

# TESTS D'INTERFACE ET AUTOMATISATION

# Test d'interface Homme-Machine

Les différentes activités:

- **Test fonctionnels**

Testeurs

Exigences : comportement des objets graphiques, chartes à respecter, navigation etc ...

- **Test ergonomique**

Spécialistes (ergonomes)

Mesure de l'utilisation du système

# Test d'interface: outils W3C

- (X)HTML valide: Markup Validation Service du W3C.
- CSS valide: valideur CSS du site W3C.
- URL, liens: Link Checker du site W3C.

<http://www.w3.org/QA/2002/07/WebAgency-Requirements.html>.fr

# Test d'interface: utilisabilité

Objectifs: Compréhensible, intuitive

- Temps d'accès aux informations
- Vocabulaire utilisé, acronyme, aide
- Taille des pages (scrolling, nombre de pages)
- ...
- Certains aspects peuvent être validés en Bêta Test, ergonome, questionnaire.

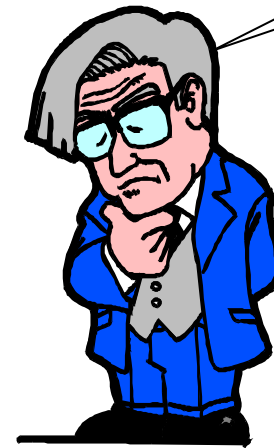
# Exemple

## MSISDN Identification

Please enter a valid MSISDN and press "Connect" to access to subscriber management

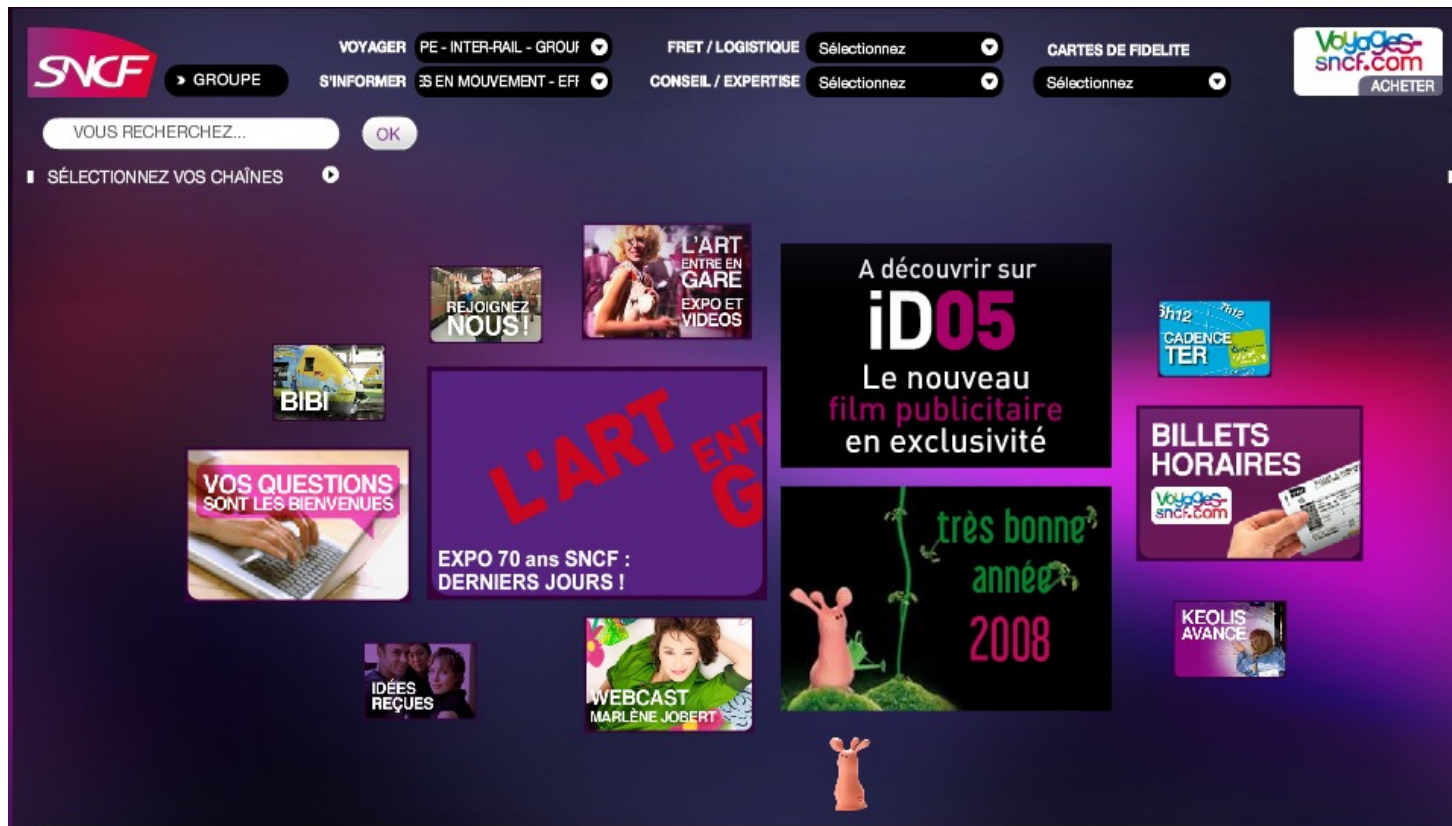
MSISDN

Connect



msisdn ?

# Exemple



# Utilisabilité : définition

- Un système est utilisable lorsqu'il permet à l'utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction dans le contexte d'utilisation spécifié.
- ISO 9241-11
  - Efficacité
  - Efficience
  - Satisfaction

# Définir ses besoins

- Efficacité : l'utilisateur va utiliser le service web pour gérer ses comptes (il ne va plus utiliser les services de l'employé)
- Efficience : l'opératrice va prendre moins de temps pour réaliser une tâche (gain de productivité)
- Satisfaction : l'utilisateur trouve mon site agréable à utiliser et ne va pas sur celui du concurrent.



# Quand tester

- Avant
  - Prototype : détecter au plus tôt les problèmes d'usabilité
- Après
  - Mesurer l'usabilité du système
- Métier => ergonomes

# Cibler les utilisateurs

Impossible d'avoir un site 100% adéquat à tous les différents profils d'utilisateurs.

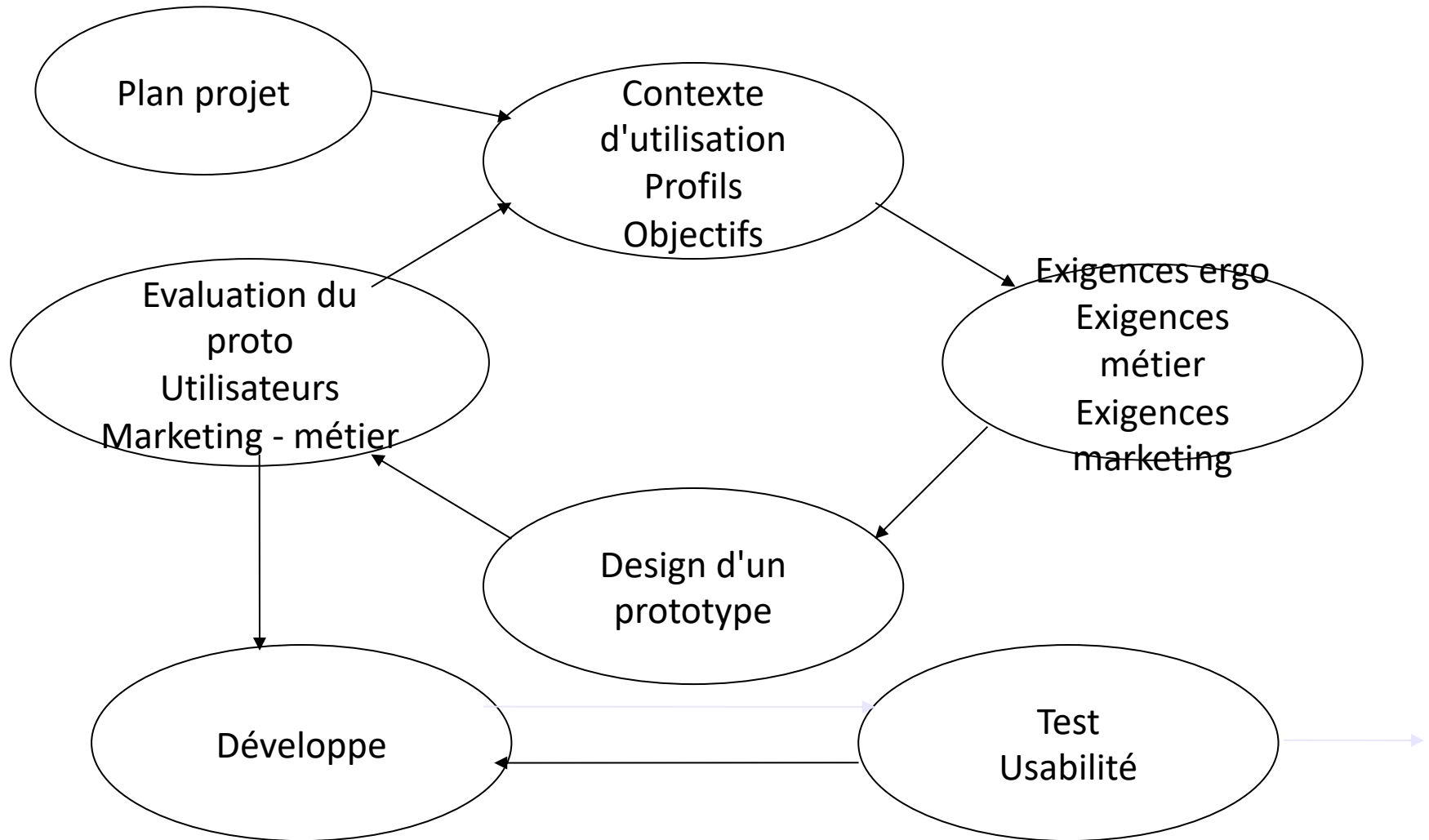
Exemple de profils:

- Ménagère
- Geek informatique
- Grand-mère
- Etc ...

# Comment tester

- Faire appel a des ergonomes
  - Prototypage
  - Audit
  - Test d'usabilité
- Comment
  - Grilles
  - Observation de panel d'utilisateur

# Cycle de vie



# Exemple de grille de mesure

- Feedback du système:

Les fenêtres/pop-ups/boîtes de dialogue s'ouvrent près de l'aire cliquée

- Sensation de contrôle – CTRL

Si le site dépasse les 100 pages, un moteur de recherche est présent

- Gestion des Erreurs – ERR

Si une erreur est commise dans un formulaire, elle est clairement décrite et expliquée

AUTOMATISATION

# Rappel sur les tests fonctionnels automatisés

- **Objectif**

exprime la condition de test

- **Procédure**

pré requis (Pré-conditions)

étape (attendus)

- **Jeux d'essais**

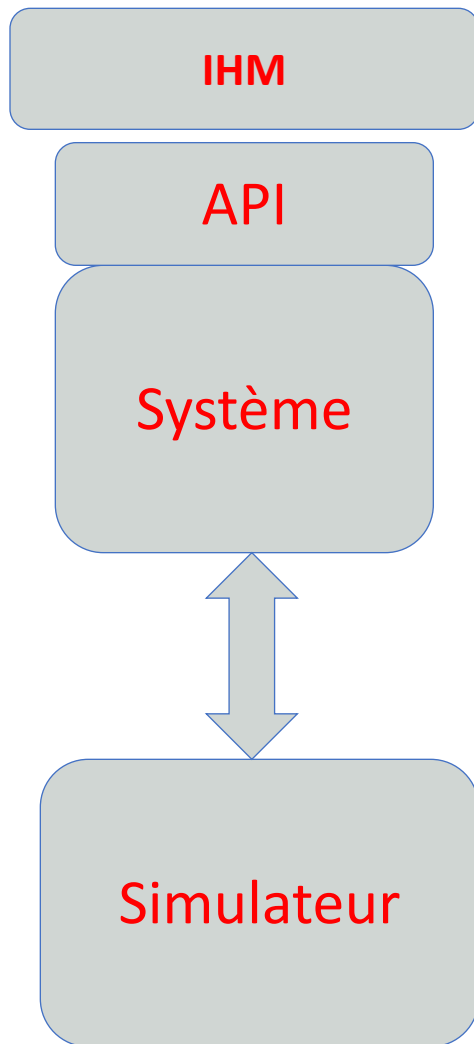
Valorisation (conception ou à l'exécution)

- **Attributs**

criticité, priorité



# Notion sur les tests fonctionnels automatisés



Tester des API nécessite des compétences en programmation.

Tester des IHM nécessite

1. Peu de compétences en programmation.
2. Des IHMs stables.
3. Des outils performants pour simuler les actions souris et claviers et pour capturer des éléments graphiques pour des vérifications.



# Notion sur les tests fonctionnels automatisés

- Outils qui manipulent les objets graphiques (mono techno ou multi techno).
  - Auto IT (IHM Windows, langage dédié).
  - Selenium (Web, java, python, etc.).
  - QTP (multi techno, VB).
- Outils qui se basent sur la position ou l'image  
Sikuli (multi techno, langage dédié).
- Tous ces outils possèdent un enregistreur d'actions.

# Selenium

- IDE
  - Enregistrement et rejeu
  - Plugin Firefox
  - Code html
- RC
  - Serveur
  - Contrôle et envoie les cmds aux navigateurs
  - Choix du langage
- GRID
  - Tests en parallèle

# Principe

1. Faire un enregistrement des actions et vérifications sur les objets graphiques
2. Enregistrer au format désiré
3. Retravailler pour intégrer les bonnes pratiques :
  1. Jeux d'essais
  2. Modularité

# Selenium : IDE

Extension : (Selenium IDE) - Selenium IDE - essaiEnregist

Project: essaiEnregist

Tests ▾ +

Search tests... 🔍

http://prestashop.qualifiez.fr ▾

	Command	Target	Value
1	open	/fr/	
2	set window size	800x541	
3	click	name=s	
4	type	name=s	MUG
5	click	css=button > .search	

Command: open

Target: /fr/

Value:

Description:

Log Reference

Enregistrement

Code enregistré

Liste des cmds

# Selenium: structure d'une commande

- Command
  - Actions (manipulent les objets graphiques)
  - Accessors (examinent l'objet graphique et stocke)
  - Assertions (vérifie l'objet graphique)
- Target
  - Élément de localisation (id, Xpath, Dom, etc.)
  - Utilisation des patterns
- Value
  - Optionnel (dépend des commandes)

# Identification d'un élément par sélénium dans une page

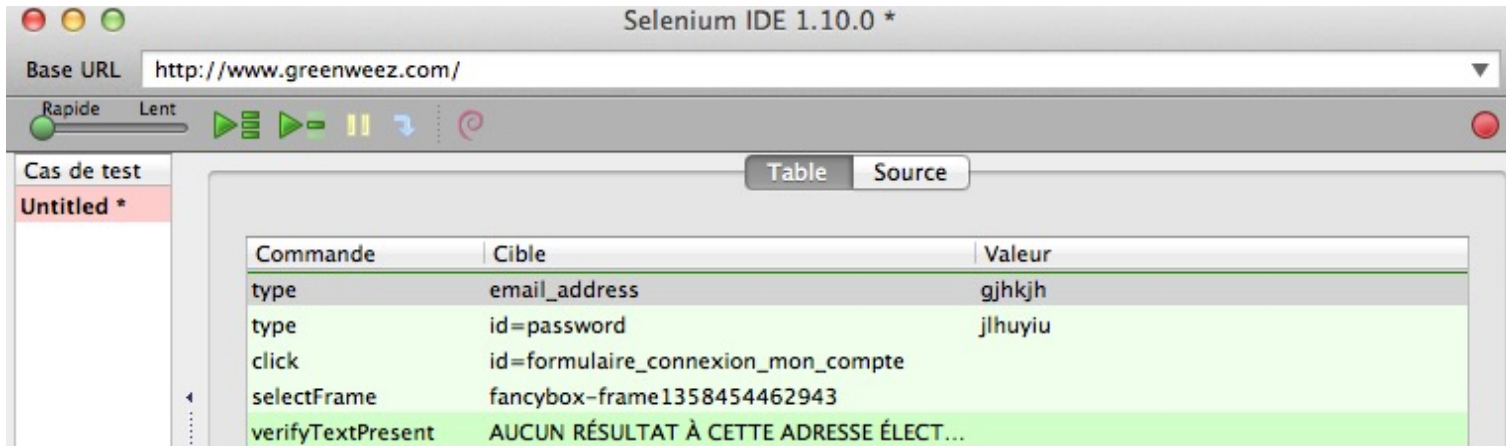
- Element locator

- Un élément fixe id, name ...
- Un élément dépendant de la structure de la page XPATH, DOM ...

id=lst-ib	
id=lst-ib	id
name=q	name
css=#lst-ib	css
document.forms['gs'].elements[...	dom:name
//input[@id='lst-ib']	xpath:attributes
//td[@id='sftab']/div/table/tbo...	xpath:idRelative
document.forms['gs'].elements[5]	dom:index
//div/input	xpath:position

- **Bonne pratique : imposer au développeur d'utiliser les id ou les name dans les pages HTML pour au niveau des tests automatiques être indépendant de la structure.**

# Identifiant par id ou name



- Identifier, id, name
- Identifier: recherche par id puis par name
- Type Identifier=email\_address
- Type Id=email\_address

# XPath définition

- XPath est une syntaxe pour définir des contenus d'un document XML.
- XPath utilise des expressions (requêtes) pour naviguer dans des documents XML.
- XPath possède une librairie de fonctions standard.
- XPath est un élément majeur de XSLT
- XPath est une recommandation W3C
- [W3schools.com](http://W3schools.com)



```
72 <body>
73   <div class="fullpage_head">
74     <p>
76     <br />TestLink 1.9.4 (Prague - Speedy)</p>
77   </div>
78   <div class="forms" id="login_div">
79     <p class="title">Reset password</p>
80
81     <form method="post" action="lostPassword.php">
82       <div class="messages" style="text-align:center;">User not found, please try again</div>
83
84       <p class="label">Login Name<br />
85       <input type="text" name="login" id="login"
86         size="32" maxlength="30" /></p>
87       <p><input type="submit" name="editUser" value="Send" /></p>
88     </form>
89
90
91     <hr />
92     <p><a href="login.php">Back to login</a></p>
93
94   </div>
95 </body>
96 </html>
```

Table

Source

Commande	Cible	Valeur
open	/testlink/lostPassword.php	
type	id=login	tgoto@free.fr
clickAndWait	name=editUser	
verifyTextPresent	User not found, please try again	

Commande

type

Cible

id=login

Valeur

Rechercher

id=login

id

name=login

name

css=#login

css

//input[@id='login']

xpath:attributes

//div[@id='login\_div']/form/p/input

xpath:idRelative

//input

xpath:position

UI-Element

ue)

```

72 <body>
73   <div class="fullpage_head">
74     <p>
76     <br />TestLink 1.9.4 (Prague - Speedy)</p>
77   </div>
78   <div class="forms" id="login_div">
79     <p class="title">Reset password</p>
80
81     <form method="post" action="lostPassword.php">
82       <div class="messages" style="text-align:center;">User not found, please try again</div>
83
84       <p class="label">Login Name<br />
85       <input type="text" name="login" id="login"
86         size="32" maxlength="30" /></p>
87       <p><input type="submit" name="editUser" value="Send" /></p>
88     </form>
89
90
91     <hr />
92     <p><a href="login.php">Back to login</a></p>
93
94   </div>
95 </body>
96 </html>

```

//input: sélectionne un élément de type input.

```

72 <body>
73 <div class="fullpage_head">
74 <p>
76     <br />TestLink 1.9.4 (Prague - Speedy)</p>
77 </div>
78 <div class="forms" id="login_div">
79     <p class="title">Reset password</p>
80
81     <form method="post" action="lostPassword.php">
82         <div class="messages" style="text-align:center;">User not found, please try again</div>
83
84         <p class="label">Login Name<br />
85         <input type="text" name="login" id="login"
86             size="32" maxlength="30" /></p>
87         <p><input type="submit" name="editUser" value="Send" /></p>
88     </form>
89
90
91     <hr />
92     <p><a href="login.php">Back to login</a></p>
93
94 </div>
95 </body>
96 </html>

```

//div[@id=« login\_div »]/form/p/input : sélection en relatif à partir de l'élément de type div dont l'attribut id est égal à « login\_div ».

```

72 <body>
73   <div class="fullpage_head">
74     <p>
76     <br />TestLink 1.9.4 (Prague - Speedy)</p>
77   </div>
78   <div class="forms" id="login_div">
79     <p class="title">Reset password</p>
80
81     <form method="post" action="lostPassword.php">
82       <div class="messages" style="text-align:center;">User not found, please try again</div>
83
84       <p class="label">Login Name<br />
85       <input type="text" name="login" id="login"
86         size="32" maxlength="30" /></p>
87       <p><input type="submit" name="editUser" value="Send" /></p>
88     </form>
89
90
91     <hr />
92     <p><a href="login.php">Back to login</a></p>
93
94   </div>
95 </body>
96 </html>

```

//input[@name='login'] : sélection d'un élément de type input dont l'attribut name est égal à « login ».

# Xpath en résumé

- Semblable au DOS
  - .. -> on revient sur le parent
  - / on descend sur l'enfant
  - // un descendant
  - \* n'importe quelle balise
  - @ spécifie un attribut

# Stratégie de localisation

- Identifier :
  - Le plus simple → règle de codage à mettre en place
  - Le moins sensible au changement
- Xpath
  - Mode relatif moins sensible
  - Offre de nombreuses possibilités
  - Outil: Xpath Checker, Firebug

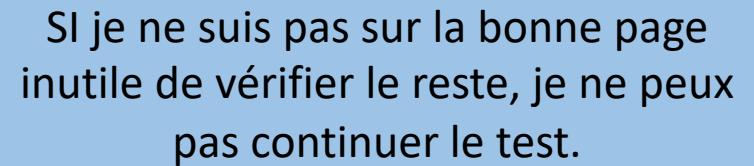
# Assertion ou Vérification

- Test → contrôle
- Présence d'un élément
- Valeur d'un attribut (texte)
- Si le contrôle est faux
  - Assert → échec du test et sortie du test
  - Verify → échec et le test continue



# Stratégie

- Certaines vérifications nécessitent d'être sûr que la page est affichée:
  - AssertTitle ...
  - VerifyText ...
  - VerifyText ...



Si je ne suis pas sur la bonne page  
inutile de vérifier le reste, je ne peux  
pas continuer le test.

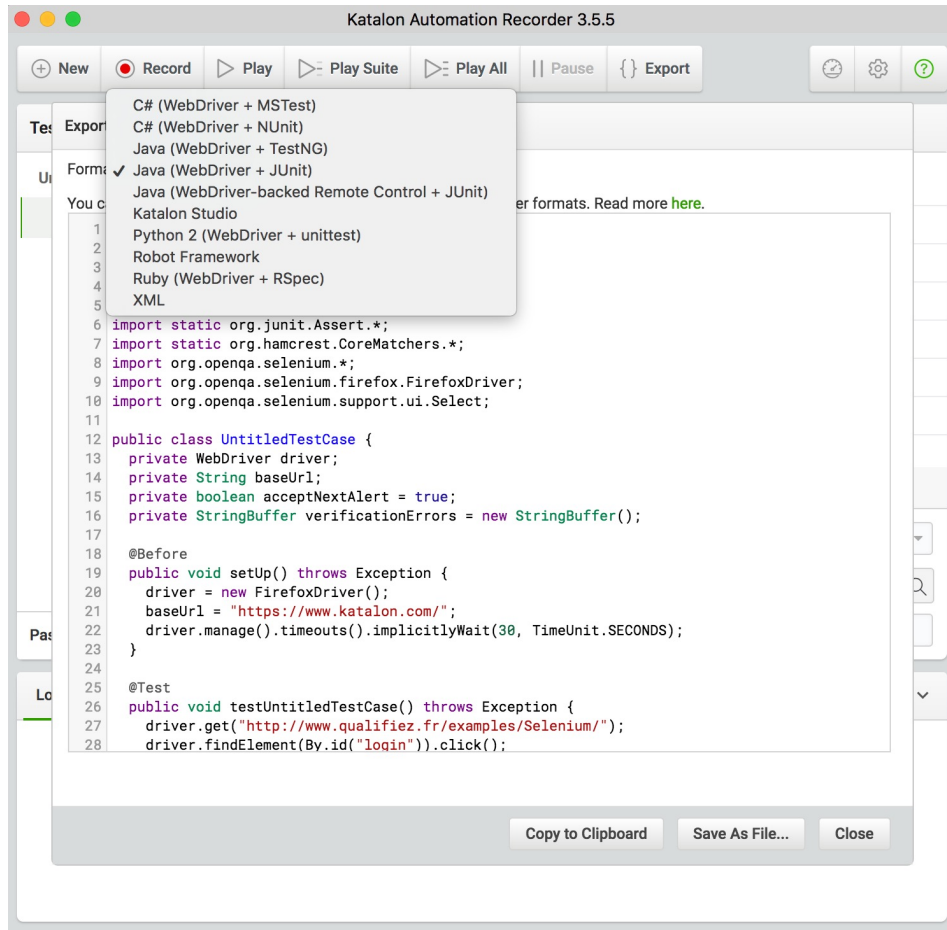
# La synchronisation: un problème typique du test d'IHM

- Changement de page
- Pop up
- Element graphique qui change
- Etc ...

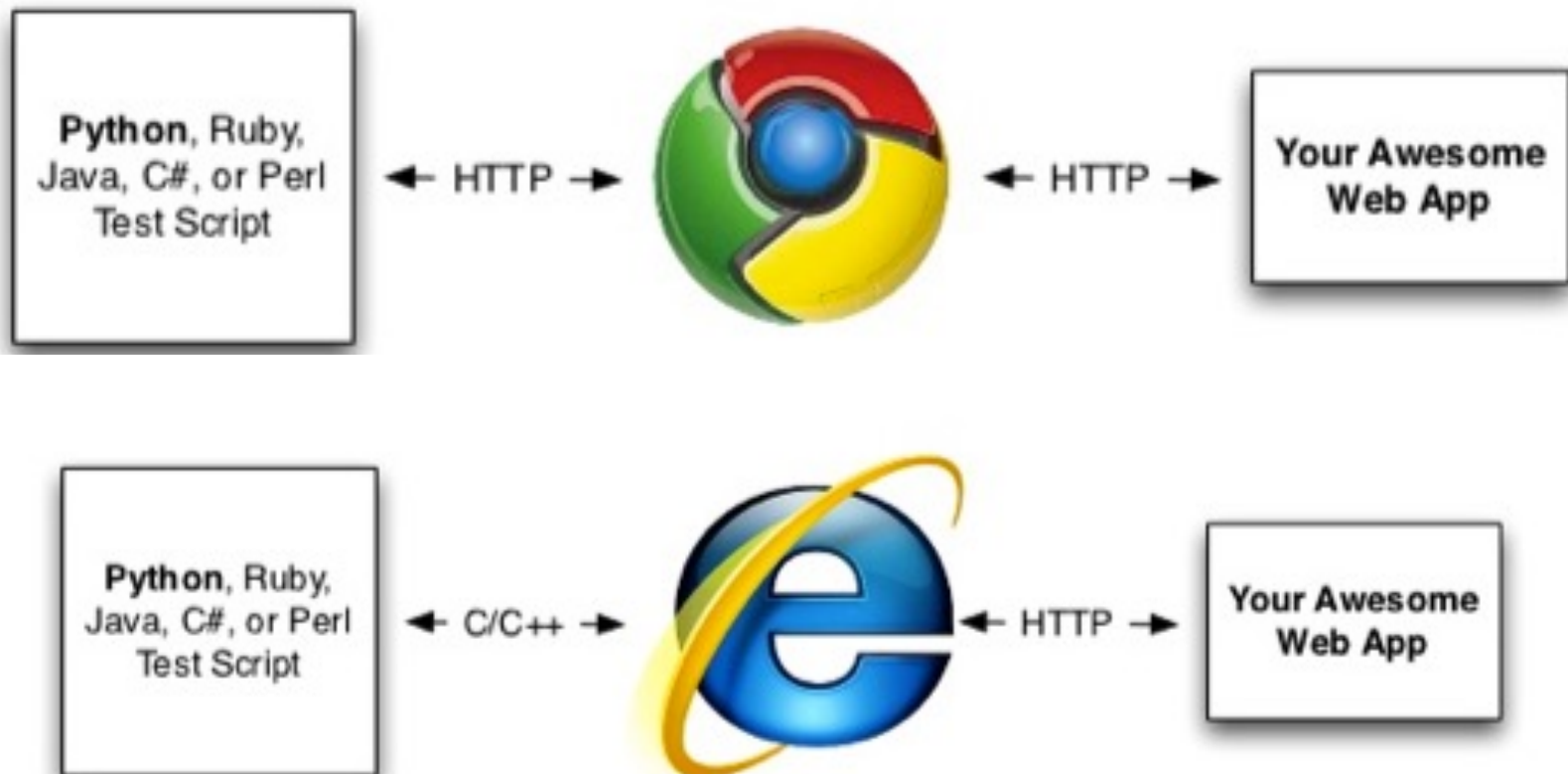
→ Il faut être synchronisé sur par exemple

- Présence d'un élément
- Bouton qui devient actif
- Etc ...

# Selenium : les différents formats



# Test Architecture : Web driver



# Robot Selenium Library

- Elle supporte les méthodes de localisation suivantes:

Strategy	Match based on	Example
id	Element <code>id</code> .	<code>id:example</code>
name	<code>name</code> attribute.	<code>name:example</code>
identifier	Either <code>id</code> or <code>name</code> .	<code>identifier:example</code>
class	Element <code>class</code> .	<code>class:example</code>
tag	Tag name.	<code>tag:div</code>
xpath	XPath expression.	<code>xpath://div[@id="example"]</code>
css	CSS selector.	<code>css:div#example</code>
dom	DOM expression.	<code>dom:document.images[5]</code>
link	Exact text a link has.	<code>link:The example</code>
partial link	Partial link text.	<code>partial link:he ex</code>
sizzle	Sizzle selector provided by jQuery.	<code>sizzle:div.example</code>
jquery	Same as the above.	<code>jquery:div.example</code>
default	Keyword specific default behavior.	<code>default:example</code>

# Robot Selenium Library : localisation

- Exemple de localisation:

```
lancerUneRecherche
  [Arguments]    ${mclef}
  Input Text     id=search_query_top    ${mclef}
  Click Element  name=submit_search
```

- Name et id  
→ l'id est une méthode de localisation intéressante car unique.
- Avec css

```
AllerALaPageDaccueil
  Click Image    css=img.img-responsive
```

# Robot Selenium Library

- Les commandes Open Browser/Close Browser
  - Les drivers (gecko,chrome ...) doivent être accessibles (variable environnement Path)
- Open Browser prend en argument l'Url, le navigateur ...
- Il est possible de travailler avec GRID en utilisant le paramètre Remote\_url

```
Ouvrir Application  
Open Browser    ${URL}    ${Browser}
```

---

```
*** Settings ***
```

```
Test Setup      Ouvrir Application  
Test Teardown   Fermer Application  
Library         Selenium2Library
```

```
*** Variables ***
```

```
${URL}          http://localhost/prestashop/index.php  
${Browser}      gc
```

# Robot Selenium Library

- Les commandes qui permettent de simuler des actions sur le navigateur:
  - Click Element, Click Button, Click Image, Drag And Drop, Input Text, Input Password, Select CheckBox ...

```
AllerALaPageDaccueil
    Click Image      css=img.img-responsive

lancerUneRecherche
    [Arguments]      ${mclef}|
    Input Text       id=search_query_top      ${mclef}
    Click Element    name=submit_search
```



# Robot Selenium Library

- Les commandes qui permettent de faire des contrôles:
  - Page Should Contain, Title Should Be, Table Should Contain, Radio Button Should Be Set To ...

---

```
*** Keywords ***  
Aller a la page de connexion  
    Click Link      Link = Connexion  
    Title Should Be Connexion - presta
```

Les commandes qui permettent de récupérer des informations de la la page:

- Get Text, Get Alert Message, Get Selected List Value, Get Window Titles, Get Value, Get Table Cell, Get Element Attribute ...

```
controle titre  
    ${res}    Get Text      css=h1.page-heading  
    Should Be Equal    ${res}    IDENTIFIEZ-VOUS
```

# Robot Selenium Library

- Les commandes qui permettent de se synchroniser:
  - Wait for condition, Wait Until Element Is Visible, Wait Until Element Is Enabled ...
  - Il peut être nécessaire avant de continuer le test de se mettre en attente (button qui devient actif, popup ...) du au temps de traitement d'une requête par le serveur.
  - La valeur du timeout peut-être préciser sinon c'est celle par défaut qui est utilisée.
  - De même chaque fois qu'un élément est recherché dans une page le timeout par défaut est appliqué, il est donc inutile et contre-productif de mettre des cmds « wait » dans un script de test.

# Robot Selenium Library

- Les commandes annexes:
  - Add Cookie, Capture Page Screenshot, Create WebDriver, Execute Javascript, Get Cookies, Get Selenium Speed, Go Back, Go To, Log Source, Set Browser Implicit Wait, Set Screenshot Directory, Set Selenium Speed ...
  - Permettent de gérer le comportement du navigateur et de selenium.