

# Rapport de Site

Lecarpentier Jeremie

Pinçon Antoine

17 décembre 2014

## Table des matières

<b>1</b>	<b>Intro</b>	<b>2</b>
1.1	Titre, description du contenu . . . . .	2
1.2	Pourquoi ces choix ? . . . . .	2
1.3	Logiciels et langages utilisés . . . . .	2
<b>2</b>	<b>Structures du Site</b>	<b>2</b>
2.1	Plan sémantique du site . . . . .	2
2.2	Structure du menu de navigation . . . . .	2
2.3	Plan navigationnel du site . . . . .	2
2.4	Organisation des dossiers . . . . .	2
<b>3</b>	<b>Qu'avons nous voulu faire ?</b>	<b>3</b>
3.1	Première version en HTML/CSS . . . . .	3
3.2	Ajout de javascript . . . . .	3
<b>4</b>	<b>Solutions techniques pour réaliser le site</b>	<b>3</b>
4.1	Mise en page . . . . .	3
4.1.1	Responsive design . . . . .	3
4.1.2	Grande Version . . . . .	4
4.1.3	Version Mobile . . . . .	4
4.2	Menu de navigation . . . . .	4
4.2.1	Génération Automatique . . . . .	4
4.2.2	Placement Automatique . . . . .	7
4.2.3	Style Associé . . . . .	8
4.3	"Image Slider" . . . . .	8
4.3.1	Première Version : Animations CSS . . . . .	8
4.3.2	Deuxième Version : Objet Javascript . . . . .	11

## Liste des tableaux

1	Méthodes de manipulation du DOM . . . . .	5
---	---	---

## Table des figures

### 1 Intro

#### 1.1 Titre, description du contenu

- thème : évènements Normands
- 3 sous thèmes choisis, avec 5 événements au total
  - Festivals musicaux
    - Festival de Beauregard
    - Festival Jazz sous les pommiers
  - Festival de cinéma
    - Festival du cinéma américain de Deauville
  - Festivals traditionnels
    - Fetes médiéval de Bayeux
    - Festival Cidre et Dragons

#### 1.2 Pourquoi ces choix ?

#### 1.3 Logiciels et langages utilisés

HTML5, CSS3, JavascriptX, gimp et photoshop pour l'édition d'images, latex pour le compte rendu (éditeur : teXshop / teXmaker)

### 2 Structures du Site

#### 2.1 Plan sémantique du site

figures

#### 2.2 Structure du menu de navigation

figures

#### 2.3 Plan navigationnel du site

figure

#### 2.4 Organisation des dossiers

figure

### 3 Qu'avons nous voulu faire ?

Je propose de ne pas faire cette section mais de passer directement à la suivante, que j'ai modifié pour que ce soit possible : en gros, au lieu de dire ce qu'on a voulu faire, on dit direct ce qu'on a fait et comment, et par themes...

#### 3.1 Première version en HTML/CSS

- premiers schémas
- éléments permanents (header par exemple)
- éléments provisoires (slider avec transitions dans le CSS par exemple)

#### 3.2 Ajout de javascript

- refonte du "slider"
- ajout de styles dynamiques
- ajout ou retrait d'éléments de la page de façon dynamique ?
- ajout de trucs marrants qui ne servent a rien ?

### 4 Solutions techniques pour réaliser le site

Du coup la liste suivante ne sert pas du tout...

- menus déroulants
- slider
- affichage d'un bloc en passant sur un autre
- mise en page des blocs (schémas, images...)
- avec à chaque fois des snippets, des HTML tag content is not necessary, CSS, javascript
- 4- W3C
- erreurs rencontrées, solutions apportées.

#### 4.1 Mise en page

##### 4.1.1 Responsive design

Nous avons souhaité que notre mise en page puisse s'adapter à la taille d'écran. Celle-ci peut varier grandement selon que le site soit visité sur un ordinateur de bureau, une tablette ou un ordiphone. Nous avons donc décidé que la mise en page en elle-même devait changer selon la taille d'écran, et pas simplement les proportions des différents éléments. Pour atteindre ce but, nous avons utilisé les `@media` queries. Celles-ci permettent de spécifier pour quelles types de medias, ou tailles d'écran, ou orientations doit s'appliquer le style :

- contenu dans la feuille de style liée au HTML dans la balise `<link>`, si l'attribut `media` est donné dans cette balise ;
- contenu entre accolades si le `@media` query est dans la feuille de style.

Ces deux utilisations possibles sont illustrées dans les Listings 1 et 2.

#### Listing 1 – Illustration des @media queries dans le HTML

```
<!--feuille de style commune a tous medias-->
<link rel="stylesheet" href="../style/commun.css" media="all">

<!--feuille de style pour les ecrans de largeur inferieure a 960px-->
<link rel="stylesheet" href="../style/commun-small.css"
      media="(max-width: 960px)">

<!--feuille de style pour les ecrans de largeur superieure a 960px-->
<link rel="stylesheet" href="../style/commun-big.css"
      media="(min-width: 960px)">
```

#### Listing 2 – Illustration des @media queries dans le CSS

```
@media(orientation: portrait){
    /*toutes les proprietes enoncees ici*/
    /*seront appliquees si la largeur du media est*/
    /*inferieure a sa longueur*/
}
```

### 4.1.2 Grande Version

Du coup ca tu peux t'en occuper, c'est juste le css normal sauf le style du header, ca on en parle dans la sous-section suivante

### 4.1.3 Version Mobile

Et ca aussi, c'est juste les trucs "-small"... Et dans ces deux sous-sous-sections tu peux aussi parler du peu de javascript qu'on a utilisé pour les hauteurs..

## 4.2 Menu de navigation

### 4.2.1 Génération Automatique

Nous avons voulu faire en sorte que notre menu de navigation, placé dans une balise <nav> soit généré automatiquement par un script. Cela a pour avantages de faciliter la maintenance de celui-ci et l'homogénéité du site. Il n'est ainsi plus nécessaire de modifier toutes les pages HTML si on souhaite ajouter un élément à ce menu par exemple.

Ce menu de navigation se présente pour la version Desktop de notre site sous la forme d'un menu déroulant. Nous avons choisi de le structurer sous la forme d'une liste non ordonnée. À chaque niveau correspond une liste. Ainsi, le premier niveau est la liste principale, le second niveau est une liste dans un élément de la premiere liste. Il n'y a que deux niveaux de listes par souci de clarté dans la navigation.

Nous avons donc structuré notre Array de la façon suivante :

- chaque élément est une sous-liste correspondant à un élément de la liste principale (premier niveau);
- chaque sous-liste contient un String correspondant au texte affiché pour le premier niveau et autant de sous-sous-listes qu'il y a d'éléments dans le deuxième niveau;
- chaque sous-sous-liste contient deux Strings, correspondant à la cible et au texte du lien qui sera un `<li>` de la liste de deuxième niveau.

Le Listing 3 détaille le Array que nous avons utilisé dans notre script.

Listing 3 – Array utilisé dans notre Script

```
var navMenu = [ ["Musique", [ "beauregard.html", "Beauregard" ],
                [ "jazz.html", "Jazz sous les Pommiers" ] ],
                [ "Cinema", [ "deauville.html", "Festival de deauville" ] ],
                [ "Tradition", [ "medieval.html", "Festival medieval de Bayeux" ],
                [ "cidre.html", "Cidre et Dragons" ] ] ];
```

Nous avons pu générer la structure de notre menu en utilisant les méthodes de création d'éléments HTML et d'insertion de ceux-ci dans le DOM associées à l'objet Document. La Table 1 liste les méthodes que nous avons utilisé.

TABLE 1 – Méthodes de manipulation du DOM

Méthode	Description
<code>document.createElement("nom de balise")</code>	Crée et retourne un élément HTML du type spécifié.
<code>élément1.appendChild(élément2)</code>	Ajoute <i>élément2</i> à <i>élément1</i> en tant que dernier fils.
<code>élément.setAttribute("attribut", "valeur")</code>	Ajoute un <i>attribut</i> à un <i>élément</i> .
<code>élément1.insertBefore(élément2, élément3)</code>	Ajoute <i>élément2</i> aux fils de <i>élément1</i> avant <i>élément3</i> .

L'utilisation de ces méthodes et l'exploitation des données du menu contenues dans un Array, qui se fait au moyen de boucles, sont détaillés dans le listing 4, qui correspond à la fonction générant notre menu dans notre script. Les commentaires explicitent le code.

Listing 4 – Exploitation des données d'un Array

```
function header(){
    //cette fonction construit le menu dans son ensemble

    //la balise nav contiendra le menu dans son ensemble
    // et elle doit etre stockee dans une variable globale
    dropDown = document.createElement("nav");
```

```

//on cree la liste principale
var firstUl = document.createElement("ul");
dropDown.appendChild(firstUl);

//le style du menu utilise des elements flottants , il faut donc une div c
var clearer = document.createElement("div");
clearer.setAttribute("class", "clearer");
dropDown.appendChild(clearer);

//cette boucle parse le Array
//cree un li pour chaque element
// et un ul dans chaque li
for (var element in navMenu){
    var li = document.createElement("li");
    var ul = document.createElement("ul");
    li.appendChild(ul);
    firstUl.appendChild(li);
}

//a ce stade la structure de la liste est faite
// il suffit de la remplir

for (var i=0; i<navMenu.length; i++){

    //le premier niveau
    var firstLi = firstUl.childNodes[i];
    var anchor = document.createElement("a");
    anchor.href = "#";
    anchor.appendChild(document.createTextNode(navMenu[i][0]));
    firstLi.insertBefore(anchor, firstLi.firstChild);

    //et le second
    var secondLevel = firstLi.lastChild;
    for (var j=1; j<navMenu[i].length; j++){
        var a = document.createElement("a");
        a.href = navMenu[i][j][0];
        a.appendChild(document.createTextNode(navMenu[i][j][1]));
        child = document.createElement("li");
        child.appendChild(a);
        secondLevel.appendChild(child);
    }
}

```

```

// et pour finir on place la structure cree dans le header
document.getElementsByTagName("header")[0].appendChild(dropDown);

}

```

#### 4.2.2 Placement Automatique

Pour des raisons de cohérence structurelle, lors du passage de la version desktop à la version mobile, il est important que le menu de navigation change de place dans le DOM. En effet, dans la version desktop, ce menu fait partie du header, alors que dans la version mobile, il est sur le coté de la page. Il devrait donc logiquement être placé dans une balise `<aside>`. De plus, il doit revenir dans le `<header>` lors d'un passage de la version mobile à la version desktop. Ces changements sont aussi pris en charge par notre script dans deux fonctions, détaillées dans le Listing ??.

Listing 5 – Fonctions Assurant le Mouvement du Menu de Navigation

```

// placer le nav dans un aside
function moveNavToAside(){

    // creation d'un aside avec le id asideNav dans lequel on placera le menu
    var aside = document.createElement("aside");
    aside.setAttribute("id", "asideNav");

    // la methode removeChild retourne l'element ;
    // c'est un peu comme un couper (le coller se fait plus tard)
    var nav = document.getElementsByTagName("header")[0].removeChild(document

    // l'element id=beforeFooter est une div class=clearer, car le menu sera f
    document.body.insertBefore(aside, document.getElementById("beforeFooter"));
    document.getElementById("asideNav").appendChild(nav);

}

function moveNavToHeader(){

    // cette fonction fait en gros l'inverse de la precedente
    var nav = document.getElementById("asideNav").removeChild(document.getElement
    document.getElementsByTagName("header")[0].appendChild(nav);
    document.body.removeChild(document.getElementById("asideNav"));
    document.getElementsByTagName("header")[0].style.width = "100%";

}

```

### 4.2.3 Style Associé

c'est ici que tu peux reprendre le CR TP 11 je pense, a moins que tu aies déjà parlé du style du menu déroulant avant, mais je pense que ce serait bien de le faire ici, et en meme temps ici tu peux parler des styles pour les deux versions (grande et mobile..) Et si tu peux corriger le mot en italique dans le paragraphe après celui ci ce serait miraculeux, parce que moi je le connais pas en français..

## 4.3 "Image Slider"

Un des *features* de notre page d'accueil est d'avoir des images défilantes, sur lesquelles on peut cliquer pour accéder à une présentation des themes dont traite notre site (Musique, Cinéma et Tradition). Ce défilement d'images est un effet assez difficile à mettre en oeuvre sans utiliser de plugin javascript. Nous avons au cours du développement de notre site testé plusieurs façons de mettre cet effet en place.

### 4.3.1 Premiere Version : Animations CSS

Le CSS permet de faire des transitions et des animations portant sur une ou plusieurs propriétés particulières. L'avantage de ce procédé est qu'il ne requiert pas de javascript, l'inconvénient est

```
/* appel aux animations */

#first {
    -webkit-animation: cycle 18s linear infinite;
    -moz-animation: cycle 18s linear infinite;
    animation: cycle 18s linear infinite;
}

#second {
    -webkit-animation: cycletwo 18s linear infinite;
    -moz-animation: cycletwo 18s linear infinite;
    animation: cycletwo 18s linear infinite;
}

#third {
    -webkit-animation: cyclethree 18s linear infinite;
    -moz-animation: cyclethree 18s linear infinite;
    animation: cyclethree 18s linear infinite;
}

/* ANIMATION */
```



```
/* et oui, en trois fois pour que ca marche avec plein de navigateurs !!*/  
/* meme si le validator css n'est pas content, c'est comme ca qu'il faut le fa  
/* http://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp*/
```

```
@-webkit-keyframes cycle {  
    0% {opacity: 1; }  
    22.2% {opacity: 1;}  
    27.8% {opacity: 0;}  
    28% {opacity: 0;}  
    82% {opacity: 0;}  
    90%{opacity: 1;}  
  
}
```

```
@-webkit-keyframes cycletwo {  
    0% {opacity: 0; }  
    22.3% {opacity: 0;}  
    27.8% {opacity: 1;}  
    50% {opacity: 1;}  
    55.5% { opacity: 0;}  
  
}
```

```
@-webkit-keyframes cyclethree {  
    0% {opacity: 0; }  
    55.5% {opacity: 0;}  
    61% { opacity: 1;}  
    80% { opacity: 1;}  
    100% {opacity: 0;}  
  
}
```

```
@-moz-keyframes cycle {  
    0% {opacity: 1; }  
    22.2% {opacity: 1;}  
    27.8% {opacity: 0;}  
    28% {opacity: 0;}  
    82% {opacity: 0;}  
    100%{opacity: 1;}  
  
}
```

```
}
```

```
@-moz-keyframes cycletwo {  
    0% {opacity: 0; }  
    22.3% {opacity: 0;}  
    27.8% {opacity: 1;}  
    50% {opacity: 1;}  
    55.5% { opacity: 0;}  

```

```
}
```

```
@-moz-keyframes cyclethree {  
    0% {opacity: 0; }  
    55.5% {opacity: 0;}  
    61% { opacity: 1;}  
    80% { opacity: 1;}  
    100% {opacity: 0;}  

```

```
}
```

```
@keyframes cycle {  
0% {opacity: 1; }  
    22.2% {opacity: 1;}  
    27.8% {opacity: 0;}  
    28% {opacity: 0;}  
    82% {opacity: 0;}  
    100%{opacity: 1;}  

```

```
}
```

```
@keyframes cycletwo {  
    0% {opacity: 0; }  
    22.3% {opacity: 0;}  
    27.8% {opacity: 1;}  
    50% {opacity: 1;}  
    55.5% { opacity: 0;}  

```

```
}
```

```
@keyframes cyclethree {  
  0% { opacity: 0; }  
  55.5% { opacity: 0;}  
  61% { opacity: 1;}  
  83% { opacity: 1;}  
  90% { opacity: 0;}  
  
}
```

#### **4.3.2 Deuxième Version : Objet Javascript**