

# TÖL306G Vefforritun



**Prófdagur og tími:** 08.12.2016 13:30-16:30

**Prófstaður:**

Aðalbygging - A222 (fjöldi: 6)  
Háskólatorg - HT302 (fjöldi: 3)  
VR-2 - V152 (fjöldi: 36)  
VR-2 - V156 (fjöldi: 22)  
VR-2 - V258 (fjöldi: 22)  
VR-2 - V261 (fjöldi: 29)

HÁSKÓLI ÍSLANDS

**Iðnaðarverkfræði-, vélaverkfræði- og tölvunarfræðideild**

**Skriflegt próf**

**Skráðir til prófs: 118**

**Kennari:**

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

**Kennslumisseri:** Haust 2016

**Úrlausnir skulu merktar með nafni**

**Prófbók/svarblöð:**

Prófbók óþörf

**Hjálpargögn:**

engin hjálpargögn

**Önnur fyrirbætur:**

**Aðgangur að prófverkefni að loknu prófi:**

Kennslusvið sendir eintak í prófasafn

**Einkunnir skulu skráðar í Uglu eigi síðar en 29.12.2016.**

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: 02.12.16

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



# HÁSKÓLI ÍSLANDS

## Verkfræði- og náttúruvísindasvið Vefforritun

**TÖL306G**

### Lokapróf

Kennari: Ólafur Sverrir Kjartansson  
Dagur: 8. desember 2016  
Klukkan: 13:30 – 16:30  
Hjálpargögn: Engin hjálpargögn

Nafn: \_\_\_\_\_

Kennitala: \_\_\_\_\_

Prófið er 18 blaðsíður.

Gangi ykkur vel!

## 1. Krossaspurningar, 30%

Fyrsti hluti inniheldur 15 krossaspurningar sem hver um sig gildir 2%.

Aðeins er eitt rétt svar við hverri spurningu, ef fleiri en eitt svar eru *tæknilega* rétt, er rétt svar *það réttasta* m.v. námsefni og umfjöllun. Ekki er dregið niður fyrir vitlaus svör, en ef merkt er við fleira en eitt svar eru engin stig gefin fyrir þá spurningu.

### 1.1 (2%) Ef við ætlum að tengjast við `example.org`, væri best að nota? / If we are linking to `example.org`, it would be best to use

- ☐ a. `<a href="http://example.org">example.org</a>`
- ☐ b. `<form method="post" action="http://example.org/">`  
`<button>example.org</button>`  
`</form>`
- ☐ c. `<form method="get" action="http://example.org/">`  
`<button>example.org</button>`  
`</form>`
- ☐ d. Allar aðferðir eru jafn góðar / All methods are equally good

### 1.2 (2%) *Code coverage* er hugtak sem hjálpar okkur að skrifa unit test því að / *code coverage* is a concept that helps us to write unit tests by

- ☐ a. Það keyrir öll prófin okkar og lætur vita af útkomum úr þeim / It runs all our tests and lets us know their results
- ☐ b. Við fáum að vita hve stórt hlutfall kóðans okkar er prófað og komumst þá hjá því að skrifa mörg próf fyrir sömu föllin, greinar eða segðir / We get to know the percentage of our code is under test and can get by without writing tests for the same functions, branches or statements
- ☐ c. Það keyrir valin prófin og lætur vita af útkomum úr þeim / It runs selected tests and lets us know their results
- ☐ d. Við fáum að vita hve margar línur í kóðanum okkar eru prófaðar og komumst þá hjá því að skrifa mörg próf fyrir sömu föllin, skipanir, greinar eða segðir / We get to know how many lines of our code are under test and can get by without writing tests for the same functions, branches and statements

**1.3 (2%) Arrow functions í ECMAScript 2015 eru öðruvísi en föll, hvernig? / Arrow functions in ECMAScript 2015 are different from functions, how?**

- ☐ a. Endurskilgreina ekki `this` og ekki arguments  
Do not bind `this` nor arguments
- ☐ b. Endurskilgreina `this` en ekki arguments  
Do bind `this` but not arguments
- ☐ c. Endurskilgreina ekki `this` aðeins arguments  
Do not bind `this` only arguments
- ☐ d. Endurskilgreina `this` og arguments  
Do bind `this` and arguments

**1.4 (2%) Ef við höfum eftirfarandi CSS, hver er breidd X hæð boxsins í pixlum skv. box módelinu? / For the following CSS, what are the width X height of the box in pixels according to the box model?**

```
.item {  
  width: 500px;  
  height: 300px;  
  margin: 20px;  
  margin-right: 0;  
  border: 1px solid #000;  
  padding: 10px;  
  padding-bottom: 20px;  
}
```

- ☐ a. 542x372
- ☐ b. 552x342
- ☐ c. 352x542
- ☐ d. 342x552

**1.5 (2%) Þegar við athugum aðgengi vefja getum við nýtt okkur: / When we are checking for web accessibility, we can use:**

- ☐ a. Staðla sem innihalda leiðbeiningar og tilmæli um aðgengi / Standards containing guidelines and recommendations on accessibility
- ☐ b. Að hlusta á vef með skjálesara / Listening to the website with a screen reader
- ☐ c. Notað eingöngu lyklaborð við að skoða vef / Using only the keyboard to navigate the website
- ☐ d. Allt að ofan / All of the above

**1.6 (2%) Hvað er V8? / What is V8?**

- ☐ a. JavaScript vél sem Chrome notar / JavaScript engine that Chrome uses
- ☐ b. HTML þáttari sem Chrome notar / HTML parser that Chrome uses
- ☐ c. CSS túlkur sem Chrome notar / CSS interpreter that Chrome uses
- ☐ d. Ekkert að ofan / None of the above

**1.7 (2%) Hvaða grundvallar munur, ef einhver, er á HTML 4.01 og XHTML? What fundamental difference is there between HTML 4.01 and XHTML?**

- ☐ a. HTML 4.01 er byggt á XML en XHTML á SGML / HTML 4.01 is based on XML but XHTML on SGML
- ☐ b. XHTML er byggt á HTML5 en HTML 4.01 á SGML / XHTML is based on HTML5 but HTML 4.01 on SGML
- ☐ c. XHTML er byggt á XML en HTML 4.01 á SGML / XHTML is based on XML but HTML 4.01 on SGML
- ☐ d. Engin grundvallar munur / No fundamental difference

**1.8 (2%) Hvert af eftirtöldum gildum eru *false* í JavaScript? / Which of the following values are *false* in JavaScript?**

- ☐ a. "0"
- ☐ b. []
- ☐ c. NaN
- ☐ d. 1

**1.9 (2%) Hugtökin *framendi*, *bakendi* og *HTTP* tengjast á hvaða hátt? / The concepts *frontend*, *backend* and *HTTP* are related in what way?**

- ☐ a. *Framendi* er birtur notanda, gögnin sem eru birt eru sótt með *HTTP* á *bakenda* / *Frontend* is shown to the user, the data is retrieved with *HTTP* on the *backend*
- ☐ b. *Bakendi* er birtur af notanda, gögnin sem eru birt eru sótt með *HTTP* á *framenda* / *Backend* is shown by the user, the data is retrieved with *HTTP* on a *frontend*
- ☐ c. *Framendi* er birtur af vefþjón, gögnin sem eru birt eru sótt með *HTTP* af *bakenda* / *Frontend* is shown by the web server, the data is retrieved with *HTTP* from the *backend*
- ☐ d. *Bakendi* útbýr gögn sem send eru með TCP/IP yfir á *framenda*, *HTTP* er tækni sem við notum til að útbúa form / *Backend* creates the data that is sent with TCP/IP to the *frontend*, *HTTP* is a technology used to create forms

**1.10 (2%) Hvernig tengjast *querySelector*, *querySelectorAll* og *jQuery*? / How do *querySelector*, *querySelectorAll* and *jQuery* relate?**

- ☐ a. Nota öll DOM tré þegar leitað er að elementum í vefsíðum / All use the DOM tree when searching for elements in websites
- ☐ b. *querySelector* og *querySelectorAll* eru föll í *jQuery* forritasafninu / *querySelector* and *querySelectorAll* are methods in the *jQuery* library
- ☐ c. Nota öll *selector syntax* til að finna element í DOM trénu / All use *selector syntax* to find elements in the DOM tree
- ☐ d. Tengjast ekkert / Not related

**1.11 (2%) Þegar við útbúum *XMLHttpRequest* og framkvæmum *Ajax* beiðni: / When we create an *XMLHttpRequest* and perform an *Ajax* request:**

- ☐ a. Fáum við alltaf JSON til baka og verðum að breyta því í JavaScript hluti til að vinna með / We always get JSON as a response and must change it into a JavaScript object to work with
- ☐ b. Verður vefþjónninn sem við köllum í að skila Cross-origin resource sharing (CORS) hausum / The web server we call must have a Cross-origin resource sharing (CORS) header
- ☐ c. Verðum við að huga að því að þegar við sækjum efni yfir net, getur það verið hægt / We must note that when we retrieve resources over a network, it can be slow
- ☐ d. Fáum við aldrei JSON til baka og verðum því að þátta svar í JavaScript hlut / We never get JSON as a response and must parse it into a JavaScript object

### 1.12 (2%) Hvað er flexbox? / What is flexbox?

- ☐ a. Leið til að útbúa sveigjanlegt útlit sem kemur í staðinn fyrir `position` og leyfir okkur að skilgreina útlitið auðveldlega á tveimur ásum / Way to create flexible layouts that replaces `position` and allows us to define layout easily on two axes
- ☐ b. Leið til að útbúa sveigjanlegt útlit sem virkar með `position` og leyfir okkur að skilgreina útlitið auðveldlega á tveimur ásum / Way to create flexible layouts that works with `position` and allows us to define layout easily on two axes
- ☐ c. Leið til að útbúa sveigjanlegt útlit sem kemur í staðinn fyrir `position` og leyfir okkur að skilgreina útlit auðveldlega á einum ás / Way to create flexible layouts that replaces `position` and allows us to define layout easily on one axis
- ☐ d. Leið til að útbúa fast útlit sem virkar með `position` og leyfir okkur að skilgreina útlit auðveldlega á tveimur ásum / Way to create fixed layouts that works with `position` and allows us to define layout easily on two axes

### 1.13 (2%) Þegar við nýtum nýja virkni ættum við: / When we use new features, we should:

- ☐ a. Alltaf að setja upp *polyfill* fyrir virknina / Always setup a *polyfill* for the feature
- ☐ b. Fletta virkninni upp á *caniuse.com* og ef hún er með yfir 90% stuðning, getum við notað hana / Check the feature on *caniuse.com* and if it has over 90% support, use it
- ☐ c. Meta virkni út frá verkefninu og þeim vöfrum sem við ætlum að styðja / Evaluate the feature based on the project and the browsers supported
- ☐ d. Allt að ofan / All above

### 1.14 (2%) Til að láta vefina okkar verða skemmtilegri getum við notað breytingar og hreyfingar, sér í lagi ættum við að / To make our websites more fun, we can use transtions and animations, in particular, we should:

- ☐ a. Forðast að nota `transform` í CSS, þar sem það getur valdið hægagangi / Stay clear of `transform` in CSS, since it can cause jank and lag
- ☐ b. Láta allt hreyfast, alltaf, allstaðar / Have everything move, everywhere, all the time
- ☐ c. Halda okkur við hluti sem er ódýrt og hagstætt að hreyfa og breyta / Stick to transitioning and transforming things that are inexpensive & reasonable
- ☐ d. Nýta 3D getu CPU með því að færa núll transformið á nýjan layer / Use the 3D processing of the CPU by moving the null transform to a new layer

**1.15 (2%) Hvað skrifast út þegar eftirfarandi JavaScript kóði er keyrður? / What is logged when the following JavaScript code is run?**

```
const values = [1, 2, 3, 4, 5, 6, 7, 8, 9].map((value) => {  
  return value * value;  
});  
const results = [];  
for (let i = values.length - 1; i>=0; i--) {  
  const value = values[i];  
  results.push(value);  
}  
console.log(results.join(', '));
```

- ☐ a. 81, 64, 49, 36, 25, 16, 9, 4, 1
- ☐ b. 1, 2, 3, 4, 5, 6, 7, 8, 9
- ☐ c. 1, 4, 9, 16, 25, 36, 49, 64, 81
- ☐ d. 9, 8, 7, 6, 5, 4, 3, 2, 1



## 2. Forritunarspurningar, 40%

### 2.1 (10%) JavaScript

Finna má ýmislegt að eftirfarandi JavaScript kóði í ES2015, nefndu a.m.k. fimm atriði og útskýrðu hvað mætti betur fara:

There is a plethora of errors in the following JavaScript code in ES2015, name at least five of them and describe why and how to fix them

```
document.addEventListener('DOMContentLoaded', function() {  
  let buttons = document.querySelectorAll('button');  
  
  for(var foo = 0; foo < buttons.length; foo++) {  
    button = buttons[foo];  
    button.addEventListener('onclick', (e) => {  
      button.setAttribute('data-clicked', true);  
    });  
  }  
});
```

1.

2.

3.

4.

5.

## 2.2 (10%) CSS

Verið er að útbúa vef sem mun birta upplýsingar í *kortum* (e. card) sem raðast hlið við hlið einsog pláss leyfir. Skrifðu CSS sem uppfyllir eftirfarandi:

- card skal hafa
  - hvítan bakgrunn (#ffffff)
  - 10px margin til hægri og 10px margin fyrir neðan
  - 20px padding á alla kanta
  - 1px fastann svartan ramma (border) (1px solid #000000)
  - takmörkun á hámarksbreidd, það má aldrei verða breiðara en 400px
  - takmörkun á lágmarksbreidd, það má aldrei verða minna á breiddina en 200px
- Almennt skal card taka 1/3 af því lárétta plássi sem í boði er eru
- Undir 800px skal card taka 1/2 af því lárétta plássi sem er í boði er
- Undir 500px skal card taka allt það lárétta pláss sem er í boði er og ekki hafa neitt margin

You are setting up a web which will display info in *cards* that will stack side by side as room allows. Write CSS that satisfies the following:

- card shall have
  - white background (#ffffff)
  - 10px margin to the right and 10px margin at the bottom
  - 20px padding on all sides
  - 1px solid black border (1px solid #000000)
  - limitation of the maximum width, it can never be wider than 400px
  - limitation of the minimum width, it can never be less than 200px
- In general, card shall take up 1/3 of the horizontal space available
- Under 800px card shall take up 1/2 of the horizontal space available
- Under 500px card shall take up all available horizontal space and not have any margin

```
/* width, max-width, min-width, padding, margin, border, background-color*/  
.card {  
  box-sizing: border-box;
```

```
}
```

```
@media only screen and (max-width: 800px) {  
  .card {
```

```
  }
```

```
}
```

```
@media only screen and (max-width: 500px) {  
  .card {
```

```
  }
```

```
}
```

## 2.3 (10%) HTML

Þú ert að setja upp fréttasíðu þar sem eftirfarandi þarf að koma fram fyrir frétt:

- Titill
- Meginmál fréttar
- Mynd með frétt, ásamt myndatexta
- Listi af lýsigögnum fyrir frétt
  - Fréttamaður
  - Dagsetning
  - Flokkar sem frétt á við (t.d. „innlent, stjórnmál, kosningar“)

Skrifaðu upp HTML beinagrind fyrir frétt þar sem öll atriðin koma fram. Notaðu merkingarfræðileg element og láttu inndrátt vera augljósan. Ekki þarf að hugsa um efni í meginmáli, nóg að skilgreina element sem mun innihalda efnið.

You are setting up a news site where the following must be included for the news:

- Title
- News body
- Image with news, along with a caption
- List of metadata for the news
  - Reporter
  - Date
  - Categories (e.g. "domestic, politics, election")

```
<!doctype html>
<html lang="is">
  <head>
    <meta charset="utf-8">
    <title>Er þetta frétt?</title>
  </head>
  <body>
```

```
</body>
</html>
```

## 2.4 (10%) node.js

Þú færð það verkefni að bæta við síðu á vef sem keyrir í Express. Síðan á að svara á slóðinni `/form` og innihalda form sem er gefið í skránni `form.pug`. Þegar búið er að fylla út í formið er því *postað* á `/process-form` sem vistar gögnin með því að kalla í þakkann `save-form-data` og er uppsettur í verkefninu. Fallið sem þakkinn gefur skilar *promise*, ef það er uppfyllt er gögnum skilað sem senda skal í template `result.pug`, ef villa kemur upp skal birta `error.pug` með villu. Fyllið inn í á þeim stöðum þar sem undirstrik eru og þar sem athugasemdir biðja um það.

- Promise hafa `then` og `catch`
- Til að birta template notum við `res.render(templateName, templateData)`

You get the task of adding a page to a website running in Express. The page should be accessible on the URI `/form` and contain a form, given in `form.pug`. When the form has been filled it is posted to `/process-form` that saves the data by calling the `save-form-data` that is setup in the project. The module returns a promise, if it is resolved you should render the template `result.pug` along with the data. If an error occurs, render `error.pug` with the error.

Fill in the underlined areas and where comments ask for it.

- Promise have `then` and `catch`
- To render a template we use `res.render(templateName, templateData)`

```

const save = require(_____);
const app = express();

//... setup of app

app._____(_____, (req, res, next) => {
  // show form

});

app._____(_____, (req, res, next) => {
  const data = req.body;

  // handle promise and show data or error
  save(data)

});

```

### 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.

Vandið uppbyggingu og frágang. Stutt og hnitmiðuð svör.

#### 3.1 (10%) Hvaða merkingu leggur þú í orð Sir Tim Berners-Lee hér að neðan? Hvernig tengjast þau því námsefni sem farið hefur verið yfir?

How do you interpret Sir Tim Berners-Lee's words below? How do they relate to the material we've covered?

*The primary design principle underlying the Web's usefulness and growth is universality. [...] And it should be accessible from any kind of hardware that can connect to the Internet: stationary or mobile, small screen or large.*

— Tim Berners-Lee: Long Live the Web



**3.2 (10%) *eslint*, *stylelint*, *babel* og *sass* eru tól sem við höfum skoðað og notað.**

**Hvaða verkefni leysa þau? Hvernig getum við sett upp verkefni þannig að þau hjálpi okkur sem mest? Hafa þau ókosti?**

*eslint*, *stylelint*, *babel* and *sass* are tools we've looked into and used.

What tasks do they solve? How can we setup our projects so they help us the most? Do they have any disadvantages?

**3.3 (10%) Injection árásir eru flokkaðar sem helsta öryggishætta vefforrita skv. OWASP.**

**Í hverju felast þessar árásir? Hvað er ráðist á? Hvernig komum við í veg fyrir þær?**

Injection attacks are classified as the main security threats to web apps according to OWASP. What are these attacks? What is attacked? How can we prevent them?

**3.4 (10%) Eftir glimrandi frammistöðu í vefforritun, færð þú ásamt öðrum samnemendum þínum það verkefni að útbúa kennsluvef fyrir námskeiðið. Það sem vefurinn þyrfti að útfæra væri:**

- Aðgengi að fyrirlestrum ásamt upptökum af þeim á einni síðu
- Gagnvirk dæmi þar sem hægt er að eiga við kóða til að skilja hvernig hann virkar

**Hvernig myndir þú setja verkefnið m.v. að bæði væri skrifaður framenda og bakenda kóði og að margir myndu vinna í því? Leitast er eftir almenntri lýsingu sem sýnir heildar skilning á efni.**

After a stellar performance in web development, you along with your classmates are tasked with preparing a website for the course. The website should implement:

- Access to lectures along with recordings on one page
- Interactive examples where you can change the code to understand how it works

How would you structure the project if both backend and frontend code would be written and a team of people would be working on it? General descriptions showing the overall understanding of the course.