

2014

GMIN 309 - TAWeb

RESPONSIVE 3

Bérenger ARNAUD
berenger.arnaud@itkweb.com



Sommaire (cours 4)

- évènements
- pointeurs sur fonctions (handlers)
- cycle de vie des composants
- glisser-déposer
- habillages



ÉVÈNEMENTS

Évènements

- événement → quelque chose s'est passé
- causes
 - framework
 - résultat d'exécution
 - vie d'un composant
 - ...
 - asynchrone
 - progression
 - résultat
 - échec
 - utilisateur
 - entrée utilisateur

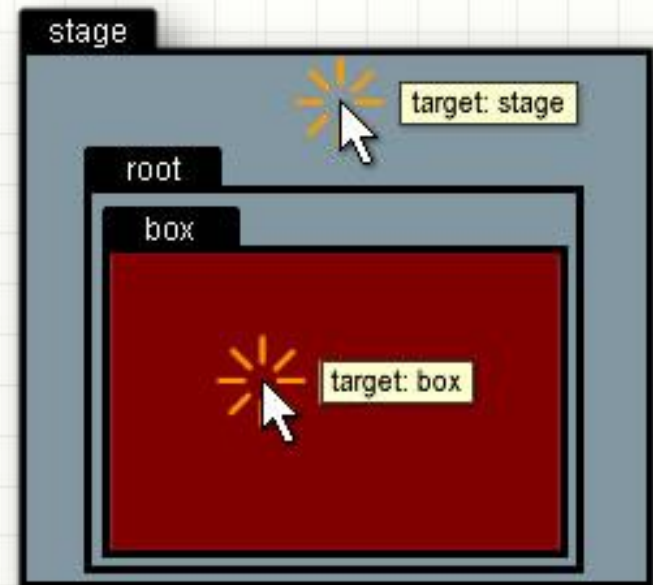


Image par Trevor McCauley (Adobe)

Actions

- Dispatcher
 - Mettre un écouteur
 - Enlever un écouteur
 - Présence d'un écouteur
- Laissons
 - liaison fortes (**par défaut**)
 - liaison faibles (**JS : bientôt**)
- Diffusion
 - priorité (**Priority**)
 - ordre d'ajout des écouteurs

```
<fx:Metadata>  
    [Event(name="submitted", type="event.Event")]  
</fx:Metadata>
```

```
document.addEventListener("click", function(){  
    document.getElementById("demo").innerHTML = "Hello World";  
});
```

Exemples

```
<Button id="myButton" click="Alert('Clique sur le Bouton')"/>
```

```
<Button id="myButton" click="myButton_clickHandler(event)"/>
```

```
function myButton_clickHandler(event:MouseEvent):void {  
    Alert ('Clique sur le Bouton');  
}
```

```
myButton.onClick = function(event:MouseEvent):void {  
    Alert ('Clique sur le Bouton');  
}
```

```
myButton.addEventListener('click', function(event:MouseEvent):void {  
    Alert ('Clique sur le Bouton');  
});
```


Cycle

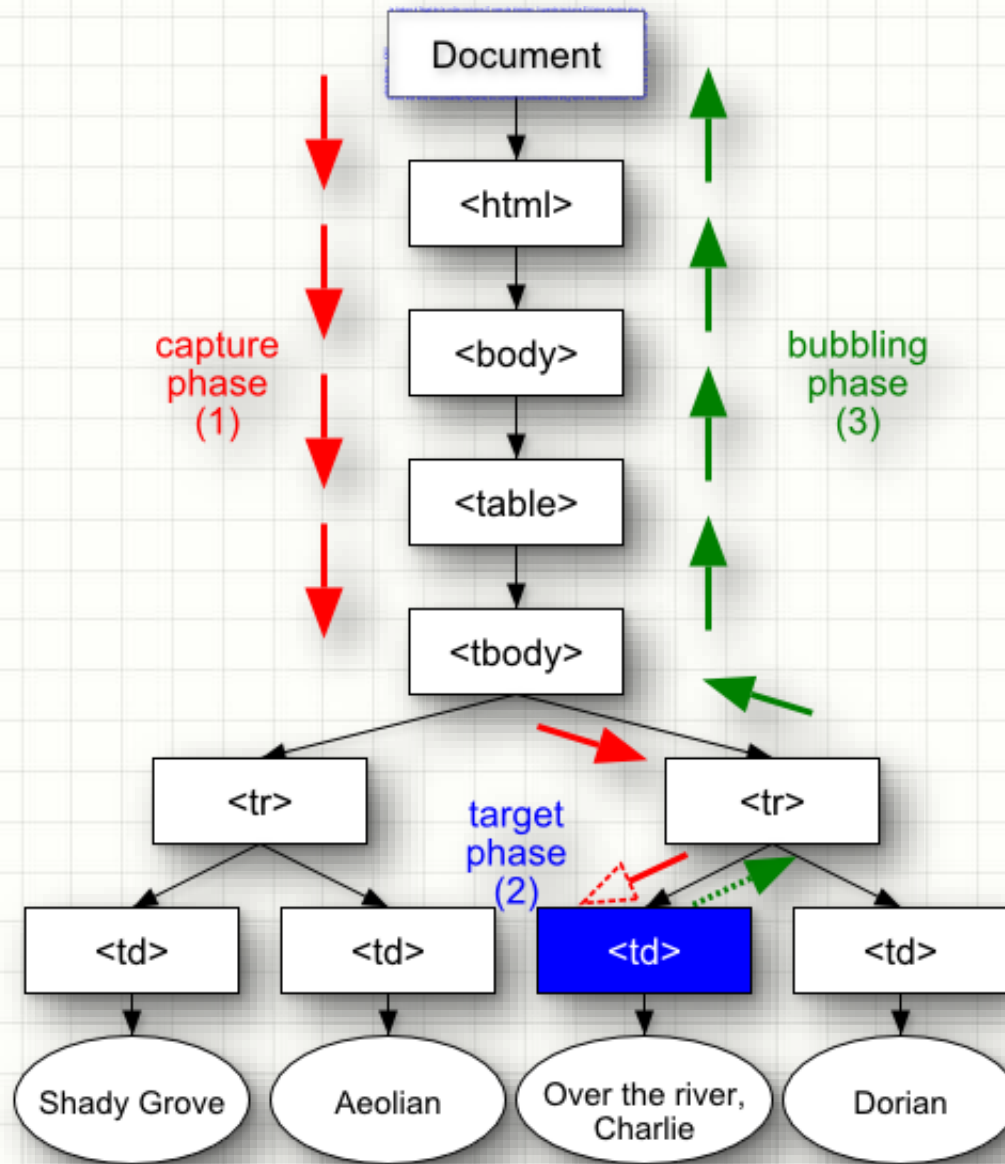
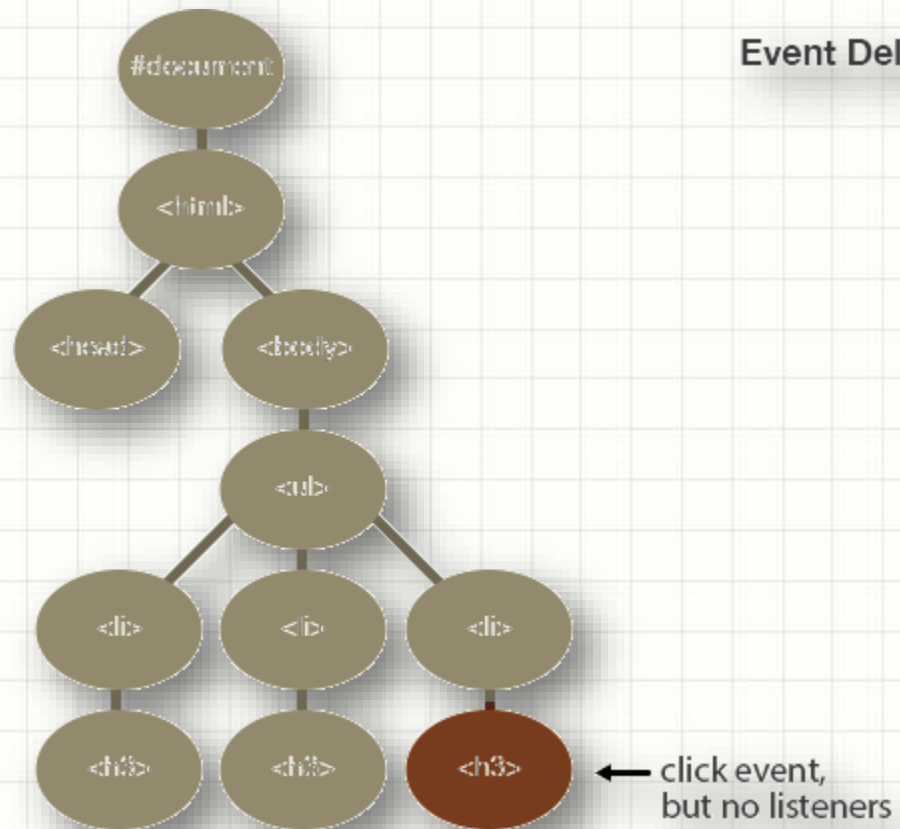


image W3C

Écoute



Stopper

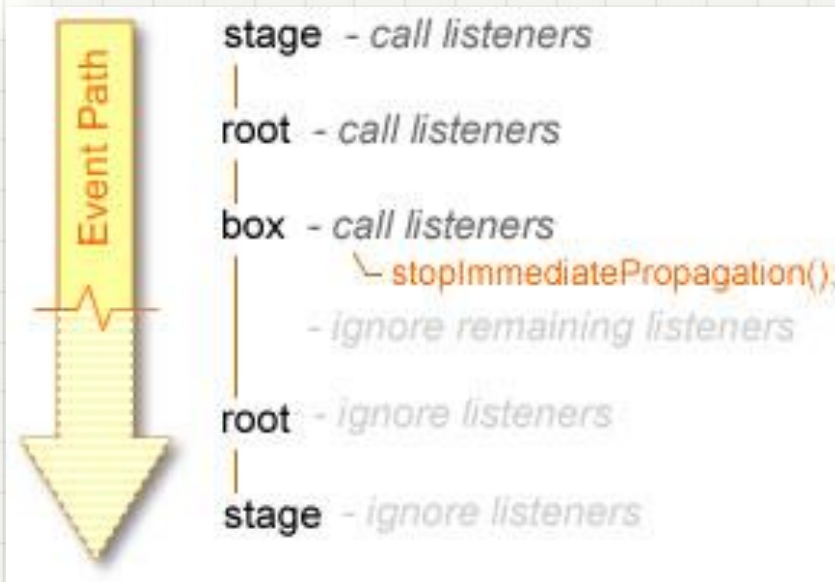


Image par Trevor McCauley (Adobe)

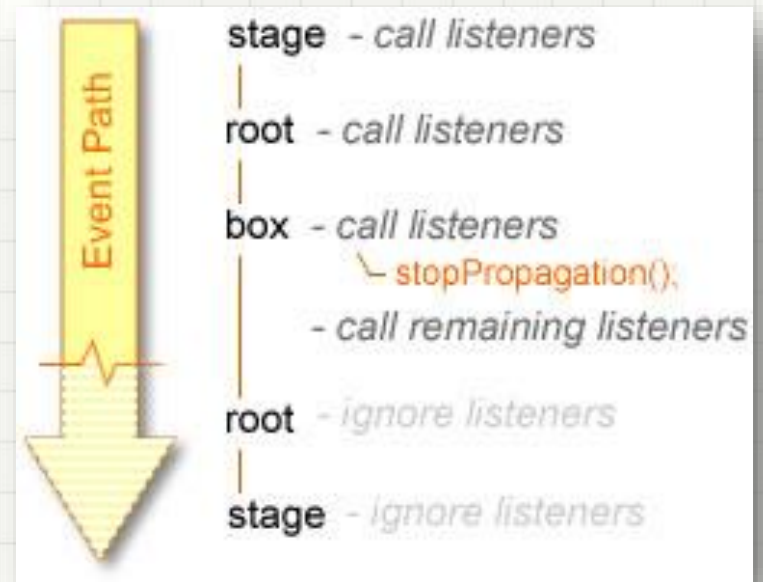


Image par Trevor McCauley (Adobe)

| | |
|--------------------------------|---|
| <code>preventDefault()</code> | To cancel the event if it is cancelable, meaning that any default action normally taken by the implementation as a result of the event will not occur |
| <code>stopPropagation()</code> | To prevent further propagation of an event during event flow |



ÉVÈNEMENTS VERSUS POINTEURS FONCTION

Évènements et pointeurs

| | Évènements | Pointeurs sur fonctions |
|-------------------|------------------|--|
| Classe | Event et dérivés | Function |
| Efficacité | lourd | spartiate |
| Écouteur | normalisé | libre |
| Dispatch | multiple (natif) | simple (natif) ou multiple (à implémenter) |
| Type de dispatch | (a)synchrone | synchrone |
| Ordre de dispatch | approximatif | maîtrisable |
| Support AS | écouteurs | pointeur |
| Support MXML | automatique | AS en ligne |

Data binding

- ActionScript

- Utilitaire

```
BindingUtils.bindProperty(replica, "text", textarea, "text");
```

- "À la main"

```
textarea.addEventListener("textChanged", textareaTextChangeListener);  
textareaTextChangeListener(event : Event) {replica.text = textarea.text; }
```

- MXML

- Simple

```
<mx:TextArea id="textarea" />  
<mx:TextArea id="replica" text="{textarea}" />
```

- Double

```
<mx:TextArea id="replica1" />  
<mx:TextArea id="replica2" text="@{replica1}" />
```

JAVASCRIPT
→ framework



VIE DES COMPOSANTS

Phases

- Composants → 3 phases
 - Naissance
 - Construction appel au constructeur
 - Configuration assignation des propriétés & styles
 - Attachement ajout du composant à la displayList
 - Initialisation phase d'initialisation
 - Vie ∞
 - Invalidation différer les processus de mise à jour
 - Validation mise à jour et (re)construction
 - Interaction événements du composant
 - Mort
 - Détachement suppression de la displayList
 - Poubelle libération mémoire

Cycle de vie

Création

- Création
 - constructeur
- Initialisation
 - pré-initialisation
 - graphismes
 - initialisation
- Validation
 - propriétés
 - taille(s)
 - rendu
 - enfants
- Finalisation
 - créé
 - à jour

Vie

- Invalidation
 - graphismes / textes
 - enfants
 - propriétés
 - taille
- Validation
 - graphismes / textes
 - enfants
 - propriétés
 - taille

Mort

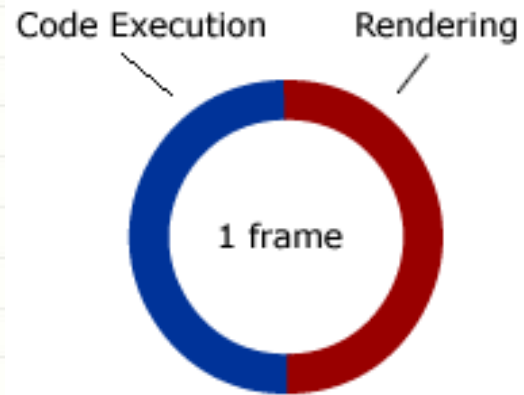
- Destruction
 - destructeur

Vie d'un composant

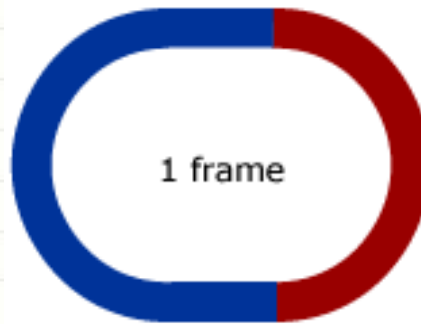


Schéma de Deepa Subramaniam

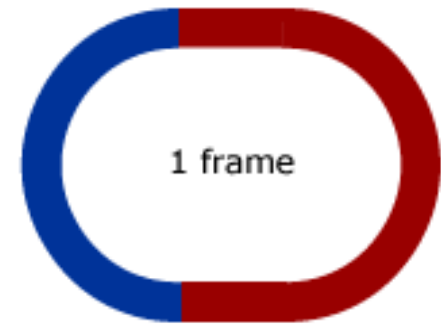
Proportion code/rendu



Heavy Code Execution



Heavy Rendering



Images de Sean Christmann



GLISSER - DÉPOSER

HTML ? JS ?

- les deux !

Youhou !

- HTML

- propriété : `draggable="true"`

- JS w3schools.com/html/html5_draganddrop.asp

- allow
 - drag
 - drop



HABILLAGE

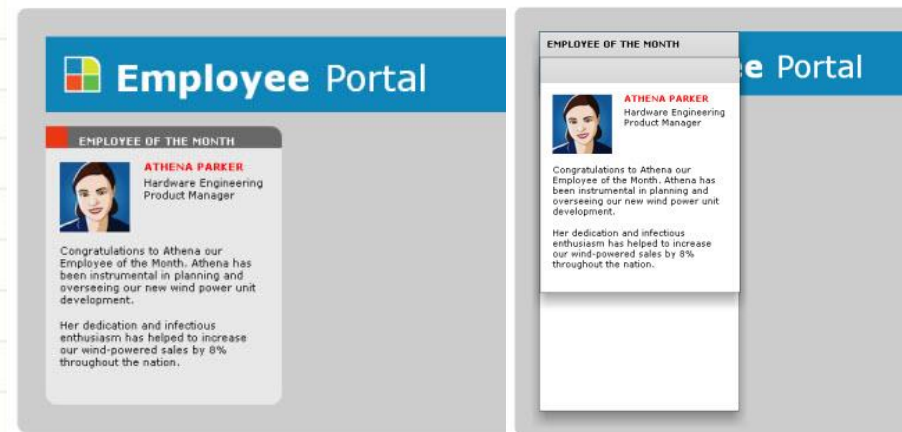
Habillage / Skin

- Définition
 - changement apparence par remplacement des objets visuels
- inclus
 - tous éléments graphiques
 - layout
- Utilité
 - UX
 - différentiation
 - charte graphique
 - fonctionnalité
design
 - dynamisme
- Outil bonus
 - Flash Catalyst

Examples



images par Adobe



Pour Flex

- Remplacement
 - classe
 - Flash (swf)
 - image (png, gif, ...)
- Par code
 - créer classe habillage
 - surcharge méthode
 - de rendu
 - de placement
 - API de dessin (pinceau)
 - scènes 2D/3D
- Compatible
 - thèmes
 - styles
 - états
 - total (automatique)
 - partiel / redéfini
 - héritage
- Affectation
 - CSS / Style
 - MXML
 - AS

Pour JS





**AVEZ-VOUS DES
QUESTIONS ?**

TD / TP

- créer un composant avec x-tags [x-tags.org]
 - carrousel simple ^^
 - éléments : cartes de visite
 - image
 - nom
 - description
 - simplification : un seul élément (au lieu du carroussel)
- utiliser une propriété **state** pour forcer un état
 - états : seulement les images, ..., tout
- bonus : responsive automatique
 - tailles : bureau (grand), tablette, téléphone (petit)