

## Feuille N° 6 – Java Script

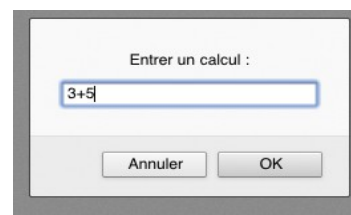
## Exercice 1

En utilisant la classe `Date` écrire un script qui affiche en format texte la date et l'heure courante (voir exemple ci dessous).

En utilisant les fonctions : `prompt` et `eval`, ajouter un évaluateur d'expression arithmétique. Une boîte de dialogue s'ouvre, l'utilisateur saisit sa formule puis valide et la résultat s'affiche. Ajouter un lien « Nouveau calcul » pour pouvoir relancer la page sur elle même.

Nous sommes le 20/10/2014.  
Il est 17h 49mn 25s.  
3+5 = 8

[Nouveau calcul...](#)



## Exercice 2

Écrire un script permettant de convertir en Francs une valeurs donnée (via une boîte de dialogue) en Euros. Puis ajouter un script permettant de calculer le périmètre et l'aire d'un carré à partir de la longueur de celui. Cette valeur sera également saisie dans une boîte de dialogue. On n'utilisera pas de formulaire.

Modifier votre affichage pour avoir un affichage dans un tableau en utilisant le style « `tab` » défini dans le fichier CSS fourni.

| Longueur | Périmètre | Aire |
|----------|-----------|------|
| 5        | 20        | 25   |

## Exercice 3

En utilisant des zones de saisie de formulaire, construire les scripts suivants :

Entrer un mot et le mettre en majuscule en appuyant sur un bouton. On utilisera la fonction `toUpperCase()` de l'objet `String`.

Entrer un prix hors taxe puis calculer et afficher le prix TTC pour un taux de TVA de 19,6% lorsqu'on appuie sur un bouton.

Entrer 3 notes d'examen et afficher la moyenne obtenue en appuyant sur un bouton.

Utiliser le fichier CSS fourni pour aboutir à l'affichage ci-dessous.

Mise en majuscules

Texte minuscule

Texte en majuscule

=>

Effacer

prix TTC

Prix Hors Taxe :

Prix TTC :

Calculer

Effacer

Calcul moyenne

Note du DS1 :

Note du DS2 :

Note du DS3 :

Moyenne obtenue :

Calculer

Effacer

#### Exercice 4

Écrire une page qui affiche deux nombres aléatoires entre 1 et 9, demande à l'utilisateur de saisir dans une zone de saisie la valeur du produit puis de valider en cliquant sur le bouton « Vérifier ».

Si l'utilisateur saisit autre chose qu'un nombre alors on affiche dans une boîte de dialogue le message « format incorrect ». S'il se trompe dans le résultat, on met en rouge la couleur de la zone de saisie et on affiche une boîte de dialogue avec un message d'erreur. Quand il clique sur le bouton « ok » de la boîte de dialogue, on efface sa saisie, on vide la zone de texte et on lui donne le focus. Tant que l'utilisateur ne trouve pas le résultat, on reste sur le même produit. Dès que l'utilisateur trouve le bon produit, on affiche une boîte de dialogue avec un message de confirmation et on recharge la page pour un nouveau produit une fois la boîte de dialogue validée.

5 x 8 =  Vérifier

On utilisera :

- `Math.floor(Math.random()*9)+1` pour la génération de nombres aléatoires entre 1 et 9.
- `isNaN(x)` pour savoir si x n'est pas numérique.
- `document.location.href= "url"` pour envoyer vers l'url.
- `(le chemin de votre champ).style.backgroundColor= "red"` pour colorier le fond du champ.
- `(le chemin de votre champ).focus()` pour donner le focus au champ.

## Exercice 5

Écrire une page où l'ordinateur choisit un nombre entre 0 et 100; l'utilisateur essaie de deviner ce nombre en saisissant une valeur dans une zone de saisie. Si l'utilisateur saisit autre chose qu'un nombre alors on affiche « format incorrect » dans une boîte de dialogue. Si le nombre est trop grand par rapport à la proposition de l'ordinateur alors on affiche une phrase comme « essai 2 : Trop grand » dans la zone de texte. De même dans le cas où le chiffre est plus petit que celui de l'ordinateur. L'utilisateur a droit à 10 essais. Au delà, on affiche le message « perdu » dans la zone de texte. Si l'utilisateur trouve la bonne réponse, on lui affiche le nombre d'essais et le mot « gagné » dans la zone de texte. Lorsque l'utilisateur a gagné ou perdu, on désactive le bouton « Contrôle » via la propriété `chemin.disabled=true`. La page contient également un lien permettant de rejouer.

|                  |    |          |                       |
|------------------|----|----------|-----------------------|
| Nombre proposé : | 19 | Contrôle | Essai 2 : Trop grand. |
|------------------|----|----------|-----------------------|

## Autre partie

## Exercice 6

Créer un document HTML qui demande un nombre dans une boîte de dialogue puis affiche la table de multiplication correspondante. En utilisant le CSS fourni, améliorer votre affichage sous forme d'un tableau stylisé comme ci dessous.

| Table de multiplication |        |          |
|-------------------------|--------|----------|
| Votre nombre            | Indice | Resultat |
| 4                       | 1      | 4        |
| 4                       | 2      | 8        |
| 4                       | 3      | 12       |
| 4                       | 4      | 16       |
| 4                       | 5      | 20       |
| 4                       | 6      | 24       |
| 4                       | 7      | 28       |
| 4                       | 8      | 32       |
| 4                       | 9      | 36       |
| 4                       | 10     | 40       |

## Exercice 7

Deux joueurs simulés par l'ordinateur lancent un dé l'un après l'autre. Les points obtenus s'ajoutent, le premier qui atteint ou dépasse 21 a gagné. Créer le document HTML permettant de lancer une partie. En utilisant le CSS fourni, afficher les points obtenus par chacun des joueurs ainsi que le total après chaque coup dans un tableau comme ci-dessous. A la fin de la partie, on affiche une boîte de dialogue avec le numéro du joueur gagnant. La page contient également un lien permettant de rejouer.

| Jeux du 21 |          |       |
|------------|----------|-------|
| Joueur 1   | Joueur 2 | Total |
| 6          |          | 6     |
|            | 1        | 7     |
| 2          |          | 9     |
|            | 3        | 12    |
| 5          |          | 17    |
|            | 4        | 21    |

## Exercice 8

Écrire une fonction `remplir(t)` qui remplit un tableau de 10 cases avec des nombres aléatoires compris entre 1 et 100. Cette fonction ne renvoie rien. Écrire une fonction `afficher(t)` qui prend en paramètre un tableau et l'affiche en utilisant le CSS fourni. Écrire une page HTML qui crée un tableau, l'affiche, le trie par ordre croissant et l'affiche de nouveau.

On souhaite maintenant trier le tableau par ordre décroissant puis l'afficher.

Pour cela, écrire une fonction `compare(a,b)` qui sera passée en paramètre de la méthode `sort`.

## Exercice 9

Écrire une page qui affiche un tirage du loto : 6 numéros **distincts** entre 1 et 49.

### Loto

39 49 35 22 48 37

[Autre tirage...](#)

## Exercice 10

Écrire une page qui affiche un tableau à deux dimensions de taille 10x10 rempli de nombres aléatoires entre 0 et 9 puis un deuxième tableau d'une dimension de taille 10 contenant le nombre d'occurrences de chaque nombre dans le tableau à deux dimensions. Utiliser le CSS fourni pour obtenir la mise en forme ci-dessous.

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 4 | 7 | 3 | 4 | 0 | 5 | 0 | 3 | 0 | 3 |
| 5 | 6 | 6 | 5 | 5 | 1 | 8 | 9 | 7 | 6 |
| 2 | 5 | 1 | 9 | 2 | 6 | 0 | 8 | 6 | 5 |
| 5 | 6 | 5 | 7 | 7 | 6 | 4 | 7 | 8 | 3 |
| 8 | 9 | 4 | 3 | 8 | 1 | 1 | 0 | 6 | 7 |
| 6 | 4 | 6 | 0 | 3 | 1 | 9 | 1 | 7 | 3 |
| 0 | 9 | 4 | 2 | 4 | 8 | 7 | 5 | 8 | 9 |
| 7 | 0 | 7 | 6 | 3 | 2 | 6 | 9 | 9 | 5 |
| 0 | 3 | 5 | 3 | 4 | 8 | 4 | 9 | 8 | 5 |
| 0 | 2 | 3 | 4 | 6 | 7 | 5 | 4 | 6 | 8 |

| Nombre         | 0  | 1 | 2 | 3  | 4  | 5  | 6  | 7  | 8  | 9 |
|----------------|----|---|---|----|----|----|----|----|----|---|
| Nb occurrences | 10 | 6 | 5 | 11 | 11 | 13 | 14 | 11 | 10 | 9 |

## Exercice 11

Un restaurant propose des menus composés d'une entrée d'un plat principal et d'un dessert.

Entrée : E1 – E2 – E3 – E4. Plat principal : P1 – P2 – P3 – P4. Dessert : D1-D2-D3-D4.

Affecter à chaque plat un prix.

Écrire une page HTML permettant de choisir un menu et d'afficher l'addition comme ci-dessous

### Composez votre menu

Choisissez une entrée, un plat et un dessert, puis appuyez sur le bouton "Envoi".

Entrée  Plat principal  Dessert

Entrée :  Plat :  Dessert  Total

## Exercice 12

Construire, en utilisant le CSS fourni, un formulaire contenant un pseudo, un age, un numéro de téléphone et un bouton valider de type "submit".

- Écrire la fonction `verifPseudo()` qui colore la zone de texte contenant le pseudo si la longueur du pseudo n'est pas comprise entre 2 et 8 caractères et renvoie `false` sinon elle renvoie `true`. On rappelle que si votre formulaire se nomme « `f` » et votre champ « `ch` » alors pour colorer votre zone de texte avec la couleur `#fba`, il suffit d'écrire :  
`document.f.ch.style.backgroundColor = "#fba";`  
Si l'utilisateur saisit un nouveau pseudo avec une longueur comprise entre 2 et 8 caractères, la couleur doit disparaître.  
On appellera : `document.f.ch.style.backgroundColor = "";`
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie du pseudo afin que la vérification se fasse à chaque fois que la zone de saisie perd le focus.
- Écrire la fonction `verifAge()` qui si l'age saisi n'est pas un nombre compris entre 5 et 100 colore la zone de texte contenant l'age avec la couleur `#fba` et renvoie `false` sinon elle renvoie `true`. On utilisera `isNaN`.
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie de l'age.
- Écrire la fonction `verifTel()` qui si le numéro de téléphone n'est pas un entier sur 10 chiffres dont le premier est un zéro colore la zone de texte contenant le numéro de téléphone avec la couleur `#fba` et renvoie `false` sinon elle renvoie `true`. On utilisera `isNaN`.
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie du numéro de téléphone.
- Écrire la fonction `verifForm(f)` qui prend en paramètre un formulaire, exécute les trois fonctions précédentes et renvoie `false` en affichant également une boîte de dialogue d'erreur si une des vérifications échoue. Elle renvoie `true` sinon.
- Ajouter dans la balise `form` : `onsubmit="return verifForm(this) "` pour bloquer l'envoi en cas d'erreur.

The image shows a web form titled "Formulaire de connexion". It has three input fields with labels "Pseudo :", "Age :", and "Num Tel :". Each label is on a yellow background. Below the input fields is a button labeled "Valider". The form is enclosed in a light gray border.

Nous allons maintenant voir une deuxième manière de procéder en ajoutant des attributs à nos fonctions :

- Écrire la fonction `surligne(champ, erreur)` qui prend en entrée une zone de saisie de formulaire et un booléen, et modifie sa couleur de fond si le booléen est à `vrai`. Cette fonction remet la couleur de fond à sa valeur par défaut si le booléen est `faux`.

- Construire la fonction `verifPseudo(champ)` qui en cas d'erreur sur le pseudo appelle la fonction `surligne(champ, true)` sinon elle appelle la fonction `surligne(champ, false)`.
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie en lui affectant « `this` » comme paramètre.
- D'une manière similaire à celle utilisée dans la fonction `verifPseudo(champ)`, écrire une fonction `verifAge(champ)` qui vérifie que l'age saisi est bien un nombre compris entre 5 et 100.
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie en lui affectant « `this` » comme paramètre.
- D'une manière similaire à celle utilisée dans la fonction `verifPseudo(champ)`, écrire une fonction `verifTel(champ)` qui vérifie que le numéro de téléphone est un entier sur 10 chiffres dont le premier est un zéro.
- Appeler cette fonction en interceptant l'événement `onblur` de la zone de saisie en lui affectant « `this` » comme paramètre.
- Écrire la fonction `verifForm(f)` qui prend en paramètre un formulaire, exécute les trois fonctions précédentes et renvoie `false` en affichant également une boîte de dialogue d'erreur si une des vérifications échoue. Elle renvoie `true` sinon.
- Ajouter dans la balise `form` : `onsubmit="return verifForm(this) "` pour bloquer l'envoi en cas d'erreur.
- 

### Exercice 13

Construire, en utilisant le CSS fourni, un formulaire contenant un login, un mot de passe, une date de naissance, un bouton valider de type "submit" et un bouton effacer.

On suppose que le login a une taille fixe de six caractères et que le mot de passe a une taille de 10 caractères. Indiquer ces contraintes en utilisant l'attribut `maxlength`.

En interceptant l'événement « `onKeyUp` », provoquer le changement de champ automatiquement du login vers le mot de passe et du mot de passe vers la date de naissance quand le nombre de caractères maximum fixé est atteint. Écrire le code sans utiliser de fonction puis refaire l'exercice en créant une fonction `suivant(encours, suivant, limite)`.

### Exercice 14

Faire une page pour la saisie des informations personnelles comme celle que l'on utilise lors de la déclaration de revenus ou à la CAF. Respecter le rendu demandé en utilisant le CSS fourni.

La figure ci-dessous illustre trois zones de saisie : Nom, Prénom et Profession. Dès que l'utilisateur a saisi son nombre  $X$  d'enfants et a donné le focus à un autre champ du formulaire, on a  $X$  zones de texte qui vont s'insérer entre la zone de saisie du nombre d'enfants et la zone de saisie de la profession. Notez bien que les zones de texte ajoutées ont chacune le texte « `Enfant i` » qui les précède avec « `i` » un entier correspondant au rang de l'enfant. Les zones de texte contiennent aussi des balises « `label` » qui permettent à l'utilisateur de cliquer simplement sur « `enfant i` » pour que le focus soit dans la zone de saisie relative au  $i^{\text{ème}}$  enfant.

Page d' accueil à son état initial :

Informations personnelles

Nom :

Prénom :

Nb enfant :

Profession :

Ok

Effacer

Si l'utilisateur a saisi 2 dans Nb enfant alors dès qu'il quitte ce champ, on obtient (sans recharger la page) l'état suivant :

Informations personnelles

Nom :

Prénom :

Nb enfant :

Enfant1 :

Enfant2 :

Profession :

Ok

Effacer

Si on remet 0 dans Nb enfant et qu'on libère le focus de Nb enfant, on revient à l'état initial de la page.

Proposer deux solutions : une en utilisant un div déjà présent mais vide que l'on remplira via `innerHTML` et une seconde solution en utilisant une création propre d'un nouveau nœud `DIV` du DOM au moment de la perte du focus (cf exemple du cours).