COMS 465:

Computer Mediated Communication

Plan

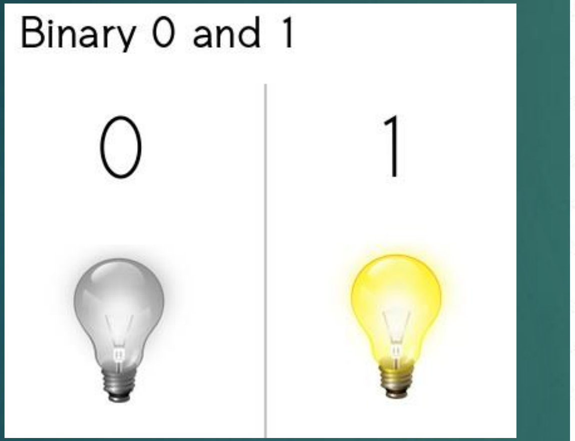
- Review
- Software
- Maker Exercise #1 – Code
- Preview

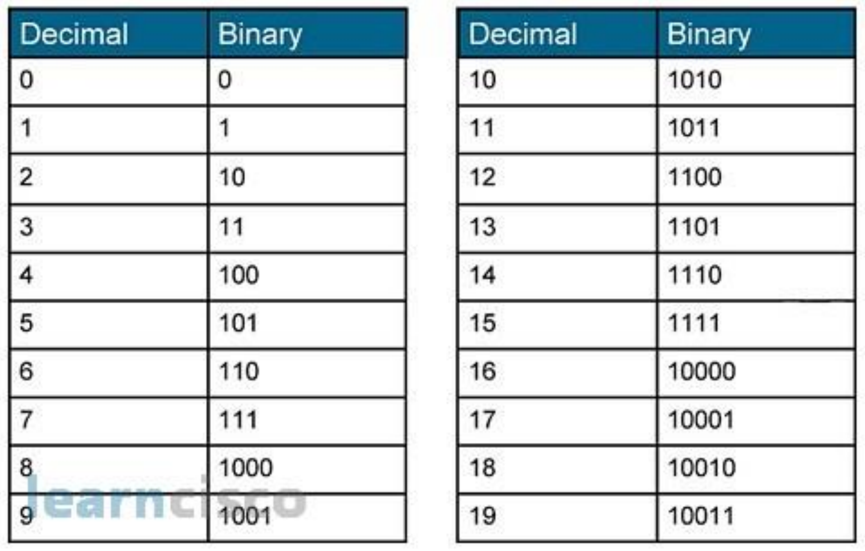
Review - Hardware

- Data Representation
- Element of the IPOS
  - Input
  - Processing
  - Output
  - Storage

Review - Hardware

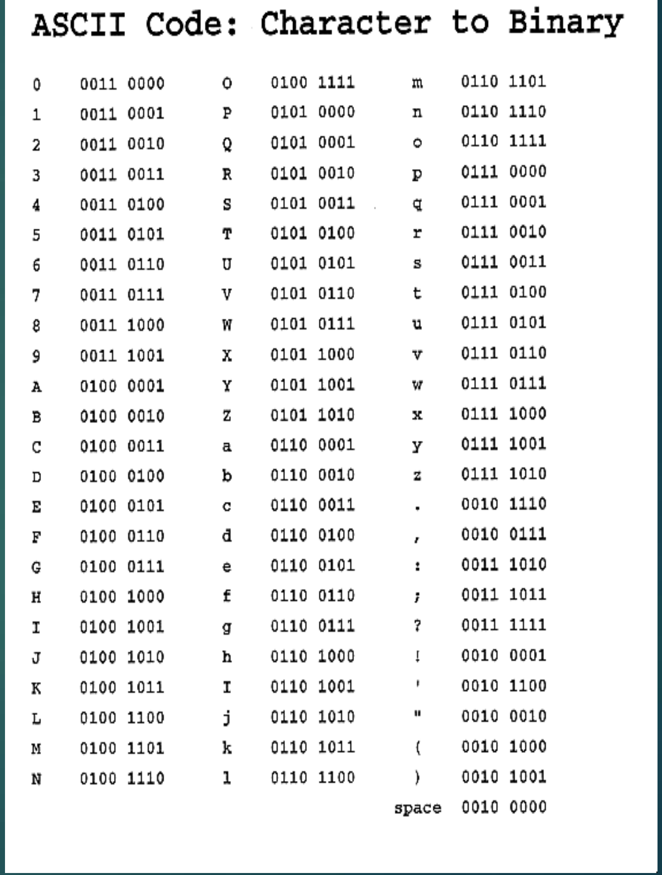
- Data Representation





Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Decimal	Binary
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011

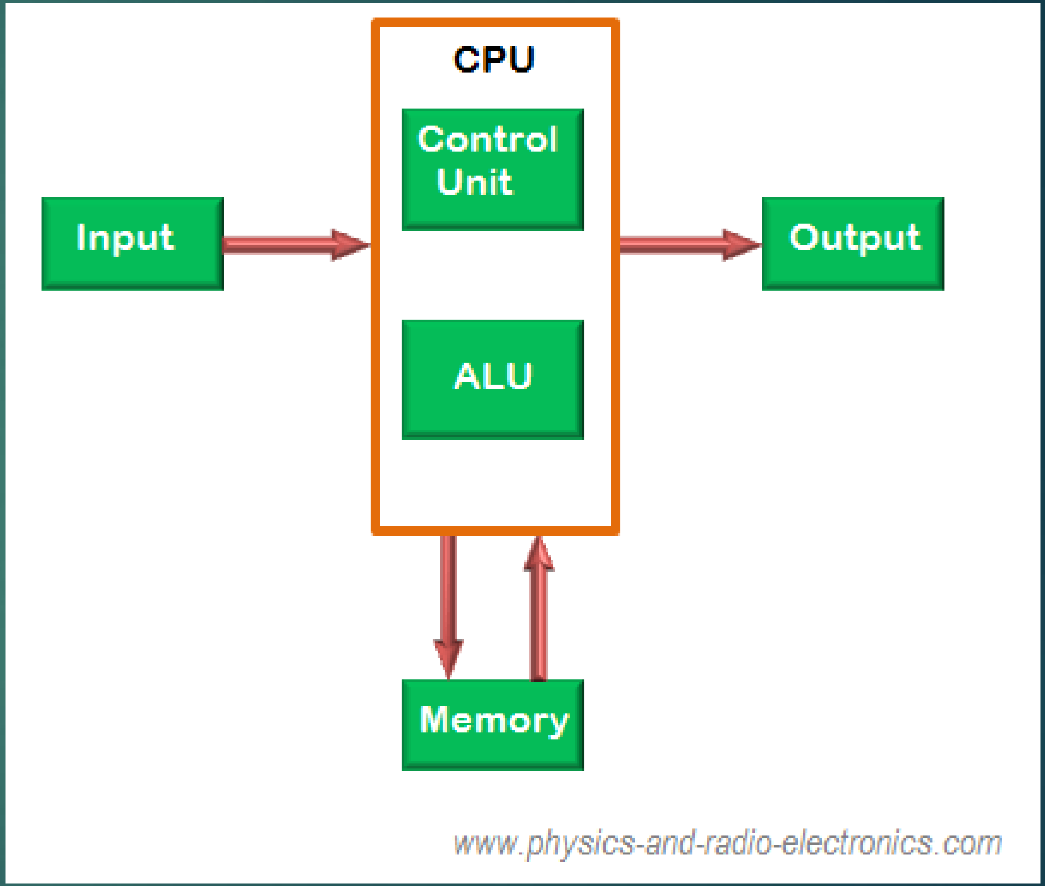


ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Review - Hardware

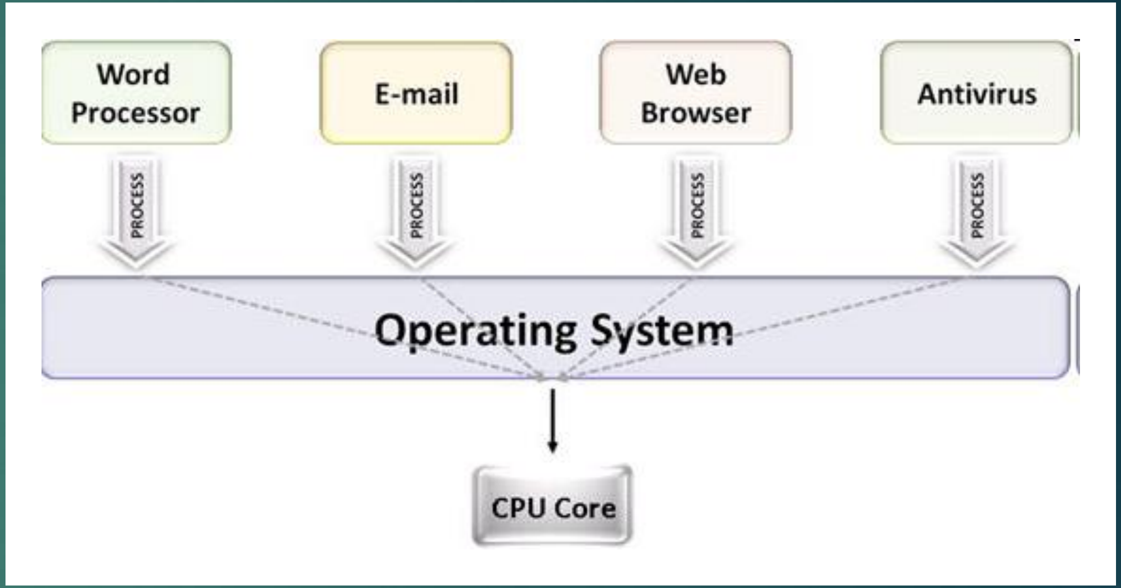
- Elements of the IPOS Cycle
  - Input devices
  - Output devices
  - Processing
  - Storage (memory)



www.physics-and-radio-electronics.com

Software

- Purpose
  - Software tells the computer what to do
  - Software dictates how the computer executes the IPOS cycle



Software

- Two Basic Types

System Software

Operating System
Utility Software

Application Software

Tools for specific tasks

Software

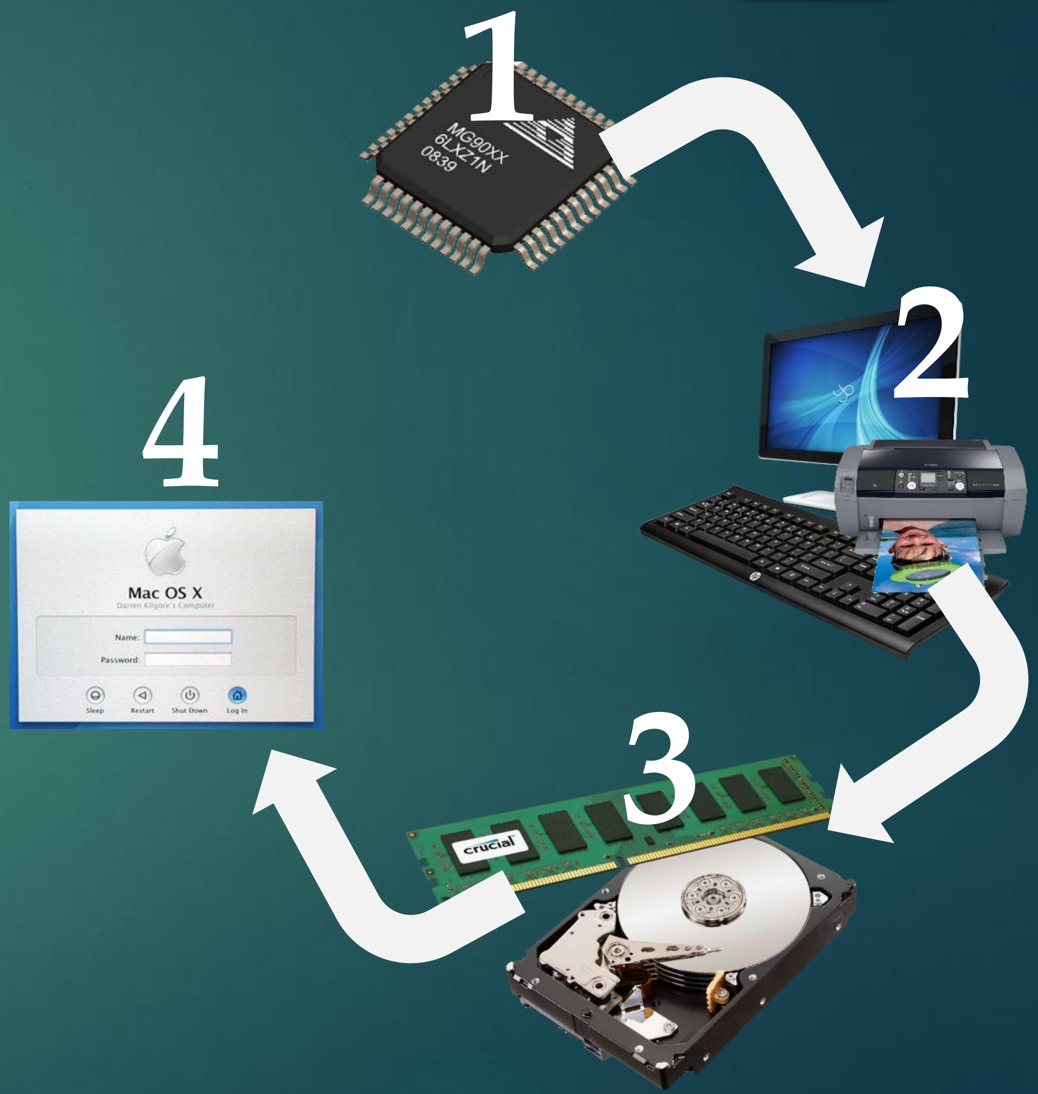
- Operating System – 5 Functions
  - Boot the Computer
  - Manage Programs
  - Manage Memory
  - Deal with I/O Devices
  - User Interface

OS is the traffic cop
of the computer

Software

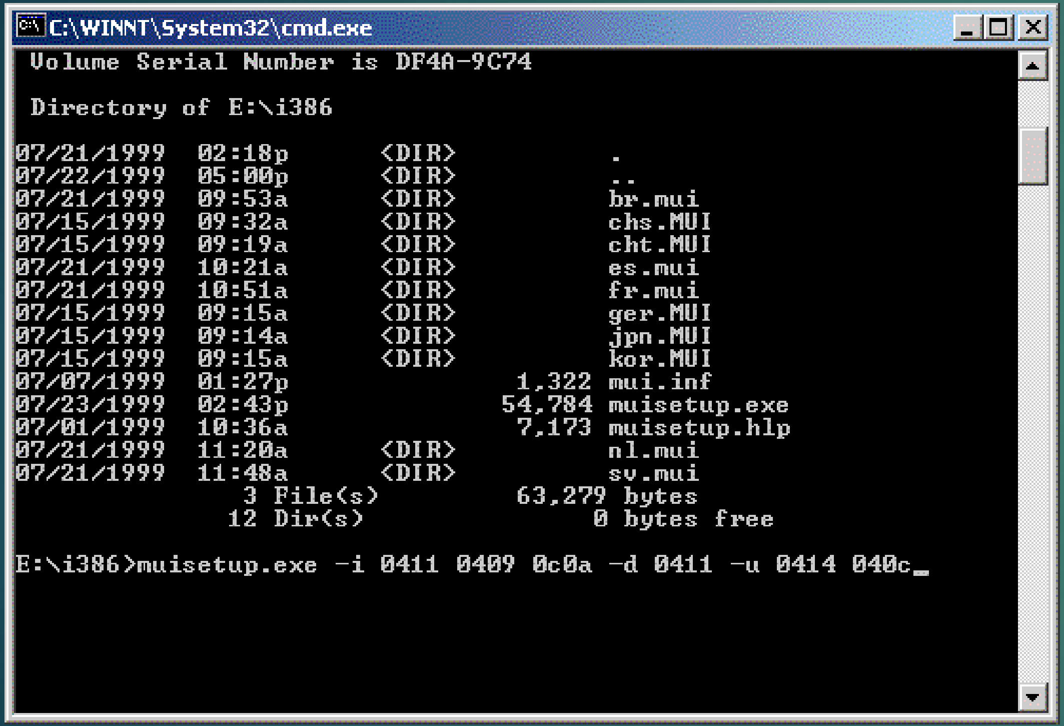
- Boot Process – 4 Basic Steps

  1. **BIOS** - Basic Input Output System is activated
  2. **POST** – Power-On Self Test checks for attached hardware devices
  3. **OS** – The OS kernel is loaded into RAM from hard disk
  4. **Setup** - Configuration and customization



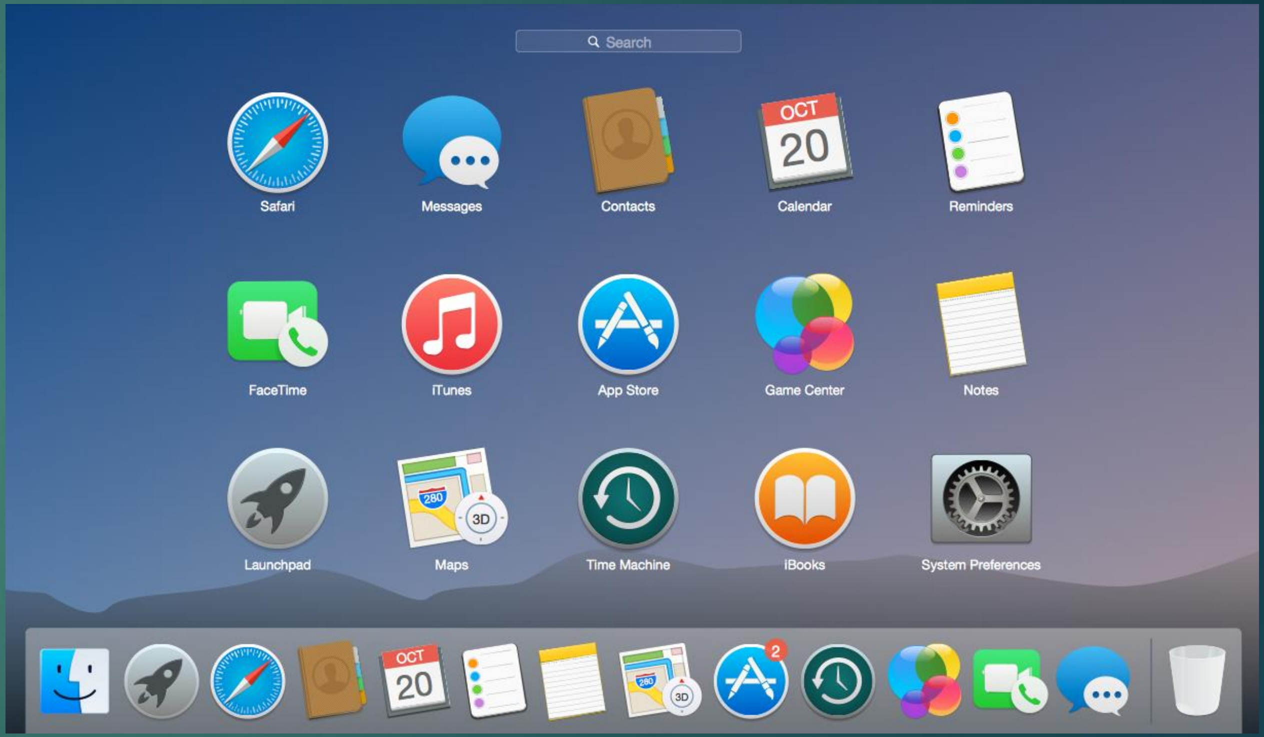
Software

- User Interface



```
 Volume Serial Number is DF4A-9C74

 Directory of E:\i386

07/21/1999  02:18p      <DIR>          .
07/22/1999  05:00p      <DIR>          ..
07/21/1999  09:53a      <DIR>          br.mui
07/15/1999  09:32a      <DIR>          chs.MUI
07/15/1999  09:19a      <DIR>          cht.MUI
07/21/1999  10:21a      <DIR>          es.mui
07/21/1999  10:51a      <DIR>          fr.mui
07/15/1999  09:15a      <DIR>          ger.MUI
07/15/1999  09:14a      <DIR>          jpn.MUI
07/15/1999  09:15a      <DIR>          kor.MUI
07/07/1999  01:27p               1,322 mui.inf
07/23/1999  02:43p              54,784 muisetup.exe
07/01/1999  10:36a               7,173 muisetup.hlp
07/21/1999  11:20a      <DIR>          nl.mui
07/21/1999  11:48a      <DIR>          sv.mui
               3 File(s)         63,279 bytes
              12 Dir(s)               0 bytes free

E:\i386>muisetup.exe -i 0411 0409 0c0a -d 0411 -u 0414 040c_
```

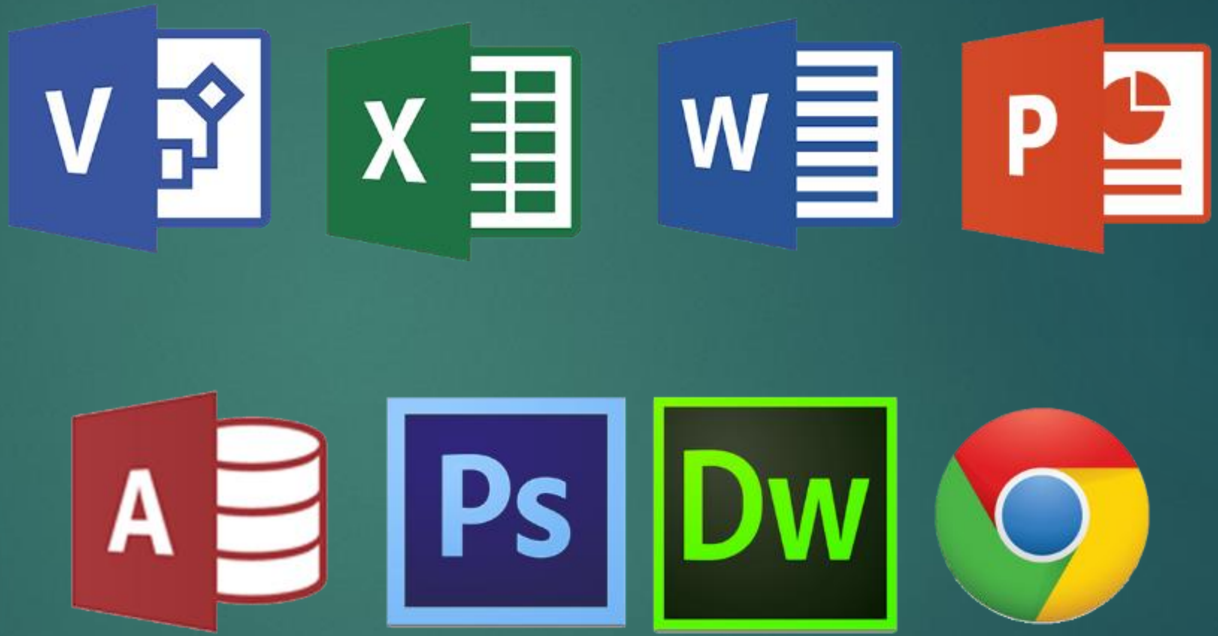
Command Line Interface



Graphical User Interface - GUI

Software

- Application Software



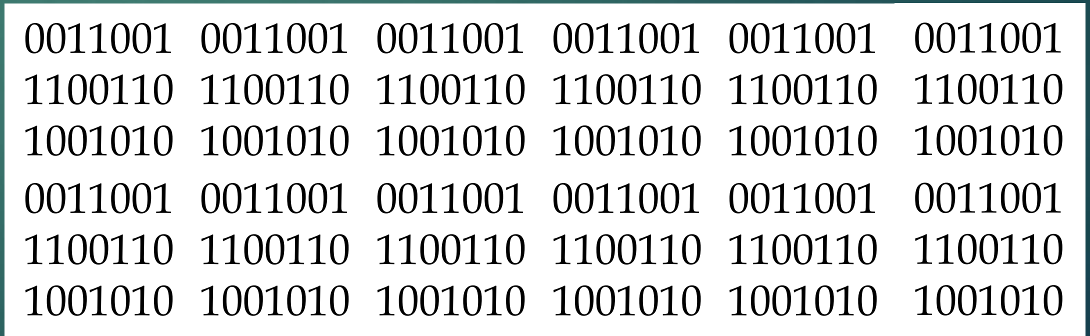
Software

- Programming Languages
  - Software is written in a programming language
  - Artificial language that tells a computer what to do
  - Four Generations

Software

1GL - Machine Language

- Sets of 0s and 1s that directly correspond to the machine’s electrical states
- Understood by machines; very difficult for human operators
- Machine specific - Each machine has its own language

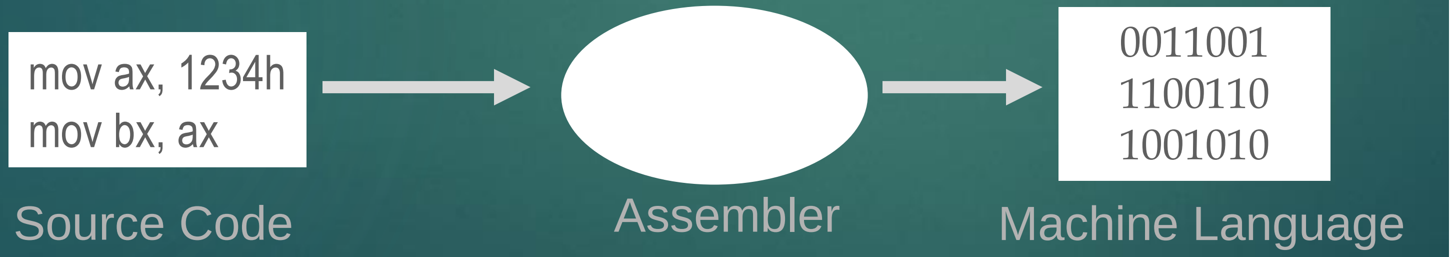


```
0011001  0011001  0011001  0011001  0011001  0011001
1100110  1100110  1100110  1100110  1100110  1100110
1001010  1001010  1001010  1001010  1001010  1001010
0011001  0011001  0011001  0011001  0011001  0011001
1100110  1100110  1100110  1100110  1100110  1100110
1001010  1001010  1001010  1001010  1001010  1001010
```

Software

2GL - Assembly Language

- Programmer uses not binary numbers but brief abbreviations for commands
- These abbreviations are called “Mnemonics”
- Assembler = translates assembly source code into machine language



```
mov ax, 1234h
mov bx, ax
```

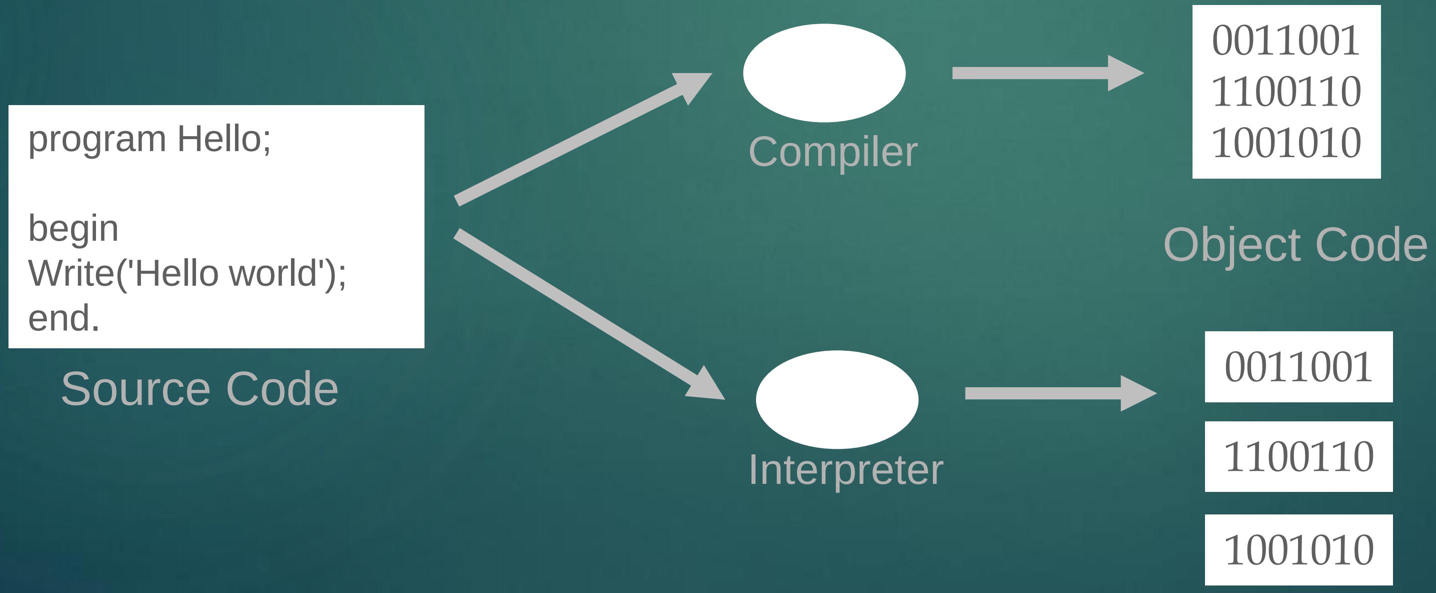
```
0011001
1100110
1001010
```

```
/* Hello world for IA64 (Itanium) Assembly
*/

.HW:
        stringz "Hello World"
        .text
        .align 16
        .global main#
        .proc main#
main:
        .prologue 14, 32
        .save ar.pfs, r33
        alloc r33 = ar.pfs, 0, 4, 1, 0
        .vframe r34
        mov r34 = r12
        adds r12 = -16, r12
        mov r35 = r1
        .save rp, r32
        mov r32 = b0
        .body
        addl r14 = @ltoffx(.HW), r1
        ;;
        ld8.mov r14 = [r14], .HW
        ;;
        st8 [r34] = r14
        ld8 r36 = [r34]
        br.call.sptk.many b0 = puts#
        mov r1 = r35
        ;;
        mov ar.pfs = r33
        mov b0 = r32
        .restore sp
        mov r12 = r34
        br.ret.sptk.many b0
```

Software

3GL - High-Level Language

- Use common words instead of mnemonics
- Source code translated into machine language – two methods



```
program Hello;

begin
Write('Hello world');
end.
```

Source Code

```
0011001
1100110
1001010
```

Object Code

```
0011001
```

```
1100110
```

```
1001010
```

```
program HelloWorld;

begin
  writeln('Hello World');
end.
```

Pascal

```
#include <stdio.h>

int main() {
  printf("Hello World!\n");
  return 0;
}
```

C

Software

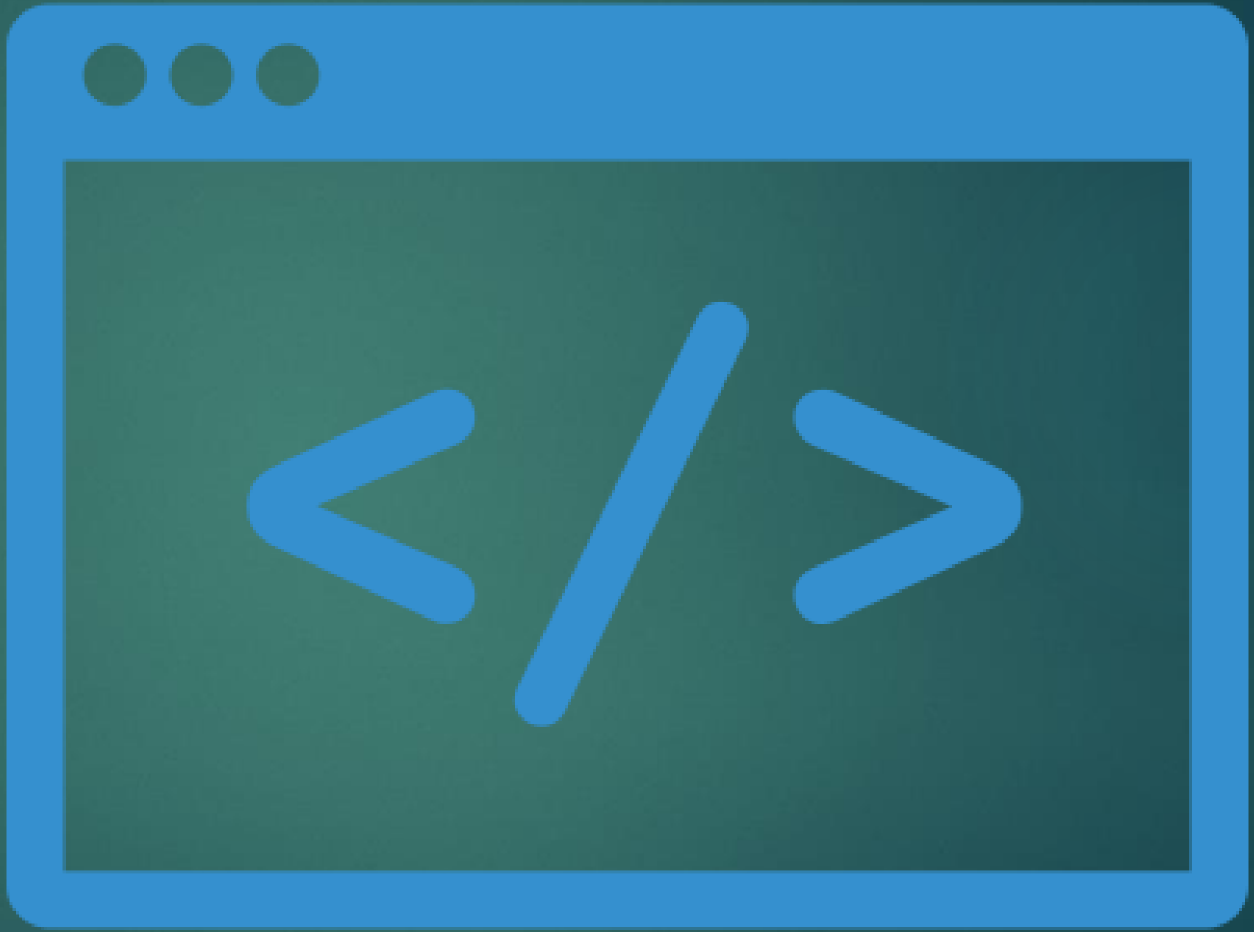
4GL – Object Oriented Programming

- Objects (nouns)
  - document
  - window
- Methods (verbs)
  - write( )
  - open( )

```
document.write(“Hello World”);
```

Javascript

Code Exercise

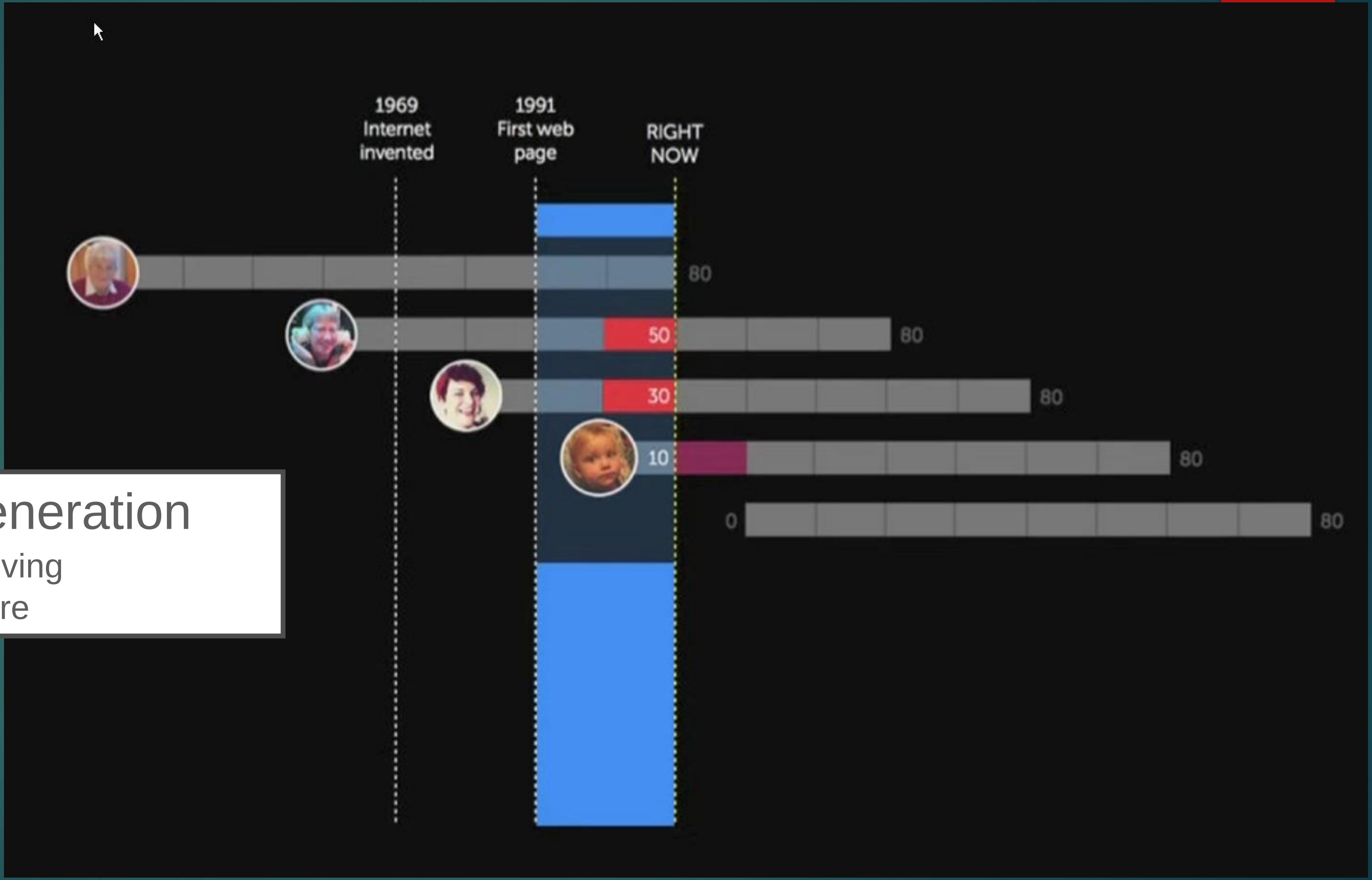


Importance of teaching children to write code

Global Facts

- Scotland – Begin programming at age 3
- Estonia – National educational objective

Hacking the future: Clare Sutcliffe at TEDxBrighton



Code Illiterate Generation

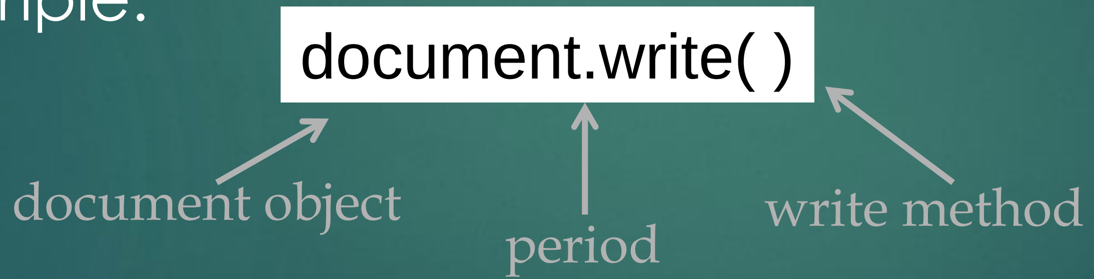
- Problem that needs solving
- Opportunity for the future

Code Exercise

- Task / Exercise
  - Hands-on experience with programming
  - JavaScript
    - Brief introduction
    - Exercises

Code Exercise

- JavaScript
  - Object Oriented Programming – OOP
    - **Objects** / **Method** = “nouns” / “verbs”
    - **Objects** and **Methods** are put together using “dot syntax” – the grammar of JavaScript
  - Example:

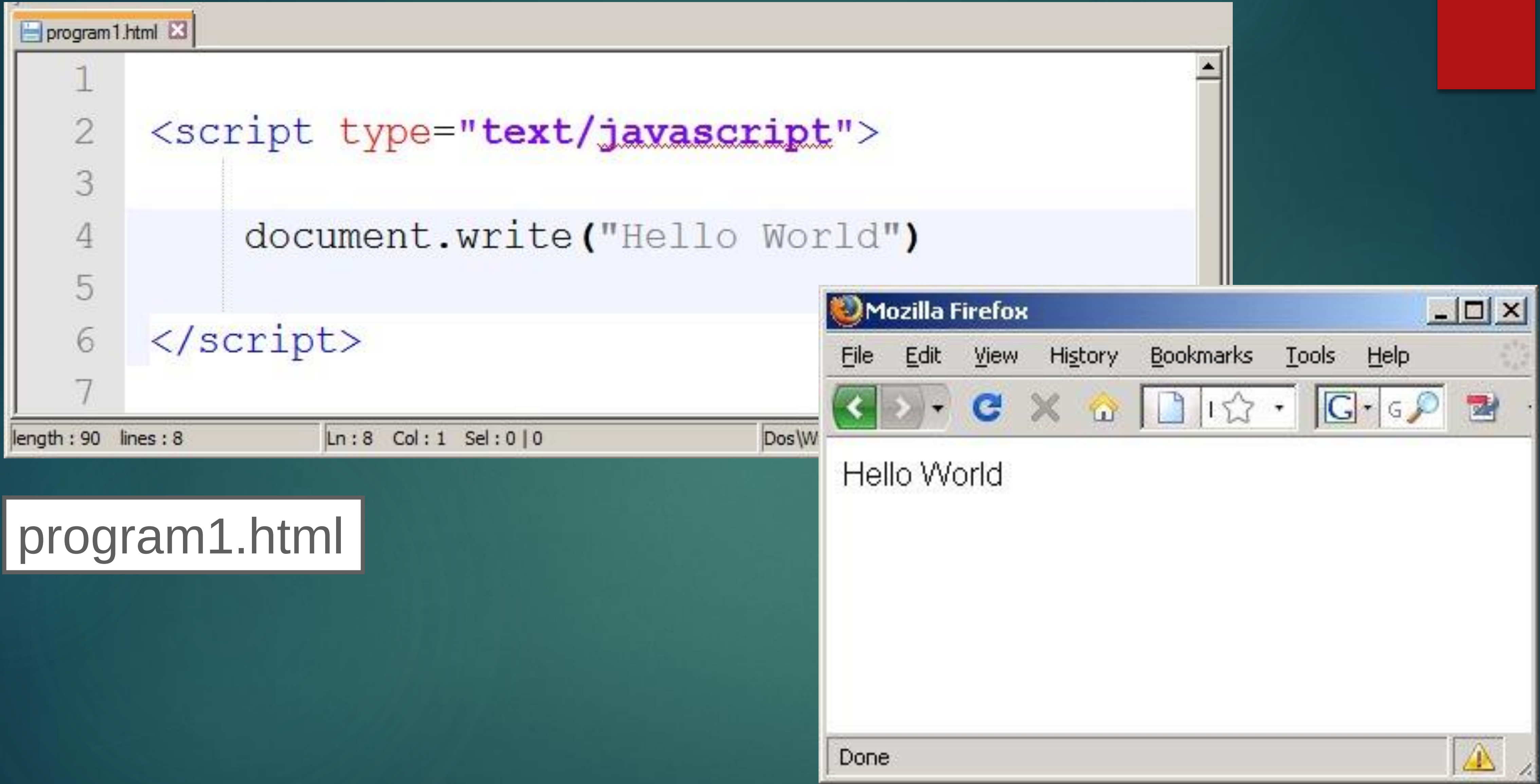


```
document.write( )
```

document object

period

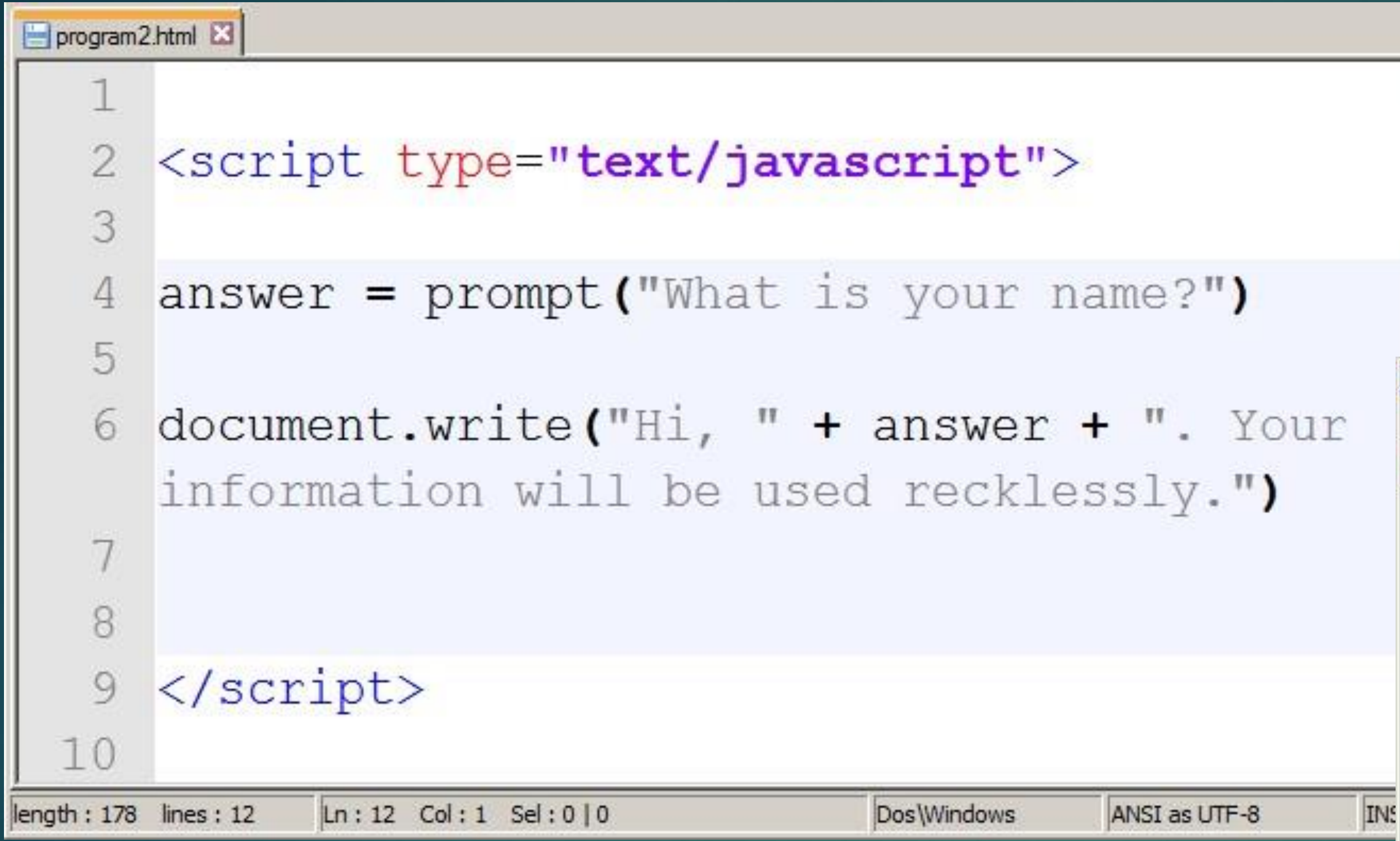
write method



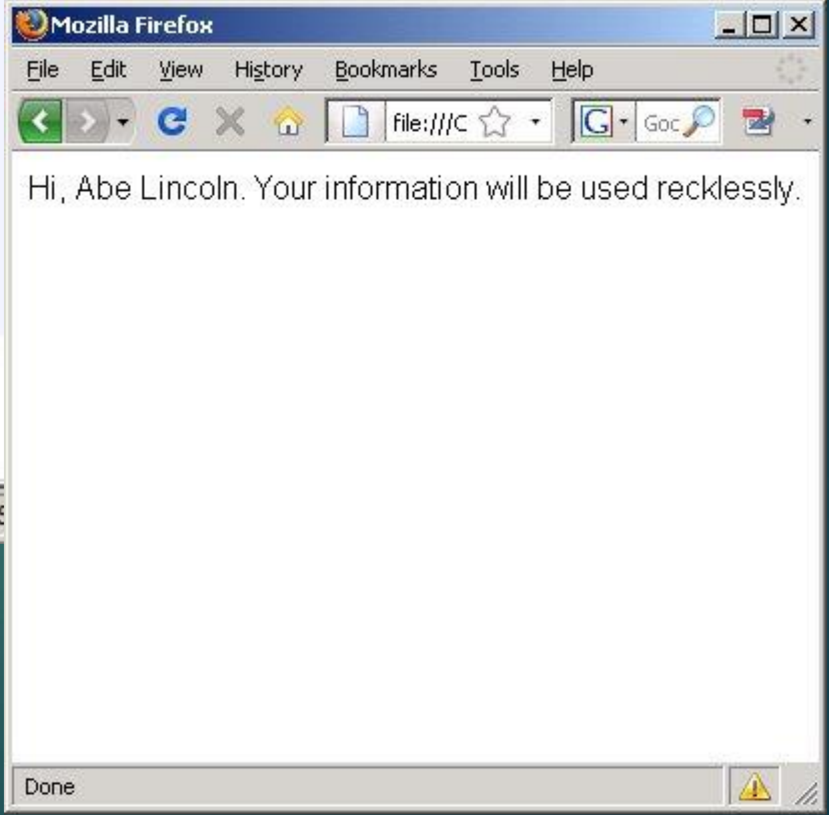
```html

<script type="text/javascript">

    document.write("Hello World")

</script>

```

```
Hello World
```

program1.html

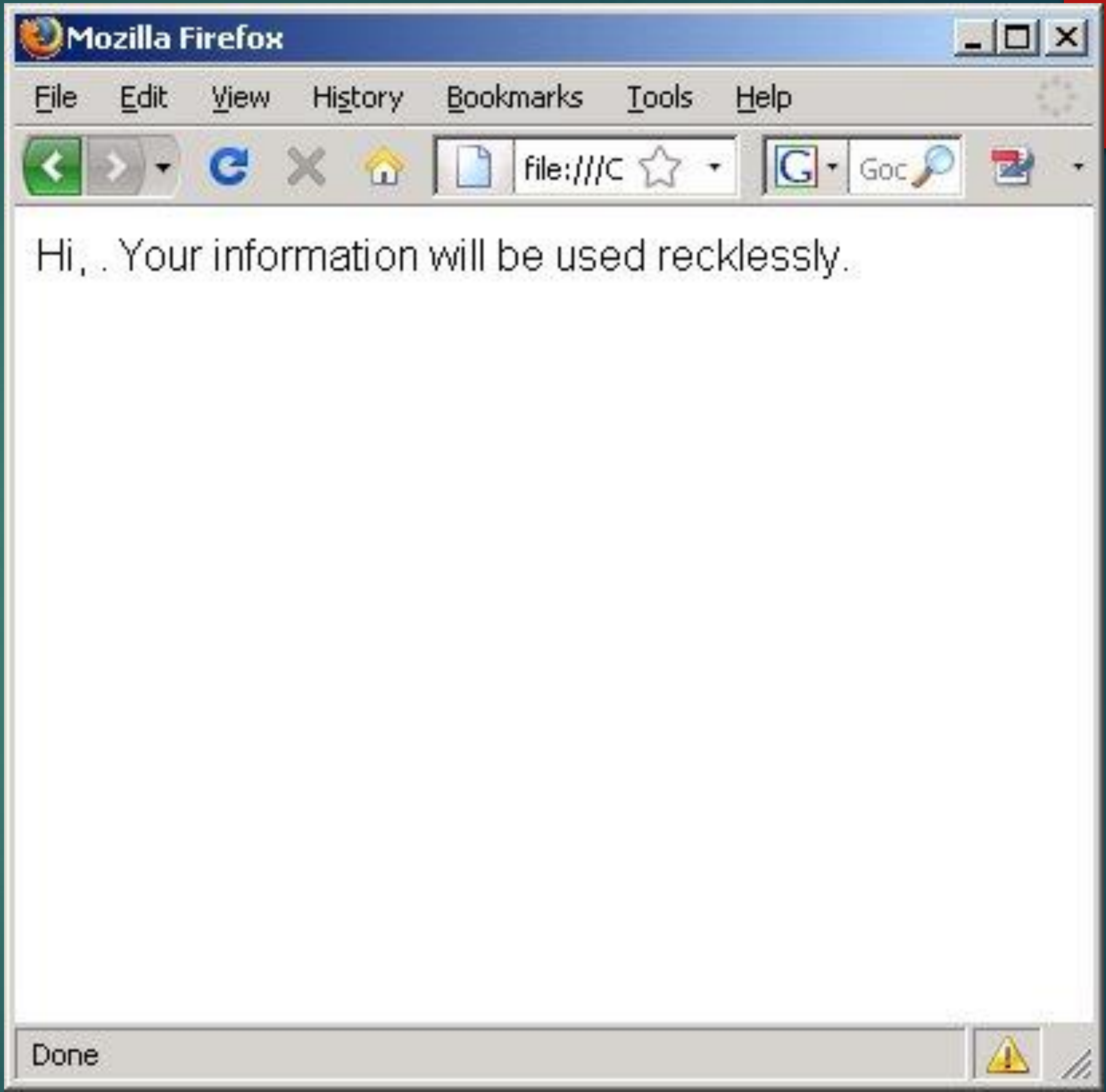


```

<script type="text/javascript">

answer = prompt("What is your name?")

document.write("Hi, " + answer + ". Your information will be used recklessly.")


</script>

```



```
Hi, Abe Lincoln. Your information will be used recklessly.
```

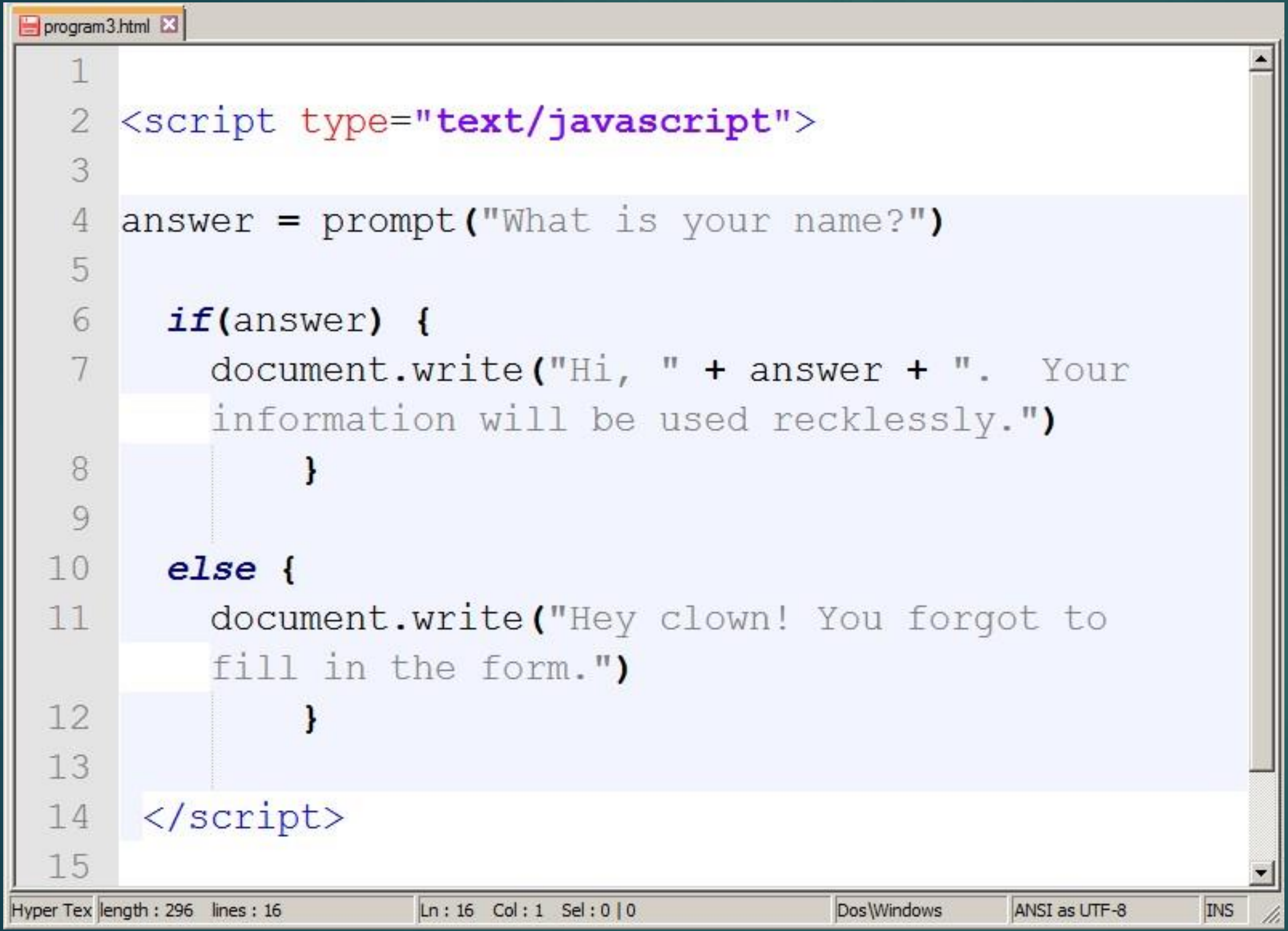
program2.html



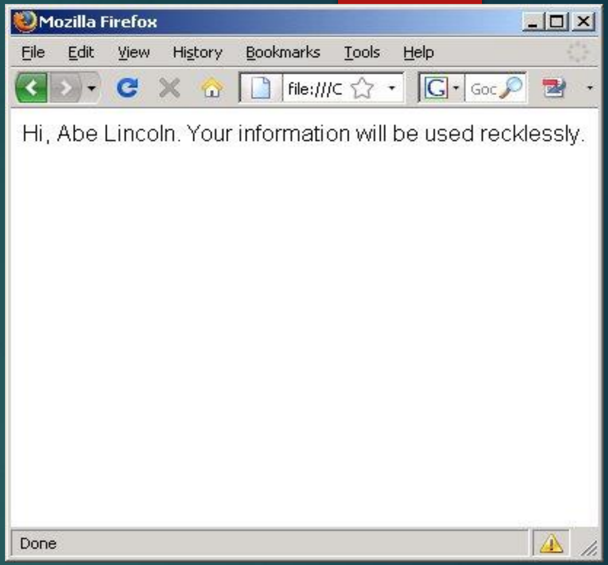
Hi, . Your information will be used recklessly.

Potential Problem

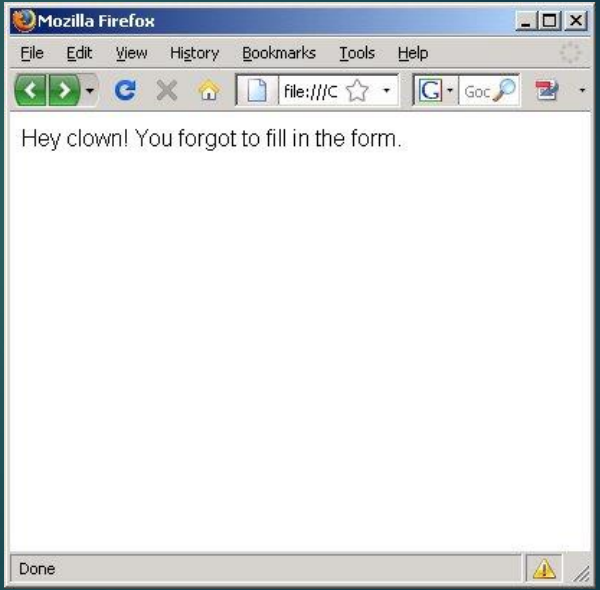
Fix = conditional statement



```

<script type="text/javascript">

answer = prompt("What is your name?")

  if(answer) {
    document.write("Hi, " + answer + ".  Your information will be used recklessly.")
        }

  else {
    document.write("Hey clown! You forgot to fill in the form.")
        }

 </script>

```

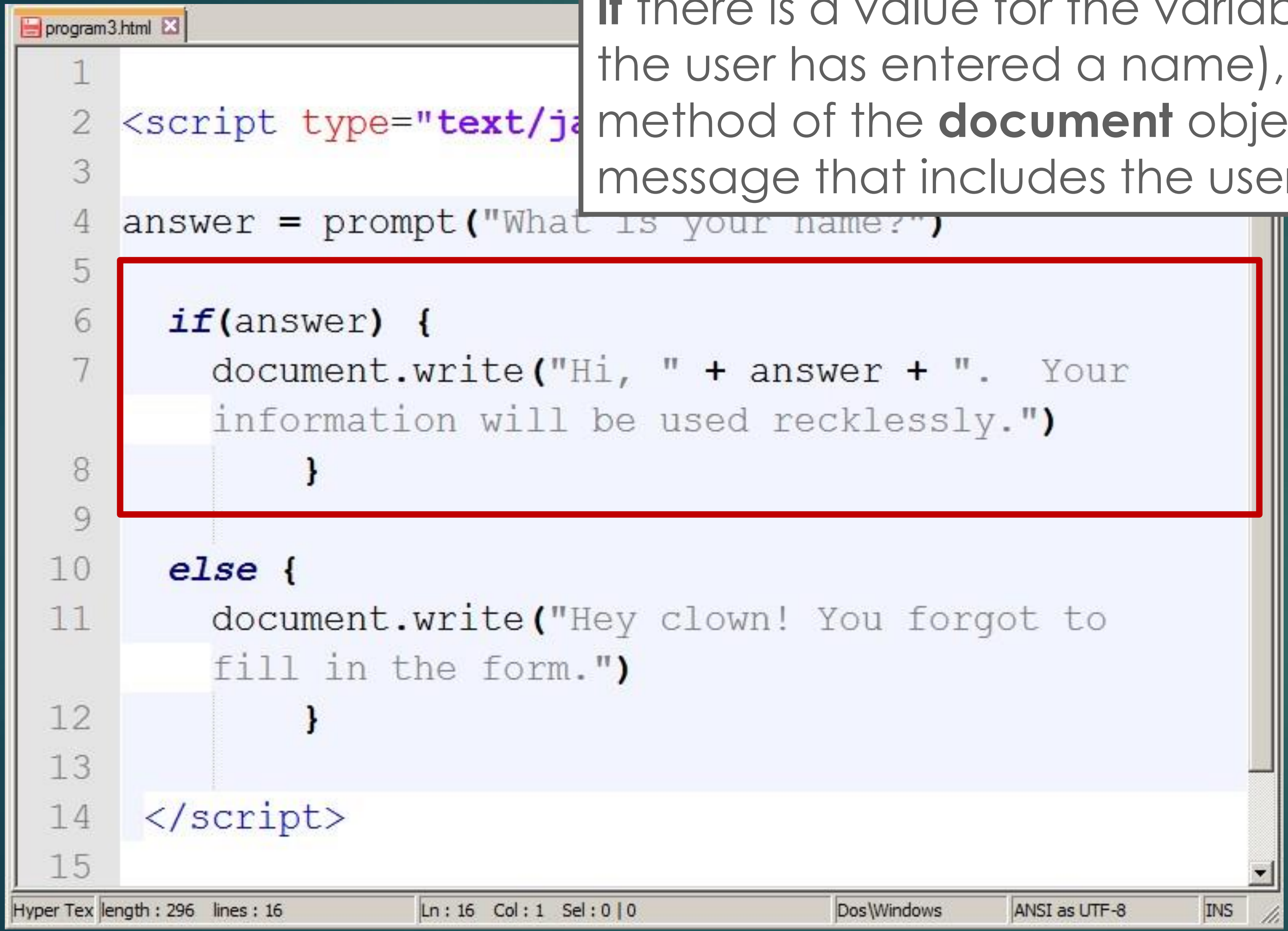


```
Hi, Abe Lincoln. Your information will be used recklessly.
```

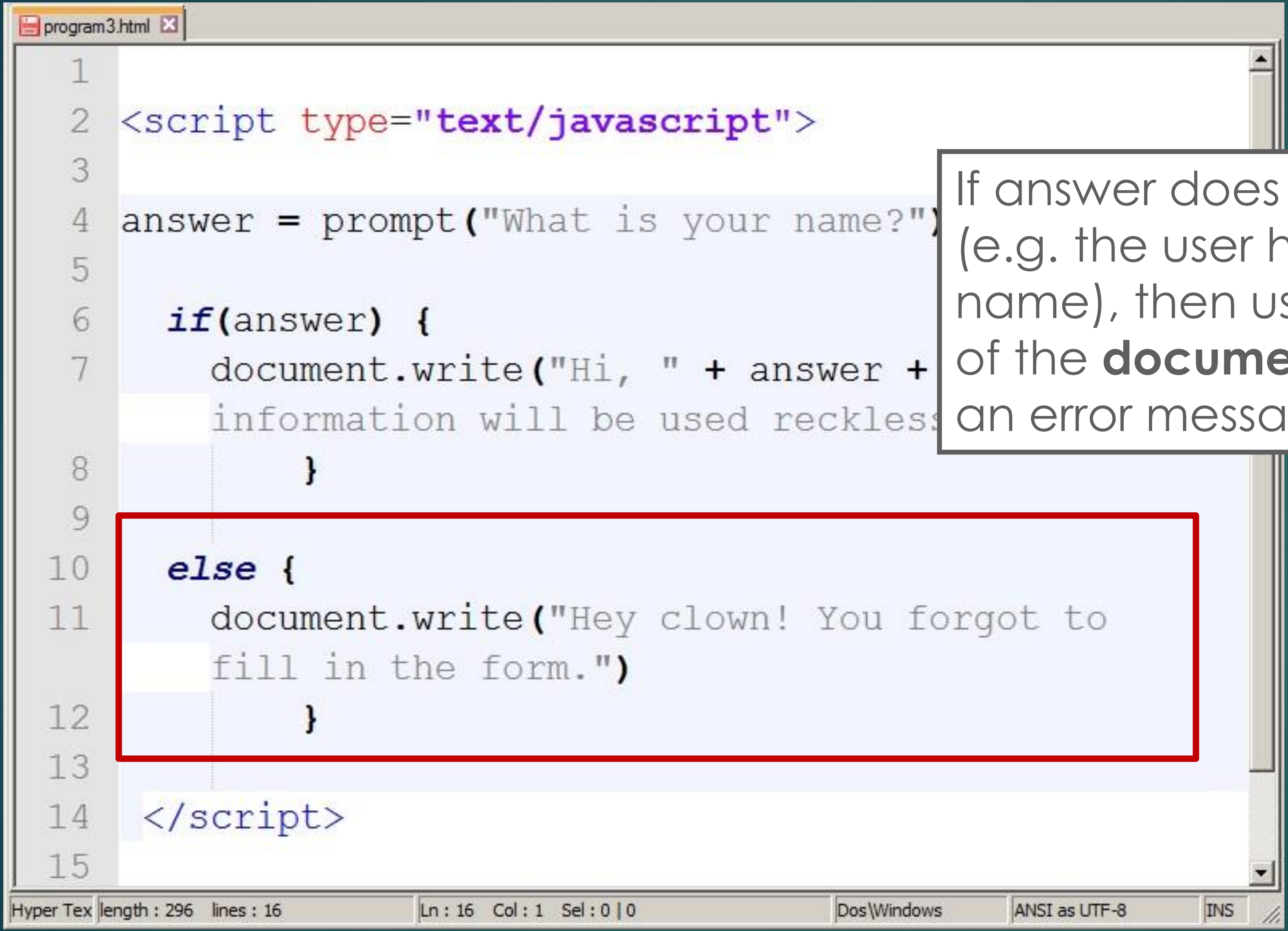


```
Hey clown! You forgot to fill in the form.
```

Include a conditional statement.
If there is a value for the variable **answer** (e.g. the user has entered a name), then use the **write** method of the **document** object to display a message that includes the user name.



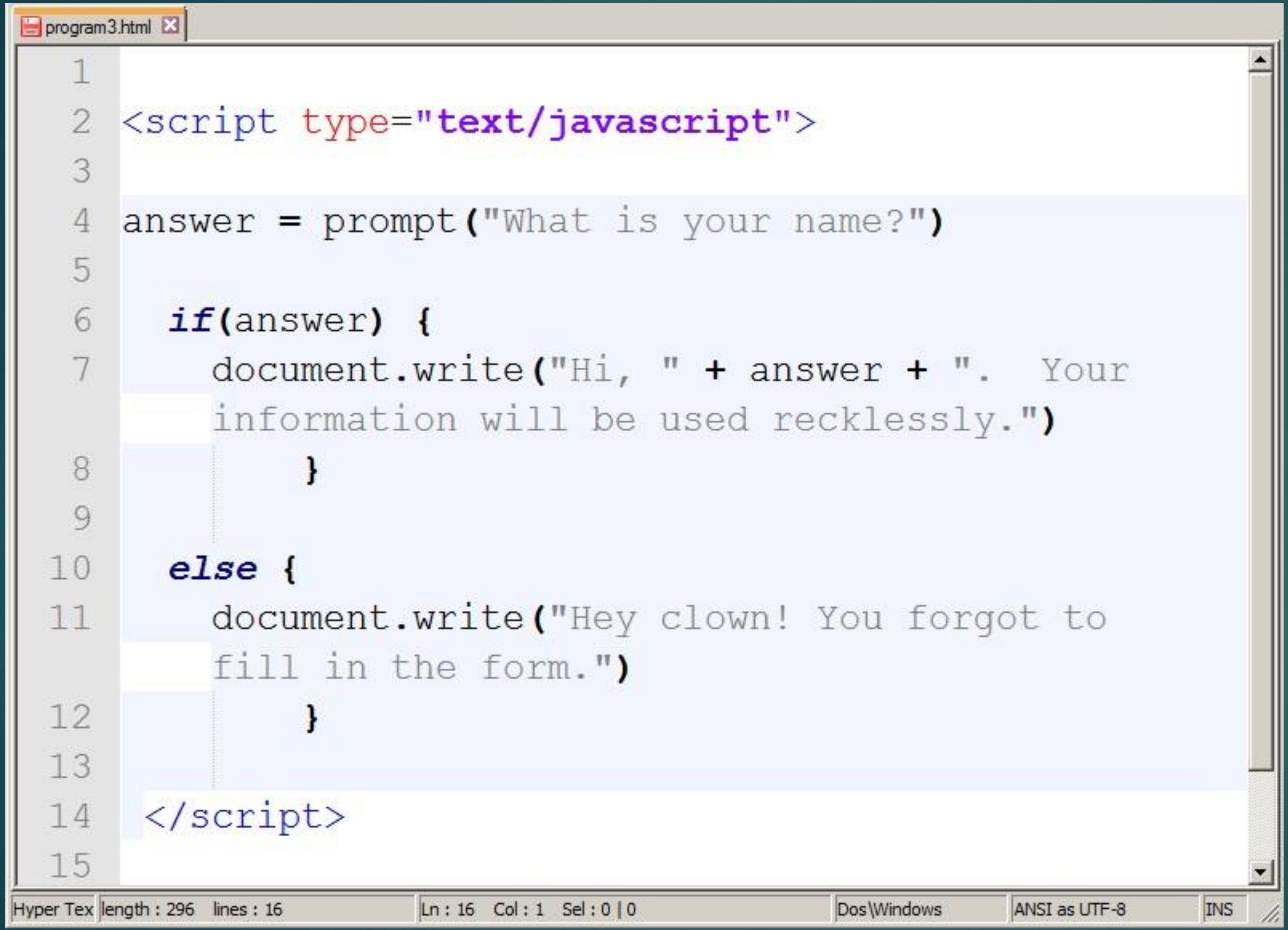
```

<script type="text/ja

answer = prompt("What is your name?")

  if(answer) {
    document.write("Hi, " + answer + ".  Your information will be used recklessly.")
        }

  else {
    document.write("Hey clown! You forgot to fill in the form.")
        }

 </script>

```



```

<script type="text/javascript">

answer = prompt("What is your name?")

  if(answer) {
    document.write("Hi, " + answer +
    information will be used reckless
        }

  else {
    document.write("Hey clown! You forgot to
    fill in the form.")
        }

 </script>

```

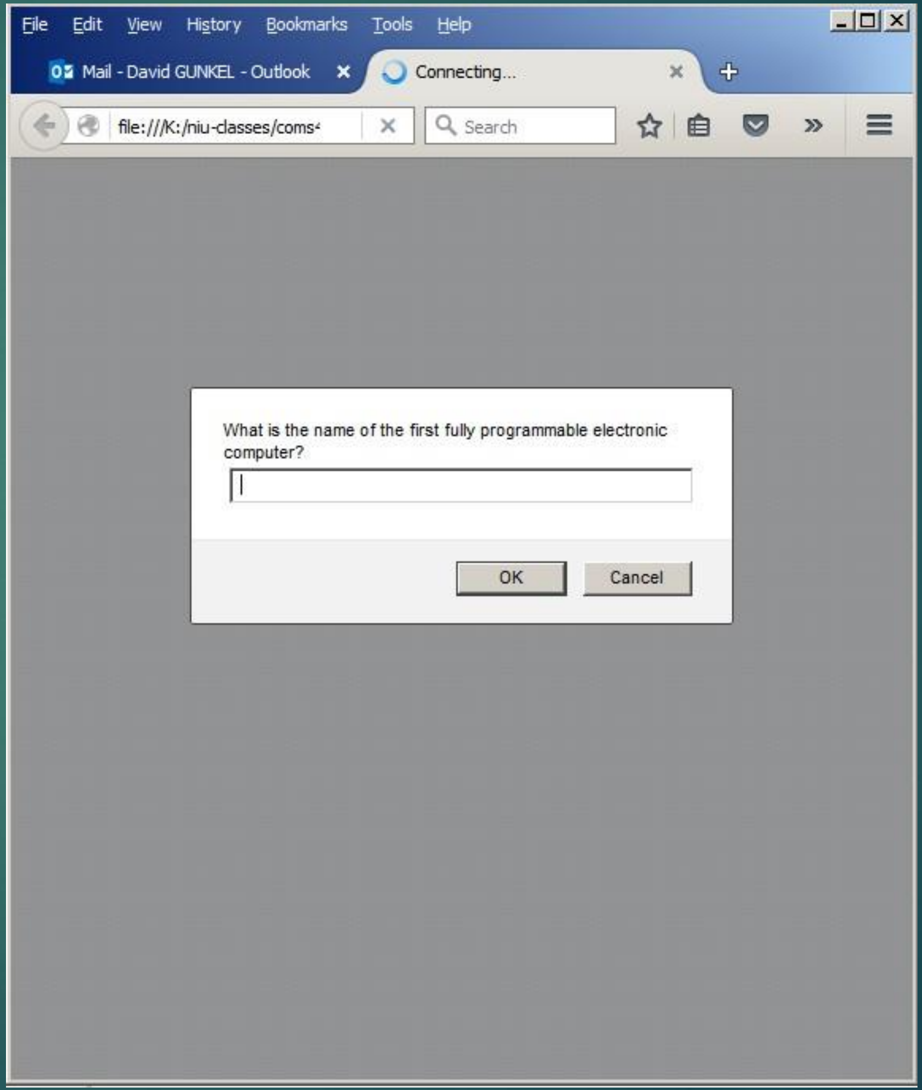
If answer does not have a value (e.g. the user has NOT entered a name), then use the **write** method of the **document** object to display an error message.

program3.html



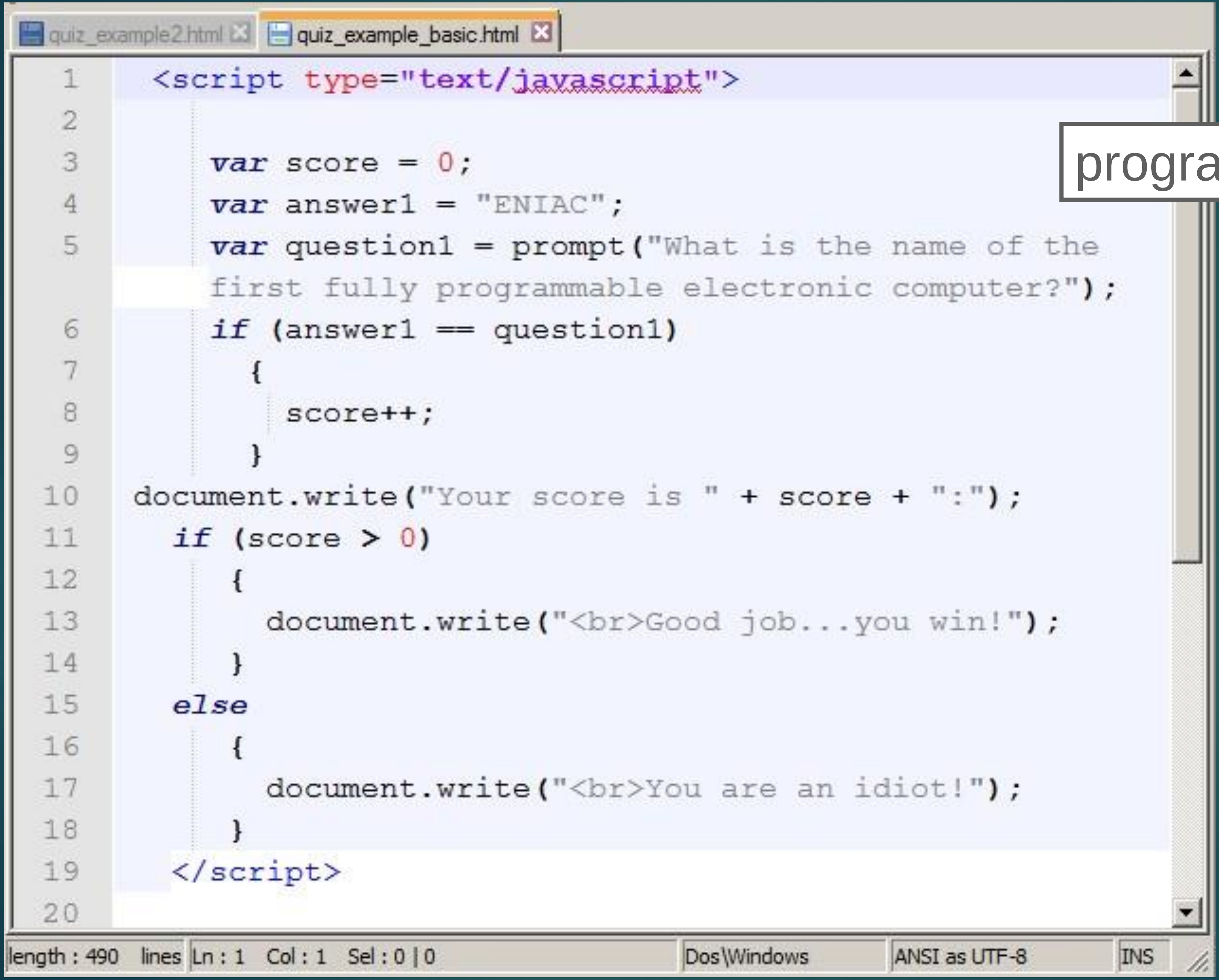
```

<script type="text/javascript">

answer = prompt("What is your name?")

  if(answer) {
    document.write("Hi, " + answer + ".  Your information will be used recklessly.")
        }

  else {
    document.write("Hey clown! You forgot to fill in the form.")
        }

 </script>

```

Trivia Quiz



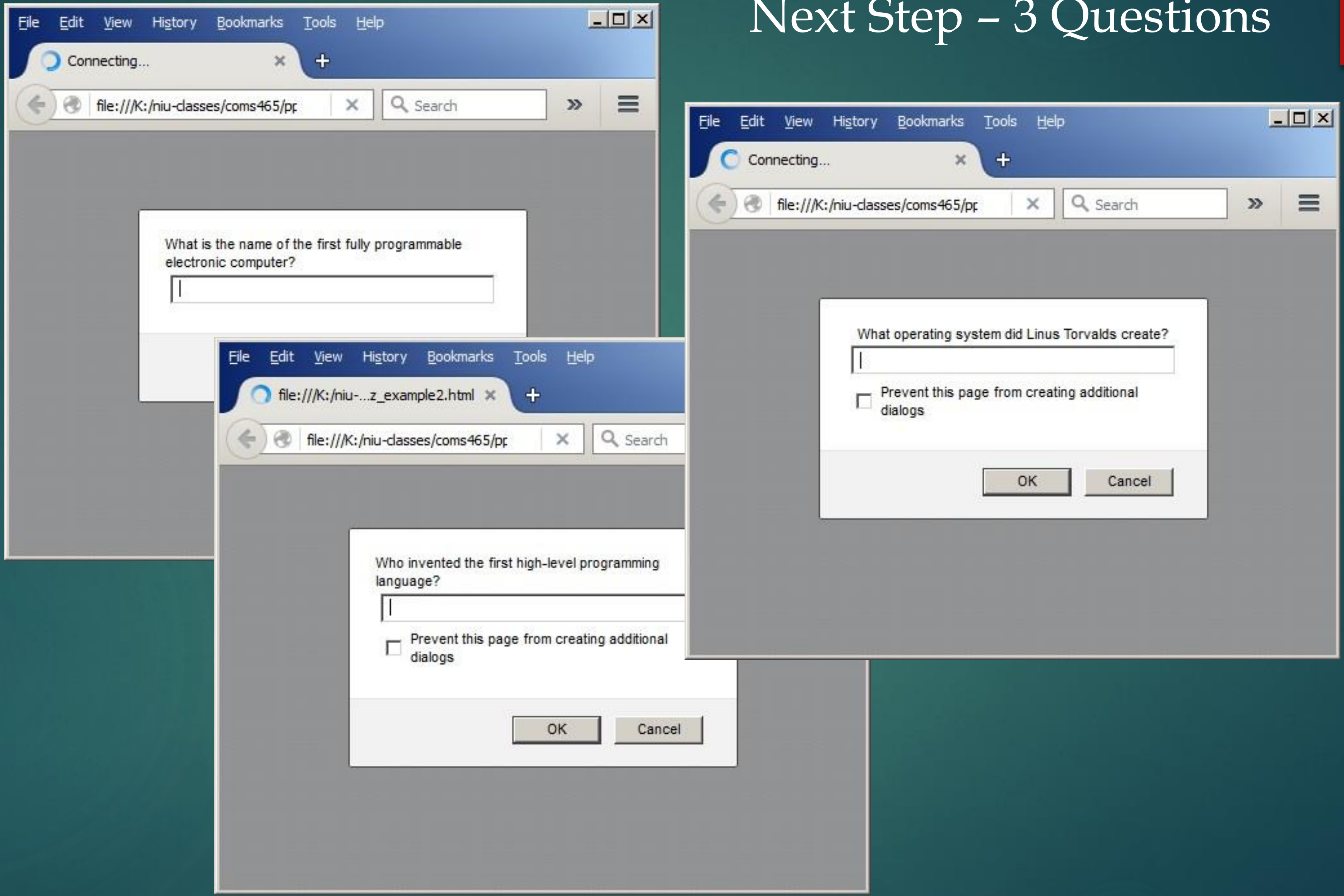
What is the name of the first fully programmable electronic computer?

program4.html



```
 <script type="text/javascript">

  var score = 0;
  var answer1 = "ENIAC";
  var question1 = prompt("What is the name of the first fully programmable electronic computer?");
  if (answer1 == question1)
   {
    score++;
   }
document.write("Your score is " + score + ":");
 if (score > 0)
   {
    document.write("<br>Good job...you win!");
   }
 else
   {
    document.write("<br>You are an idiot!");
   }
 </script>
```

Next Step – 3 Questions



File Edit View History Bookmarks Tools Help
Connecting...
file:///K:/niu-classes/coms465/pr
Search

What is the name of the first fully programmable electronic computer?

File Edit View History Bookmarks Tools Help
file:///K:/niu-...z_example2.html
file:///K:/niu-classes/coms465/pr
Search

Who invented the first high-level programming language?

□ Prevent this page from creating additional dialogs

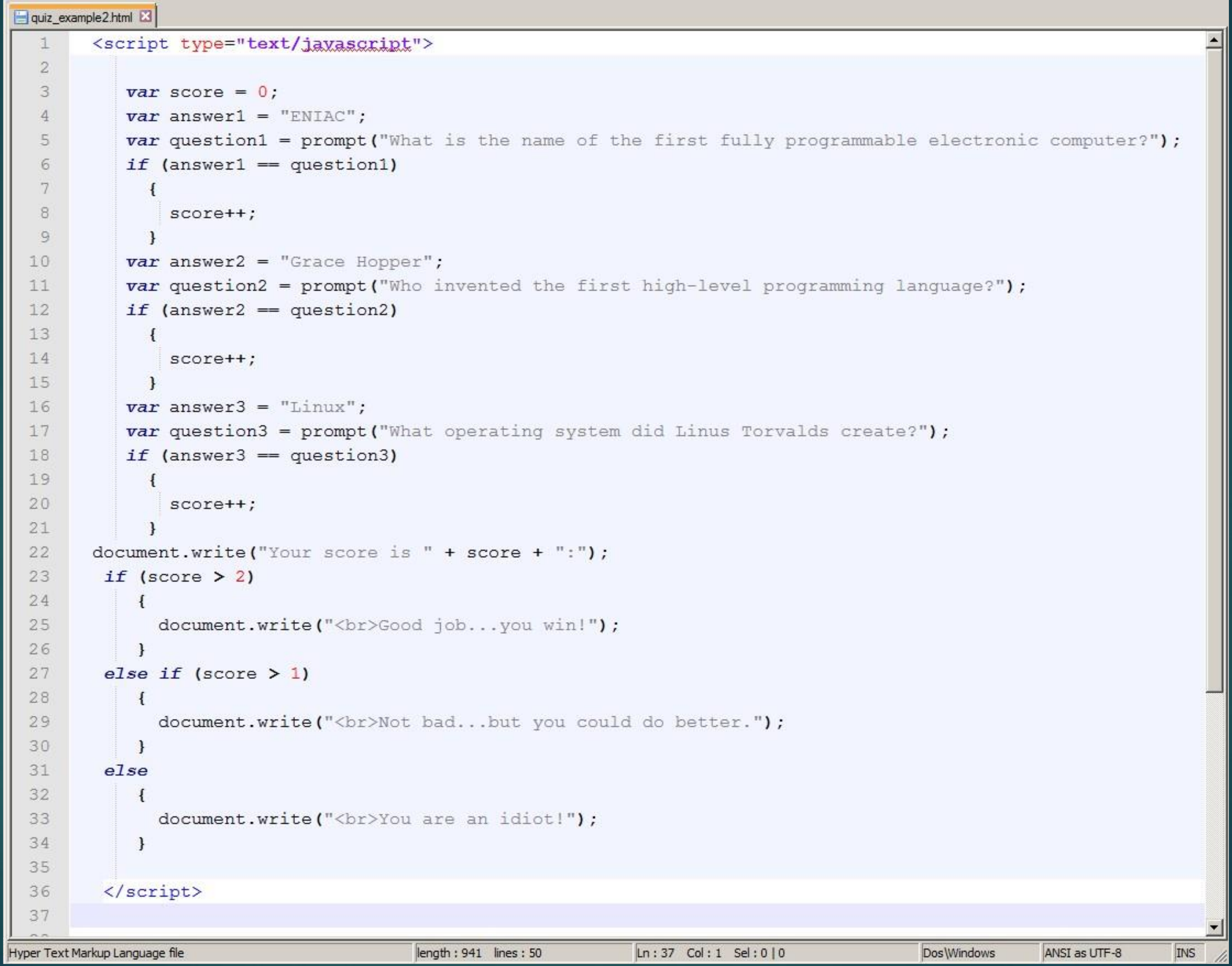
OK Cancel

File Edit View History Bookmarks Tools Help
Connecting...
file:///K:/niu-classes/coms465/pr
Search

What operating system did Linus Torvalds create?

□ Prevent this page from creating additional dialogs

OK Cancel



```
<script type="text/javascript">

   var score = 0;
   var answer1 = "ENIAC";
   var question1 = prompt("What is the name of the first fully programmable electronic computer?");
   if (answer1 == question1)
     {
       score++;
     }
   var answer2 = "Grace Hopper";
   var question2 = prompt("Who invented the first high-level programming language?");
   if (answer2 == question2)
     {
       score++;
     }
   var answer3 = "Linux";
   var question3 = prompt("What operating system did Linus Torvalds create?");
   if (answer3 == question3)
     {
       score++;
     }
document.write("Your score is " + score + ":");
 if (score > 2)
    {
      document.write("<br>Good job...you win!");
    }
 else if (score > 1)
    {
      document.write("<br>Not bad...but you could do better.");
    }
 else
    {
      document.write("<br>You are an idiot!");
    }

 </script>
```

Power User

Technology = Magic

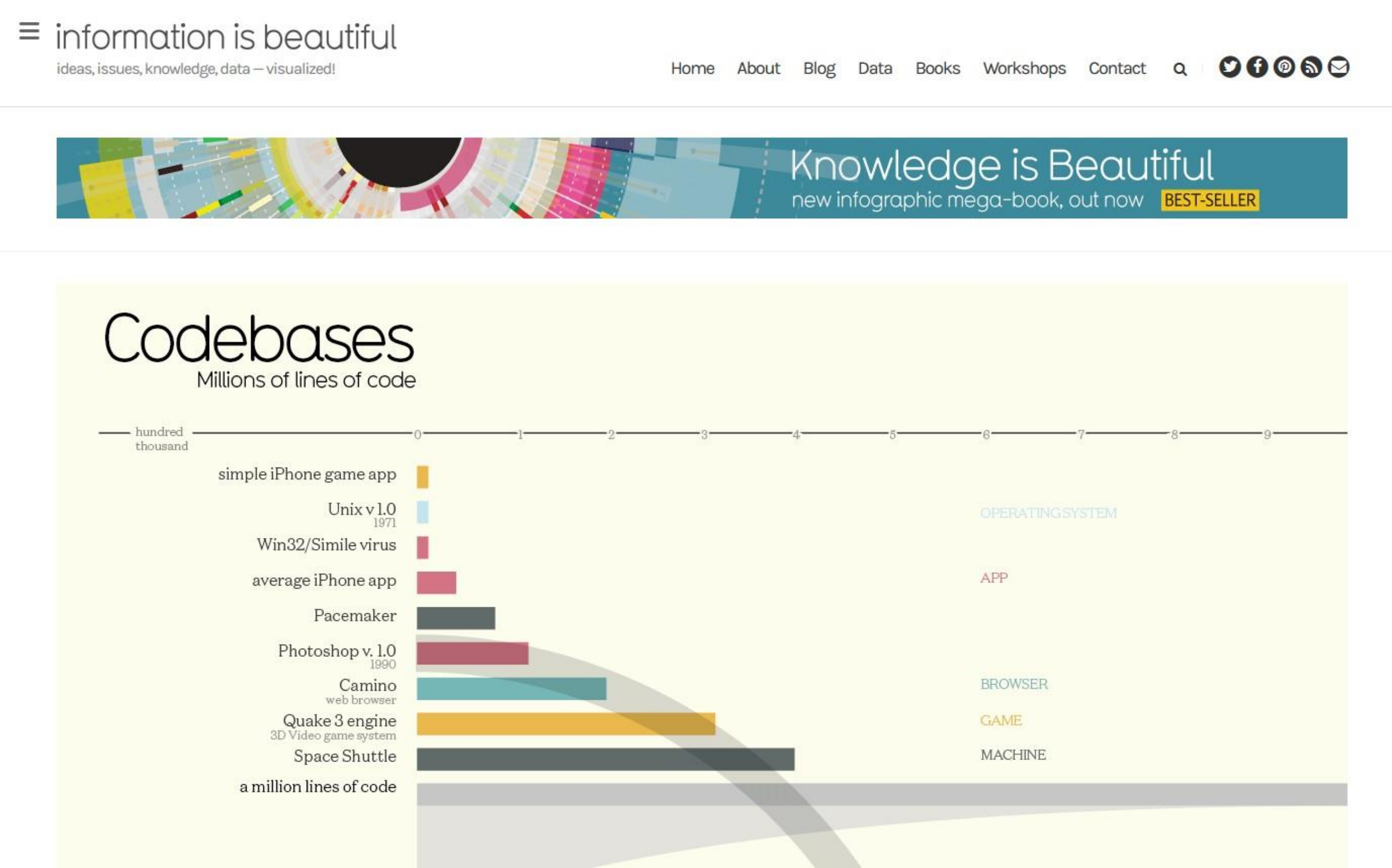
The tool is in control

Critical User

Technology = Tool

We control our tools

How many lines of code?



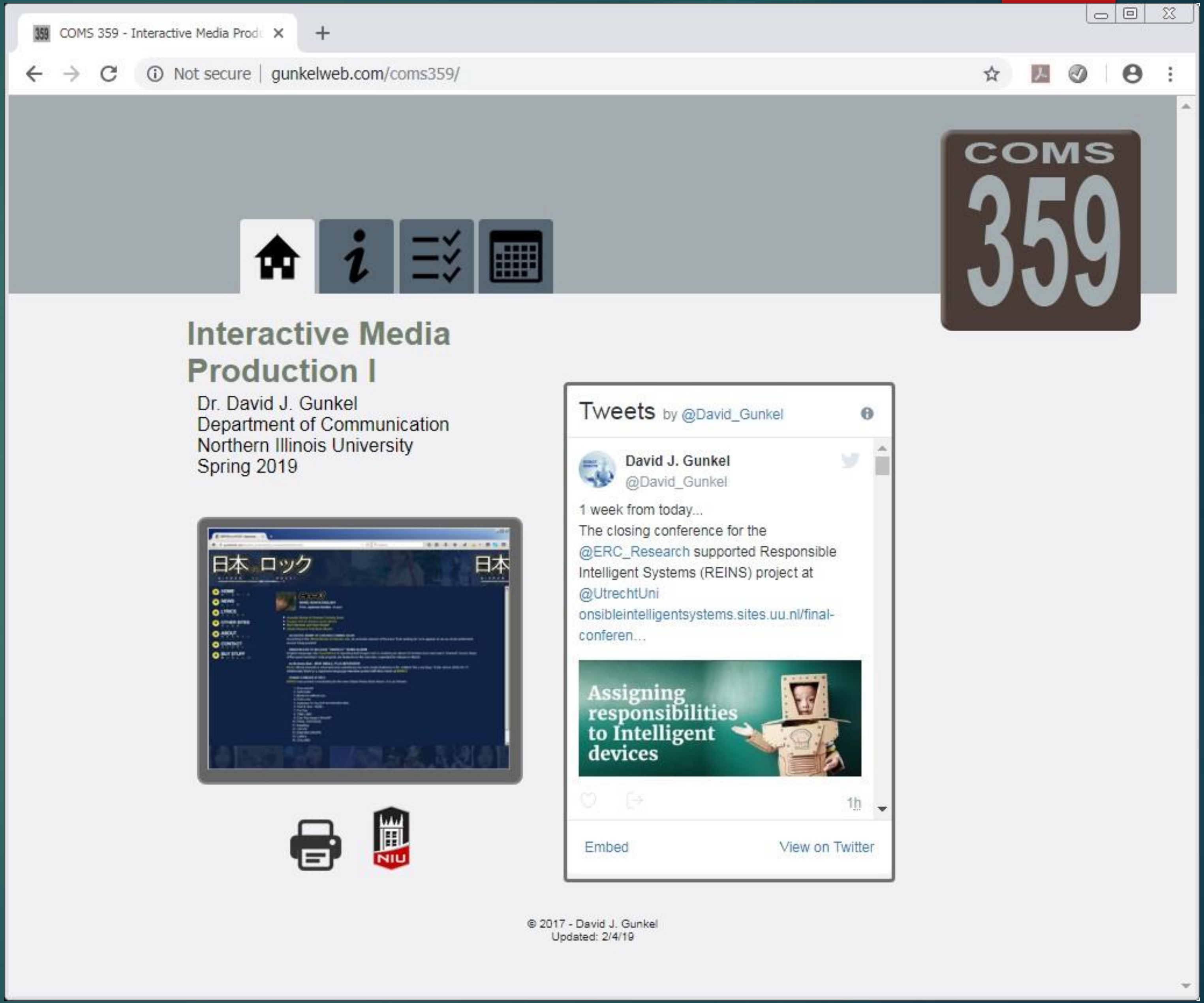
information is beautiful

ideas, issues, knowledge, data — visualized!

Codebases

Millions of lines of code

http://www.informationisbeautiful.net/visualizations/million-lines-of-code


COMS 359 - Interactive Media Prod
Not secure | gunkelweb.com/coms359/
COMS 359
日本 ロック 日本
NIU

Interactive Media Production I

Dr. David J. Gunkel
Department of Communication
Northern Illinois University
Spring 2019

Tweets by @David_Gunkel

David J. Gunkel
@David_Gunkel

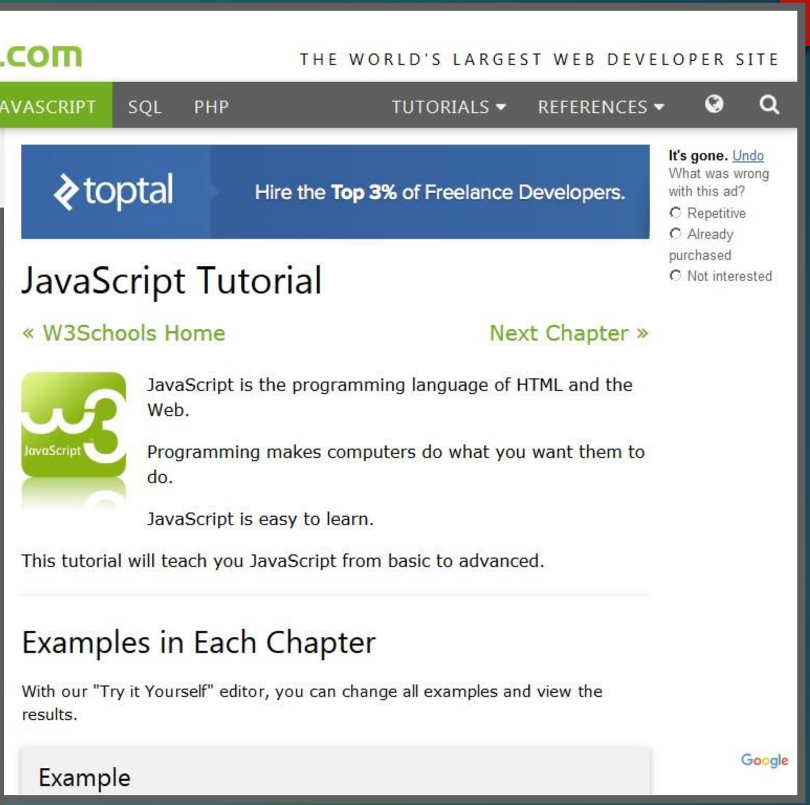
1 week from today...
The closing conference for the @ERC_Research supported Responsible Intelligent Systems (REINS) project at @UtrechtUni onsibleintelligentsystems.sites.uu.nl/final-conferen...

Assigning responsibilities to Intelligent devices

1h

Embed View on Twitter

© 2017 - David J. Gunkel
Updated: 2/4/19



w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

HTML CSS JAVASCRIPT SQL PHP TUTORIALS ▾ REFERENCES ▾

JS Array Methods

JS Booleans

JS Comparisons

JavaScript Tutorial

« W3Schools Home Next Chapter »

JavaScript is the programming language of HTML and the Web.

Programming makes computers do what you want them to do.

JavaScript is easy to learn.

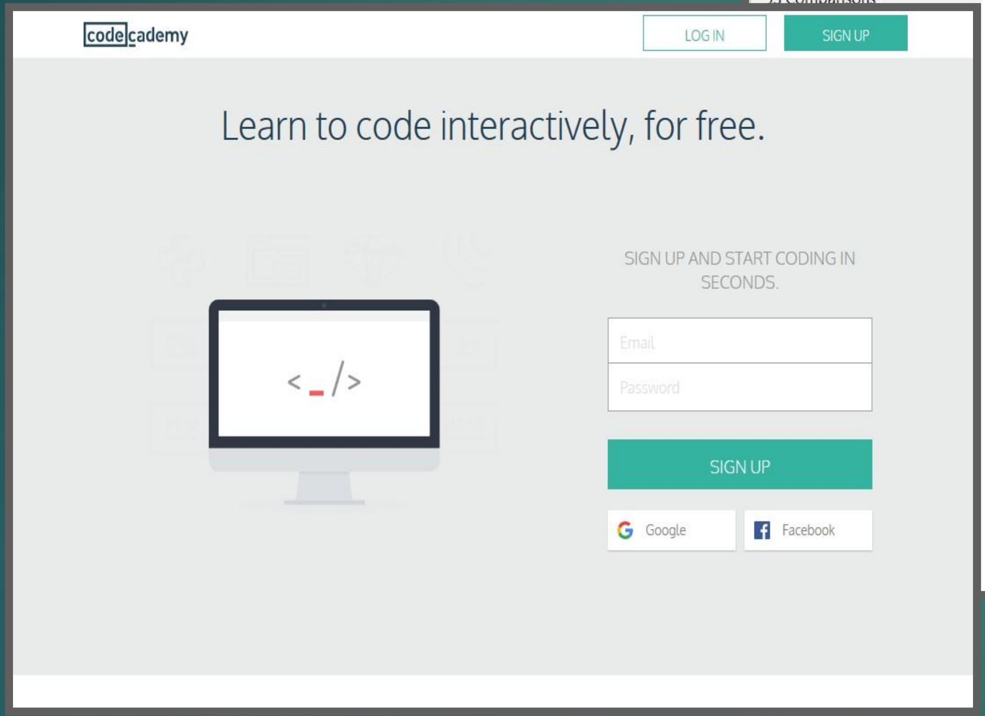
This tutorial will teach you JavaScript from basic to advanced.

Examples in Each Chapter

With our "Try it Yourself" editor, you can change all examples and view the results.

Example

http://www.w3schools.com



codecademy

LOG IN SIGN UP

Learn to code interactively, for free.

SIGN UP AND START CODING IN SECONDS.

Email

Password

SIGN UP

Google Facebook

https://www.codecademy.com

Preview

- 11-13 February
  - Robot/AI Conference - Utrecht
  - No class meeting
- 18 February
  - The Internet
  - Texts
    - Ceruzzi - Computing - ch. 6
    - Brief History of the Internet