COMP303

Internet Computing

Different Ways of Handling Web Page Code

David Rossiter

This Presentation

- Some features of handling web page code/content
  1) Dynamic creation of web page/JavaScript content
  2) Using base64 for representing anything

1. Dynamic Creation of Content

- The complete textual contents of a web page can be dynamically created by JavaScript
- If the code which generates the page ‘hides’ the text which gets generated by encoding it in some way, then it is not possible to see the code even if the user views the page source

Hiding Output in document.write()

- document.write interprets HTML commands in its argument
  - its output is part of the DOM
- You can see from the program text what the first two document.write() print; what about the third one?
  - This is a way to confuse human code reader

```
var plain_str="<p>This is plain text.</p>";
var hexadecimal_str
="\x3c\x62\x3e\x44\x6f\x20\x79\x6f\x75\x20\x6b\x6e\x6f\x77\x20\x77\x68\x
61\x74\x20\x49\x20\x61\x6d\x3f\x3c\x2f\x62\x3e";
document.write(plain_str);
document.write("<p style='color:blue'>This is plain text.</p>")
document.write(hexadecimal_str);
```

Convert Hexadecimal to Text

```
function process_str(input_str) {
var result_str = "";
for (var i = 0 ; i < input_str.length; ++i)
   result_str += String.fromCharCode(input_str.charCodeAt(i));
return result_str;    }

var hexadecimal_str = "\x3c\x62\x6f\x64\x79 … \x79\x3e";

var decoded_str = process_str(hexadecimal_str); // Convert to text
document.write(decoded_str); // Make the page
```

Returns the Unicode/ASCII value of the character/byte at position i of the string; e.g., i=0 => char=\x3c => 60_{10}

Convert Unicode value to character

Hex code of the string:

```
<body onload="alert('This is a demo of hidden JavaScript code. Even if you look at the web page source code, you will not see this alert command.')"></body>
```

- The ASCII table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Result

- The previous code generates this entire sequence:

```
<body onload="alert('This is a demo of hidden JavaScript code.
    Even if you look at the web page source code, you will not see
    this alert command.')"></body>
```

Why Bother ?

- document.write(hexadecimal_str) produces the same alert box
- String.fromCharCode (input_str.charCodeAt(i)) converts the string to Unicode and back to the same string again
  - E.g., i=0 ⇒ \x3c ⇒ 60_{10} ⇒ \x3c ⇒ '<'
- The example makes the code even harder to understand for human and even for machines!

You can Encode Functions Too

```
<script>
function_str='<SCRIPT>function helloworld()
{document.write("hello world");}<\/SCRIPT>'
document.write(function_str);
helloworld();
</script>
```

- A function can be encoded in a string (use hex if you like)
- The function is added to DOM via document.write
- You can invoke the function subsequently

2. Using Base64 Encoding

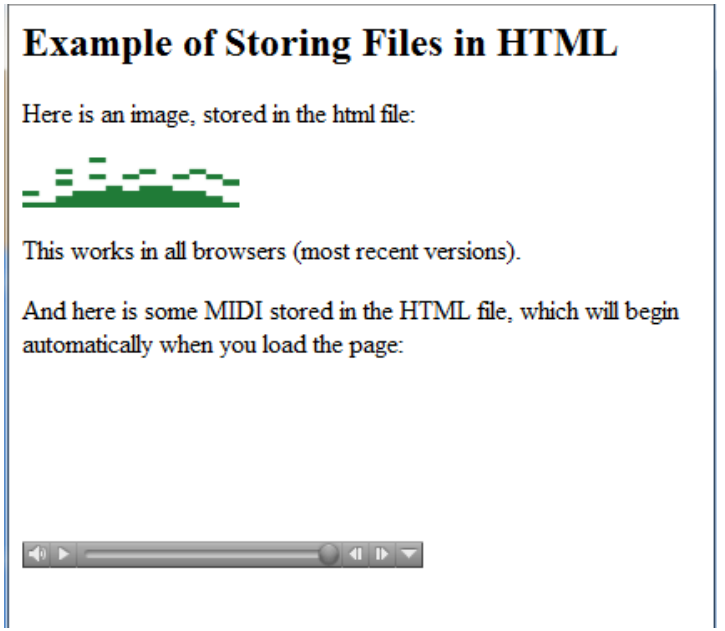
- base64 encodes binary data with printable characters, i.e., A-Z, a-z, 0-9 (totally 62 symbols) plus two more symbols (+ and /) in 6 bits
- 'data:' and base64 can be used to store any kind of file in the HTML file, including JavaScript
  - E.g., store an image and a MIDI file (see next slide)
- This feature is great because only one single file needs to be sent from the server to the browser, so loading the web page is much faster

Base64 Encoding

- Some ascii values are impossible to type directly into a web page i.e. ascii value 7 (a bell sound)
- We can use the Base64 method to encode anything using the 64 printable characters in base64
- To convert to base64, we group the bits into 6 bits, then put the bits into ascii, like this:

Text content	M			a			n		
ASCII	77			97			110		
Bit pattern	0 1 0 0 1 1			0 1 0 1 1 0			0 0 0 1 0 1	1 0 1 1 1 0	
Index	19			22			5		46
Base64-Encoded	T			W			F		u

```
<h2>Example of Storing Files
    in HTML</h2>
    <p>Here is an image, stored in the html
    file:</p>
<img src="
    iVBO . . .==" width="130" height="30" />
<p> And here is some MIDI stored in the
    HTML file, which will begin automatically
    when you load the page:</p>
<embed src="data:audio/mid;base64,
    TVRo . . ." autostart="true" >
</embed>
```



PNG image file stored in base 64 format (only first few characters shown here)

Start playing as soon as the MIDI file has loaded

MIDI file stored in base 64 format (only the first few characters shown here)

Take Home Message

- JavaScript must be interpreted by the browser, so complete hiding of source code is impossible although you can make it difficult to understand
- Being able to encode functions in hard-to-understand strings poses security threats to browsers
- Encoding is an important issue for the web since it has to deal with different human languages (more to be discussed in later lectures)