

markup-storage guestbook disadvantages

disadvantages

disadvantages

- let's say we need to change the markup and we've already got entries

disadvantages

- let's say we need to change the markup and we've already got entries
 - remove the "By" from the byline

disadvantages

- let's say we need to change the markup and we've already got entries
 - remove the "By" from the byline
 - add a field to capture, like location

disadvantages

- let's say we need to change the markup and we've already got entries
 - remove the "By" from the byline
 - add a field to capture, like location
 - add avatars

disadvantages

disadvantages

- what if we need to delete/edit a single entry?

disadvantages

- what if we need to delete/edit a single entry?
 - or add an admin interface to do it?

disadvantages

- let's say we have a LOT of entries and we want to paginate them

disadvantages

- let's say we want to be able to look at a group of entries by date, eg
 - `/guestbook.php?year=2012&month=9`

disadvantages

- all comes down to the importance of content vs presentation
- or "model" versus "view"

Javascript

well, a wee bit of it

Javascript

- client-side script language of choice
- used to manipulate the DOM
- and make asynchronous HTTP requests
- gaining popularity as server-side scripting language (with Node.js)
- and general-purpose scripting

variables in Javascript

- similar to PHP

variable assignment

console

output

```
var sum = 1 + 2;  
console.log(sum);  
sum = 4 + 5;  
console.log(sum);  
sum = sum + sum;
```

variable assignment and use

console

output

```
→ var sum = 1 + 2;  
      console.log(sum);  
      sum = 4 + 5;  
      console.log(sum);  
      sum = sum + sum;
```



note! doesn't exist until after 1st statement!

variable assignment and use

console

3

output

```
→ var sum = 1 + 2;  
      console.log(sum);  
      sum = 4 + 5;  
      console.log(sum);  
      sum = sum + sum;
```



variable assignment and use

console

```
var sum = 1 + 2;  
console.log(sum);  
→ sum = 4 + 5;  
console.log(sum);  
sum = sum + sum;
```

3

output

3



variable assignment and use

console

```
var sum = 1 + 2;  
console.log(sum);  
sum = 4 + 5;  
→ console.log(sum);  
sum = sum + sum;
```

9

output

3



variable assignment and use

console

```
var sum = 1 + 2;  
console.log(sum);  
sum = 4 + 5;  
console.log(sum);  
→ sum = sum + sum;
```

9

output

3
9



variable assignment and use

console

```
var sum = 1 + 2;  
console.log(sum);  
sum = 4 + 5;  
console.log(sum);  
sum = sum + sum;
```

18

output

3
9



linear arrays in Javascript

```
var pickleNames = [  
  'Gherkin',  
  'Polish',  
  'Hungarian',  
  'Swedish'  
];
```



accessing arrays

```
var pickleNames = [  
  [0]  'Gherkin',  
  [1]  'Polish',  
  [2]  'Hungarian',  
  [3]  'Swedish'  
];
```

```
console.log(pickleNames[2]);
```



zero-based index

adding to a JS array

```
var pickleNames = [  
  'Gherkin',  
  'Polish',  
  'Hungarian',  
  'Swedish'  
];  
pickleNames.push('Kosher Dill');
```



refers to the “stack” data structure

other array operations

- https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global_Objects/Array

“associative arrays” in Javascript

- Javascript’s equivalent to an associative array is called a “hash” or “plain object” or just “object”
- use is similar to PHP

JS hashes

```
var pickleOrigins = {  
    Gherkin: 'West Indian',  
    Polish: 'Poland',  
    Hungarian: 'Hungary',  
    Swedish: 'Sweden',  
    'Kosher Dill': 'Germany'  
};
```



properties



values

JS hashes

```
var pickleOrigins = {  
    Gherkin: 'West Indian',  
    Polish: 'Poland',  
    Hungarian: 'Hungary',  
    Swedish: 'Sweden'  
    'Kosher Dill': 'Germany'  
};
```

↑ ↑
properties values

quotes necessary for
properties
with spaces

JS hashes

```
var pickleOrigins = {  
    Gherkin: 'West Indian',  
    Polish: 'Poland',  
    Hungarian: 'Hungary',  
    Swedish: 'Sweden'  
    'Kosher Dill': 'Germany'  
};
```



accessing hashes

```
var pickleOrigins = {  
    Gherkin: 'West Indian',  
    Polish: 'Poland',  
    Hungarian: 'Hungary',  
    Swedish: 'Sweden'  
};  
  
console.log(pickleOrigins['Polish']);  
console.log(pickleOrigins.Polish);  
pickleOrigins.Gherkin = 'Alaska';  
pickleOrigins['Kosher Dill'] = 'Germany';
```

JSON

- Javascript Object Notation
- combination of arrays and hashes,
arbitrarily nested
- concise, easy to understand
- very popular, very useful

JSON

```
var users = [
  {
    'username': 'eric',
    'firstName': 'Eric',
    'lastName': 'Miller',
    'email': 'eric@patternleaf.com'
  },
  {
    'username': 'fred',
    'firstName': 'Fred',
    'lastName': 'Miller',
    'email': 'fred@patternleaf.com'
  },
  {...},
  {...}
];
```

JSON

```
var users = [ ← open array
  {
    'username': 'eric',
    'firstName': 'Eric',
    'lastName': 'Miller',
    'email': 'eric@patternleaf.com'
  },
  {
    'username': 'fred',
    'firstName': 'Fred',
    'lastName': 'Miller',
    'email': 'fred@patternleaf.com'
  },
  {...},
  {...}
]; ← close array
```

each item in array
is an object

JSON data access

```
var users = [
  {
    'username': 'eric',
    'firstName': 'Eric',
    'lastName': 'Miller',
    'email': 'eric@patternleaf.com'
  },
  {
    'username': 'fred',
    'firstName': 'Fred',
    'lastName': 'Miller',
    'email': 'fred@patternleaf.com'
  },
  {...},
  {...}
];
```

```
console.log(users[0].username);
console.log(users[1].firstName);
console.log(users[1]['email']);
```

exercise

- what would the data format for our guestbook be in JSON?

Assignment

- convert guestbook from storing HTML markup to storing JSON data
- useful PHP functions:
 - `json_encode`
 - `json_decode`

email field

- form + list (view)
- JSON data (model)
- “glue” code (controller)

mvc

