

Comp 125 - Visual Information Processing

Spring Semester 2019 - Week 7 - 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)
```



```
> // 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

Fun exercise - using objects

many possible updates...

- add `bookCheck.length` to variable before use in `for` loop
- abstract repetitive logic to functions
- use object destructuring to access nested objects, properties, and values
- use of `Object` methods, e.g. `keys()`, `values()`, `entries()`...
- various alternative `for` loops
 - *introduced with ES2015 (ES6 JavaScript)*
- ...

HTML & JS - Random Greeting Generator - variant I

example solution - full HTML

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- title -->
    <title>Random Greeting Generator</title>
  </head>
  <body>
    <header>
      <h2>Create a random greeting...</h2>
    </header>
    <main>
      <!-- elements for getting user input -->
      <section id="generator">
        <header>
          <h3>Enter a name for the greeting</h3>
        </header>
        <form>
          <!-- player input for guessing a letter -->
          <input name="customName" placeholder="enter a name" type="text" autofocus id="name" class="t
          <!-- send guess letter -->
          <button type="button" id="greetingBtn">create greeting</button>
        </form>
      </section>
      <!-- elements for outputting generated random greeting -->
      <section id="output">
        <header>
          <h3>Greeting...</h3>
        </header>
        <p id="greeting"></p>
      </section>
    </main>
    <footer>
      <p>developed by ancientlives</p>
    </footer>
    <!-- script files -->
    <script src="./assets/js/greeting.js"></script>
  </body>
</html>
```

HTML & JS - Random Greeting Generator - variant I

example solution - JS logic

- start by defining an array for the random greeting

```
// define random greetings - initial fixed examples...  
var greetings = [  
  "Hello ",  
  "Bonjour ",  
  "Guten Tag ",  
  "Χαίρετε ",  
  "Salve ",  
  "Ciao ",  
  "こんにちは "  
];
```

- we'll need to get a random value from this array
 - called in an event listener for the form button

```
// pick a random greeting message  
var greeting = greetings[Math.floor(Math.random() * greetings.length)];
```

HTML & JS - Random Greeting Generator - variant I

example solution - JS logic

- add event listener for `click` on form button

```
// select guess button in document
var greetingBtn = document.getElementById('greetingBtn');

// listen for user click on `greeting` button
greetingBtn.addEventListener('click', function() {
  ...
}, false);
```


HTML & JS - Random Greeting Generator - variant I

example solution - JS logic

- add greeting variable to event listener for form button
- get value from form input text field - name entered by user
 - concatenate *greeting* and *name*
 - creates greeting to output to user

```
// pick a random greeting message
var greeting = greetings[Math.floor(Math.random() * greetings.length)];
// get name value from input field
var name = document.getElementById('name').value;
// create greeting message
var greetingMessage = greeting + name;
```

HTML & JS - Random Greeting Generator - variant I

example solution - JS logic

- add usability updates to application
 - reset form input field
 - reset focus on input field

```
// reset input field
document.getElementById('name').value = '';
// reset focus on input field
document.getElementById('name').focus();
```

- output greeting message to user

```
// output greeting message to user
document.getElementById('greeting').innerHTML = 'random greeting: ' + greetingMessage;
```

HTML & JS - Random Greeting Generator - variant 2

example solution - update JS logic

- abstract JS logic with function generateGreeting()
 - *add greetings array*
 - *get random greeting*
 - *return greeting value from function*
 - *accept parameter for name*
 - *use name with*

```
// FN: greetings generator
function generateGreeting(name) {
  // define random greetings - initial fixed examples...
  let greetings = [
    `Hello ${name}, how are you?`,
    `Bonjour ${name}, ça va? `,
    `Guten tag ${name}, wie geht es ihnen?`,
    `Χαίρετε ${name}, Πώς είσαι;`,
    `Salve ${name}, quid agis?`,
    `Ciao ${name}, come va?`,
    `こんにちは ${name}, お元気ですか?`
  ];
  // pick a random greeting message
  let greeting = greetings[Math.floor(Math.random() * greetings.length)];
  // return greeting message
  return greeting;
}
```

Resources

- W3Schools - HTML5
 - *headings*
 - *tables*
- W3Schools - JS
 - *conditionals*
 - *For loop*
 - *functions*
 - *JS - Math Object*