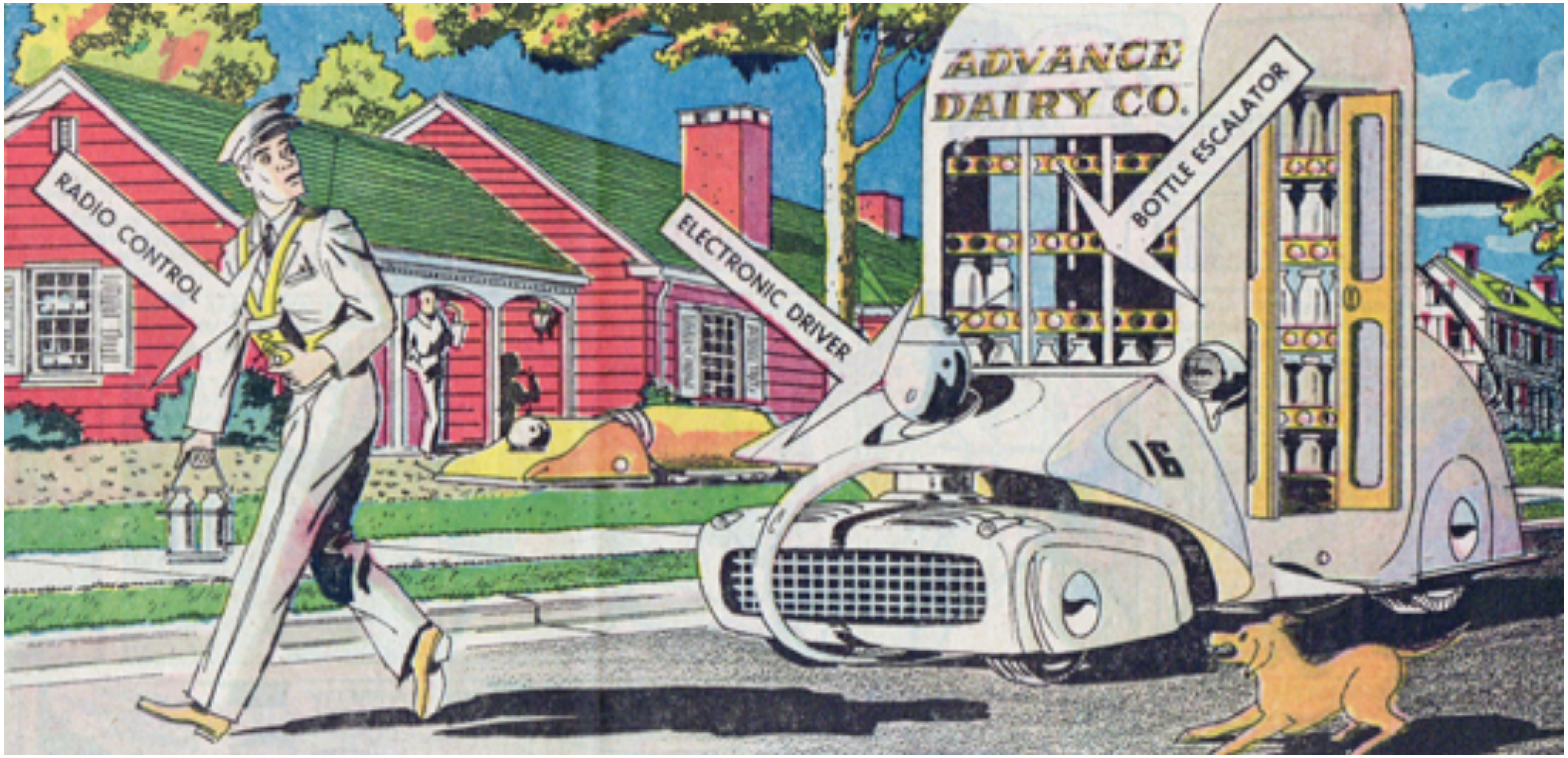
CM1015 SOFTWARE DESIGN AND DEVELOPMENT

Lecture 1

Introduction to JavaScript

The Future


RADIO CONTROL
ELECTRONIC DRIVER
ADVANCE DAIRY CO.
BOTTLE ESCALATOR
16

Alan Kay

Xerox PARC (a computer science think tank for which Kay was a founding principal in 1970) was set up in Palo Alto, California to be as far away from corporate headquarters as possible and still be in the continental U.S. [Executives who visited] were worried about the future, and they would badger us about what's going to happen to us.

Finally, I said:

'Look, **the best way to predict the future is to invent it**.

This is the century in which you can be proactive about the future; you don't have to be reactive. The whole idea of having scientists and technology is that those things you can envision and describe can actually be built.'

Content

- Introduction to programming
- Software to be used
- Output
- Strings
- Control Characters
- Concatenation
- Input

- Exercises

Introduction

- Programming is the creation of instructions for an *agent* to carry out. E.g.
  - A recipe for a cook
  - A blueprint for a builder
  - A score for a musician
- In each case we need to provide an unambiguous list of instructions in sequence in a language that the agent can understand
- For a computer program we carry out a similar task however computers are different to human agents.

Computer Programs

- Computers only understand one language – binary – which is just a sequence of 1s and 0s.
- Fortunately for us software has been written which will translate between binary and a language that we can understand.
- This means we can write programs in a *high-level* language

- Another difference between computers and a human agent is reasoning.
- If a omelette recipe asked you to add “2 iggs” then you would probably realise this should be “2 eggs” – a computer won’t do this.

Software 1: Browser

- The only piece of software we really need to run JavaScript is a web browser.
- You can use your own favourite browser but be aware that different browsers have added extra functionality to JavaScript.
- I will use Chrome for most of my examples but may use Safari as well.

Example 1: Hello Program (I)

- Our first program will simply output some text
- A group of characters in programming is call a **string**
- We enclose strings between “….“

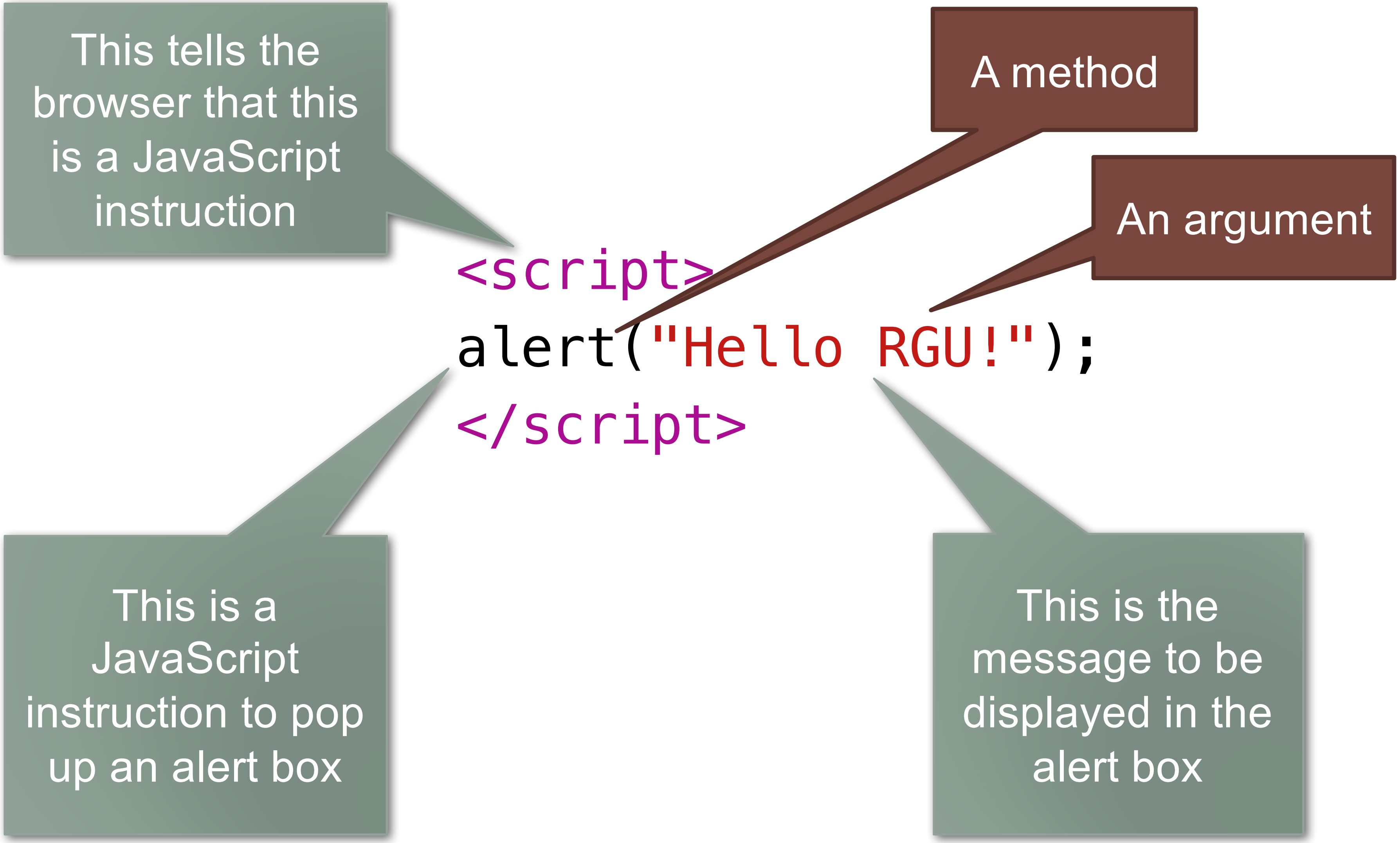
- Type the following into a text file (with the extension html):

```
<script>
alert("Hello RGU!");
</script>
```

- Open the file with a browser

Example 1: Discussion

- What did the statement actually do?



This tells the browser that this is a JavaScript instruction

A method

An argument

```
<script>
alert("Hello RGU!");
</script>
```

This is a JavaScript instruction to pop up an alert box

This is the message to be displayed in the alert box

Software 2: Runner

- A JavaScript runner allows us to run JavaScript instructions without having to type them into the browser bar
- We will use the following:

  http://javascript.cs.lmu.edu/runner/
- Go to the site now and save it in your favourites
- Try entering the following instructions and clicking on Run

```
alert("Hello")
alert("Goodbye")
alert(2)
```

- You should find that with the runner you can enter more than one instruction in the window but each instruction will only be carried out one line at a time.

Files

- As you start to write more instructions its seems sensible to keep a copy of them.
- To do this choose your favorite text editor, e.g. Notepad
- Type the instructions into a file and save it with an appropriate filename.

Software 3: Shell

- A third approach to running JavaScript statements is to use a Shell. One such shell is available at:

  http://www.squarefree.com/shell/shell.html

- Again save this in your favourites.
- Try typing in a few commands to see how this works.
- Note that you can use the up-arrow to get your input history – very useful if you make mistakes.

Example 2: Concatenation

- Our examples have all been outputting a single string but we can “add” strings together using the concatenation operator (+).
- For example, try the following commands:

```
alert("Hello " + "RGU!")
alert(3 + " is the magic number")
```

- We will make more use of this in the second half of the class.
- As the concatenation operator is the same as addition this can have some odd effects – see the exercises.

Example 3: Control Characters

- All of our examples have produced separate alert windows for each line but we can use the following control characters to output text with some “texture”.
  - `\n` gives a new line
  - `\t` gives a tab space (but may not work on some browsers)
  - `\"` displays as "
  - `\'` displays as `
- Use control characters to output:

```
1
      "2"
            3
```

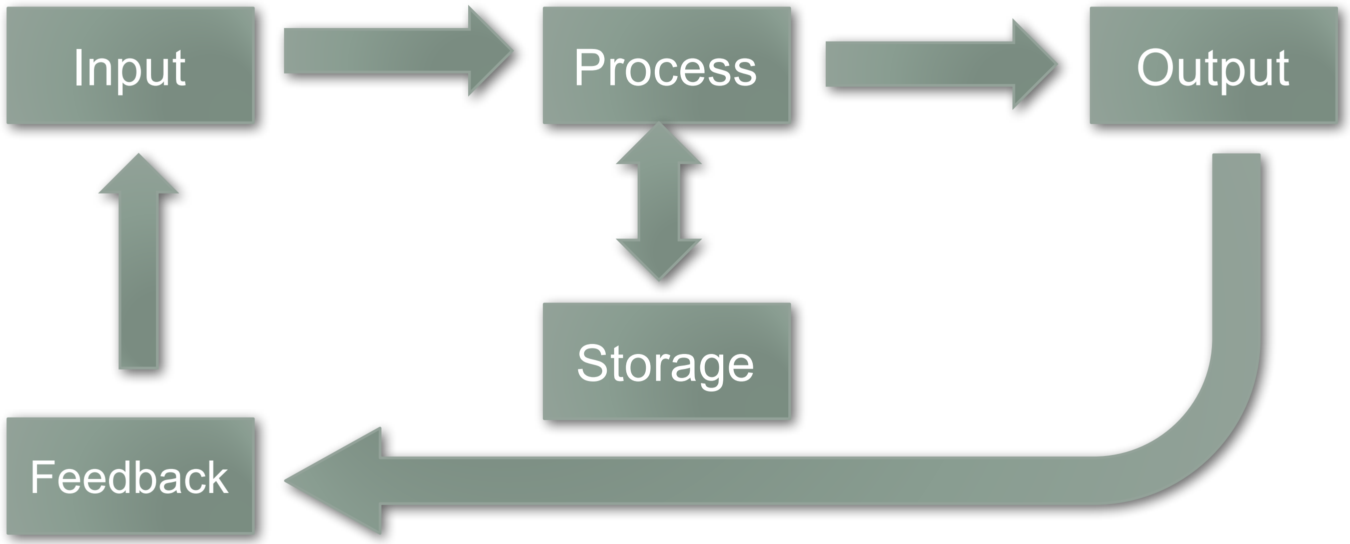
- with a single statement

Different flavours of JavaScript

- Note that using the shell we can display a string on the same page using the `print()` method
- Also in EJ there is a similar method
- Both of these are examples of non-standard JavaScript where the authors of the site/book have added some extra functionality to make life easier for the programmer.
- There are also browser-dependent versions of JavaScript
- Beware of these little modifications – they can cause big problems.
- Try `print("Hello")` – what happens?

The IPO model

- One model of a general computer system is the Input-Process-Output model. There are some additions to this as well.



- We have dealt with the output (and processing) side of things so now for input.

Input

- To get input in JavaScript we will rely on pop-up boxes.
- Type the following into JavaScript to see how this works:

```
prompt("What is you name? ")
```

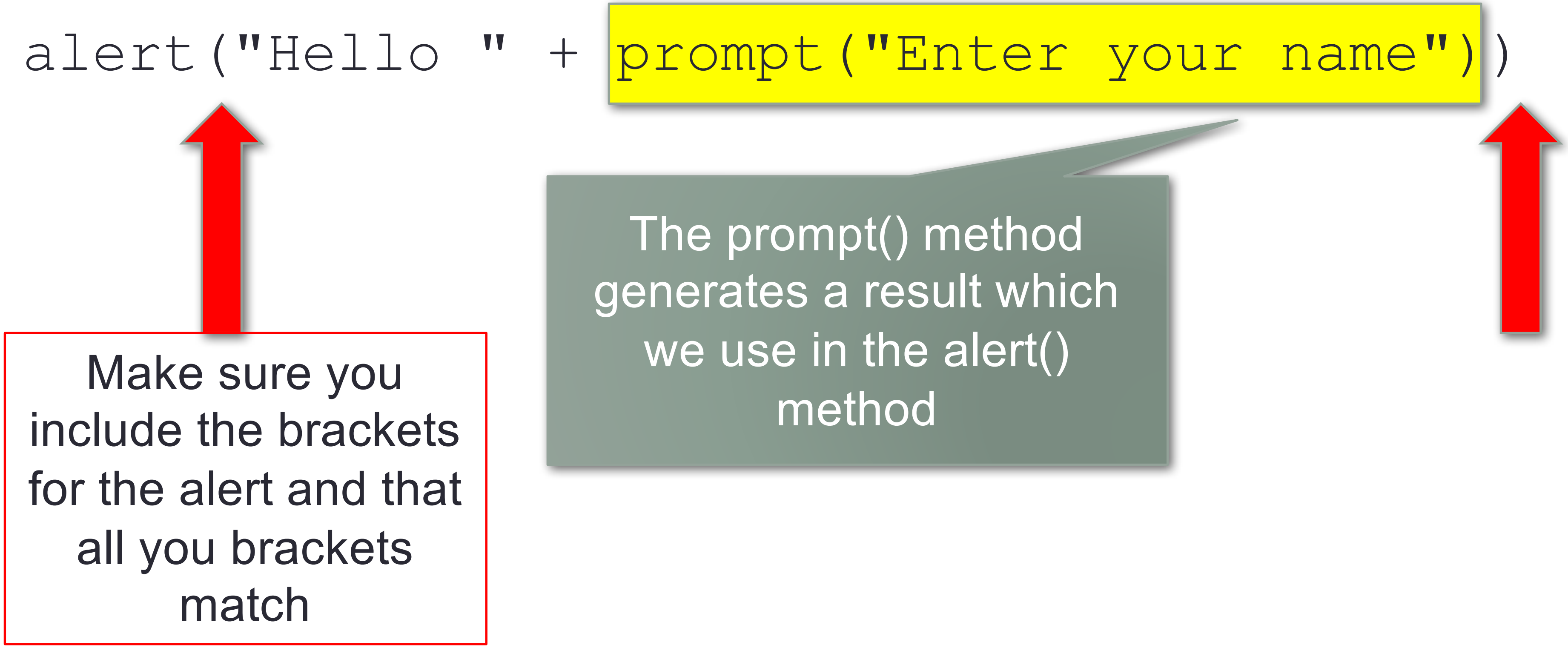
- Another example of input which we will use later is a pop-up box which asks for a yes/no answer

```
confirm("Join LoveJavaScript.org?")
```

- If you run these in the Shell you will see a response but this is of little use unless we do something with the input.

Using the input

- In order to use the input we can start to embed methods.
- Try the following to see how this might work.



```
alert("Hello " + prompt("Enter your name"))
```

Make sure you include the brackets for the alert and that all you brackets match

The prompt() method generates a result which we use in the alert() method

Terminology

- The `alert()` method generates a result – we say that it **returns** this value.
- What we are doing with the returned value at the moment is not very efficient.
- Next time we will look at storing the value generated by an input and using this effectively.

Summary

- Introduction to programming
- Software to be used
- Output
- Strings
- Control Characters
- Concatenation
- Input

Exercises

- Now try the exercises on the lab sheet.
- Try to get through as many of the exercises that you can using the browser address bar, a JavaScript runner and a JavaScript shell
- Experiment and investigate.
- If you don’t complete the exercises in class try to do them before the next CM1015 class

- Read chapter 1 of EJ for background (you might want to look through chapter 2 to see what we will be doing next)

Concatenation (Extra Note)

- When concatenating strings and numbers we need to make sure that we use parentheses (brackets) to make sure the output is as intended.
- JavaScript will work from left to right interpreting the + signs as addition or concatenation depending on the type of value it is working with (as soon as the result is a string it will assume concatenation)
- For example:

```
alert(1 + 2 + "some" + " words" + 1 + 2)
```

- Produces

```
3some words12
```