

TÖL306G Vefforritun



Prófdagur og tími: 08.12.2015 13:30-16:30

Prófstaður:

Aðalbygging - A222 (fjöldi: 8)

Lögberg - L-204 (fjöldi: 1)

VR-2 - V138 (fjöldi: 16)

VR-2 - V147 (fjöldi: 15)

VR-2 - V152 (fjöldi: 21)

VR-2 - V156 (fjöldi: 21)

VR-2 - V258 (fjöldi: 20)

VR-2 - V261 (fjöldi: 29)

HÁSKÓLI ÍSLANDS

Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild

Skriflegt próf

Skráðir til prófs: 131

Kennari:

Ólafur Sverrir Kjartansson (osk@hi.is / GSM: 6922349) Umsjónarkennari

Kennslumisseri: Haust 2015

Úrlausnir skulu merktar með nafni

Prófbók/svarblöð:

Prófbók óþörf

Hjálpargögn:

Engin leyfileg hjálpargögn

Önnur fyrirmæli:

Aðgangur að prófverkefni að loknu prófi:

Kennslusvið sendir eintak í prófasafn

Einkunnir skulu skráðar í Uglu eigi síðar en 29.12.2015.

ATHUGIÐ að einhverjar úrlausnir úr fjölmönnum prófum geta verið í þunnum umslögum sem auðvelt er að yfirsjá. GÓÐ VINNUREGLA er að byrja á því að opna öll umslög, telja úrlausnir og athuga hvort fjöldi stemmir við uppgefinn fjölda sem kvittað var fyrir.

Prentað: 04.12.15

Samkvæmt 60. grein Reglna fyrir Háskóla Íslands skulu einkunnir birtar í síðasta lagi tveimur vikum eftir hvert próf, nema eftir desemberpróf, þá eftir þrjár vikur. Einkunnir skulu skráðar í Uglu.



HÁSKÓLI ÍSLANDS

Verkfræði- og náttúruvísindasvið Vefforritun

TÖL306G

Lokapróf

Kennari: Ólafur Sverrir Kjartansson
Dagur: 8. des 2015
Klukkan: 13:30 – 16:30
Hjálpargögn: Engin hjálpargögn

Nafn: _____

Kennitala: _____

Prófið er 15 blaðsíður.

Gangi ykkur vel!

1. Krossasurningar / multiple choice, 30%

Fyrsti hluti Inniheldur 15 krossasurningar sem hver um sig gildir 2%. Veljið aðeins eitt svar fyrir hverja spurningu. Ekki er dregið niður fyrir vitlaus svör, en ef merkt er við fleira en eitt svar eru engin stig gefin fyrir þá spurningu.

The first part has 15 multiple choice questions, each gives 2%. Please choose only one answer per question, wrong answers do not incur a penalty but if a question has more than one answer, no points will be given for that question.

1.1 (2%) Til hvers er Gulp notað? / What is Gulp used for?

- ☐ a. Það er framework fyrir Node.js til að útbúa vefforrit / It's a framework for Node.js
- ☐ b. *Task runner* sem gerir handvirku hlutina sjálfvirka / Task runner that makes the manual things automatic
- ☐ c. Aðstoða okkur við að innbyrða *meira* af drykkjum *hraðar* / It helps us consume *more* drinks *faster*
- ☐ d. *Task runner* sem passar upp á að við skrifum góðan JavaScript kóða / Task runner that makes sure we write good JavaScript code

1.2 (2%) JSHint er dæmi um tól sem getur hjálpað okkur við að skrifa betri JavaScript kóða, hvernig virkar það? / JSHint is a tool that can help us write better JavaScript code, how does it work?

- ☐ a. Breytir týpukerfi JavaScript úr því að vera *dýnamísk* yfir í að vera *statísk* / It changes the type system of JavaScript from being *dynamic* into being *static*
- ☐ b. Bætir við virkni í ritli sem klárar fyrir okkur algengar skipanir (code completion) / Adds functionality to our text editor that allows us to do code completion
- ☐ c. Sendir kóðann okkar á jshint.com þar sem reyndir forritarar fara yfir kóða og senda tillögur til baka / Sends our code to jshint.com where veteran programmers go over our code and send back improvements
- ☐ d. Lætur okkur vita af algengum villum sem við gerum við smíði á JavaScript kóða / Alerts us to common errors we can do while writing JavaScript

1.3 (2%) Hver af eftirtöldum fullyrðingum um HTTP er rétt?

- ☐ a. HTTP krefst þess eingöngu að client geymi stöðu / HTTP requires that only the client stores state
- ☐ b. HTTP notar JavaScript fyrir virkni á framenda / HTTP uses JavaScript for frontend functionality
- ☐ c. HTTP byggir á að client sendi request og server sendi response / HTTP is based on clients sending requests and servers sending responses
- ☐ d. HTTP hefur aðgerðir sem eru flokkaðar, t.d. eru aðgerðir í 300 flokknum aðgerðir til að áframsenda / HTTP has methods that are categorized, e.g. methods in the 300 category are used for redirection

1.4 (2%) Þegar við setjum forrit sem tengjast gagnagrunni upp á mismunandi kerfum, geta tengistillingar verið mismunandi. Til að auðvelda okkur að breyta þeim er gott að: / When we deploy our programs that connect to a database on different platforms, the connection strings can be different. To help us change them it is good to:

- ☐ a. Safna öllum stillingum saman á einum stað í forriti, harðkóða þar og breyta fyrir hverja uppsetningu / Collect all the settings in one place in the program, hardcode them there and change for each deployment
- ☐ b. Geyma stillingar í sér falli og breyta fyrir hverja uppsetningu / Store settings in an isolated function and change for each deployment
- ☐ c. Harðkóða stillingar þar sem þær eru notaðar í forriti og breyta þeim fyrir hverja uppsetningu / Hardcode settings where they're used in the program and change them for each deployment
- ☐ d. Geyma upplýsingar í umhverfi og breyta fyrir hverja uppsetningu / Store settings in the environment and change for each deployment

1.5 (2%) <div> og eru element sem standa sér, um þau gildir að: / <div> and are elements that stand alone, about them we can assert that:

- ☐ a. Þau bera ekki merkingu og því ætti alls ekki að nota þau vegna merkingarfræði / They do not have any semantic meaning so they should absolutely not be used
- ☐ b. Þau bera merkingu en hún er óræð og því ætti alls ekki að nota þau / They have semantic meaning but it is not rational so they should absolutely not be used
- ☐ c. Þau bera ekki merkingu en ef svo ber við er í lagi að nota þau / They do not have any semantic meaning but if needed, they're okay to use
- ☐ d. Þau bera merkingu og er hún það almenn að við getum notað þau fyrir allt / They do have semantic meaning and it's so general that we can use them for anything

1.6 (2%) Hugtakið um „flóðið“ (cascade) í CSS skilgreinir / The concept of the cascade in CSS defines:

- ☐ a. Gildi fyrir hvern selector í CSS sem ræður því hversu mikilvægur hann er / Value for each selector in CSS that determines how important it is
- ☐ b. Röðina á skilgreiningum fyrir hvert element / The order of definitions for each element
- ☐ c. Gildi fyrir hverja reglu sem skilgreinir hvernig sjálfgefin gildi flæða yfir hana / Value for each rule that defines how default values *cascade* over it
- ☐ d. Röðina á skilgreiningum í skjalinu / The order of definitions in the document

1.7 (2%) Mörg framework bjóða upp á einhverskonar template mál, hvert að eftirfarandi er góð ástæða til að nota þau? / Many frameworks offer some kind of a templating language, which of the following is a good reason to use them?

- ☐ a. Tryggja okkur að SEO (search engine optimization) og aðgengi vefsins okkar sé gott / They guarantee that the SEO (search engine optimization) and accessibility of our website is good
- ☐ b. Sjá til þess að HTML vefsins okkar sé alltaf gilt / They guarantee that the HTML of our website is always valid
- ☐ c. Leyfa okkur að senda gögn á template og birta þau óháð virkni forrits / They allow us to send data to the template og display it, regardless of the functionality of the program
- ☐ d. Allt að ofan / All of the above

1.8 (2%) Einingapróf eru notuð til að: / Unit tests are used to:

- ☐ a. Auka öryggi okkar á breytingum sem við gerum á forriti / Increase our confidence while making changes to programs
- ☐ b. Skjala raunverulega virkni forrits / Document the real functionality of a program
- ☐ c. Gefa okkur hraða endurgjöf á virkni forrits meðan við erum að smíða það / Give us fast feedback on the functionality of a program while we write it
- ☐ d. Allt að ofan / All of the above

1.9 (2%) Þegar við útfærum notendaumsjón er að mörgu að huga, eitt af því sem við ættum að hafa sérstaklega í huga: / When we implement a user system, one thing we need to think specially about is:

- ☐ a. Nota traust og gott forritunarmál / Using a trusted and good programming language
- ☐ b. Dulkóða lykilorð / Encrypting passwords
- ☐ c. Nota trausta notendaumsjón frá þriðja aðila ef möguleiki er / Use trusted user systems from a third party
- ☐ d. Passa upp á að fela öll URL sem notandi má ekki komast inn á / Hide all URLs the user cannot see

1.10 (2%) Hvert af eftirtöldu er gilt URL? / Which of the following is a valid URL?

- ☐ a. http://example.org/foo?bar=baz
- ☐ b. example.org/foo?bar=baz
- ☐ c. http://example.org:foo/?bar=baz
- ☐ d. example.org:foo/?bar=baz

1.11 (2%) Hver af eftirfarandi staðhæfingum er ekki rétt um einingakerfi Node.js / Which of the following is not correct about the module system in Node.js

- ☐ a. Hægt er að sækja einingar sóttar með NPM með því að skrifa
`require('<nafn á einingu>');`
You can include a module from NPM by using
`require('<name of module>');`
- ☐ b. Hægt er að sækja einingu úr möppu sem inniheldur index.js með
`require('./<nafn á möppu>');`
You can include a module stored in a directory with index.js by doing
`require('./<name of directory>');`
- ☐ c. Hægt er að sækja forrit úr möppu sem inniheldur app.js með því að gera
`require('<nafn á möppu>');`
You can include a program from a directory that includes app.js with
`require('<name of directory>');`
- ☐ d. Hægt er að sækja JSON skrár með því að gera
`require('./<nafn á JSON skrá>.json');`
You can include a JSON file by doing
`require('./<name of JSON file>.json');`

1.12 (2%) Eitt af því sem Sass býður okkur ekki upp á er að: / One of the things Sass does not offer is:

- ☐ a. Útbúa *grindur* / Define *grids*
- ☐ b. Útbúa *mixin* fyrir aðgerðir / Define *mixin* for functions
- ☐ c. Skilgreina breytur / Define variables
- ☐ d. Skilgreina *nestaða* selectora / Defining *nested* selectors

1.13 (2%) Eftirfarandi CSS selector velur / The following CSS selector selects:

`main > section h2.important`

- ☐ a. Allar h2 fyrirsagnir með class important sem er afkomandi section sem er barn main / All h2 headings with class important that are a descendant of section that is the child of main
- ☐ b. Allar h2 fyrirsagnir með id important sem er afkomandi section sem er barn main / All h2 headings with id important that are a descendant of section that is the child of main
- ☐ c. Allar h2 fyrirsagnir með class important sem er afkomandi section sem er afkomandi main / All h2 headings with class important that are the descendant of section that is the descendant of main
- ☐ d. Allar h2 fyrirsagnir með id important sem er afkomandi section sem er afkomandi main / All h2 headings with id important that are the descendant of section that is the descendant of main

1.14 (2%) Hvaða verkefni leysa NPM og Bower? / What tasks do NPM and Bower solve?

- ☐ a. Skilgreina hvaða einingar frá þriðja aðila við notum / Define what modules from third parties we use
- ☐ b. Aðstoða okkur við að finna, sækja og uppfæra einingar frá þriðja aðila / Assist us in finding, fetching and updating modules from third parties
- ☐ c. Gerir okkur kleift að bæta okkar eigin einingum í almennar kóðageymslur þar sem aðrir geta sótt og nýtt / Allow us to publish our own modules to public repositories for others to use
- ☐ d. Allt að ofan / All of the above

1.15 (2%) Hver verður niðurstaðan ef eftirfarandi JavaScript kóði er keyrður? / What is the result of running the following JavaScript code?

```
function calc(f, g, x) {  
    return f(x) + g(x);  
}  
  
var x = 5;  
var square = function (x) { return x*x; }  
var result = calc(square, function (y) {  
    return x + y + 10;  
}, 2);  
  
console.log(result);
```

- ☐ a. 21
- ☐ b. 18
- ☐ c. NaN
- ☐ d. 42

2. Forritunarspurningar / Programming questions, 40%

2.1 (15%) HTML & CSS

Fyrir eftirfarandi HTML búi: / For the following HTML snippet:

```
<div class="card">
  <div id="heading">Lorem ipsum</div>
  <div class="content">Lorem ipsum dolor sit amet.</div>
  <figure id="image">
    
    <p>Hr. Foo</p>
  </figure>
</div>
```

er skilgreint eftirfarandi CSS: / the following CSS is defined:

```
.card {
  padding: 0;
  width: 200px;
  margin: 10px auto 0 auto;
  border: 1px solid #000;
  border-radius: 5px;
  padding: 10px 15px 10px 15px;
}
.card #heading { text-align: center; font-size: 20px;}
.card figure { margin: 0 auto; width: 50%; margin: 0 auto; }
```

Hvernig lítur .card út? Teiknið einfalda mynd. / How does .card look like? Draw a simple picture.

Hægt er að gagnrýna a.m.k. fimm atriði við bútana að ofan, gerðu grein fyrir þeim í stuttu máli: / You could criticize at least five things about the above snippets, elaborate in a few words:

1.

2.

3.

4.

5.

Ef að við vildum að .card myndi aðlagast sig að og virka í sem flestum aðstæðum – vera sem mest *skalanlegt* (*responsive*) – þyrftum við að: / If we wanted .card to adapt and work in as many scenarios as possible – be responsive – we'd need to:

Hver væri stærðin á .card skv. box módelinu (width og height) ef það væri sett inn í element sem hefði breidd 500px? / What is the size of .card according to the box model (width & height) if it's contained within an element that has a width of 500px?

2.2 (10%) JavaScript

Útfærið virkni sem finnur alla takka (<button>) á síðu og festir við þá fallið `buttonHandler` þegar smellið er á þá (*click*). Aðeins skal nota JavaScript. Verið nákvæm og uppfyllið málfræðireglur.

Implement functionality that finds all buttons (<button>) on a page and attaches the function `buttonHandler` to them when they are clicked. Only use JavaScript. Be precise and adhere to the syntax of JavaScript.

```
A.addEventListener('DOMContentLoaded', B () {  
    var buttons = document.C(D);  
    if (!E) {  
        return;  
    }  
  
    F (var i = 0; i<G.length; i++) {  
        buttons[i].H(I, J);  
    }  
})
```

A =

B =

C =

D =

E =

F =

G =

H =

I =

J =

2.3 (15%) Node.js & Express

Stillið upp virkni sem tekur við gögnum úr formi, staðfestir þau og ef þau eru í lagi, vistar þau í grunni. Gefið er: / Setup functionality that receives data from a form, validates them and if they're okay, saves them in a database. Given is:

`validate(data, callback)` - tekur við fylki af gögnum og staðfestir. Callback inniheldur `true` ef gögn eru í lagi, `false` annars
`validate(data, callback)` - takes an array of data and validates. Callback contains `true` if data is okay, `false` otherwise

`save(data, callback)` - tekur við gögnum og vistar í grunni. Callback inniheldur `true` ef gögn eru vistuð, `false` annars
`save(data, callback)` - takes an array of data and saves them in database. Callback contains `true` if data is okay, `false` otherwise

Callback eru á forminu `function (err, data) { }` þar sem `err` er `null` ef engin villa kom upp, annars villan sjálf
Callbacks are in the form `function (err, data) { }` where `err` is `null` if no error occurred, otherwise the error itself.

Middleware `isLoggedIn` er til sem sér um að passa upp á að notandi sé innskráður og megi sjá form og vista gögn. `dataForm` middleware birtir form og `saveForm` middleware vistar gögn.
Middleware `isLoggedIn` exists to make sure the user is logged in and can see form and save data. `dataForm` middleware shows the form and `saveForm` middleware saves data.

`res.render('form', { data: data, posted: <bool>, valid: <bool> });`
sýnir form rétt m.v. gögn og hvort það hafi verið send gögn og hvort þau séu gild. / this shows the form correctly according to the data and if data has been posted and if they're valid.

Ef taka á við get og post beiðnum frá innskráðum notendum, hvernig stillum við upp *router*? / If we want to receive get and post requests from logged in users, how should we structure our routes?

`router.`

`router.`

Útfærið eftirfarandi m.v. gefnar forsendur, ekki þarf að sýna nákvæman kóða. Hugið sérstaklega að *callback flæði* / Implement the following according to the given assumptions, the code does not need to be perfect. Focus on the *callback flow*.

```
function saveForm(req, res, next) {  
  var data = req.body;
```

3. Ritgerðarspurningar/ Essay questions, 30%

Þriðji hluti inniheldur fjórar spurningar en aðeins þarf að svara þrem sem hver um sig gildir 10%. Ef öllum spurningum er svarað gilda þrjú bestu svörin. Vandlð uppbyggingu og frágang. Stutt og hnitmiðuð svör.

The third part has four questions but you only need to answer three, each of which gives 10%. If you answer all the questions, the three best answers count. Clear structure and delivery. Short and concise answers.

3.1 (10%) *Stigvaxandi aukning* (progressive enhancement) og *tignarlegt niðurbrot* (graceful degradation) eru hugtök sem skilgreina leiðir til að eiga við stuðning vafra. Hvað þýða þau og hver er helsti munurinn? Gefið dæmi. / Progressive enhancement and graceful degradation are concepts that define ways to deal with browser support. What do they mean and what is their greatest difference? Give an example.

3.2 (10%) Node.js er *atburðadrifið* og er *event loop* notuð. Hvernig virkar hún og hvaða áhrif hefur það á það hvernig við skrifum forrit? / Node.js is *event driven* and uses an *event loop*. How does that work and what effects does it have on how we write programs?

3.3 (10%) Hvað er XSS (Cross-Site Scripting)? Hvernig virka *persistent* og *reflected* XSS árásir? Hvernig getum við varið okkur fyrir XSS? Gefið dæmi. / What is XSS (Cross-Site Scripting)? How do persistent and reflected XSS attacks work? How can we defend against XSS? Give an example.

3.4 (10%) Þú ásamt nokkrum öðrum aðilum í hóp fáið það verkefni að útfæra vefkerfi. Kerfið á að bjóða upp á að svara spurningum út frá vel skilgreindum lista þar sem hvert svar leiðir þig að ákveðinni útkomu (t.d. eins og persónuleikapróf, tveir aðilar sem svara alveg eins fá nákvæmlega sömu útkomu.)

Hvernig myndir þú:

- a. Setja upp verkefnið þ.a. auðvelt sé fyrir hópinn að vinna það?**
- b. Haga útfærslu á framenda og bakenda? Leitast er eftir almennri lýsingu sem sýnir heildarskilning á efni**

You along with others in a team get the project of building a web app. The system should allow people to answer questions from a well defined list, where every answer leads you towards a certain outcome (e.g. a personality test, where two parties who give the same answers get exactly the same result).

How would you:

- a. Structure the project so that the team can easily work together?**
- b. Implement the frontend and backend? Answer should be general and show an overall understanding of the material.**