

JavaScript



TP 3: JAVASCRIPT



- Partie 1: Manipulation du langage JavaScript
- Partie 2: JavaScript & DOM
- Partie 3: Mini-Application



- **Partie 1: Manipulation du langage JavaScript**
- Partie 2: JavaScript & DOM
- Partie 3: Mini-Application



Exercice 1

Les Bases de la Programmation Côté Client

Exercice 1 Les Bases de la Programmation Côté Client

À vous

HTML

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <button>Afficher</button>
  </body>
</html>
```



Afficher

1

Dans un fichier exo1.html, écrire le code HTML suivant

2

Vous devriez obtenir le résultat suivant.

Exercice 1

Les Bases de la Programmation Côté Client

HTML

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <button>Afficher</button>
  </body>
</html>
```



Afficher

1

Nous allons demander au navigateur d'exécuter un code JavaScript, lorsque le bouton "Afficher" est cliqué.

2

Dans premier temps, nous allons lui demander de faire un simple traitement.

Pour cela, nous aurons besoin de déclarer ce traitement JavaScript. Il existe 3 méthode pour ça.



Exercice 1 Les Bases de la Programmation Côté Client

Rappel Déclaration du code JavaScript

Méthode 1: En ligne

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1 id="compt_H1">Compteur=0</h1>
    <button onclick="document.getElementById('compt_H1').innerHTML='Hello!!!'">
      Incrémenter
    </button>
  </body>
</html>
```

Méthode 2: Dans un élément

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1 id="compt_H1">Compteur=0</h1>
    <button onclick="doIncrement()">
      Incrémenter
    </button>
  </body>
  <script>
    /*Traitement*/
    function doIncrement() {
      /* do Increment*/
    }
  </script>
</html>
```

Méthode 3: Fichier JS

```
<!DOCTYPE html>
<html>
  <head>
    <script src="mon_script.js"></script>
  </head>
  <body>
    <h1 id="compt_H1">Compteur=0</h1>
    <button onclick="doIncrement()">
      Incrémenter
    </button>
  </body>
</html>
```

mon_script.js

```
/*Traitement*/
function doIncrement() {
  /* do Increment*/
}
```

Où mettre le Script ?



Exercice 1 Les Bases de la Programmation Côté Client

À vous

HTML

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <button>Afficher</button>
  </body>
</html>
```



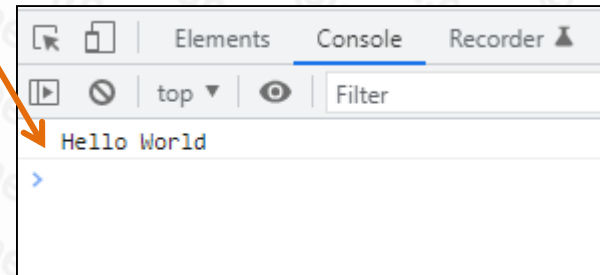
Afficher

2

Dans les outils de développement, aller sur l'onglet Console, puis cliquer sur le bouton. Vous devriez voir apparaître le chaîne Hello World.

1

En utilisant les 3 méthodes décrites dans le slide précédent, rajouter le traitement **`console.log("Hello World");`** qui va afficher *Hello World* dans la console, lorsque le bouton "Afficher" est cliqué.





Exercice 2

Manipulation des Variables & Constantes

Exercice 2 Manipulation des Variables & Constantes

À vous



Nom

Prénom

Nom & Prénom

Age

Note

Pi

1

Dans un fichier exo2.html, créer une page qui contient les 6 boutons suivants.



Exercice 2

Manipulation des Variables & Constantes

À vous

1

Dans un fichier script.js, déclarer les variables & constantes suivante:

```
nom="John";
prenom="Doe";
age=30;
note=15;
const pi=3.14;
```



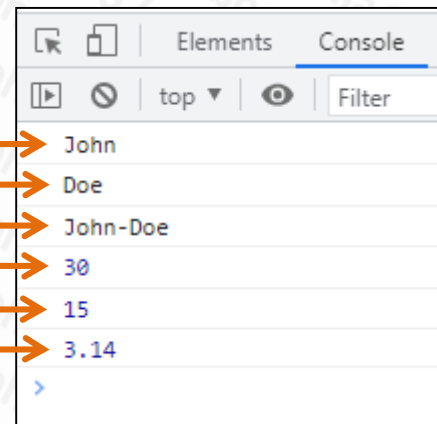
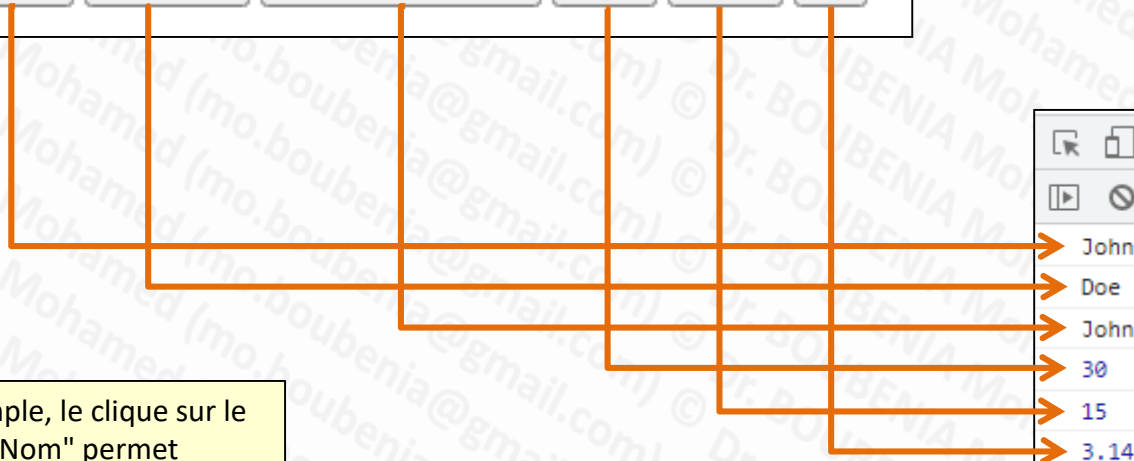
Exercice 2

Manipulation des Variables & Constantes

À vous

1

Après avoir lié le fichier exo2.html à script.js, on voudrait maintenant qu'un clique sur chacun des boutons permettent l'affichage de la valeur de la variable/constante correspondante, comme suit:



2

Par exemple, le clique sur le bouton "Nom" permet l'affichage de la valeur de la variable Nom



Exercice 3

Manipulation des Type complexes, Fonctions & Boucles



Exercice 3

Manipulation des Type complexes, Fonctions & Boucles

À vous

1

Dans un fichier exo3.html, écrire le code HTML permettant d'obtenir l'affichage suivant

Tableau Etudiants

Objet Etudiant

Tableau d'objets Etudiants



Exercice 3

Manipulation des Type complexes, Fonctions & Boucles

À vous

1

Dans un fichier script.js, déclarer un tableau **etudiantsTab**, dans lequel sera rangé les 3 valeur "**Etudiant1**", "**Etudiant2**", "**Etudiant3**".

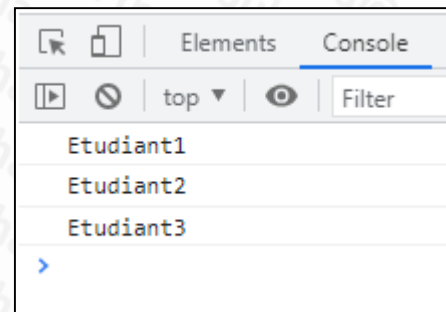
Tableau Etudiants

Objet Etudiant

Tableau d'objets Etudiants

2

En utilisant une boucle, on souhaiterait maintenant qu'un clique sur le bouton "Tableau Etudiants", affiche le contenu de ce tableau.





Exercice 3

Manipulation des Type complexes, Fonctions & Boucles

À vous

1

Dans le même fichier script.js, déclarer un objet **etudiantObjet**, dans lequel sera rangé les valeurs `{nom: 'John', prenom: 'DOE', age: 30}`.

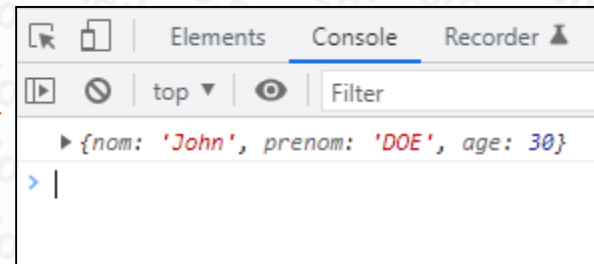
Tableau Etudiants

Objet Etudiant

Tableau d'objets Etudiants

2

En utilisant une boucle, on souhaiterait maintenant qu'un clique sur le bouton "Objet Etudiant", appelle une fonction "afficherObjet()" qui affichera le contenu de cet Objet.





Exercice 3

Manipulation des Type complexes, Fonctions & Boucles

À vous

1

Dans le même fichier script.js, déclarer un tableau d'objets **etudiants**, dans lequel sera rangé les valeurs

```
{nom: 'nom1', prenom: 'prenom1', age:21} .  
{nom: 'nom2', prenom: 'prenom2', age:22} .  
{nom: 'nom3', prenom: 'prenom3', age:23} .
```

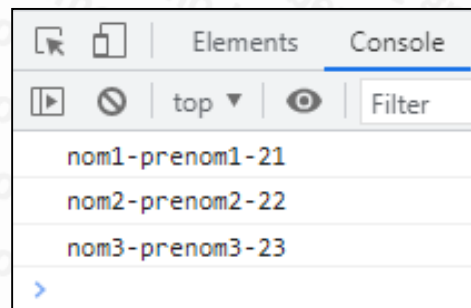
Tableau Etudiants

Objet Etudiant

Tableau d'objets Etudiants

2

En utilisant une boucle, on souhaiterait maintenant qu'un clique sur le bouton "Tableau d'objets Etudiants", appelle une fonction "afficherTableauObjets()" qui affichera le nom, le prénom et l'âge de chaque étudiant, séparés par des tirets.





Exercice 4

Les opérations et les Tests



Exercice 4 Les opérations et les Tests

À vous

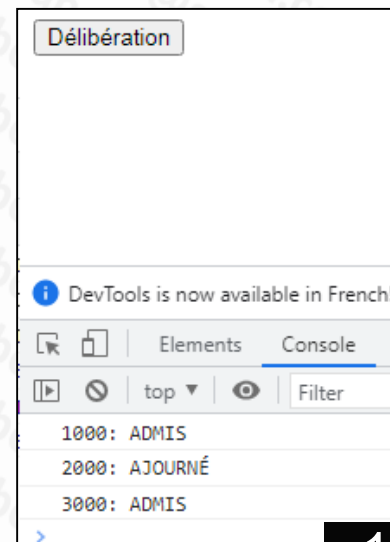
- On voudrait créer le script suivant qui :
 - Déclare un tableau "etudiants" contenant 3 objets étudiants.
 - Chaque étudiant est caractérisé par son matricule, nom, prénom, note, assiduité.
 - Le tableau contient initialement les 3 étudiants suivants:----->

(1000, JOHN, DOE, 14, 5)

(2000, BOB, CARLTON, 7, 1)

(3000, RAYANE, SMITH, 13, 3)

- On voudrait maintenant écrire une fonction **deliberation()**, qui parcourt, avec une boucle, les étudiants du tableau "etudiants", et pour chaque étudiant, elle appelle la fonction "estAdmissible(noteFinale)".
- Si "estAdmissible(noteFinale)" retourne "Vrai" alors on affiche l'étudiant « ADMIS » sinon « AJOURNÉ ».
- La fonction estAdmissible(noteFinale) est une fonction qui vérifie si la note finale est supérieure ou égale à 10.
- L'exécution du programme commence lorsqu'on clique sur un bouton « Délibération »



JavaScript



TP 3: JAVASCRIPT

1. Manipulation du langage JavaScript
2. Manipulation du DOM (à venir)
3. Utilisation du BOM (à venir)
4. Un Site Statique Interactif (à venir)