WFDD Week 4

Google Camp of Tongji University

By Breezewish

Week3 Review

Outline

- Embedding Js
- Math
- Casting
- JSON
- References
- Timeout & Interval
- DOM & Events
- AJAX & Crossdomain
- jQuery
- XSS

A handy way to write & test Js

http://repl.it

PS: REPL = Read-Eval-Print Loop

Warm-up

Implement method chaining:

```
console.log(MyCalculator
  .set(0)
  .add(10)
  .multiply(20)
  .minus(5)
  .divide(2)
  .get())  // => 97.5
```

Embedding JavaScript

- Embeded:

```
<script type="text/javascript">
// JavaScript code here
</script>
```

- External:

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

Math

- `Math.PI`
- `Math.random()`
- `Math.round()`
- `Math.floor()`
- `Math.ceil()`
- `Math.abs()`

Read more:
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

A reminder

null is **NOT** a object although:

```
typeof null === 'object'
```

A reminder

‘The ECMAScript language types are Undefined, Null, Boolean, String, Number, and Object. ’

http://www.ecma262-5.com/ELS5_HTML.htm#Section_8

Read more:
http://www.2ality.com/2013/10/typeof-null.html
http://roshanca.com/2012/traps-in-javascript-part-I/
http://coolshell.cn/articles/688.html

Casting

- Convert to integer number

  `parseInt(string, radix)`

- Convert to float number

  `parseFloat()`

- Convert to string

  ```
  Number.toString([radix])
  String(object)
  ```

valueOf() & toString()

- `valueOf()`

  returns the primitive value of the specified object

- `toString()`

  returns a string representing object

Task 0.9

```
function funcGenerator1() {
  // write code here
}
funcGenerator1()(4);  // => 4
```

Task 0.99

```
function funcGenerator2(v) {
  // write code here
}
funcGenerator2(4)();  // => 4
```

Task 0.999

```
var x = {
  // write code here
};
alert(x);  // => Hello World
```

Task 1

```
function mul(v) {
  // write code here
}
mul(3)(4)(5);  // => 60
mul(1)(2);     // => 2
mul(1)(2)…(N); // => 1*2*…*N
```

JSON

JavaScript Object Notation

- a lightweight data-interchange format

Serialize & Deserialize JSON

- `JSON.stringify()`

  Object to JSON string (serialize)

- `JSON.parse()`

  JSON string to object (deserialize)

References

```
var a = 10;
var b = a;
a = 0;
console.log(b); // => ?
```

References

```
var a = {
    p: 1
};
var b = a;
a.p = 10;
console.log(b.p); // => ?
```

References

```javascript
var a = [1, 2, 3];
var b = a;
a[1] = 10;
console.log(b[1]); // => ?
```

References in Arguments

An ancient problem:

```
function foo(x) {
    x = 1;
}
var a = 10;
foo(a);
// what is the value of a now?
```

Copy (Clone) Objects

- A simple & naive solution:

  ```
  JSON.parse(JSON.stringify(obj))
  ```

- A complex & naive solution:

  ```
  for (......) { }
  ```

setTimeout & clearTimeout

- `setTimeout`: Calls a function or executes a code snippet after a specified delay.

  ```
  var timeoutID = setTimeout(func, delay);
  ```

- `clearTimeout`: Clears the delay set by setTimeout.

  ```
  clearTimeout(timeoutID);
  ```

setInterval & clearInterval

- `setInterval`: Calls a function or executes a code snippet ***repeatedly***, with a fixed time delay between each call to that function. Returns an intervalID.

  ```
  var intervalID = setInterval(func, delay);
  ```

- `clearInterval`: Cancels repeated action which was set up using setInterval.

  ```
  clearInterval(intervalID);
  ```

Task 2

- Simple:

  Output(`console.log`) number 1 to 5, one second each.

How to manipulate HTML?

- The Document Object Model (DOM) is a programming interface for HTML, XML and SVG documents. It provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed from programs so that they can change the document structure, style and content.

- The DOM is a fully object-oriented representation of the web page, and it can be modified with a scripting language such as JavaScript.

How to manipulate HTML?

- Access the document element:

```
document.getElementById()
document.getElementsByName()
document.getElementsByTagName()
document.querySelector()
document.querySelectorAll()
```

How to manipulate HTML?

- Create new node:

```
document.createElement()
document.createTextNode()
```

Task 3

- Easy as well:

  Count `<p>` in the document.

Common APIs

- `parentNode.appendChild(node)`
- `element.innerHTML`
- `element.style`
- `element.setAttribute()`
- `element.getAttribute()`

Task 4

- Yet another simple task:

  Show “Hello World” on the page after 1 second.

Interact with user: Events

- inline:

```
<p onclick="……"></p>
```

- via DOM:

```
element.addEventListener
(type, listener[, useCapture])
element.removeEventListener
(type, listener)
```

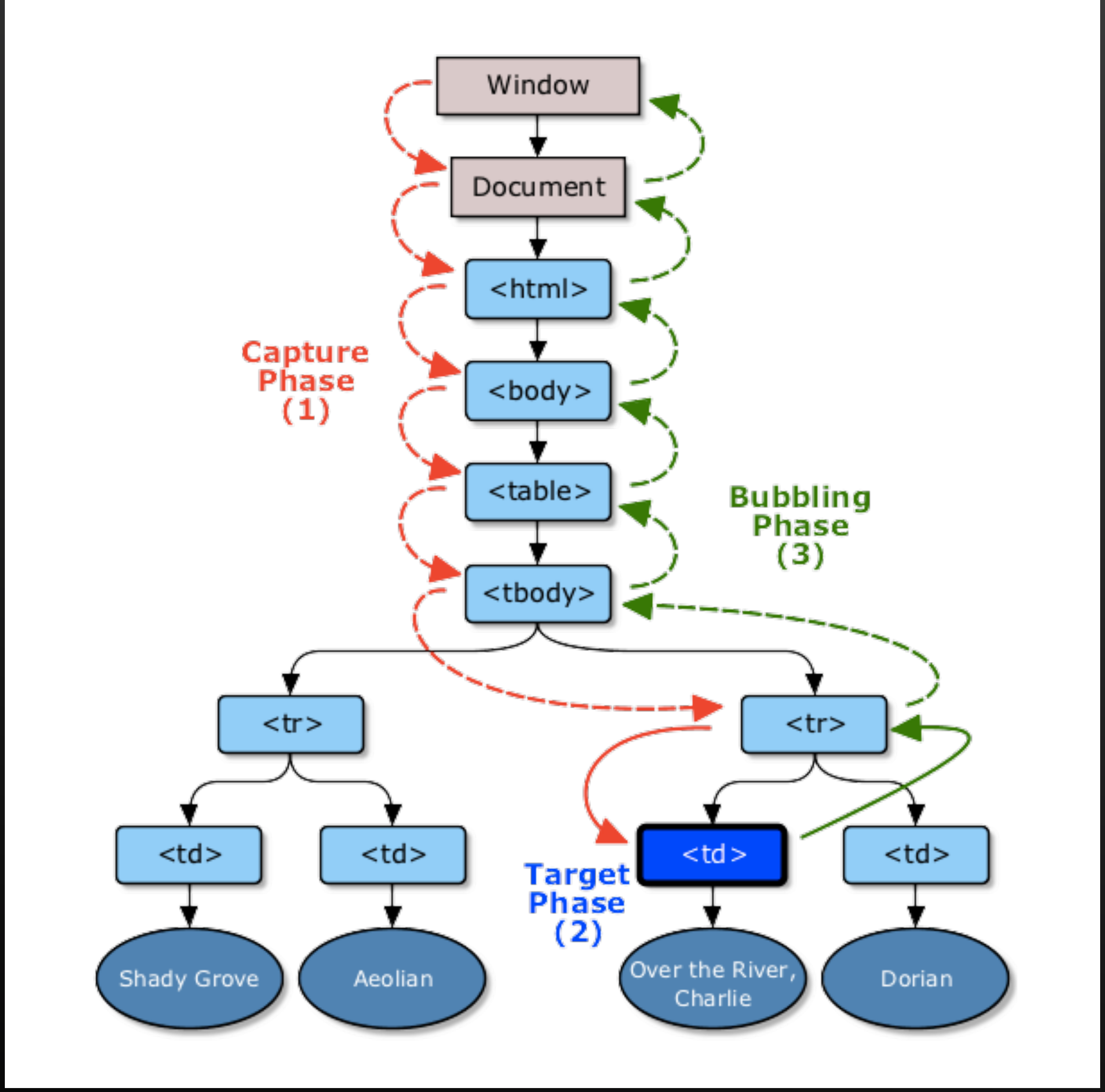
Common event types

- blur
- change
- click
- dblclick
- focus
- keydown
- keypress
- keyup
- mousedown
- mouseenter
- mouseleave
- mousemove
- mouseout
- mouseover
- mouseup
- paste

Task 5

- Alert user’s input when clicking the button

Capture & Bubble



Read more:

http://www.smashingmagazine.com/2013/11/12/an-introduction-to-dom-events/

AJAX

- Asynchronous JavaScript and XML:
  Send requests asynchronously.
- We will discuss this technology later.

Crossdomain Security

Origin: http://store.company.com/dir/page.html

URI	Outcome	Reason
http://store.company.com/dir2/other.html	Success	
http://store.company.com/dir/inner/another.html	Success	
https://store.company.com/secure.html	Failure	Different protocol
http://store.company.com:81/dir/etc.html	Failure	Different port
http://news.company.com/dir/other.html	Failure	Different host

jQuery

‘jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.’

Official website: http://jquery.com/

Select elements

```
$("#id")

$(DOM)
```

Bind event listeners

```
$jqObj.click(callback);

$jqObj.on('click', callback);
```

Emulate user’s action

```
$jqObj.click();
```

Task 6

- Print user’s input to the page when clicking the button

  use jQuery

XSS (Cross-Site Scripting) Attack

```
<div class="content">{ user input here }</div>

<img src="{ user input here }">
```

XSS (Cross-Site Scripting) Attack

Real world examples:

http://ceonline.tongji.edu.cn

http://sse.tongji.edu.cn/icss

http://4m3.tongji.edu.cn

Prevent XSS Attack

- Never trust user’s input.
- Escape everything carefully.

CSRF Attack

- CSRF = Cross-Site Request Forgery