Macintosh HD:Users:ecstv:Documents:Matt Gussin:DOCUMENTS:Open Day Posters and banners etc:2016:5275_UoS_OpenDayTimetable_PSE Folder:Links:black.epsMacintosh HD:Users:ecstv:Documents:Matt Gussin:Media peices/logos/patterns:Logos:University Logos:university_southampton_black.eps

**Functional Programming and Big Data Processing**

This practical exercise exposes students to functional programming, and allows them to write functional code that is used in the context of a distributed data processing problem. Students will get hands on experience of using map, reduce, filter and sort operations with functions they define.

Time needed: ~1 hour

**Learning outcomes**

Having completed this practical, students will be able to:

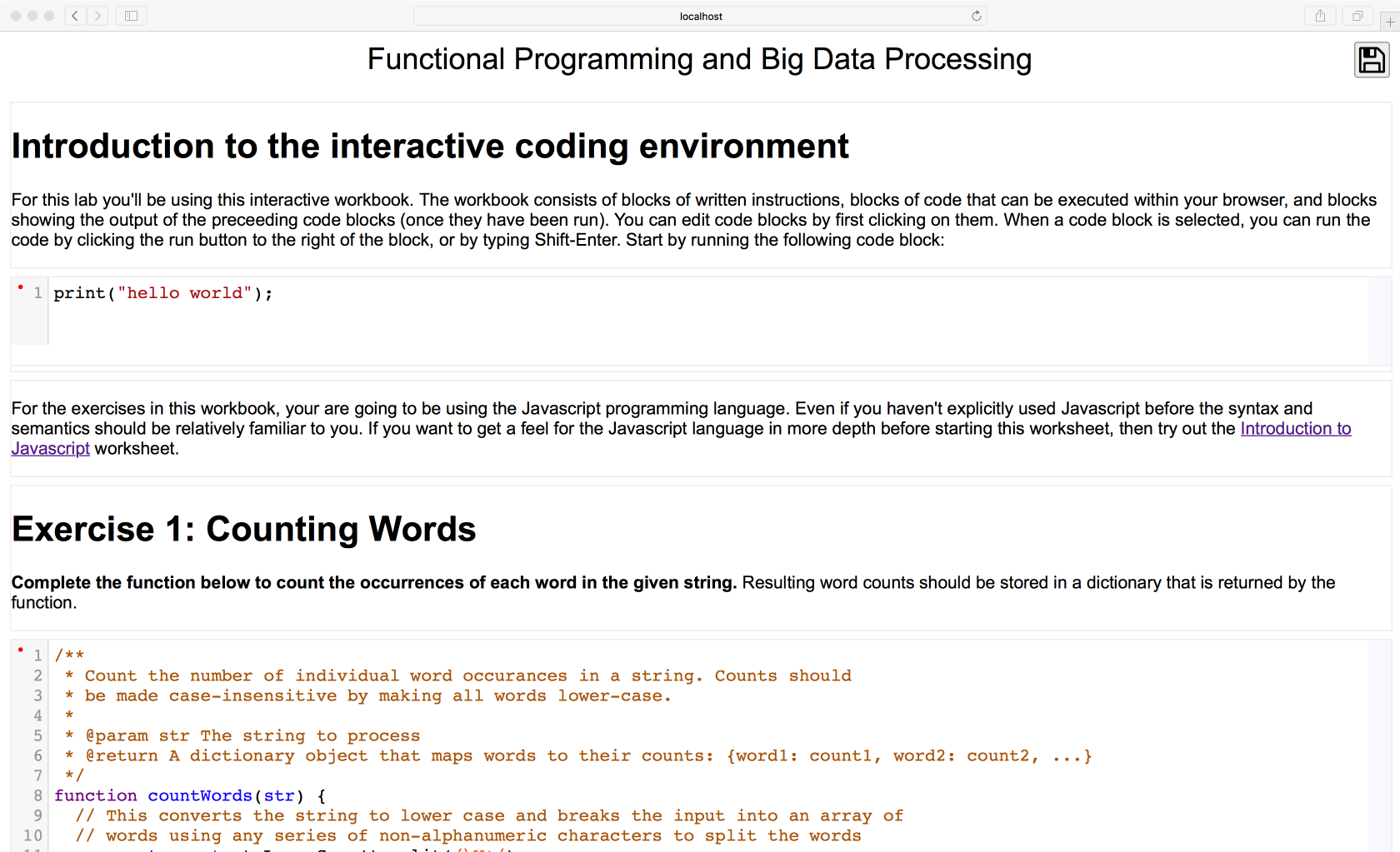
* Demonstrate understanding some of the challenges associated with big data and distributed data processing.
* Demonstrate practical use of `map`, `reduce`, `filter` and `sort` in a functional programming language.
* Show experience writing functions for distributed processing.

**Syllabus mappings**

* AQA: 4.4.1.9 Decomposition; 4.4.1.10 Composition; 4.11.1 Big Data (volume; distributed processing; functional programming as a solution); 4.12.2.1 Functional language programs (use of map, filter, reduce).
* OCR: 2.1.5 Thinking concurrently; 2.2.2 Computational methods (problem decomposition; divide and conquer)

**Equipment needed**

* Each student needs access to a computer with a modern web browser



The aim of this activity is to expose students to functional programming concepts in the context of processing distributed data effectively. Students should undertake to the “Distributed data processing” activity before starting this one to understand the base concepts with respect to distributing computation.

This activity takes the form of an interactive worksheet which is available at: <http://js.compsci.school/FunctionalBigData>

The worksheet covers:

* Writing a function to count words in a string, storing results in a dictionary object.
* Counting words across an array of strings,
  + using a loop, and,
  + using map.
* Aggregating word counts from different strings into a single accumulator using reduce.
* Sorting word counts with a custom comparator function.
* Composing multiple functional calls in the Map-Reduce methodology to achieve the task of computing sorted aggregate word counts from multiple strings.
* Applying the above composition to a large body of data stored on several remote servers,
  + using a *pull* methodology, where all the data is copied to the students’ browser, and the computation happens serially;
  + using a *push* methodology, where the raw data remains on the remote servers, and the function to be used in a map operation is pushed to each server, and the results returned for aggregation and sorting; and
  + using a *push* methodology, where the raw data remains on the remote servers, and the functions to be used in a both a map and intermediate reduce operation are pushed to each server, and the results returned for final aggregation and sorting.

Whilst the worksheet walks the students through the key concepts, however additional classroom activities might be required to explain the theory around functional programming in more detail. The worksheet itself requires the students to use the JavaScript programming language. Whilst it is unlikely that the students have used JavaScript extensively in the past the syntax should be familiar and the code that must be written to complete the exercises is relatively straight-forward. In addition, the worksheet contains a link to another worksheet (<http://js.compsci.school/JSIntro>) which introduces JavaScript fundamentals if required.

The worksheet should work in most modern browsers, but it has only been extensively tested in Chrome and Safari.

Should you wish to modify or customise a copy of the worksheet, you can edit and save by visiting:

<http://js.compsci.school/FunctionalBigData?expert>

The <http://js.compsci.school/> website also allows you to create your own custom worksheets should you so wish. The source code for the website is available on-line at <https://github.com/jonhare/jsworksheets> should you wish to see how they work. Please let us know if you find this tool is useful for your teaching. If you find any bugs or have any feature requests, please log them in the GitHub issues tracker: <https://github.com/jonhare/jsworksheets/issues>.

The source code for servers that hosts the data and performs the remote map and reduce procedures in the Functional Big Data worksheet is available at <https://github.com/jonhare/js-mr-demo>.

In Electronics and Computer Science (ECS), we have created an environment that has inspired an extraordinary level of achievement. ECS is one of the best places in the world to study electronics and electrical engineering, computer science and Software Engineering.

