```php
<?php
/*
Fichier: sessions.php
Auteur: Kevin Zaffino
Date: 15/06/2016
Version:1.10
Description: Gere les sessions
Copyright (TPI 2016 - Kevin Zaffino © 2016)
*/

session_start();
?>

<?php
require_once './phpScript/constants.php';

/**
 * Est-ce que l'utilisateur est loggé
 * @return {boolean}   True is loggé, autrement false
 */
function isLogged() {
    return (isset($_SESSION["logged"])) ? $_SESSION["logged"] : false;
}

/**
 * Definis l'etat "logged"
 */
function setLogged() {
    $_SESSION['logged'] = true;
}

/**
 * Recupere le niveau de privilège de l'utilisateur
 * @return (int)
 */
function getPrivilege() {
    return (isset($_SESSION["privilege"])) ? $_SESSION["privilege"] : PRIV_UNKNOWN;
}

/**
 * Assigne le nieau de privilege en session
 * @param (int) $value
 */
function setPrivilege($value) {
    $_SESSION["privilege"] = $value;
}

/**
 * Assigne le chemin de l'avatar de l'utilisateur en session
 * @param (string) $path
 */
function setImagePath($path) {
    $_SESSION['image'] = $path;
}

/**
 * Retourne le chemin de l'avatar de l'utilisateur contenu en session
```

```php
     * @return (string)
     */
    function getImagePath() {
        return (isset($_SESSION["image"])) ? $_SESSION["image"] : NULL;
    }

    /**
     * Est-ce que l'utilisateur est un admin
     * @return (bool) TRUE si il est admin, sinon FALSE
     */
    function isAdmin() {
        return getPrivilege() == PRIV_ADMIN;
    }

    /**
     * Est-ce que l'utilisateur est un simple utilisateur
     * @return (bool) TRUE si oui, sinon FALSE
     */
    function isUser() {
        return getPrivilege() == PRIV_USER;
    }

    /**
     * Recupere l'id utilisateur contenu en session
     * @return (int)
     */
    function getUserID() {
        return (isset($_SESSION["uid"])) ? $_SESSION["uid"] : -1;
    }

    /**
     * Assigne l'id de l'utilisateur en session
     * @param type $uid
     */
    function setUserID($uid) {
        $_SESSION["uid"] = $uid;
    }

    /**
     * Retourne le nom de l'utilisateur enregistré en session
     * @return type
     */
    function getUserName() {
        return (isset($_SESSION["uname"])) ? $_SESSION["uname"] : "";
    }

    /**
     * Assigne le nom d'utilisateur a une variable de session
     * @param type $name
     */
    function setUserName($name) {
        $_SESSION["uname"] = $name;
    }

    /**
     * Retourne le mail de l'utilisateur
     * @return type
```

```php
 */
function getUserMail() {
    return (isset($_SESSION["umail"])) ? $_SESSION["umail"] : "";
}

/**
 * Assigne le mails de l'utilisateur dans une variable de session
 * @param type $mail
 */
function setUserMail($mail) {
    $_SESSION['umail'] = $mail;
}

/**
 * Recupere le numero de l'utilisateur
 * @return type
 */
function getUserTel() {
    return (isset($_SESSION["utel"])) ? $_SESSION["utel"] : "";
}

/**
 * Assigne le numero de l'utilisateur dans une variable de session
 * @param type $tel
 */
function setUserTel($tel) {
    $_SESSION['utel'] = $tel;
}

/**
 * Recupere l'adresse de l'utilisateur
 * @return type
 */
function getUserAdress() {
    return (isset($_SESSION["uadress"])) ? $_SESSION["uadress"] : "";
}

/**
 * Assigne l'adresse de l'utilisateur dans une variable de session
 * @param type $mail
 */
function setUserAdress($mail) {
    $_SESSION['uadress'] = $mail;
}

/**
 * Detruit la session
 */
function destroySession(){
    session_destroy();
}

```