COPYING JAVASCRIPT OBJECTS

Ultimate JavaScript Objects

COPYING JAVASCRIPT OBJECTS

- Learn about the complexities of copying JavaScript objects
- Learn several strategies for achieving end goals
- Apply strategies in a few demos
- Resolve complex copying situations involving nested children

COPYING VS REFERENCING

Copying

- Default for strings, numbers, Booleans
- Changing the original will not change any other copies
- Creates another variable in memory
- Destroying a reference will not destroy the object in memory

Making a Reference

- Default for arrays and objects
- Changing the original will change any other references
- Requires minimal additional memory

COPYING CONFUSION

- Classes sharing references to an object may affect each-other
  - i.e., global configuration object
- Some values (references to other objects) cannot be truly copied to another object
- Truly copying requires fully copying all children and their children, which may not be the intended result
  - May also be impossible – e.g., any object with a reference to self
- Objects with only primitives (strings, numbers, etc.) as properties can be copied without any complication at all

COPYING AN OBJECT

- An object can be copied in a number of ways
  - Create an empty object and loop through the properties of the original, adding them to the copy
  - Create an object with object literal syntax and include the copied variables as key-value pairs
- Object properties will be copied as references only (fine if this is the intended result)

INTO THE DEEP
COPYING NESTED PROPERTIES

- A copy of an object which also attempts to copy referenced values is called a “deep” copy
- Deep copying is not possible if loops exist within references
- Can lead to confusing results
  - i.e., reference to config object being deep copied, then the config is changed but the object’s behavior does not change
- OK as long as all references are eventually resolved
- Code must recurse through all children until each is resolved
- Can be handled by external libraries (next chapters)

COPYING CONCLUSION

- Copying objects is more difficult than copying strings or floats
- Objects are not copied when creating a new reference
- Copying objects with no references as properties is straightforward
- Copying objects with nested properties is challenging but not impossible
- Objects with loop references cannot be deep copied