COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

Table des matières

[Chapitre 1 : LA POO 3](#_bookmark0)

1. [Notion de classe 3](#_bookmark1)
2. [Encapsulation 4](#_bookmark2)
3. [Expression de classe 5](#_bookmark3)
4. [Méthodes et propriétés statiques 5](#_bookmark4)
5. [l’héritage 6](#_bookmark5)

[Chapitre 2 : Les Objets prédéfinis 8](#_bookmark6)

1. [L’objet String 8](#_bookmark7)

[1.1- Création 8](#_bookmark8)

[1.2- Taille d’une chaine 8](#_bookmark9)

[1.3- Comparaison des caractères 8](#_bookmark10)

[1.4- Les Méthodes de l’objet String 8](#_bookmark11)

1. [L’objet Math 11](#_bookmark12)

[2.1- Les Constantes de l’objet Math 11](#_bookmark13)

[2.2- Les Méthodes de l’objet Math 12](#_bookmark14)

1. [L’objet Date 14](#_bookmark15)

[3.1- Création de date 14](#_bookmark16)

[3.2- Les getters 14](#_bookmark17)

[3.3- Les setters 14](#_bookmark18)

[Chapitre 3 : Le DOM HTML (Document Object Model) 15](#_bookmark19)

1. [Introduction 15](#_bookmark20)
2. [Récupérer des éléments du DOM 15](#_bookmark21)
3. [Accéder au contenu d’un nœud (Node) 16](#_bookmark22)

[3.1- La propriété innerHTML 16](#_bookmark23)

[3.2- La propriété innerText 16](#_bookmark24)

[3.3- La propriété textContent 16](#_bookmark25)

1. [Accéder et modifier les attributs html d’un nœud 17](#_bookmark26)
2. [Ajouter et supprimer les classes d’un nœud 17](#_bookmark27)
3. [Naviguer dans le DOM 18](#_bookmark28)

**BGRAS**

1

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

##### [Les Evénements 24](#_bookmark29)

##### [Evénement de nature click 24](#_bookmark30)

##### [Evénement de nature souris 26](#_bookmark31)

##### [Evénement de nature clavier 28](#_bookmark32)

##### [Evénement de nature focus 29](#_bookmark33)

##### [Evénements liés au formulaire 30](#_bookmark34)

##### [Evénement lié à la barre de défilement 32](#_bookmark35)

##### [Evénement lié au chargement de la page 33](#_bookmark36)

**BGRAS**

2

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

# Chapitre 1 : LA POO

## Notion de classe

##### La programmation orientée objet (POO) est un paradigme de programmation fondamental pour de nombreux langages de programmation, dont Java et C++. La programmation orientée objet consiste à modéliser un système comme une collection d'objets, où chaque objet représente un aspect particulier du système. Les objets contiennent à la fois des fonctions (ou méthodes) et des données.

##### Vous pouvez déclarer une classe à l'aide du mot- clé [**class**.](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/class) Une classe contient des propriétés et des méthodes parmi les methodes on peut retrouver le **constructeur** de la classe, une méthode qui permet de créer des objets de la classe.

class Personne {

/\* pour les propriétés pas de mot clé var ou let chaque propriété doit Obligatoirement se terminer par le ;

\*/

nom; prenom; age;

/\* constructeur de la classe personne attention pas le nom de la classe on utilise le mot clé constructor

\*/

constructor(nom, prenom, age)

{

this.nom = nom this.prenom = prenom this.age = age

}

/\* une méthode affiche ici aussi on n’utilise pas le mot clé function\*/ affiche()

{

print('Nom : ' + this.nom) print('Prenom : ' + this.prenom) print('Age : ' + this.age)

}

}

##### NB :

##### Il également possible d’initialiser une propriété avec une valeur par défaut.

##### Pour utiliser cette classe il suffit de créer des objets de cette classe en faisant appel à son constructeur via le mot clé **new**. Ainsi on pourra accéder a toutes les methodes et propriétés.

**BGRAS**

3

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

var P = new Personne('BAH', 'Tidiane', 56) P.affiche()

P.age = 80

P.affiche()

## Encapsulation

##### L’encapsulation en JavaScript est une technique qui consiste à rendre une propriété inaccessible en dehors de la classe. Les propriétés de données privées doivent être déclarées dans la déclaration de classe et leurs noms commencent par ***#***.

class Personne { #nom; #prenom; #age;

constructor(nom, prenom, age)

{

this.#nom = nom this.#prenom = prenom this.#age = age

}

affiche()

{

print('Nom : ' + this.#nom) print('Prenom : ' + this.#prenom) print('Age : ' + this.#age)

}

}

#### Remarque :

##### Vous pouvez avoir des méthodes privées ainsi que des propriétés de données privées. Tout comme les propriétés de données privées, leurs noms commencent par ***#***, et elles ne peuvent être appelées que par les propres méthodes de l'objet.

##### A l’instar des objets une classe peut aussi disposer des setters et des getters pour ces attributs encapsulés

**BGRAS**

4

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

## Expression de classe

##### Une **expression de classe** est une autre façon de définir une classe. Les expressions de classe peuvent être nommées ou non.

Personne = class { #nom;

#prenom; #age;

constructor(nom, prenom, age)

{

this.#nom = nom this.#prenom = prenom this.#age = age

}

affiche()

{

print('Nom : ' + this.#nom) print('Prenom : ' + this.#prenom) print('Age : ' + this.#age)

}

}

## [Méthodes et propriétés statiques](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes#static_methods_and_properties)

##### Le mot- clé [static définit une méthode ou une propriété statique pour une classe.](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes/static) Les membres statiques (propriétés et méthodes) sont appelés sans instancier leur classe et ne peuvent pas être appelés via une instance de classe. Les méthodes statiques sont souvent utilisées pour créer des fonctions utilitaires pour une application, tandis que les propriétés statiques sont utiles pour partager des informations entre les différents objets d’une classe. Pour préciser qu’un membre est statique il suffit de précéder sa déclaration du mot clé **static.**

**BGRAS**

5

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

Calculatrice = class

{

static somme(...array)

{

let som = 0 array.forEach(element => {

som += element

});

return som

}

static produit(...array)

{

let prod= 1 array.forEach(element => {

prod \*= element

});

return prod

}

}

print('somme(4, 5, 6, 8) = '+Calculatrice.somme(4, 5, 6, 8))

print('produit(4, 5, 6, 8) = '+Calculatrice.produit(4, 5, 6, 8))

## [l’héritage](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes#static_methods_and_properties)

##### L’héritage consiste à créer une nouvelle classe (classe dérivée) à partir d’une classe déjà existante (classe de base). La nouvelle classe héritant des propriétés et des méthodes de sa classe de base. Pour préciser qu’une classe hérite d’une autre classe on utilise le mot clé **extends.**

**BGRAS**

6

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

class Animal { constructor(couleur, poids)

{

this.poids = poids; this.couleur = couleur;

}

boire() { print('les animaux boivent tous');}

}

class Chien extends Animal { constructor(couleur, poids, nbdents)

{

super(couleur, poids); this.nbdents = nbdents;

}

aboyer() {print('seul le chien aboie');}

}

var animal = new Animal('noir', 32) animal.boire()

var milou = new Chien('blanche', 23, 24) milou.boire()

milou.aboyer()

#### Remarque :

##### Les objets de la classe de base n’ont pas accès aux membres de la classe de base encapsulée ou non.

##### Les objets de la classe dérivée ont accès seulement aux membres publiques de la classe de base mais pas aux membres privés c’est à dire encapsulés.

##### Les methodes de la classe de dérivée ont accès seulement aux membres publiques de la classe de base mais pas aux membres privés c’est à dire encapsulés.

##### Pour faire appel à une méthode de la classe de base dans une méthode de la classe dérivée on utilise aussi le mot clé super.

**BGRAS**

7

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

# Chapitre 2 : Les Objets prédéfinis

## L’objet String

##### L'Objet **String** est utilisé pour représenter et manipuler une séquence de caractères. Les chaînes sont utiles pour contenir des données qui peuvent être représentées sous forme de texte.

### 1.1- Création

##### Les chaînes peuvent être créées en tant que primitives, à partir de littéraux de chaîne ou en tant qu'objets, à l'aide du constructeur [**String()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/String) :

const string1 = "Bonjour"; const string2 = 'Bonjour';

const string3 = new String('bonjour');

const string4 = new String("bonjour");

### 1.2- Taille d’une chaine

##### La propriété **length** permet de déterminer le nombre de caractère d’une chaine de caractère.

const string1 = "Bonjour"; print (string1.length)

// affiche << 7 >>

### 1.3- Comparaison des caractères

##### Lorsque vos chaines sont construites sous forme littérale vous pouvez utiliser les opérateurs de comparaison pour comparer vos chaines. Mais lorsque vous utilisez le

constructeur dans ce cas c’est l’adresse des objets qui seront comparer. Si vous voulez comparer le contenu des objets de type String utiliser la méthode **localCompare () qui renvoie 0 si les deux objets ont même contenu. 1 si le contenu de l’objet appelant la méthode est inférieur au contenu de l’objet transmis en argument -1 sinon.**

### 1.4- Les Méthodes de l’objet String

##### o La méthode **charAt ()** : renvoie un caractère de rang donné sachant que le premier caractère a pour index 0.

const string1 = "Bonjour"; print(string1.charAt(3))

// affiche << j >>

##### o La méthode **charCodeAt ()** : renvoie le code ASCII d’un caractère de rang donné sachant que le premier caractère a pour index 0.

**BGRAS**

8

const string1 = "Bonjour"; print(string1.charCodeAt(0))

// affiche << 66 >>

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

##### o La méthode **concat ()** : permet de concaténer une série de chaine de caractère et renvoie le résultat de la concaténation.

const string1 = "Bonjour" const string2 = " Monsieur"

print(string1.concat(string2, ' Diallo'))

// affiche << Bonjour Monsieur Diallo >>

##### o La méthode **endsWith ()** : renvoie true si une chaine de caractère se termine par une sous chaine false sinon.

const string1 = "Bonjour" print(string1.endsWith('jour'))

// affiche << true >>

##### o La méthode **includes ()** : renvoie true si une sous chaine se trouve dans une chaine de caractère false sinon.

const string1 = "Bonjour" print(string1.includes('jour'))

// affiche << true >>

const string1 = "Bonjour"

print (string1.includes('jour', 4))

// affiche << false >>

##### La méthode **indexOf ()** : renvoie la position de la première occurrence d’une sous chaine dans une chaine de caractère. Si la sous chaine n’existe pas -1 sera renvoyer

const string1 = "Bonjour" print(string1.indexOf('jour'))

// affiche << 3 >>

const string1 = "Bonjour" print(string1.indexOf('jour', 4))

// affiche << -1 >>

##### **NB :** pour la position de la dernière occurrence on utilisera la méthode **lastIndexOf**

##### La méthode **repeat ()** : crée une nouvelle chaine de caractère en répétant plusieurs fois une chaine de caractère.

const string1 = "Bonjour " print(string1.repeat(3))

// affiche << Bonjour Bonjour Bonjour >>

##### La méthode **replace ()** : crée une nouvelle chaine de caractère en remplaçant la première occurrence d’une sous chaine par une autre dans une chaine de caractère.

9

**BGRAS**

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

const string1 = "Bonjour " print(string1.replace('jour', 'soir'))

// affiche << Bonsoir >>

**NB :** utiliser la méthode **replaceAll ()** pour remplacer toutes les occurrences.

##### La méthode **slice ()** : extrait une sous chaine dans une chaine en spécifiant une position de debut et de fin(exclus). Si aucune position n’est fournie toute la chaine est extraite.

const string1 = "Bonjour" print(string1.slice())

// affiche << Bonjour >>

const string1 = "Bonjour" print(string1.slice(3))

// affiche << jour >>

const string1 = "Bonjour" print(string1.slice(3, 7))

// affiche << jour >>

##### La méthode **split ()** : découpe une chaine de caractère en un tableau de chaine en utilisant un séparateur.

const string1 = "Bon jour soir nuit" var array = string1.split(' ')

// crée le tableau suivant : ['bon', 'jour', 'soir', 'nuit']

##### La méthode **startsWith ()** : renvoie true si une chaine de caractère commence par une sous chaine false sinon.

const string1 = "Bonjour" print(string1.startsWith('Bon'))

// affiche : << true >>

##### La méthode **substring ()** : extrait une sous chaine dans une chaine en spécifiant une

##### position de debut et de fin(exclus). Si aucune position n’est fournie toute la chaine est extraite.

const string1 = "Bonjour" print(string1.substring())

// affiche << Bonjour >>

const string1 = "Bonjour" print(string1.substring(3))

// affiche << jour >>

**BGRAS**

10

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

const string1 = "Bonjour" print(string1.substring(3, 7))

// affiche << jour >>

* La méthode **toLowerCase ()** : renvoie une chaine de caractère en minuscule

const string1 = "BonJOur" print(string1.toLowerCase())

// affiche : << bonjour >>

* La méthode **toUpperCase ()** : renvoie une chaine de caractère en majuscule

const string1 = "BonJOur" print(string1.toUpperCase())

// affiche : << BONJOUR >>

* Les méthode **trim (), trimStart () et trimEnd ()** : éliminent respectivement les espaces de début et fin de chaine, les espaces de début de chaine et les espaces de fin de chaine.

##### Les méthode **valueOf () :** convertie une chaine de caractère Object en une chaine de caractère littérale

var s1 = new String('bonjour') var s2 = new String('bonjour') print(s1 === s2)

// affiche << false >> print(s1.valueOf() === s2.valueOf())

// affiche <<true>>

##### **Remarque** : en JS il existe une fonction très spéciale qui évalue une chaine de caractère littérale transmise en argument comme du code source il s’agit de la fonction **val.**

var chaine = '5 \* 5' print(eval(chaine))

// affiche << 25 >>

## L’objet Math

##### **Math** est un objet intégré qui possède des propriétés et des méthodes static pour les constantes et les fonctions mathématiques. Ce n'est pas un objet fonction. **Math** fonctionne avec le type [**Number**.](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Number)

### 2.1- Les Constantes de l’objet Math

##### [**Math.PI**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/PI) : valeur de du symbole mathématiques π

* + [**Math.SQRT1\_2**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/SQRT1_2) : Racine carrée de ½ ; environ 0.707.
  + [**Math.SQRT2**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/SQRT2) : Racine carrée de 2; environ 1.414.

**BGRAS**

11

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

### 2.2- Les Méthodes de l’objet Math

o [**Math.abs(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/abs) **:** Renvoie la valeur absolue de x.

print(Math.abs(-5))

// affiche << 5 >>

o [**Math.cbrt(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie la racine cubique de x.

print(Math.cbrt(9))

// affiche << 2.080083823051904 >>

##### o [**Math.ceil(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie le plus petit entier supérieur ou égal à x.

print(Math.ceil(4.67))

// affiche << 5 >>

o [**Math.floor(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie le plus grand entier inférieur ou égal à x.

print(Math.floor(4.67))

// affiche << 4 >>

##### o [**Math.hypot()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie la racine carrée de la somme des carrés de ses arguments.

print(Math.hypot(3, 4))

// affiche << 5 >>

##### o [**Math.max()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : renvoie le plus grand des nombres donnés comme paramètres d'entrée, ou **-** [**Infinity**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Infinity) s'il n'y a pas de paramètres.

print(Math.max(3, 4, 12))

// affiche << 12 >> print(Math.max(...[3, 4, 12, 78, 65]))

// affiche << 78 >>

##### o [**Math.min()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : renvoie le plus petit des nombres donnés comme paramètres d'entrée, ou **+** [**Infinity**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Infinity) s'il n'y a pas de paramètres.

print(Math.min(3, 4, 12))

// affiche << 3 >>

print(Math.min(...[1, 3, 4, 12, 78, -65]))

// affiche << -65 >>

**BGRAS**

12

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

o [**Math.pow(x, y)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie la base x à la puissance de l'exposant y.

print(Math.pow(3, 2))

// affiche << 9 >>

* [**Math.random()**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie un nombre pseudo-aléatoire entre 0 et 1.

##### NB :

##### Math.floor(Math.random() \* max) ; // renvoie un nombre aléatoire entre 0 et max.

##### Math.floor(Math.random() \* (max - min) + min) ;

##### // renvoie un nombre aléatoire entre min et max

* [**Math.round(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie l’entier le proche de x

print(Math.round(2.65))

// affiche << 3 >>

print(Math.round(2.25))

// affiche << 2 >>

##### .

##### [**Math.sign(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie le signe de x (1 s’il est positif et -1 s’il est négatif)

print(Math.sign(-2.65))

// affiche << -1 >>

* [**Math.sqrt(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie la racine carré de x

print(Math.sqrt(9))

// affiche << 3 >>

* [**Math.trunc(x)**](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math/cbrt) : Renvoie la partie entière de x

print(Math.trunc(9.987766))

// affiche << 9 >>

**BGRAS**

13

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

## L’objet Date

##### Une date JavaScript est fondamentalement le nombre de millisecondes qui se sont écoulées depuis l' [époque ECMAScript](https://tc39.es/ecma262/multipage/numbers-and-dates.html#sec-time-values-and-time-range) , qui est définie comme le 1er janvier 1970.

### 3.1- Création de date

var d = Date() //renvoie la date et l'heure actuelle

var d = Date.now() //nombre de seconde écouler depuis 1-jan-1970 var d = Date.parse('MM JJ AAAA')

//retourne le nombre de seconde écouler depuis 1-jan-1970 jusqu'a cette date var d = new Date() //renvoie la date et l'heure actuelle

var d = new Date('MONTH JJ, AAAA') var d = new Date(millisecondes)

var d = new Date(AAAA, MM [, JJ, HH, MN, SS, MS]) //MM(commence par 0), var d = new Date("AAAA-MM-JJ") //MM(commence par 1)

var d = new Date("AAAA-MM-JJ HH:MN:SS")

var d = new Date("MM/JJ/AAAA")

var d = new Date("MONTH JJ AAAA HH:MN:SS")

var d = new Date("DAY MON JJ AAAA HH:MN:SS GMT+0n")

### 3.2- Les getters

print(d.getFullYear()) //l'année print(d.getMonth()) //le mois commence par 0 print(d.getDate()) //numéro de jour du mois

print(d.getHours()) // l'heure print(d.getMinutes()) // la minute print(d.getSeconds()) // les secondes

print(d.getMilliseconds()) // les millisecondes

### 3.3- Les setters

d.setFullYear(AAAA [, MM, JJ]) //modifie l'année, le mois et le jour d.setDate(JJ) //modifie le numéro de jour du mois JJ > 0 d.setMonth(mm) //modifie le mois m >= 0

d.setHours(hh) // modifie l'heure d.setMinutes(mm) // modifie la minute d.setSeconds(ss) // modifie les secondes

d.setMilliseconds(ms) // modifie les millisecondes

**BGRAS**

14

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

**Chapitre 3 : Le DOM HTML (**Document Object Model)

## Introduction

##### Le **Document Object Model** (DOM) est la représentation des données des objets qui composent la structure et le contenu d'un document sur le Web. Le DOM est une interface de programmation pour les documents Web. Il représente la page afin que les programmes puissent modifier la structure, le style et le contenu du document. Le DOM représente le document sous forme de nœuds et d'objets ; il permet aux langages de programmation d’interagir avec la page.

## Récupérer des éléments du DOM

##### Il existe plusieurs méthodes permettant de récupéré des éléments dans le DOM :

* **document.getElementById(‘id’)** : c’est la méthode la plus utilisée elle permet de récupérer un élément HTML à partir de son identifiant (attribut id).
* **document.getElementsByName(‘name’) :** elle permet de récupérer dans un tableau des éléments HTML à partir de leur nom (attribut name).
* **document.getElementsByClassName(‘class’) :** elle permet de récupérer dans un tableau des éléments HTML à partir de leur classe (attribut class).

##### **document.getElementsByTagName(‘balise’) :** elle permet de récupérer dans un tableau des éléments HTML à partir de leur nom de balise. Cette méthode n’est pratique qui si l’on veut connaitre le nombre de balise de type à partir d’un nœud.

#### Remarque :

##### On peut aussi utiliser **querySelector ()**, qui renvoie le premier élément trouvé correspondant au sélecteur CSS spécifié, ou **querySelectorAll ()**, qui renvoie tous les éléments (sous forme de tableau) correspondant au sélecteur CSS spécifié entre parenthèses.

**BGRAS**

15

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

**Rappel sur quelques sélecteurs CSS**

##### \* : cible tous les éléments HTML

##### Balise1 > balise2 : cible les balise2 qui sont dans enfants direct de balise1.

##### Balise1 balise2 : cible les balise2 qui sont enfants direct ou indirect de balise1.

##### Balise1.maclasse : cible les balise1 qui ont une classe nommée maclasse.

##### Balise1#monid : cible la balise1 qui a un identifiant nommé monid.

##### Balise1, balise2 : cible les balise1 et balise2

##### Balise1 + balise2 : cible les balise2 qui sont immédiatement précédées de balise1.

## Accéder au contenu d’un nœud (Node)

##### Il existe trois propriétés JS du DOM HTML permet d’accéder et de modifier le contenu d’un nœud :

### 3.1- La propriété innerHTML :

##### Cette propriété renvoie tout le contenu d’un nœud c’est-à-dire ce qui se trouve entre la

##### balise d’ouverture et fermeture sous forme de texte.

### 3.2- La propriété innerText :

##### Cette propriété renvoie seulement le texte contenu dans un nœud.

### 3.3- La propriété textContent :

##### Identique à **innerText** mais compatible a tous les navigateurs.

<body>

<div>

<p id="matin" class = "moi">bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.getElementById('matin') console.log(p.innerHTML+'\n'+p.innerText+'\n'+p.textContent)

</script>

</body>

**BGRAS**

16

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

## Accéder et modifier les attributs html d’un nœud

On peut manipuler sur les attributs d'une balise HTML avec les méthodes **getAttribute ()** et **setAttribute ()**

<body>

<div>

<p id="matin" class = "moi">bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.getElementById('matin')

// console.log(p.getAttribute('id')) ou encore console.log(p.id)

p.setAttribute('name', 'midi')

</script>

</body>

## Ajouter et supprimer les classes d’un nœud

On peut ajouter un supprimer les classes d'un nœud avec les méthodes **classList.add ()** et **classList.remove().** La méthode **classList.toggle()** vérifie si la classe existe il la supprime sinon il l’ajoute

<body>

<div id="container">

<div id="items">

<span class="item supprime">pause</span>

<span class="item">play</span>

<span class="item">muet</span>

</div>

</div>

<script>

const container = document.getElementById('container') container.style.background = "green"

const items = document.getElementById('items') items.classList.add('ajout') items.firstElementChild.classList.remove('supprime') items.lastElementChild.classList.toggle('ajoute\_supprime')

</script>

</body>

**BGRAS**

17

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

## Naviguer dans le DOM

##### La propriété **parentNode** permet d'accéder à l'élément parent d'un élément

<body>

<div>

<p id="matin" class = "moi">bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.getElementsByTagName('mark') console.log(p[0].parentNode.innerText)

</script>

</body>

##### Les propriété **nodeName** renvoie le nom d'un nœud :

<body>

<div>

<p id="matin" class = "moi" >bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.getElementById('matin') console.log('Nom : '+p.nodeName)

</script>

</body>

**BGRAS**

18

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

* **firstChild** et **lastChild** permettent d'accéder au premier et au dernier élément d'un nœud.

<body>

<div>

<p id="matin" class = "moi" >bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.querySelector('div') console.log('Premier enfant : '+p.firstChild.innerHTML) console.log('Dernier enfant : '+p.lastChild.innerHTML)

</script>

</body>

***ATTENTION*** : Avec **firstChild** et **lastChild** les espaces entre les balises sont considérés comme des nœuds. Pour éviter cela utilisez plutôt les méthodes **firstElementChild** et **lastElementChild**

<body>

<div>

<p id="matin" class = "moi" >bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.querySelector('div')

console.log('Premier enfant : '+p.firstElementChild.innerHTML) console.log('Dernier enfant : '+p.lastElementChild.innerHTML)

</script>

</body>

**BGRAS**

19

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

##### **childNodes** retourne un tableau contenant la liste des enfants d'un élément. Les espaces entre les balises sont considérés comme des nœuds.

<body>

<div>

<p id="matin" class = "moi" >bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.querySelector('div') var array = p.childNodes

console.log('Nombre d\'enfants : '+p.childElementCount) // affiche 4 console.log('Nombre d\'enfants : '+array.length) // afiche 9

</script>

</body>

**nextSibling** et **previousSibling** permettent d'accéder à l'élément suivant, et au précédent . Les espaces entre les balises sont considérés comme des nœuds. Pour éviter cela utilisez plutôt les méthodes **nextElementSibling** et **previousElementSibling.**

<body>

<h1>Naviguation DOM</h1>

<div>

<p id="matin" class = "moi" >bonjour <mark>monsieur</mark></p>

<p name = "soir">bonsoir</p>

<p name = "soir">bonne nuit</p>

<span>

<p>aurevoir</p>

</span>

</div>

<script>

var p = document.querySelector('div') console.log('Element suivant : '+p.nextSibling.innerHTML)

console.log('Element precedent : '+p.previousSibling.innerHTML) console.log('Element suivant : '+p.nextElementSibling.innerHTML) console.log('Element precedent : '+p.previousElementSibling.innerHTML)

</script>

</body>

**BGRAS**

20

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

#### Créer et insérer des éléments

##### Avec le DOM, l'ajout d'un élément se fait en trois temps :

* + - On crée l'élément avec la méthode **createElement('type\_element')**
    - On lui affecte des attributs avec la méthode **setAttribute (attribut, valeur)**

##### On l'insère dans un nœud avec la méthode **appendChild (nœud).** L’élément inséré sera le dernier enfant du nœud ou il a été insérer.

<script>

var p = document.querySelector('div') var h1 = document.createElement('h1')

// creer un noeud de simpble type texte

var textedeh1 = document.createTextNode('NAVIGUER DANS LE DOM')

// ajouter le texte dans le nouveau noeud h1.appendChild(textedeh1)

// ajout d'attribut au nouveau noeud h1.setAttribute('style', 'color : red')

// ou encore h1.style.backgroundColor = 'blue' h1.style.fontStyle = 'italic' h1.style.textAlign = 'center'

// ajout du nouveau noeud dans son noeud parent p.appendChild(h1)

</script>

#### Cloner, remplacer et supprimer

##### Pour cloner un élément, on utilise **cloneNode** (), et on choisit avec (true) ou sans (false) ses enfants et ses attributs.

##### Pour remplacer un élément par un autre, on utilise **replaceChild** (), avec deux paramètres, le nouvel élément et l'élément qu'on veut remplacer.

##### Pour supprimer un élément, on utilise **removeChild** (), avec le nœud enfant à retirer.

**BGRAS**

21

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

<body>

<h1>LE DOM <mark>HTML</mark></h1>

<div id="test">

<p>Un peu de texte <a href="#">lien HTML</a></p>

<p class="myclasse"> C'est genial le dom </p>

</div>

<script>

/ remplacer un noeud

var link = document.getElementsByTagName('a')[0]; var newLabel= document.createTextNode('Lien JS'); link.replaceChild(newLabel, link.firstChild);

//cloner un noeud

var titre = document.getElementsByTagName('h1')[0]

var clone = titre.cloneNode(true) //cloner avec les enfants var madiv = document.getElementById('test') madiv.appendChild(clone)

//supprimer

var para = madiv.getElementsByClassName('myclasse')[0]

// ou encore

var para = madiv.lastElementChild.previousElementSibling madiv.removeChild(para)

</script>

#### Vérifier la présence d’un élément

##### Pour vérifier la présence d'éléments enfant, on utilise **hasChildNodes** ().

#### Insérer un élément avant un autre

##### Pour insérer un élément avant un autre, on utilise **insertBefore** ().

**BGRAS**

22

<body>

<h1>LE DOM <mark>HTML</mark></h1>

<div id="test">

<p>Un peu de texte <a href="#">lien HTML</a></p>

<p class="myclasse"> C'est genial le dom </p>

</div>

<script>

//verifier l'existance d'un element

var madiv = document.getElementById('test')

var para = madiv.getElementsByClassName('myclasse')[0] console.log(para.hasChildNodes() ? 'Le paragraphe existe' : 'le paragraphe

n\'existe pas')

//inserer avant le paragraphe

var titre = document.getElementsByTagName('h1')[0] var clone = titre.cloneNode(true) madiv.insertBefore(clone, para)

</script>

</body>

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

**Exercice 1.**

Passez ce code HTML en script :

<div id="divTP1">

Le <strong>World Wide Web Consortium</strong>, abrégé par le sigle

<strong>W3C</strong>, est un

<a href="<http://fr.wikipedia.org/wiki/Organisme_de_normalisation>" title="Organisme de

normalisation">organisme de standardisation</a> à but non-lucratif chargé de promouvoir la compatibilité des technologies du

<a href="<http://fr.wikipedia.org/wiki/World_Wide_Web>" title="World Wide Web">World Wide Web</a>.

</div>

**Exercice 2.**

Passez le code HTML en script, en utilisant une boucle for :

<div id="divTP2">

<p>Langages basés sur ECMAScript :</p>

<ul>

<li>JavaScript</li>

<li>JScript</li>

<li>ActionScript</li>

<li>EX4</li>

</ul>

</div>

**Exercice 3.**

Passez le code HTML en script :

<div id="divTP4">

<form enctype="multipart/form-data" method="post" action="upload.php">

<fieldset>

<legend>Uploader une image</legend>

<div style="text-align: center">

<label for="inputUpload">Image à uploader :</label>

<input type="file" name="inputUpload" id="inputUpload" />

<br /><br />

<input type="submit" value="Envoyer" />

</div>

</fieldset>

</form>

</div>

**BGRAS**

23

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

1. Les Evénements

##### En JS il existe plusieurs type d’événements classer par nature :

* 1. Evénement de nature click

##### click : se déclenche lorsqu’on clique sur un élément

<body>

<div id="divTP4">

<form enctype="multipart/form-data" method="post" action="#">

<fieldset>

<legend>Uploader une image</legend>

<div style="text-align: center">

<label for="inputUploadrrr" onclick="console.log('clic sur le label')"> Image à uploader :

</label>

<input type="file" name="inputUpload" id="inputUpload" />

<br /><br />

<input type="submit" value="Envoyer"/>

</div>

</fieldset>

</form>

</div>

</body>

##### NB : il existe plusieurs moyens pour relier un événement a un élément :

##### La première consiste à faire appel à l’attribut html **onevenement** par exemple **onclick** et lui passer comme valeur le code JS à exécuter ou la fonction JS à exécuter c’est ce nous avons utilisé dans l’exemple précédent.

##### La deuxième solution consiste à récupérer l’objet dans le DOM et utiliser la propriété JS **onevenement** de l’objet puis lui passer une fonction anonyme ou fléchée à exécuter.

**BGRAS**

24

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

<div id="divTP4">

<form enctype="multipart/form-data" method="post" action="#">

<fieldset>

<legend>Uploader une image</legend>

<div style="text-align: center">

<label for="inputUploadrrr">Image à uploader :</label>

<input type="file" name="inputUpload" id="inputUpload" />

<br /><br />

<input type="submit" value="Envoyer"/>

</div>

</fieldset>

</form>

</div>

<script>

var label = document.querySelector('fieldset > div').firstElementChild label.onclick = () =>{console.log('click sur le label')}

</script>

</body>

##### La troisième solution qui est la plus répandue consiste à récupérer l’objet dans le DOM et lui ajouter un gestionnaire d’événement via la méthode **addEventListener** qui reçoit deux arguments le premier est le nom de l’événement sans le mot **on** et le second la fonction anonyme ou fléchée à exécuter.

<body>

<div id="divTP4">

<form enctype="multipart/form-data" method="post" action="#">

<fieldset>

<legend>Uploader une image</legend>

<div style="text-align: center">

<label for="inputUploadrrr"> Image à uploader :</label>

<input type="file" name="inputUpload" id="inputUpload" />

<br /><br />

<input type="submit" value="Envoyer"/>

</div>

</fieldset>

</form>

</div>

<script>

var label = document.querySelector('fieldset > div').firstElementChild label.addEventListener('click',(e) =>{

e.target.style.color = "red"; e.target.style.fontSize = "2rem"

})

</script>

</body>

**BGRAS**

25

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

* 1. Evénement de nature souris

##### mouseover : Faire entrer le curseur sur l'élément

##### mouseout : Faire sortir le curseur de l'élément

<body>

<div id="mouse">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore fugiat, sit doloribus molestias error consequuntur totam iusto iure laudantium, Lorem ipsum dolor sit, amet consectetur adipisicing elit. Recusandae odio soluta vero at

</div>

<script>

var div = document.querySelector('#mouse') div.style.fontSize = '2rem'

div.addEventListener('mouseover',() =>{div.style.background = 'red'}) div.addEventListener('mouseout',() =>{div.style.background = 'blue'})

</script>

</body>

##### **mousedown :** Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément .

##### **mouseup :** Relâcher le bouton gauche de la souris sur l'élément

<div id="mouse">

Lorem ipsum dolor sit amet consectetur adipisicing elit. Tempore fugiat, sit doloribus molestias error consequuntur totam iusto iure laudantium, Lorem ipsum dolor sit, amet consectetur adipisicing elit.

</div>

<script>

var div = document.querySelector('#mouse') div.style.fontSize = '2rem'

div.addEventListener('mousedown',() =>{div.style.background = 'red'}) div.addEventListener('mouseup',() =>{div.style.background = 'blue'})

</script>

</body>

* + - **mousemove :** Faire déplacer le curseur sur l'élément

**BGRAS**

26

<style type="text/css">

.mouse

{

width: 100%;

background-color : rgb(241, 8, 8); border: 2px solid;

}

</style>

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

<body>

<div id="mouse"> Bonjour

</div>

<script>

var div = document.querySelector('#mouse') div.style.fontSize = '2rem'

div.addEventListener('mousemove',() =>{div.classList.add('mouse')})

</script>

</body>

##### **Remarque :** il est possible de passer un argument de type Event a notre fonction anonyme ou fléchée pour accéder à l’ensemble des propriétés de l’événement tel que les coordonnées de la souris l’élément sur lequel l’événement est réaliser etc….

<style type="text/css">

.mouse

{

width: 100%;

background-color : rgb(241, 8, 8); border: 2px solid;

}

#ico{

width: 10px; position: absolute; top: -20px;

}

</style>

<body>

<div id="mouse"> Bonjour

<img src="./ico.png" alt="" id="ico">

</div>

<script>

var div = document.querySelector('#mouse') var icone = document.getElementById('ico') div.style.fontSize = '2rem' window.addEventListener('mouseover',(event) =>

{

div.classList.add('mouse') icone.style.top = event.pageY+'px' icone.style.left = event.pageX+'px'

})

</script>

</body>

**BGRAS**

27

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

* 1. Evénement de nature clavier

##### **keydown** Appuyer (sans relâcher) sur une touche de clavier sur l'élément

##### **keyup** Relâcher une touche de clavier sur l'élément

<style type="text/css">

.mouse

{

width: 100%;

background-color : rgb(241, 8, 8); border: 2px solid;

}

</style>

<body>

<div id="mouse"> Bonjour

</div>

<script>

var div = document.querySelector('#mouse') div.style.fontSize = '2rem' window.addEventListener('keydown',(event) =>

{

div.classList.add('mouse')

div.innerHTML += " vous avez appuyez sur la lettre " + event.key div.innerHTML += " <br> son code ascii est " + event.keyCode

})

window.addEventListener('keyup',(event) =>

{

div.classList.remove('mouse') div.innerHTML = "Bonjour"

})

</script>

</body>

**BGRAS**

28

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

##### **keypress** Frapper (appuyer puis relâcher) une touche de clavier sur l'élément

<body>

<div id="mouse"> Bonjour

</div>

<script>

var div = document.querySelector('#mouse') div.style.fontSize = '2rem' window.addEventListener('keypress',(event) =>

{

div.style.width = "100%"

div.style.backgroundColor = "rgb(241, 8, 8)" div.style.border = "2px solid"

div.innerHTML += " vous avez appuyez sur la lettre " + event.key div.innerHTML += " <br> son code ascii est " + event.keyCode

})

</script>

</body>

* 1. Evénement de nature focus
* **focus** : « Cibler » l'élément

##### **blur** : Annuler le « ciblage » de l'élément

**BGRAS**

29

<body>

<div id="mouse" style="width: 500px;">

<form action="" method="post">

<input type="text" size = "20" id="txt1"><label id="start">\*</label>

</form>

</div>

<script>

var texte = document.querySelector('#txt1') var start = document.getElementById('start') texte.style.outline = "0px" texte.style.borerRadius = "3px" start.style.visibility = "hidden" texte.addEventListener('focus',() =>

{texte.style.border = "2px solid blue" start.style.visibility = "hidden"}) texte.addEventListener('blur',() =>

{if (texte.value === ''){ texte.style.border = "2px solid red"

texte.placeholder = 'Ce champ est obligatoire' start.style.visibility = "visible" start.style.color = "red"}})

</script>

</body>

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

* 1. Evénements liés au formulaire

##### **change** : Changer la valeur d'un élément spécifique aux formulaires (input, checkbox, etc.)

##### **input** : idem que change mais facile

##### **select** : Sélectionner le contenu d'un champ de texte (input, textarea, etc.)

##### Il existe une méthode du DOM **select ()** qui permet de sélectionner tout le contenu de l’élément.

###### Exemple 1 : afficher le genre sélectionner

<div id="mouse" style="width: 500px;">

<form action="#" method="post">

<fieldset>

<legend>Genre</legend>

<input type="radio" name="sexe" id="m" value="Maculin">

<label for="m">Masculin</label>

<input type="radio" name="sexe" id="f" value="Feminin">

<label for="f">Feminin</label>

</fieldset>

</form>

</div>

<div id="out"></div>

<script>

var out = document.getElementById('out')

// m.addEventListener('change',()=>{

// if (m.checked) out.innerHTML = "vous etes un garçon"

// })

// f.addEventListener('change',()=>{

// if (f.checked) out.innerHTML = "vous etes une fille"

// })

// ou encore m.addEventListener('input', (e)=>

{

out.innerHTML = 'Vous ete de sexe '+ e.target.value

})

f.addEventListener('input', (e)=>

{

out.innerHTML = 'Vous ete de sexe '+ e.target.value

})

</script>

###### Exemple 2 : afficher le texte sélectionner dans la zone de saisie

**BGRAS**

30

<div id="mouse" style="width: 500px;">

<form action="#" method="post"> <input type="text" size="=50" id ="text">

</form> </div> <div id="out"></div>

<script>

var texte = document.getElementById('text') var out = document.getElementById('out')

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

texte.addEventListener('select', ()=>

{ if (texte.selectionStart !== undefined)

{

let startPos = texte.selectionStart ; let endPos = texte.selectionEnd let selectedtext = texte.value.substring(startPos, endPos); out.innerHTML = selectedtext

}

})

</script>

###### Exemple 3 : afficher le pays sélectionner dans la zone de liste

<body>

<div id="mouse" style="width: 500px;">

<form action="#" method="post">

<select name="" id="pays">

<option value="Guinee">Guinee</option>

<option value="Mali">Mali</option>

<option value="Senegal">Senegal</option>

</select>

</form>

</div>

<div id="out"></div>

<script>

var pays = document.getElementById('pays') ; var out = document.getElementById('out')

/\* pays.addEventListener('change', ()=>

{

out.innerHTML = "Votre Pays est : "+pays.options[pays.selectedIndex].value

}) \*/ Ou encore

pays.addEventListener('input', (e)=>

{

out.innerHTML = 'Votre Pays est : '+e.target.value

})

</script>

</body>

##### **submit** : Envoyer le formulaire correspond à la méthode submit() (balise <form>)

##### **reset** : Réinitialiser le formulaire correspond à la méthode reset() (balise <form>)

**BGRAS**

31

<body>

<form id="form1" action="cible.php" method="post">

<input type="text" placeholder="Entrez un texte" /><br /><br />

<input type="submit" value="Envoyer" />

<input type="reset" value="Annuler" />

</form>

<script>

var myForm = document.getElementById('form1');

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

myForm.addEventListener('submit', function(e)

{ e.preventDefault(); // pour bloquer le formulaire

//form1.reset() pour reinitialiser automatiquement le formulaire

});

myForm.addEventListener('reset', function(e) { alert('Vous avez réinitialisé le formulaire !');

});

</script>

</body>

* 1. Evénement lié à la barre de défilement

##### **scroll** : c’est un évènement de l’objet **window** se déclenche lorsque vous faite défiler la barre de défilement.

<style>

\*{margin: 0; padding: 0; outline: 0;} body{height: 200vh;}

.navbar

{

background: rgb(172, 164, 172); position: fixed; width: 100%; display: none; font-size: 2rem;

}

.menu { display: flex; justify-content: space-evenly;} li{list-style-type: none; cursor: pointer;}

</style>

<body>

<nav class="navbar">

<ul class="menu">

<li>Acceuil</li><li>Services</li>

<li>Contacts</li><li>Actualites</li>

<li>Forum</li>

</ul>

</nav>

<script>

var navbar = document.querySelector('.navbar') window.addEventListener('scroll', ()=>

{

(window.scrollY > 200) ? navbar.style.display = "block"

: navbar.style.display = "none"

})

</script>

</body>

**BGRAS**

32

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

* 1. Evénement lié au chargement de la page

##### **load** : c’est un évènement de l’objet **window** se déclenche lorsque le chargement de la page se termine.

<body>

<div>

<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quae fuga at quis maxime ut ratione perspiciatis, tempora ad eveniet voluptatum illo officia modi animi placeat iste molestias sit nemo enim.</p>

</div>

<script>

window.alert('chargement terminer !')

</script>

</body>

**BGRAS**

33

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

**Travaux Pratiques**

**TP1 :**

Écrire une page HTML 5 telle que celle-ci :

contienne un élément de type <div> tel que :

cet élément contienne du texte et des sous éléments de type <strong>, cet élément ait un attribut id="ma\_div",

cet élément ait un attribut class="ma\_classe",

affiche par un script javascript l'ensemble des attributs et des enfants de cette div dans la page HTML.

**TP2 :**

Écrire une page HTML 5 telle que celle ci verifie dès son ouverture si le navigateur ne supporte ni les placeholders ni l'autofocus dans les champs d'entrée (input) des formulaires (form).

**TP3 :**

1. Créer un document html contenant des champs de type range et un titre dans un tableau html
2. Créer le script qui permet de changer les propriétés (background,padding,height,width,arrondi,rotation) du titre en milieu en changeant

leurs valeurs via les inputs de type range.

**TP4 :**

soit le document html suivant :

<body>

<p id="parag1">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo onsequat. </p>

<button onclick="changer\_style()">Changer Style</button>

</body>

Ecrire la fonction changer\_style qui permet de styler le paragraphe au clic du bouton par :

une couleur blanche. un background noir.

une bordure noire pointillée de 1px. un retrait de 5px.

Définir les propriétés précédentes dans une classe "active" et modifier la fonction changer\_style de telle façon qu'elle ajoute la classe "active" au

paragraphe.

**BGRAS**

34

COURS DE JAVASCRIPT 2021-2022 LICENCE 2 MIAGE/GENIE INFORMATIQUE

**Projet 1 :** créer une calculatrice avec les opérations de base (+, -, \*, /). Ensuite ajouter les fonctionnalités ci-dessous :

* C : (clear pour nettoyer)

o = : pour afficher les résultats

o +/- : pour la valeur absolue

* % : pour calculer le pourcentage
* , : pour le traitement des nombres décimaux
* MS : (Mémoire sauvegarde)
* MR : (Mémoire retour)
* MC :( Mémoire clear)

**Projet 2 :** créer un menu dynamique flexible.

**Projet 3 :** créer un carrousel.

**Projet 4 :** créer un chronomètre avec les boutons 'sart', 'stop' et 'reset'

**Projet 5 :** créer une horloge (hh, mm, ss)

**BGRAS**

35