

CHALMERS

EXAMINATION / TENTAMEN

Course code/kurskod	Course name/kursnamn		
DIT 341	Web and Mobile Development		
Anonymous code Anonym kod		Examination date Tentamensdatum	Number of pages Antal blad
761		2019 04 24	12

* I confirm that I've no mobile or other similar electronic equipment available during the examination.
 Jag intygar att jag inte har mobiltelefon eller annan liknande elektronisk utrustning tillgänglig under
 eximinationen.

Solved task Behandlade uppgifter No/nr	Points per task Poäng på uppgiften	Observe: Areas with bold contour are to completed by the teacher. Anmärkning: Rutor inom bred kontur ifylls av lärare.
1	X 8	
2	X 11.5	
3	X 4	
4	X 5	
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
Bonus poäng		
Total examination points Summa poäng på tentamen	18.5	

1. Dom → Document object model

The dom is the building blocks of HTML which are used to build and make layout of a webpage.

Example

```

  {<div class="demo"> // container
   <p>Text inside div</p> // paragraph.
  </div>
  
```

That's not really the DOM

Without styling the example above "Text inside div" will be displayed on the website.

2

2. Http requests

https://www.coolwebsite.com/page1/?query1=test&query2=8

protocol

domain

path

parameters

In the above example two parameters are sent, query1 and query2.

I expect to get an HTTP 200 response along with some data.

This is a request



3. Rest constraints

statelessness

The server should not store any state from the caller.
 Each request should be independant.
 If data is needed it should be sent with each request.

cachelessness

The server should return if it is cacheless or not.

2

~~separation of concern ?~~

One way entry point

The caller should only know one entry point!

for example ... api/users

~~(crud?)~~

4. Lists in HTML

To create a list in HTML you could use
for example a list or a table amongst others

List

~~~~ // declare the list

~~<item>~~ | ~~</item>~~ // declare an item

~~<item> 2</item>~~

~~...~~

~~~~

~~~~

~~..~~

~~~~

~~~~

~~..~~

~~~~

Table

~~<table>~~ // creates a table

~~<tr>~~ // creates a new row

~~</tr>~~ item 1

~~<tr>~~

~~item 2~~

~~</tr>~~

~~<table>~~

~~O~~

Above creates a table without any headers or columns

5. AJAX ASynchronous Javascript and XML //could also be Json.
Ajax is used to make async calls to collect more data.
with the help of Ajax we can change the content in the dom, for example in a `<div>` without the need of reloading the website. ✓

2

6. A process is usually started when we start a program.

Threads run "inside" the process to perform different tasks. ✓

For example, when starting a browser the computer starts a process. If you would open several different tabs within the process you would start a new thread for each tab. Threads, can run concurrently to each other.

2

1. <!doctype html> // ~~HTML~~ 5
 <head>

<title> MyPage </title> ✓
 </head>

<body>
 <h1 class="class1"> PageTitle </h1> ✓
 <p id="para1">Text1</p> ✓

</body>

</html>

2. CSS Cascade

1. Browser default

2. styling in header.

→ can't remember exactly but something like this

3. based on Attributes

pageTitle will be in blue. ✓

The h1 tag has the class "class1", which in the CSS is written .class. This has priority over browser default and header styling. ✓

Text1 will be displayed in green. The p tag has an id, which in CSS is written #para1
 ↓id ✓

perfect

3. The output will be "True" ✓

JavaScript does not compare types when using the `==` operator, and in this case the `Value` will be the same. However if we were to be `if(x==y)` the output would have been `false`.

perfect

1

4. Callbacks

Advantages

Since javascript is only able to run on thread at the time, callbacks are good because it can help to webpage to run fluently as the program is able to continue when making for example a longrunning api call. The program will continue and return to the callback function when the api call is finished. ✓

Disadvantages

The disadvantage of callbacks is that the code can be really messy and hard to understand if there are a lot of "nested" callbacks.

2

5. The output will most likely be

4 ✓

6 ✓

5

1

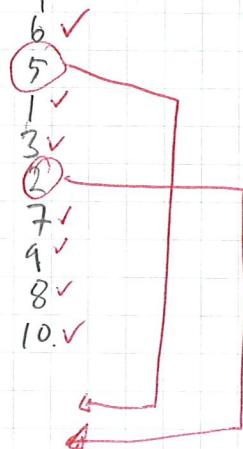
35

27

9 ✓

8 ✓

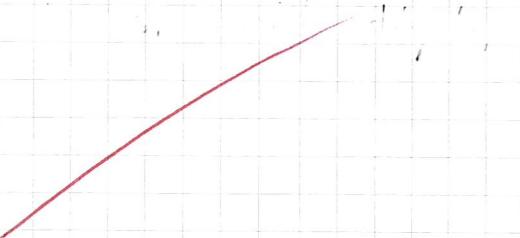
10. ✓



2

JavaScript is run synchronously, however when it comes to a setTimeout it will "jump" over to the next sequence of the code until the timeout has passed. When the timeout is done it will run the inside the block.

6



0

7. The application will display in the div "app"
 $n_1 + n^2 * 2$ Where n_2 is the number (✓)
 of clicks.

1st output = 5

~~Q~~

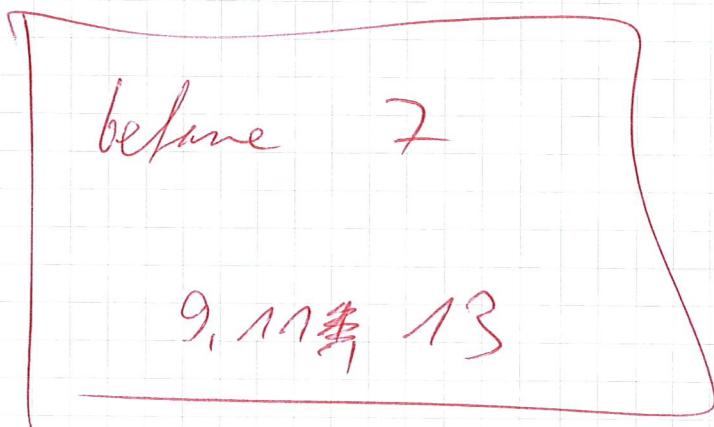
2nd output = 7

~~X~~

3rd output = 9.

~~X~~

0.5



8. A program is executed as a process.

✓

A thread inside a process could be used to for example
making an api call or sending data to google.

✓

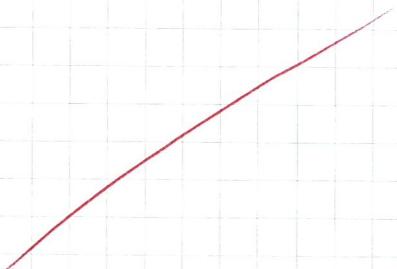
1

9. The UI thread is the main thread we start
when we start an mobile application. Blocking
this leads to the ui stops responding. Thus the app will
look/feel that it is halted for user.

✓

10.

A



1. I expect a 404 not found

↳ first result of expression

1

2. They are both used to update the object.

I would use put if I where to change ~~almost~~ ① the entire object and patch if I would update only one field.

2

3. I expect to get a maximum amount of hits,
on the "keyword"

1 page is likely how pages i would like and {
1 limit is the total amount of hits i expect. } quick
→ paginat₁₀

4. I expect the the api ~~/will~~ create two diffunt
posts and recive the id of each post in return.

Ø

Ø

Ø Sets created once ! Ø

1. CI

The idea of CI is to make several small releases instead of one big one. One advantage to this is time to market. Another pro is for example testability and if something goes wrong reverting to previous version is a lot easier and fixing small problems is easier than to fix a big release.

~~What does it mean?~~

~~CI~~

✓

In Practice

A developer implements a feature. When finished with "first" testing and when the a code review has been conducted, the developer then merges his branch to example the develop branch. If on a build server like Jenkins will take over and run unit tests hopefully. If the unit test are approved the buildserver will publish the code to a server. It could be either directly to the production server but usually to a test server where for example integration testing could be done before publishing to the production server. And then repeat for next feature/bug fix.

2. No, tests are always useful to conduct, no matter the type of program. It is great to perform for example monkey test on a website, ideally by someone other than the developer. It is a great opportunity to see if the webpage works as expected and to the user understands and perceives the functionality the same as the developer.

Smoke tests are also good for websites and also for example APIs. It is good to test parts of the code to see that they are functioning as expected.

Comparison of techniques 7