

Exploration

Projets d'apprentissages en autonomie

Rédigé par

David ROUMANET
Professeur BTS SIO



Changement

Date	Révision
2024-10-13	Ajout des projets sur les itérations
2024-10-18	Correction D2.3 Liste joueur et ajout question

Sommaire

A Introduction.....	1
A.1 Présentation.....	1
A.2 Préparation.....	1
A.3 Assistance.....	1
B Activité sur les variables.....	2
B.1 Manipulation de variables.....	2
B.1.1 Calcul de TVA.....	2
B.1.2 Données des utilisateurs.....	3
C Activité sur les conditions.....	4
C.1 Préparation.....	4
C.2 Gestion de conditions.....	4
C.2.1 Programme de remise.....	4
C.2.2 Programme de genre.....	5
C.2.3 Programme comptage de caractères.....	6
C.2.4 Mot de passe.....	7
C.2.5 Deviner mon nombre.....	8
D Activité sur les itérations.....	9
D.1 Préparation.....	9
D.2 Gestion de tableaux.....	9
D.2.1 Lecture et affichage de températures.....	9
D.2.2 Affichage de la plus basse et la plus haute température.....	9
D.2.3 Créer une liste de joueurs.....	10
D.2.4 Affichage liste de joueurs.....	11
D.2.5 Gestion de notes.....	11
D.2.6 Quel jour ?.....	14

Nomenclature :

- **Assimiler** : cours pur. Explication théorique et détaillée (globalement supérieur à 4 pages).
- **Décoder** : fiche de cours, généralement inférieure à 5 pages.
- **Découvrir** : Travaux dirigés. Faisable sans matériel.
- **Explorer** : Travaux pratiques. Nécessite du matériel ou des logiciels.
- **Mission** : Projet encadré ou partie d'un projet.
- **Voyager** : Projet en autonomie totale. Environnement ouvert : Vous êtes le capitaine !

A Introduction

A.1 Présentation

Dans cette activité, vous allez vérifier vos connaissances et vos capacités de raisonnement sur les différents thèmes abordés en cours :

- Les variables
- Les conditions
- Les fonctions
- Les itérations
- Etc.

Vous évoluerez à votre vitesse et pourrez apprendre et valider vos capacités à votre rythme.

A.2 Préparation

Dans votre éditeur de code (VSCode, Sublim Text ou n'importe quel éditeur de texte), se placer dans votre répertoire de travail (par exemple `Z:\workspace\JavaScript`)

Dans ce répertoire, créer un dossier `90-Mes projets`. À l'intérieur de ce répertoire, vous créez les fichiers tels que décrits dans les activités à suivre.

A.3 Assistance

En cas de difficulté, vous regarderez les aides proposées, mais pour demander de l'aide au professeur, il faut impérativement que :

1. Votre code soit correctement indenté/formaté (rappel : `CTRL+A` puis `CTRL+K`, `CTRL+F` ou encore `Alt+Maj+F`)
2. Votre code soit sauvegardé et que celui qui est affiché dans le navigateur soit le même fichier
3. La console de débogage soit ouverte (`F12` ou `CTRL+Maj+I` ou **Inspecter l'élément**), sur l'onglet `Console`.

Il est également fortement recommandé de vérifier le nombre de parenthèses et d'accolades



En effet, vous êtes plusieurs en classe et le professeur ne peut passer tout son temps avec vous : c'est une ressource rare et précieuse.

B Activité sur les variables

Créer un fichier **01-ManipulationsVariables.html** et utiliser le langage Emmet pour créer la structure HTML, avec le caractère `!` Puis `Entrée`.

01-ManipulationsVariables.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

À l'intérieur des balises `<body>` ajouter les balises `<script>` et `</script>` et placer le curseur entre ces deux balises, comme ci-dessous :

```
<body>
  <script>
    █
  </script>
</body>
```

B.1 Manipulation de variables

B.1.1 Calcul de TVA

Créez les variables suivantes :

prixHT	200
tva	20
prixTTC	0

Calculer le prixTTC en multipliant prixHT avec $(1+tva/100)$

Afficher sur une même ligne, dans la console, le prixHT, le taux de TVA et le prixTTC

 Le prixTTC doit être égal à 240

Créer une nouvelle variable de type texte qui affichera le résultat ci-dessous, en concaténant les variables :

`200` € x TVA `20` % = `240` €

B.1.2 Données des utilisateurs

Ajouter les variables suivantes :

prenom	Chuck
nom	NORRIS
fullName	''

Puis changer la valeur de `fullName` en y plaçant le `nom` puis un espace puis le `prénom` de l'utilisateur.

Sur une ligne, afficher le nom, le prénom et enfin `fullName` pour obtenir l'affichage suivant dans la console.

```
Prénom : Chuck, Nom : NORRIS (NORRIS Chuck).
```

La fonction pour passer un texte en minuscule s'appelle `toLowerCase()` et la fonction pour le passer en majuscule s'appelle `toUpperCase()`. La syntaxe pour l'utiliser sur une variable `nomVariable` est `nomVariable.toLowerCase()` ou `nomVariable.toUpperCase()`.

Changer le prénom en majuscules (la variable `prenom` devra contenir "CHUCK" et plus "Chuck").

Vérifier en affichant la variable `prenom` dans la console.

C Activité sur les conditions

C.1 Préparation

Comme dans l'activité sur les variables, créer un fichier HTML ayant pour nom :

02-ManipulationsConditions.html.

Créer la zone script comme précédemment et placer le curseur à l'intérieur.

C.2 Gestion de conditions

C.2.1 Programme de remise

Vous allez réaliser un programme capable d'appliquer une remise différente en fonction du prix total d'un achat.

> 1000 €	20 % de remise
>700	10 % de remise
Autres montants	0 % de remise

Il faut créer les variables suivantes :

prixTotal	Placer un prix pour tester
PrixFinal	0
nomClient	"Chuck Norris"

L'objectif est d'afficher

```
Client : Chuck Norris, bénéficie d'une remise de xx % !
Prix total = xxxx € (prix avant remise = xxxx €)
```

💡 Il faut créer un texte concaténé (utilisant donc le symbole `+` entre les variables et chaînes)

💡 Il faut afficher deux lignes séparées (ligne 1 : "Client...", ligne 2 : "Prix total...")

💡 La structure conditionnelle est `if – else if – else if – ... – else`

💡 la condition est toujours contenue par deux parenthèses (revoir le support de cours)

C.2.2 Programme de genre

Sauvegardez le code suivant sous le nom **03-manipulationsConditions.html**

03-manipulationsConditions.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Définir genre</title>
</head>
<body>
  <input type="text" id="genre" placeholder="Saisir votre genre">
  Bonjour <span id="reponse"></span>
  <script>
    // recherche de l'objet 'input' ayant pour id 'genre'
    let oGenre = document.getElementById("genre")
    // Recherche de l'objet 'span' ayant son id défini à 'reponse'
    let oReponse = document.getElementById("reponse")
    // Ajoute un événement lorsqu'on saisi du texte dans 'genre'
    oGenre.addEventListener("keyup", DefinirGenre)

    // Créer notre propre fonction DefinirGenre()
    function DefinirGenre() {
      // Récupère la première lettre du champ de saisie HTML
      let lettre = oGenre.value[0]
      // ----- complétez les tests ci-dessous -----
      if () {
        oReponse.innerHTML = "Monsieur"
      } else if () {
        oReponse.innerHTML = "Madame"
      } else {
        oReponse.innerHTML = "Iel"
      }
      // ----- fin des modifications -----
    }
  </script>
</body>
</html>
```

Compléter les tests pour que les conditions respectent le tableau suivant

Lettre	Résultat
"F" ou "f"	Madame
"M" ou "m"	Monsieur
Autres réponses	Iel

- 💡 Une réponse doit s'afficher dès qu'une touche est appuyée dans le champ de saisie
- 💡 Pour changer la casse d'un texte, on peut utiliser `lettre.toUpperCase()`
- 💡 Vous ne devez modifier QUE la partie concernant les tests (conditions)
- 💡 Pour récupérer la première lettre d'un texte, on utilise son indice entre crochet, exemple : `mot[0]`

C.2.3 Programme comptage de caractères

Sauvegardez le code suivant sous le nom **04-manipulationsConditions.html**

04-manipulationsConditions.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Comptage de lettres</title>
</head>

<body>
  <input type="text" id="txtphrase" placeholder="écrire une phrase" size="35"><br>
  Votre phrase compte <span id="reponse">aucun</span> caractères.
  <script>
    // Créer notre propre fonction compterCaracteres()
    function compterCaracteres() {
      // Récupère la première lettre du champ de saisie HTML
      let nombre = oPhrase.value.length

      // ----- complétez les tests ci-dessous -----
      if (nombre == 0) {
        oReponse.innerHTML = // <-- mettre une chaine
      } else {
        oReponse.innerHTML = // <-- mettre une variable
      }
      // ----- fin des modifications -----
    }
    // recherche de l'objet 'input' ayant pour id 'txtphrase'
    let oPhrase = document.getElementById("txtphrase")
    // Recherche de l'objet 'span' ayant son id défini à 'reponse'
    let oReponse = document.getElementById("reponse")
    // Ajoute un événement lorsqu'on saisi du texte
    oPhrase.addEventListener("keyup", compterCaracteres)
  </script>
</body>

</html>
```

Ce compte doit permettre de compter le nombre de caractères dans une phrase, grâce à la propriété `.length` de JavaScript.

Modifiez le code pour qu'il affiche 'aucun' s'il n'y a pas de caractères dans le champ de saisie, et une valeur si des caractères sont présents.

💡 Une réponse doit s'afficher dès qu'une touche est appuyée dans le champ de saisie

💡 L'instruction `oPhrase.value.length` récupère la valeur contenue dans l'objet `oPhrase` et `.length` en donne la longueur

💡 Vous ne devez modifier QUE la partie concernant les tests (conditions)

C.2.4 Mot de passe

Sauvegarder votre code précédent sous un nouveau nom : **05-manipulationsConditions.html**

(Rappel, vous pouvez passer par le menu **Fichier > Enregistrer sous...** ou bien utiliser `Ctrl+Maj+S`)

Votre objectif est de modifier le code pour afficher un message de bienvenue, si l'utilisateur a tapé le bon mot dans le champ de saisie, sinon votre programme doit afficher "Accès refusé".

💡 Il faut modifier le texte HTML de base, dans la partie **body**, ainsi que la consigne dans la propriété `placeholder`.

💡 Il faut comparer le contenu de `oPhrase.value` avec votre mot de passe voulu

💡 Il faut corriger les tests, pour afficher le bon message

💡 Un test d'égalité utilise presque toujours deux symboles : `==` pour égal, `!=` pour n'est pas égal.

💡 Une chaîne de caractères doit toujours être encadrées : `'mon Texte'`, ou `"mon Texte"` ou ``mon Texte``.

C.2.5 Deviner mon nombre

Créer un fichier **06-manipulationsConditions.html** et coller le code ci-dessous :

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Jeu deviner un nombre</title>
</head>

<body>
  <input type="text" id="txtNombre" placeholder="nombre entre 1 et 10" size="25"><br>
  Résultat <span id="reponse"></span>.
  <script>
    // Créer un premier nombre aléatoire entre 1 et 10
    let unNombre = Math.trunc(Math.random()*10)+1
    console.log(unNombre)
    // Créer notre propre fonction verifierNombre()
    function verifierNombre() {
      // Récupère le contenu du champ de saisie HTML
      let nombre = oNombre.value
      // ----- complétez les tests ci-dessous -----
      if () {
        oReponse.innerHTML =
      } else if () {
        oReponse.innerHTML =
      } else {
        oReponse.innerHTML =
        unNombre = Math.trunc(Math.random()*10)+1
      }
      oNombre.value=""
      // ----- fin des conditions -----
    }
    // recherche de l'objet 'input' ayant pour id 'txtNombre'
    let oNombre = document.getElementById("txtNombre")
    // Recherche de l'objet 'span' ayant son id défini à 'reponse'
    let oReponse = document.getElementById("reponse")
    // Ajoute un événement lorsqu'on saisi du texte
    oNombre.addEventListener("keyup", verifierNombre)
  </script>
</body>
</html>
```

Grâce à la bibliothèque mathématique de JavaScript, on peut créer des nombres aléatoires (comme tiré au hasard). Comme le tirage par défaut crée un nombre entre 0.0000001 et 0.9999999, il faut le multiplier par une valeur, puis l'arrondir pour obtenir un nombre entier.

```
// Créer un nombre aléatoire entre 1 et 10
let unNombre = Math.trunc(Math.random()*10)+1
```

L'objectif est que le joueur tape un nombre dans le champ et le programme le compare à celui choisi.

💡 Vous devez tester et afficher si `nombre` est plus petit que `unNombre`, s'il est plus grand et sinon afficher que nombre est le bon.

💡 Penser à concaténer le nombre et la phrase " est trop petit' ou " est trop grand ou " est le bon nombre. On recommence" dans la réponse

D Activité sur les itérations

D.1 Préparation

Comme dans l'activité sur les conditions, créer un fichier HTML ayant pour nom :

11-ManipulationsIterations.html.

Créer la zone script comme précédemment et placer le curseur à l'intérieur.

D.2 Gestion de tableaux

D.2.1 Lecture et affichage de températures

Un site météorologique vous donne les températures à venir, une semaine à l'avance. Votre travail est d'afficher ces températures à côté du jour (le premier jour de la semaine est toujours un lundi).

Voici le code de la partie script à insérer.

```
let semaine = [a_remplacer]
let temperatures = [18, 17, 21, 18, 14, 11, 12]

for (let i = 0; i < temperatures.length; i = i + 1) {
  document.write("Le "+semaine[i]+" il fera "+a_remplacer+"°C"+"<br>")
  // <br> permet le retour à la ligne
}
```

Vous devez remplacer les parties en orange pour que le script affiche :

```
Le lundi il fera 18°C
Le mardi il fera 17°C
...
```

D.2.2 Affichage de la plus basse et la plus haute température

Sauvegardez le fichier précédent en **12-tempMinMax.html** pour pouvoir le modifier.

Pour trouver la valeur la plus petite d'un tableau, il faut prendre la première valeur de ce tableau, puis la comparer à toutes les autres valeurs du tableau. Lorsqu'on parle de comparaison, il faut penser à une condition :

```
let semaine = [déjàFait précédemment]
let temperatures = [18, 17, 21, 18, 14, 11, 12]
let tempMin = temperatures[0]

for (let i = 1; i < temperatures.length; i = i + 1) {
  document.write("Le "+semaine[i]+" il fera "+déjàFait précédemment+"°C"+"<br>")
  if (temperatures[i] < b_compléter) {
    tempMin = temperatures[b_compléter]
  }
}
document.write("La température la plus froide est "+a_compléter+"°C"+"<br>")
```

Essayez de faire de même pour la valeur maximale.

D.2.3 Créer une liste de joueurs

Un ami à vous veut préparer une LAN party et veut savoir qui sera inscrit dans la classe. Il vous demande de créer un code HTML+JS qui sera affiché sur un ordinateur en libre-service.

Dans un premier temps, la page ne permet que la saisie d'un nom et son insertion dans un tableau JavaScript. On utilise un seul fichier ici :

13-ProjetLanParty.html

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Inscription LAN Party</title>
</head>
<body>
  <h1>13 - Inscription LAN Party</h1>

  <div>
    <input type="text" id="saisieNom" placeholder="Entrez votre nom">
    <button id="btnAjouter">S'inscrire</button>
  </div>

  <h2>Liste des inscrits :</h2>
  <ul id="listeInscrits"></ul>

  <script>
    // Tableau pour stocker les noms des inscrits
    let inscrits = à compléter

    // Fonction pour ajouter un nom à la liste
    function ajouterNom() {
      const objSaisieNom = document.getElementById('saisieNom');
      const nom = objSaisieNom.value.trim();

      // Créer une condition qui vérifie si nom n'est pas vide (!="")
      inscrits.à compléter;
      objSaisieNom.value = ''; // Vider le champ de saisie
      // fin si
    }

    // Ajouter un écouteur d'événement au bouton
    document.getElementById('btnAjouter').addEventListener('click', ajouterNom);
  </script>
</body>
</html>
```

Dans un premier temps, assurez-vous que l'initiation de la variable `inscrits` soit bien un tableau.

Dans la fonction `ajouterNom()`, utilisez la fonction `push` pour ajouter le nom du joueur saisi.

Enfin, toujours sans la fonction `ajouterNom`, vous ajouterez un test (une condition) qui vérifie que le nom n'est pas vide.

💡 Un tableau se déclare avec deux crochets, par exemple `let unTableau = []`

💡 Pour ajouter un élément à `unTableau`, on utilise la fonction `push()`, par exemple `unTableau.push(variable)`.

D.2.4 Affichage liste de joueurs

Dans le code précédent, vous avez ajouté des joueurs, sans les afficher. Il est toujours possible dans la console de débogage, d'écrire 'inscrits' pour afficher le tableau.

Désormais, essayez de créer une nouvelle fonction capable d'afficher chaque nom, les uns sous les autres dans la page web en vous basant sur le code suivant :

```
function AfficherJoueurs() {  
    for (let i = 0; i < a_replacer.length; i = i + 1) {  
        document.write(`Joueur ${i} est ${inscrits[a_replacer]}<br>`)  
        // <br> permet le retour à la ligne  
    }  
}
```

D.2.5 Gestion de notes

Dans ce scénario, une institutrice veut que ses élèves puisse comprendre comment fonctionne la moyenne des notes. Elle vous demande de rédiger une page web qui demande la saisie des notes et qui affiche celles-ci ainsi que la moyenne.

Nous allons créer notre premier programme en deux parties : une page web en HTML et un fichier de code en JavaScript.

Voici le fichier HTML :

14-saisieNotes.html

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>12 - saisie notes</title>  
</head>  
<body>  
    <div>  
        <input type="number" id="saisieNote" placeholder="Entrez une note" min="0" max="20" size="12">  
        <button id="btnAjouter">Ajouter</button>  
    </div>  
    <div id="notesList">(zone d'affichage de la liste des notes)</div>  
<hr>  
    <div id="txtMoyenne">(zone d'affichage de la moyenne)</div>  
  
    <!-- Séparation du code HTML et du code JavaScript -->  
    <script src="12-saisieNotes.js"></script>  
</body>  
</html>
```

La ligne `<script src="14-saisieNotes.js"></script>` permet d'appeler un fichier qui contiendra le script que nous écrivons dans le fichier HTML.

Il faut donc ici, sauvegarder votre fichier HTML (qui ne sera plus modifié pour la suite du projet).



On utilise une balise HTML qui s'appelle `<div>` et qui crée un bloc pouvant contenir du texte ou des composants graphiques. Ici, ce sera seulement du texte. Ce qui est différent avec la balise `` c'est que par défaut, on ne peut pas mettre deux blocs côte à côte (ils sont forcément l'un sous l'autre). Vous pouvez tester les différences sur le site de w3schools : https://www.w3schools.com/tags/tag_span.asp et https://www.w3schools.com/html/html_div.asp

Dans le même dossier, il faut créer un fichier JavaScript qui s'appelle **14-saisieNotes.js** comme dans la ligne du fichier HTML.

Voici le code à placer dans le fichier JavaScript

14-saisiNotes.js

```
// Code de manipulation de boucle
// Création du tableau 'mesNotes' pour stocker les notes
let mesNotes = [];

document.getElementById('btnAjouter').addEventListener('click', AjouterNote);

// La fonction ajouter une note
function AjouterNote() {
  const objSaisieNote = document.getElementById('saisieNote');
  const note = parseFloat(objSaisieNote.value);

  // Condition si note >=0 et note <= 20
  if (!isNaN(note) && complétez cette partie) {
    mesNotes.push(note); // insère la note dans le tableau 'notes'
    AfficherNotes();
    objSaisieNote.value = ''; // vide le champ de saisie
  } else {
    alert('Veuillez entrer une note valide entre 0 et 20.');
```

💡 Pour la condition, le symbole `&&` signifie 'ET'. `If (argent > 0 && argent <5)` vérifiera que la quantité d'argent sera strictement supérieure à 0 et strictement inférieure à 5. On peut utiliser autant de `&&` que l'on veut



Dans la boucle for, en JavaScript les tableaux commencent toujours à l'indice 0 !



Dans la boucle for, en JavaScript, il faut préciser de combien on augmente le compteur (ici, i)

💡 Pour augmenter une variable de 1, on peut écrire `laVariable = laVariable + 1` ou bien choisir la forme abrégée : `laVariable++`.

💡 Rappel : une moyenne est égale à la somme de toutes les notes, divisée par le nombre de notes.

D.2.6 Quel jour ?

Vous travaillez pour une entreprise d'affichage publicitaire. Les écrans de cette société affichent toujours un message de bienvenue, en fonction du jour de la semaine :

Le lundi : "Lundi, je vous souhaite un bon début de semaine !"

Du mardi au jeudi : "xxxx, je vous souhaite une bonne journée !"

Le vendredi : "Vendredi, c'est bientôt le weekend !"

Le samedi et le Dimanche : "xxxx, c'est le week-end : prenez le temps de vous reposer !"

Le script qui vous renvoie le jour courant, le fait sous la forme d'un nombre entre 0 et 6. Le lundi correspond à 0, le mardi à 1... le dimanche à 6.

Créez la page web capable d'afficher le message en fonction du jour courant :

15-bonjour.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>15 - Bonjour</title>
</head>

<body>
  <script>
    function aujourd'hui() {
      let dateCourante = new Date();
      let jour = dateCourante.getDate();
      let mois = dateCourante.getMonth() + 1; // Les mois commencent à 0, donc on ajoute 1
      let annee = dateCourante.getFullYear();
      return "La date d'aujourd'hui est : " + jour + "/" + mois + "/" + annee
    }

    function jourSemaine(numero) {
      let semaine = ["lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"]
      let texte = à compléter

      if (numero == 0) {
        texte = semaine[numero] + ", je vous souhaite un bon début de semaine."
      } else if (numero > 0 && numero < 4) {
        texte = à compléter + ", je vous souhaite une bonne journée."
      } else if (à compléter) {
        texte = semaine[numero] + ", c'est bientôt le weekend."
      } else {
        texte = à compléter
      }
      return texte
    }

    // appel de la fonction aujourd'hui() puis jourSemaine()
    document.write(aujourd'hui())
    document.write("<br>")
    document.write(jourSemaine(4))

  </script>
</body>
</html>
```


