

Dans cet exercice nous allons utiliser:

- Utilisation des Hooks

Interactions

Qu'est-ce qu'un HOOK ?

Un **hook** (littéralement « crochet » ou « hameçon ») permet à l'utilisateur d'un logiciel de personnaliser le fonctionnement de ce dernier, en lui faisant réaliser des actions supplémentaires à des moments déterminés. Le concepteur du logiciel prévoit des **hooks** au long du fonctionnement de son programme, qui sont des points d'entrée vers des listes d'actions. Par défaut, le hook est généralement vide et seules les fonctionnalités de base de l'application sont exécutées.



Cependant, l'utilisateur peut « accrocher » des morceaux de programme à ces hooks pour personnaliser le logiciel. *Source wikipédia*

Dans Ludiscape il existe des hooks

Ces hooks ne sont appelés par le moteur que **si ils été déclarés** à un endroit (par exemple Script Manager, ou dans un fichier JS embarqué)

Dans ce TP vous devez afficher dans une boîte de message (fonction alert) les différents paramètres de chaque hook.

Puis dans un deuxième temps :

Vous allez faire une copie du fichier ludiscape réalisé dans le **TP11** et afficher la progression et les remarques dans la zone de messages de manière automatique.

Ajouter plusieurs écrans QCM de façon à rendre le test plus réaliste.

Les Trois **hooks** de Ludiscape :

//A la fin du contenu (appelé par l'objet **fin de TP** uniquement);

```
function ludiscapeFinishActivity(login,activityId,title,success,duration,score_raw){  
  
}
```

//A la fin du contenu (appelé par : Note, Fin de TP, Bilan, Rapport)

```
function ludiscapeFinishAll(login,activityId,title,duration,score_raw){  
  
}
```

//A la fin d'un écran (

```
function ludiscapeFinishAll(login,activityId,title,duration,score_raw){  
  
}
```

duration = temps en millisecondes de l'exécution de l'écran

Pour surcharger la zone de message du TP11:

```
document.getElementById("messages-list").innerHTML = nouvelle valeur;
```