DESCRIBING XML DATA: SCHEMA

- DTD
  - Document Type Definition
  - describes type, sequence and requirements of the data (meta data)
  - can be internal or external
  - <!DOCTYPE book SYSTEM "..\books.dtd">
- Schema
  - .xsd
  - Formatted as XML
  - Supports namespaces and user defined data types
  - Built-in support for modularity

41

41

STYLING THE DISPLAY OF THE XML FILE

- An XML file will display in most browsers as a hierarchical tree view
- You can use CSS or XSLT to "style" the display
- CSS only allows basic styling
  - Link the stylesheet using a processing instruction inside the XML file
    <?xml-stylesheet type="text/css" href = "file.css"?>
- XSLT goes much further allowing filtering and broader transformations in addition to styling

42

42

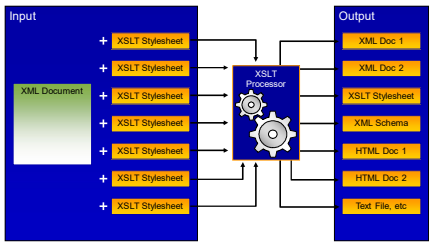
XSLT – EXTENSIBLE STYLESHEET LANGUAGE

- More options than CSS
- Many transformation options
  - XML to XML
    - Can combine with CSS for formatted output
  - XML to HTML

43

43

XSLT TRANSFORMATIONS



44

44

POP QUIZ

1. True or false: XML is the technology of choice when your primary goal is display of information.
2. Which of the following is not a technology associated with XML?
XSLT - ASP - DTD - SAX - CSS
3. True or false: an XML document is considered to be well formed if it has been validated against a DTD.
4. True or false: A document can be well-formed and not valid.
5. True or false: DTD is in XML format but Schema is not
6. True or false: A document can be validated against a DTD or a Schema

45

45

POP QUIZ – CONT.

Consider the following XML code:

1. <Employee>
2. <Name>
3. <FirstName>Bill</FirstName>
4. <LastName>Blass</LastName>
5. </Name>
6. <Title>Manager</Title>
7. </Employee>

Identify the line number that contains each of the following elements:

a. Root element
b. Child element
c. Parent element
d. End tag

46

46

BEST AND WORST WEBSITES

- Worst:
  - https://www.uat.edu/#
  - http://www.gatesnfences.com/
- Best
  - https://www.anandupender.com/index.html
  - https://www.julianabicycles.com/en-US

47

47

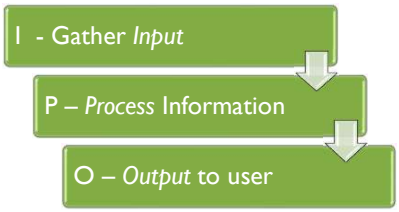
PROGRAMMING ON THE WEB

- Follows same principles/best practices as programming for other media
- Different dev environments
- Mostly scriting languages
- Robust IDE may not always be available

48

48

BEST PRACTICE: PROGRAM STRUCTURE



49

49

INPUT ON WEB

- Forms / GUI elements
- Database
- File
- Sensor

- Gather input ***before*** attempting to process

50

50

PROCESS

- Calculations
- Functions
- Get result from web service / API

51

51

OUTPUT ON WEB

- Display on screen
- Write to GUI element
- Output to database
- Output to file
- Make element on page *do* something

- *Bottom line – the user is able to see or inspect the output*

52

52

CLIENT SIDE PROGRAMMING WITH JAVASCRIPT

- Similar to C/C++
- Scripting language
- Interpreted
- Supports OOP
- Object libraries: DOM (document object model)

Incorporate into HTML page using <script> tag:

```
<script language="javascript">
document.write('hello world');
</script>
```

53

53

I/O: OUTPUT

- Output to a popup
  - alert()
    - alert("hello");
- Output to the page
  - document.write(), document.writeln()
    - document.write ("hello");
- Write to an element on the page
  - document.getElementById("result").innerHTML = "hello";

54

54

I/O: INPUT

- Yes or no
  - confirm
    - answer = confirm ("Are you sure?");
- Random value
  - prompt(question, default)
    - answer = prompt ("School name?", "Tufts University");
    - answer = prompt ("School name?")

55

55

VARIABLES IN JAVASCRIPT

- Use the keyword "var"
  - var x;
- Can declare several variables together
  - var x, y;
- Can include initialization
  - var x=0, y=1;
- Value of *null* can be used to assign a defined but non specific value to a variable
  - var x=null;
- Not required if the first use of the variable is assignment
  - x = 5;

56

56

DATA TYPES IN JAVASCRIPT

- Javascript is loosely typed
- Variables in Javascript are shape shifters!
- Numbers
  - Integral (no decimal point)
  - Floating point (*may* have a decimal point)
- Text (Strings)
  - Enclosed in quotes
  - Both single and double quotes are valid (end what you start with)
- Boolean
  - True or false

57

57

WORKING WITH DATA

- Converting a string to a number
  - parseInt
  - parseFloat
  - NaN (not a number)

```
alert(parseInt("1 is the loneliest number"));
  //result is 1
alert(parseInt("The loneliest number is 1"));
  // result is NaN
alert(parseInt("3.14 is my favorite kind of pi"));
  // result is 3
alert(parseFloat("3.14 is my favorite kind of pi"));
  // result is 3.14
```

58

58

STRINGS AND NUMBERS

- When concatenating strings with numbers, the result is *ALWAYS* a string
  - `alert("The answer is " + answer + 2);`
- Convert a number to a string
  - `var n = 4;`
    `n = n + "";`
- Use Parenthesis to calculate the math problem first:
  `alert("The answer is " + (answer + 2));`

59

59

PROBLEM:

- Create a program with 3 variables called name1, name2, name3. Each should contain the name of a friend.
- Create three more variables called age1, age2, age3. Each of these variables should store the age of the corresponding friend
- *USING THE VARIABLES*- display the name and age of each friend as:
  Bill is 21 years old.

60

60

SOLUTION …

```
var name1 = "bill", name2="bob", name3="mary";
var age1= 23, age2=17, age3=21;

document.write(name1 + " is " + age1 + " years old<br />");
document.write(name2 + " is " + age2 + " years old<br />");
document.write(name3 + " is " + age3 + " years old<br />");
```

61

61

OPERATORS

- Arithmetic
- Arithmetic with assignment
- Assignment
- Comparison
- Logical
- Concatenation

62

62

ARITHMETIC

+	addition	Adds numeric expressions.
-	subtraction	Used for negating or subtracting.
++	increment	Adds 1 to an expression.
--	decrement	Subtracts 1 from the operand.
*	multiplication	Multiplies two numerical expressions.
/	division	Divides first expression by second expression
%	modulus	Calculates the remainder of first expression divided by second expression.

63

63

ORDER OF OPERATIONS

- Please
- Excuse
- My
- Dear
- Aunt
- Sally

64

64

INTRO TO THE DOM: GETTING A "RANDOM NUMBER"

- Math.random()
- Gets a number between 0 and 1
- Need to multiply it to get a larger number
- Example- random number from 1 to 10
```
n = Math.random() * 9 + 1;
```
- Example- random number from 1 to 20
```
n = Math.random() * 19 + 1;
```

65

65

EXAMPLE: TIP CALCULATOR

```
var checkAmount,tipAmount, TIP_PERCENT=.17;
checkAmount = prompt("What is the check amount?","20");
tipAmount = checkAmount * TIP_PERCENT;
document.write("The tip for a check of "
    + checkAmount
    + " is $ " + tipAmount);
```

66

66

EXAMPLE: IS A NUMBER EVEN?

```
var num, result;
num = prompt("Enter a number", "5");
result = num % 2;
alert("The remainder of " + num
                + " divided by 2 is " + result);
```

67

67

COMPOUND ASSIGNMENT OPERATORS

- +=, -=, *=, /=, %=
- Performs the operation and does the assignment
- Example:
```
x = x + 3;
```
is the same as
```
x += 3;
```

68

68

COMPARISON

== equality True if two expressions have the same value (data conversion may occur)

!= inequality Opposite of the equality (==) operator.

> greater than True if first expression is larger

>= greater than or equal to True if first expression is greater than or equal to a second expression

< less than True if first expression is smaller

<= less than or equal to True if first expression is less than or equal to a second expression

69

69

CONDITIONAL OPERATOR ?:

- Test ? Result1 : Result2
- Do the test, if the test is true, then the value of the operator is Result1 otherwise it is Result2
- Remember this is an operator, there is no implied assignment
- "Absolute value" example:
  - n = n<0? –n : n;

70

70

EXAMPLE: EVEN REVISITED

```
var num, result;
num = prompt("Enter a number", "5");
result = num % 2;
alert("The number: " + num
      +  (result==0? " is " : " is not ")
      + "even";
```

71

71

LOGIC OPERATIONS

- Most programming languages support at least three logical operations:
  - AND (both)
  - OR (either)
  - NOT (opposite)
- Logical operations "glue" two comparison operators together.
- For example, there is no "between", but you can check if both greater than a starting value and less than an ending value are true
- Deciphering the logic
  - AND is true when two expressions are **both** true
  - OR is true when **either** of two expressions is true
  - NOT is true when a single expression is false

72

72

TRUTH TABLES

- Truth Tables are another way to represent the outputs of a logical operator.
  - You can also think of False as 0 and True as 1.
- Truth Table for AND

exp1	exp2	result
F	F	F
F	T	F
T	F	F
T	T	T

Truth Table for OR

exp1	exp2	result
F	F	F
F	T	T
T	F	T
T	T	T

73

73

"FORCED" VALUES

- When a value of one input expression forces the output result of a logical operator, it is called a "Forcing Function"
- The forcing function for AND is an input is False.
- The forcing function for OR is an input is True.

74

74

LOGICAL OPERATORS

- AND → && (shift-7)
- OR → || (shift \)
- NOT → ! (shift 1)

- These are binary operators meaning that they always require two operands

- Example: Is num *between* 1 and 10?

```
result = num >= 1 && num <=10;
```

75

75

MIXING NUMBERS, STRINGS AND OPERATORS

- + operator
  - Two strings → concatenation
  - String + number → convert number to string and concatenate
  - Two numbers → addition
- Comparison operators
  - Two strings → alphabetical order
  - String + number → convert string to number and compare numerically
  - Two numbers → numerical order

76

76

FIGURE IT OUT

- What is:
  - 3 > 4 && 6 < 10
  - "6" == 6 || 8< 0
  - ! (5 > 2)
- Assume n = 10
  What is:
  - n > 0
  - n <= 5 && n >= 10
  - n >= 1 && n<= 10
  - n == "10" || n < 3

77

77

EVENT DRIVEN PROGRAMMING – SNEAK PREVIEW

- Linear programming is sequential
  - One instruction cannot execute until the last has completed
  - Every instruction in the sequence will execute
- Event Programming executes code only when an "event" occurs
  - Events can be a key click, a page loaded, a window resized
  - If the event doesn't happen, the code will not run
  - *Event Handlers* are the code that execute in response to an event

78

78