CS 416
Web Programming

Intro to Javascript
and dynamic web pages

Dr. Williams
Central Connecticut State University

Agenda

- Javascript
  - Language basics
  - Document.write()
  - Events
  - Mixed types

Static vs dynamic pages

- **HTML**
  - defines content and formatting
  - produces **static** pages (behave in same way every time they are loaded by the browser)
- **Javascript** introduced in 1995 to produce **dynamic** web pages – pages that could change their appearance
  - Overtime – different each time loaded
  - Respond to user's actions

Programming languages

- JavaScript is a client side scripting/programming language
  - Each statement specifies a particular action that the **browser** is to carry out (e.g., changing an image or opening a window when a button is clicked)
- Some programming languages are multi-purpose (C++, Java)
- **Javascript created specifically for making web pages dynamic**

What can JavaScript do?

- Gives HTML developers a **programming tool** - a scripting language with a very simple syntax
- React to **events** – page load, clicking, typing
- **Read and write** HTML elements
- **Validate form data** before it is submitted to a server, saving the server from extra processing
- Detect the visitor's **browser**
- Create **cookies** - Store and retrieve information on the visitor's computer

Adding Javascript (one of several)

- Add JavaScript statements to a Web page using the HTML tags

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

- When the browser displays the page, any statements inside the SCRIPT tags are executed and the result is displayed

Simple dynamic page

```
<html>
<head><title>simple page</title></head>
<body>
<script type=“text/javascript”>
  /*
   * my script
   */
  var firstName = prompt(“Name”,””); // set name
  document.write(“<p>Hello “ + firstName +”</p>”);
</script>
<p> some plain html </p>
</body>
</html>
```

While each of the main browsers (IE, Firefox, Chrome) support Javascript they all implement it a little differently

Comments

- Comments within Javascript are the same style as in C++ and Java
- Multi-line comment

```
/*
 * my comment
 */
```

- Single line comment

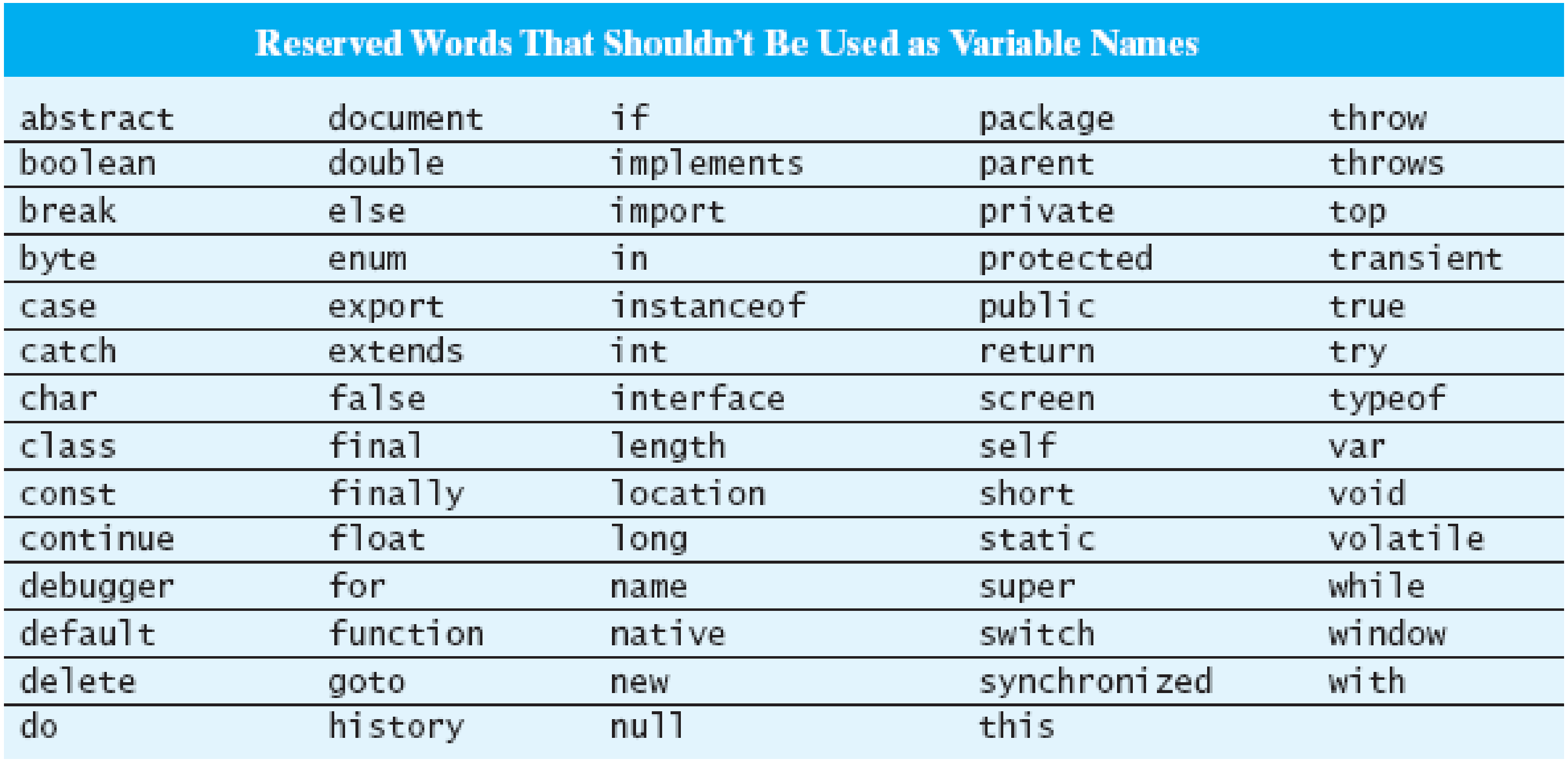
```
x = y;  // my comment
```

Javascript variables

- a variable name can be any sequence of letters, digits, and underscores (but must start with a letter)
  - valid: tempInFahr SUM current_age Sum2Date x
  - invalid: 2ShotsForU salary$
    - two words "sum_to_date"
- variable names are case sensitive, so Sum and SUM are treated as different variables

Reserved words

Because these words have special meaning they should not be used as variable names:



Reserved Words That Shouldn’t Be Used as Variable Names

abstract	document	if	package	throw
boolean	double	implements	parent	throws
break	else	import	private	top
byte	enum	in	protected	transient
case	export	instanceof	public	true
catch	extends	int	return	try
char	false	interface	screen	typeof
class	final	length	self	var
const	finally	location	short	void
continue	float	long	static	volatile
debugger	for	name	super	while
default	function	native	switch	window
delete	goto	new	synchronized	with
do	history	null	this	

Variable creation

```
var firstName = prompt(“Name", "");
var total;
```

- Variables are specified in Javascript by adding **`var`** ahead of the variable name
- Some browsers don’t require **`var`** before the variable name, but some versions of IE will report a script error without it

Using variables

- In addition to prompting for variables they can be created by assigning values to them

Ex.

```
var pi = 3.14;
var radius = prompt(“radius”,””);
var circumference = 2 * pi *
                 parseDouble(radius);
```

Reusing variables

- The value of a variable can be set multiple times
- Example

```
<script type=“text/javascript”>
  var pie = prompt(“Favorite pie”,””);
  document.write(“favorite is “ + pie);
  pie = prompt(“Hated pie”,””);
  document.write(“hated is “ + pie);
  pie = 3.14;
</script>
```

Watch out JavaScript types are NOT fixed which can lead to unexpected behavior and difficult debugging

Prompt assignment statement

```
var firstName = prompt(“Name", "");
```

- when an assignment statement involving prompt is executed by the browser
  - a separate window is opened/or shown with a text box for the user to enter text
  - when the user is done typing, he/she can click on the OK button
  - when OK is clicked, the text entered is assigned to the variable

Prompts

- General form with prompt

```
VARIABLE = prompt(“prompt message”,
 “default value”);
```

- VARIABLE – should be meaningful name representing what it contains
- Prompt message – message to let the user know what information you are expecting
- Default value – if the prompt should default to a particular text or “” leaves it blank

Write statement

```
document.write(“<p>Hello “ +
                  firstName +”</p>”);
```

- when a write statement is executed by the browser
  - the message specified in the statement is written into the HTML page
  - a message can include
    - a string literal – text enclosed in quotes
    - a variable
    - a combination of strings and variables, connected via '+'
  - when a variable is encountered, the browser substitutes the value currently assigned to that variable

Write statement (cont.)

- The string (text) output by the write statement is processed by the browser just as if it was HTML
  - As a result any formatting you want to add needs to be in the write statement as well
- Statement can be broken across lines except within a string

Creating a javascript string

- Creating it directly

```
 tempString = “Hello”;
```

- Reading from user

```
 tempString = prompt(“name”,”default text”);
```

- Appending strings

```
 tempString = “Hello” + “ World”;
```

- Appending values of variables

```
 tempString = “Hello” + name;
```

- Combining elements

```
tempString=“Hello”+first+last+”today”;
```

Putting it together

- Example

```
<script type=“text/javascript”>
  var food = prompt(“Favorite food”,””);
  // statement continued on next line
  document.write(“<p>Favorite food is “ +
 food+”<br />”);
  food = prompt(“Least favorite food”,””);
  document.write(“Least favorite food is “
 + food+”</p>”);
</script>
```

Special characters

Code	Outputs
\’	Single quote
\”	Double quote
\&	Ampersand
\\	Backslash
\n	New line
\r	Carriage return
\t	Tab
\b	Backspace
\f	Form feed

Using special characters

- Why needed, HTML output from document.write()

- Ex. <table border=“2”>

“<table border=\“2\”>”

- Ex. <table border=“ +value+”>

“<table border=**\”**”+value+”**\”**>”

Example

- Write javascript to prompt the users for 2 values and output the values in the following format

B	A	
	Value1	Value2