

API Specification

Silhouette

Group 12:
Mats Engelen
Lars Erik Faber
Håkon Marthinsen

Contents

1	Intro	3
2	Group Description	3
2.1	Delegation of Work	3
3	Background	3
4	Design Specification	3
4.1	Design Patterns	3
4.2	Design Decisions	3
4.3	Personal Decisions	3
5	Project Structure (File Structure)	3
5.1	Type Reference Documentation	3
6	Client Code	3
6.1	Scenarios and Solutions	3
7	User Testing	5
7.1	Description of setup	5
7.2	The Code	5
7.3	Feedback	5
8	Revised API	5
9	Project Discussion and Conclusion	5

1 Intro

2 Group Description

2.1 Delegation of Work

Description of Work

Each students writes about scenarios they have contributed with... Workload...

3 Background

Establish existing solutions ...

4 Design Specification

High Level Design Principles...

4.1 Design Patterns

4.2 Design Decisions

4.3 Personal Decisions

5 Project Structure (File Structure)

5.1 Type Reference Documentation

Link to type doc...

6 Client Code

6.1 Scenarios and Solutions

Scenario 1

Make two rulesets, one that is a regular ruleset and one that is a grid ruleset. Give each ruleset a unique selector. For the regular ruleset, add a blue background and change the text color to #32a852. For the grid, define three columns and two rows of varied size. Lastly, apply both of the rulesets to a Container of type "header".

Scenario 1 - Proposed Solution

```
RuleSet color = new RuleSet(".color");
color.addRule("background-color", "blue");
color.addRule("color", Color.Hex(#32a852));

Grid grid = new Grid("#grid");
grid.setColumns("1fr", "100px", "2em");
grid.setRows("50%", "120px");

Container myHeader = new Container("header");
myHeader.applySelector(".color");
myHeader.applySelector("#grid");
```

Scenario 2

Make a table whose size changes dynamically, add values to the header row and add values to the rest of the rows as they are generated. Apply a class to the table and set a header color for the table.

Scenario 2 - Proposed Solution

```
Table table = new Table();
int col = headerArray.length;
int row = array.length;
RuleSet color = new RuleSet(".tableClass");
color.addRule("background-color", Color.RGB(255,255,255));
ColGroup headerCol = new ColGroup();

table.setSize(row, col);
table.applySelector(".tableClass");

for(int i= 0; i<row; i++){
    for(int j= 0; j<col; j++){
        while(i<1){
            table.insertHead(i,j, headerArray[j]);
        }
        table.insertValue(i,j, array[j])
        if(i<1){
            headerCol.addCol(col, color.toString());
        }
    }
}
```

Scenario 3

Make a frontpage and two article pages and share a common stylesheet between them. Add a simple ruleset that targets the body element and give it a lightgreen background to see that the changes have applied to all the pages. Lastly, generate the code.

Scenario 3 - Proposed Solution

```
HTML myWebsite = new HTML("example website");

Page frontPage = new Page("frontpage", "My Front Page");
Page articlePage1 = new Page("article", "My First Article");
Page articlePage2 = new Page("article", "My Second Article");

StyleSheet style = new StyleSheet("style.css");

RuleSet bg = new RuleSet("body");
bg.addRule("background-color", "lightgreen");

style.append(bg);

website.link("style.css");
website.addPage(frontPage);
website.addPage(articlePage1);
website.addPage(articlePage2);

website.initialize();
```

Scenario 4

Make a simple form consisting of three input fields for username, password and submit. Label each of the fields accordingly and give the fieldset a legend of "Login Credentials".

Scenario 4 - Proposed Solution

```
FieldSet myFields = new FieldSet();

Input username = new Input("text");
username.addLabel("Type Username");

Input password = new Input("password");
password.addLabel("Type Password");

Input submit = new Input("submit");

myFields.addFields(username, password, submit);
myFields.addLegend("Login Credentials");
```

7 User Testing

7.1 Description of setup

7.2 The Code

7.3 Feedback

8 Revised API

9 Project Discussion and Conclusion