

AKADEMICKÝ INFORMAČNÝ SYSTÉM

SvF | SjF | FEI | FCHPT | FAD | MTF | FIIT

Prihlásený: Rosaldo 0 správ 0 dokumentov 0 úloh



Testy a skúšanie

Testy k vypracovaniu

Odozvané testy

Podrobnosti o teste VAVJS Skuska OT

Pri otázkach, u ktorých bolo upravené bodové hodnotenie, sú vypísané tri bodové hodnoty, v poradí pôvodných výsledkov otázky, zvýšené alebo znížené hodnotenie a ich súčet - hodnota výsledná.



Z teste bolo získaných 35,00 z maximálnych 50 bodov, t.j. podľa nastavenia prepočítaná úspešnosť 70,00 %.



Pri vyhodnotení budú udelené body iba za úplne správne zodpovedanú otázku. Za čiastočné odpovede nebudú žiadne body udelené.

Pomocou nasledujúceho filtra môžete obmedziť zobrazené otázky.

Zobraziť iba otázky s úspešnosťou

--



%

Obmedziť

1. Napiste funkciu sum1(arr) ktora scita vsetky neparne cisla v poli "arr".

2 z 2 b.

Priklad:

```
var a = sum1([1,2,3,4]); // a == 4;
```

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
function sum1(arr){  
  return arr.filter((item) => item % 2 == 1).reduce((acc,it) => acc+it,0)  
}
```

Komentár: Vyhodnotené bez komentára.

2. Ktore z moznosti su platne definicie premennej v JavaScripte.

1 z 1 b.

Vyberte ľubovoľný počet možných odpovedí. Správna nemusí byť žiadna, ale tiež môžu byť správne všetky.

Správna

Vybraná

define a = 4;

number a = 4;

int a = 4;

let a = 4;

3. Aka je uloha V8 v JavaScripte?

1 z 1 b.

Vyberte ľubovoľný počet možných odpovedí. Správna nemusí byť žiadna, ale tiež môžu byť správne všetky.

Správna

Vybraná

V8 je verzia jazyka JavaScript

V8 je JavaScript kniznica pre manazment asynchronnych funkcii

V8 je JavaScript engine pouzivany v node.js

V8 je JavaScript engine pouzivany v prehliadacoch

Správna	Vybraná	
---------	---------	--

4. 1 z 1 b.

Aky je rozdiel medzi == a ===?

Vyberte iba jednu z nasledujúcich možných odpovedí.

Správna	Vybraná	
<input type="radio"/>	<input checked="" type="radio"/>	== porovnáva hodnoty, === porovnáva hodnoty aj typy
<input type="radio"/>	<input type="radio"/>	== porovnáva typy, === porovnáva hodnoty aj typy
<input type="radio"/>	<input type="radio"/>	== porovnáva typy, === porovnáva hodnoty
<input type="radio"/>	<input type="radio"/>	== porovnáva hodnoty, === porovnáva typy

5. 0 z 2 b.

Aky je rozdiel medzi promise-ami a callback-ami v JavaScripte;

Vyberte ľubovoľný počet možných odpovedí. Správna nemusí byť žiadna, ale tiež môžu byť správne všetky.

Správna	Vybraná	
<input type="radio"/>	<input checked="" type="radio"/>	promisy sa vykonávajú v presnom poradí, callbacky nie
<input type="radio"/>	<input type="radio"/>	callbacky je možné použiť s async/await, promisy nie
<input type="radio"/>	<input checked="" type="radio"/>	promisy je možné použiť s async/await, callbacky nie
<input type="radio"/>	<input checked="" type="radio"/>	promisy môžu byť rejectnuté, callbacky nie

6. 2 z 2 b.

Ako JavaScript reaguje pri chybe

Vyberte ľubovoľný počet možných odpovedí. Správna nemusí byť žiadna, ale tiež môžu byť správne všetky.

Správna	Vybraná	
<input type="radio"/>	<input type="radio"/>	JavaScript zaznamená chybu a pokračuje vo vykonávaní
<input type="radio"/>	<input checked="" type="radio"/>	JavaScript prestane vykonávať ďalšie príkazy a vyhodí chybu
<input type="radio"/>	<input type="radio"/>	JavaScript zastaví vykonávanie a vyzve používateľa na opravu chyby
<input type="radio"/>	<input type="radio"/>	JavaScript ignoruje chybu a pokračuje vo vykonávaní

7. 1 z 1 b.

Co je node.js.

Vyberte iba jednu z nasledujúcich možných odpovedí.

Správna	Vybraná	
<input type="radio"/>	<input type="radio"/>	knihovna jazyka JavaScript pre tvorbu grafických aplikácií
<input type="radio"/>	<input checked="" type="radio"/>	prekladac jazyka JavaScript umoznujuci beh JavaScript aplikacii na zariadeniach (bez nutnosti prehliadaca)
<input type="radio"/>	<input type="radio"/>	specifikacia jazyka JavaScript
<input type="radio"/>	<input type="radio"/>	nastroj na optimalizaciu webovych stranok

8. 0 z 2 b.

Co je TypeScript a ako sa odlišuje od JavaScriptu?

Vyberte ľubovoľný počet možných odpovedí. Správna nemusí byť žiadna, ale tiež môžu byť správne všetky.

Správna	Vybraná	
<input type="radio"/>	<input type="radio"/>	TypeScript je jazyk transpilovaný do jazyku Java
<input type="radio"/>	<input checked="" type="radio"/>	TypeScript je programovací jazyk, ktorý umožňuje využívanie a kontrolu typov pri preklade do jazyka JavaScript

Správna

Vybraná



TypeScript je špecifikácia jazyka JavaScript

TypeScript je transpilátor ktorý preklada JavaScript do JavaScriptu

9.

0 z 4 b.

Predpokladáme stránku obsahujúcu:

```
<button id="prev">prev</button>
```

```
<img id="current" src="" />
```

```
<button id="next">next</button>
```

```
<script></script>
```

Napíšte obsah "script" elementu tak, aby tlačidlá vedeli meniť obrázky v img elemente podľa odpovede zo serveru. Pri stlačení tlačidla "prev" sa má načítať predosly obrázok, pri stlačení tlačidla "next" sa má načítať nasledujúci obrázok. Adresa obrázka sa získava zo serveru na adrese localhost:8089 pri požiadavke na /img kde parameter id bude obsahovať id obrázku. ID obrázku je 0+. Podtečenie nemusíte ošetrovať.

Príklad požiadavky na obrázok: http://localhost:8089/img?id=5

Príklad odpovede na predoslu požiadavku: "https://www.example.com/cat.jpg"

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne. Študent na otázku neodpovedal.

10.

4 z 5 b.

Napíšte HTTP server pre node.js ktorý bude počúvať na adrese http://localhost:8089. Po prijatí požiadavky na "/quote" metódou GET bude server posielat požiadavku na server http://quotes.example.com/qoute a preposielat túto odpoveď klientovi tak ako ju dostal. Predpokladajte že node.js má fetch rovnakej syntaxe ako u prehliadačov. Všetky odpovede sú typu JSON.

Komunikácia: požiadavka 1.:klient -> localhost:8089, požiadavka 2.: localhost:8089 -> quotes.example.com

Odpoveď na požiadavku 2. quotes.example.com -> localhost:8089, odpoveď na požiadavku 1. localhost:8089 -> klient

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
const http = require("http")

const server = http.createServer((req,res)=>{
  if(req.url == "/quote" && req.method == "GET"){
    fetch("http://quotes.example.com/qoute").then(response => response.json()).then(data => {
      res.writeHead(200,{"Content-Type": "application/json"})
      res.write(JSON.stringify(data))
      res.end()
    })
  }
}).listen(8089)
```

Číslo
otázky

Otázka

Body

komentár: response.json() je funkcia

11.

2 z 2 b.

Napíšte promise ktorý bude po 2s vykonávať resolve s reťazcom "promise resolvnuty". Použite tento promis v async funkcii ktorá bude do konzoly vypisovať hodnotu vrátenú z promise.

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
function pr(){
  return new Promise((resolve,reject)=> setTimeout(()=>{resolve("promise resolvnuty")},2000))
}

async function fun(){
  pr().then((data)=> console.log(data))
}

fun()
```

Komentár: Vyhodnotený bez komentára.

12.

5 z 5 b.

Napíšte React aplikáciu, ktorá bude mať komponent <But> a komponent <Out>. Komponent <But> má obsahovať tlačidlo. Po stlačení tlačidla sa má vykonať požiadavka na HTTP server (použite syntax fetch z prehliadaču) s adresou http://localhost:8089/req. Server na tejto adrese vráti JSON hodnotu "resp" (príklad: {"resp": "ok"}). Komponent out bude obsahovať HTML element <div>. Po prijatí odpovede zo serveru aplikácia nastaví elementu <div> komponentu <Out> hodnotu "resp" z JSONu. Príklad: z odpovede serveru {"resp": "ok"} bude komponent <Out> obsahovať <div>ok</div>.

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
import React,{useState} from "react"
import ReactDOM from "react-dom"

function But(props){ return <button onClick={props.onClick}>Submit</button>}

function Out(props){ return <div>{props.value}</div>}

function App(){
  const [text,setText] = useState("")
  return(<div><But onclick={() => fetch("http://localhost:8089/req").then(data =>
data.json).then(response =>
setText(response.resp))} /><Out value={text} /></div>)
}

const root = ReactDOM.createRoot(document.getElementById("root"))
```

Číslo
otázky

Otázka

Body

13.

1 z 3 b.

Napiste funckiu addAd(img, link), ktora po 3 sekundach bude do tela stranky pridavat obrazok s adresou "img" odkazujuci na stranku s adresou "link". Po kliknutí na reklamu sa musi zvysit interne pocitadlo a otvorit odkaz. Pridanie obrazku ma byt jednorazove, nie opakovane.

Priklad: addAd('https://www.example.com/dog.jpg','https://www.example.com/about/dogs');

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
let obj = {}

function helper(img,link){
  var a = document.createElement("img")
  a.src = link;
  document.body.appendChild(a)
}

function addAd(img,link){
  setTimeout(helper(img,link),3000)
}
```

Komentár:

helper sa zavola ihned

chyba click listener

14.

3 z 3 b.

Napiste funkciu straeiou, ktora berie ako parameter retazec str a vracia pocet samohlasok v tomto retazci. Pre zjednodusenie moze mat retazec str iba znaky [a-z].

Priklad volania: straeiou('retazec'); // 3

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
function straeiou(str){
  let count = 0;
  for(i=0;i<str.length;i++){
    if(str.charAt(i) == "a" || str.charAt(i) == "e" || str.charAt(i) == "i" || str.charAt(i) ==
"o" || str.charAt(i) ==
"u" || str.charAt(i) == "y"){count++}
  }
  return count;
}
```

Komentár: Vyhodnotené bez komentára.

15.

2 z 2 b.

Napiste funkciu rWords(str) ktora bude brat parameter "str" obsahujuci slova oddelene medzerou a vracat pocet slov v retazci "str" zacinajucich pismenom "s".

var a = rWords('raketa strom ryba chlieb'); // a == 1

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
function rWords(str){
  count = 0;
  str = str.split(" ")
  str.forEach((item)=> {if(item.charAt(0) == "s"){count++}})
  return count;
}
```

Komentár: Vyhodnotené bez komentára.

16.

4 z 4 b.

Vyberte 4 fazy event-loop v JavaScripte a vysvetlite ich.

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

The Poll Phase - vykonava I/O operacie, ktore vykonavame (napr vypis console logu)

Timers - Po vykonani synchronneho kodu, event loop pokracuje do dalsich faz, vo faze timers prejde danu frontu a vykona dane funkcie. Preto napr. pokiaľ máme `setTimeout(()=>{console.log("hey")},0); console.log("a");` Prvy sa vykona `console.log("a")`, aj napriek tomu ze timeout nema nastavene oneskorenie. Okrem `setTimeout` sem patri aj `setInterval`

Check - sa vykonava v check faze, hned po poll faze, patri sem napr funkcia `setImmediate`

Close Callback - vsetky ukoncovacie callback eventy - napr. `process exit`, zatvorenie websocketu, tu event loop ukoncuje svoj cyklus a zacina znova

Číslo
otázky

Otázka

Body

Komentár: Vyhodnotené bez komentára.

17.

0 z 3 b.

Popiste ako sa pracuje s eventami (ako napr. "click") v jazyku JavaScript v prehliadacoch a v node.js.

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

No ze sa klikne a jede to ... 3/3

Komentár: Vyhodnotené bez komentára.

18.

3 z 3 b.

Ako sa pouziva funkčna a class syntax pri definicii tried. Uvedte priklady na triede Auto, ktora moze mat atributy "vyrobca" a "rokVyroby".

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
/* funkčna */
function Auto(vyrobca,rokVyroby){
    this.vyrobca = vyrobca
    this.rokVyroby = rokVyroby
}

let a = new Auto("lorem",2009)

/* Class */

class Auto{
    constructor(vyrobca,rokVyroby){
        this.vyrobca = vyrobca
        this.rokVyroby = rokVyroby
    }
}

let a = new Auto("ipsum",2010)
```

Komentár: Vyhodnotené bez komentára.

19.

3 z 4 b.

Napiste funkciu `mostProductive` pre kniznicny system, ktora ako parameter bude brat pole objektov library so strukturou: `[{"autor": "Ezop", "titul": "Vlk a zajac"},...]`, spočíta počet titulov pre danyh autorov a vracať pole objektov so strukturou: `[{"autor": "Ezop", "pocet": 4},...]`, zotriedene od najproduktivnejšieho autora (najvacsi počet titulov) po najmenej produktivneho (najmenej titulov).

```
var library = [{"autor": "Ezop", "titul": "Vlk a zajac"}, {"autor": "Tolkien", "titul": "Hobit"}, {"autor": "Ezop", "titul": "Medved a vlk"}];
```

```
mostProductive(library);
```

```
//priklad vystupu: [{"autor": "Ezop", "pocet": 2}, {"autor": "Tolkien", "pocet": 1}]
```

Na túto otázku môžete odpovedať ľubovoľne dlhým textom. Odpoveď bude vyhodnotená skúšajúcim ručne.

```
function mostProductive(library){
    let obj = {}
    let finalObj = {}
    library.forEach((item) => {if(!obj.hasOwnProperty(item.autor)){obj[item.autor] = 1}else{obj[item.autor] += 1}})
    var keys = Object.keys(obj)
    var values = Object.values(obj)
    keys.forEach((item,index) => {finalObj[item] = values[index]})
    return finalObj
}
```

Komentár:

malo vracať pole, bue objekt

- *Späť na zoznam napísaných testov*
- *Späť na zoznam testov k vypracovaniu*
- *Návrat do osobnej administratívy*

Číslo otázky	Otázka	Body
-----------------	--------	------