Comp 125 - Visual Information Processing

Spring Semester 2018 - week 5 - wednesday

Dr Nick Hayward

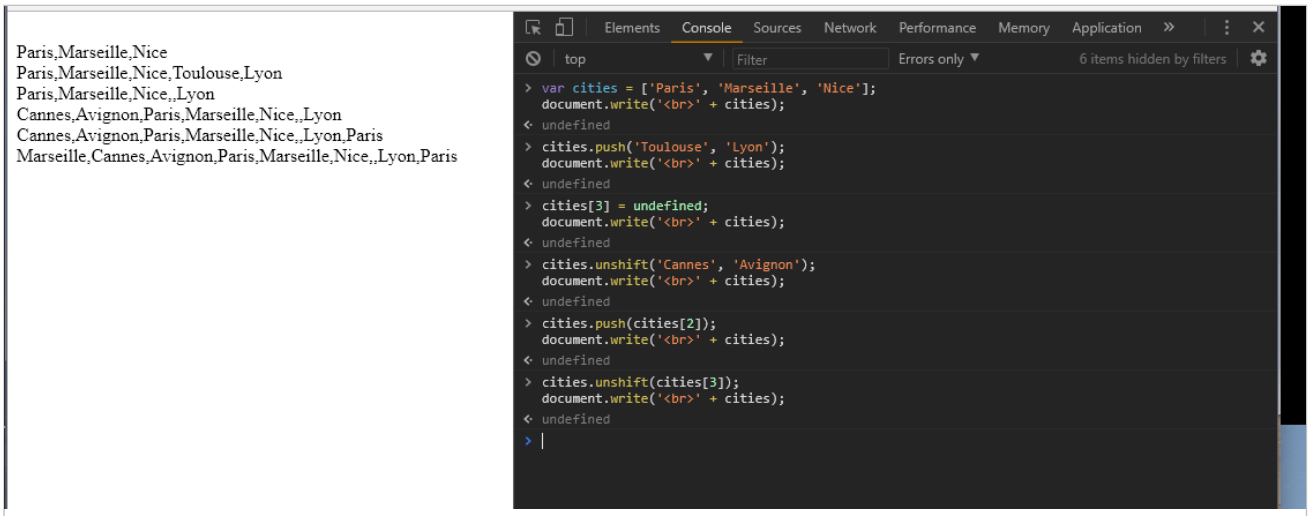
Fun exercise - using arrays

- create a new array, named **cities**, with the following values
  - *Paris, Marseille, Nice*
- add the following values to the end of the array
  - *Toulouse, Lyon*
- remove the fourth value from the array
- add the following values to the start of the array
  - *Cannes, Avignon*
- move the third value in the array to the end of the array
- move the fourth value in the array to the start of the array

Output each answer to the document with a line break between each result.

Fun exercise - using arrays

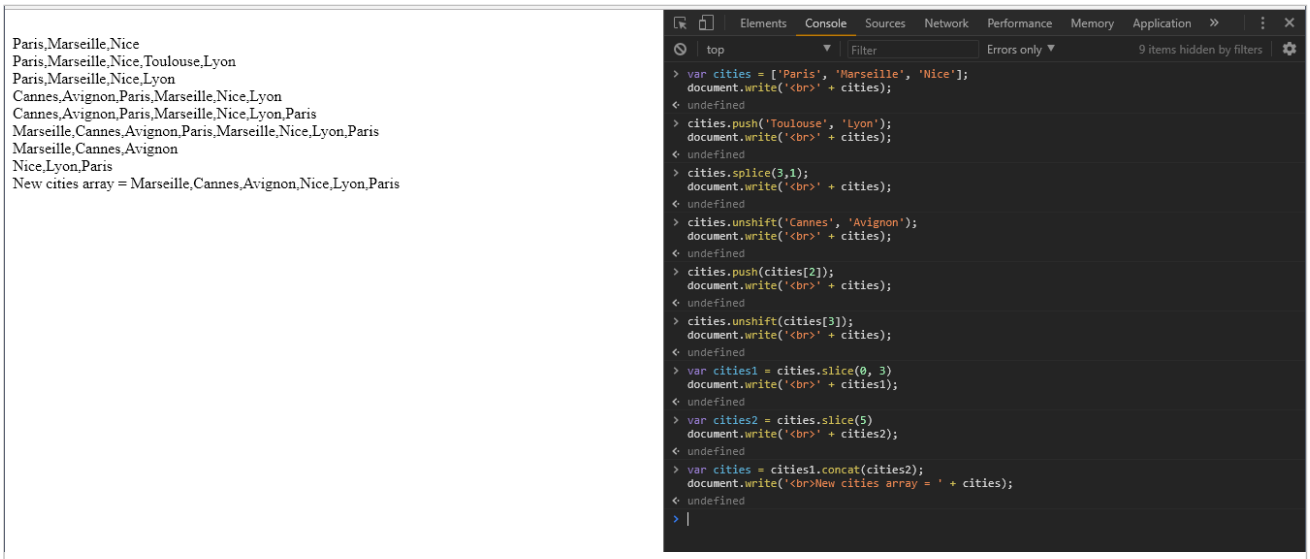
answer variant 1...



Fun Exercise - using arrays - variant 1

Fun exercise - using arrays

answer variant 2...



Fun Exercise - using arrays - variant 2

HTML & JavaScript - embed JavaScript

- start by embedding JavaScript in HTML
  - *add a `<script>` element to the HTML document*

```
<script>
var title = "Welcome to Castalia...";
console.log(title);
</script>
```

- in a browser's console
  - *each line would be executed with Return keypress*
  - *in a HTML file*
  - *JS run from top to bottom at one time*
  - *`console.log` - value will be output to browser's console*

HTML & JavaScript - create a game

- common first game to create with many languages is **Hangman**
  - *a word-guessing game*
  - *one player picks a secret word*
  - *the second player tries to guess*
  - *a word is chosen with a known length, e.g.* ***WALDZELL***
  - *8 letters in the word expressed using empty characters*

```
--------
```

- as second player guesses a correct letter
  - *we can add it to the output, e.g.*

```
 L ZE
--------
```

- good test of JavaScript usage and structure
  - *data usage*
    - interaction and input
    - output and updates...

HTML & JavaScript - create a game - basic HTML page

v0.1

- start by creating a basic HTML page for the game
  - *add header for page*
    - text input for player guess
    - render hangman data to document

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <!-- gaming title -->
    <title>Hangman Game</title>
  </head>
  <body>
    <header>
      <h3>Waldzell Gaming - Hangman</h3>
    </header>
    <main>
      <section>
        <header>
          <h4>play a game</h4>
        </header>
      </section>
      <section>
        <header>
          <h4>game updates</h4>
        </header>
      </section>
      <aside>
        <!-- add some game instructions... -->
      </aside>
    </main>
  </body>
</html>
```

HTML & JavaScript - create a game - game logic

- JavaScript - game logic includes
  - *player picks a random word for the game*
  - *logic needs to accept a player's guess*
  - *check if guess is a valid letter*
  - *record correct letters chosen by player*
  - *record counter of incorrect letters chosen by player*
  - *output game progress to player*
  - *finish the game*
    - either the player guesses the word correctly
    - or the player guesses incorrectly too many times...

HTML & JavaScript - create a game - add JS file

- create new JavaScript file for game logic
  - *e.g. game.js*
  - *add standard reference to JS file in* `index.html`

```
  <head>
  <meta charset="UTF-8">
  <!-- gaming title -->
  <title>Hangman Game</title>
      <!-- script files -->
      <script src="./assets/js/game.js"></script>
</head>
```

- we'll move this script element later in the development

HTML & JavaScript - create a game - game logic

part 1 - random word

- use JS built-in `Math` object
  - *use `random` method to get value*
  - *round the value down with `floor` method*

```
// random words for game
var gameWords = [
  "dragon",
  "wizard",
  "eagle",
  "hobbit",
  "earth",
  "planets",
  "geography"
];

// pick a random word for a new game
var gameWord = gameWords[Math.floor(Math.random() * gameWords.length)];

// check random word in console
console.log('game word = ' + gameWord);
```

- W3Schools - Math object

References

- W3Schools - Math object