Assignment 6: Forms, XML and Javascript

No one is above the law, particularly not those who enforce it. This is especially true when it comes to tax law. In this exercise you are going to create a website that will enable superheroes to submit their tax returns.

Due to missing form validation support for certain fields we will only grade your assignment using Chrome (version 60).

Part 1: Create a Form (15%)

Please note that this part of the exercise has to be completed **without Javascript**.

Prior to submitting tax returns, a superhero must provide some personalia along with information regarding his or her superpower. Modify the empty form in `index.html` to include fields with appropriate input types and labels given the following questions.

Label	Name	Description	Example
Real name	name	A string of alphabetical characters	Peter Parker
Gender	gender	Either "male" or "female"	Male
E-mail	email	An e-mail address	spinning@web.com
Birthdate	birthdate	DD.MM.YYYY prior to 1988	28.12.1984
Hero name	hero	A string of characters	Spider-Man
Do you wear spandex?	spandex	A boolean	True
Strength	strength	Integer between 1 and 10 (inclusive)	10
		Integer between 1 and	

| Speed | speed | 10 (inclusive) | 1 |
| Intelligence | intelligence | Integer between 1 and 10 (inclusive) | 8 |

Note: The format of "Birthdate" can vary depending on your operating system. The format "DD/MM/YYYY" is also accepted.

The "Description" and "Example" columns are only intended for you as the developer, they do not need to be present in the form.

Finally, a button saying "Submit tax form" must be located at the end of the form.

Part 2: Style it (5%)

Style the form and its elements using CSS to make it more user friendly and better looking.

Part 3: Form validation (10%)

Please note that this part of the exercise has to be completed **without Javascript**.

The Norwegian Tax Administration (Skatteetaten) requires that the superhero fills in all the fields in the form. This means that the superhero should not be able to submit the form prior to answering all the questions.

Answers that violates the constraints expressed in the "Description" column in Part 1 should render the form unsubmittable. For example, it should not be possible to submit the form if the birthdate is set to `1989-01-01`.

Part 4: Add more fields to the form (20%)

To enable the Norwegian Tax Administration to receive the actual tax returns you must append three more required fields to the form.

Label	Name	Description	Example
Income	income	An integer greater or equal to 0	123 000
Wealth	wealth	An integer greater or equal to 0	21 000

| Tax | tax | Disabled: Not possible to enter value | 44 790 |

`income` and `wealth` are required fields. Also, ensure that the values can not violate the constraints expressed in the "Description" column.

Note that a disabled form element is an element that can not have its value changed by the user.
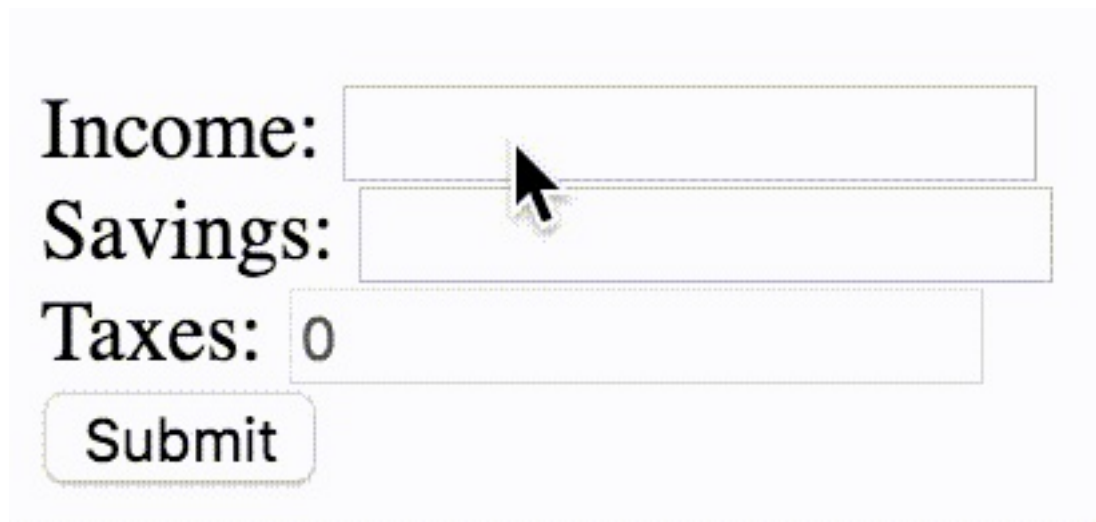
Part 5: Binding elements together (20%)

Now that all the form elements are in place it is time to calculate how much the superhero must pay in taxes! The following formula calculates how much tax a superhero has to pay given his or her income and wealth.

```
tax(income, wealth) = (0.35 * income) + (0.25 * wealth)
```

When a superhero changes the income or wealth form fields, the value of the disabled tax field should be updated to the result of the above formula.

This exercise needs to be solved *with* Javascript.

Part 6: XML structure (20%)

This part is independent of all previous parts.

The Norwegian Tax Administration want to store their data in XML documents. Your job is to

show them how they should structure the XML documents. Convert the following tax forms (see table below) into one well formed XML document. You do not have to worry about XML Schema Definitions (XSD).

Store the file in the root directory of your deliverable and name it `taxes.xml`.

Real name	Gender	E-Mail	Birthyear	Hero name
Nick Nedd	Male	nick@tick.com	1976	The Tick
Bruce Wayne	Male	bruce@wayneenterprises.com	1974	Batman
Selina Kyle	Female	selina@whiskas.com	1977	Catwoman

Note: The table is also available in the attachments folder.

Questions (10%)

1. Why does the following expression `0.1 + 0.2 === 0.3` evaluate to `false` in Javascript?
2. What does the `method` attribute on a `form` element decide?
3. Explain why the form fields appear in the URL when the `method` on `form` element is changed to `GET`?
4. Why is it considered bad practice to only validate form input with HTML5 and/or Javascript?
5. When is it appropriate to use `textarea` over `input`?

Deliver the answers as a .txt file together with your other deliverables.

Deliverables

Submission should be uploaded as a zip file into Blackboard before the deadline. Submissions are ONLY accepted via Blackboard. We DON'T accept late assignments. Emails or any other messages with late assignments are automatically discarded without further communication.