

# Test de l'application javascript

- Introduction
- Liste des assertions
- Fonctionnement

## Introduction

J'ai mis en place un système très simple de test pour cette application entièrement en javascript.

## Fonctionnement

Les fiches de test doivent se trouver dans le dossier `./template/tests/tests/`. Dès qu'une fiche de test est créée, il faut lancer en console (Terminal) l'utilitaire `./utils/test.rb` qui va préparer les tests.

Un test se compose de cette manière :

```
var test = new Test('Nom de mon test');

test.run = function(){

    // Je suis la méthode qui sera appelée par le moteur de test

    // Une méthode définie plus bas (pour la clarté)
    this.un_test_particulier();
}

test.un_test_particulier = function(){
    given('Une situation de départ');
    // Le travail
    // Les assertions pour vérifier
}
```

Ensuite, il suffit de charger le fichier `test.html` (créé par `test.rb`) et de lire la console de Firebug pour avoir le résultat du test.

## Assertions

**assert(evaluation, message\_success, message\_failure)**

Assertion de base qui génère le message de succès `message_succes` quand `evaluation` est `true` et génère le message de failure `message_failure` dans le cas contraire.

```
assert(  
    true,  
    'Je suis vrai',  
    'Je suis faux'  
)  
  
assert(  
    2 > 4,  
    'Deux est bien inférieur à 4',  
    'Deux ne devrait pas être supérieur à 4'  
)
```

Noter que pour les assertions négatives, il suffit d'utiliser `!evaluation`.

**assert\_classes(jqDes, classes) / assert\_not\_classes()**

Produit un succès si l'élément ou la liste d'éléments désignés par `jqDes` possèdent les classes (Array) désignées par `classes`. Produit une failure dans le cas contraire.

```
var mesDivs = document.getElementsByClassName('divisors');  
assert_classes(mesDivs, ['good', 'one'])
```

**assert\_position(jqDes, hposition[, tolerance]) / assert\_not\_position()**

Produit un succès si l'élément désigné par `jqDes` se trouve dans les positions définies par `hpositions` avec une tolérance optionnelles de `tolerance`.

`hpositions` est une table définissant les valeurs (nombre de pixels, sans unité) de `x` ou `left`, `y` ou `top`, `w` ou `width`, `h` ou `height`.

```
var monDiv = document.getElementById('monDiv');  
assert_position(monDiv, {x: 100, y: 200});
```

L'assertion ci-dessus génère un succès si le node `#monDiv` se trouve a un `left` de 100 pixels et un `top` de 200 pixels.

```
var monDiv = document.getElementById('monDiv');  
assert_not_position(monDiv, {w: 100, h:30}, 10);
```

Le code ci-dessus génère une failure si l'objet `#monDiv` a une largeur comprise entre 90 et 110 et une hauteur comprise entre 20 et 40.

On peut transmettre une liste (Array) de nœuds à la méthode.

**`assert_visible(jqDes) / assert_not_visible`**

Produit un succès si l'élément désigné par `jqDes` est visible, une failure dans le cas contraire.

```
assert_visible('#monDiv');  
  
assert_not_visible('.mesDivs');
```