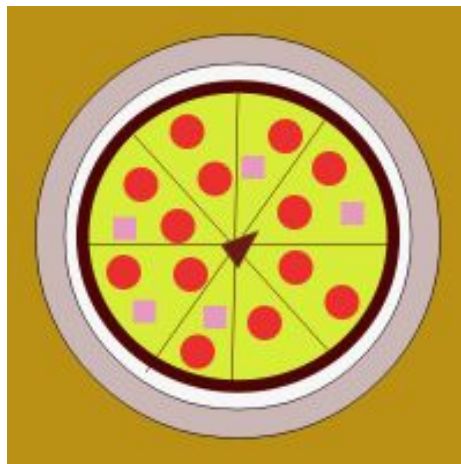
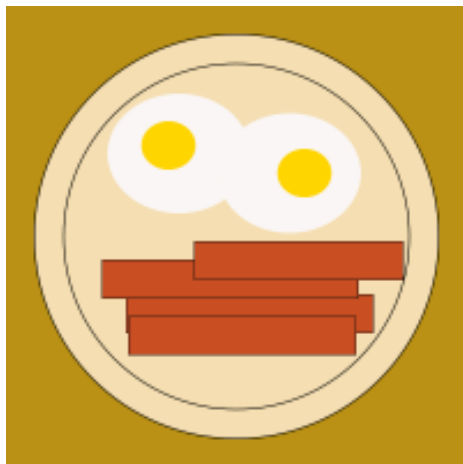
- À gauche, c'est ton plat sur une table en bois. Qu'est-ce-que tu mange pour le dîner?
- Utilisez les figures que vous avez apprises pour décorer le plat.
- Pour commencer, voici le programme:

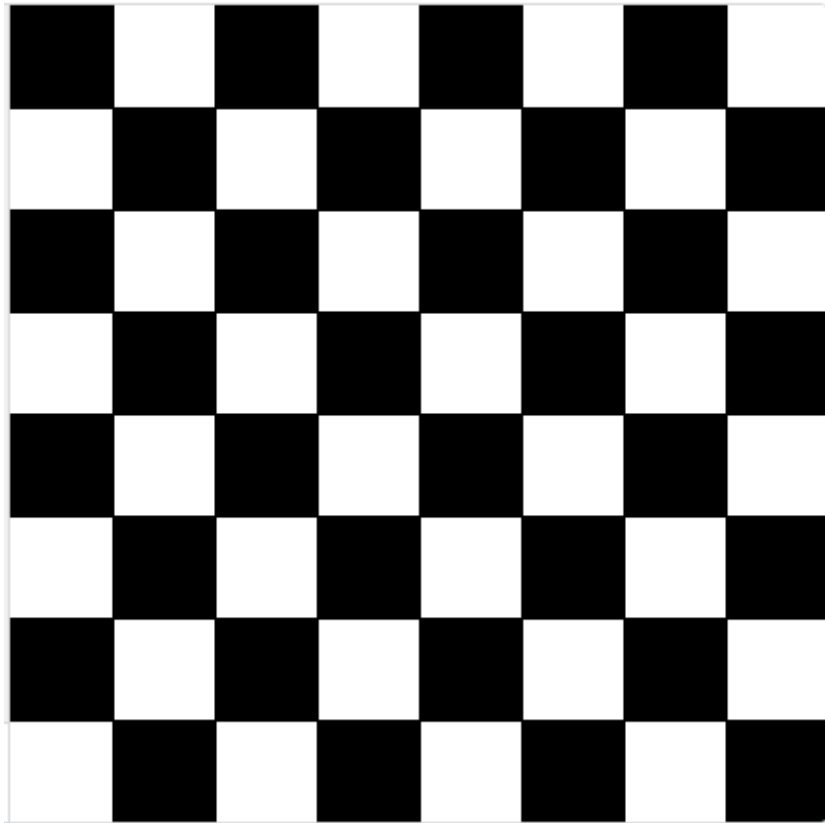
```
background(186, 145, 20); // table en bois
```

```
ellipse(200, 200, 350, 350); // plat
```

```
ellipse(200, 200, 300, 300);
```


Quelques exemples pour le challenge #1





Challenge #2

Générez un damier 8 x 8

Conseils:

- 1) Utilisez une boucle dans une autre.
- 2) Faites votre programme en 2 parties.

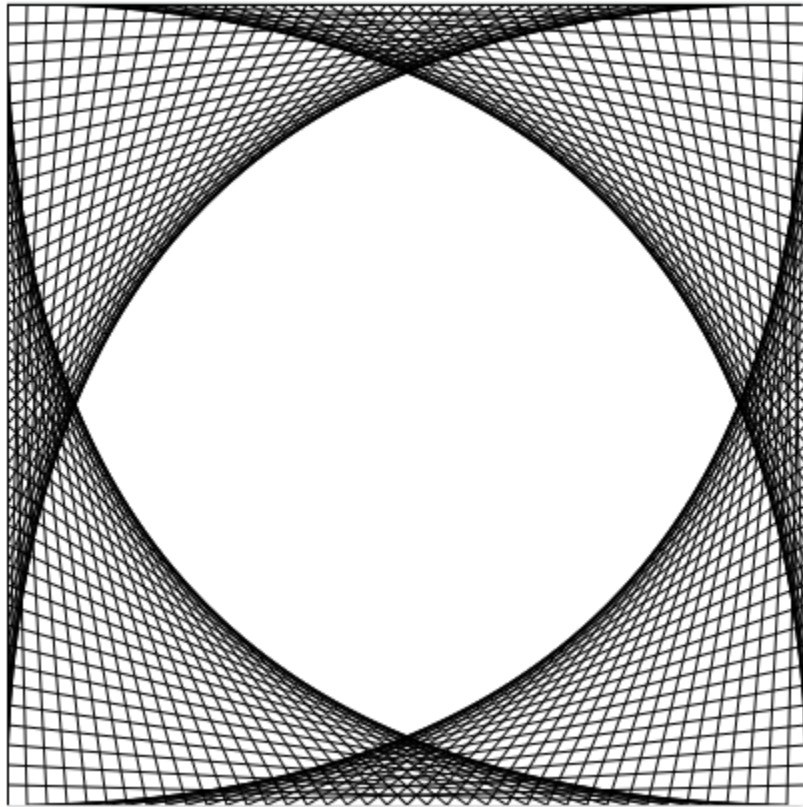
La première affiche les carrés trouvant sur les lignes 1, 3, 5, 7.

Affichez ensuite les carrés trouvant sur les lignes 2, 4, 6, 8.

Challenge #2 solution

```
var x, y;  
  
y = 0;  
while (y < 400) {  
  x = 0;  
  while (x < 400) {  
    fill(0, 0, 0);  
    rect(x, y, 50, 50);  
    x = x + 100;  
  }  
  y = y + 100;  
}
```

```
y = 50;  
while (y < 400) {  
  x = 50;  
  while (x < 400) {  
    fill(0, 0, 0);  
    rect(x, y, 50, 50);  
    x = x + 100;  
  }  
  y = y + 100;  
}
```



Challenge #3

- Si vous ne savez pas comment faire ce dessin, essayez sur un papier en premier.
- Ensuite, écrivez un programme pour répliquer vos actions que vous avez faites sur le papier.

Challenge #3 solution

```
var x = 0;
while (x < 400) {
  line(0, 400-x, x, 0);
    line(x, 400, 400, 400-x);
    line(x, 0, 400, x);
    line(0, x, x, 400);

  x = x + 10;
}
```

Challenge #4

- Utilisez les techniques d'animations pour créer un aquarium avec plein de poissons!
- Utilisez les images, figures géométriques, couleurs pour construire votre aquarium.
- Soyez créatif!

BRAVO!
TU ES MAINTENANT
UN JAVASCRIPT NINJA!