Exercices : javascript

**Exo 0 : de l'art avec Javascript (analyse de code)**

On donne en annexe le code d'une page html permettant de dessiner sur la page. (fichier **exo\_0\_Js.html** ).

1. Après avoir testé la page :

* Dans le code, entourer/surligner en vert les instructions créant les composants graphiques permettant d'interagir avec la page
* Dans le code, entourer/surligner en rouge les instructions permettant de gérer le comportement de la page.

1. L'interaction est-elle gérée au niveau du navigateur ou du serveur (justifier) ?
2. Souligner en noir , les instructions permettant d'accéder à un élément de la page web

En bleu, celles déclenchant une action suite à un évènement (vous donnerez le nom et la signification des évènements en question)

1. Compléter le code afin de permettre de dessiner en vert et en bleu.
2. Modifier le code afin que l'on dessine une succession de points au lieu de cercle.

**Exo 1 : Conversion degré -> radian (\*)**

Écrire le code permettant d'entrer un angle en degré et lorsque l'on appuie sur un bouton intitulé convertir, cela affiche l'angle correspondant en radian.

Rq : pour le bouton, utiliser une balise <input type='button'> ou <button type='button>..</button> afin de ne pas soumettre le formulaire après clic.

**Exo 2 : vérification de données de formlaire** (\*)

1. Écrire le code HTML et JavaScript permettant à un utilisateur de rentrer son âge via un formulaire, et le script js doit vérifier que l'âge est bien compris entre 18 et 120 ans lorsque l'on quitte l'élément (évènement **onblur** correspondant à la perte de focus). Le code enverra un message d'erreur (via l'instruction alert) pour un âge au-dessus de 120, et un message indiquant qu'il faut être adulte pour un âge inférieur à 18 ans.
2. De la même manière, ajouter le code permettant de donner un mot de passe, en vérifiant que le mot de passe soit de au moins 8 caractères.

**Ex 3 : modification de programme (\*) (source lycée la Martinière)**

Modifier le programme **exo\_3\_JS.html**, de manière à :

* Lorsque l'on survole le paragraphe : modifier le texte (cf commentaire), mettre le fond en jaune.
* Lorsque l'on ne survole plus : remettre les paramètres initiaux

**Ex 4 : zoom (\*)**

Écrire le code permettant de faire un zoom sur une image (celle que vous voulez) de taille initiale (100px×100px) lorsqu'on la survole (on agrandira l'image à la taille 300par 300px). Elle doit retrouver sa taille originale lorsque le pointeur ne la survole plus.

1. Avec la méthode que vous voulez.
2. En utilisant une fonction admettant en arguments la largeur et la hauteur à imposer à l'image.

Uniquement, pour ceux qui veulent aller plus loin que ce que l'on peut vous demander cette année :

1. En utilisant la méthode addEventListener et une fonction admettant en arguments l'élément sur lequel intervenir (grâce à this), la largeur et la hauteur à imposer à l'image.

Pour cela, il faut utiliser les fonctions anonymes : ex

Element\_a\_choisir.addEventListener("mouseout",function(){changerTaille(this,100,100)},false);

1. Modifier le code HTML, pour insérer des images supplémentaires. Le code Javascript doit maintenant agrandir toutes les images de la page.

**Ex 5 : diaporama (\*)**

Écrire le code permettant de créer un diaporama : un clic sur le bouton suivant fait apparaitre l'image suivante, un clic sur le bouton retour fait apparaitre l'image précédente. (Vous choisirez 5 images de votre choix)

Rq : Vous devrez gérer l'aspect '"circulaire" du diaporama (arrivé à la dernière image, le clic sur suivant affiche la première image)

Aide : vous pourrez utiliser un tableau contenant les url des 5 images



**Ex6 : Chiffrement code césar \*\***

Le code permet de coder du texte en décalant simplement les lettres de l'alphabet d'un certain nombre.

Exemple, le mot **souriez** codé en code césar avec un décalage de 1 donne **tpvsjfa.**

1. Créer la page permettant de saisir un texte et un décalage, et qui lorsque l'on appuie sur un bouton "coder" donne le code César correspondant. (texte d'entrée en minuscule)

Aide :

* en Javascript, on peut accéder aux lettres d'une chaîne de caractères de la même manière que l'on accède à un tableau:

let mot='coucou';

mot[0]=='c' return true

* On peut obtenir le code ASCII d'une lettre ( le nombre correspondant à la lettre dans le système de codage ASCII) grâce à la fonction (méthode pour être plus juste car programmation objet) : **caractere.charCodeAt()**

Exemple : 'a'.charCodeAt() renvoie 97

Pour obtenir le caractère correspondant à un code ascii : **String.fromCharCode(codeAscii)**

Exemple : String.fromCharCode(97) renvoie 'a'

Vous pourrez le tableau des codes Ascii à la page <https://fr.wikipedia.org/wiki/Fichier:ASCII-Table.svg>

* Les valeurs d'un champ de formulaire sont du texte (même avec l'attribut type='number') Pour convertir une chaine de caractère en nombre, vous pouvez utiliser la fonction **parseInt(string)** (de la même manière que int() en python)

1. Afin d'utiliser votre programme en décodage, ajouter un bouton décodage qui essaiera tous les décalages possibles afin de trouver celui qui convient:

Essayer de décoder la phrase suivante : tjsng ngmk snwr jwmkka

1. Modifier votre programme afin qu'il gère les majuscules ( question facultative)

**Ex 7 : (d'après lycée La martinière)**

Lire les informations concernant la balise détail : <https://developer.mozilla.org/fr/docs/Web/HTML/Element/details>

Exécuter puis comprendre ce que déclenche l'événement dans le code : **exo\_7\_JS.html**

Modifier le code afin qu'un survol de l'énoncé par la souris remette en place le texte de l'aide (qui est visible dans le code source).

**Annexe : code ex 0**

<!DOCTYPE html>

<html lang=**"fr"** >

<head>

<meta charset=**"UTF-8"**>

<title>**Dessiner et programmer avec canvas\_TP1**</title>

</head>

<body>

<h1>**De l'art avec Javascript !**</h1>

<canvas id=**"caneva"** width=**"400"** height=**"400"** style=**"border: solid black 1px"**></canvas>

<div>

<input id=**"bouton\_rouge"** type=**"button"** value=**"rouge"** onclick=**"changer\_rouge()"**>

<input type=**"button"** value=**"vert"**>

<input type=**"button"** value=**"bleu"**>

<input type=**"button"** value=**"Effacer"** onclick=**'effacer()'**>

</div>

<script>

**var** cadre**=**document.getElementById**(**"caneva"**);**

cadre.addEventListener**(**"mousemove"**,**dessiner**,false);**

// on déclare un espace de dessin 2d

**const** ctx **=** cadre.getContext**(**"2d"**);**

// on initialise les variables globales de compostantes couleur

**var** rouge**=**5**;**

**var** vert**=**19**;**

**var** bleu**=**50**;**

**function** effacer**()** // efface la zone de dessin

**{**ctx.clearRect**(**0**,** 0**,** caneva.width**,** caneva.height**);}**

**function** dessiner**(**evt**){**

// on récupère les coordonnées de la souris dans le canvas

**var** mousePos **=** getMousePos**(**cadre**,** evt**);**

**let** x**=**mousePos**[**0**];**

**let** y**=**mousePos**[**1**];**

// on dessine un disque

ctx.beginPath**();**

ctx.strokeStyle**=**couleur\_rvb**(**rouge**,**vert**,**bleu**);**

ctx.arc**(**x**,**y**,** 15**,** 0**,** 2**\***Math.PI**);** // on trace un cercle

ctx.stroke**();**

**}**

**function** changer\_rouge**(){**

rouge**=**255**;**

vert**=**0**;**

bleu**=**0**;**

**}**

// fonction qui retourne une chaine de caractere "rgb(r,v,b)

// on convertit un entier n en str avec la methode n.toString()

**function** couleur\_rvb**(**r**,**v**,**b**){**

**var** couleur**=**"rgb("**+**r.toString**()+**","**+**v.toString**()+**","**+**b.toString**()+**")"**;**

**return** couleur**;**

**}**

**function** getMousePos**(**canvas**,** evt**)** **{**

**var** rect **=** canvas.getBoundingClientRect**();**

**return** **[**evt.clientX **-** rect.left**,**evt.clientY **-** rect.top**];**

**}**

</script>

</body>

</html>