Constructing Code: Who Does What?

CREATING FUNCTIONS

Simon Allardice

STAFF AUTHOR, PLURALSIGHT

@allardice   www.pluralsight.com

Calling Functions

```
// C-style function call
showMessage(123);
```

Calling Functions

```
// C-style function call
showMessage(123);

// Swift style function call
showMessage(number: 123)
```

argument label

Calling Functions

```
// C-style function call
showMessage(123);

// Swift style function call
showMessage(number: 123)
```

argument label

```
// Swift style function call with three arguments
showMessage(number: 123, name: "Grace", largeFont: true)
```

Calling Functions: Example

```
// C-style function call
stride(0, 256, 16)
```

Calling Functions: Example

```
// C-style function call
stride(0, 256, 16)

// Swift style function call
stride(from: 0, through: 256, by: 16)
```

Function parameters are **constants**, and immutable by default.

Functions Without Return Values

```
func simpleFunction(number: Int) {

     // code
     // ...

}
```

Functions Without Return Values

```
func simpleFunction(number: Int) {

    // code
    // ...

}
```

Functions Without Return Values

```
func simpleFunction(number: Int)         {

     // code
     // ...

}
```

Functions Without Return Values

```
func simpleFunction(number: Int) -> Void {

     // code
     // ...

}
```

Functions Without Return Values

```
func simpleFunction(number: Int) -> Void {

    // code
    // ...

}
```

no return value

```
// example functions
playMP3(filename: String) -> Bool {
   ...
}

playOGG(oggFile: String) -> Bool {
   ...
}

showImage(at url: String) -> Bool {
   ...
}

loadVector(_ url: String) -> Bool {
   ...
}
```

Swift Function Type

```
( )  ->
```

Swift Function Type

```
(parameter_type) -> return_type
```

Swift Function Type

```
// example functions
playMP3(filename: String) -> Bool { ... }

playOGG(oggFile: String) -> Bool { ... }

showImage(at url: String) -> Bool { ... }

loadVector(_ url: String) -> Bool { ... }
```

Swift Function Type

```
(String) -> Bool
```

```
// example functions
playMP3(filename: String) -> Bool { ... }

playOGG(oggFile: String) -> Bool { ... }

showImage(at url: String) -> Bool { ... }

loadVector(_ url: String) -> Bool { ... }
```

Swift Function Type

(*parameter_type*) -> *return_type*

```
(String) -> Bool
```

```
// example functions
playMP3(filename: String) -> Bool { ... }

playOGG(oggFile: String) -> Bool { ... }

showImage(at url: String) -> Bool { ... }

loadVector(_ url: String) -> Bool { ... }
```

Swift Function Type

```
(parameter_type) -> return_type
```

```
(String) -> Bool
```

Swift Function Type

```
(parameter_type) -> return_type
```

```
(String) -> Bool
```

Swift Function Type

```
(parameter_type) -> return_type
```

```
(String) -> Bool
```

"a function that takes a String and returns a Bool"

Swift Function Type

```
(parameter_type) -> return_type
```

```
(String) -> Bool
```

"a function that takes a String
and returns a Bool"

```
(Int) -> String
```

"a function that takes an Int
and returns a String"

Swift Function Type

```
(parameter_type) -> return_type
```

```
(String) -> Bool
```

"a function that takes a String and returns a Bool"

```
(Int) -> String
```

"a function that takes an Int and returns a String"

```
(Double, Double) -> [String]
```

"a function that takes two Doubles and returns an Array of Strings"

Functions Without Return Values

```
func simpleFunction(number: Int) {
     // code goes here
     // ...
}
```

Functions Without Return Values

```
func simpleFunction(number: Int) {
     // code goes here
     // ...
}
```

no return arrow

Functions Without Return Values

```
func simpleFunction(number: Int) {
     // code goes here
     // ...
}
```

no return arrow

Has the Function Type:

```
(Int) -> Void
```

"a function that takes an Int and returns nothing"

Functions Without Return Values

```
func verySimpleFunction() {
     // code goes here
     // ...
}
```

Functions Without Return Values

```
func verySimpleFunction() {
     // code goes here
     // ...
}
```

Has the Function Type:

```
() -> Void
```

"a function that takes no parameters and returns nothing"

Defining and Calling Functions

```
func showMessage(number: Int, name: String) {
    ...
}
```

Defining and Calling Functions

```
func showMessage(number: Int, name: String) {
    ...
}




// later..
showMessage(number: 123, name: "Grace")
```

Defining and Calling Functions

```
func showMessage(number: Int, name: String) {
    ...
}
```

parameter name

```
// later..
showMessage(number: 123, name: "Grace")
```

Defining and Calling Functions

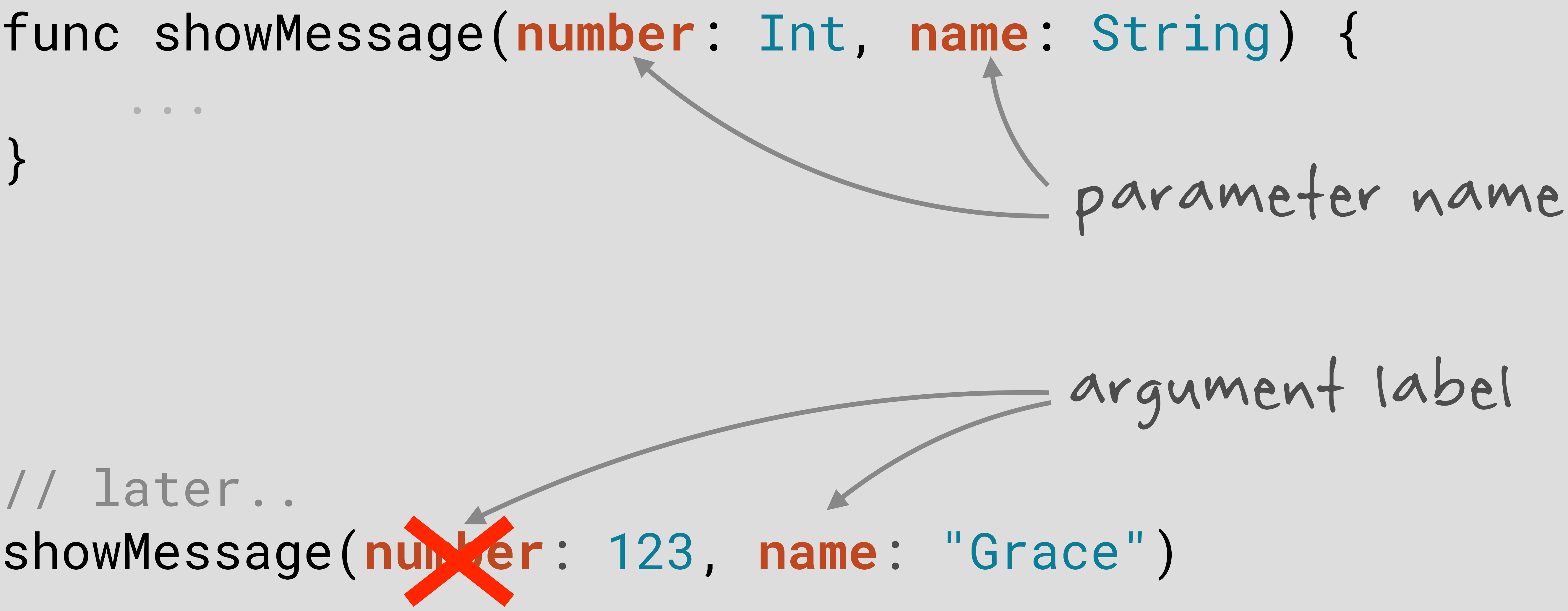
```
func showMessage(number: Int, name: String) {
    ...
}
```

parameter name

argument label

```
// later..
showMessage(number: 123, name: "Grace")
```

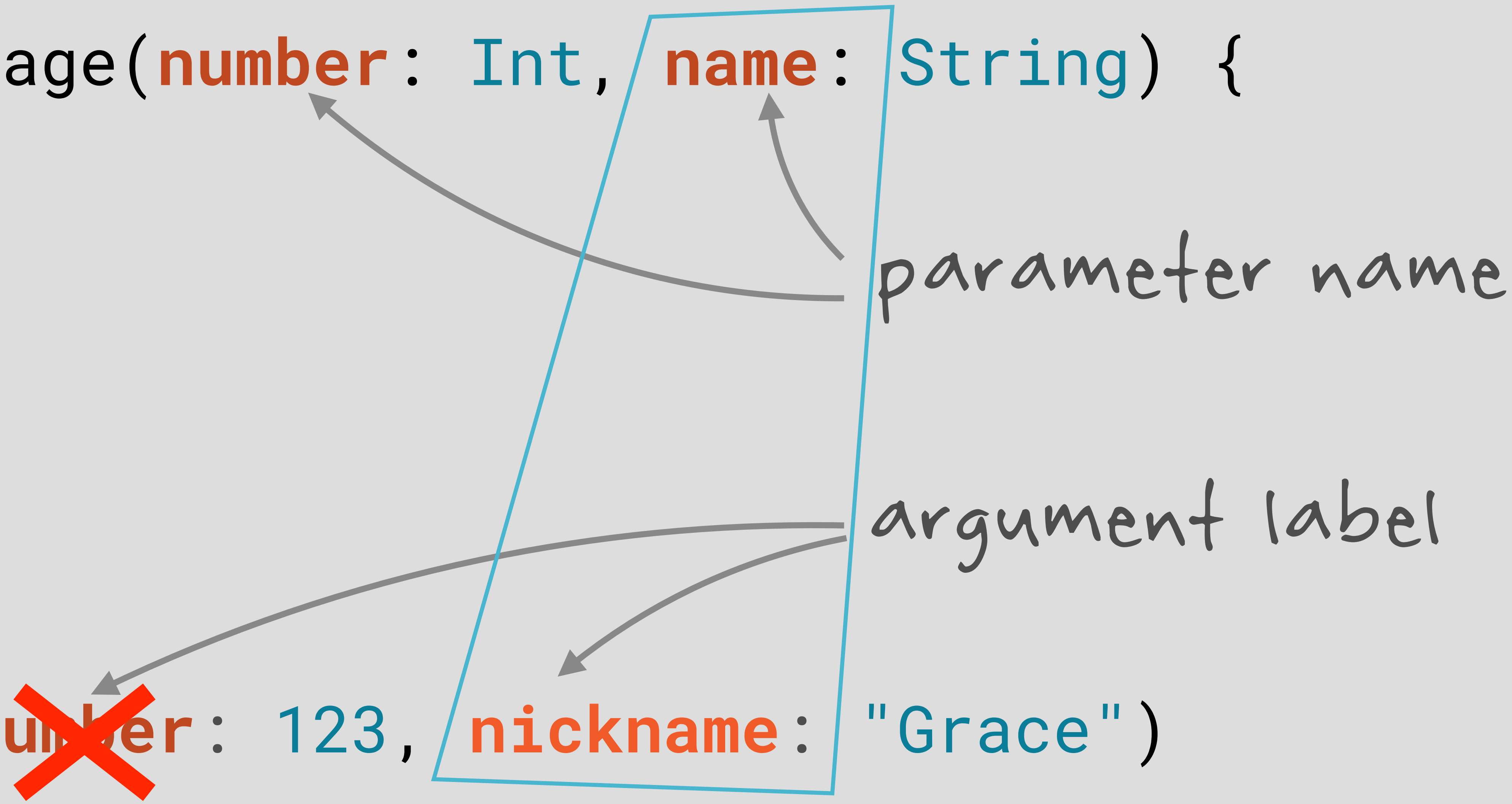
Defining and Calling Functions



```
func showMessage(number: Int, name: String) {
    ...
}


// later..
showMessage(~~number~~: 123, name: "Grace")
```

Defining and Calling Functions



```
func showMessage(number: Int, name: String) {
    ...
}



// later..
showMessage(~~number~~: 123, nickname: "Grace")
```

Functions With Multiple Parameters

```
// define the function
func purchase(item: Int, account: String, isDigital: Bool) {
    ...
}


// call the function
purchase(item: 23728, account: "ADF543", isDigital: false)
```

Functions With Multiple Parameters

```
// define the function
func purchase(_ item: Int, account: String, isDigital: Bool) {
    ...
}


// call the function
purchase(23728, account: "ADF543", isDigital: false)
```

Functions With Multiple Parameters

```
// define the function
func purchase(_ item: Int, account: String, isDigital: Bool) {
    ...
}


// call the function
purchase(23728, account: "ADF543", isDigital: false)
```

Functions With Multiple Parameters

```
// define the function
func purchase(_ item: Int, _ account: String, _ isDigital: Bool) {
    ...
}


// call the function
purchase(23728,"ADF543",false)
```

Functions With Multiple Parameters

```
// define the function
func purchase(_ item: Int, _ account: String, _ isDigital: Bool) {
    ...
}


// call the function
purchase(23728,"ADF543",false)
```

Calling Functions: Example

```
// C-style
process("A674", 4, false, true, 8.2)
```

Calling Functions: Example

```
// C-style
process("A674", 4, false, true, 8.2)

// Swift style
process(itemID: "A674", quantity: 4, expedite: false,
        freeShipping: true, tax: 8.2)
```

Argument Labels

```
func showMessage(textToShow: String) {
    ...
}
```

Argument Labels

```
func showMessage(textToShow: String) {
      ...
}
```

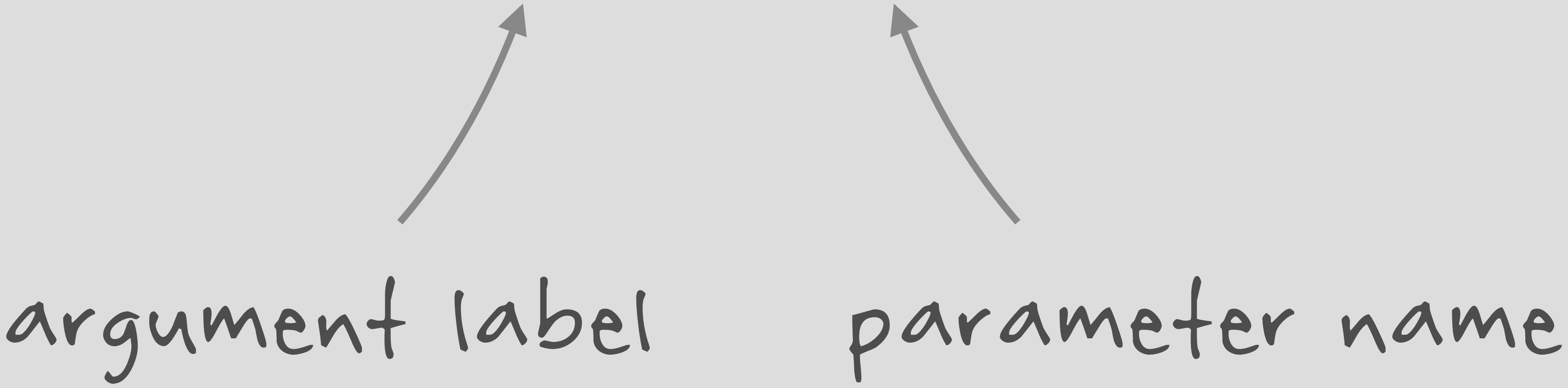
parameter name and argument label

Argument Labels

```
func showMessage(message textToShow: String) {
    ...
}
```

Argument Labels

```
func showMessage(message textToShow: String) {
    . . .
}
```

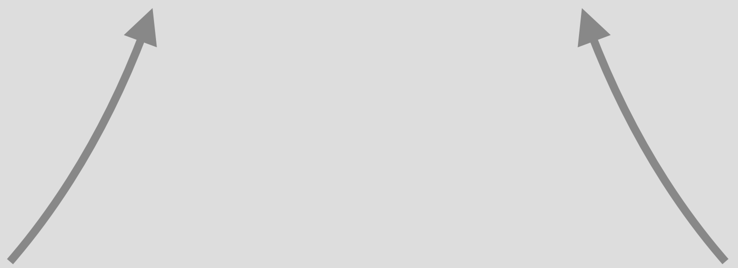


argument label

parameter name

Argument Labels

```
func showMessage(message textToShow: String) {
    ...
}
```



argument label
(used outside the function)

parameter name
(used inside the function)

Naming Functions

```
// in JavaScript
typeof someVariable
```

Naming Functions

```
// in JavaScript
typeof someVariable

// in C#
typeof(someVariable)
```

Naming Functions

```
// in JavaScript
typeof someVariable
```

```
// in C#
typeof(someVariable)
```

```
// in Swift
type(of: someVariable)
```

Naming Functions

```
// could be written as a choice of
strideThrough(0,256,16)
// or
strideTo(0,256,16)
```

Naming Functions

```
// could be written as a choice of
strideThrough(0,256,16)
// or
strideTo(0,256,16)


// in Swift
stride(from: 0, through: 256, by: 16)
// or
stride(from: 0, to: 256, by: 16)
```