|  |  |
| --- | --- |
| …..  <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>  JS лаб. работа №7  </title>  </head>  <body>  <button name='button' onclick="langSwitch();"></button>  <p>  <b><a name="task"></a>1:</b><br>  </p>  <table border="solid">  <tr id="month">  </tr>  <tr id="day">  </tr>  </table>  <p>  <b><a name="task"></a>2:</b><br>  </p>  <a name="before"></a>  <input type="text" id="session" disabled="disabled">  <a name="days"></a>  <p>  <b><a name="task"></a>3:</b><br>  </p>  <button onclick="days();" name="button"></button>  <a name="passed"></a>  <input type="button" id="days" disabled="disabled">  <a name="days"></a>  <p>  <b><a name="task"></a>4:</b><br>  </p>  <table border="solid">  <tr>  <td>  <p><a name="element"></a>1:</p>  </td>  Module.createNamespace("interface.lang.ru");  Module.createNamespace("interface.lang.eng");  Module.registerInitializationFunction(initNamespaceRu);  Module.registerInitializationFunction(initNamespaceEng);  Module.registerInitializationFunction(init);  Module.runInitializationFunctions();  date();  session();  render();  function initNamespaceRu() {  var nsRu = interface.lang.ru;  nsRu.lang = function() {  return "ru-Ru"  }  nsRu.task = function() {  return "Задание ";  }  nsRu.before = function() {  return "До сессии осталось: ";  }  nsRu.passed = function() {  return "Уже прошло: ";  }  nsRu.result = function() {  return "Результат: ";  }  nsRu.days = function() {  return " дней";  }  nsRu.button = function() {  return "Кнопка";  }  nsRu.element = function() {  return "Элемент ";  }  nsRu.error = function() {  return "Некоректный ввод";  }  nsRu.inputDate = function() {  return "Введите дату в формате ГГГГ-ММ-ДД";  }  nsRu.date = function() {  return "Введите дату в формате ГГГГ/ММ/ДД";  }  }  function initNamespaceEng() {  var nsEng = interface.lang.eng;  nsEng.lang = function() {  return "en-Us";  }  nsEng.task = function() {  return "Task ";  }  var month = document.getElementById("month");  var year = document.getElementById("year");  day.innerHTML = date.toLocaleString(interface.lang.current.lang(), {weekday: 'long'});  month.innerHTML = date.toLocaleString(interface.lang.current.lang(), {day: 'numeric', month: 'long', year: 'numeric'});  }  function session() {  var date = new Date();  var s = new Date(2017, 5, 16);  date.setHours(0, 0, 0, 0);  date = s - date;  var session = document.getElementById("session");  session.value = (date / 86400000 - 1);  }  function days() {  var s = prompt(interface.lang.current.inputDate());  s = Date.parse(s);  var date = new Date();  date.setHours(0, 0, 0, 0);  date = date - s;  var days = document.getElementById("days");  days.value = (Math.floor(date / 86400000) + 1);  }  function sum() {  var first = document.getElementById("first").value;  var last = document.getElementById("last").value;  if (isNaN(first) || isNaN(last) || first == "" || last == "" || first == null || last == null) {  alert(interface.lang.current.error());  return;  }  var r = 0;  for (var i = Number(first); i <= Number(first) + Number(last); i++)  r += (Math.pow((-1), (i - 1)) \* Math.pow((i + 4), 2));  sumresult.value = r;  }  function search() {  var found = document.getElementById("found");  var res = document.getElementById("res");  res.innerHTML = " " + found.value.search(/\d\d\d\d[/]\d\d[/]\d\d/);  }  if (typeof from == "string") from = Module.modules[from];  if (!from || typeof from != "object")  throw new Error("Module.importSymbols(): требуется указать объект пространства имен");  var to = Module.globalNamespace;  var symbols = [];  var firstsymbol = 1;  if (arguments.length > 1 && typeof arguments[1] == "object") {  if (arguments[1] != null) to = arguments[1];  firstsymbol = 2;  }  for (var a = firstsymbol; a < arguments.length; a++)  symbols.push(arguments[a]);  if (symbols.length == 0) {  if (from.EXPORT) {  for (var i = 0; i < from.EXPORT.length; i++) {  var s = from.EXPORT[i];  to[s] = from[s];  }  return;  }  else if (!from.EXPORT\_OK) {  for (s in from) to[s] = from[s];  return;  }  }  var allowed;  if (from.EXPORT || from.EXPORT\_OK) {  allowed = {};  if (from.EXPORT)  102  for (var i = 0; i < from.EXPORT.length; i++)  allowed[from.EXPORT[i]] = true;  if (from.EXPORT\_OK)  for (var i = 0; i < from.EXPORT\_OK.length; i++)  allowed[from.EXPORT\_OK[i]] = true;  }  for (var i = 0; i < symbols.length; i++) {  var s = symbols[i];  if (!(s in from))  throw new Error("Module.importSymbols(): символ " + s + " не определен ");  if (allowed && !(s in allowed))  throw new Error("Module.importSymbols(): символ " + s +  " не является общедоступным " + "и не может быть импортирован.");  to[s] = from[s];  }  };  Module.registerInitializationFunction = function(f) {  Module.\_initfuncs.push(f);  Module.\_registerEventHandler();  }  Module.runInitializationFunctions = function() {  for (var i = 0; i < Module.\_initfuncs.length; i++) {  try {  Module.\_initfuncs[i]();  } catch (e) { } | <td>  <input type="text" id="first">  </td>  </tr>  <tr>  <td>  <p><a name="element"></a>2:</p>  </td>  <td>  <input type="text" id="last">  </td>  </tr>  </table>  <p>  <button onclick="sum();" name="button"></button>  <a name="result"></a>  <input type="text" id="sumresult" disabled="disabled">  </p>  <p>  <b><a name="task"></a>5:</b>  </p>  <a name="date"></a>  <p>  <button onclick="search();" name="button"></button>  <input type="text" id="found">  <a id="res"></a>  </p>  <script type="text/javascript" src="js/module.js"></script>  <script type="text/javascript" src="js/script.js"></script>  </body>  </html>  nsEng.before = function() {  return "Before session: ";  }  nsEng.passed = function() {  return "Already passed: ";  }  nsEng.result = function() {  return "Result: ";  }  nsEng.days = function() {  return " day's";  }  nsEng.button = function() {  return "Button ";  }  nsEng.element = function() {  return "Element ";  }  nsEng.error = function() {  return "Incorrect input";  }  nsEng.inputDate = function() {  return "Enter the date in the format YYYY-MM-DD";  }  nsEng.date = function() {  return "Enter the date in the format YYYY/MM/DD";  }  }  function init() {  interface.lang.current = interface.lang.ru;  }  function render() {  var names = ["before", "passed", "days", "result", "task", "button", "element", "date"];  for (var k in names) {  var elem = document.getElementsByName(names[k]);  for (var i in elem)  elem[i].innerHTML = interface.lang.current[names[k]]();  }  }  function langSwitch() {  if (interface.lang.current == interface.lang.ru)  interface.lang.current = interface.lang.eng;  else  interface.lang.current = interface.lang.ru;  date();  render();  }  function date() {  var date = new Date();  var day = document.getElementById("day");  var Module;  if (Module && (typeof Module != "object" || Module.NAME))  throw new Error("Пространство имен 'Module' уже существует");  Module = {};  Module.NAME = "Module";  Module.VERSION = 0.1;  Module.EXPORT = ["require", "importSymbols"];  Module.EXPORT\_OK = ["createNamespace", "isDefined",  "registerInitializationFunction",  "runInitializationFunctions",  "modules", "globalNamespace"  ];  Module.globalNamespace = this;  Module.modules = {  "Module": Module  };  Module.createNamespace = function(name, version) {  if (!name) throw new Error("Module.createNamespace(): не указано имя");  if (name.charAt(0) == '.' ||  name.charAt(name.length - 1) == '.' ||  name.indexOf("..") != -1)  throw new Error("Module.createNamespace(): неверное имя: " + name);  100  var parts = name.split('.');  var container = Module.globalNamespace;  for (var i = 0; i < parts.length; i++) {  var part = parts[i];  if (!container[part]) container[part] = {};  else if (typeof container[part] != "object") {  var n = parts.slice(0, i).join('.');  throw new Error(n + " уже существует, но не является объектом");  }  container = container[part];  }  var namespace = container;  if (namespace.NAME) throw new Error("Модуль " + name + " уже определен ");  namespace.NAME = name;  if (version) namespace.VERSION = version;  Module.modules[name] = namespace;  return namespace;  }  Module.isDefined = function(name) {  return name in Module.modules;  };  Module.require = function(name, version) {  if (!(name in Module.modules)) {  throw new Error("Модуль " + name + " не определен");  }  if (!version) return;  var n = Module.modules[name];  if (!n.VERSION || n.VERSION < version)  throw new Error("Модуль " + name + " имеет версию " +  n.VERSION + " требуется версия " + version + " или выше.");  };  Module.importSymbols = function(from) {  }  Module.\_initfuncs.length = 0;  }  Module.\_initfuncs = [];  Module.\_registerEventHandler = function() {  var clientside =  "window" in Module.globalNamespace && "navigator" in window;  if (clientside) {  if (window.addEventListener) {  window.addEventListener("load", Module.runInitializationFunctions, false);  } else if (window.attachEvent) {  window.attachEvent("onload", Module.runInitializationFunctions);  } else {  window.onload = Module.runInitializationFunctions;  }  }  130  Module.\_registerEventHandler = function() {};  } |