// JAVASCRIPT OCH PROGRAMMERINGSMETODIK
// COURSEWORK ASSIGNMENT 1 (CW1)

// TEACHER : JASON DE DONNO
// BY: SANDRA FILIPSSON, MWD 2015
// DATE: 2016-02-23

BACKGROUND

The assignment was to develop a meaningful application that has either a practical use or educational value. The application should be able to be later implemented on an appropriate website. The aim was to develop my knowledge and practical experience of programming, also having an application, I could show for possible employers to be able to explain my knowledge within JavaScript. I desided to program a calculator, a calculator which was smaller than usual so it could be open in a new window without taking up the whole screen of the user. I decided this because the calculator in Windows 8 takes up the whole screen, without any possibilty to making it smaller. As a user of calculator, I was bothered by this.

PROBLEM

I decided to do a basic calculator, that have the most common operands and solves the most common problems. I divided the functions into different groups. This because I wanted the design and code to be easy to develop/change further on.

- Operands; plus, minus, divide, multiply, comma and equal to.
- Deletes; backspace, CE and C.
- Extras; plus minus, square root, percent and raised to.
- Numbers; zero, one, two, three, four, five, six, seven, eight, nine.

I wanted the calculator to calculate a whole set of numbers, for example a set of 5 numbers with operands, first when the user clicks the equal button, depending on the rules for which operand should be calculated first in mathemathics. There was some problems with this directly from the start, when I added the numbers or operands in a varible I had to convert it to a String, to make it not calculate itself directly. Then when I wanted it to calculate, it wouldn't. I tried to convert back with **`Number()`** and **`parseInt()`** without any luck.

In my vision of the calculator (in a smaller window that wouldn't take up so much space from the users screen) I wanted to make the window that hold the calculator without an adressbar and as minimalistic as possible. After research about windows adressbar I came with the solution that not all browsers allow this anymore. Which mean it would work in some browsers but not all, and the program wouldn't be uniform if I would implement it. In Chrome for example doesn't allow you to remove the adressbar, it will only set it to be unchangeable. This with the code:

```
window.open("", "", "location=no");
```

In the beginning there was a problem with having them both in one file so the quick solution was to have them in one each. The feedback I got was why I had two js files in my project. I decided to fix the problem the hard way. The problem was that all the code

that was executed directly gave an error, when the window was active and the code that was executed for the non-active window. This meant I had to have a way of checking which window was active. The very simple solution I came up with in the end, after testing window.onload without success, was to check a specific id in that window, if it didn't return null, the code would execute.

PROPOSED SOLUTION

I was inspired by the Windows 8 calculator for doing the functions in the calculator and which functions there should be in the calculator. After the problems with making the numbers calculate, I did som more testing of the Windows 8 calculator in how it worked. I realised it always calculated the numbers by default at the fourth input. This got me to change my solution. In many ways, the best way to design, is the most common way to design. This because the user is used to how something works at most cases and will anticipate a new calculator to work in the same way. In the meantime the equal button calculates a set of numbers at the third input. This made me to set an input in **`addtoInt()`** function. In the function you input an array(the array of the different expressions your calculating) and a value for when the functions gonna start calculating. In my cases, either three or four.

RESULT AND TESTING

I started to test the calculator right from the start and when the design was set, I tested if the buttons were appropriate, by imagining to click and the number would output. When I had written the functions for outputing in textarea I would test the calculator by clicking buttons to see the output in the textarea. After the most of my functions where done, while testing them, I realised there was much validation to be done. I had to define rules for what the user was allowed to input in the testarea. After having grand ideas at the beginning I thought about what I was programming. I am doing a basic calculator, which means for me that I should have more defined rules. Instead of allowing the user to input everything and then giving the user a message that he/she input wrong, I decided to have the possible input very strict. The **`kHandler()`** function recognize your keystrokes and inputs them, but only the valid ones. The **`validate()`** function goes through what button or key you have pressed and inputs them in the textarea depending on the rules.

- The user shouldn't be able to input 2 operands in a row. (If not pressed the extra plusminus function, which inputs 2 operands by itself) The user can't input for example 2 + + 5 but with plusminus can input 5+-2, the second number becomes a negative number.
- The user shouldn't be able to input more than one comma in every expression. The user can't input for example 2.2.2 + 5, but can input 2.34 - 0.34
- If the user removes all the numbers/signs, the textarea should show a zero.
- If the user input a point without anything before the point, it should be added a zero before the point.
- If the calculated answer is Infinity then make the answer zero.