



PROGRAMMATION I

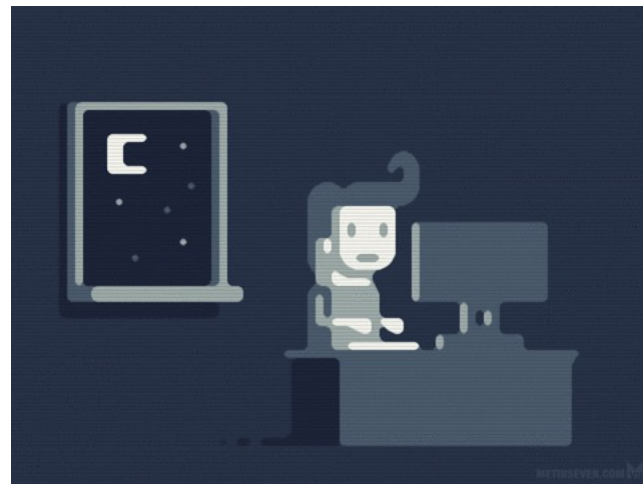
Session 1





Au programme :

- Les fonctions (*prompt*, *alert*, *parseInt...*)
- Les opérateurs de comparaison
- Les opérateurs logique
- Les conditions



C'est parti !

Fonctions



```
alert("Bonjour Tumo");  
console.log("Hello!");  
let z = Math.sqrt(25);
```



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



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



Les fonctions prennent des données et réalisent une certaine action avec ces données.



“Bonjour Tumo”, “Hello!” et 25 sont les **arguments** de ces fonctions.



L'utilisation d'une fonction dans un code est appelée **appel de fonction** (function call en anglais).



```
let a = prompt("Entrez le nombre a");  
let b = prompt("Entrez le nombre b");  
  
let sum = a + b;  
console.log(sum);
```

```
let a = prompt("Entrez le nombre a");  
let b = prompt("Entrez le nombre b");  
  
let sum = parseInt(a) + parseInt(b);  
console.log(sum);
```



La fonction `prompt()` prend les données de l'utilisateur et les affecte à une variable.



Les valeurs de `prompt()` sont toujours des strings (chaînes de caractères).

Cela signifie que même si l'utilisateur saisit des chiffres, `a` et `b` seront des strings.



La fonction `parseInt()` transforme une string en un nombre entier.

La fonction `parseFloat()` transforme une string en un nombre à virgule.

Opérateurs de comparaison



```
let a = 10;  
let b = 100;
```

```
let i = (a == b);  
let j = (a > b);
```



Les **opérateurs de comparaison** comparent les valeurs.



Le résultat de l'action des opérateurs de comparaison est une valeur booléenne : **true** ou **false**.



Opérateur	Signification	Utilisation
==	Égal à	10 == 4
>	Supérieur à	10 > 4
<	Inférieur à	10 < 4
>=	Supérieur ou égal à	10 >= 4
<=	Inférieur ou égal à	10 <= 4
!=	Pas égal à	10 != 4



Opérateur	Signification	Action
=	Affectation	Donne une nouvelle valeur à une variable.
==	Égalité	Vérifie si les valeurs sont égales.
===	Égalité stricte	Vérifie si les valeurs sont égales et si les types de données sont identiques.



Ne confonds pas les opérateurs = et ==, ils ne signifient pas du tout la même chose.
Cette erreur peut être très difficile à déboguer.



```
let a = 10;  
let b = 100;  
let c = 10;  
let d = 10;
```

```
console.log(a == b);  
console.log(a > b);  
console.log(d > c);  
console.log(c >= d);
```

```
let a = "Tumo";  
let b = "Lyon";  
let c = 15;  
let d = "15";
```

```
console.log(a == b);  
console.log(a != b);  
console.log(c == d);  
console.log(c === d);
```

Opérateurs logiques



```
let a = 14;
let b = 20;
let c = 100;

console.log(a < b && b < c);
console.log(a < b && b > c);
console.log(a < b || b == c);
```



Les **opérateurs logiques** sont permettent de créer de conditions complexes..



Comme les opérateurs de comparaison, ils vont valoir **true** ou **false**.

Opérateur	Nom
& &	And (et)
	Or (ou)



Attention, les opérateurs & et | existent mais leur fonction est très différente.



```
let a = 10;
let b = 20;
let c = 100;

console.log(a < b && c > a);
console.log(a < b || c > a);
console.log(a == b && c != a);
console.log(a == b || c != a);
console.log(a != b && b > a);
console.log(a != b || b > a);
```

```
let type = "monkey";
let age = 7;

console.log(type == "monkey" && age == 7);
console.log(type == "monkey" || age == 8);
console.log(type == "elephant" && age == 7);
console.log(type == "elephant" || age == 7);
```

Structures conditionnelles



```
let type = "monkey";  
let age = 7;  
  
console.log(type);  
  
let result = Math.sqrt(25);  
console.log(result);
```



Les programmes séquentiels, que nous avons vu jusqu'à présent sont des programmes dans lesquels toutes les commandes sont exécutées l'une après l'autre.



Dans ces séquences, il n'est pas possible d'utiliser des conditions.



```
let age = prompt("How old are you?");  
  
age = parseInt(age);  
  
if (age < 18) {  
    console.log("Bienvenue à TUM0 !");  
}
```



script.js



```
let age = prompt("How old are you?");  
  
age = parseInt(age);  
  
if (age < 18) {  
    console.log("Bienvenue à TUM0 !");  
}
```

Condition

Scope



1. La structure conditionnelle commence par l'expression if.
2. La **condition** est écrite dans les parenthèses.
3. Le **corps de la structure** est écrit entre accolades.



Si la **condition** est vraie, alors la séquence de commande dans le **corps** est exécutée.



```
let age = prompt("How old are you?");

age = parseInt(age);

if (age < 18) {
  console.log("Bienvenue à TUM0 !");
}
else {
  console.log("TUM0 est réservé aux moins de 18 ans !")
}
```



Ajoute les parties manquantes du code précédent.



script.js



```
let age = prompt("How old are you?");
```

```
age = parseInt(age);
```

```
if (age < 18) {  
  console.log("Bienvenue à TUM0 !");  
}
```

```
else {  
  console.log("TUM0 est réservé aux moins de 18 ans !");  
}
```

Condition

Scope du if

Scope du else



- Si la **condition** est vraie, la séquence de commande écrite dans le **scope du if** est exécutée.
- Sinon celui du else est exécuté.

Exercice : réécris le code



```
let word = prompt("Entrez le nom d'un animal");

if (word == "lion") {
  console.log("Le lion est un mammifère");
}
else if (word == "serpent") {
  console.log("Le serpent est un reptile");
}
else if (word == "aigle") {
  console.log("L'aigle est un oiseau");
}
else {
  console.log("Je ne connais pas le(s) " + word);
}
```



Ajoute des commentaires.



script.js



```
public class Player : MonoBehaviour
{
    public float health = 100;
    public Vector3 SpawnPosition;

    void Start ()
    {
        SpawnPosition = transform.position;
    }

    void Update()
    {
        if ( health < 0 )
        {
            transform.position = SpawnPosition;
            health = 100;
        }
    }
}
```

Et s'il n'y avait pas de portail ?



```
let word = prompt("Entrez le nom d'un animal");

if (word == "lion") {
  console.log("Le lion est un mammifère");
}
else if (word == "serpent") {
  console.log("Le serpent est un reptile");
}
else if (word == "aigle") {
  console.log("L'aigle est un oiseau");
}
else {
  console.log("Je ne connais pas le(s) " + word);
}
```



Avec plusieurs if et else if, toutes les conditions sont vérifiées l'une par l'autre.



Seule la première condition vraie est exécutée.



Si aucune condition n'est vraie, seul le scope du dernier else s'exécute.



script.js



```
let word = prompt("Entrez le nom d'un animal");

if (word == "lion") {
  console.log("Le lion est un mammifère");
}
else if (word == "serpent") {
  console.log("Le serpent est un reptile");
}
else if (word == "aigle") {
  console.log("L'aigle est un oiseau");
}
else {
  console.log("Je ne connais pas le(s) " + word);
}
```



Ajoute l'information suivante

cheval	mammifère
écrevisse	crustacé
églantine	plante

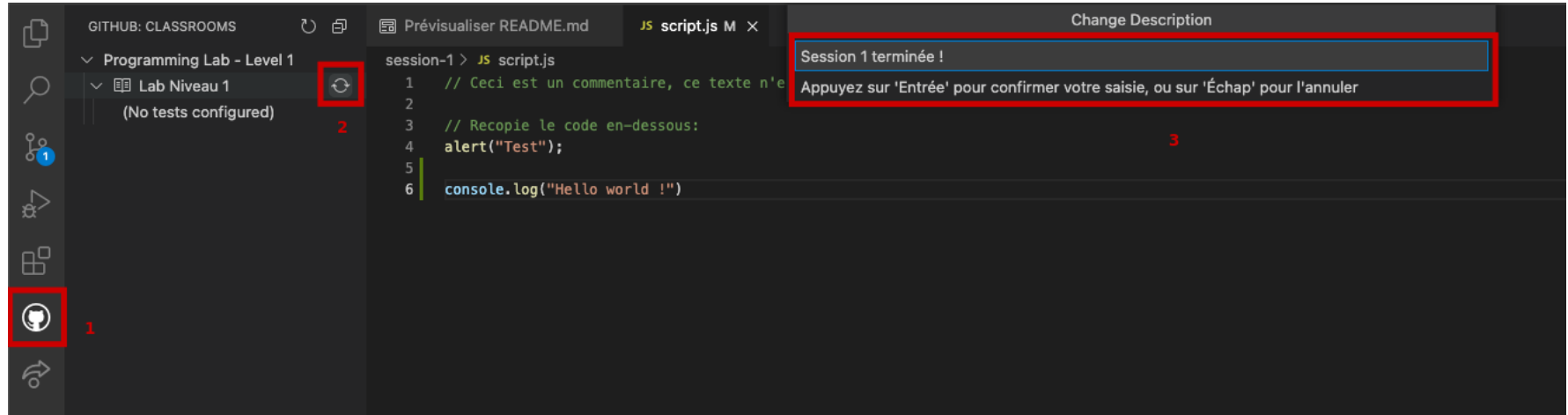


script.js

Enregistrer ton travail



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



The screenshot shows the GitHub Classroom interface. On the left sidebar, under 'GITHUB: CLASSROOMS', there is a section 'Programming Lab - Level 1' containing 'Lab Niveau 1'. A red box labeled '1' highlights the GitHub icon in the sidebar. In the main area, the 'Lab Niveau 1' is selected, and a red box labeled '2' highlights the refresh icon. The code editor shows a file named 'JS script.js' with the following content:

```
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 !")
```

On the right, a 'Change Description' dialog box is open, 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 in the instructions.

Apprends à apprendre !

Nouveaux termes



Termes
if
else
alert

Ressources supplémentaires



+



COPIER



+



COUPER



+



COLLER



+



ENREGISTRER



+



REVENIR EN ARRIERE



+



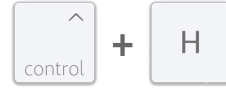
REVENIR EN AVANT



+



RECHERCHER



+



REEMPLACER



+



{ / }



+



[/]



Mac



=>





Opérateurs de comparaison et logiques, structures conditionnelles

1. http://eloquentjavascript.net/02_program_structure.html#h_rDxYNPd65Z (en)

Algèbre logique et tableaux de true/false

- <http://silon.slaks.net/#gates>

Comment JavaScript fait la comparaison

- <https://dorey.github.io/JavaScript-Equality-Table/>



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/>



if

if-else

Fin de session

À la prochaine !