EECS 368
Programming Language Paradigms

Dr. Andy Gill

Department of Electrical Engineering & Computer Science
University of Kansas

September 10, 2015

KU

Syntax for Languages

C++

```
#include <iostream.h>
main()
{
    cout << "Hello World!";
    return 0;
}
```

Java

```
public class HelloWorld {
    public static void main (String[] args) {
        System.out.println ("Hello World!");
    }
}
```

JavaScript

```
alert("Hello, World!");
```

Scheme

```
(display "Hello World!")
(newline)
```

Haskell

```
main :: IO ()
main = putStrLn "Hello World!"
```

KU

EBNF - Extended BNF

One or more of something

X+

Zero or more of something

X*

Optional elements

X?

Grouping

(...)

KU

Homework

```
S ::= A S | A
A ::= A S | B
B ::= C S D | E
```

- What are the terminals and non-terminals for this grammar? (In class)
- Show that this grammar is ambiguous by giving a token sequence that has two possible concrete syntax trees (two derivations)
- Construct an unambiguous grammar to describe the same language, using BNF.
- Construct another unambiguous grammar to describe the same language, using EBNF.

(Adapted from Fundamental Structures of Computer Science, Wulf, Shaw, Hilfinger, Flon, pp 370)

KU

Programming Languages

- For the sake of getting things done
- Let us look at some examples, in Haskell and JavaScript

KU

Haskell Sieve

```
import System.Environment

primes :: [Int]
primes = sieve [2..]
    where sieve (p:xs) = p : sieve [x | x<-xs, x `mod` p /= 0]

main = do
   [n] <- getArgs
   print (last (takeWhile (<= (read n)) primes))
```

Can you play this game?

Then you can understand JavaScript!

JavaScript Gameplan

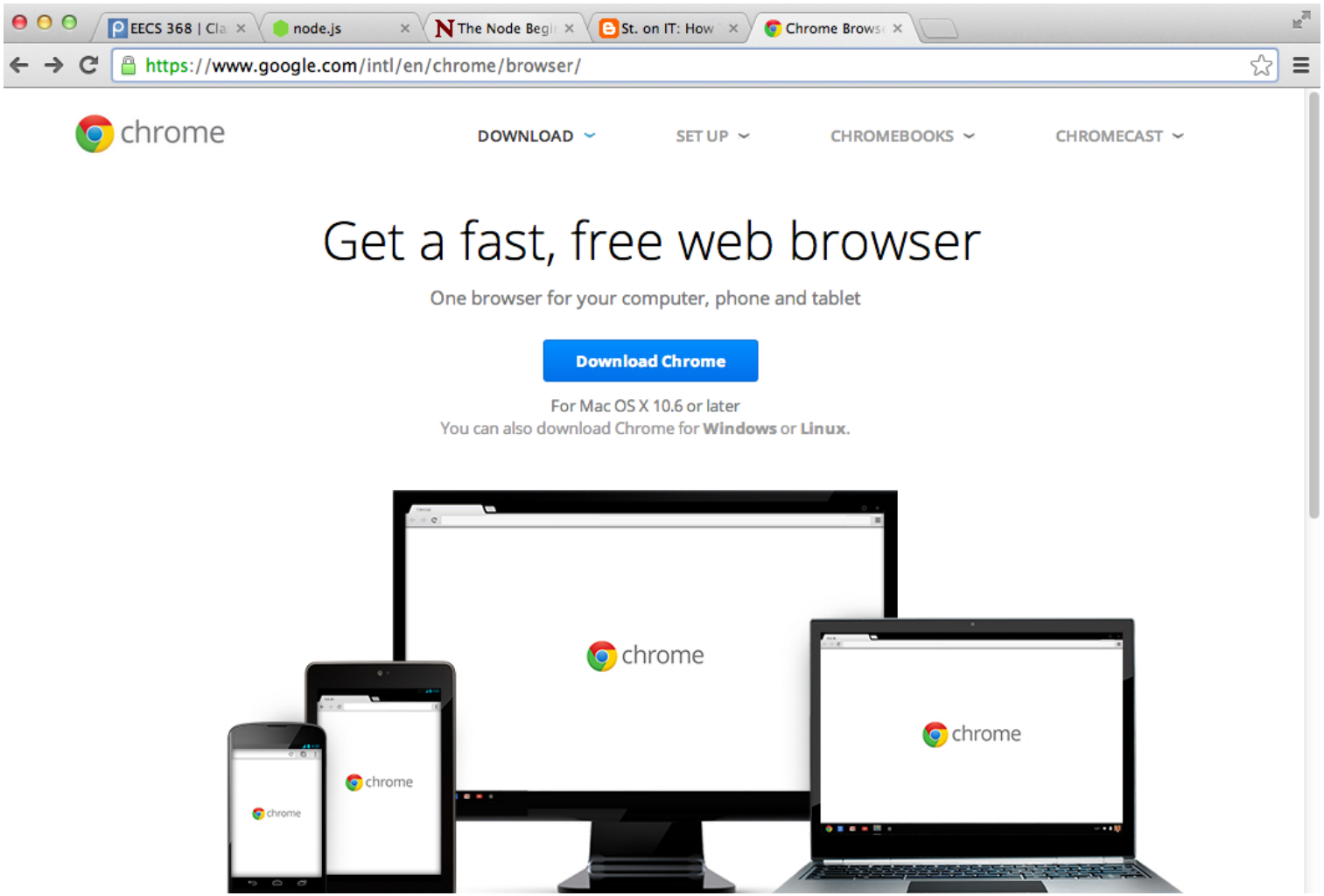
- We are going to work through some examples, for the sake of getting a feel for the language
- We will then step back (in future classes) and examine different important attributes of the JavaScript language.
- Topics will include
  - Syntactical Structure
  - Input/Output
  - Environment
  - Object Oriented Programming
  - Threads and Concurrency
  - Browser DOM and jquery
  - Collections

KU

Where does JavaScript fit into the computer universe?

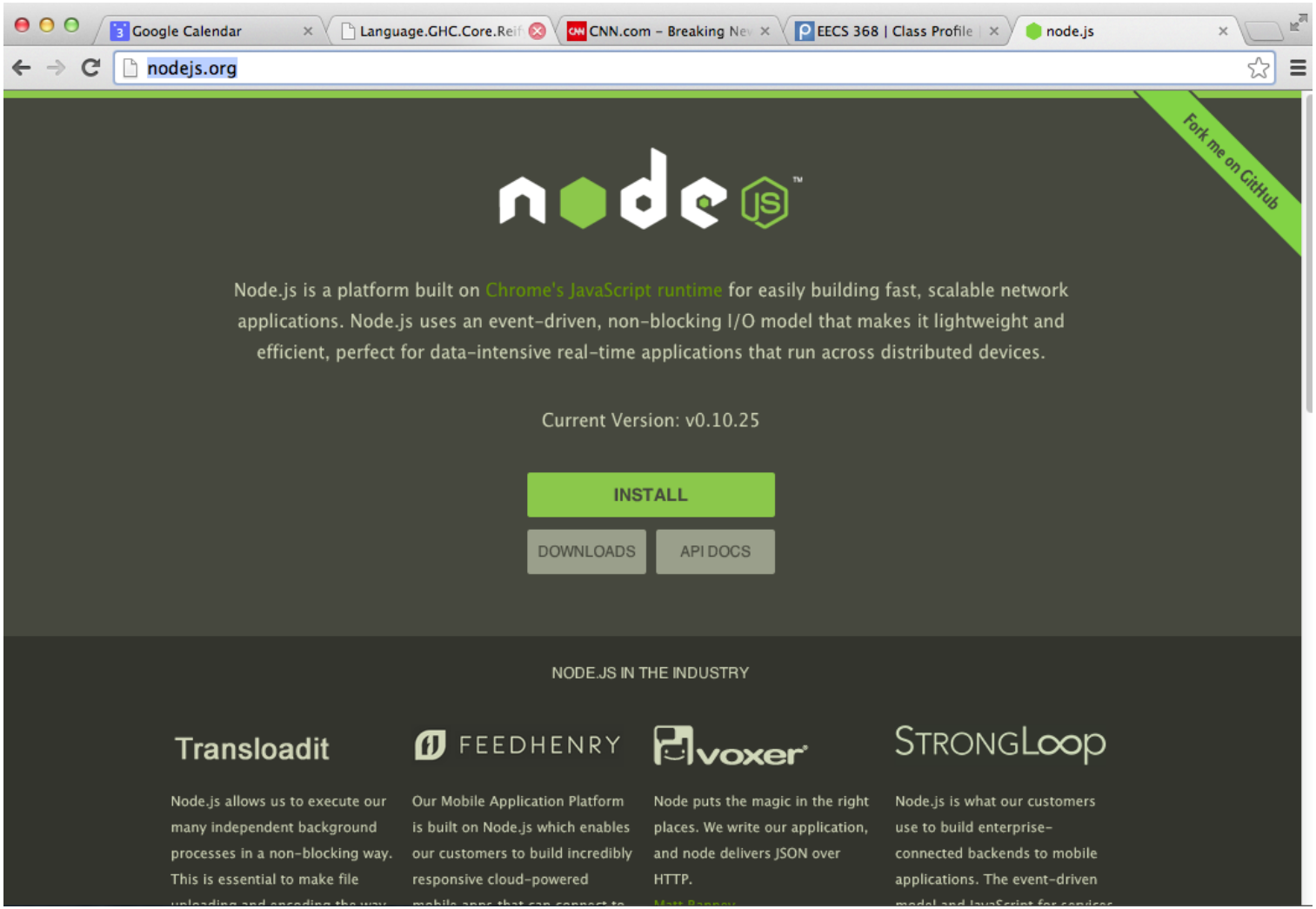
KU

JavaScript Interpreter : inside Chrome



JavaScript Interpreter : node.js

http://nodejs.org/



Your first JavaScript program in Chrome

index.html

```
<html>
  <script src="hello.js"></script>
</html>
```

hello.js

```
alert("Hello!");
```

Your first JavaScript program in node

world.js

```
console.log("Hello World!");
```

```
$ node world.js
Hello World!
$
```

JavaScript Stack

JavaScript

```
alert("Hello, World!");
```

Library

```
// Example: jQuery
function $(...) { ... }
```

Browser

- Can modify DOM (Document Object Model)
- Can request specific JavaScript be run when things happen, for example a mouse click.

KU

Gameplan for JavaScript

- We will be writing JavaScript, at first for the sake of understanding syntaxtical structure.
- I will provide a library to make JavaScript easier. Eventually, you will understand the whole library. This is a platform to stand on, for now.
- We will use node.js and the browser, as necessary.

KU

Your first JavaScript program in Chrome

index.html

```
<html>
  <script src="hello.js"></script>
</html>
```

hello.js

```
alert("Hello!");
```

Your first JavaScript program in node

world.js

```
console.log("Hello World!");
```

```
$ node world.js
Hello World!
$
```

Your first JavaScript program in Chalk

index.html

```
<html>
  <body>
  </body>
  <div id="chalk"></div>
  <script src="http://code.jquery.com/jquery-1.10.2.min.js"></scrip
  <script src="chalk.20140207.js"></script>
  <script src="hello.js"></script>
</html>
```

hello.js

```
// Chalk calls main when it is completed initalization
main = function(){
  chalk.println("Hello, World!");
};
```

Chalk JavaScript Commands

chalk-commands.js

```
// Chalk calls main when it is completed initalization
main = function(){
  chalk.print("Hello, "); // print without space
  chalk.println("World!");
  chalk.newline();
  chalk.println("Hello, World!");
  chalk.hr();
  chalk.println("That's all folks");
};
```

Chalk JavaScript Commands

for-loop.js

```
main = function() {
    for(var i = 0;i < 10;i++) {
        chalk.println("i = " + i);
    }
}
```

Chalk JavaScript Commands

while-loop.js

```
// print all square numbers less that 10000
main = function() {
    var x = 1;
    while(x * x < 10000) {
        chalk.print(x * x + " ");
        if (x % 8 == 0) {
            chalk.newline();
        }
        x++;
    }
}
```