

# Rapport du projet Javascript

Pour la conception de notre formulaire Bootstrap de calcul de bulletin de salaire nous avons édité trois fichiers : un fichier Javascript, un fichier CSS et un fichier HTML tout en utilisant du Bootstrap.

Pour la partie design(HTML) du formulaire nous avons utilisés la classe `container` de Bootstrap qui permet de centrer le contenu et un `background-color` gris ensuite une balise `h1` pour le titre.

Nous avons séparés les informations personnelles des informations financières avec une date tout au début en `float-right`, le tout regroupé dans une balise `form` qui représente le formulaire. Pour les titres de chaque partie nous avons utilisés la balise `nav`. Pour les informations personnelles nous avons utilisés une balise `div` pour y regrouper le contenu avec comme id `contenu` appliqué aux styles CSS `background-color` de couleur white et un padding de 5px. Pour les champs Prénom et Nom nous avons utilisés les classes `form-row` et `form-group col-md-6` pour les mettre sur une même ligne avec deux blocs de 6 colonnes ainsi que des `label` et `input` de même que pour les autres champs de saisies. Pour le sexe nous avons utilisés des `button radios` et pour le supplément nous avons utilisés la balise `select` avec des `option` selon les choix aucun ,bonus ou allocation.

Après la partie informations personnelles nous avons sauté d'une ligne avec la balise `br` pour passer au champ Salaire brut en utilisant un `label` et `input` de type `number` (pour le contrôle de saisie) ainsi qu'une balise `button` avec l'id `button1` appliqué aux styles CSS `float-right` et une option `onclick` qui appelle la fonction `afficherInfos()`.

Pour la partie des informations financières nous avons utilisés deux balises `div` avec comme id `bloc1` et `bloc2` qui occupent chacun 48% de la largeur, une hauteur 365 px, un padding de 5px ainsi qu'un `background-color` white. Le `bloc1` contient les informations liées aux impôts, assurances, régimes, réduction et impôt final et est appliqué au style CSS `float-left`. Le `bloc2` contient les informations liées aux pourcentages avec des valeurs ainsi que le style CSS `float-right`. Les `input` contenus dans les blocs ont comme option `readonly` (ces champs permettent juste d'afficher les valeurs après calcul). Pour le calcul du salaire net nous avons une balise `label` et un `input` de type `number` avec une option `readonly`.

Après nous avons une balise `div` qui regroupe trois `button` : calculer salaire qui appelle la fonction `calculerSalaire ()` au clic ; réinitialiser avec comme type `reset` qui permet d'effacer toutes les informations saisies du formulaire ; imprimer qui permet l'impression avec la fonction `print ()` tout en cachant les `button`.

Pour le pied de page nous avons utilisé une balise `footer` qui contient une balise `p` ainsi qu'une balise `ul` qui a comme class `list-inline` qui permet de regrouper les balises `li` sur une même ligne qui ont comme class `list-inline-item` qui contiennent les prénoms des membres du projet.

Par la suite nous allons parler du fichier JavaScript qui contient les fonctions :

La fonction `supplément ()`

Cette fonction retourne une valeur de pourcentage suivant le choix récupéré au niveau du champ `supplément`.

Elle retourne **0.20**, **0.25** et **0** respectivement selon les choix `"bonus"`, `"allocation"` et `"aucun"`.

La fonction `montantRelative ()`

Cette fonction prend en paramètre deux nœuds (par exemple l'impôt et son pourcentage). Elle ajoute le supplément au salaire brut (appelle de la fonction supplément) puis récupère le pourcentage du nœud pourcentage en argument. Elle calcule ensuite la valeur de l'autre nœud et le met dans l'élément correspondant.

La fonction **calculerReduction ()**

On commence par déclarer une variable **reduction** initialisée à 0. Si la personne est une femme on incrémente de 2 la variable **reduction**. On récupère le nombre de personnes en charge : si ce nombre est égal à 3 on incrémente la variable de 1 ou si le nombre est égal à 4, on incrémente de 2. Et enfin, on stocke la valeur de **reduction** dans le champ **reduction**.

La fonction **afficherInfos ()**

Cette fonction nous permet de calculer les 5 champs suivants : **impot, assurance, regime, reduction** et **impot final**.

Pour l'impôt on appelle la fonction **montantRelative ()** et on lui passe en arguments le nœud de l'impôt et celui de son pourcentage. On fait idem pour l'assurance et le régime. On appelle la fonction **calculerReduction ()** pour remplir le champ **reduction**. Enfin avec cette **reduction** et l'**impôt** calculé ci-dessus on calcule l'impôt final.

Cette fonction s'exécute suite à l'évènement **"On click"** du bouton **valider**.

La fonction **calculerSalaire ()**

On récupère le salaire de base avec le supplément. On récupère ensuite le **régime**, l'**assurance** et l'**impôt final** qu'on somme et on retranche cette valeur dans la première valeur (salaire de base et supplément). Pour terminer on met la valeur finale dans le champ du salaire.

Cette fonction s'exécute suite à l'évènement **"On click"** du bouton **calculerSalaire**.

Pour le bouton imprimer, on fait appel à la méthode **print** de window à l'aide de l'évènement **"Click"**.

La classe **"ne-pas-imprimer"** est ajouté à, tous les éléments qui ne doivent pas apparaître au moment de l'impression.

Pour l'initialisation, on choisit **reset** comme type du bouton **réinitialiser**.

**Fatou Binetou Gueye : [fatoubintou967@gmail.com](mailto:fatoubintou967@gmail.com)**

**Marieme Diop : [diopmareme529@gmail.com](mailto:diopmareme529@gmail.com)**

**Ansoumana Ndiaye : [ndiayetacky@gmail.com](mailto:ndiayetacky@gmail.com)**

**Demba Soumaré : [bigsoomis@gmail.com](mailto:bigsoomis@gmail.com)**