Structures and Enumerations Are Value Types

A value type sends a copy of a value
when assigned or passed to a function.
(Not a reference to the original value.)

Classes are Reference Types

Rather than a copy, a reference to the same existing instance is used when assigned or passed to a function.

Comparing Structures and Classes

Both have properties, methods, subscripts, initialization, extensions, and protocols.

Only Classes have inheritance, type casting, deinitializers, and automatic reference counting.

Definition Syntax

```
struct SomeStructure {
    // CODE
}

class SomeClass {
    // CODE
}
```

Structs and Classes are types.

Types start with a capital letter.
Properties and methods start with a
lower case letter.
(This capitalization style is a common human convention and not hard-coded into Swift.)

For example:

```
struct Resolution {
    var width = 0
    var height = 0
}
class VideoMode {
    var resolution = Resolution()
    var interlaced = false
    var frameRate = 0.0
    var name: String?
}
```

Structure and Class Instances

```
let someResolution = Resolution()

let someVideoMode = VideoMode()
```

Accessing Properties

Use dot syntax:

```
instanceName.propertyName
```

For example:

```
someResolution.width
```

Multiple dots can be used for a drill down into subproperties. For example:

```
someVideoMode.resolution.width
```

Assignment can also be used with dot notation.
For example:

```
someVideomode.resolution.width = 1280
```

Memberwise Initializers for Structure Types

A Structure's memberwise initializer can receive values by name. For example:

```
let vga = Resolution(width: 640, height: 480)
```

Class instances don't receive a default memberwise initializer.

Identity Operators

```
classInstanceOne === classInstanceTwo
// Identical to. Refers to the same instance.

classInstanceOne !== classInstanceThree
// Not identical to. Does not refer to the same instance.
```

Pointers

A Class Reference is similar to a pointer, but is not a direct pointer to a memory address, and does not require an asterisk (*) to create a reference.