

Ed : les formulaires (2)(Verification Javascript)

exercice 1

1. ecrire un programme JS général qui parcourt le premier formulaire d'un document et affiche pour chacun de ses éléments les valeurs principales *vues* par Javascript.
2. appeler ce programme lors de la soumission du formulaire vu en cours
3. appeler ce programme lors de la soumission du formulaire vu dans le premier td.

exercice 2

On travaille sur l'exemple de formulaire vu en cours, dans lequel on donne aux boutons checkbox le meme nom *rt1*. Ecrire un programme JS qui s'exécute lors de la soumission du formulaire et qui affiche les données des cases à cocher *rt1* et des boutons radio *rt11*. Ce programme ne doit pas parcourir le formulaire élément par élément.

exercice 3

On travaille sur le formulaire de réservation de gîte.

1. Ecrire une fonction JS qui fait les vérifications concernant la désignation des gîtes. Cette fonction devra retourner true ou false.
2. Ecrire une fonction JS qui fait les vérifications concernant les dates. Elle retournera 0 si tout est ok, 1 ou 2 s'il n'y a pas de sélection de date de début ou de fin et 3 si la date de fin est antérieure à la date de début.
3. Ecrire une fonction JS qui fait les vérifications concernant les données personnelles. elle retournera une chaîne de caractères contenant soit ok soit une indication sur toutes les erreurs rencontrées.
4. Utiliser ces fonctions pour faire toutes les vérifications du formulaire lors de la soumission.

Une solution exercice 1

1. ecrire un programme JS général qui parcourt le premier formulaire d'un document et affiche pour chacun de ses éléments les valeurs principales *vues* par Javascript.

```
function voirselect(s){
var t=(" selectedIndex: " + s.selectedIndex + " long : " + s.length + "\n" );
for (var j=0;j<s.options.length;j=j+1){
    t=(t+ "option[" + j + "] text : " + s.options[j].text
        + " selected : " + s.options[j].selected
        + " value : " + s.options[j].value + "\n");
}
return t;
}

function voirRadio(s){
var t=(" value : " + s.value
        + " defaultChecked : " + s.defaultChecked
        + " checked : " + s.checked);
return t;
}

function voirPasswdTextSubmitTextarea(s){
var t=(" value : " + s.value + " defaultValue : " + s.defaultValue );
```

```

return t;
}

function voirFormulaire1(){
var elts=window.document.forms[0].elements;
var texte= " ";
var x,y,z;
texte=(texte + " nombre d'elemt :" + elts.length + "\n" );
texte=(texte + " description des elements :" + "\n");
for (var i=0;i<elts.length;i=i+1){
    x= (elts[i].type );
    y= (elts[i].tagName );
    z= (elts[i].name );
    texte=(texte+ " elements[" + i +"] ");
    texte=(texte + " balise :" + y + " type :"+ x + " name :" + z );
    if (x=="select-one"){
        texte=(texte + voirselect(elts[i]) );
    }
    if (x=="checkbox" || x=="radio"){
        texte=(texte+voirRadio(elts[i]) );
    }
    if ((x=="text" || x=="textarea")|| x=="submit" || x=="password"){
        texte=(texte + voirPasswdTextSubmitTextarea(elts[i]) );
    }
    texte=(texte + "\n");
}
alert(texte);
return false;
}

```

2. appeler ce programme lors de la soumission du formulaire vu en cours

```

<head>
    <script type="text/javascript" src="verifForm.js"></script>
    <title>premier formulaire </title>
</head>
<body>
    <form action="rien.html" method="post" onsubmit="return voirFormulaire1();"
    enctype="multipart/form-data" >
    ...

```

3. appeler ce programme lors de la soumission du formulaire vu dans le premier td.

```

<head>
    <script type="text/javascript" src="verifForm.js"></script>
    <title>formulaire de reservation</title>
</head>
<body>
    <form action="rien.html" method="post" onsubmit="return voirFormulaire1();"
    ...

```

Une solution exercice 2

On travaille sur l'exemple de formulaire vu en cours, dans lequel on donne aux boutons checkbox le meme nom *rt1*. Ecrire un programme JS qui s'exécute lors de la soumission du formulaire et qui affiche les données des cases à cocher *rt1* et des boutons radio *rt11*. Ce programme ne doit pas parcourir le formulaire élément par élément.

Pour manipuler les boutons radio ou checkbox de même nom comme une même entité, il faut y accéder par nom. `window.document.forms[0].elements["nomEntite"]` est alors un tableau de tous les éléments de même nom ;

```
function voirCheckrt1(){
var s=window.document.forms[0].elements["rt11"];
var texte=" ";
for( i =0;i<s.length; i=i+1){
    texte=(texte + "rt1[" + i + "]"  value : " + s[i].value
        + " defaultChecked : " + s[i].defaultChecked
        + " checked : " + s[i].checked + "\n");
}
alert( texte);
return false;
}
```

```
function voirRadiort1(){
var s=window.document.forms[0].elements["rt1"];
var texte=" ";
for( i =0;i<s.length; i=i+1){
    texte=(texte + "rt1[" + i + "]"  value : " + s[i].value
        + " defaultChecked : " + s[i].defaultChecked
        + " checked : " + s[i].checked + "\n");
}
alert( texte);
return false;
}
```

Une solution exercice 3

On travaille sur le formulaire de réservation de gîte.

1. Ecrire une fonction JS qui fait les vérifications concernant la désignation des gîtes. Cette fonction devra retourner true ou false.

```
function vGite(){
// on regarde si un gite a été choisi
/* 3 input checkbox avec le même nom :nomGite-> nomGite tableau a 3 cases
Pour savoir qui est coché : .checked */
var s=window.document.forms[0].elements["nomGite"];
var choix = false;
for(i=0; i<s.length; i++) {
    if(s[i].checked) choix = true;
}
return choix;
}
```

2. Ecrire une fonction JS qui fait les vérifications concernant les dates. Elle retournera 0 si tout est ok, 1 ou 2 s'il n'y a pas de sélection de date de début ou de fin et 3 si la date de fin est antérieure à la date de début.

```
function vDate(){
/* on vérifie que la date début est antérieure à la date de fin et
qu'elles sont sélectionnées toutes les deux.
on retourne 0(ok)* 1 2 3 (erreur début fin postérieur)*/
var f=window.document.forms[0];
var d=f.elements["dateDebut"].options.selectedIndex;
```

```

var f=f.elements["dateFin"].options.selectedIndex;
if(d==1){return 1;}
if(f==1){return 2;}
if(f<=d){return 3;}
return 0;
}

```

3. Ecrire une fonction JS qui fait les vérifications concernant les données personnelles. elle retournera une chaîne de caractères contenant soit ok soit une indication sur toutes les erreurs rencontrées.

```

function vCoord(){
// retourne une string : " "ok sinon les messages d'erreur rencontrés
var f=window.document.forms[0];
var mess="";
if(f.elements["nom"].value.length == 0)
    {mess=mess+"veuillez entrer votre nom \n";
    }
    if(f.elements["adresse"].value.length == 0)
    {mess=mess+"veuillez entrer une adresse \n";
    }
    if(f.elements["email"].value.length == 0)
    {mess=mess+"veuillez entrer une adresse email \n";
    }
    else {
var a,p;
a=f.elements["email"].value.indexOf('@');
p=f.elements["email"].value.indexOf('.');
if (a<1 || p<=a){mess=mess+"adresse email incorrecte";}
    }
    return mess;
}

```

4. Utiliser ces fonctions pour faire toutes les vérifications du formulaire lors de la soumission.

```

function verification(){
if (!vGite()){
    alert("veuillez choisir un gîte");
    return false;
}
var vd=vDate();
if(vd==1){
    alert("entrer une date de debut");
    return false;
}
if(vd==2){
    alert("entrer une date de fin");
    return false;
}
if(vd==3){
    alert("la fin du séjour n'est pas postérieure au début");
    return false;
}
var message=vCoord();
if (message!=""){

```

```
    alert(message);  
    return false;  
}  
return true;  
}
```