JavaScript

Formularprüfung

```
function emailverify(email) {
    return (email.search(/^w+((-w+)|(.w+))*@[A-Za-z0-9]+((.|-)[A-Za-
z0-9]+)*.[A-Za-z0-9]+$/) != -1);
}
function CheckRegister(f)
{
  if (f.username.value==""){alert('Benutzername fehlt');}
  else if (!f.username.value.match(/^[a-zA-Z0-9_-]+$/)){alert('Benutzername
ungültig');}
  else if (f.password.value==""){alert('Passwort fehlt');}
  else if (f.password.value.length<8){alert('Passwort zu kurz');}
  else if (f.password.value!=f.password2.value){alert('Passwörter nicht
gleich');}

  else return true;
  return false;
}
```

```
<form>
  <label for="username">Benutzername: </label><input type="text"
name="username" id="username" maxlength="20" />
  <label for="password">Passwort: </label><input type="password"
name="password" id="password" />
  <label for="password2">PW wiederholen: </label><input type="password"
name="password2" id="password2" />
  <input type="submit" name="register" value="registrieren" onclick="return
CheckRegister(this.form);"/>
</form>
```

Datumsfunktionen (de):

```
function GerDate(d) //formatiert ein JavaScript-Date-Objekt nach deutschem
Format dd.mm.yyyy (wg. Probleme mit dateobj.toLocaleStr)
{
  var month = d.getMonth()+1;
  var year = d.getYear();
  var day = d.getDate();
  if(day<10) day = "0" + day;
  if(month<10) month= "0" + month;
  if(year<1000) year+=1900;
  return day+'.'+month+'.'+year;
}
```

```
function heute() //liefert das aktuelle Datum in deutscher Formatierung
{
  var today = new Date();
  var month = today.getMonth()+1;
  var year = today.getYear();
  var day = today.getDate();
  if(day<10) day = "0" + day;
  if(month<10) month= "0" + month;
  if(year<1000) year+=1900;
  return day+'.'+month+'.'+year;
}

function GerDay(mod) //Datumsmanipulation vom aktuellen Datum +/- x Tage
{
  var now = new Date();
  var today = now.getTime();
  if (mod)
  {
    var than = today + (mod * 24 * 60 * 60 * 1000);
    now.setTime(than);
  }
  var month = now.getMonth()+1;
  var year = now.getYear();
  var day = now.getDate();
  if(day<10) day = "0" + day;
  if(month<10) month= "0" + month;
  if(year<1000) year+=1900;
  return day+'.'+month+'.'+year;
}

function jetzt(sec)
{
  var today = new Date();
  var hour = today.getHours();
  var minute = today.getMinutes();
  var second = today.getSeconds();
  if(hour<10) hour= "0" + hour;
  if(minute<10) minute="0" + minute;
  if(second<10) second="0" + second;
  if (sec)
    return hour+':'+minute+':'+second;
  else
    return hour+':'+minute;
}

function calc_endtime(f) //rechnet x minuten (feld duration) auf datum in
starttime
{
    var dtstr = f.starttime.value;
    if (dtstr.match(/^\d\d\.\d\d\.\d\d\d\d \d\d:\d\d$/))
    {
```

```
        var dt = dtstr.split(" ");//datum und zeit trennen
        var d=dt[0].split(".");
        var t=dt[1].split(":");
        var ts  = new Date(d[2], d[1]-1, d[0]);
        //alert(t[0]+'h'+t[1]+'+'+f.duration.value);
        ts.setHours(t[0],parseInt(t[1])+parseInt(f.duration.value));
        var endtime = ("0" + ts.getDate()).slice(-2) + "." +
("0"+(ts.getMonth()+1)).slice(-2) + "." +
            ts.getFullYear() + " " + ("0" + ts.getHours()).slice(-2) + ":"
+ ("0" + ts.getMinutes()).slice(-2);
        f.endtime.value=endtime;
    }else alert(dtstr+' hat falsches Format!');
}
```

label-Text

```
function GetLabelText(cb_id)
{
  var t="";
  var labels=document.getElementsByTagName('label');
  for (var i = 0; i < labels.length; i++)
  {
    if (labels[i].htmlFor==cb_id)
      t=labels[i].innerHTML;
    //alert(labels[i].title);
  }
  return t;
}

function SetLabelText(cb_id,txt)
{
  var labels=document.getElementsByTagName('label');
  for (var i = 0; i < labels.length; i++)
  {
    if (labels[i].htmlFor==cb_id)
      labels[i].innerHTML=txt;
    //alert(labels[i].title);
  }
}
```

label zu Input finden

https://stackoverflow.com/a/285608

Abmessungen/Position

```
function GetRect (o)
{
  var r = { top:0, left:0, width:0, height:0 };

  if(!o) return r;
  else if(typeof o == 'string' ) o = document.getElementById(o);
  if( typeof o != 'object' ) return r;

  if(typeof o.offsetTop != 'undefined')
  {
    r.height = o.offsetHeight;
    r.width = o.offsetWidth;
    r.left = r.top = 0;

    while (o)
    {
      r.top  += parseInt( o.offsetTop );
      r.left += parseInt( o.offsetLeft );
      o = o.offsetParent;
    }
  }
  return r;
}

function SetRect(e,l,t,w,h)
{
  //alert('[SetRect] '+l+' '+t+' '+w+' '+h);
  if(typeof e == 'string' ) e = document.getElementById(e);
  if (typeof e == 'object')
  {
    e.style.left=l+'px';
    e.style.top=t+'px';
    if (w>0) e.style.width=w+'px';
    if (h>0) e.style.height=h+'px';
  } else alert('[SetRect] e not found!\n'+typeof e);
}

function SetHeight(e,v)
{
  if(typeof e == 'string' ) e = document.getElementById(e);
  if (typeof e == 'object')
  {
    e.style.height=v+'px';
  } else alert('[SetHeight] e not found!\n'+typeof e);
}

function SetWidth(e,v)
{
  if(typeof e == 'string' ) e = document.getElementById(e);
  if (typeof e == 'object')
  {
```

```
    e.style.width=v+'px';
  } else alert('[SetWidth] e not found!\n'+typeof e);
}
```

AJAX

```
var loading_html='<img src="images/loading.gif">';

function setInnerText(divID, text)
{
  if (divID)
  {
    var el = (document.getElementById) ? document.getElementById(divID) :
document.all[divID];
    if(el) { el.innerHTML = text; }else alert('[AJAX] div "'+divID+'" not
found');
  }
}

// create Object
function GetHttpRequest()
{
  if (window.XMLHttpRequest)
    return new XMLHttpRequest();// Mozilla, Safari, Opera, IE>=7
  else if (window.ActiveXObject)
  {
    try
    {
      return new ActiveXObject('Msxml2.XMLHTTP'); // IE 5
    } catch (e)
    {
      try
      {
        return new ActiveXObject('Microsoft.XMLHTTP'); // IE 6
      } catch (e) {}
    }
  }else
  {
    alert("Your browser does not support AJAX.");
  }
  return null;
}

function request_get(xmlHttp,serverscript,params)
{
  if(xmlHttp)
  {
    xmlHttp.open("GET", serverscript+'?'+params, true);
    xmlHttp.send(null);
```

```
  }else alert('HttpRequest wurde nicht erstellt!');
}
function request_post(xmlHttp,serverscript,params)
{
  if(xmlHttp)
  {
    xmlHttp.open("POST", serverscript, true);

    //Send the proper header information along with the request
    xmlHttp.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    xmlHttp.setRequestHeader("Content-length", params.length);
    xmlHttp.setRequestHeader("Connection", "close");
    xmlHttp.send(params);
  }else alert('HttpRequest wurde nicht erstellt!');
}

function ajax_request(divID, url, params,callback)
{
  var xmlHttp = GetHttpRequest();
  if (xmlHttp)
  {
    if(callback == undefined) //if callback defined no loading-image
    setInnerText(divID,loading_html);
    xmlHttp.onreadystatechange = function()
    {
      if(xmlHttp.readyState == 4)
      {
        if(callback != undefined && typeof callback == 'function')
            callback(divID, xmlHttp.responseText);
        else
            setInnerText(divID, xmlHttp.responseText);
      }
    }
      request_post(xmlHttp,url,params);
  }
  return false;
}
```

json-parsing

```
function applyState(div,data)
{
    if(data) {
        try {
            obj = JSON.parse(data);
        } catch(e) {
            alert(e);
        }
```

```
    }
    if (obj)
    {
        console.log(obj);
        if (obj.hasOwnProperty('checks'))
        {
            for (var key in obj.checks) {
                console.log(key + " -> " + obj.checks[key].state);
            }
        }
        if (obj.hasOwnProperty('trails'))
        {
            for (var key in obj.trails) {
                console.log(key + " -> " + obj.trails[key].working);
            }
        }
    }
    setInnerText(div, '<pre>[applyState_'+jetzt(true)+']\n'+data+'</pre>');
}
```

ein/ausblenden

```
function ShowDivTag(e,show)
{
  if (e)
  {
    if (show=='toggle') show=(e.style.display=='none');
    if (show)
      e.style.display='';
    else
      e.style.display='none';
  }
}

function ShowDiv(id,show)
{
  var e=document.getElementById (id);
  ShowDivTag(e,show);
}
```

Radiobutton/Checkbox

```
function GetRadioValue(radioname)
{
  //alert(radioname);
  if (radioname)
  {
    if(radioname.length==undefined)//wenn nur ein radio-button existiert
```

```
    {
      if(radioname.checked==true)
        return radioname.value;
    }else
    {
      for(var i=0; i<radioname.length; i++)
      {
        if(radioname[i].checked == true)
            return radioname[i].value;
      }
    }
  }else alert('radiobutton nicht gefunden!');

  return -1;//if nothing found
}

function SetRadioValue(radioname,val)
{
  //alert(radioname);
  if (radioname)
  {
    if(radioname.length==undefined)//wenn nur ein radio-button existiert
    {
      if(radioname.value==val)
      {
        radioname.checked=true;
        return true;
      }
    }else
    {
      for(var i=0; i<radioname.length; i++)
      {
        if(radioname[i].value==val)
        {
      radioname[i].checked=true;
      return true;
    }
      }
    }
  }else alert('radiobutton nicht gefunden!');

  return false;//if nothing found
}

function GetCheckboxValues(chk_name,sep)
{
  //var s=GetCheckboxValues('chk[]',',');
  var sOut = "";
  if (!sep) sep=',';
  var cb = document.getElementsByName(chk_name);
  if (cb!=null)
```

```
  {
    for(var i=0;i<cb.length;i++)
    {
      if (cb[i].checked==true)
      {
        if (sOut!="") sOut+=sep;
        sOut += cb[i].value;
      }
    }
  }
  return sOut;
}
```

Class-Manipulation (css)

```
function addClass (element, className) {
    if (!hasClass(element, className)) {
        if (element.className) {
            element.className += " " + className;
        } else {
            element.className = className;
        }
    }
}

function removeClass (element, className) {
var regexp = addClass[className];
    if (!regexp) {
        regexp = addClass[className] = new RegExp("(^|\\s)" + className +
"(\\s|$)");
    }
    element.className = element.className.replace(regexp, "$2");
}

function hasClass (element, className) {
    var regexp = addClass[className];
    if (!regexp) {
        regexp = addClass[className] = new RegExp("(^|\\s)" + className +
"(\\s|$)");
    }
    return regexp.test(element.className);
}

function toggleClass (element, className) {
    if (hasClass(element,className)) {
        removeClass(element,className);
    } else {
        addClass(element,className);
    }
```

```
}
```

Zufallsfarbe

```
function get_random_color()
{
  return '#' + (Math.floor(Math.random() * 2 **
24)).toString(16).padStart(0, 6);
}

function randomColor()
{
color='rgb('+Math.round(Math.random()*255)+','+Math.round(Math.random()*255
)+','+Math.round(Math.random()*255)+')';

     return color;
}
```

Elemente hinzufügen

```
function myclick(e){alert('click '+e.id+' img-src:'+e.src);}

function createnode()
{
    var e=document.getElementById('icons');
    if (e)
    {
        var node=document.createElement("div");
        node.style.height='16px';
        node.style.width='16px';
        node.style.border = "1px solid #000000";
        node.style.display = "inline-block";
        node.style.backgroundColor=randomColor();
        node.title='hint '+node.style.backgroundColor;
        e.appendChild(node);
        //second node, an img
        var node2=document.createElement("img");
        node2.id='img_'+Math.round(Math.random()*255);
        if (Math.random()*2 > 1)
        {
            node2.src='burger.png';
        }else{
            node2.src='coffee.png';
        }
        node2.setAttribute('onclick', 'myclick(this);');
        e.appendChild(node2);
    }
```

```
}
```

```
<button onclick="createnode();">add child</button>
<button onclick="var e=document.getElementById('icons');if (e)
e.innerHTML='';">delete all childs</button>
<div id="icons" style="max-width:200px"></div>
```

From:
http://fw-web.de/dokuwiki/ - **FW-WEB Wiki**

Permanent link:
http://fw-web.de/dokuwiki/doku.php?id=programming:javascript

Last update: **2023/06/08 17:06**