

Comp 125 - Visual Information Processing

Spring Semester 2018 - week 5 - friday

Dr Nick Hayward

Fun exercise - using objects

- create an object or objects with information about an archive
 - *include name and location of the archive*
- use a combination of arrays and objects to store information about books in the archive - minimum five books
 - *include author's name, book title, date of publication, number of pages...*
- output to the document all of the names of the books in the archive
 - *output to the document all information for at least one book in the archive*

Output answers to the document with line breaks between results.

Fun exercise - using objects

answer variant I...

```
archive = waldzell
location = castalia
book 1 = antigone
book 2 = iliad
book 3 = the birds
book 4 = odyssey
book 5 = the persians
the birds by aristophanes, which was purchased in 1996, has 591 pages.
```



```
> // define variable with object for archive
var archive = {
  name: 'waldzell',
  location: 'castalia',
  books: [
    {
      author: 'ophocles',
      title: 'antigone',
      date: '1983',
      pages: 352
    },
    {
      author: 'homer',
      title: 'iliad',
      date: '1987',
      pages: 272
    },
    {
      author: 'aristophanes',
      title: 'the birds',
      date: '1996',
      pages: 591
    },
    {
      author: 'homer',
      title: 'odyssey',
      date: '2001',
      pages: 647
    },
    {
      author: 'aeschylus',
      title: 'the persians',
      date: '2005',
      pages: 128
    }
  ]
};

// output name and location of archive
document.write('<br>archive = ' + archive.name);
document.write('<br>location = ' + archive.location);

// create variable to store archive.books for easy use
var bookCheck = archive.books;

// output names of each book in archive
document.write('<br>book 1 = ' + bookCheck[0].title);
document.write('<br>book 2 = ' + bookCheck[1].title);
document.write('<br>book 3 = ' + bookCheck[2].title);
document.write('<br>book 4 = ' + bookCheck[3].title);
document.write('<br>book 5 = ' + bookCheck[4].title);

// output all information for book 3
document.write('<br>' + bookCheck[2].title + ' by ' + bookCheck[2].author + ',
which was purchased in ' + bookCheck[2].date + ', has ' + bookCheck[2].pages +
' pages.');
```

Fun Exercise - using objects - variant I

Fun exercise - using objects

answer variant 2...

```
archive = waldzell
location = castalia
antigone was written by sophocles (date = 1983, pages = 352)
iliad was written by homer (date = 1987, pages = 272)
the birds was written by aristophanes (date = 1996, pages = 591)
odyssey was written by homer (date = 2001, pages = 647)
the persians was written by aeschylus (date = 2005, pages = 128)
```

```
Elements Console Sources Network Performance Memory »
top Filter Default levels ▾ Group similar
> // define variable with object for archive
var archive = {
  name: 'waldzell',
  location: 'castalia',
  books: [
    {
      author: 'sophocles',
      title: 'antigone',
      date: '1983',
      pages: 352
    },
    {
      author: 'homer',
      title: 'iliad',
      date: '1987',
      pages: 272
    },
    {
      author: 'aristophanes',
      title: 'the birds',
      date: '1996',
      pages: 591
    },
    {
      author: 'homer',
      title: 'odyssey',
      date: '2001',
      pages: 647
    },
    {
      author: 'aeschylus',
      title: 'the persians',
      date: '2005',
      pages: 128
    }
  ]
};

// output name and location of archive
document.write('<br>archive = ' + archive.name);
document.write('<br>location = ' + archive.location);

// create variable to store archive.books for easy use
var bookCheck = archive.books;

for (i = 0; i < bookCheck.length; i++) {
  document.write('<br>' + bookCheck[i].title + ' was written by ' +
    bookCheck[i].author + ' (date = ' + bookCheck[i].date + ', pages = ' +
    bookCheck[i].pages + ')');
};
```

Fun Exercise - using objects - variant 2

HTML & JavaScript - create a game - game logic

part 2 - array for the answers

- create initial empty array for characters in random word
 - *get length of random word*
 - *use string `length` property*
- use for loop to add underscore per character
 - *index `i` used to add value to `answers` array*
 - *`lettersToGuess` value decremented*
 - *decrement by 1 for each correctly guessed letter*

```
// define empty array for characters in random word
var answers = [];

// set value for letters to guess from random word
var lettersToGuess = gameWord.length;

// loop through answers array - add underscore for each letter in gameWord
for (var i = 0; i < lettersToGuess; i++) {
    answers[i] = "_";
}
```

HTML - better markup

- web standards are crucial for understanding markup
 - *markup that goes beyond mere presentation*
- improved usage and structure, accessibility, integration...
- with standards, maintenance and extensibility becomes easier
- improved page structure and styling
 - *helps web designers and developers update and augment our code*
- poor markup usage
 - *to achieve a consideration and rendering of pure design*
 - *e.g. nesting tables many levels deep*
 - *adding images and padding blocks for positioning...*
- support for web standards continues to grow in popular browsers
- gives developers option to combine markup and styling
 - *HTML with CSS to achieve greater standards-compliant design*

HTML - markup and standards

- many benefits of understanding and using web standards, e.g.
- *reduced markup*
 - *less code, faster page loading*
 - *less code, greater server capacity, less bandwidth requirements...*
- *separation of concerns*
 - *content, structure, and presentation separated as needed*
 - *CSS used to manage site's design and rendering*
 - *quick and easy to update efficiently*
- *accessibility improvements*
 - *web standards increase no. of supported browsers & technologies...*
- *ongoing compatibility*
 - *web standards help improve chances of compatibility in the future...*

HTML - better structure

- consider *semantic* or *structured* markup
 - *within the context of app usage and domain requirements*
- trying to impart a sense of underlying meaning with markup
 - *correct elements for document markup*
- for a list
 - *use correct list group with list items - e.g. `ul`, `li`...*
- for a table
 - *consider table for data purposes*
 - *structure table & then consider presentation...*
- semantic markup helps create *separation of concerns*
 - *separate content and presentation*
 - *improves comprehension and usage*

Semantic HTML - intro

- importance of web standards
 - *and their application to HTML markup and documents*
- standards help drive a consideration of markup, e.g. HTML
 - *usage for what they mean*
 - *not simply how they will look...*
- semantic instead of purely presentational perspective
 - *introduction of meaning and value to the document*
- when pages are processed
 - *impart structure and meaning beyond mere presentation*
- a core consideration for usage of markup languages
- issues persist with HTML element usage
 - *e.g. inline elements such as `` and `<i>`*

Semantic HTML - a reason to care

- Semantic HTML - opportunity to convey meaning with your markup
 - *meaning may be explicit due to the containing element*
 - *implicit due to a structured grouping of elements*
- markup makes it explicit to the browser
 - *underlying meaning of a page and its content*
- notion of meaning and clarity also conveyed to search engines
 - *fidelity with query and result...*
- semantic elements provide information beyond page rendering and design
- use semantic markup correctly
 - *create more specific references for styling*
 - *greater chance of rendering information correctly*

HTML & JavaScript - create a game - HTML

update game page

- update HTML for game
 - add *id* attributes with unique reference values
 - values act as unique selectors for elements

```
<section id="play">
  <header>
    <h3>play a game</h3>
  </header>
</section>
<section id="updates">
  <header>
    <h3>game updates</h3>
  </header>
  <p id="wordStatus"></p>
</section>
```

- add unique *id* references for each section

HTML & JavaScript - create a game - game update

output start of game

- output game word to player in the updates section of HTML

```
// output game progress to player  
var lettersOutput = answers.join(" "); // create string from answers array  
document.getElementById('wordStatus').innerHTML = 'guess word: ' + lettersOutput;
```

- use `join()` method to create string from answers array
 - use paragraph with ID `wordStatus`

HTML & JavaScript - create a game - user input

add input for letter guess

- add a text input field
 - *allows player to guess a letter in the random word*
 - *add useful attributes to `input`*
 - *`placeholder` - sets default text for input (helper text)*
 - `maxlength` - sets maximum characters permitted in input

```
<section id="play">
  <header>
    <h3>play a game</h3>
  </header>
  <form>
    <input name="guess" placeholder="guess a letter" type="text" maxlength="1" id="guess" class="txtInput" />
  </form>
</section>
```

- W3Schools - HTML Form Attributes

HTML & JavaScript - create a game - guess a letter

add button to make a guess

- add a simple button
 - *player may submit letter in input field as their **guess***

```
<form id="">
  <input name="guess" placeholder="guess a letter" type="text" maxlength="1" id="guess" class="txtInput"/>
  <button type="button" id="guessBtn">guess</button>
</form>
```

- [W3Schools - HTML Form Elements](#)

Semantic HTML - example usage

```
<!-- incorrect element chosen -->  
<div id="code">  
  document.addEventListener('click', function () {  
    console.log('Click received...');  
  });  
</div>
```

```
<!-- correct element chosen -->  
<code>  
  document.addEventListener('click', function () {  
    console.log('Click received...');  
  });  
</code>
```

- semantic example usage

References

- W3Schools - HTML Form Attributes
 - *W3Schools - HTML Form Elements*
- W3Schools - Math object