

HINTS

1. Fixed-size collection of values of any type **Tuple**
2. A custom type with cases and associated values **Enum**
3. Makes a method shared across all instances of a class or struct **Static**
4. How we check for and extract the value inside an optional **Unwrapping**
5. Type that stores data as pairs of keys and values **Map**
6. Places variables into strings easily **String interpolation**
7. Loop type commonly used to make infinite loops **While True**
8. A list of criteria that a type must conform to **Protocol**
9. Evaluates multiple conditions in one block of code **Switch**
10. Special method that creates instances of structs and classes **Init**
11. Functions that accept one or more parameters of a specific type **Variadic fn**
12. Code to handle errors thrown by do **catch**
13. Value passed into a function **parameter**
14. Unwrapping alternative to if let **guard**
15. Sends back a value from a function **return**
16. A catch-all case for switch blocks **default**
17. An anonymous function that you can pass around as data **closure**
18. A whole number **int**
19. Telling Swift the specific type a variable should be **type definition**
20. Question marks after optionals **?**
21. Code that is triggered when properties change **property observer**
22. May or may not exist **optional**
23. Keyword that lets function parameters be modified outside the function **inout**
24. Access control that restricts a property to being used only inside its type **private**
25. Skips the rest of the current loop iteration **continue**
26. Loop that always executes at least once **repeat loop**
27. The return type of a function that returns nothing **void**
28. How we refer to the current instance of an object **self**
29. Struct initializer that assigns values to all properties **memberwise init**
30. The name for how values used in a closure are stored for later use
31. A variable attached to a struct or class **property**
32. A type that spans values between two numbers **range**
33. Special syntax for final parameter closures **trailing closure syntax**
34. Type that holds a large floating-point number **double**
35. The name for math symbols like + and - **operators**
36. The ability to treat an object of one type as another type **polumorphysm**
37. Adds extra functionality to a type **extension**
38. The name for a function that exists inside a struct or class **method**
39. Exits a loop immediately **break**
40. Keyword for a function that can trigger errors **throws**
41. Key that lets us replace a method inherited from a superclass **override**
42. When one class builds on another **inheritance**
43. Reads the length of a string **count**
44. Apple's all-in-one code editing environment **xCode**
45. Operator that takes three operands **ternary operator**
46. Puts off work until later **lazy**
47. Name for a method called before class destruction **deinit { ... }**
48. Ordered collection of values stored in a single value **array**
49. A class that cannot be inherited from **final**
50. Holds either true or false **boolean**