Javascript - Objets

Mise en pratique: Annuaire téléphonique

But: développer un programme qui permet d'afficher le numéro de téléphone d'un ami (à l'aide de `alert`) dont le nom a été saisi par l'utilisateur (à l'aide de `prompt`).

Afin que l'annuaire soit extensible à l'avenir, aucune condition `if` ne doit être utilisée. Nous allons donc stocker les noms et numéros de téléphones dans un objet, et utiliser l'adressage par crochets pour trouver un numéro à partir d'un nom.

```
let annuaire = {
  patricia: '06 66 66 66 66',
  david: '07 77 77 77 77',
};
```

1. Afficher dans la console le numéro de téléphone de `patricia`, en utilisant la notation pointée sur l'objet `annuaire`;
2. Demander à l'utilisateur de saisir un prénom, puis afficher le numéro de téléphone associé à ce prénom.

Hiérarchie d’objets

Comme il est possible de stocker un objet comme valeur d’une propriété d’objet, cela veut dire que nous pouvons définir des hiérarchies / arbres d’objets.

Exemple:

```
let compteFacebook = {
  amis: {
    'Luke Skywalker': true,
    'Dark Vador': false,
  },
  groupes: {
    maitresJedi: {
      titre: 'Groupe des maîtres Jedi',
      membres: [ 'Yoda', 'Obi Wan' ],
    },
    lolcats: {
      titre: 'Vive les chats !',
      membres: [ 'Patrick' ],
    },
  },
};
```

Hiérarchie d'objets

Dans l'exemple ci-dessus, nous avons jusqu'à trois niveaux d'imbrication d'objets (voire quatre, si on compte les tableaux comme des objets): l'objet `compteFacebook` contient une propriété `groupes` (de type objet) contenant une propriété `maitresJedi` (aussi de type objet) contenant deux propriétés (de type chaîne de caractères et tableau).

Cet objet peut être représenté visuellement sous forme d'un arbre.

Pour accéder au premier élément du tableau `membres` de la sous-propriété `maitresJedi`, on doit écrire le cheminement à suivre, de propriété à sous-propriétés.

Exemple:

```
// adressage par notation pointée:
compteFacebook.groupes.maitresJedi.membres[0]; // => 'Yoda'

// adressage par crochets:
compteFacebook['groupes']['maitresJedi']['membres'][0]; // (idem)
```

Dans cet exemple aussi, on peut utiliser indifféremment la notation pointée ou les crochets, pour accéder à la propriété d'un objet.

Énumérer les propriétés d'un objet

Il existe deux manières d'énumérer les propriétés d'un objet:

1. utiliser une boucle `for-in`;
2. ou itérer sur le tableau des clés de l'objet, à l'aide de la fonction `Object.keys()`.

Une boucle `for-in` s'utilise de cette façon:

```
let mesAmis = {
  'Luke Skywalker': true,
  'Dark Vador': false,
};
for (let cle in mesAmis) {
  console.log(cle, '->', mesAmis[cle]);
}
// => lignes affichées dans la console:
//    Luke Skywalker -> true
//    Dark Vador -> false
```

Suppression d’une propriété

Pour supprimer une propriété d’un objet, il faut utiliser le mot clé `delete` de la manière suivante:

```
let objet = {
  prop1: 'a',
  prop2: 'b',
  prop3: 4,
};
delete objet.prop2;
// => objet === { prop1: 'a', prop3: 4 }
```

Exercice: Répertoire téléphonique

Sur la base de l'annuaire téléphonique réalisé plus haut, développer un programme qui propose les fonctionnalités suivantes:

- recherche et affichage d'un numéro de téléphone en saisissant un nom,
- liste des contacts de l'annuaire (nom + numéro de téléphone, à afficher dans la console à l'aide d'une boucle `for..in`),
- ajout d'un contact (nom + numéro de téléphone) dans l'annuaire,
- suppression d'un contact.

Pour permettre à l'utilisateur de choisir la fonctionnalité qu'il souhaite utiliser, lui demander de saisir la première lettre de la fonction:

- `'r'` pour rechercher,
- `'l'` pour afficher la liste,
- `'a'` pour ajouter,
- `'s'` pour supprimer.
- 'q' pour arrêter

Afin que l'utilisateur puisse utiliser plusieurs fonctionnalités d'affilée, placer le code du programme dans une boucle qui ne s'arrêtera que lorsque l'utilisateur aura tapé `q` au lieu de choisir une fonctionnalité.