1.

I have been using Chrome ever since I started web programming and had not have a lot of experience with the other ones.

For Chrome, there are two things I love the most -

1. Debugging - Chrome made it super easy for developers to spot bugs by using the web page inspection. The tools I usually use are writing to the console (console.log()) and assertions (console.assert()) made it really convenient to detect JavaScript bugs such as mis-saving a NULL object to a variable, off by one error, or incorrect variable usage and etc.

2. Styling - When I code in CSS, I've always enjoyed using the inspector's tool to check if I have the correct appearance I wanted it to be. For those small errors I make, I can always try it out and fix it while watching the changes I make instantly.

2.

For a single event, if the there is an event handler at the DOM object level, then the object will be handled as assigned; if there is no event handler specified, the event will be passed out to its parent object....parent's parent object...until it has been handled or reaches the window document. Cases where I would want to use it including

1. assigning several DOM Objects capturing one single event and each of the objects do its own job when the event happens

2. having several similar events being handled together. For example, if we want to make a huge table with each of the cell inside turn yellow when clicked. The easiest approach will be putting event listeners into every single cell inside the table, but this would lead to huge pain and can pretty easily crash the browser. Instead, we could have the event handler at table element level and check which cell has been clicked.

3.

JSONP has an extra padding over JSON that is it has an additional callback function wrapping the data itself. When it is requested, JSON passes back JS objects and JSONP passes back a callback parameter. In this case, we are able to load the JSON as a script file so that the same origin policy no longer enforced. As a result, we can cross domain boundaries and load data from other domains.

4.

CSS3 is always the best way to style an HTML document because it is really simple and selectors for generalized or complex objects is always so helpful. I would say only use JS/JQuery for styling when the object is unpredictable, appearance changes when event happens, or special effects happen.

5.

JavaScript

// question not clearly stated so I am assuming there are

// only strings and array of strings in the initial array

// all special cases such as empty string are treated the

// same way as regular string

function flattenString(arr) {

var result = "";

for (var i = 0; i < arr.length; i++) {

var current = arr[i];

if (typeof current == 'string' || current instanceof String) {

result += " " + current;

} else if (current instanceof Array){

for (var j = 0; j < current.length; j++) {

result += " " + current[j];

}

}

}

console.log(result);

}

6.

A small example I have done was summing all elements in a large array. My approach was to have several threads simultaneously sum part of the array and call start() on each thread object to actually run it in parallel. I used the fork join algorithm that can really simply avoid the issue of race condition.

7.

I have been using DropBox for a really long time. I choose DropBox because it is super easy and convenient. It synchronizes the files to my local folder automatically. I can just use it as it is my local folder and it automatically updates whenever I make a change to my files. Additionally, there are many add-ons available such as the Mover which enables online backup of Git, mysql and etc.

8.

There were, in fact, two difficult projects I have encountered.

The first one was schoolwork – where I had to build a campus navigation map from scratch in Java. It is the first time I fully designing such a big project so that there are many problems I encountered during that time. For example, for my basic OOD model, I did not fully copy in and copy out that led to a pretty hard to find bug later on in the process. I also had to deal with generics and tried using MVC model, which are all fresh new skills for me at that time. If I could have done it again, I will test through the process as I code even more carefully so that I could have caught bugs earlier to avoid wasting of time later on.

The second one was during my internship at Baidu. The project itself may not be as difficult, however I find it hard because the projects I do at work is so different from those I did at school – at school, whenever I get assigned sometime, I am for sure know that it can be done and there must be a way to solve it; however at work, sometimes though I seek for the solution real hard and look for all kinds of solution online, the problem can still not be solved. This is a transaction from school to work, and the only solution for me is just to stay strong and work through it.