



PROGRAMMATION I

Session 5



0
1
0
0
1
1
1
0
0
1
0
1

{ _ _ }

< _ .

</>

{

0
1
0
1
1
0
0
0
0
1
1
0
1
1
0
1
1

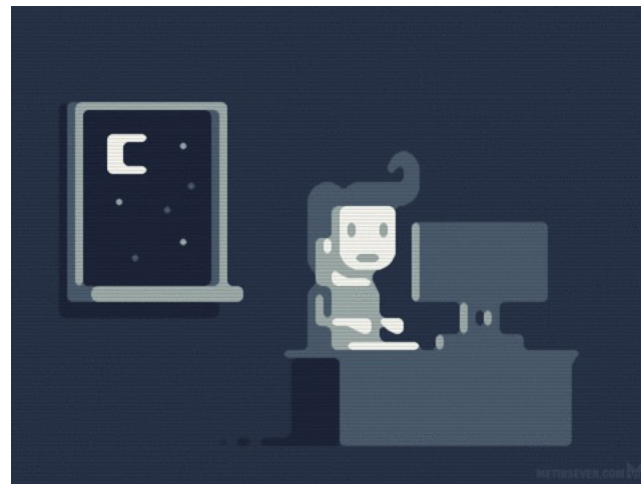
{





Au programme :

- Boucles et géométrie
- Boucles et tableaux
- Break et Continue

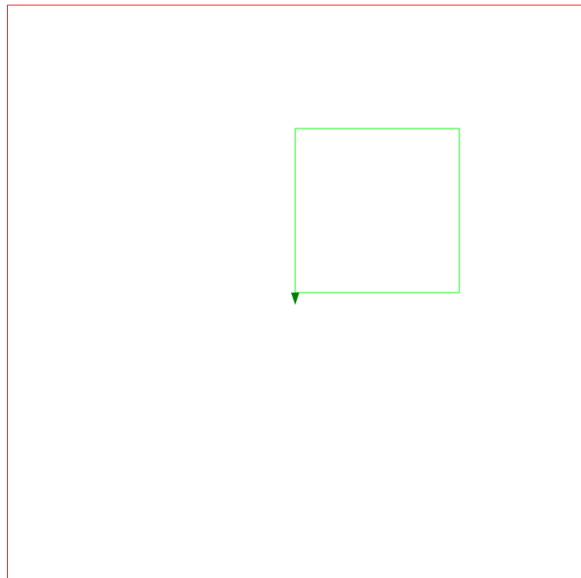


C'est parti !

Géométrie (Turtle)



Canvas

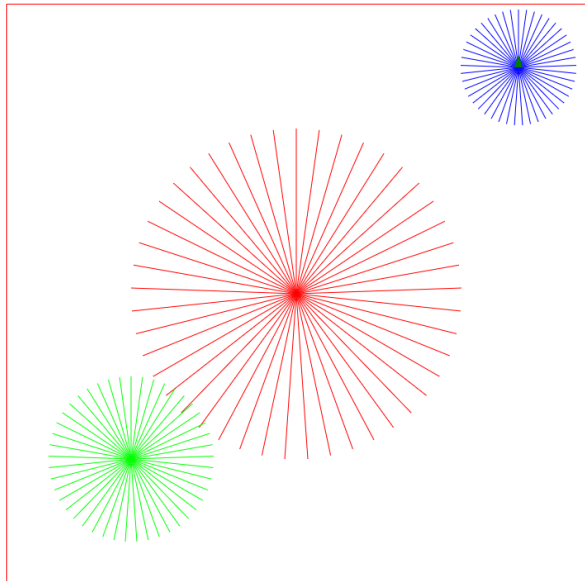


Écris un programme qui dessinera un carré en utilisant une boucle

La longueur du côté est 200



Canvas



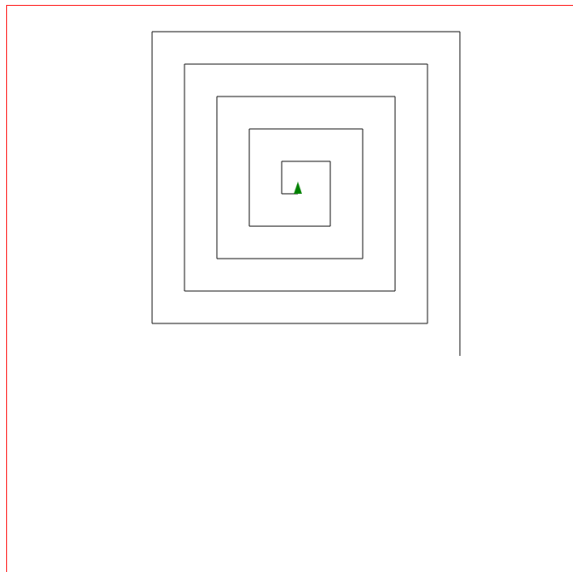
Écris un programme qui dessine les formes suivantes :

Couleur	Longueur	Angle	Coordonnées
red	200	10	0, 0
green	100	8	-200, -200
blue	70	8	269, 274

Dessine ces formes



Canvas



Écris un programme qui dessinera la forme suivante

Tableaux et boucles

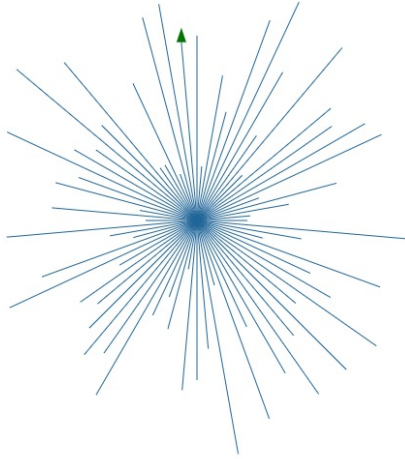


```
let arr = [];  
  
for (let i = 0; i < (100 / 3); i++) {  
  arr[i] = (i + 1) * 3;  
}  
  
console.log(arr);  
// [3, 6, 9, 12, 15, 18, ..... ]
```

Ce programme va remplir le tableau avec tous les multiples de 3 jusqu'à 100.

[[pytutor](#)]

Dessine cette étoile grâce au tableau



```
let arr = [191, 56, 152, 116, 220, 249,  
177, 107, 233, 66, 180, 170, 200, 210,  
68, 149, 96, 55, 52, 218, 109, 70, 201,  
129, 159, 226, 133, 218, 155, 219, 182,  
121, 218, 69, 245, 133, 165, 176, 51,  
116, 84, 108, 208, 167, 181, 157, 134,  
147, 92, 213, 170, 68, 91, 197, 53,  
150, 60, 151, 130, 216, 146, 127, 243,  
139, 213, 66, 66, 156, 51, 217, 227,  
185];
```



```
let arr = [191, 56, 152, 116, 220, 249,  
177, 107, 233, 66, 180, 170, 200, 210,  
68, 149, 96, 55, 52, 218, 109, 70, 201,  
129, 159, 226, 133, 218, 155, 219, 182,  
121, 218, 69, 245, 133, 165, 176, 51,  
116, 84, 108, 208, 167, 181, 157, 134,  
147, 92, 213, 170, 68, 91, 197, 53,  
150, 60, 151, 130, 216, 146, 127, 243,  
139, 213, 66, 66, 156, 51, 217, 227,  
185];
```

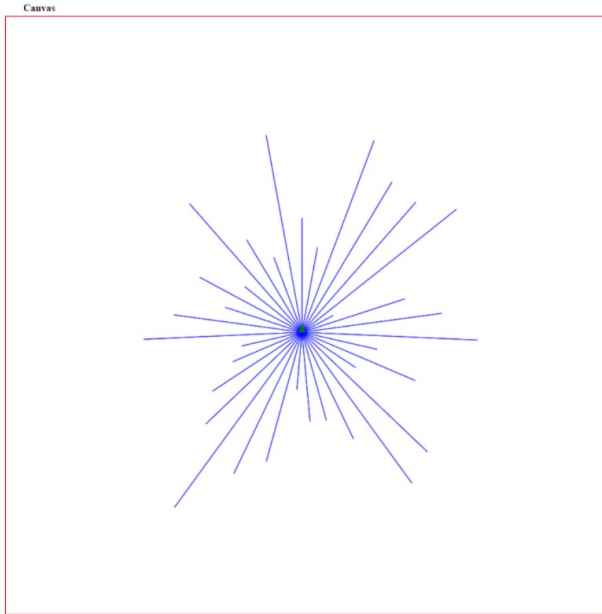


Écris un programme qui compte la quantité de nombres dans le tableau qui sont plus grands que 100. Le résultat doit être affiché dans la console.

Remplis le tableau avec des nombres aléatoires



Écris un programme qui remplit un tableau avec 36 nombres entiers aléatoires.
Puis le programme dessine une étoile basée sur ce tableau.



La fonction `Math.random()` retourne un nombre aléatoire de 0 à 1.



`Math.floor()` arrondit un nombre à la baisse.

1,5 - 1

2,4 - 2

2,1 - 2

2,9 - 2

```
let x = Math.floor(101 * Math.random());  
console.log(x);
```



```
let animals = ['Lion', 'Monkey', 'Unicorn', 'Human', 'Dragon', 'Eagle'];  
  
for (let i = 0; i < animals.length; i++) {  
  if (animals[i] == "Human") {  
    console.log("Human was found !");  
    break;  
  }  
}
```



La propriété `.length` donne la longueur d'un tableau.



Lorsque l'expression `break` est rencontrée, la boucle s'arrête.



```
let animals = ['Lion', 'Monkey', 'Unicorn', 'Human', 'Dragon', 'Eagle'];  
  
for (let i = 0; i < animals.length; i++) {  
  if (animals[i] == "Human") {  
    continue; // Ignoring humans  
  }  
  console.log(animals[i]);  
}
```

[Pythontutors : boucles et continue](#)

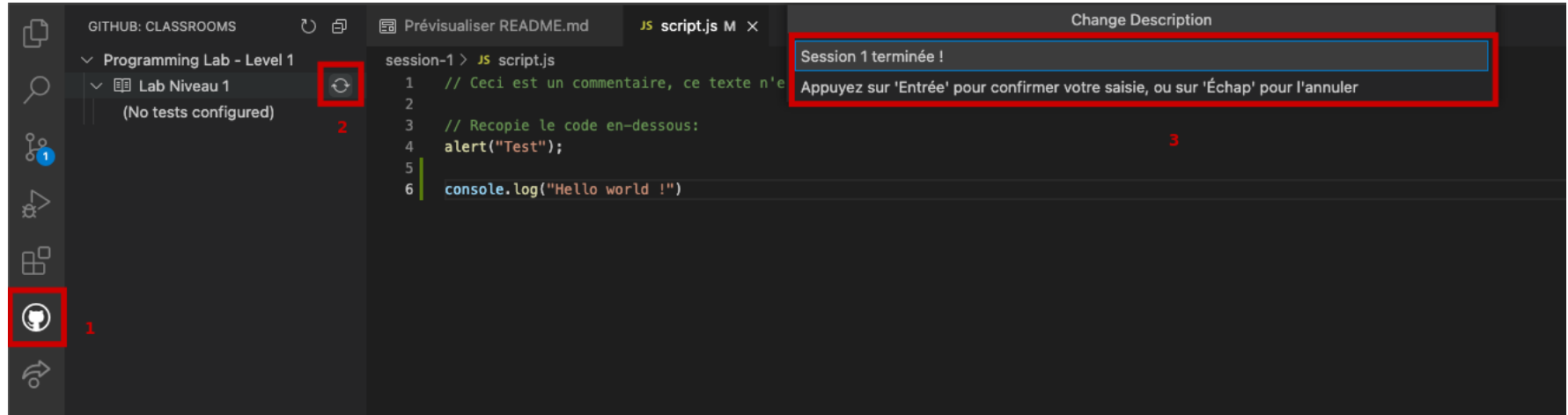


Lorsque l'expression **continue** est rencontrée, la boucle passe directement à l'itération suivante.

Enregistrer ton travail



Retourne dans l'onglet GitHub et clique sur l'icone  correspondant à ce lab.



The screenshot displays the GitHub Classroom interface. On the left sidebar, under 'GITHUB: CLASSROOMS', the 'Programming Lab - Level 1' is expanded, showing 'Lab Niveau 1' with '(No tests configured)'. A red box labeled '1' highlights the GitHub icon in the sidebar. A red box labeled '2' highlights the refresh icon next to 'Lab Niveau 1'. The main area shows a code editor for 'JS script.js' with the following content:

```
session-1 > JS script.js
1 // Ceci est un commentaire, ce texte n'e
2
3 // Recopie le code en-dessous:
4 alert("Test");
5
6 console.log("Hello world !")
```

A 'Change Description' dialog box is open on the right, containing the text: 'Session 1 terminée !' and 'Appuyez sur 'Entrée' pour confirmer votre saisie, ou sur 'Échap' pour l'annuler'. A red box labeled '3' highlights the 'Entrée' key instruction.

Ressources supplémentaires



+



COPIER



+



COUPER



+



COLLER



+



ENREGISTRER



+



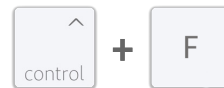
REVENIR EN ARRIERE



+



REVENIR EN AVANT



+



RECHERCHER



+



REEMPLACER



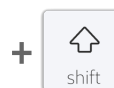
+



{ / }



+



[/]



Mac



=>



Description des fonctions de Turtle



Fonction	Signification	Paramètres
forward(distance)	Avance par distance	Distance
right(angle)	Tourne vers la droite d'un angle donné (en degrés)	Angle
left(angle)	Tourne vers la gauche d'un angle donné (en degrés)	Angle
goto(x,y)	Déplace la tortue au point (x, y)	x-coordonnée, y-coordonnée
clear()	Efface	
penup()	Relève le stylo	
pendown()	Abaisse le stylo	
reset()	Commence dès le début	
width(width)	L'épaisseur du stylo	Épaisseur
color(r,g,b)	Couleur	Rouge, vert, bleu



Labs et livres importants

1. <http://eloquentjavascript.net> (en)
2. <https://www.w3schools.com/js/default.asp> (en)

Information

- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference> (en)
- <https://www.w3schools.com/jsref/default.asp> (en)
- <https://hanumanum.github.io/js-turtle/>

Fin de session

À la prochaine !