JavaScript: Coercion and Functions

Computer Science and Engineering
College of Engineering
The Ohio State University

Lecture 23

Conversion of Primitive Values

		string	number	boolean
numbers	0	"0"		false
	-0	"0"		false
	1	"1"		true
	NaN	"NaN"		false
	Infinity	"Infinity"		true
	-Infinity	"-Infinity"		true
	6.022e23	"6.022e+23"		true

Conversion of Primitive Values

		string	number	boolean
boolean	true	"true"	1	
	false	"false"	0	
strings	11 11		0	false
	11 11		0	true
	"1.2"		1.2	true
	"0"		0	true
	"one"		NaN	true

Conversion of Primitive Values

		string	number	boolean
undefined	undefined	"undefined"(NaN	false
null	null	"null" (0	false

Summary of (Simple?) Rules

- How do numbers convert to things?
 - Boolean: 0 is false, non-0 is true (exception: NaN)
- How do strings convert to things?
 - Numbers: non-valid syntax give NaN (exception: empty/blank give 0)
 - Boolean: true, only empty string is false
- How does undefined convert to things?
 - Number: NaN
- How does null convert to things?
 - Number: 0

Easier? Column-Major View

Computer Science and Engineering
The Ohio State University

- □ How do things convert to boolean?
 - Empty string is false
 - Numbers (+/-)0 and NaN are false
 - undefined and null are false
- □ Aka "falsy" (vs. "truthy")
- Importance: Boolean contexts

if (pet)... // evaluate pet as a boolean

- □ Pitfall: &&, || may not result in a boolean
 - **x** || y means x ? x : y (first x converted)

p = "cat" || "dog" //=> p == "cat"

Old idiom: !!x forces conversion to boolean
p = !!("cat" || "dog") //=> p == true

Easier? Column-Major View

Computer Science and Engineering
The Ohio State University

How do things convert to Numbers?

- Empty (and whitespace) string is 0
- Non-numeric strings are NaN
- undefined is NaN
- null is 0
- □ Importance: Used in == evaluation

== Evaluation is... Different

Computer Science and Engineering
The Ohio State University

- When types do not match, coerce:
 - null & undefined (only) equal each other
 - Strings & booleans converted to numbers "1.0" == true && "" == false
 - " == false // but " " is truthy!
 - Pitfall: NaN is not equal to NaN
- □ When *one* operand is an object:
 - Convert via valueOf (fall back toString)
 - Result then compared with usual == rules
 - Note: no coercion when *both* operands are references (== means reference equality)
- □ Sanity:

11

Use === since it never coerces

Your Turn

Computer Science and Engineering
The Ohio State University

Evaluate: True or false?

- true == '1'
- 'false' == false
- 0 == '0'
- 0 == ''
- NaN == NaN

Surprising Consequences

Computer Science and Engineering
The Ohio State University

false == 'false' //=> false == '0'//=> //=> !!'0' ('0' == 0) && (0 == '') && ('0' != '') //=>(NaN == true) || (NaN == false) //=> //=> !!NaN (NaN != 0) && (!!NaN == !!0)//=> dorey.github.io/JavaScript-Equality-Table

Functions are People too

Computer Science and Engineering
The Ohio State University

Named functions: declaration & use function foo(a, b) { ... } foo("hi", 3); Anonymous functions function $(a, b) \{ \dots \}$ // how is such a thing invoked? Functions are objects (first-class citizens) They can be assigned to variables! let foo = function(a, b) {...}; foo("hi", 3); let bar = foo; // cf. let bar = foo(); bar("world", 17);

Functions are Objects



Functions Can Be Arguments

Computer Science and Engineering
The Ohio State University

function apply(f, a) {
 return f(a); // f is a function!
}

```
function square(i) {
   return i * i;
}
```

square(3); //=> 9
apply(square, 5); //=> 25
apply(square, 12); //=> 144

Summary

Computer Science and Engineering
The Ohio State University

□ Truthy, falsey, and friends

- Type coercion is everywhere
- Coerce to boolean in conditionals
- Coerce to number for ==

Functions as first-class citizens

- Can be passed as arguments
- Can be returned as return values!
- Closure: carry their context