

JavaScript I.b

CSCI311

Learning Objectives

Learn basic JavaScript control and data structures

- arrays
- objects
- loops

JavaScript used for

user events and reactions

compute values and display results

change style and position of elements of HTML

display prompts or warnings

pop up windows or menus

detect info about browser

generate html

modify/transform page content

validate user entry

load pages

control media playback and loading

control CSS transitions and animations

A first JavaScript program

```
<body style="margin: 50px">
  <section><h1>Hello JavaScript</h1>
  <p>JavaScript programming is fun.</p>
  <p>Time = <span class="generated">
    <script type="text/javascript">
      var dt = new Date();
      var time = dt.getHours() + ":" + dt.getMinutes();
      document.write(time);</script></span></p>
  <p>UA = <span class="generated">
    <script type="text/javascript">
      document.write(navigator.userAgent);
    </script></span></p></section>
</body>
</html>
```

<https://developer.mozilla.org/en-US/docs/Web/API/NavigatorID/userAgent>

Where do we put JavaScript code?

embedded in a webpage

- in `<script>` elements
 - any number of them
 - in head, body, flow or phrasing elements
- JavaScript code is executed as it is encountered
- put functions or data structures in the head
- put `<script>` elements in the body where needed
- event actions can be put in attributes of tabs
- as values of event-handling attributes of HTML elements

kept in an external file (typically `.js` or `.es` suffix)

- shared across multiple pages

Image Rollover example

Use a mouse event in the image element to change the image when the mouse enters/leaves an element

```

```

two events: onmouseover, onmouseout

JavaScript expression right in the event attribute

Better practice is to put this code in a function

Can also be achieved using opacity:

```

```

general format:

- target_obj.style.property=value

Arrays in JavaScript

List-like objects

Length of array and type of elements are not fixed

Elements are indexed using integers

Declaring an array:

- `let myArray = ['tree', 'rock'];`

Accessing an index:

- `let myValue = myArray[0];`
- `myArray[0] = 'flower';`

Array Operations and Properties

Looping with foreach

- `myArray.forEach(function(item, index, array){`
- `console.log(item, index);`
- `})`

A few Array functions:

- `push`, `pop`
- `shift`, `unshift`
- `indexOf`
- `splice(position, n)`
- `Slice`

Array property:

- `length`

Accessing elements using strings

Arrays in JS are not associative, but we can “fake” key access by defining properties of an existing array

```
let anotherArr = [];  
anotherArr.thing1 = "banana";  
anotherArr.thing2 = "anana";  
anotherArr[0]; //will not access anything  
anotherArr["thing1"]; //will output "banana"
```

Arrays in JavaScript

Accessing an unassigned element returns *undefined*

Assigning to an element beyond the length of an array lengthens the array

Arrays

Arrays can contain Objects

Arrays are (a kind of) Object

Therefore, arrays can contain values of differing types:

- `var myArr = ["this", "is", "gr", 8, "!"];`

for loop

JavaScript support foreach style loops

especially useful for associative arrays

works on both indexed and associative arrays

example:

```
ans=0;
arr=[7,8,9];
for(var k in arr) { ans+=arr[k]; }
```

general syntax:

- `for (var key in arr){ /* do stuff */ }`

Array oddities

You can add elements to arrays beyond their initial length:

- `var myArr = [2, 4, 100];`
- `myArr[6] = 44;`
- `//contains: 2, 4, 100, , , , 44`

This creates “holes” in the array, so be careful!!

Arrays are sorted by string order by default

JS Loops

Javascript also supports:

- For loops
 - `for(var i=0; i<10, i++)...`
- While loops
- Do while loops

JS Conditional statements

JavaScript supports:

- if else, if else if
- switch statements
- Try catch

Objects

To create an object:

- `var myObj = new Object();`

To set properties:

- `myObj.name = "Bob";`
- `myObj["name"] = "Bob";`

To get properties:

- `var myName = myObj.name;`
- `myName = myObj["name"];`

Objects in JavaScript

One way of creating an object in Javascript:

- Pass in an object literal

Example:

- `var flower = {colour:"white", name:"rose", edible:true};`

Object methods are also defined as properties

Example:

- `var flower = {`
- `colour:"white", name:"rose", edible:true,`
- `display:function(){return "a "+ this.colour + " " + this.name;}`
- `};`

Using JavaScript with forms

We can use javascript to validate the data in a form before submitting

To access the value of an input:

- Need to get access to that input element
- Then need access to its value

Ways to access html elements:

- By id: `document.getElementById("idName");`
- By tag name: `document.getElementsByTagName("tagName");`
- By class: `document.getElementsByClassName("className");`

Validation

A simple example...