SOFTWARE PROGRAMMING

EGCO342 INFORMATION TECHNOLOGY IN DAILY LIFE

KANAT POOLSAWASD
DEPARTMENT OF COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Programming (1)

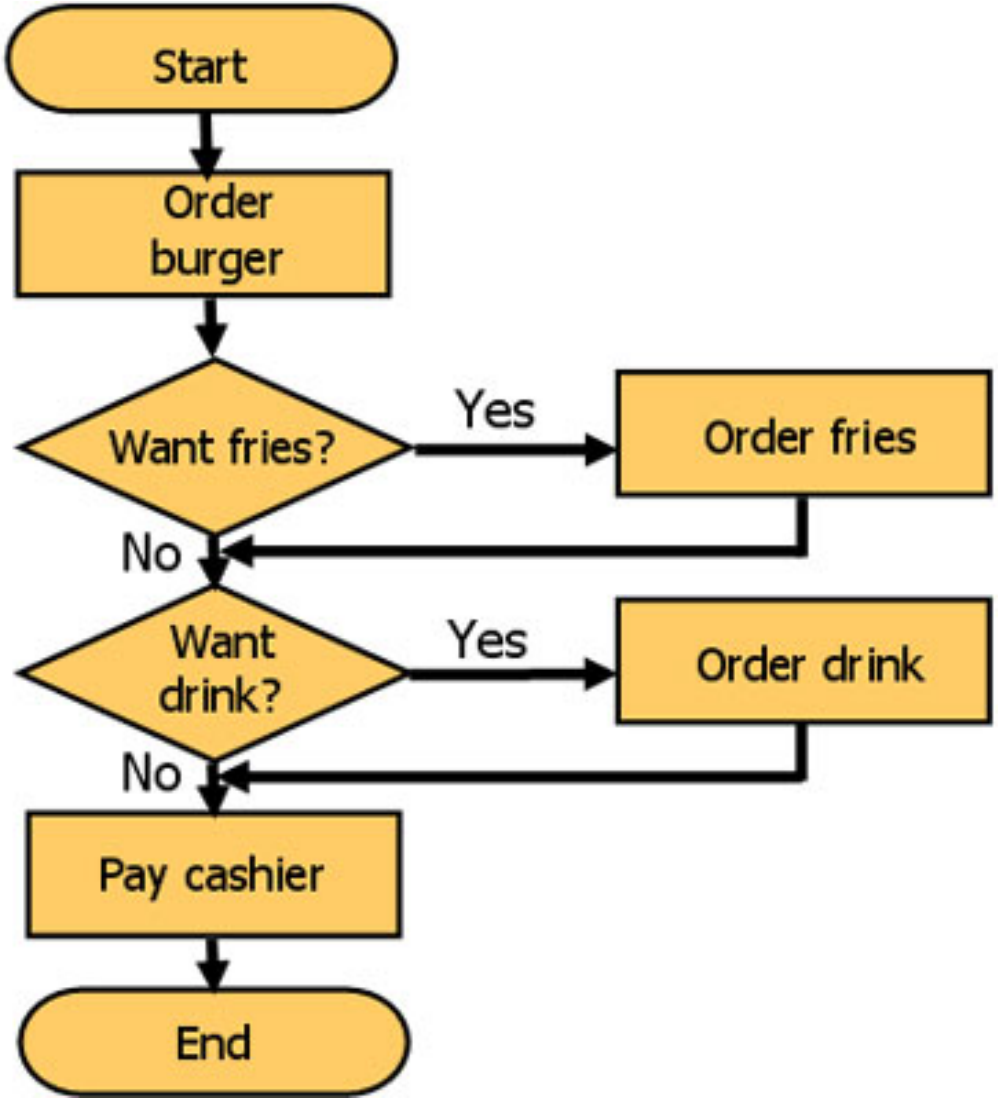
- Telling a computer what to do.
- Just moving number around and manipulate them.
  - Yes. That's all.
- But I can type and see pictures!
  - It's all ILLUSIONs. A character is just a number to computer.

    A picture is just a matrix of numbers each representing a color.

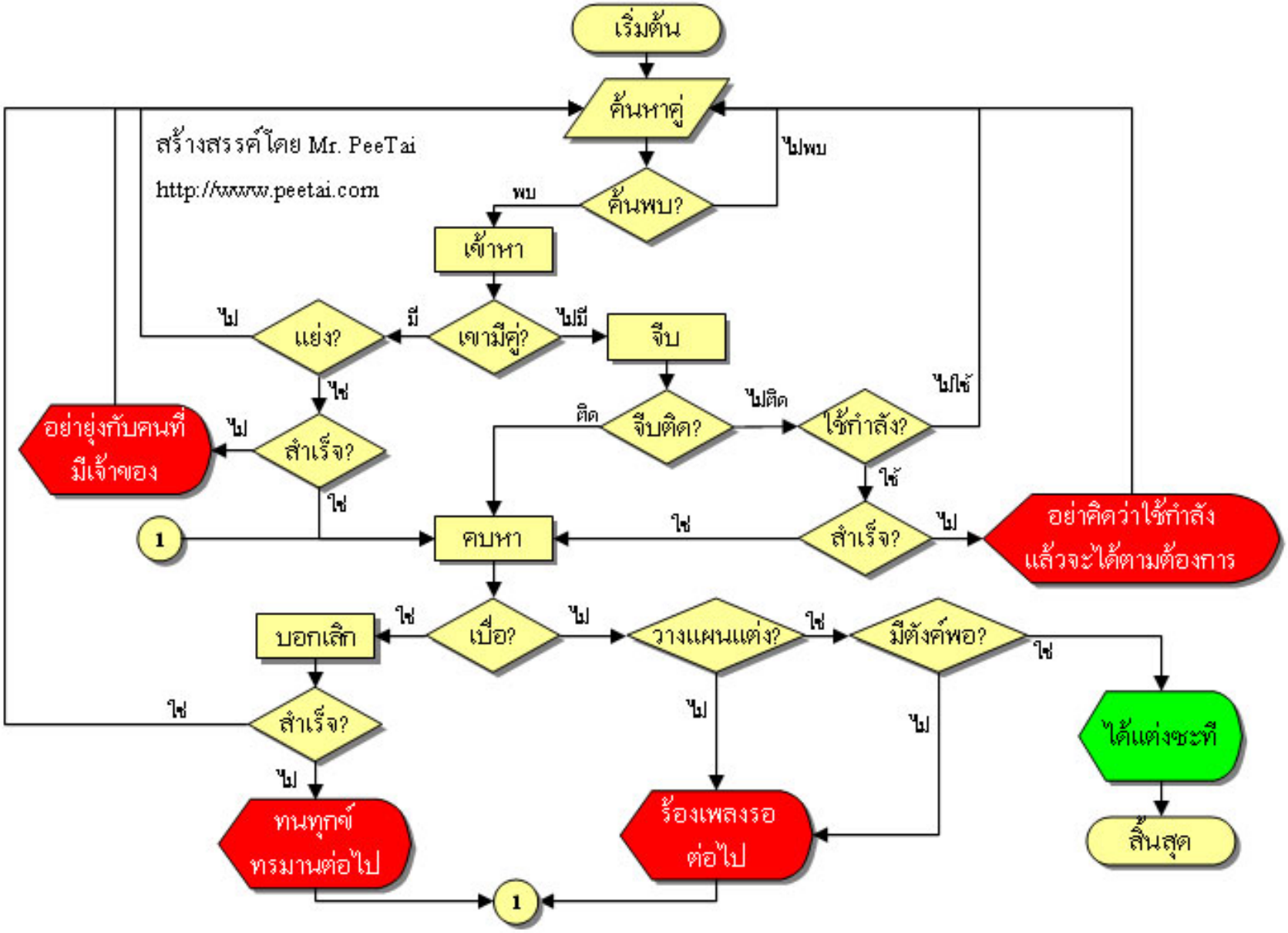
Programming (2)

- Algorithms are what move everything
- No you don’t need to be a genius to program. (You only need a genius to come up with a new cool algorithm.)
- To program you need to speak computer language.
  - 011001 is too hardcore.
  - We will go with a modified English.

Flowchart (1)


Start
Order burger
Want fries?
Yes
Order fries
No
Want drink?
Yes
Order drink
No
Pay cashier
End

Flowchart (2)


เริ่มต้น
ค้นหาคู่
สร้างสรรค์โดย Mr. PeeTai
http://www.peetai.com
ไม่พบ
ค้นพบ?
พบ
เข้าหา
ไม่
แย่ง?
มี
เขามีคู่?
ไม่มี
จีบ
ไม่
ใช่
อย่ายุ่งกับคนที่มีเจ้าของ
ไม่
สำเร็จ?
ใช่
ติด
จีบติด?
ไม่ติด
ใช้กำลัง?
ไม่ใช้
ใช้
1
คบหา
ใช่
สำเร็จ?
ไม่
อย่าคิดว่าใช้กำลังแล้วจะได้ตามต้องการ
ใช่
บอกเลิก
เบื่อ?
ไม่
วางแผนแต่ง?
ใช่
มีตังค์พอ?
ใช่
ใช่
สำเร็จ?
ไม่
ไม่
ไม่
ได้แต่งซะที
ทนทุกข์ทรมานต่อไป
ร้องเพลงรอต่อไป
สิ้นสุด
1

Pseudocode

Bold terms show actions that are common in programming, such as reading data, making decisions, printing, and so on.

1. **Ask the user** how many hours they worked today
2. If the number of hours worked < = 8,
   compute total pay without overtime
   otherwise,
   compute total pay with overtime pay
3. **Print** total pay

Underlined words are information items that appear repeatedly in the algorithm.

Coding

- Speaking the Language of the Computer
  - Syntax
    - Agreed-upon set of rules of language used
  - Keywords
    - Set of words with predefined meanings
  - Data types
    - Describe the kind of data being stored in memory
  - Operators
    - Coding symbols that represent fundamental actions

Compilation

- Compilation is the process of converting code into machine language
- A compiler reads the source code and translates it into machine language
- After compilation, programmers have an executable program

Interpreter

- Some programming languages do not have a compiler, but use an interpreter instead
  - The interpreter translates source code into a line-by-line intermediate form
  - Each line is executed before the next line is compiled
  - Programmers do not have to wait for the entire program to be recompiled each time they make a change
  - Programmers can immediately see the results of changes as they are making them

Coding Tools

- Integrated Development Environment (IDE) or Interactive Development Environment
  - Developmental tool that helps programmers write, compile, and test programs
- Every language has its own specific IDE
- Example: Visual Studio .NET, Xcode, Eclipse, Text Editor, etc.

Debugging

- Getting Rid of Errors
  - Process of running program over and over
    - To find errors
    - To make sure the program behaves the way it should

Finishing the Project

- Testing and Documentation
  - Internal testing – a group with the software company uses program in every way possible
  - External testing – people like those who will eventually purchase the program work with it

Programming Language

- There are about 50 well-known languages
  - http://www.levenez.com/lang/
- Total number of languages is estimated to be about 2500+
  - http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm
- You don’t need to learn all.
- You learn one you can do most of them.

Current Popular Languages

- Java
- JavaScript
- C#
- PHP
- Ruby
- Python
- C/C++

Generations of Programming Languages

Level	Generation	Example
Low	1GL	Machine
	2GL	Assembly
High	3GL	FORTRAN, BASIC, C, Java
	4GL	SQL
Natural	5GL	PROLOG

Programming Paradigm

- Procedural
  - Input -> Processing -> Output
- Object Oriented (OOP)
- Event Driven
- Functional
- Logical

Selecting the Right Language

- Programming team considers several factors
  - Space available
  - Speed required
  - Organizational resources available
  - Type of target application

Visual Studio .NET

- Builds object-oriented applications for:
  - Windows
  - The Web
  - Mobile Devices
- Easy to drag and drop entire programming components into application

C and C++

- C
  - Developed for system programmers
  - Provides higher-level programming features
  - if statements and for loops
- C++
  - Uses same symbols and keywords as C
  - Better security
  - Support for reuse of existing code
  - Includes object-oriented design

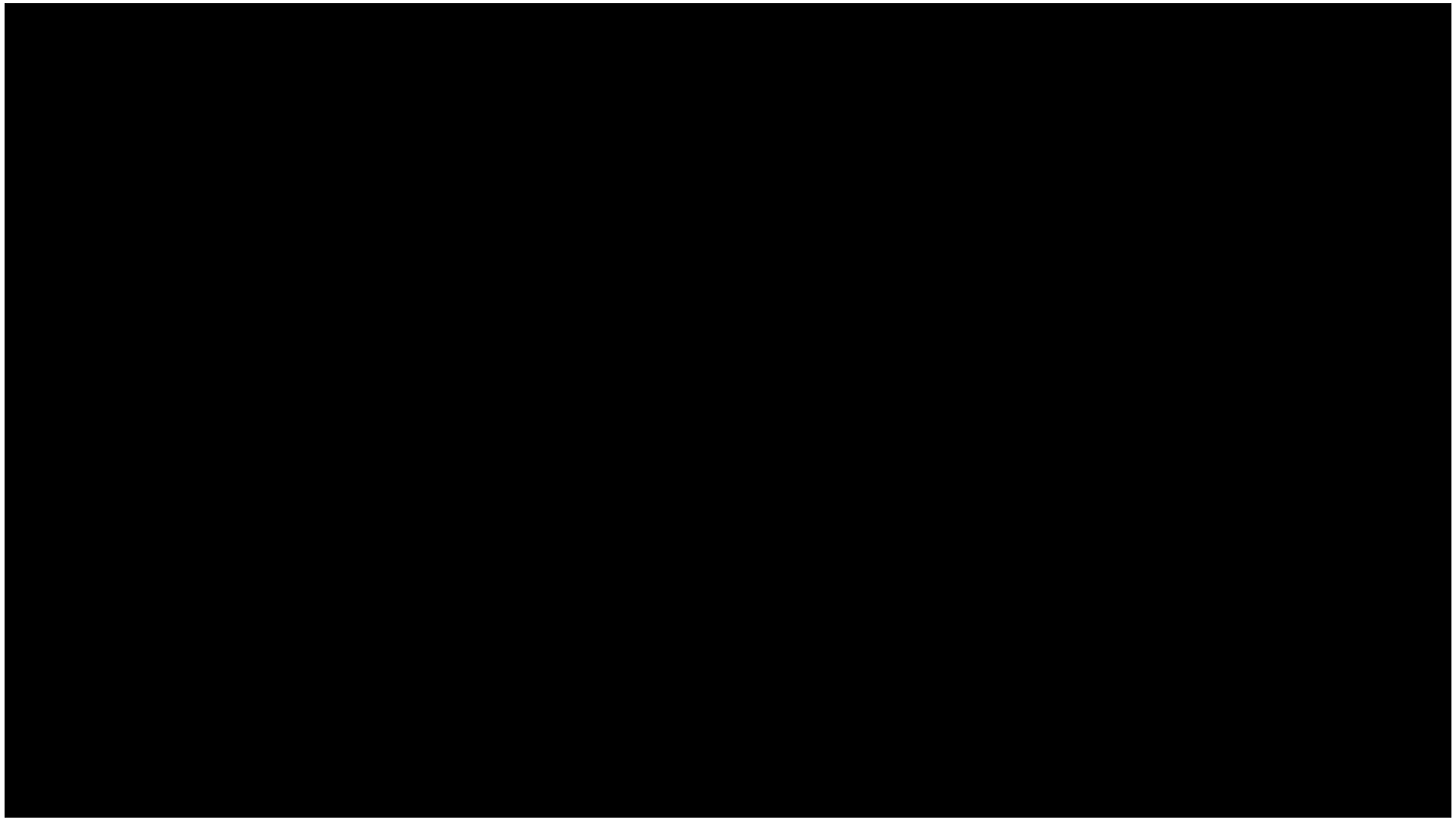
Java and C#

- Java
  - Object-oriented features
  - Large set of existing classes
  - Architecture neutral
  - Java applets: Small Java-based programs
- C#
  - Completing program released by Microsoft

Objective-C / Swift

- Language most often used to program applications to run under Mac OS X
  - Object-oriented language
  - Superset of the C language
  - Often used with library called Cocoa
- Swift is a general-purpose, multi-paradigm, compiled programming language developed by Apple Inc. for Apple Platform (iOS, macOS, watchOS, and tvOS)

Swift Playground



Building Web Applications

- HTML/XHTML
  - HyperText Markup Language/eXtensible HyperText Markup Language
  - Not a true programming language
  - Uses special symbols (tags) to control how Web pages are viewed

ASP, JSP, and PHP

- Used by programmers to build Web sites with interactive capabilities
- User supplies information that is translated into a request.
- Scripting code controls automatic writing of the custom page returned to user's computer

Flash and XML

- Adobe Flash
  - Used to develop Web-based multimedia
  - Includes its own scripting language, ActionScript
- XML (eXtensible Markup Language)
  - Enables designers to define data-based tags
  - Makes it easier for Web site to transfer key information on its page to another site

AJAX

- Asynchronous JavaScript And XML
  - Allows creation of Web applications that can update information without requiring a page refresh
  - Uses existing technologies to do more processing in the browser
  - Users have a more responsive experience

Building Mobile Applications

- Special languages and supporting tools help speed development of applications for mobile devices like smart phones and tablets
- Specific features include GPS capability, software keyboards, and touch-sensitive screens
- User interface must take smaller screen size into account

The Next Great Language

- Never easy to predict which language will become the next "great" language
- Experts predict that as projects grow in size, time to compile will also grow
- Interpreted languages could become more important because they have virtually no compile time
  - Python, Ruby, and Smalltalk

HTML (1)

- HyperText Markup Language
  - = Mark up the text with the format you want
- Look like this

```
<!DOCTYPE html>
<html>
<body>
  <h1>Hello Header</h1>
  <p>This is the content.</p>
</body>
</html>
```

HTML (2)

- Old Day
  - Just formatting text and pictures
- Now
  - Allow interaction
  - Behave like a program
- Interaction and Animation
  - Adobe Flash
  - HTML 5
  - WebGL
  - Java Applet
  - Microsoft Silverlight

Programming with HTML 5

- FireFox (Web Browser)
- FireBug (Debugging tool integrated into the FireFox)
- Text Editor (Coding Tool)

References

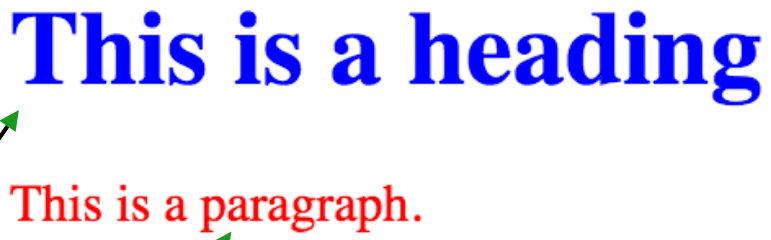
- JavaScript
  - http://www.w3schools.com/js/default.asp
- HTML5 Tags
  - http://www.w3schools.com/html/default.asp
- CSS
  - http://www.w3schools.com/css/default.asp

First HTML (1)

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>
</body>
</html>
```

First HTML (2)

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
  <h1 style="color:blue">This is a heading</h1>
  <p style="color:red">This is a paragraph.</p>
</body>
</html>
```



HTML Lists

```
<!DOCTYPE html>
<html>
<body>
<h2>Unordered List with Default Bullets</h2>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
</body>
</html>
```

HTML Table

```
<!DOCTYPE html>
<html>
<body>
<table border="1" style="width:100%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
</body>
</html>
```

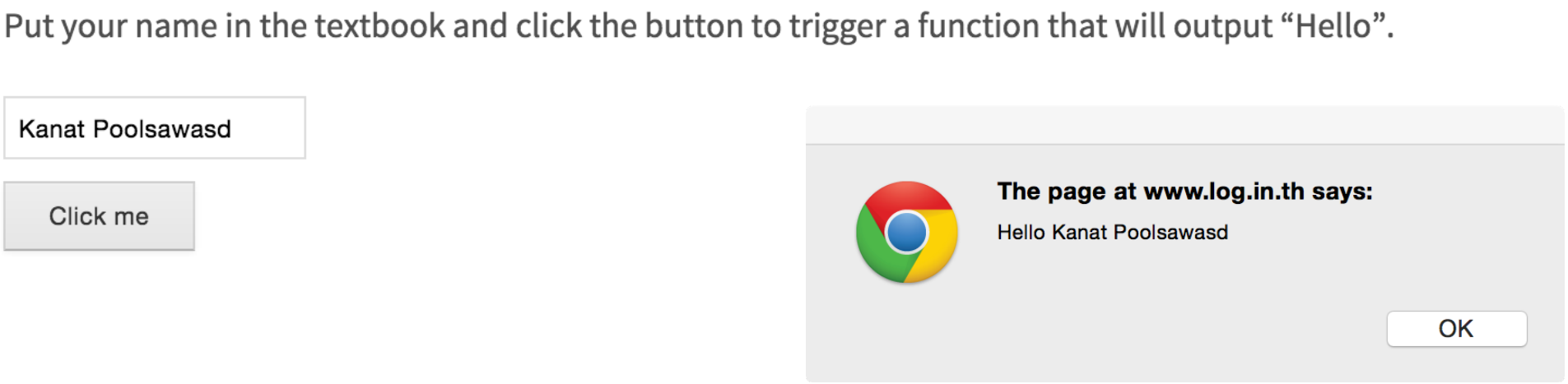
Jill	Smith	50
Eve	Jackson	94
John	Doe	80

HTML Table

Put your name in the textbook and click the button to trigger a function that will output “Hello”.

```
<input id="textbox" type="text" value=“”></input>
<button onclick="myClick()">Click me</button>

<script>
function myClick()
{
  var n;
  n = document.getElementById("textbox").value;
  alert("Hello " + n);
}
</script>
```



Assignment 7

- ให้ download ไฟล์ JavaScript Exercise.docx จากเว็บไซต์ของรายวิชา และตอบคำถามในเอกสารดังกล่าว ส่งกลับทาง e-mail (kanat@log.in.th) พร้อมไฟล์ .html ที่สร้างขึ้น