

**JavaScript**

Client-side dynamic documents

Netprog 2002 JavaScript 1

---

---

---

---

---

---

---

---

**Smart Browsers**

- Most *browsers* support a `<SCRIPT>` tag that is used to include executable content in an HTML document.
- There are a number of *scripting* languages that are supported

Netprog 2002 JavaScript 2

---

---

---

---

---

---

---

---

**Script Languages**

- Netscape
  - JavaScript
- Internet Explorer
  - Jscript
  - VBScript
  - PerlScript

Netprog 2002 JavaScript 3

---

---

---

---

---

---

---

---

## JavaScript Capabilities

- Add content to a web page dynamically.
- Alter a web page in response to user actions.
- React to user events.
- Interact with frames.
- Manipulate HTTP cookies

Netprog 2002 JavaScript

4

---

---

---

---

---

---

---

---

## JavaScript is not Java

- JavaScript is a very simple scripting language.
- Syntax is similar to a subset of Java.
- Interpreted language.
- Uses objects, but doesn't really support the creation of new object types\*

\*It almost does, but it's cumbersome.

Netprog 2002 JavaScript

5

---

---

---

---

---

---

---

---

## Language Elements

- Variables
- Literals
- Operators
- Control Structures
- Objects

Netprog 2002 JavaScript

6

---

---

---

---

---

---

---

---

## JavaScript Variables

- Untyped!
- Can be declared with var keyword:  
`var foo;`

- Can be created automatically by assigning a value:

```
foo=1;   blah="Hi Dave";
```

Netprog 2002 JavaScript

7

---

---

---

---

---

---

---

---

## Variables (cont.)

- Using `var` to declare a variable results in a *local* variable (inside a function).
- If you don't use `var` – the variable is a global variable.

Netprog 2002 JavaScript

8

---

---

---

---

---


---

---

---

## Literals

- The typical bunch:
  - Numbers `17 123.45`
  - Strings `"Hello Dave"`
  - Boolean: `true false`
  - Arrays: `[1, "Hi Dave", 17.234]`

  
Arrays can hold anything!

Netprog 2002 JavaScript

9

---

---

---

---

---

---

---

---

## Operators

- Arithmetic, comparison, assignment, bitwise, boolean (pretty much just like C).

```
+ - * / % ++ -- == != > <  
&& || ! & | << >>
```

Netprog 2002 JavaScript

10

---

---

---

---

---

---

---

---

## Control Structures

- Again – pretty much just like C:  
`if if-else ?: switch`

```
for while do-while
```

- And a few not in C  
`for (var in object)`

```
with (object)
```

Netprog 2002 JavaScript

11

---

---

---

---

---

---

---

---

## Objects

- Objects have attributes and methods.
- Many pre-defined objects and object types.
- Using objects follows the syntax of C++/Java:

```
objectname.attributename  
objectname.methodname()
```

Netprog 2002 JavaScript

12

---

---

---

---

---

---

---

---

## Array Objects

- Arrays are supported as objects.
- Attribute `length`
- Methods include:  
`concat join pop push reverse sort`

Netprog 2002 JavaScript

13

---

---

---

---

---

---

---

---

## Array example code

```
var a = [8,7,6,5];  
  
for (i=0;i<a.length;i++)  
  a[i] += 2;  
  
b = a.reverse();
```

Netprog 2002 JavaScript

14

---

---

---

---

---

---

---

---

## Many other pre-defined object *types*

- **String**: manipulation methods
- **Math**: trig, log, random numbers
- **Date**: date conversions
- **RegExp**: regular expressions
- **Number**: limits, conversion to string

Netprog 2002 JavaScript

15

---

---

---

---

---

---

---

---

## Predefined Objects

- JavaScript also includes some objects that are automatically created for you (always available).
  - `document`
  - `navigator`
  - `screen`
  - `window`

Netprog 2002 JavaScript

16

---

---

---

---

---

---

---

---

## The `document` object

- Many attributes of the current document are available via the `document` object:
  - Title
  - URL
  - Forms
  - Colors
  - Referrer
  - Images
  - Links

Netprog 2002 JavaScript

17

---

---

---

---

---

---

---

---

## `document` methods

- `document.write()` like a print statement – the output goes into the HTML document.

```
document.write("My title is" +  
document.title);
```

↑  
string concatenation!

Netprog 2002 JavaScript

18

---

---

---

---

---

---

---

---

## JavaScript Example

```
<HEAD>
<TITLE>JavaScript is Javalicious</TITLE>
</HEAD>
<BODY>
<H3>I am a web page and here is my
  name:</H3>
<SCRIPT>
document.write(document.title);
</SCRIPT>
<HR>
```

[j1.html](#)

Netprog 2002 JavaScript

19

---

---

---

---

---

---

---

---

## JavaScript and HTML Comments

```
<SCRIPT>
<!-- ←
document.write("Hi Dave");
document.bgColor="BLUE";
--> ←
</SCRIPT>
```

HTML comment

[j2.html](#)

Netprog 2002 JavaScript

20

---

---

---

---

---

---

---

---

## JavaScript Functions

- The keyword `function` used to define a function (subroutine):

```
function add(x,y) {
  return(x+y);
}
```

Netprog 2002 JavaScript

21

---

---

---

---

---

---

---

---

## JavaScript Events

- JavaScript supports an event handling system.
  - You can tell the browser to execute javascript commands when some event occurs.
  - Sometimes the resulting *value of the command* determines the browser action.

Netprog 2002 JavaScript

22

---

---

---

---

---

---

---

---

## Simple Event Example

```
<BODY BGCOLOR=WHITE onUnload="restore()">
<H5>Hello - I am a very small page!</H5>
<SCRIPT>
savewidth = window.innerWidth;
saveheight = window.innerHeight;
function restore() {
    window.innerWidth=savewidth;
    window.innerHeight=saveheight;
}
// Change the window size to be small
window.innerWidth=300; window.innerHeight=50;
document.backgroundColor='cyan';
</SCRIPT>
```

[j3.html](#)

Netprog 2002 JavaScript

23

---

---

---

---

---

---

---

---

## Buttons

- You can associate buttons with JavaScript events (buttons in HTML forms)

```
<FORM>
<INPUT TYPE=BUTTON
VALUE="Don't Press Me"
onClick="alert('now you are in trouble!')" >
</FORM>
```

[j4.html](#)

Netprog 2002 JavaScript

24

---

---

---

---

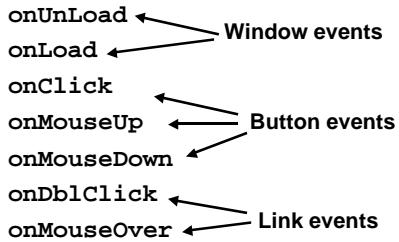
---

---

---

---

## Some Events (a small sample)



Netprog 2002 JavaScript

25

---

---

---

---

---

---

---

---

## Document Object Model

- Naming hierarchy used to access individual elements of a HTML document.
- Netscape D.O.M. is a little different than IE D.O.M. (D.A.M.)!!!\*
- Easy to use if you name all entities:
  - Forms, fields, images, etc.

Things are getting better all the time – there are standard DOMs defined by The W3C

Netprog 2002 JavaScript

26

---

---

---

---

---

---

---

---

## DOM example

```
<FORM NAME=myform ACTION=...  
Please Enter Your Age:  
<INPUT TYPE=TEXT NAME=age><BR>  
And your weight:  
<INPUT TYPE=TEXT NAME=weight><BR>  
</FORM>
```

From javascript you can get at the age input field as: `document.myform.age.value`

Netprog 2002 JavaScript

27

---

---

---

---

---

---

---

---

## Form Field Validation

- You can have JavaScript code that makes sure the user enters valid information.
- When the submit button is pressed the script checks the values of all necessary fields:
  - You can prevent the request from happening.

Netprog 2002 JavaScript

28

---

---

---

---

---

---

---

---

## Checking Fields

```
function checkform() {  
  if (document.myform.age.value == "") {  
    alert("You need to specify an age");  
    return(false);  
  } else {  
    return(true);  
  }  
}
```

**Needs to return true or false!**

Netprog 2002 JavaScript

29

---

---

---

---

---

---

---

---

## The Form

```
<FORM METHOD=GET ACTION=foo.cgi  
  NAME=myform  
  onSubmit="return(checkform())">  
  
  AGE: <INPUT TYPE=TEXT NAME=Age>  
  <INPUT TYPE=SUBMIT>  
</FORM>
```

*Needed to prevent the browser from submitting!*

Netprog 2002 JavaScript

30

---

---

---

---

---

---

---

---

## Complete Form Example

- Check the CGI example named "JavaScript" for a complete example:
  - Student grade database with form field validation in the form.

Netprog 2002 JavaScript

31

---

---

---

---

---

---

---

---

## Important Note about Form Validation!!!

- It's a good idea to make sure the user fills out the form before submitting.
- Users can bypass your form – they can create requests manually (or their own forms).
- Your CGI programs cannot rely (solely) on Client-Side JavaScript to validate form fields!

Netprog 2002 JavaScript

32

---

---

---

---

---

---

---

---

## Lots of JavaScript

- There are many javascript examples available via the course home page:

"Stupid JavaScript Tricks"

Got one of your own? Send it to Dave!

Netprog 2002 JavaScript

33

---

---

---

---

---

---

---

---