

# ElectronJS - Examen blanc : ElectronZoo - durée : 5 heures

---

Vous allez créer une application Electron permettant de gérer les animaux d'un zoo, nommée "ElectronZoo"

Les utilisateurs pourront :

- consulter la liste des animaux
- ajouter des animaux avec un nom, description, taille, poids et photo

Des éléments d'interface graphique et de CSS sont demandés : vous êtes **libres** sur la manière dont vous les intégrez. Vous pouvez utiliser les librairies JS/CSS de votre choix ou bien aucune si vous voulez. Une note d'appréciation visuelle sera donnée : le visuel peut être très simple mais ce que vous faites doit être soigné.

## Exercice 1 - Fenêtre principale (Total: 5 points)

1. Créez une fenêtre principale de format 1024x768 qui contiendra la liste des animaux. - **(1 point)**
2. Dans cette fenêtre, on doit avoir : - **(2 points)**
  - une barre de navigation contenant le titre de l'application "HBZoo" et un **lien** "Ajouter un animal"
  - Une liste des animaux présents sous la forme de petites "cartes"
  - Ces "cartes" contiendront :
    - Le nom de l'animal
    - Sa taille
    - Son poids
    - Sa description
    - Sa photo (la photo doit venir d'un lien vers une photo sur Internet)

*Comme vous n'avez pas encore géré l'ajout d'animaux, vous pouvez ajouter des animaux "en dur" dans le HTML en attendant d'afficher ceux enregistrés, ce que vous ferez dans un des exercices suivants.*

3. Cette fenêtre doit avoir une taille **maximale** de 1024x768 et une taille **minimale** de 1024x768 (on ne peut ni agrandir ni réduire la taille de la fenêtre). - **(2 points)**

## Exercice 2 - Ajout d'un animal (Total: 8 points)

1. Vous devez faire en sorte que le bouton **Ajouter un animal** ouvre une nouvelle fenêtre de taille 800x600 qui contiendra le formulaire d'ajout d'un animal. - **(2 points)**
  - Sur le même principe que l'exercice 1.3., la taille doit être fixe.
  - Dans cette page, vous aurez :
    - Un titre de la page en H1
    - Des champs de formulaire permettant de saisir :
      - Le nom de l'animal

- Sa taille
  - Son poids
  - Sa description
  - Sa photo (la photo doit venir d'un lien vers une photo sur Internet)
  - Un bouton "Ajouter l'animal"
2. Lors du clic sur "Ajouter l'animal", l'animal doit être ajouté à un tableau qui sera stocké dans le LocalStorage. - **(2 points)**
  3. Lors de l'ajout d'un animal, vous devez afficher une alerte Javascript avec la phrase : "L'animal nom\_de\_lanimal a bien été ajouté." - **(2 points)**
  4. La fenêtre "Ajout de l'animal" doit se refermer une fois l'animal enregistré. - **(2 points)**

### Exercice 3 - Affichage des données (Total: 4 points)

1. Vous devez afficher sur la page d'accueil les données issues du LocalStorage. - **(2 points)**
2. À chaque fois que l'on ajoute un animal, la page d'accueil doit toujours afficher la liste complète des animaux, mais contenant également le nouvel animal ajouté. - **(2 points)**

### Exercice 4 - Rendu du projet (Total: 3 points)

1. Appréciation du code : le code HTML/CSS/JS doit être propre et sans erreurs et bien indenté. L'aspect visuel doit être soigné (il peut être simpliste mais pas bâclé). Le nommage des variables doit être cohérent. - **(2 points)**
2. Vous devez rendre un fichier zippé contenant tous vos fichiers **SAUF** le dossier `node_modules`. Le fichier zippé devra être nommé comme suit : `hbzoo-prenom`. Le projet doit pouvoir être lancé grâce aux commandes `npm install` puis `npm start` (ou bien vous devez fournir la documentation des commandes à saisir si vous utilisez des librairies requérant d'autres commandes). - **(1 point)**

### Exercice 5 - Exercices bonus (Total: 2 points)

1. Vous devez ajouter un bouton de menu d'application (comme les boutons "Fichier", "Editer"... dans les applications habituelles) qui permette de fermer l'application quand on clique dessus. - **(2 points)**