

today

view: studio 1

topic: html forms, js basics

introduce studio 2: madlibs 2.0

tuesday

read: eloquent js, ch 1, 2, 3 (ch 18 optional)

due: book ex ch 2 and ch 3

studio 2: sketches

studio

view: st 1 (portal and blog)

describe: intent, content

show: validation

code view and any questions/problems
or special discoveries/successes

interactive grading: with Spencer during studios

the bottom line

you have to write and understand your own code

problems? questions?

1. post questions to Discussion forum on Canvas
2. email (Spencer && glenda)...!(Spencer || glenda)

beware of copying/pasting my code due to quote issues

studio 2: madlibs 2.0



MAD LIBS
VACATIONS

A vacation is when you take a trip to some _____ place
with your _____ family. Usually you go to some place
that is near a/an _____ or up on a/an _____
A good vacation place is one where you can ride _____
or play _____ or go hunting for _____ I like
to spend my time _____ or _____
When parents go on a vacation, they spend their time eating
three _____ a day, and fathers play golf, and mothers
sit around _____ Last summer, my little brother
fell in a/an _____ and got poison _____ all
over his _____ My family is going to go to (the)
_____ and I will practice _____ Parents
need vacations more than kids because parents are always very
_____ and because they have to work _____
hours every day all year making enough _____ to pay
for the vacation.

From VACATION FUN MAD LIBS • Copyright © 1988 by Price Stern Sloan,
a division of Penguin Putnam Books for Young Readers, New York.

mad libs 2.0: previous student examples

Jane Go
Tim Stapleton
Jennifer Wu
James Kim
Patrice Ehlert
Tonya Rodham
Deborah Bazsuly (error detection)

after this, experienced js students may leave

bad habits are hard to

common problems

...no longer allowed, use the box model instead

```
li {  
  margin-bottom:24px;  
}
```

centering elements

```
article {  
  width: 800px; // required  
  margin:auto;  
}
```

centering content

```
article {  
  text-align:center;  
}
```

open link in a new tab

```
<a href="#" target="_blank"></a>
```

we want to have good <form>

html forms

The `<form>` tag is used to create an HTML form for user input.

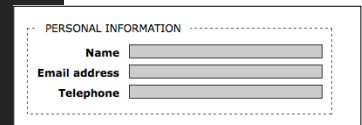
The `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<textarea>`
- `<button>`
- `<select>`
- `<option>`
- `<optgroup>`
- `<fieldset>`
- `<label>`

note: input is self-closing

form example

```
<form name="f"> <!--js needs the form name-->
<fieldset>
  <legend>PERSONAL INFORMATION</legend>
  <label>Name</label><input type="text" name="username" />
  <label>Email address</label><input type="text" name="email" />
  <label>Telephone</label><input type="text" name="phone" />
</fieldset>
</form>
```

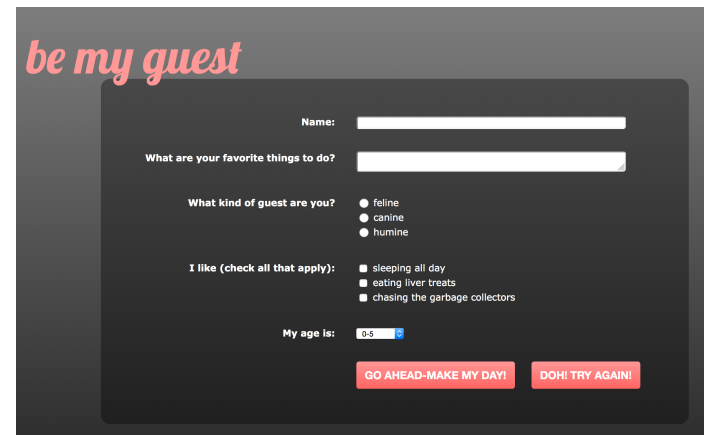


form example reset and submit

```
<form name="f">
...
<input type="submit" value="submit" />
<input type="reset" value="reset" />
</form>
```

submit reset

be my guest



codepen: form basic



javascript!

javascript

considered **object-based** (create objects that are then targeted with methods and properties) & **event driven**

most popular development language in the world :)

content / presentation / behavior

content / presentation / behavior

.html

.css

.js

browser wars

DOM scripting: the code only modifies pages via manipulation of the W3C DOM (no use of proprietary, non-standard or deprecated properties)

unobtrusive scripting: keeping the behavior entirely separate from the content (external files). users don't need JavaScript to view/interact with the site, but the experience is richer with the js

like css, there are several locations for js

in studio1, we put the link to the external js right before </body>

```
...  
<script type="text/javascript" src="script.js">  
</script>  
</body>
```

inline javascript

index.html

```
<body>
<script>
  console.log("hello world!");
</script>
</body>
```

this is how you will do the book exercises

javascript console window

send messages to help with debugging

```
console.log("check it out");
```

practice

open "internal js" from today's date on our schedule page (redrocketmedia.com/des157)
add a console.log message

```
console.log("hello world");
```

external javascript (better practice)

index.html

```
<head>
<script type="text/javascript" src="script.js">
</script>
</head>
```

script.js

```
// JavaScript Document
// call to function
popUpAlert();
// define function
function popUpAlert() {
  alert("Take me to New York, I'd love to see LA");
}
```

practice

update js to external

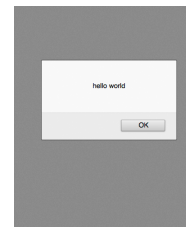
update console.log() to alert()

link from the <head> of html document

```
<script type="text/javascript" src="script.js"></script>
```

script.js

```
alert("hello world");
```



js overview

variable names

case sensitive!

```
myContainer, my_container, my_Container
```

```
theScore, the_score, the_Score
```

use only letters, numbers and underscores

cannot start with a number

must be unique names
(no 2 variables/functions/objects with same name)

cannot be a keyword

variable declaration

keyword var

```
var userName;
```

multiple declarations on one line

```
var userName, userAge, highScore;
```

scope

range within which a variable or function is available...
global or local, depends on where defined

```
//global variable  
var i = 5;  
function moveMC(){  
  myGraphic.x += i;  
  //update global variable  
  i++;  
}
```

```
function moveMCmore(){  
  //local variable  
  var myNum = 7;  
  myGraphic.x += myNum;  
  //update local var  
  myNum++;  
  //update global var  
  i++;  
}
```



```
console.log ("Hello " + "DES 157");  
> Hello DES 157
```

```
var name = "Adrian"  
console.log(name);  
> Adrian  
  
var name += " Ang"  
console.log(name);  
> Adrian Ang
```

strings

concatenation with variables

```
var className="DES 157";  
console.log ("Hello " + className);  
> Hello DES 157
```

booleans

true, false

```
console.log (3 > 2);  
> true
```

javascript loops

iterative tasks

```
for (integer variable; condition that checks value of integer; expression that updates value) {  
  //code statements;  
}
```

```
function initAll(){  
  for (var i=0; i<24; i++) {  
    var newNum=Math.floor(Math.random()*75) + 1;  
    myGraphic.x += newNum  
  }  
}
```

practice: count me in (looping numbers), template on schedule page

javascript functions

organize & reuse code

```
var n="";  
logNums();  
function logNums(){  
  for (var i=1; i<=7; i++){  
    n+=i;  
    console.log(n);  
  }  
}
```

one function

can have many statements

```
function getGroceries  
  go to the store and buy eggplant
```

```
function getGroceries  
  go to the store and buy eggplant  
  while at the store buy chocolate
```

can make changes to multiple elements

```
function changePage  
  change the background color
```

```
function changePage  
  add padding
```

```
function changePage  
  change the background color  
  add padding
```

functions with arguments

```
addNums(10,20);
```

```
function addNums(num1, num2){  
  var sum = num1 + num2;  
  console.log ("the sum of " + num1 + " and " + num2 + " is: " + sum);  
}
```

functions that return values

```
var newSum = addNums(10,20);  
console.log("newSum: " + newSum);  
  
function addNums(num1, num2){  
  var sum = num1 + num2;  
  return sum;  
}
```

practice: update looping file by adding a function that accepts two parameters, adds the two parameters together and returns the sum

call the function from a new variable and print the sum to the console