(cc) BY-NC-ND

On associe souvent la **géolocalisation** aux systèmes de navigation embarqués comme le GPS. Mais le GPS n'a pas que le calcul d'itinéraire comme application, on voit apparaître de nos jours des applications utilisant ce système qui permet par exemple d'annoter des photos avec des informations de **géolocalisation**, ou d'informer en temps réel un interlocuteur de sa localisation. La seule limite étant votre imagination. Je vous montre comment utiliser l'API Javascript de **géolocalisation**.

Prérequis

– au moins un navigateur compatible sous la main, Firefox de préférence;
– savoir ce qu'est le **code html** et en avoir pratiqué un peu;
– savoir ce qu'est le **code Javascript** et en avoir pratiqué un peu.

La **géolocalisation**, c'est quoi ?

La **géolocalisation** est un système permettant à un utilisateur de se localiser sur une carte. Un système de **géolocalisation** renvoie 3 informations nécessaires à sa localisation :
– la longitude;
– la latitude;
– et l'altitude.

La réalité augmentée, c'est quoi ?

La réalité augmentée consiste à superposer à la réalité et en temps réel, des éléments virtuels pertinents pour l'utilisateur. Certains smartphones proposent déjà ce type d'application : en associant GPS, boussole, gyroscope, accéléromètre et capteur vidéo, l'application affiche des informations incrustées en temps réel sur ce que l'utilisateur voit au travers du capteur vidéo.

La **géolocalisation** avec **html et javascript**

La **géolocalisation** n'est pas un système propre à html5, mais la spécification permet de l'utiliser par le biais d'une **API Javascript**. Du coup, vous pouvez utiliser la **géolocalisation** que ce soit une application mobile ou une application classique.
Toutefois, la précision de la **géolocalisation** est plus ou moins parfaite.
Depuis un PC connecté à Internet, le système de **géolocalisation** utilise votre adresse IP qui dépend du routeur de votre fournisseur d'accès sur lequel vous êtes connecté. En réalité, c'est le routeur qui est géolocalisé, ce système est très imparfait.
Depuis un Mobile, le système utilise :
– soit votre position par rapport aux antennes GSM situées à proximité du mobile;
– soit le GPS du mobile, c'est dans ce dernier cas que la précision sera la meilleure.

L'**API de géolocalisation Javascript**

Javascript met à disposition du développeur une API très simple à utiliser dédiée à la

géolocalisation.

Mon navigateur supporte t-il la **géolocalisation** ?

La méthode ***geolocation*** de l'objet navigator renvoie un booléen indiquant si le navigateur supporte (***true***) ou non (***false***) la **géolocalisation**.
Un seul test suffit :

```
<script>
if ( navigator.geolocation ) {
  // geolocalisation supportée
} else {
  // geolocalisation non supportée
}
</script>
```

Préalablement à l'exécution, le navigateur vous demande si vous acceptez d'être géolocalisée.

Comment obtenir mes coordonnées de **géolocalisation** ?

La méthode ***navigator.geolocation.getCurrentPosition*** prend de un à trois paramètres :
– le premier est une fonction de callback qui est exécutée lorsque la **géolocalisation** s'est correctement passée;
– le deuxième est une fonction de callback qui est exécutée en cas d'erreur lors de la géolocalisation;
– le troisième paramètre est un objet ***PositionOptions*** qui comporte trois propriétés optionnelles : ***enableHighAccuracy***, ***timeout***, et ***maximumAge***.

```
navigator.geolocation.getCurrentPosition(
  successCallback,
  errorCallback,
  {
    enableHighAccuracy : true,
    timeout : 5000,
    maximumAge : 60000
  }
);
```

La propriété ***enableHighAccuracy*** est un indicateur de précision, s'il est fixé à true, la méthode de **géolocalisation** est la plus précise possible parmi celles disponibles (IP, GSM, GPS, ….).
La propriété ***timeout*** est le temps exprimé en ms à partir duquel la recherche de position est stoppée. Dans ce cas, la fonction callback d'erreur (deuxième paramètre) est appelée.

La propriété ***maximumAge*** est une propriété relative à l'utilisation du cache. *maximumAge* est une durée exprimée en ms en dessous de laquelle, si une recherche de position a déjà été faite, une nouvelle recherche de position n'est pas refaite : c'est la valeur stockée dans le cache qui est renvoyée. Si la durée *maximumAge* par rapport à la dernière recherche de position faite est dépassée, une nouvelle recherche est faite.
Si ***maximumAge*** est :
– fixée à 0, le cache n'est pas pris en compte,
– fixée à *Infinity* le cache est systématiquement pris en compte, sauf s'il est vide et dans ce cas la recherche de position est faite.
Lorsqu'une géolocalisation échoue, une fonction callback d'erreur (le deuxième paramètre ***afficherErreurGeoLoc***) est appelée. Cette fonction prend en paramètre un objet ***PositionError*** qui encapsule :
– un code erreur parmi ***PERMISSION_DENIED***, ***POSITION_UNAVAILABLE*** et ***TIMEOUT***;
– un message d'erreur ***message***.
Lorsqu'une géolocalisation est faite, une fonction callback (le premier paramètre ***afficherGeoLocPosition***) est appelée. Cette fonction prend en paramètre un objet ***Position*** qui encapsule les informations de **géolocalisation** sous la forme d'un objet ***Coordinates*** *coords*, entre autres :
– ***coords.altitude*** renvoie l'altitude;
– ***coords.longitude*** renvoie la longitude;
– ***coords.latitude*** renvoie la latitude;
– ***coords.accuracy*** renvoie le niveau de précision en m pour la latitude et la longitude;
– et ***coords.altitudeAccuracy*** renvoie le niveau de précision en m pour l'altitude.
En **Javascript** :

```
<script>
var afficherGeoLocPosition = function(geolocPosition) {
  document.write('<br>longitude ' + geolocPosition.coords.longitude);
  document.write('<br>latitude ' + geolocPosition.coords.latitude);
  document.write('<br>altitude ' + geolocPosition.coords.altitude);
}

if(navigator.geolocation) {
  // geolocalisation supportée
  navigator.geolocation.getCurrentPosition(afficherGeoLocPosition);
} else {
  // geolocalisation non supportée
}
</script>
```

L'objet ***Coordinates*** encapsule deux autres paramètres non pertinents avec la méthode ***getCurrentPosition*** :
– ***coords.heading*** est indicateur de direction ou send de déplacement exprimé en degrés par rapport au nord;
– et ***coords.speed*** renvoie la vitesse de déplacement exprimée en m/s.

Comment suivre un déplacement grâce à la **géolocalisation** ?

L'API de **géolocalisation** permet de suivre en temps réel un déplacement. La spécification est identique à la précédente à la différence qu'au lieu d'utiliser la méthode ***getCurrentPosition***, on utilise ici la méthode ***watchPosition*** dont les paramètres sont identiques et qui renvoie un identifiant.
La méthode ***watchPosition*** possède un timer interne (transparent pour l'utilisateur) : pour suivre un déplacement, il est nécessaire de faire des recherches de position à intervalles réguliers.
L'exécution de la méthode ***watchPosition*** n'ayant pas de limite d'exécution dans le temps, son exécution ne peut être stoppée que par la méthode ***clearWatch*** qui prend en paramètre l'identifiant renvoyé par ***watchPosition***.
Pour le reste, le fonctionnement est identique à la méthode ***getCurrentPosition*** : les callback sont appelés dans les mêmes circonstances et les paramètres s'utilisent de la même manière.
A la différence que :
– la fonction de callback de succès est rappelée uniquement si la position a changé;
– les paramètres ***heading*** et speed de l'objet ***Coordinates*** deviennent pertinents.

La gestion des erreurs

Lorsqu'une recherche de position échoue, la méthode callback d'erreur (deuxième paramètre des méthodes ***getCurrentPosition*** et ***watchPosition***) est invoquée.
Cette fonction prend en paramètre un objet ***PositionError*** qui encapsule tout deux informations expliquant l'échec :
– code qui peut prendre comme valeur ***PERMISSION_DENIED***, ***POSITION_UNAVAILABLE***, et ***TIMEOUT***;
– message qui est le message décrivant le détail de l'erreur.
PERMISSION_DENIED indique que la géolocalisation n'a pu se faire du fait d'une non autorisation de l'utilisation de l'API. N'oubliez pas que toute utilisation de l'API nécessite l'approbation de l'utilisateur.
POSITION_UNAVAILABLE indique que la géolocalisation n'a pu se faire du fait d'une erreur interne du ou des périphériques de géolocalisation.
TIMEOUT indique que le temps d'attente de la recherche de position est écoulé.

Conclusion

Comme vous pouvez le remarquer, cette API est très simple à utiliser. Cependant elle n'a une réelle utilité que dans le cadre d'une application mobile. Désormais, vous pouvez l'expérimenter.

[important]N’hésitez pas à poser vos questions ou ajouter vos remarques.[/important]

(cc) BY-NC-ND