

# Conception d'Applications Interactives : Applications Web

## Séance #1 - Côté navigateur

2/3 - HTML5, CSS3, Bootstrap

# HTML5

- Le buzzword - HTML5, c'est quoi ?
- HTML 5
- CSS 3
- Bootstrap

# HTML



## Le buzzword

## HTML5, c'est quoi ?



# L'HTML 5 n'est pas...

"L'HTML5, c'est un nouveau langage ?"

"Je débute, j'ai envie d'apprendre l'HTML5 directement,  
ça a l'air mieux que l'HTML d'avant."

"Pfff, moi qui venais d'apprendre l'HTML,  
je vais devoir tout réapprendre..."

**L'HTML5 n'est pas un nouveau langage**



# L'HTML5 est...

- Une évolution d'HTML 4
  - Qu'on a survolé précédemment
- Deux syntaxes : HTML5 et XHTML5
- Des nouvelles fonctionnalités
- Une couche d'application
  - Des APIs



# L'HTML5 et les standards

- W3C définit les standards du web
  - Chargé d'élaborer le standard HTML5
  - Processus très lent et bureaucratique

- WHATWG

*Web Hypertext Application Technology Working Group*

- Groupe dissident du W3C
  - Des développeurs des navigateurs
  - Approche pratique
- Les deux travaillent en parallèle
  - Sur le même document



# Alors, deux standards HTML5 ?

*Mais lequel a gagné ?*

- En 2012 le W3C et le WHATWG ont décidé de suivre des chemins séparés
  - W3C travaille pour un standard fixe
    - Un snapshot de l'état actuel : HTML5
  - WHATWG travaille sur un living standard
    - En évolution permanente : HTML
- Approches complémentaires



# Les nouveautés de HTML5

- Allègement du code
- Nouvelles balises sémantiques
- Disparition de balises de mise en forme
- Nouveau modèle de contenu
- Balises multimédia
- Formulaire avec sémantique
- Stockage local
- Glisser-Déposer
- Géolocalisation
- Websockets



# Les nouveautés de HTML5 - Allègement du code



- Allègement de l'entête `head`
  - Le `doctype`, les balises `meta`, l'encodage des caractères, les balises `style` et `script`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" >
<head>
    <meta http-equiv="Content-Type"
        content="text/html; charset=utf-8">
    <link rel="stylesheet" type="text/css"
href="design.css" />
    <script type="text/javascript" src="script.js"></script>
</head>
```

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet"
href="design.css" />
    <script src="script.js"></script>
</head>
```

- Simplifications en général
  - Certaines discutables  
(pas de `/>` pour balises vides)

# Les nouveautés de HTML5 - Nouvelles balises sémantiques



- Des balises avec du sens sémantique
  - Plus spécifiques que les génériques
  - Structuration du document

A la place d'un `<div id="header">` pour l'entête de la page utilisons un `<header>`

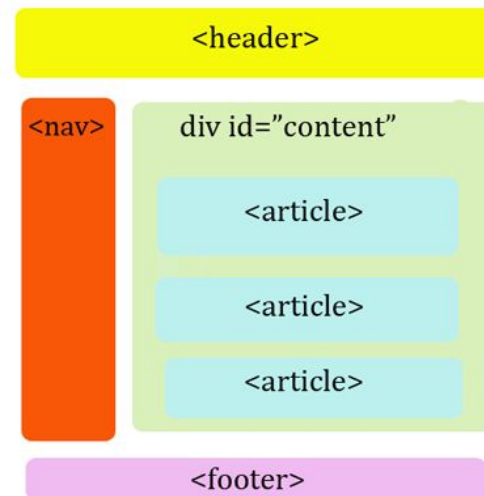


Image : [Alsa Creations](#)

- `<header>` : indique une en-tête
- `<footer>` : indique un pied de page
- `<nav>` : indique un élément de navigation (menu...)
- `<aside>` : indique une zone secondaire (sidebar, publicité...)
- `<section>` : indique une portion de la page
- `<article>` : indique une portion de la page avec du sens en lui-même

Oh, c'est EXCITANT... si c'est ça l'HTML5, je vois pas trop le buzz...

# Les nouveautés de HTML5 -

## Disparition de balises de mise en forme



- Meilleure séparation entre forme et contenu
- Disparition de balises sans sens sémantique
  - Telles que `center`, `font`, `big`, `strike` ou `u`
- La mise en forme se fait avec les CSS

# Les nouveautés de HTML5 - Balises multimédia



Pour mettre enfin de l'audio et de la vidéo  
**SANS FLASH !**

- `<video>` : introduit un lecteur vidéo ayant une URL comme source
- `<audio>` : introduit un lecteur audio ayant une URL comme source
- `<canvas>` : introduit une surface de dessin
  - Dessiner, tracer des formes, les animer...

**<Canvas>** ouvre la porte aux jeux  
Surtout couplé avec WebGL

# Les nouveautés de HTML5 - Formulaires avec sémantique



- Des nouveaux types pour la balise `<input>`

*Contrôles de surface*  
*Claviers software adaptés*

- `tel`
- `email`
- `url`
- `date`, `day`, `month`, `year`, `week`
- `number`
- `range`
- `search`
- `color`

# Les nouveautés de HTML5 - Stockage local



- Stocker des informations côté navigateur
  - Système clé-valeur
  - Chaque domaine a son *sandbox*
- Applications web déconnectés

Stocker une valeur :

⇒ `{`  
`localStorage['maCle'] = "Ma valeur";`    ou  
`localStorage.setItem("maCle", "Ma valeur");`

Récupérer une valeur :

⇒ `{`  
`localStorage['maCle'];`    ou  
`localStorage.getItem('maCle');`

Effacer une clé :

⇒ `removeItem("maCle");`

Tester si le navigateur  
supporte le stockage local  
:

⇒ `{`  
`if (localStorage) {`  
`// Le navigateur supporte le localStorage`  
`} else {`  
`// localStorage non supporté`  
`}`

# Les nouveautés de HTML5 - Géolocalisation



- Spécification W3C propre associée à HTML5
- Permet de géolocaliser le navigateur
  - GPS, triangulation GSM, triangulation wifi, adresse IP
- Pour Wifi et IP, utilisation de BDD de géolocalisation
  - E.g : `https://www.google.com/loc/json`
- API asynchrone

```
function maPosition(position) {  
    var infopos = "Position déterminée :\n";  
    infopos += "Latitude : "+position.coords.latitude +"\n";  
    infopos += "Longitude: "+position.coords.longitude+"\n";  
    infopos += "Altitude : "+position.coords.altitude +"\n";  
    document.getElementById("infoposition").innerHTML = infopos;  
}  
// Pour connaître la position  
navigator.geolocation.getCurrentPosition(maPosition);  
// Pour suivre la position  
var survId = navigator.geolocation.watchPosition(maPosition);  
// Pour annuler le suivi de position  
navigator.geolocation.clearWatch(survId);
```

# Les nouveautés de HTML5 - Glisser-Déposer



- Permet de déplacer des éléments entre des applications et le navigateur
  - API JavaScript native HTML5
- Attribute **draggable** : élément déplaçable
- Évènement **dragstart** : généré au début du transfert
- Évènement **dragover** : généré au survole d'un élément pendant la glisse
- Évènement **drop** : généré en fin de transfert

*Gare aux mauvais jeux de mots...*

```
function dragstart(target, e) {  
    e.dataTransfer.setData('text/plain', "Texte transmis");  
}  
function dragover(target, e) {  
    e.preventDefault(); // Annule l'interdiction de drop  
}  
function drop(target, e) {  
    e.preventDefault(); // Annule l'interdiction de drop  
    alert('Vous avez bien déposé votre élément !');  
}
```

```
<div id="source" draggable="true" ondragstart="dragstart(this,event)"> Élément source </div>
```

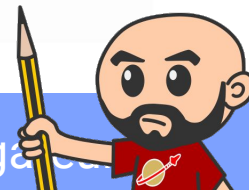
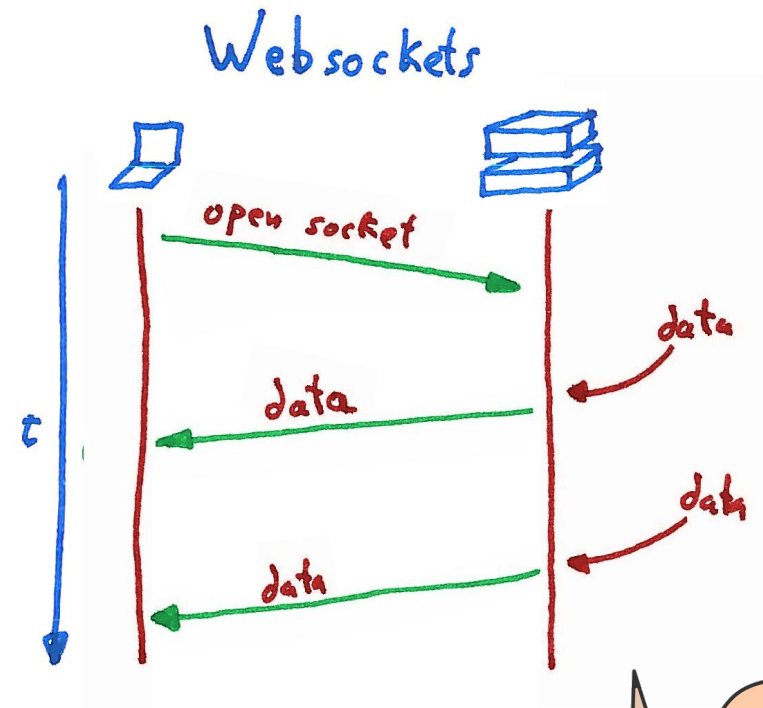
```
<div id="target" ondragover="dragover(this,event)" ondrop="drop(this,event)"> Cible </div>
```



# Les nouveautés de HTML5 - Websockets



- HTTP standard : requête-réponse
  - Du navigateur au serveur
- Websockets :  
communication  
bidirectionnelle
  - Plus besoin de  
*polling, long-polling*  
ou autres



**CSS**



**CSS3**



# Les CSS3, c'est quoi ?

*"C'est quoi le CSS3 ? Ca a un rapport avec HTML5"*

- Le CSS3 n'est pas forcément lié à HTML5
- Une évolution majeure des CSS
  - Nouveaux sélecteurs
  - Nouvelles façons de spécifier les couleurs
  - Détection des caractéristiques des terminaux
  - Des calculs dans la feuille de style
  - Des SVG en arrière plan
  - ...

*Comme les deux sont nouveaux  
on a tendance à les associer...*

# Exemple classique : Les coins arrondis



- Boîte avec coins arrondis avant CSS3
  - Une `<table>` avec 9 cases, chacune avec des images de fond
- Boîte avec coins arrondis avec CSS3

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8" />
  <style>
    .boite_arrondie {
      background: #eeeeee;
      border: 2px solid black;
      border-radius: 20px;
      width: 200px; height: 80px;
      margin: auto; padding: 20px;
    }
  </style>
</head>
<div class="boite_arrondie">Oh la jolie boîte !</div>
</html>
```





# Les nouveautés CSS3

- Effets visuels
- Sélecteurs
- Nouveaux outils

# Les nouveautés CSS3 - Effets visuels



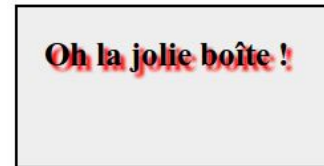
- border-radius

```
border-radius: 20px;
```



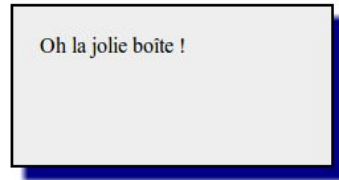
- text-shadow

```
text-shadow:  
4px 4px 3px #ff0000;
```



- box-shadow

```
box-shadow:  
10px 10px 5px #000088;
```



- font-face

```
@font-face {  
    font-family: 'Luckiest Guy';  
    src:url("luckiest-guy-regular.otf")  
}  
...  
font-family:'Luckiest Guy';
```



# Les nouveautés CSS3 - Effets visuels



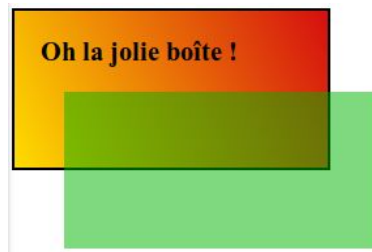
- gradient

```
background-image:  
    linear-gradient(right top,  
        #D60F0F 0%, #FFDD00 100%);
```



- opacity

```
background: rgba(0, 180, 0, 0.5);
```



# Les nouveautés CSS3 - Effets visuels : transform



- transform : rotate

```
transform: rotate(30deg)
```



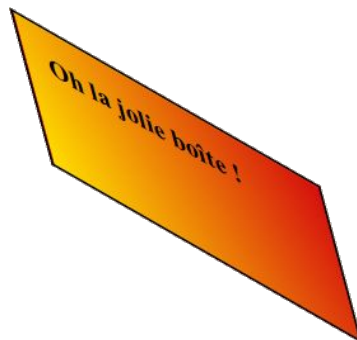
- transform : scale

```
transform: scale(1,0.25)
```



- transform : skew

```
transform: skew(15deg, 30deg);
```



- transform : translate

```
-webkit-transform: translate(30px, 30px);
```



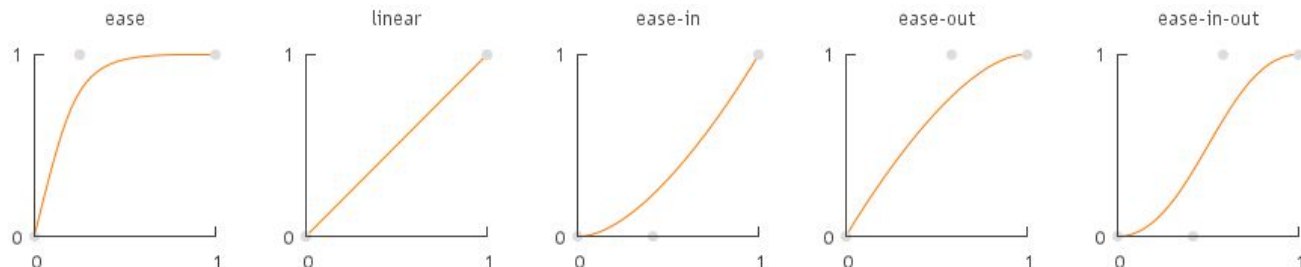


# Les nouveautés CSS3 - Effets visuels : Transitions



## ● Des propriétés

- **transition-property** : Propriétés CSS à transformer
  - couleurs, position, dimensions, transformations, visibilité, ombres, dégradés
- **transition-duration** : Durée de la transition
  - en secondes ou millisecondes
- **transition-timing-function** :
  - Fonction de transition, modèle d'interpolation (accélération, décélération...)



- **transition-delay** : Retard du départ de la transition
  - en secondes ou millisecondes

# Les nouveautés CSS3 - Effets visuels : Transitions



- Déclenchement

```
.transition {  
  background: #aaa;  
  transition-property : color;  
  transition-duration : 5s;  
  color: white;  
}  
.transition:hover {  
  transition-property : color;  
  transition-duration : 5s;  
  color: black;  
}
```



# Les nouveautés CSS3 - Transformées en 3D



- perspective

```
transform:  
    perspective(600px)  
    rotateX(40deg );
```



- rotateX, rotateY, rotateZ
- translateX, translateY, translateZ

# Les nouveautés CSS3 - Animations



- Des keyframes

```
@keyframes rotateCube {  
  0% {  
    transform: rotateX( 0deg ) rotateY( 0deg );  
  }  
  100% {  
    transform: rotateX( 360deg ) rotateY( 360deg );  
  }  
}
```

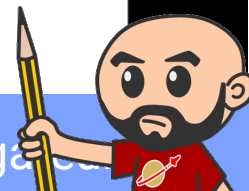
- Des animations

```
animation: rotateCube 8s infinite linear;
```

# Exemple : le cube des navigateurs



- `translateX`, `translateY`, `translateZ`
- `rotateX`, `rotateY`, `rotateZ`
- `animation`, `@keyframes`



# TD HTML5 : Le cube



# TD HTML5 : Le cube

## 1. Produire un document HTML5 avec un

élément **div** appelé *cube*

- Ouvrir le fichier avec le navigateur et vérifier qu'il est cohérent
- Valider le fichier avec [HTML5 Validator](#)
  - N'oubliez pas ajouter la balise **title**, obligatoire

## 2. Un cube a 6 faces : créer 6 éléments **div** à

l'intérieur de l'élément *cube*, tous appartenant à une classe *face*, chacun avec une **id** selon sa position :

**front, back, top, bottom, left, right**

- Les faces doivent avoir des dimensions 200x200 px, un fond gris (**background: #888**) et une bordure noire d'un pixel de large
  - Créer un élément **style** dans le head du document
  - Avec la définition de la classe *face*



# TD HTML5 : Le cube

3. Mettons les 6 *faces* les unes sur les autres :

- `position: absolute; top: 0; left: 0;`

4. Le *cube* va être centré

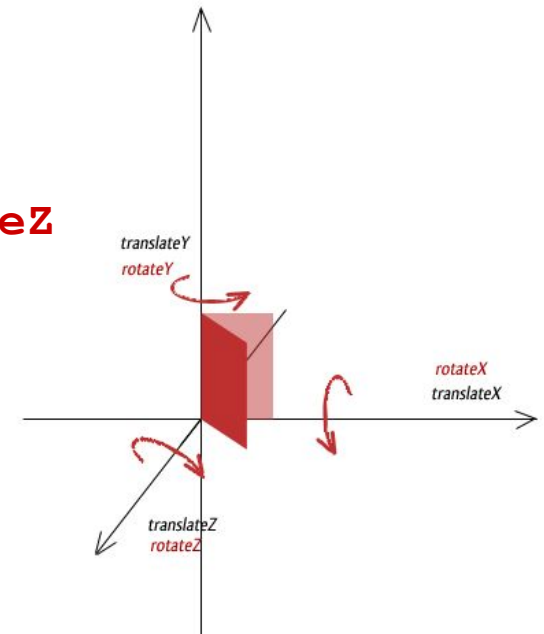
- `position: absolute; top: 200px; left: 400px;`

5. Passons sur une vue isométrique sur l'objet **cube**

- `translateX, translateY, translateZ`
- `rotateX, rotateY, rotateZ`

Exemple rotation :

`transform : rotateZ(65deg) ;`







# TD HTML5 : Le cube

6. Maintenant on a 6 faces à plat. Pour chacune des *face*, créer l'élément de style correspondant à son **id** et appliquer les transformations 3D pour le mettre à sa place

- Il ne faut pas oublier de positionner

**`transform-style: preserve-3d;`**

sur *cube*

- Pour que le cube tourne sur son centre, appliquez **`transform-origin:`**

**`transform-origin: 100px 100px;`**

- Commencez par les faces **front** et **back**

## TD HTML5 : Le cube

### 7. Ajoutez des images à chaque face

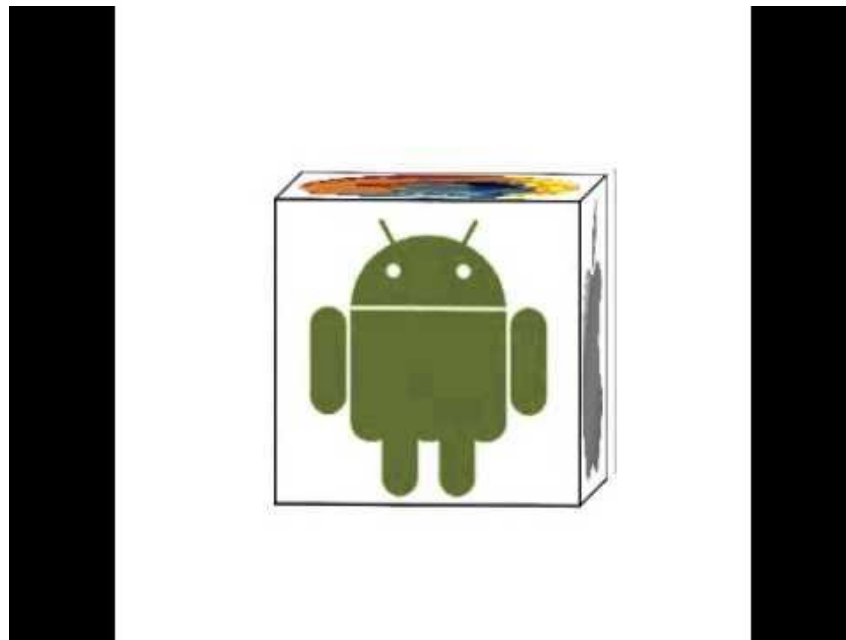
- Les images sont [ici](#)
- Bien vérifier l'orientation des faces
  - Les logos doivent être à l'endroit



## TD HTML5 : Le cube

### 8. Maintenant on le fait tourner !

- Des rotations sur le *cube*
- Utilisations d'**animation** et **@keyframe**
- Propriété utile : **transform-origin: 100px 100px;**



# Pour aller plus loin

- Pour apprendre et se tenir informés :  
[HTML5 Rocks!](#)





# Bootstrap

Car nous ne sommes pas  
tous des graphistes

# C'est très beau ça... mais le design graphique me fait peur !



- Si on fait du web en 2013 on ne peut pas se passer du HTML5/CSS3/JS
  - Mais la plupart des développeurs n'aiment pas concevoir des interfaces
  - On n'est pas des graphistes !
- Quoi faire ?
  - Utiliser Bootstrap



# C'est quoi Bootstrap

- Constats chez Twitter en 2009 :
  - Les développeurs d'applications web n'aiment pas faire de l'IHM web
  - A chaque nouvelle application, on refaisait une IHM
  - Soucis d'ergonomie, d'uniformité, d'esthétisme
- Solution : faire un boîte à utils HTML5/CSS3/JS
  - Pour IHMs performantes, ergonomiques et jolies
  - Simple à utiliser
    - Permettant à un développeur de faire des IHMs belles et efficaces
  - *Lingua franca* entre graphistes et développeurs

# La panoplie Bootstrap




The screenshot shows the Bootstrap website homepage. At the top is a dark navigation bar with links: Home, Get started, Scaffolding, Base CSS, Components, JavaScript, and Customize. The word 'Bootstrap' is on the right. The main content area has a large heading 'Introducing Bootstrap.' followed by the tagline 'Need reasons to love Bootstrap? Look no further.' Below this are three columns, each with an icon, a title, and a description. The first column features Twitter and GitHub icons, titled 'By nerds, for nerds.', and describes Bootstrap's origins. The second column features a monitor and smartphone icon, titled 'Made for everyone.', and describes its cross-device compatibility. The third column features a grid icon, titled 'Packed with features.', and lists its various components and tools.

Home Get started Scaffolding Base CSS Components JavaScript Customize Bootstrap


## Introducing Bootstrap.

Need reasons to love Bootstrap? Look no further.




### By nerds, for nerds.

Built at Twitter by [@mdo](#) and [@fat](#), Bootstrap utilizes [LESS CSS](#), is compiled via [Node](#), and is managed through [GitHub](#) to help nerds do awesome stuff on the web.



### Made for everyone.

Bootstrap was made to not only look and behave great in the latest desktop browsers (as well as IE7!), but in tablet and smartphone browsers via [responsive CSS](#) as well.



### Packed with features.

A 12-column responsive [grid](#), dozens of components, [JavaScript plugins](#), typography, form controls, and even a [web-based Customizer](#) to make Bootstrap your own.

- Des éléments de base, des composants, des widgets complexes...
- Modulable
- Responsive



# Facile à intégrer, facile à personnaliser



- Il suffit d'embarquer les JS et CSS Bootstrap
- Les composants sont des classes CSS
- Pas besoin de prise de tête avec la présentation
- Responsive, adaptable à tout terminal

Mais je veux ma présentation à moi

- Des thèmes
  - Permettant de garder la puissance Bootstrap
  - Adaptant le *look* à ce que je veux
  - Toujours sans prise de tête développeur



# *Lingua franca ?*

- Les graphistes web aiment Bootstrap
  - Ergonomie soignée
  - CSS claires et structurées
  - Simplicité d'adaptation et personnalisation des thèmes
- Les développeurs aiment Bootstrap
- Les graphistes et développeurs comprennent Bootstrap

Langage commun entre  
développeurs et graphistes
- Le graphiste fournit des maquettes en Bootstrap
- Le développeur adapte sa logique IHM à Bootstrap