



# PROGRAMMATION I

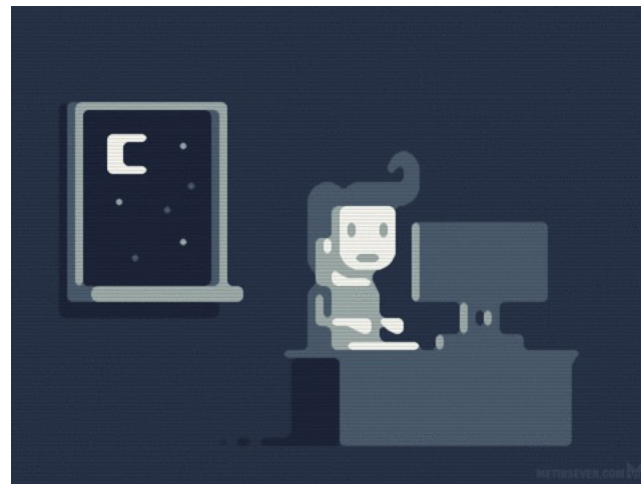
## Session 6





Au programme :

- Fonctions et Arrays
- Fonctions et mathématiques
- Création de fonctions



C'est parti !

---

# Fonctions

---



```
alert("Hello !");  
console.log("Some value");  
let x = Math.sqrt(25);
```



La fonction `Math.sqrt()` calcule la racine carrée du nombre passé en argument.



Les **fonctions** sont de petits programmes indépendants (**sous-programmes**), qui sont nommés et peuvent être utilisés dans notre code.



Les fonctions peuvent prendre des données et ou en retourner.



Dans cet exemple “Hi”, “some value” et 25 sont les **arguments** des fonctions.



L'utilisation d'une fonction dans le code est appelée **appel de fonction (call of a function)**.

# Bibliothèque standard

---

L'utilisation des fonctions existantes



Commentons ligne par ligne

Fonction	Signification
push()	Ajoute un nouvel élément à la fin du tableau
pop()	Efface le dernier élément du tableau
shift()	Efface le premier élément du tableau
unshift()	Ajoute un nouvel élément au début du tableau

Pythontutor : travaillant avec les tableaux

```
let animals = ['Lion', 'Monkey', 'Unicorn', 'Dog', 'Dragon', 'Eagle'];  
  
animals.push('Horse');  
console.log(animals); // ?  
  
animals.push('Cat');  
console.log(animals); // ?  
  
animals.pop();  
animals.pop();  
animals.pop();  
console.log(animals); // ?  
  
animals.shift();  
console.log(animals); // ?  
  
animals.unshift('Lemur');  
animals.unshift('Centaur');  
console.log(animals); // ?
```



Commentons ligne par ligne

Fonction	Signification
Math.round()	Arrondit le nombre
Math.floor()	Arrondit le nombre à la baisse
Math.ceil()	Arrondit le nombre à la hausse
Math.sqrt()	Retourne la racine carrée du nombre
Math.abs()	Retourne la valeur absolue du nombre
Math.random()	Retourne un nombre aléatoire entre 0 et 1

```
let a = Math.round(25.640);  
console.log(a); // 26
```

```
let b = Math.round(17.21);  
console.log(b); // 17
```

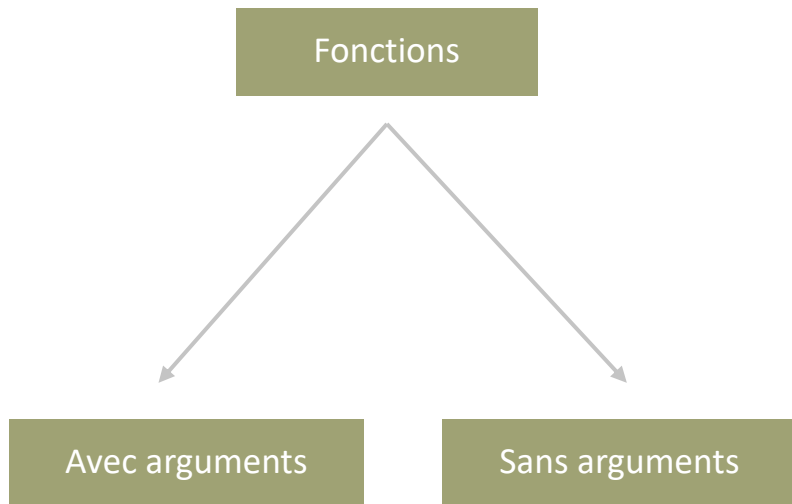
```
let c = Math.sqrt(25);  
console.log(c); // 5
```

```
let d = Math.abs(-11);  
console.log(d); // 11
```





## Types de fonctions (du point de vue d'argument)



### Classifiez les fonctions

```
let animals = ['Lion', 'Monkey', 'Unicorn', 'Dog', 'Dragon', 'Eagle'];

animals.push('Horse');
console.log(animals); // ?

animals.push('Cat');
console.log(animals); // ?

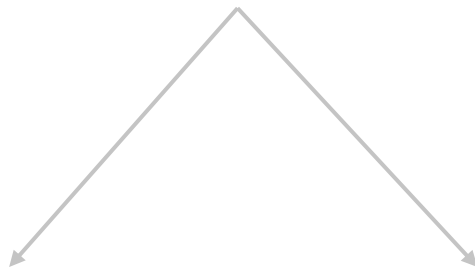
animals.pop();
animals.pop();
animals.pop();
console.log(animals); // ?

animals.shift();
console.log(animals); // ?

animals.unshift('Lemur');
animals.unshift('Centaur');
console.log(animals); // ?
```



### Fonctions



Retournent  
une valeur

Ne retournent rien



### Classifie les fonctions

```
let input = prompt("Input your age");
let age = parseInt(input);

console.log(input);
alert(age);

let rdm = Math.random();
```



Si la fonction de retourne pas de valeur,  
la valeur de retour est **null**.



Groupe	Liens
Fonctions mathématiques	<a href="https://www.w3schools.com/jsref/jsref_obj_math.asp">https://www.w3schools.com/jsref/jsref_obj_math.asp</a>
Fonctions de tableaux	<a href="https://www.w3schools.com/jsref/jsref_obj_array.asp">https://www.w3schools.com/jsref/jsref_obj_array.asp</a>
Fonctions de strings	<a href="https://www.w3schools.com/jsref/jsref_obj_string.asp">https://www.w3schools.com/jsref/jsref_obj_string.asp</a>
Fonctions de temps	<a href="https://www.w3schools.com/jsref/jsref_obj_date.asp">https://www.w3schools.com/jsref/jsref_obj_date.asp</a>

# Nos propres fonctions

---

La création des fonctions



Quand tu crées une fonction, tu dois te poser ces trois questions :

1. Quel nom fais sens pour ma fonction ?
2. Est-ce que ma fonction reçoit des arguments ? Si oui, lesquels ?
3. Ma fonction retourne-t-elle une valeur ?

Écris une fonction qui calcule le carré (square) d'un nombre



```
function square(number) {  
    return number * number;  
}  
  
square(5);
```

Écris une fonction avec une alerte « Hello Tumo ! »



```
// La définition de la fonction
function saluer() {
    console.log("Hello Tumo!");
}

// Appel de la fonction
saluer();
```



Déclaration

Titre

Arguments

```
function calcPower(number, exposant) {  
  let result = 1;  
  
  for (let i = 0; i < exposant; i++) {  
    result *= number;  
  }  
  
  return result;  
}
```



Il n'y a pas de limite au nombre d'arguments d'une fonction.

En revanche une fonction ne peut retourner qu'une seule valeur.

Corps

Retour de la valeur



# Turtle

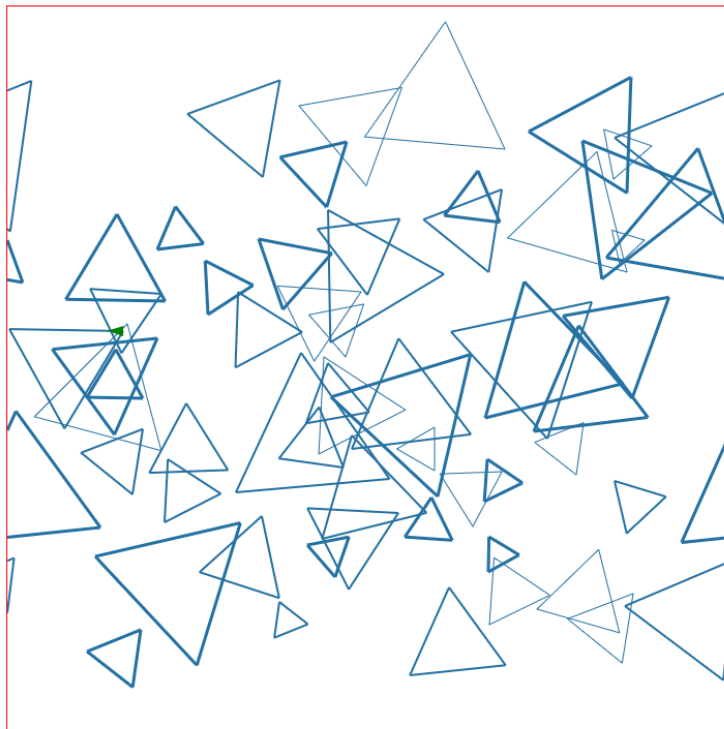
---

Et les fonctions

Discutons de cette image



Canvas



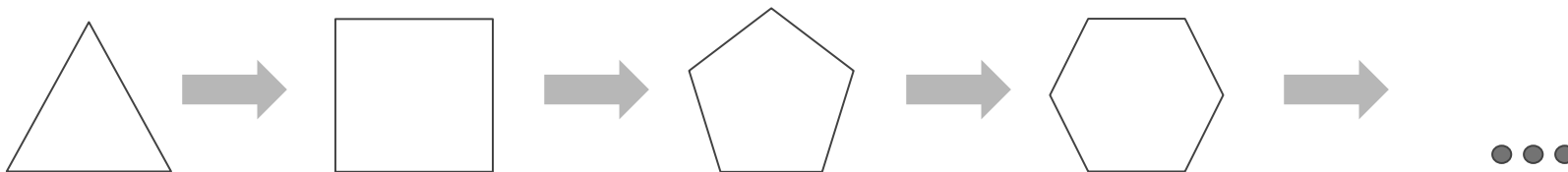


```
drawPoly(3, 100);  
drawPoly(4, 100);  
drawPoly(5, 100);  
drawPoly(6, 100);
```

La fonction s'appellera **drawPoly**.

La fonction prendra comme arguments le nombre de côtés et la taille des côtés.

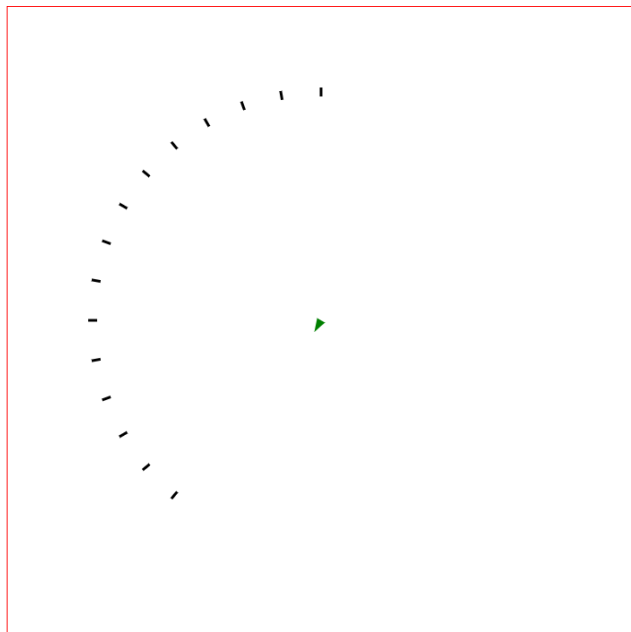
La fonction n'a pas besoin de retourner de valeur



## Créons une fonction qui dessine un cercle



Canvas



`drawCircle(x, y, r)`



```
getRandInt(0, 255);  
getRandInt(5, 10);  
getRandInt(15, 100);
```



Crée une fonction qui retourne un nombre entier aléatoire dans l'intervalle mentionné.

La fonction s'appellera `getRandInt` et prendra comme arguments les valeurs `minimum` et `maximum`.



La fonction `Math.random()` retourne un nombre aléatoire entre 0 et 1.



Écris une fonction qui crée et remplit un tableau avec 10 nombres entiers aléatoires compris dans une intervalle.

La fonction s'appellera `getRandArray` et prendra comme arguments la valeur `minimum` et `maximum`.

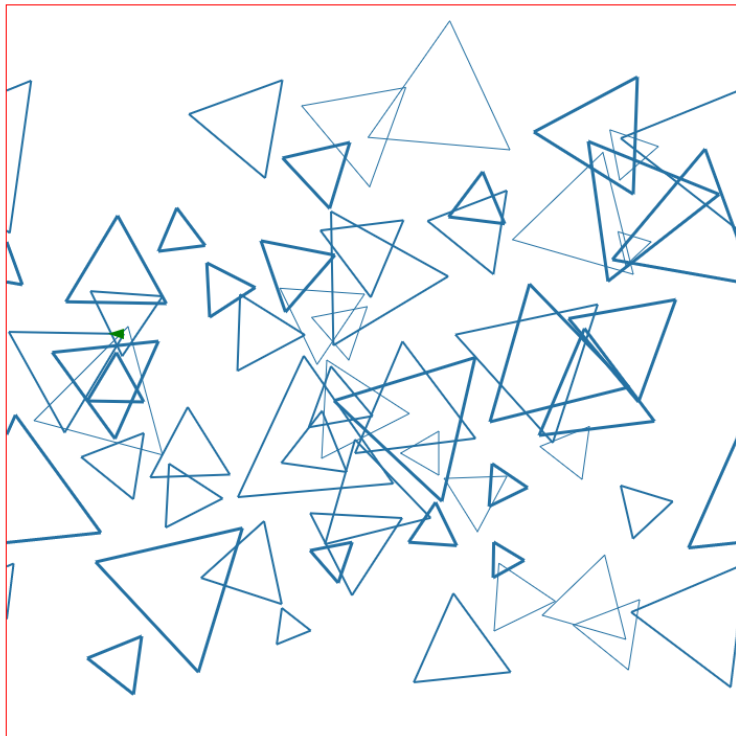
```
let diceRolls = getRandArray(1, 6);  
  
console.log(numbers);  
// [2, 5, 5, 3, 1, 4, 6, 3, 2, 4]
```



Réutilise la fonction `getRandInt` précédemment réalisée



Canvas




Écris un programme qui dessine ces formes :

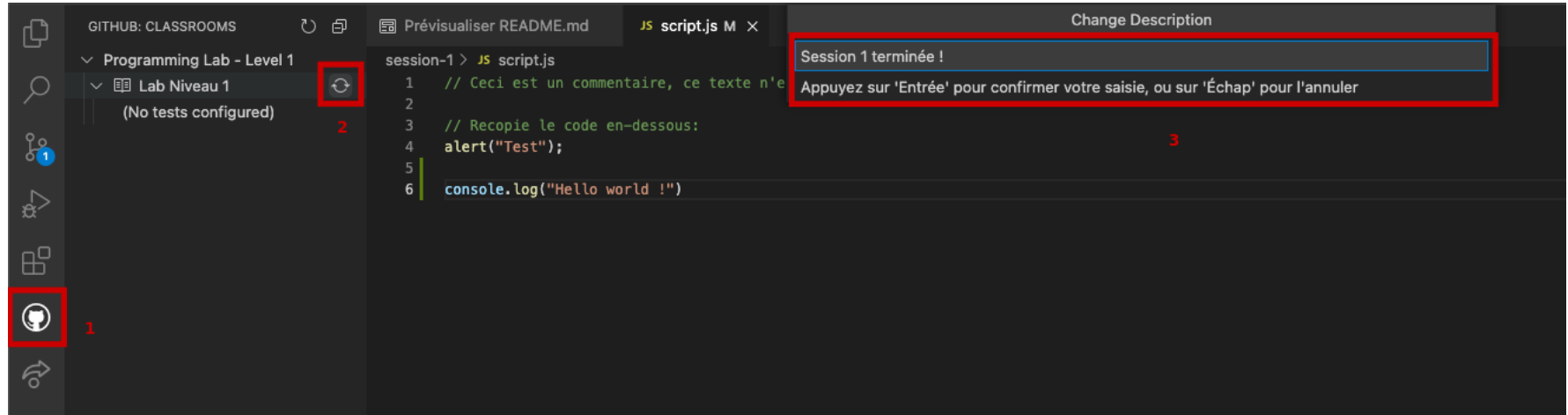
Enregistrer ton travail

---





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.

Apprends à apprendre !

---



- Parfois les problèmes ne sont pas résolus immédiatement. Ne sois pas découragé.
- Même si ça ne vient pas tout de suite, tu trouveras la solution plus tard.

# Ressources supplémentaires

---



+



COPIER



+



COUPER



+



COLLER



+



ENREGISTRER



+



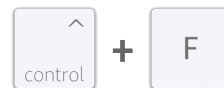
REVENIR EN ARRIERE



+



REVENIR EN AVANT



+



RECHERCHER



+



REEMPLACER



+



{ / }



+



[ / ]



Mac



=>





## Sur les fonctions

1. [http://eloquentjavascript.net/03\\_functions.html#h\\_H2WKvqbgVY\(en\)](http://eloquentjavascript.net/03_functions.html#h_H2WKvqbgVY(en))



## Liens sur les fonctions

### Fonctions mathématiques

[https://www.w3schools.com/jsref/jsref\\_obj\\_math.asp](https://www.w3schools.com/jsref/jsref_obj_math.asp)

### Fonctions de tableaux

[https://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](https://www.w3schools.com/jsref/jsref_obj_array.asp)

### Fonctions de strings

[https://www.w3schools.com/jsref/jsref\\_obj\\_string.asp](https://www.w3schools.com/jsref/jsref_obj_string.asp)

### Fonctions de temps

[https://www.w3schools.com/jsref/jsref\\_obj\\_date.asp](https://www.w3schools.com/jsref/jsref_obj_date.asp)



### 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 !