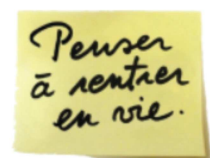


Développement :

Nolark est un site de vente de casques de moto, lié à une base de données.



Au-delà de l'obligation légale, le port d'un casque est le garant de votre sécurité. Le choix de votre casque doit se faire en fonction de vos besoins (route, cross, piste...).

Spécialiste reconnu dans l'univers de la sécurité du deux-roues, Nolark vous propose des milliers de modèles de casques disponibles au travers de plus de 50 marques. Vous trouverez ici tous les types de casques moto : jet, intégral, modulable, transformable, piste, cross ou encore un large choix de casques moto enfants (taille et poids spécialement adaptés).

Nolark c'est également la disponibilité de toutes les couleurs et matières et ce, pour toutes les bourses.

Une question ? Un conseil ? Nos conseillers sont là pour vous aiguiller afin de trouver le modèle qui correspond à vos besoins.

Nolark n'est pas seulement un magasin de casques, c'est avant tout une équipe de passionnés qui sélectionnent pour vous les meilleurs casques moto afin de répondre au mieux à vos attentes.



Catégories les plus populaires

Cadre :

Il s'agit d'un TP proposé dans le cadre du cours "Développement d'applications", par José Gil.

Support :

Développé avec NetBeans, les langages utilisés sont : html/css ainsi que JavaScript et Php. La base de données liée est gérée via PhpMyAdmin sur WampServer.

Contraintes / Organisation :

L'ordre de développement nous était fourni pas à pas, ainsi que la base de données.

Difficultés rencontrées :

Le CSS est un langage qui peut être un peu difficile à manipuler si le code html derrière n'est pas parfaitement organisé. Mais dans le cadre de ce TP, je n'ai pas rencontré de difficultés particulières.

Description détaillée du développement

J'ai donc suivi les étapes suivantes :

- la mise en forme des pages web avec le CSS
- gestion des événements avec Javascript
- liaisons avec la base de données

1. Mise en forme : le couple gagnant HTML / CSS :

L'en-tête, commune à toutes les pages : elle est bien liée à mes feuilles de style CSS, et il y a une balise <viewport> pour la mise en page spécifique pour l'affichage du site sur smartphones.

```
<html lang="fr-FR">
  <head>
    <title>"Casques NOLARK : Sécurité et confort, nos priorités !"</title>
    <link href="../CSS/styles.css" rel="stylesheet" type="text/css"/>
    <link href="../CSS/casques.css" rel="stylesheet" type="text/css"/>
    <meta charset="UTF-8">
    <meta name="author" content="Stéphanie OTTO">
    <meta name="description" content="Découvrez des casques moto dépassant même les exigences des tests de sécurité. Tous les casques Nolark au meilleur prix et avec en prime la livraison gratuite !">
    <link rel="icon" href="../favicon.ico">
    <!-- Pour que le site s'adapte à l'écran définit en CSS : -->
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <header>
```

Les informations statiques, comme le texte sur la page d'accueil, ont été saisies "en dur", c'est à dire directement dans le code de la page html

```
<section id="principal">
  <article>
    
    <p>
      Au-delà de l'obligation légale, le port d'un casque est le garant de votre sécurité. Le choix de votre casque doit se faire en fonction de vos besoins (route, cross, piste...).
    </p>
    <p>
      Spécialiste reconnu dans l'univers de la sécurité du deux-roues, Nolark vous propose des milliers de modèles de casques disponibles au travers de plus de 50 marques. Vous trouverez ici tous les types de casques moto : jet, intégral, modulable, transformable, piste, cross ou encore un large choix de casques moto enfants (taille et poids spécialement adaptés).
    </p>
    <p>
      Nolark c'est également la disponibilité de toutes les couleurs et matières et ce, pour toutes les bourses.
    </p>
    <p>
      Une question ? Un conseil ? Nos conseillers sont là pour vous aiguiller afin de trouver le modèle qui correspond à vos besoins.
    </p>
    <p>
      Nolark n'est pas seulement un magasin de casques, c'est avant tout une équipe de passionnés qui sélectionnent pour vous les meilleurs casques moto afin de répondre au mieux à vos attentes.
    </p>
  </article>
  <aside>
    
```

Pour les pages de casques, il faudra un affichage dynamique, c'est à dire qu'ils s'afficheront à partir de la bdd.

NO LARK : PROTECT YOUR MINDS !

Accueil Route Cross Piste Enfants Nous contacter

211.09€ STOCK	179.19€ STOCK	285.53€ STOCK	191.25€ STOCK	381.55€ STOCK
SCORPION Exo 1000 Air Noir	SCORPION Exo 1000 Air Twister Noir Blanc	SCORPION Exo 1000 Air Twister Noir Mat	SCORPION Exo 1000 Anthracite Mat	SCORPION Exo 1000 Blanc
★★★★☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆

Mais c'est pour plus tard, pour l'instant nous avons besoin qu'il y ait une saisie " en dur ", pour pouvoir régler correctement la mise en page

```

<section id="casques">
  <article>
    
      <p class='stockok'>
        <abbr data-tip='18 casques en stock'>stock</abbr>
      </p>
      <p class="prix">342.45€</p>
      <p class="marque">AGV</p>
      <p class="modele">skyline block blanc bleu</p>
      
    </article>
  <article>
    
      <p class="stockok">
        <abbr data-tip="Plus que 9 casques en stock...">
          stock</abbr>
      </p>
  
```

Ceci pour toutes les pages du site.
Avec ce joli code, toutes ces balises parfaitement identifiées, il est temps de passer au CSS correspondant.

Le style appliqué est différent selon les pages : une page styles.css pour la mise en page de l'en-tête et de la page d'index,

```
body {
  display : flex; /* Ce conteneur est "flexible"*/
  justify-content : center; /*Les éléments portés sont centrés à l'horizontale*/
  flex-direction : column; /* Les éléments sont empilés*/
  align-items : center; /* Les éléments portés sont centrés à la verticale*/
  width : 100%; /* La page occupe toute la largeur*/
}

header {
  width : 916px; /* Taille de la bannière */
  display : flex; /* Ce conteneur est flexible*/
  flex-direction : column; /* Les éléments sont empilés */
}

picture {
  display: flex;
}

nav ul {
  display : flex; /* Ce conteneur est flexible*/
  flex-direction : row; /* Les éléments sont mis côte à côte*/
}
```

et une feuille casques.css pour la mise en page des casques (5 par ligne, le centrage, etc...)

```
#casques {
  display : flex; /* Ce conteneur est flexible*/
  flex-wrap : wrap; /*Les élts flexibles sont disposés sur plusieurs lignes */
  justify-content : flex-start; /* les élts sont alignés à gauche */
  margin-bottom : 45px; /* marge inférieure */
}

#casques article {
  width : 163px; /* largeur de l'article*/
  height: 275px; /* hauteur de l'article*/
  margin: 10px; /* marges extérieures*/
  padding: 5px; /* marges intérieures */
  border: white thin solid; /* bordure blanche */
  position : relative; /* position relative */
}

#casques article img {
  margin-left: 6px; /* marge gauche*/
}

#casques article:hover {
  border: #bbb thin solid; /* bordure au survol*/
}

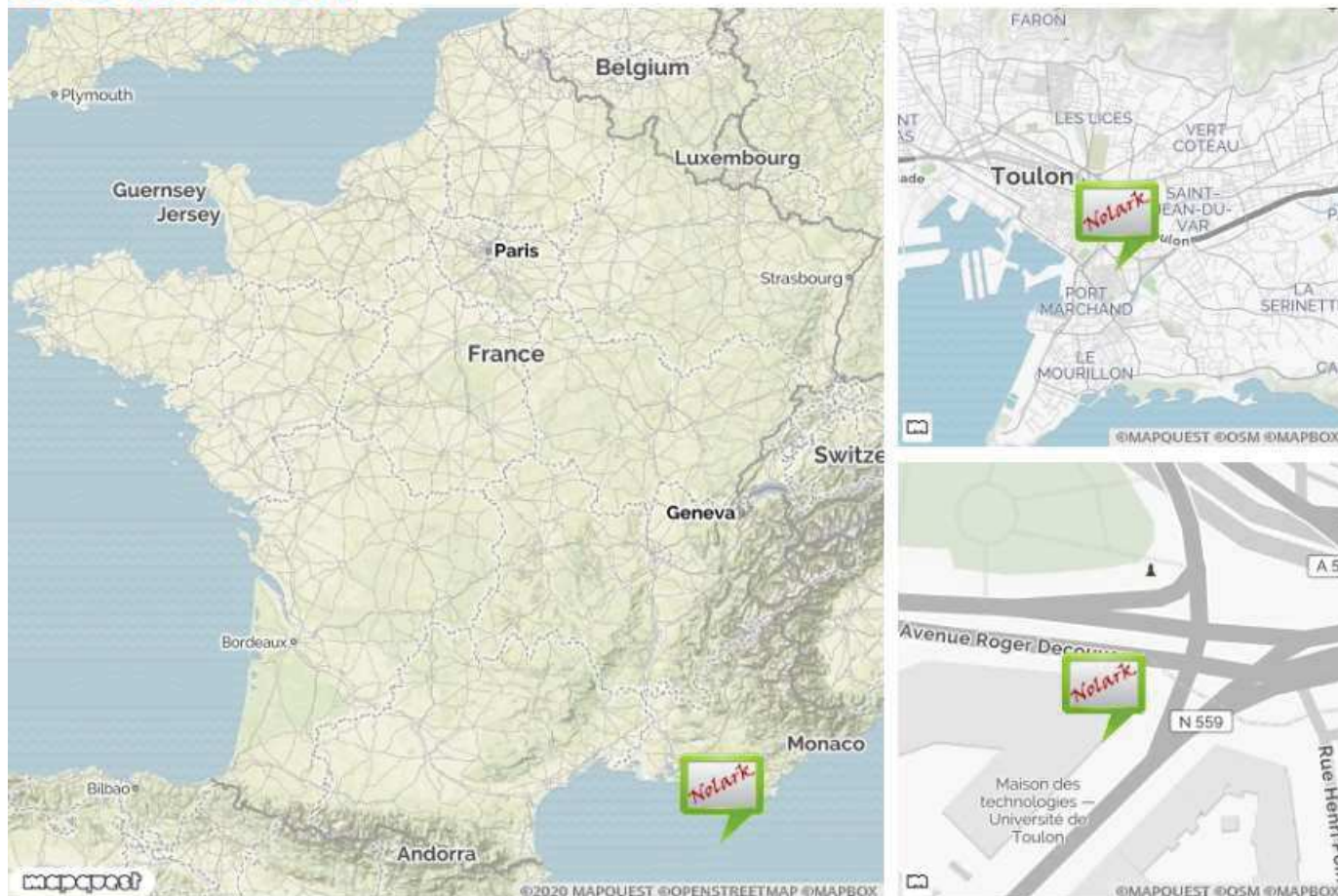
#casques .prix {
  color: red; /* couleur du texte */
  font-weight: bold; /* texte en gras*/
}
```

Et c'est là qu'on prend bien conscience de l'importance de l'organisation HTML. Pour pouvoir être aussi précis dans la mise en page en CSS, il faut, au préalable, avoir été précis dans nos balises et tous nos identifiants HTML.

Une autre page particulière, “ Nous contacter ”



Nous localiser :



Nous contacter :

Pour cette page, une API (Application Programming Interface) qui nous est fournie par Mapquest permet l’affichage des cartes : <https://developer.mapquest.com/documentation/open/static-map-api/v5/>

Sans entrer dans les détails de l’utilisation, on obtient un résultat similaire à celui de Google Map (qui demande vos coordonnées bancaires), gratuitement et relativement facilement.

2. La gestion des évènements avec JavaScript

Observer de jolies pages, c'est bien. Pouvoir interagir avec elles, leur demander par exemple de faire des calculs, c'est mieux.

Voyons donc maintenant un simulateur de commissions, destiné aux commerciaux de chez Nolark, afin de les motiver :



Simulateur de calcul pour la force de vente :

Vos résultats :

Nombre d'années d'ancienneté	<input type="text" value="10"/>
Nombre de S 20 vendus :	<input type="text" value="13"/>
Nombre de X-Spirit vendus :	<input type="text" value="0"/>
Nombre de Multitec vendus :	<input type="text" value="12"/>
Distance parcourue :	<input type="text" value="69"/>

La rémunération sera de : 1299.15 €

Pour faire fonctionner ce simulateur, il a d'abord fallu créer toutes les fonctions correspondantes. Ex avec la prime d'ancienneté :

```
* Fonction qui retourne la prime d'ancienneté
* @param {integer} nb
* @param {float} fixe
* @returns {float}
- */
] function calculPrimeAnciennete(nb, fixe) {
    const nbAncienMin = 5, txAncienMin = 0.03, nbAncienSup = 10,
        txAncienSup = 0.06;
] if (nb >= nbAncienSup) {
    return (fixe * txAncienSup);
- }
] else if (nb >= nbAncienMin) {
    return (fixe * txAncienMin);
- }
] else {
    return 0.0;
- }
- }
- }
```

Ensuite, une fonction récapitulative, qui récupère les sommes saisies dans la page HTML pour les placer dans des variables, et qui utilise les fonctions préalablement écrites :

```
function calcRemu() {
  // Déclaration des constantes :
  const fixe = 1100.0;
  // Déclaration et affectation des variables :
  let nbAncien = recupValeur("#num_ancien");
  let nbS20 = recupValeur("#num_s20");
  let nbXS = recupValeur("#num_xs");
  let nbMulti = recupValeur("#num_multi");
  let km = recupValeur("#num_km");
  let remuneration = fixe + calculPrimeAnciennete(nbAncien, fixe) +
    recupComS20(nbS20) + recupComXS(nbXS) + recupComMulti(nbMulti)
    + recupIndemKm(km);

  // Affichage du résultat
  afficheRemu(remuneration);
}
```

Et enfin, le plus intéressant, on place des “ écouteurs d’évènements ” :

```
window.addEventListener("load", function() {
  // Déclaration de l'index de parcours
  let i;

  // tabInputs est une collection de <input> :
  let tabInputs = window.document.querySelectorAll("input");

  // Parcours de tabInputs en s'appuyant sur le nbre de <input>
  for (i = 0; i < tabInputs.length ; i++) {
    // Ajout d'un listener sur ts les <input> sur l'évènement onKeyUp
    tabInputs[i].addEventListener("keyup", calcRemu);
    tabInputs[i].addEventListener("mouseup", calcRemu);
  }
});
/**
```

Donc sur chaque “ input ” de cette page, sur les évènements “ keyup ” qui se déclenchent lorsque qu'une touche du clavier qui a été pressée est relâchée et sur les évènements “ mouseup ” (idem que keyup mais avec la souris), c’est ma fonction calcRemu() qui s’exécute.

JavaScript est un langage puissant, qui permet de modifier l’affichage tel qu’il a été défini en html. Dans ce cas, une balise h2 rémunération est placée dans le code html de la page :

```
<h2 id="remuneration"> </h2>
```

Et dans la page JavaScript :

```
function afficheRemu(nombre) {
  window.document.querySelector("#remuneration").innerHTML =
    "La rémunération sera de : " + nombre.toFixed(2) + " €";
}
/**
```

3. Liaisons avec la base de données

L'objectif, ici, est de rendre les pages relatives aux casques dynamiques : afin de pouvoir modifier très rapidement les casques, en ajouter ou en supprimer, sans devoir aller "mettre les mains dans le camboui" c'est à dire sans modifier le code HTML.

La base de données nous a été fournie, il n'y avait plus qu'à l'importer. On a 3 tables, casque, marque et type. La table casque par exemple ressemble à ça :

Options				id	marque	modele	type	prix	classement	image	stock			
<input type="checkbox"/>		Éditer		Copier		Supprimer	1	2	vx3 frost silver	1	249.98	20	arai-vx3-frost-silver.jpg	-1
<input type="checkbox"/>		Éditer		Copier		Supprimer	2	4	MX ELAN MC5	1	247.1	35	CL-MX-ELAN-MC5.jpg	17
<input type="checkbox"/>		Éditer		Copier		Supprimer	3	11	ds anthracite mat	1	226.59	5	Hornet-ds-anthracite-mat.jpg	13
<input type="checkbox"/>		Éditer		Copier		Supprimer	4	11	ds blanc	1	205.67	20	Hornet-ds-blanc.jpg	14
<input type="checkbox"/>		Éditer		Copier		Supprimer	5	11	ds gris	1	433.28	20	Hornet-ds-gris.jpg	18
<input type="checkbox"/>		Éditer		Copier		Supprimer	6	11	ds noir mat	1	394.92	30	Hornet-ds-noir-mat.jpg	10
<input type="checkbox"/>		Éditer		Copier		Supprimer	7	11	ds noir	1	168.02	25	Hornet-ds-noir.jpg	18
<input type="checkbox"/>		Éditer		Copier		Supprimer	8	18	vfx w noir mat	1	303.95	10	shoei-vfx-w-noir-mat.jpg	8
<input type="checkbox"/>		Éditer		Copier		Supprimer	9	22	START F.Black	1	197.59	0	X501-START-F.Black.jpg	19
<input type="checkbox"/>		Éditer		Copier		Supprimer	10	3	blanc	2	406.14	25	bobby-blanc.jpg	4
<input type="checkbox"/>		Éditer		Copier		Supprimer	11	3	noir rose	2	290.43	45	bobby-noir-rose.jpg	14
<input type="checkbox"/>		Éditer		Copier		Supprimer	12	3	noir	2	370.88	25	bobby-noir.jpg	1
<input type="checkbox"/>		Éditer		Copier		Supprimer	13	4	xy wanted mc1	2	398.5	0	cl-xy-wanted-mc1.jpg	-1
<input type="checkbox"/>		Éditer		Copier		Supprimer	14	4	xy wanted mc2	2	447.12	40	cl-xy-wanted-mc2.jpg	12
<input type="checkbox"/>		Éditer		Copier		Supprimer	15	4	xv wanted mc5	2	202.86	10	cl-xv-wanted-mc5.jpg	12

Un curseur pour accéder à la bdd et y récupérer les bonnes informations pour les placer au bon endroit a donc été créé :

```
<?php

$cnx = new PDO('mysql:host=127.0.0.1;dbname=nolark', 'nolarkuser', 'nolarkpwd');
$req = 'SELECT casque.id, nom, modele, libelle, prix, classement, image, stock';
$req .= ' FROM casque INNER JOIN type ON casque.type=type.id';
$req .= ' INNER JOIN marque ON casque.marque=marque.id';
$req .= ' WHERE libelle="' . substr($pageActuelle, 0, -4) . "'";

$res = $cnx->query($req);

echo '<section id="casques">';
while ($ligne = $res->fetch(PDO::FETCH_OBJ)) {
    echo "\t", '<article>', "\n";
    echo "\t\t", 'modele, '" >', "\n";

    // Si le stock contient -1, c'est sur commande et stockko
    if ($ligne->stock === -1) {
        echo "\t\t", '<p class="stockko"><abbr data-tip="Sur commande uniquement">stock</abbr></p>';
    } elseif ($ligne->stock >= 10) {
        echo "\t\t", '<p class="stockok"><abbr data-tip="', $ligne->stock, ' casques en stock">stock</abbr></p>';
    } else {
        echo "\t\t", '<p class="stockok"><abbr data-tip="Plus que ', $ligne->stock, ' casques en stock...">stock</abbr></p>';
    }
    echo "\t\t", '<p class="prix">', $ligne->prix, '€</p>', "\n";
    echo "\t\t", '<p class="marque">', $ligne->nom, '</p>', "\n";
    echo "\t\t", '<p class="modele">', $ligne->modele, '</p>', "\n";

    // Si la note est de 5, il faut ajouter un 0 pour correspondre au selecteur CSS
    if ($ligne->classement === 5) {
        echo "\t\t", '<img class="classement classement0', $ligne->classement, '" src="../images/casques/etoiles.gif" alt="Cl
    } else {
        echo "\t\t", '<img class="classement classement', $ligne->classement, '" src="../images/casques/etoiles.gif" alt="Cla
    }
    echo "\t", '</article>', "\n";
}
echo '</section>';
```


Ainsi, dans la page HTML, qui est maintenant une page .php, il n'y a plus qu'à remplacer le long code qui détail les casques 1 par 1 (vous pouvez imaginer, là il n'y a que 2 casques):

```
<section id="casques">
  <article>
    
      <p class="stockok">
        <abbr data-tip='18 casques en stock'>stock</abbr>
      </p>
      <p class="prix">342.45€</p>
      <p class="marque">AGV</p>
      <p class="modele">skyline block blanc bleu</p>
      
    </article>
  <article>
    
      <p class="stockok">
        <abbr data-tip="Plus que 9 casques en stock...">
          stock</abbr>
      </p>
```

par ceci :

```
<?php
    include('../includes/header.html.inc.php');

    include('../Includes/casques.inc.php');

    include('../includes/footer.inc.php');
?>

</body>
</html>
```