

Index

■ A, B

“Ad hoc” polymorphism, 55

■ C

Calling capability methods
 capability prototype methods, 72
 constructor properties, 71
 single properties, 71

Classes, 101–102

Classic books, 98

Composition, 102

Composition *vs.* inheritance, 102

Constructors

- constructor.prototype, 90–91
- Dog.prototype, 92, 94
- dynamic prototype, 93
- inheritance, 94–95
- new operator, 89
- prototype, 92
- “[[prototype]]” property, 91
- return statement, 93
- this parameter, 90

Core Python Programming, 99

C++ Programming Language, The, 98

■ D

Designing JavaScript

- capabilities liberally, 86
- composition liberally, 84–86
- ex nihilo* constantly
 - array literals, 83
 - “styles” object, 84
 - Wobj and Wobj-based objects, 84
- inheritance conservatively, 87

Dynamic properties, 106

■ E, F, G, H

Ex nihilo objects

- class
 - function, 61
 - returning, 61
- namespace object, 59–60

■ I

Inheritance, 101–102

Inheritance alternatives

 Button_sizable

 Button_sizable.Button family, 79

 Button_sizable.panel family, 80

 choosable buttons, 78–79

 click and draw functions, 80–81

 click functions, 77–78

 constructor, 77

 calling capability methods

 capability prototype methods, 72

 constructor properties, 71

 single properties, 71

 capabilities, 68

 Closable constructor, 73

 implement() method, 72–73

 interface, 67

 Maskable, 74–76

 mixins, 70

 multiple inheritance, 65

 Window_M_BS constructor, 70

Inheritance practice

 Cascading init() methods

 A.init(), 39

 B()function, 38

 B.init() method, 38, 40

 B.init() and E.init(), 39

 call() and apply() method, 41

- Inheritance practice (*cont.*)
 - capabilities implementation, 44
 - constructor, 37
 - discrete defaults, 43
 - extending family, 41
 - JSWindows system, 42
 - toString() method, 40
 - Window_M_BS constructor, 43
- JSWindows inheritance, 49
- prototype
 - class methods, 49
 - function library, 48
 - inheritance alternatives, 46
 - lookups and performance, 48
 - prototypal inheritance, 45–46
 - theory, 45
- Inheritance theory
 - class-based inheritance
 - chains, 30
 - extending instance, 29
 - overriding properties, 29–30
 - property sets, 28–29
 - classes
 - constructor, 26
 - data, 27
 - definition, 25–26
 - instance methods, 26
 - methods, 27
 - inheritance *vs.* composition
 - see* (Inheritance *vs.* composition)
 - prototypal inheritance, 31
- Inheritance *vs.* composition
 - JSWindows
 - behaviors, 34–35
 - complex properties, 33
 - HTML element, 33
 - single constructor code, 32
 - theory, 32
- Instances, 101–102

■ J, K, L

JavaScript: *The Good Parts*, 99

■ M, N

- Master Classes, 100
- Methods, 101–102

■ O

- Object-oriented programming (OOP), 102
- Object programming
 - class software, 16
 - JSWindows, 15
 - JSWindows library
 - DOM related, 19–21
 - utility, 21, 23
 - OP removes restrictions, 15
 - properties
 - dot notation, 16
 - inheriting prototypes, 19
 - Object Sum, 17–19
 - Subscript notation, 17
- Objects creation
 - class-based *vs.* prototypal
 - C++ and Java, 3
 - self and JavaScript, 3
 - Simula, 2
 - Smalltalk, 2
 - ex nihilo*
 - arrays and functions, 7
 - Object constructor, 5–6
 - object literal, 6–7
 - objects up close
 - data properties, 4
 - methods, 4–5
 - OOP-style code
 - class statics, 9
 - constructor, 8
 - default values, 10
 - getters and setters, 9
 - initial property values, 8
 - instance methods, 9
 - prototypal object creation
 - cloning, 12–13
 - object prototypes, 10
 - prototype chain, 12
 - sing() method, 11
 - reasons
 - event-driven programming, 2
 - objects do methods, 1
 - taming exponential complexity, 2
- OOP principles
 - abstraction, 57
 - classes, 56

encapsulation
 access specifiers, 52
 closures, 53
 definition, 52
 inheritance, 52
 Java's interfaces, 57
 polymorphism
 "ad hoc" polymorphism, 55
 javaScript and polymorphism, 55
 parametric polymorphism, 54
 polymorphism, 55
 subtype, 53
 surveyed pages, 97

■ **P, Q, R**

Polymorphism, 55
 Properties configuration, 105–106

"Prototypal" object paradigm, 103
 Prototype-based
 programming, 102

■ **S, T, U**

Simple closure, 103–104
 Simula, 2
 Smalltalk, 2

■ **V**

*Visual Basic in Visual
 Studio* 2015, 99

■ **W, X, Y, Z**

Wikipedia, 99