

2014

GMIN 309 - TAWeb

RESPONSIVE 2

Bérenger ARNAUD
berenger.arnaud@itkweb.com



Présentation

- Béranger ARNAUD
- Ingénieur de recherche RIA
 - Projets Web multi-technologies
 - Expérimentations
- Docteur IHM
 - Visualisation de données inconnues
 - Interaction en collaboration

Organisation

Date		Créneau 1	Créneau 2	Créneau 3
9 octobre	responsive 2	cours	TD / TP	
16 octobre	responsive 3	cours	TD / TP	
23 octobre	mobile, 3D	cours	TD / TP	
6 novembre		soutenance projet 1		
13 novembre	data driven	Cours	TD / TP	
20 novembre	asynchrone	Cours	TD / TP	

- Salles :
 - cours 1.07
 - TD/TP bat. 16 salle informatisée

Découpage

1. ~~intro web~~
2. ~~web responsive 1~~
3. web responsive 2
4. web responsive 3
5. mobile, 3D
6. data driven
7. asynchrone
8. cloud 1
9. cloud 2

Sommaire (cours 3)

- présentation générale
- Flash/Flex
- responsive 1 → 2
- composants
- conteneurs et placements
- bonus Flex : data container



PRÉSENTATION

RIA ?

plugin

- navigation
 - barres
 - scripts
- navigateur
 - Flash
 - Silverlight
 - Java Applet
- système
 - AIR
 - JavaFX

« natif »

- commun
 - HTML + CSS
 - JavaScript
 - « HTML 5 »
- navigateur

Historique

1990

- Applet

1991 ~ 94

- HTML : tableaux & CSS

1995

- HTML : tables
- FutureSplash Animator
- Shockwave
- Java applet

1996

- Flash

Objectifs initiaux :

HTML

présentation de texte

Applet

application intégrée

Shockwave

contenu avancé

Flash

rendu vectoriel

Historique

1997

- HTML : CSS + scripts
- Java : nouvelle AWT

1998

- Shockwave : audio
- Java : SWING

1999

- Flash : audio

2000

- Java : audio

Nouveaux objectifs :

HTML

présentation de contenu

Applet

calcul + rendu élaborés

Shockwave

contenu professionnel

Flash

uniformisation UX

Historique

2001

- Shockwave : 3D

2002

- Java : Image IO + Web Start

2007

- HTML 5

2011

- Shockwave : C++
- Flash : Falcon compiler

2012

- Flash : 3D native GPU
- Flex+Falcon : libre @ Apache

Aspirations :

HTML

mise en forme riche

Applet

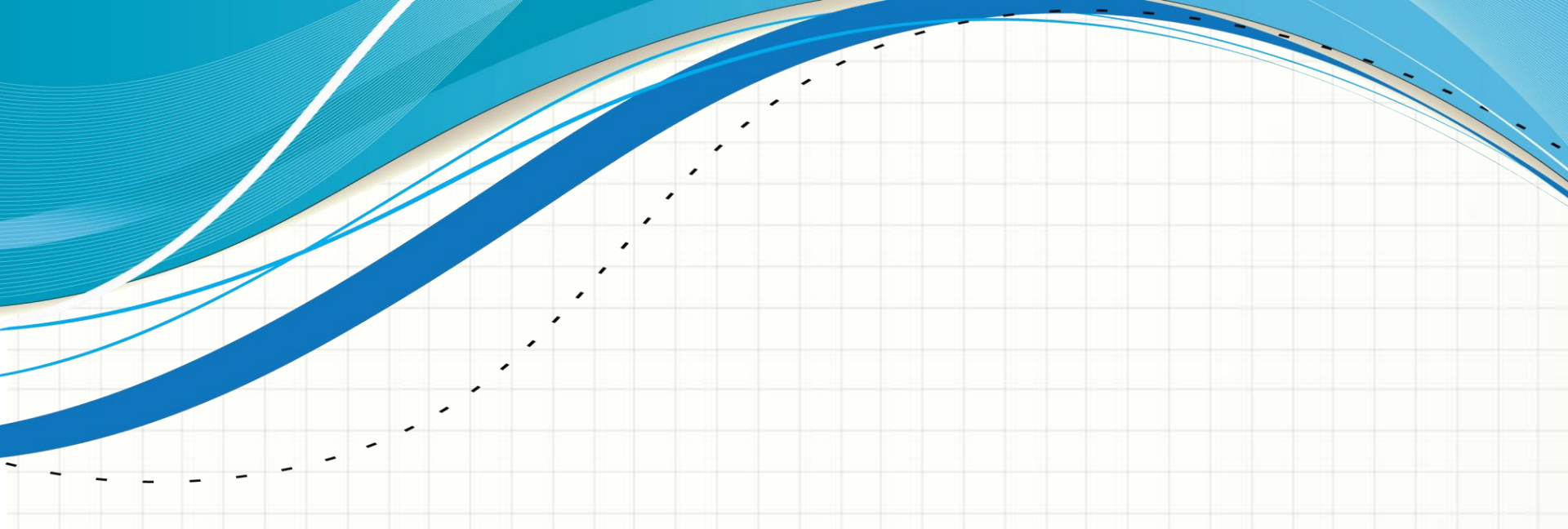
→ disparition

Shockwave

→ fusion Flash

Flash

RIA + Jeux



FLASH + FLEX

Flash ?

EST

- environnement (concept)
- machine virtuelle
 - Flash Pro / AS / C
- rendu
 - vectoriel / multimédia
- multi plateformes
- multi médias
 - Open Screen Project

N'EST PAS

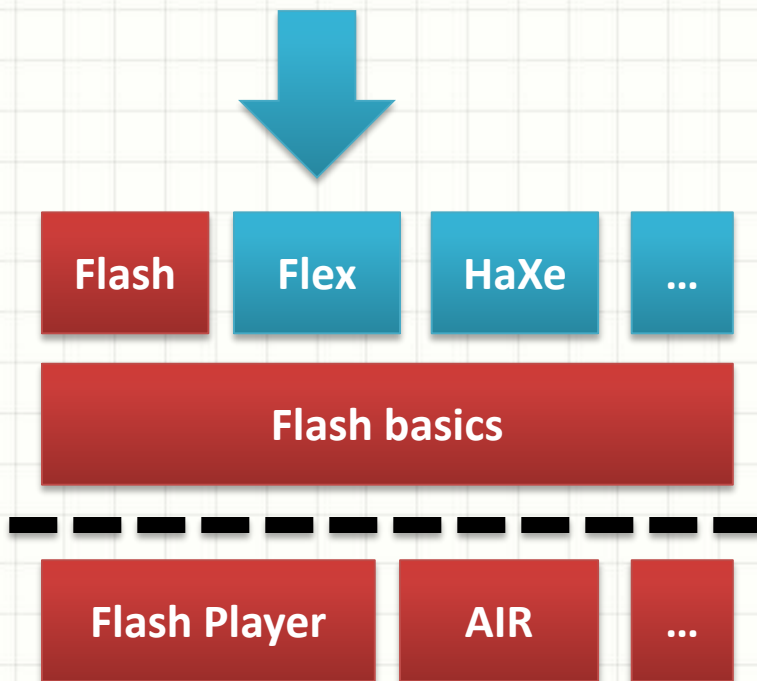
- concurrent HTML+CSS+JS
 - Web x.x
- interprété
- essentiel

Flex ?

- environnement / framework
 - surcouche de Flash
 - librairies
- programmation (patterns)
 - MVC composants graphiques
- compilateurs
 - Falcon et FalconJS
- communauté
 - programmeurs vs. designers

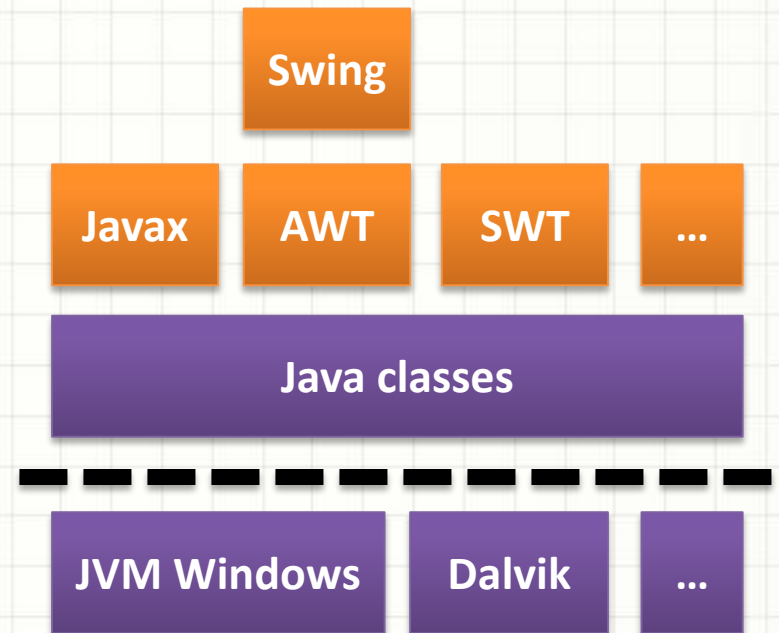
Architecture

Flash



■ Propriétaire
■ Ouvert ou Libre

Java



■ Sous licence
■ Libre

Historique Adobe

2004 : Flex 1

- serveur

2005 : Flex 2

- client
- composants graphes
- vm intégrée au plugin Flash

2007 Flex 3 (Moxie/Halo)

- composants graphiques
- Action Script 3
- AMF 3
- nouvelle vm
- AIR
- Flex Builder 3

2010 : Flex 4 (Spark)

- composants graphiques
 - séparation du rendu
- Flash Builder 4
- Flash Catalyst 1

2011 : Flex 4.5

- composants mobile
- AIR mobile
- Flash Builder 4.5
- Flash Catalyst CS 5.5

2012 : Flex 4.6 ~ 4.9

- composants mobile
- abandon propriété

Historique Apache

2012 : Flex 4.09

- reprise Apache

2013 : Flex 4.10

- travaux communauté
 - bug fix
 - nouveaux composants

2014 : Flex 4.13

- FlexJS (FalconJS)
- FlexJS Mobile (PhoneGap)

Compilateur Flex

- Flex SDK
 - classes AS
 - compilateur MXMLC
 - émulateur ADL
- <https://flex.apache.org/>
- <https://opensource.adobe.com/>
- destinations de compilation :
 - embarqué : Flash Player
 - multiplateforme : AIR
 - natif : AIR et AIR Mobile
 - conversion : HTML/JS/CSS

Environnement de développement

Flash Builder

- plus complet et efficace
- payant
- adobe.com/go/flashbuilder

FB4Linux

- fork de Flash Builder
- gratuit
- fb4linux.googlecode.com

FlashDevelop

- alternative aboutie
- Windows
- open source
- flashdevelop.org

FDT

- alternative aboutie
- multi-OS, HaXe
- version communautaire
- fdt.powerflasher.com

Essentiel

- Documentation

- flex.apache.org/asdoc/
- help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/

Package and class filters:

- ▶ Runtimes: AIR 1.5 and earlier, Flash Player 10 and earlier
- ▶ Products: Open Source Media Framework 1.0

- Démonstrations

- Tour de Flex : flex.apache.org/tourdeflex/

- S'entraîner

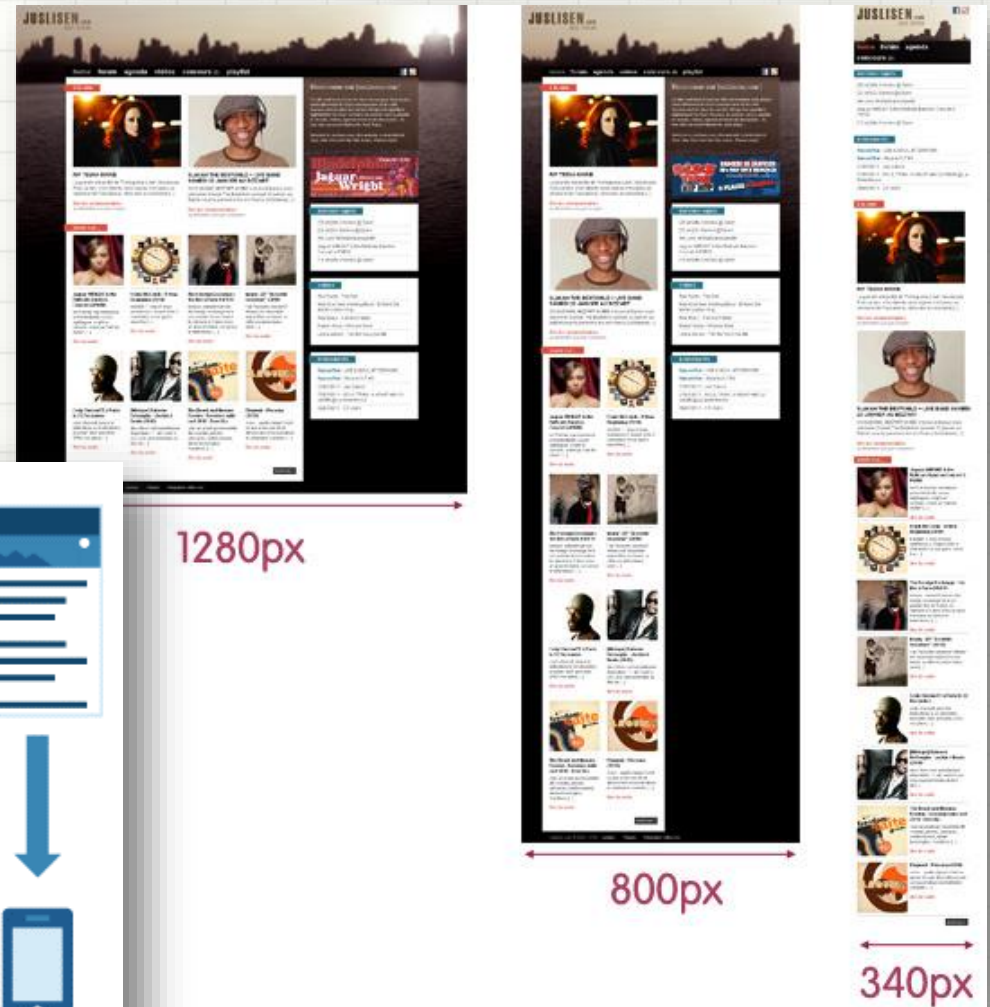
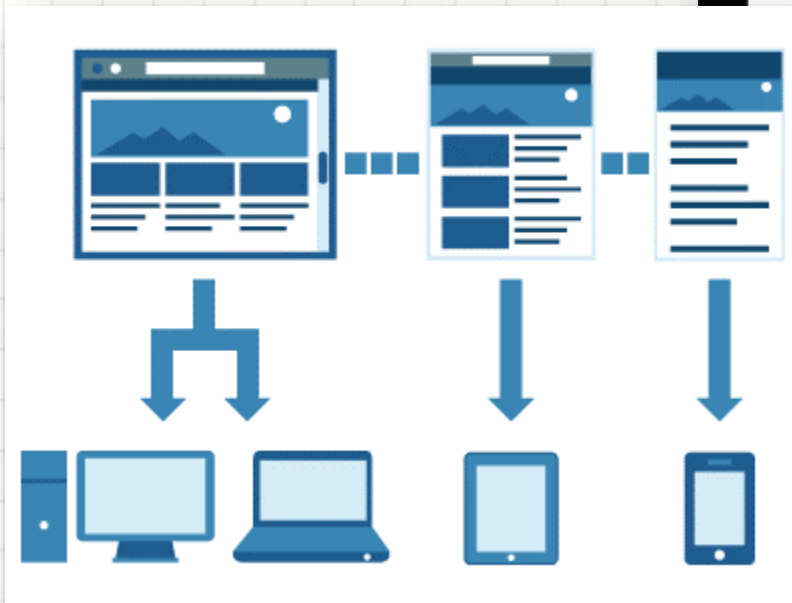
- Test drive (en une heure)
adobe.com/devnet/flex/testdrive.html
- Flex in a week
adobe.com/devnet/flex/videotraining.html



RESPONSIVE 1 → 2

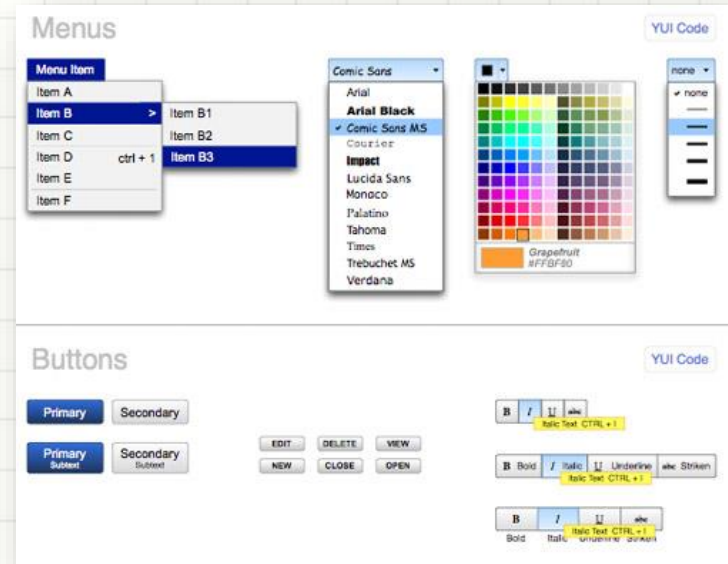
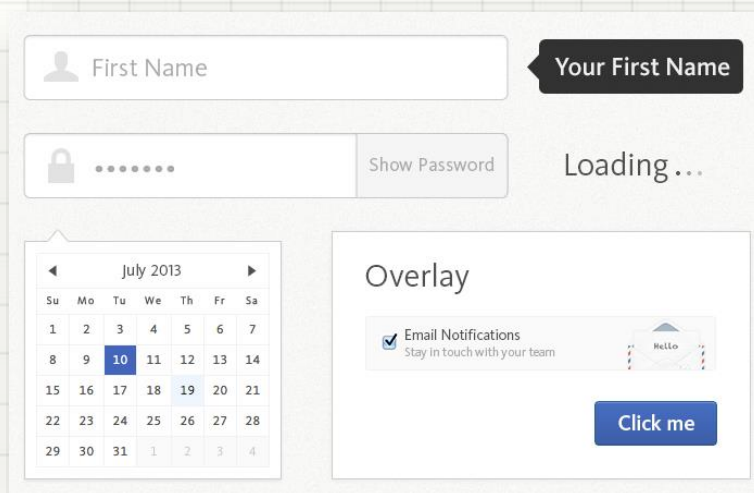
Adaptatif : placement

- placement
- CSS media queries



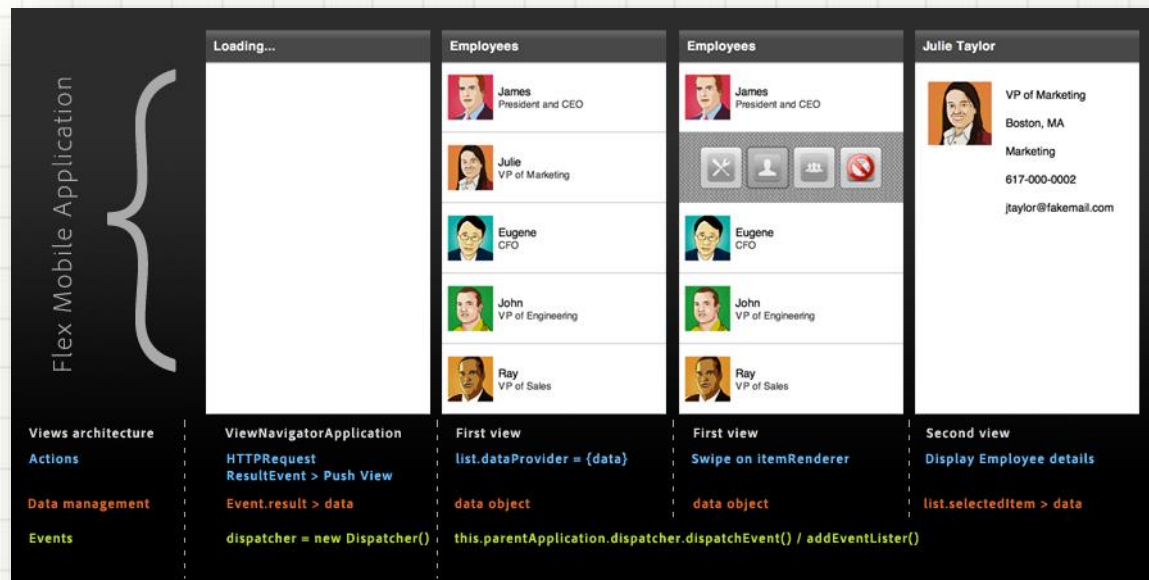
Adaptable 1 : UI (composant)

- change composant
- Web Components [draft]



Adaptable 2 : UX (scénarisation)

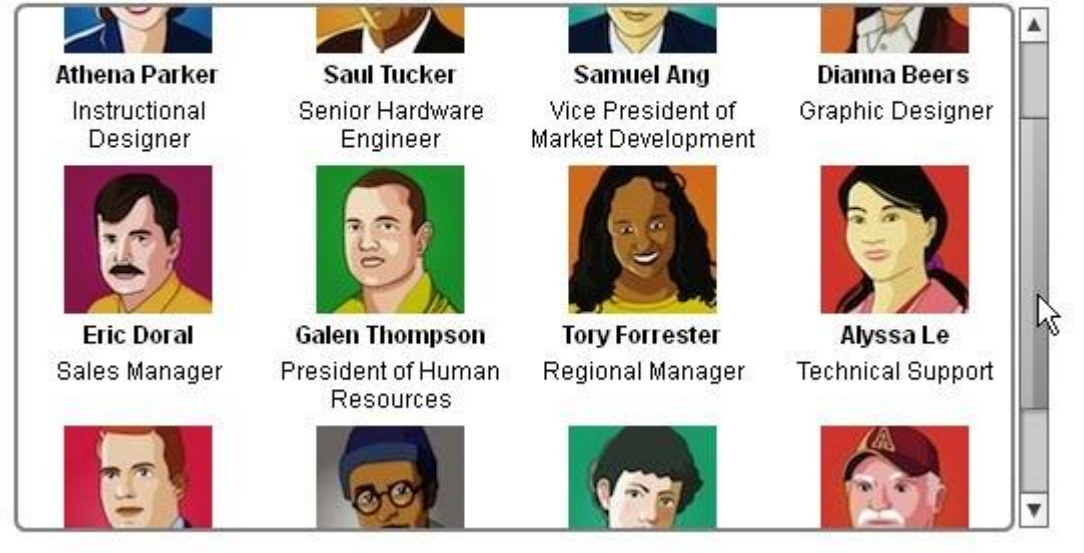
- réécrit l'interaction
- Flash : OK
- HTML : ??



Composant

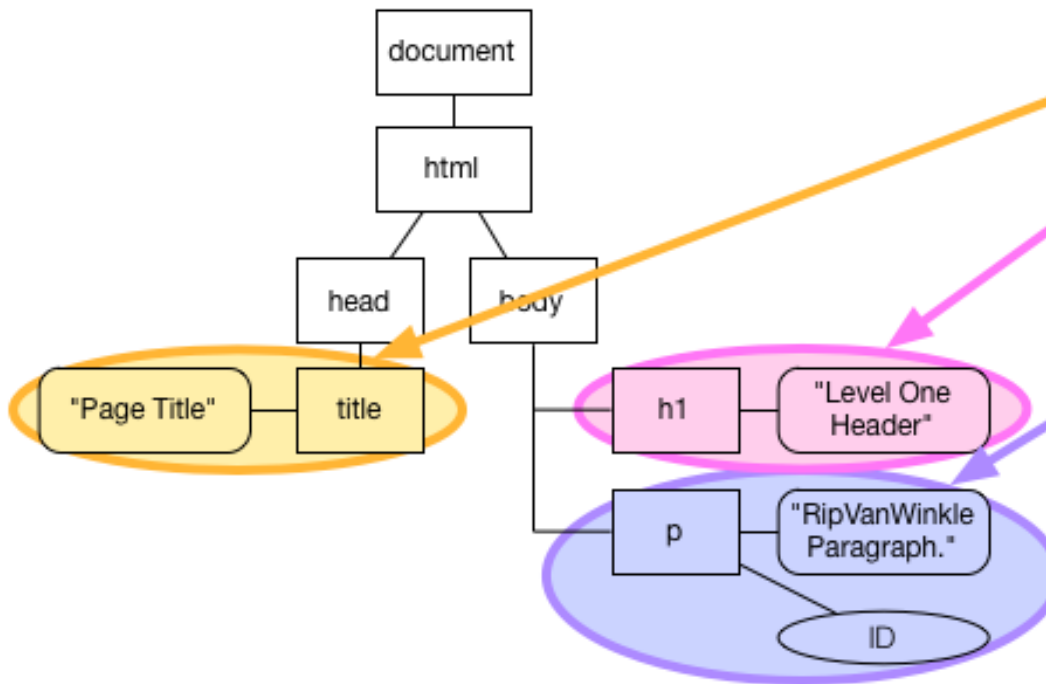
Component	Composant
Container	Conteneur
Layout	Placement
Renderer	Rendu
Skin	« Peau »
Theme	Thème

Employee Portal: Employee Directory



Hiérarchie des composants

DOM tree for simple.html



simple.html

```
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>Level-One Header</h1>
  <p id="RipVanWinkle">
    RipVanWinkle paragraph.
  </p>
</body>
</html>
```



COMPOSANTS : PROPRIÉTÉS CLÉS

Identification

<i>pointeur</i>		<ul style="list-style-type: none">• automatique• non accessible
id		<ul style="list-style-type: none">• nom de l'instance• utile pout tout
<i>caractéristique(s)</i>		<ul style="list-style-type: none">• cible un ensemble• chainage et raffinage
ne pas utiliser	uid	<ul style="list-style-type: none">• identifiant unique interne (graphique)
	name	<ul style="list-style-type: none">• identifiant interne
	automationName	<ul style="list-style-type: none">• identifiant interne dédié aux tests automatique

Taille

width

- largeur en pixels
- **graphique** : largeur en %

minWidth, maxWidth

- largeur en pixels

percentWidth

- largeur en %
- lecture à la création

height

- hauteur en pixels
- **graphique** : hauteur en %

minHeight, maxHeight

- hauteur en pixels

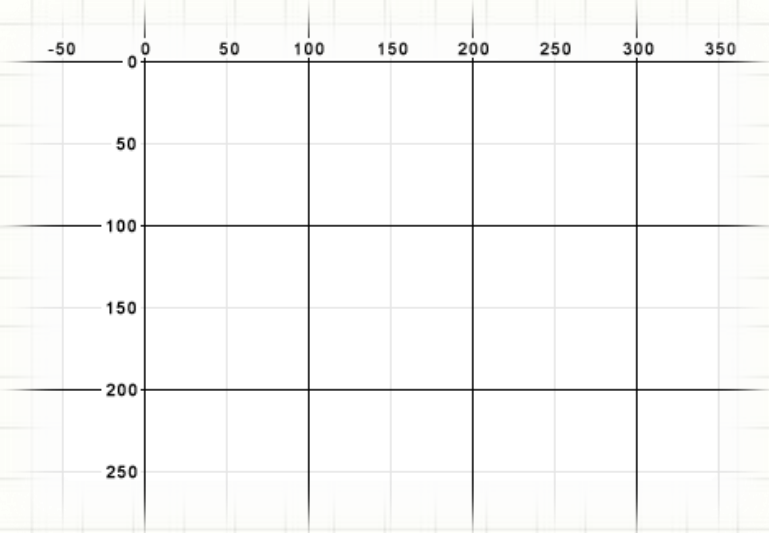
percentHeight

- hauteur en %
- lecture à la création

Position

x, y

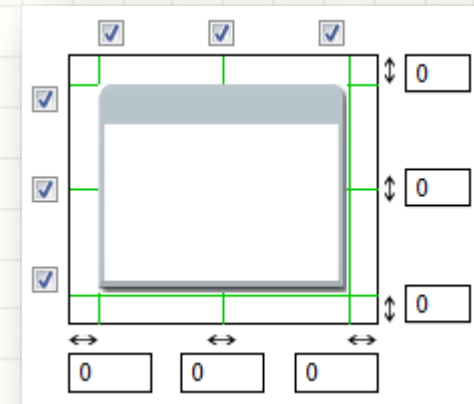
- coordonnées absolues
- point haut gauche



top, right, bottom, left

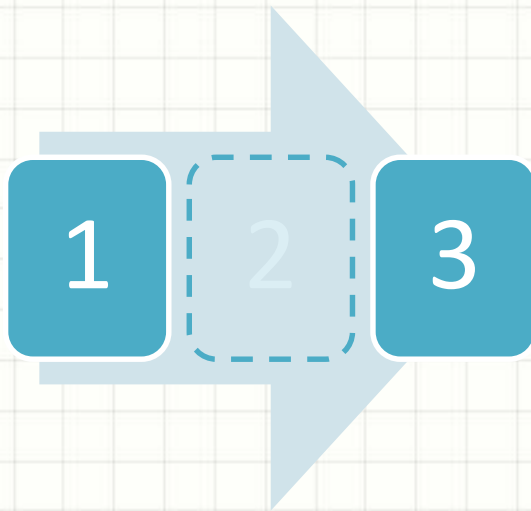
horizontalCenter, verticalCenter

- écarts absolus

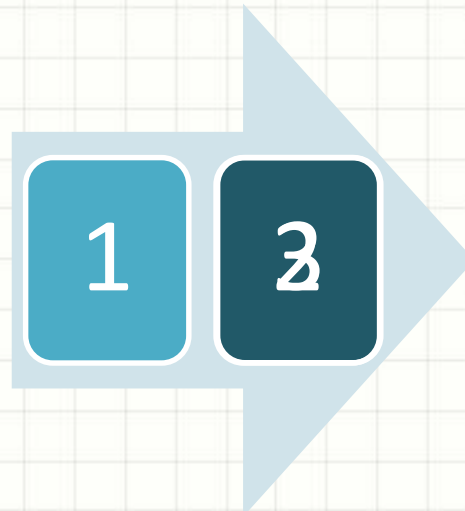


Visibilité

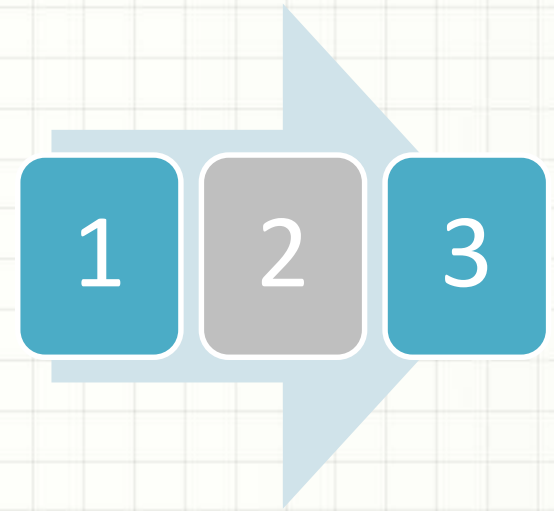
visible



include in layout



enable



Libellé

label

- titre
- support HTML allégé
- propriété recherché pour conversion

text

- multi-lignes
- support HTML allégé

Données

data

- réservé (rendus)
- libre (reste) → à éviter
- données sources
- sert au paramétrage

automation*Data

- libre
- dédié aux tests automatiques

dataProvider

- réservé (conteneur)
- libre (reste)
- lié à une source
- source directe ou non
- sert au "remplissage"



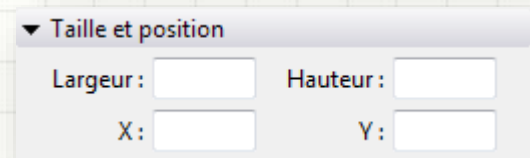
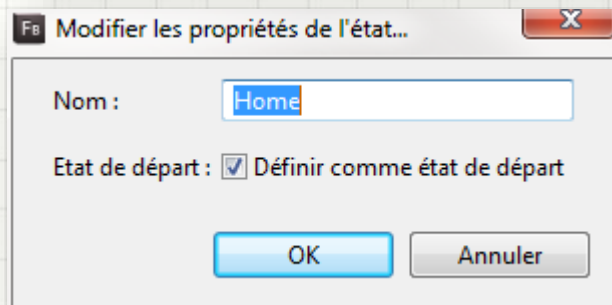
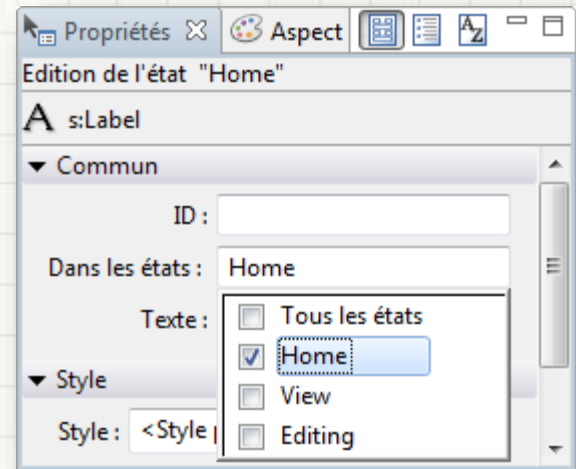
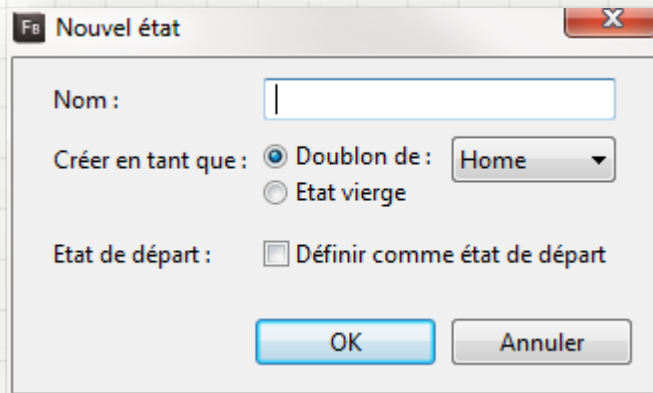
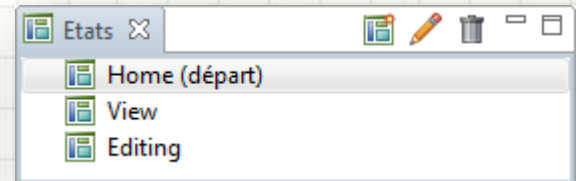
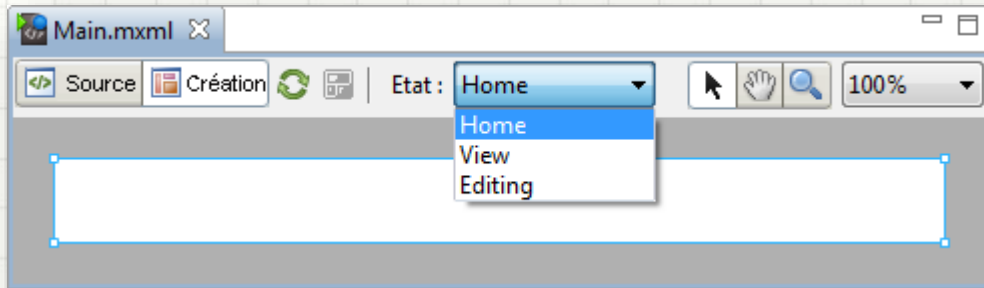
ÉTATS

INTERFACE DYNAMIQUE

Utilité

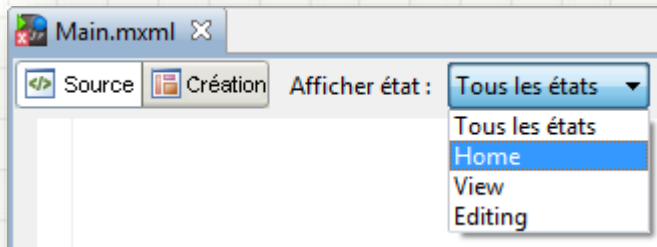
- State → état d'affichage d'un composant
- Possibilités
 - sélection des enfants
 - paramétrage (propre et enfants)
 - rendu et style
- Calcul automatique des changements

Utilisation ① Interface



Utilisation ② MXML

```
<s:Application currentState="Home">  
  <s:states>  
    <s:State id="stateHome" name="Home"/>  
    <s:State id="stateView" name="View"/>  
    <s:State id="stateEditing" name="Editing" basedOn="View"/>  
  </s:states>  
  <s:Label includeIn="View" excludeFrom="Home,Editing"  
    text="Accueil" text.View="Vue"/>  
</s:Application>
```



```
includeIn="" excludeFrom="Home"/>
```



Utilisation ③ AS

```
var stateHome:State = new State();  
stateHome.name = "Home";
```

```
var stateView:State = new State();  
stateView.name = "View";  
stateView.overrides = [ new SetProperty( label, "text", "Vue" ) ];
```

```
var stateEditing:State = new State();  
stateEditing. basedOn = "View";  
stateEditing.name = "Editing";
```

```
states = [ stateHome , stateView , stateEditing ];  
currentState = "Home";
```



CONTENEUR ET PLACEMENT

Conteneur vs. placement

Container

- composant
- possède des enfants

Layout

- pas composant
- pas d'enfants
- arrange des composants
- donne position
- Influence
 - taille
 - transformation
 - colorisation

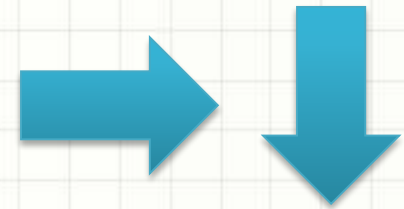
Placement absolu (Basic)

- par défaut
- aucune mise en place
- travaille sur **x** et **y** ou les contraintes
- aucune autre modification



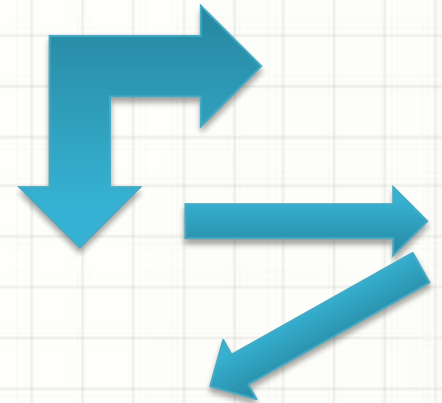
Placement orienté (H/V)

- le plus utilisé
- force placement par index
- ignore les propriétés absolues
- calcule
 - influences de tailles → **width, height**
 - %
 - overflow
 - mise en place → **x, y**
 - padding, gap
 - overflow



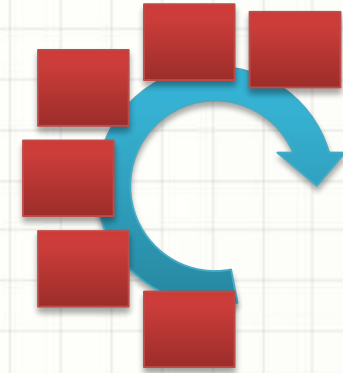
Placement bi-orienté (Tile, Flow)

- mix H/V
- exemples
 - Tile : affiche en grille
 - Flow : comprime
- taille et position au cas par cas
 - autorise les limitations (colonnes / lignes)
 - ex Tile
 - taille : max = taille cellule (grille)
 - position : relatif cellule



Placement personnalisé

- tout à penser / coder
- à calculer
 - placements
 - transformations
 - colorisation

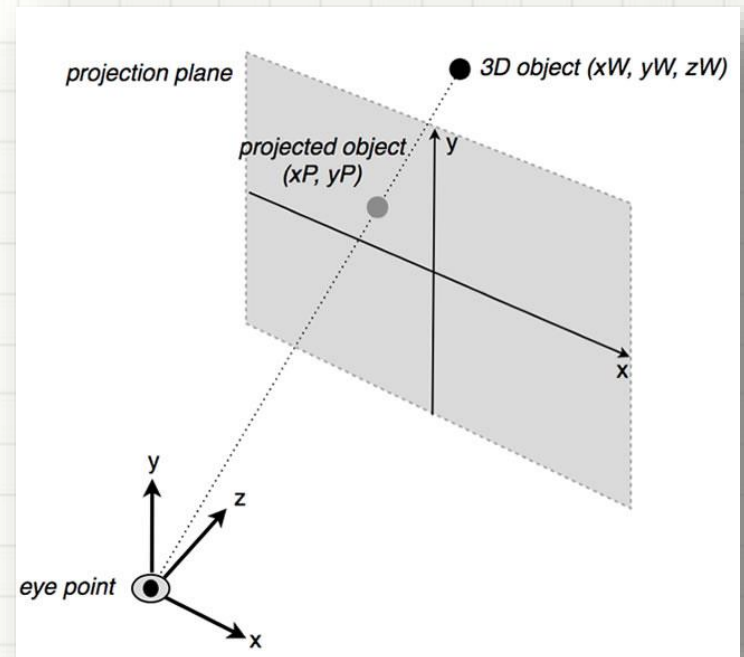
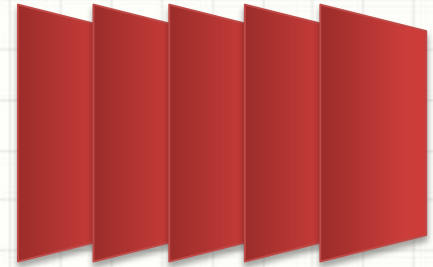


→ résultats supérieurs

Placement 3D

- tout à penser / coder en 3D
- utilisation de la 3D native
- à paramétrer
 - scène ← conteneur
 - projection
- à calculer
 - placements 3D
 - transformations 3D
 - colorisation

→ résultats supérieurs **en 3D**



Placement personnalisé (AS)

```
public class monLayout extends LayoutBase {  
    override public function updateDisplayList(width:Number, height:Number):void {  
        var element:ILayoutElement;  
        var x:Number;  
  
        super.updateDisplayList(width, height);  
  
        for (var i:int = 0 ; i < target.numElements ; i++) {  
            element = target.getElementAt(i);  
            x = (i == 0) ? 0 : target.getElementAt(i - 1).getLayoutBoundsX();  
  
            element.setLayoutBoundsPosition(x, 0);  
        }  
    }  
}
```

Matrice de transformation

2D

$$\begin{bmatrix} a & c & t_x \\ b & d & t_y \\ 0 & 0 & 1 \end{bmatrix}$$



3D

Orientation			Translation
scaleX	0	0	t_x
0	scaleY	0	t_y
0	0	scaleZ	t_z
0	0	0	t_w
x-axis	y-axis	z-axis	

$$\begin{bmatrix} \text{scaleX} & 0 & 0 & t_x \\ 0 & \text{scaleY} & 0 & t_y \\ 0 & 0 & \text{scaleZ} & t_z \\ 0 & 0 & 0 & t_w \end{bmatrix}$$



CONTENEUR DE DONNÉES ET RENDU

Généralités

- Conteneur de données = conteneur
- DataGroup, List, ...
- Différences
 - source de données externe non graphique
 - enfants : définis → générés = rendu d'éléments
- Générer ?
- Conteneur visuel ou pas ?

Rendu texte

- Convertir en texte
- Conversion
 - `String(item)`
 - `item.label`
 - `item.toString()`
 - `item[propriété]`
 - fonction de génération (`item`)
- Rendu par défaut
 - `DefaultItemRenderer`
 - `String(item)`

Rendu rapide

- MXML

```
<s:List>
  <s:itemRenderer>
    <fx:Component>
      <s:ItemRenderer>
        <s:Label text="{data.name}"/>
      </s:ItemRenderer>
    </fx:Component>
  </s:itemRenderer>
</s:List>
```

- **ItemRenderer** hérite de **Group**

Rendu personnalisé ①

- Héritage de **(I)ItemRenderer**
- Propriétés
 - **data** : item
 - **listData** : contexte de l'item dans la liste
- Interface
 - Rendu par défaut

Rendu personnalisé ② MXML

```
<?xml version="1.0" encoding="utf-8"?>
<s:ItemRenderer xmlns:fx="http://ns.adobe.com/mxml/2009"
                 xmlns:s="library://ns.adobe.com/flex/spark"
                 xmlns:mx="library://ns.adobe.com/flex/mx">
    <s:Label text="{data}"/>
</s:ItemRenderer>
```

Rendu personnalisé ③ AS

```
public class monRenderer extends ItemRenderer {  
    override protected function updateDisplayList(unscaledWidth:Number,  
                                                    unscaledHeight:Number):void {  
        var label:Label = new Label();  
        label.text = String(data);  
        addElement(label);  
  
        super.updateDisplayList(unscaledWidth, unscaledHeight);  
    }  
}
```



RAPPELS

Convention de codage

- CHOISISSEZ UNE CONVENTION DE CODAGE !
- SUIVEZ CETTE CONVENTION !
- Exemples
 - Typographie, CamelCase
 - Nommage
 - Séparation code HTML / CSS / JS, AS / MXML
 - *MVC*
 - Frameworks, API, ...
 - ...



**VOUS AVEZ DES
QUESTIONS ?**

prochaine slide :
TD / TP

TD

- Carrousel
 1. page web simple
 2. HTML : n div (un par élément)
 - élément : une image et un texte
 - flèches suivant/précédent
 3. CSS
 - image en fond, texte par-dessus
 4. JS : un div à la fois
 - propriété « display »


```

<html>
<head>
<style>
  #carrousel {
    width: 100%;
    background-color: rgba(0, 0, 0, .1);
    margin: 50px 0px auto 0px;
    padding: auto;
  }
  #carrousel .element img {
    max-width : 250px;
  }
  #carrousel .element .copyright:before {
    content: "(@ "
  }
  #carrousel .element .copyright:after {
    content: ")"
  }
  #carrousel .element .date {
    display: none;
  }
</style>
</head>
<body>
  <h1>Mon carrousel</h1>
  <div id="carrousel">
    <div class="arrow"><a id="leftarrow"
href="#">&lt;</a></div>
    <div class="element">
      
      <span class="title">Sombrero Galaxy (Messier 104) as
seen from NASA's Hubble Space Telescope</span>
      <span class="copyright">STSCI/AURA/NASA</span>
      <span class="date">October 4, 2014</span>
      <p>World Space week begins today. How to observe this
week-long educational effort, you ask? Go somewhere away from
city lights and take a look up at the night sky. With a little
research beforehand, you may be able to locate the Sombrero
Galaxy.</p>
      <p>Located in the Virgo constellation, it's one of the
easiest galaxies to view with amateur telescope equipment. And
yes, from some angles, it looks like the wide-brimmed hat from
Mexico. Happy stargazing!</p>
    </div>
    <div class="element">
      
      <span class="title">Silkorm cocoons at a weaving center
in Luang Prabang, Laos</span>
      <span class="copyright">David Noton
Photography/Alamy</span>
      <span class="date">October 2, 2014</span>
      <p>In Luang Prabang, a city in northern Laos, most silk

```

is still woven by hand, using methods refined over the centuries. These silkorm cocoons, always from the species Bombyx mori, will go from natural wonders to handcrafted textiles.</p>

<p>The commercially reared silkorm has become so specialized over years of careful breeding that Bombyx mori no longer exists in the wild. However, hundreds of other silkorm species do live in the natural world, and their cocoons are often foraged to produce a silk that's tougher, but also rougher than the silk from commercially reared silkorms.</p>

```

    </div>
    <div class="element">
      
      <span class="title">Cathedral Peak, Yosemite National
Park, California</span>
      <span class="copyright">Mark Brodtkin/Rex Features</span>
      <span class="date">October 1, 2014</span>
      <p>Seasoned climbers of North American mountains
probably know the Yosemite Decimal System, developed among
mountaineers here in Yosemite National Park. For the rest of
us: It's a rating system used to classify the difficulty of
hikes and climbs and it's used primarily in the United States
and Canada.</p>
      <p>Scaling Eichorn Pinnacle, the western spire of
Cathedral Peak here, rates a 5.4. What does that mean as far
as the severity of the ascent? Here's the definition of a
Class 5 climb: "Technical free climbing involving rope,
belaying, and other protection hardware for safety. Un-rope
falls can result in severe injury or death." Anyone up for a
nice, safe picnic?</p>
    </div>
    <div class="arrow"><a id="rightarrow"
href="#">&gt;</a></div>
  </div>
  <script>
    document.getElementById("leftarrow").onclick = function ()
    {
      alert("left click");
    }
    document.getElementById("rightarrow").onclick = function
() {
      alert("right click");
    }
  </script>
</body>

```

TP

- Application responsive
 - mode bureau :
 - carrousel automatique
 - flèches « hover » pour changer
 - mode tablette :
 - carrousel automatique
 - sans texte
 - onglets adaptés aux doigts
 - mode mobile
 - titre seulement en liste
 - click : détails avec image