

## DEVELOPPEMENT WEB FRONT-END

### TP N°3 - JQUERY -

(Initiation et/ou révision)

#### Recommandations

Lire le [tutoriel jQuery](#) du site w3Schools. Dans ce tutoriel, ne ratez pas [jQuery Examples](#).  
Une fois votre lecture terminée, testez vos compétences avec les exercices sur [jQuery Exercises](#).

NB : Un autre [tutoriel jQuery](#) du site *TutorialsTeacher* bien expliqué.

#### Exercice 1 (Diaporama d'un jeu de cartes)

En TP2, le but de l'exercice 7 était de coder un diaporama de jeu de cartes en HTML/JS/DOM.  
Reprendre cet exercice pour le coder en HTML/JS/jQuery.  
Ajouter un effet de fondu pour l'affichage de la carte sur clic des boutons Précédent et Suivant.

Utiliser :

```
$(document).ready(function(){...}) ; element.click(function(){...}) ;
```

méthodes: *val()*, *prop("disabled", true)*, *prop("disabled", false)*, *attr("src",...)*, *fadeIn(...)*, *fadeOut(...)*.



#### Exercice 2 (Shaun le mouton)

Le but est de faire avancer, stopper, retourner puis enfermer *Shaun* le mouton dans la bergerie. Shaun peut aller à droite, et à gauche après s'être retourné. Il bouge ses pattes pour marcher, résultat d'une animation à partir des 8 images se trouvant dans le dossier :

/home/public/2A\_INFO/WEB\_FRONT\_END/TP3\_JQUERY/2\_MOUTON/images.zip.

Utiliser :

**HTML :**

1 img et 4 button.

**CSS :**

img { position : relative ; } /\* Requis pour bouger l'image\*/

**JS/jQuery :**

setInterval(affiche, 500); où *affiche()* utilise les 8 images pour modifier la source de l'image et faire marcher Shaun. ClearInterval(), ...

Sur événements click sur les boutons : animate({left: "+=150px"},...), stop(), css({transform: 'scale(-1, 1)'}), ...



### Exercice 3 (Planètes du système solaire)

But : simuler le mouvement des planètes et de leurs satellites autour du soleil.

On suppose que les orbites de ces corps célestes sont des ellipses qui sont toutes dans le même plan. L'équation cartésienne d'une ellipse est :

$$x=a*\cos(t)n; y=b*\sin(t)$$

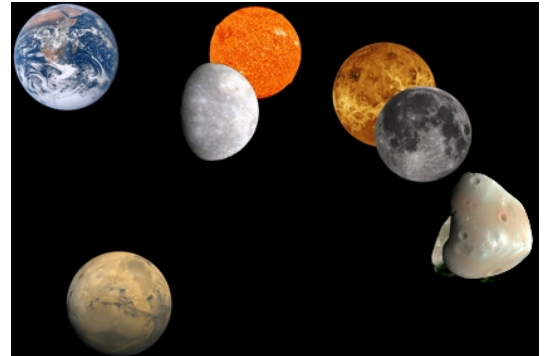
où  $a$  et  $b$  sont les "demi-grands" axes de l'ellipse et  $t$  est la période.






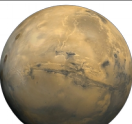


1. Commencer par placer l'image d'une planète et l'animer en lui faisant parcourir un mouvement elliptique.

2. Construire une structure de données permettant de modéliser et d'animer la liste des planètes et des satellites. Utiliser un tableau (*array*) d'objets. S'inspirer de l'Exercice 8 : (Objets, propriétés, array, boucles, fonctions) du sujet **TP2**.

3. Les images sont dans :

/home/public/2A\_INFO/WEB\_FRONT\_END/TP3\_JQUERY/3\_PLANETES/images.zip.



N°	Nom	a	b	Période	en orbite autour de	
0	<b>Soleil</b>	0	0	-		
1	<b>Mercur</b>	58	59	88	Soleil	
2	<b>Vénus</b>	109	108	225	Soleil	
3	<b>Terre</b>	150	149	365	Soleil	
4	<b>Lune</b>	0.3	0.3	27	Terre	
5	<b>Mars</b>	228	227	686	Soleil	
6	<b>Phobos</b>	0.9	0.9	0.4	Mars	
6	<b>Deimos</b>	0.3	0.3	1.2	Mars	

Utiliser :

- CSS :  
background-color : black;
- JS/jQuery :  
window.addEventListener('load', ...) ;  
querySelector("table") ;  
objets, this, ... (voir Exercice 8 du TP2).

Pour afficher l'image d'une planète et la positionner :

- <img id=...src=..
- css('marginTop', ...) , css('marginLeft',...),  
css('height')/2, css('width')/2, css('top', ...) , css('left', ...) ,  
css('position', 'fixed'), ...

Pour associer à l'image un mouvement elliptique :

- calculer (x,y) par l'équation cartésienne ; Y ajouter la distance par rapport à son orbite (distance gauche et haut).
- Bouger la planète avec animate({left: x + "px", top: y + "px"},1).

Pour animer les planètes du systèmes solaire :

- changer la position de chaque planète avec setTimeout en intégrant un compteur *temps* dans *cos* et *sin*.

4. Optionnel : intéressé (e) par les planètes du système solaire ? Consulter le site suivant :

<http://solarviews.com/french/homepage.htm>