

¡Bienvenidos!

zero.swift - MXL
Meetup #0 2/03/2019

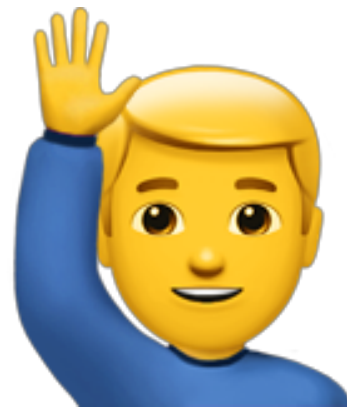
¿Qué hay para hoy?

- Avisos.
- Meetup.
- iOS: ¿Qué se necesita para desarrollar tu propia app?
- Swift: Intro al lenguaje parte I.
- Despedida.

Avisos

- Tenemos canal de slack: https://join.slack.com/t/mxlios/shared_invite/enQtNTYwMjU0NjY2OTEzLTEyZDY0ZDQ0NjA0ODIzZTgxZTJkODI2ZDUwODFiYmUxNGI1MDVmMmEyY2E2NWVI MmE3NTdhZDgyYWFkMzg1ODM
- Presentaciones & Código fuente estará en GitHub: <https://github.com/fmbosox/SwiftMeetupMexicali>
- Grupo de Facebook: [MXL] iOS Developers

¡Hola!



- ¿Cuál es tu nombre?
- ¿Cómo te enteraste del meetup?
- Tu razón de estar en el meetup es...

¿Qué se necesita para crear tu propia app para iOS/macOS?

Autor: { mxlbx }

COMMODORE BASIC V7.0 122365 BYTES FREE
(C)1986 COMMODORE ELECTRONICS, LTD.
(C)1977 MICROSOFT CORP.
ALL RIGHTS RESERVED

READY.
█

Las 4 herramientas básicas.

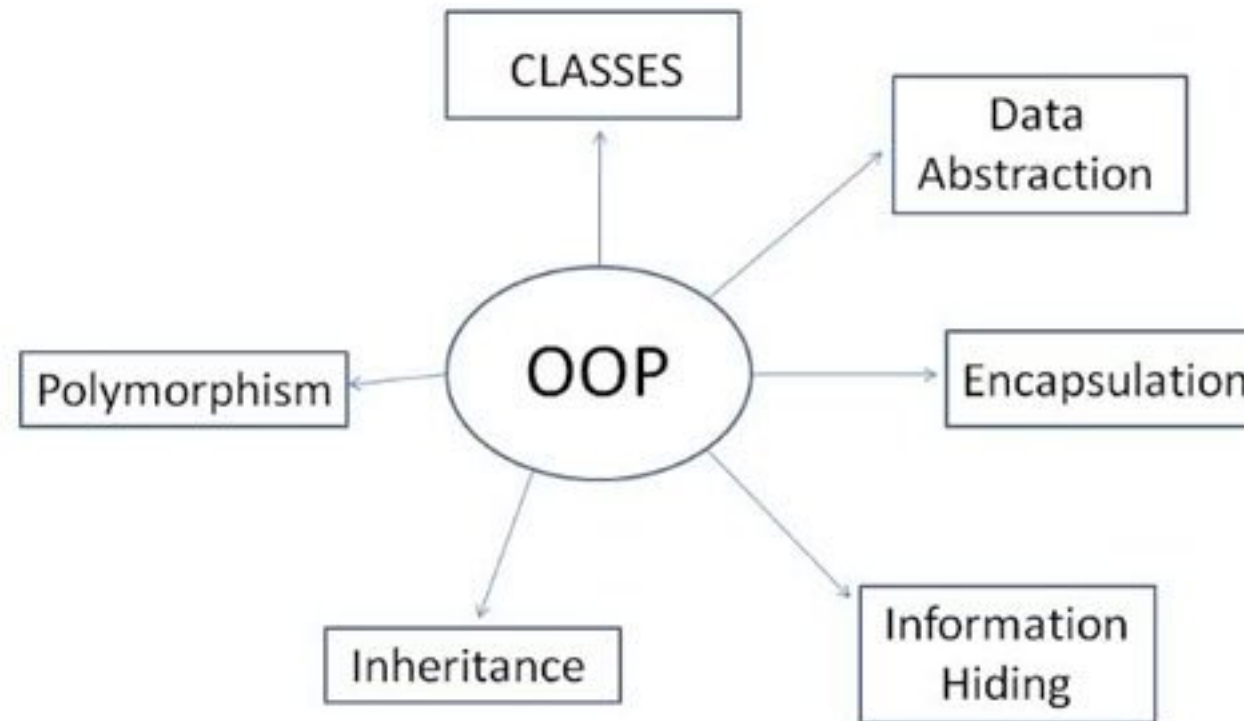
1.OOP

2.SWIFT || Objective C

3.iOS/macOS & MVC

4.XCODE

OOP



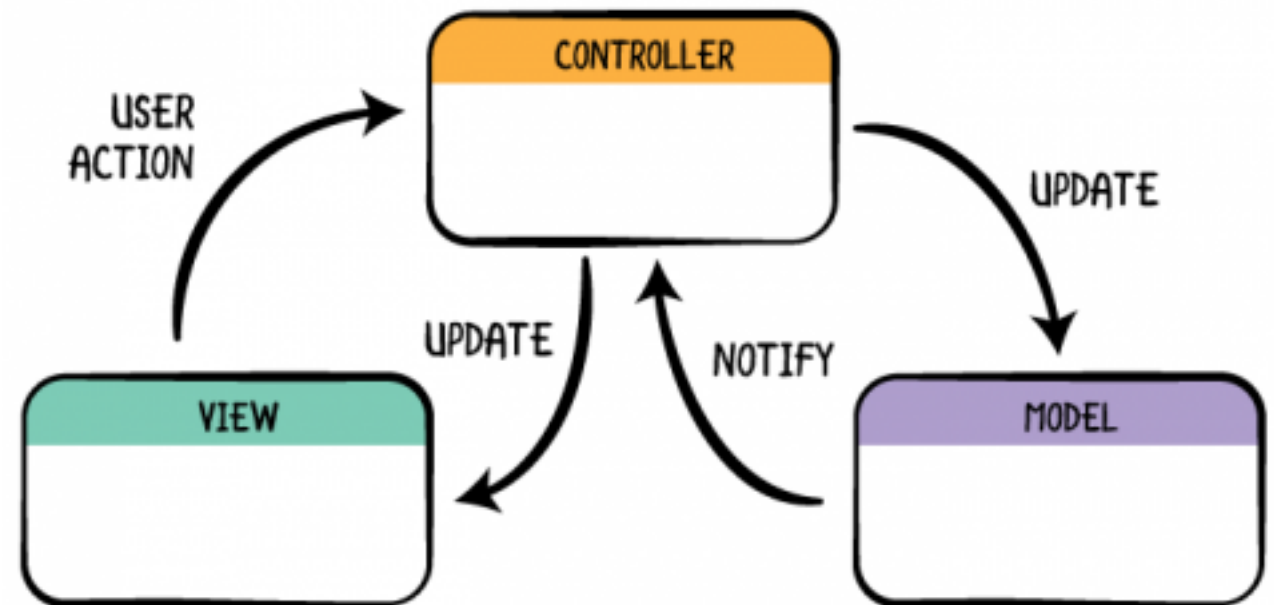
Lenguaje de programación



www.swift.org

Frameworks & MVC

- UIKit
- Foundation
- ...



Xcode



Plus

- Herramienta de diseño de UI (Sketch)
- Algoritmos.

Recomendaciones

- Curso: CS50! <https://youtu.be/5azaK2cBKgw>
- Curso: Stanford iOS Development : <https://youtu.be/ilQ-tq772VI>
- Libro: Intro to app development with Swift

Intro a Swift

Parte I

Swift

- Language moderno < 5 años.
- “*general-purpose*”
- **Amigable con programadores nuevos.**
- La asignación de memoria es manejada automáticamente.

Constant & Variables

let

- Se utilizan para almacenar y hacer referencia a valores
- Se asigna un nombre que lo identifica.
- Estos valores pueden ser declarados como “fijos” y el valor no puede cambiar
- Hacen que el código sea más seguro y claro.

var

- Se utilizan para almacenar y hacer referencia a valores
- Se asigna un nombre que lo identifica.
- Estos valores son declarados como “variables”.
- El valor puede cambiar.

***Nombre**

****Type**

*****Opcional**

```
let a: String = "hello"  
var b = "world";  
print(a)  
print(b)
```

***No:** espacio en blanco,[+,-,..],flechas, números al inicio

****No** se puede cambiar el type, redeclaración.

*****Solo:** más de una declaración en una sola linea.

Integers

1,2,3...

- No tienen componente fraccional.
- Pueden ser (signed) [-, 0, +] o unsigned [0, +]
- Tamaño en memoria: 8, 16, 32 y 64 bits para representar los números. [Int32, UInt16, etc]
- No es necesario seleccionar tamaño. Int = [Int32 ó Int64]

Floating-Points Numbers

1.0, 2.11, 3.14159265...

- Números con componente fraccional.
- **Double [64Bits]** o Float [32Bits]
- Double precision de al menos 15 números decimales.
- Float precision de 6 números decimales.

Booleans

true or false

- “Logical type” solo tiene dos valores.
- Utiles para trabajar con “conditional statements”

String

“hello, world”

- Serie de letras. “Hola”
- Cada letra puede ser representada por Character.
- Concatenación e interpolación.
- Sigue las reglas de la representación de Unicode.

Tuples

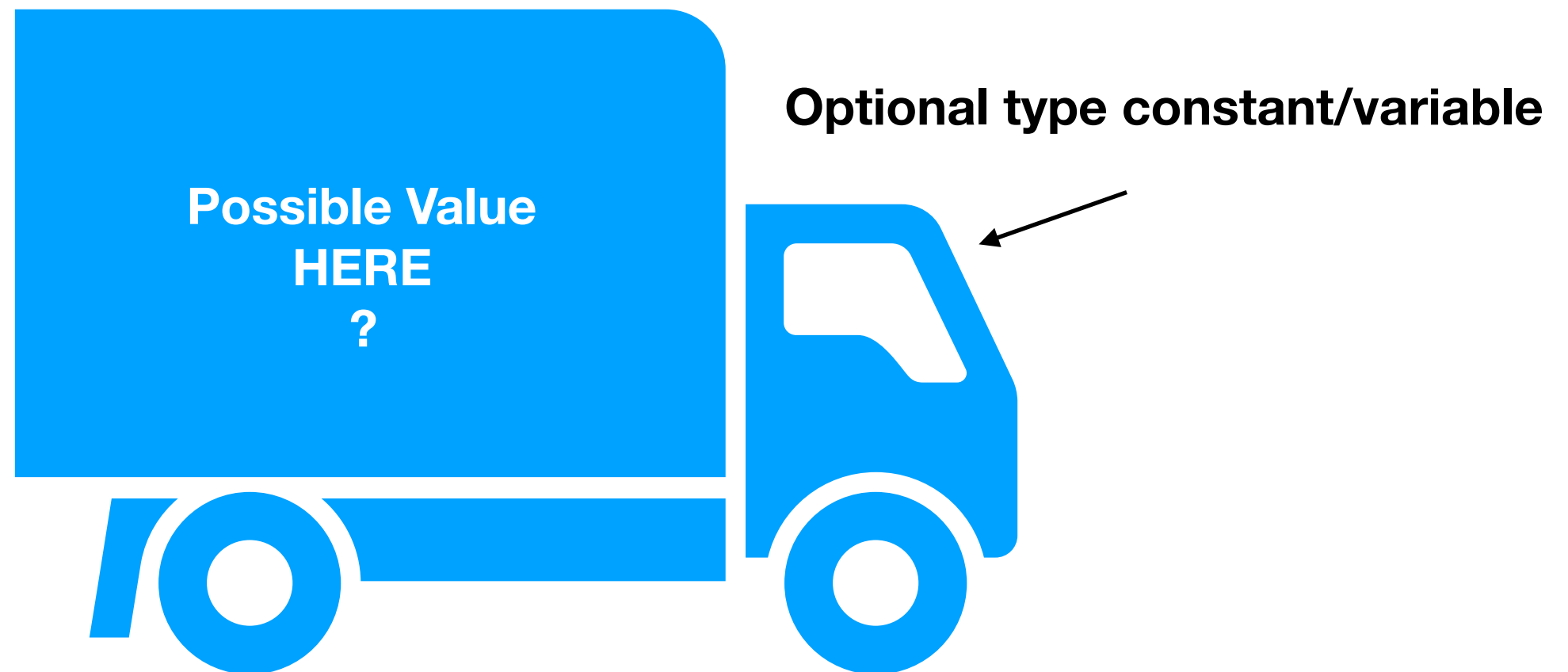
(Int,Int)

- Agrupa “values” en uno solo.
- No necesariamente los valores deben de ser del mismo tipo.
- Acceso simple.
- Solamente para uso en “Temporary scope”.

Optionals

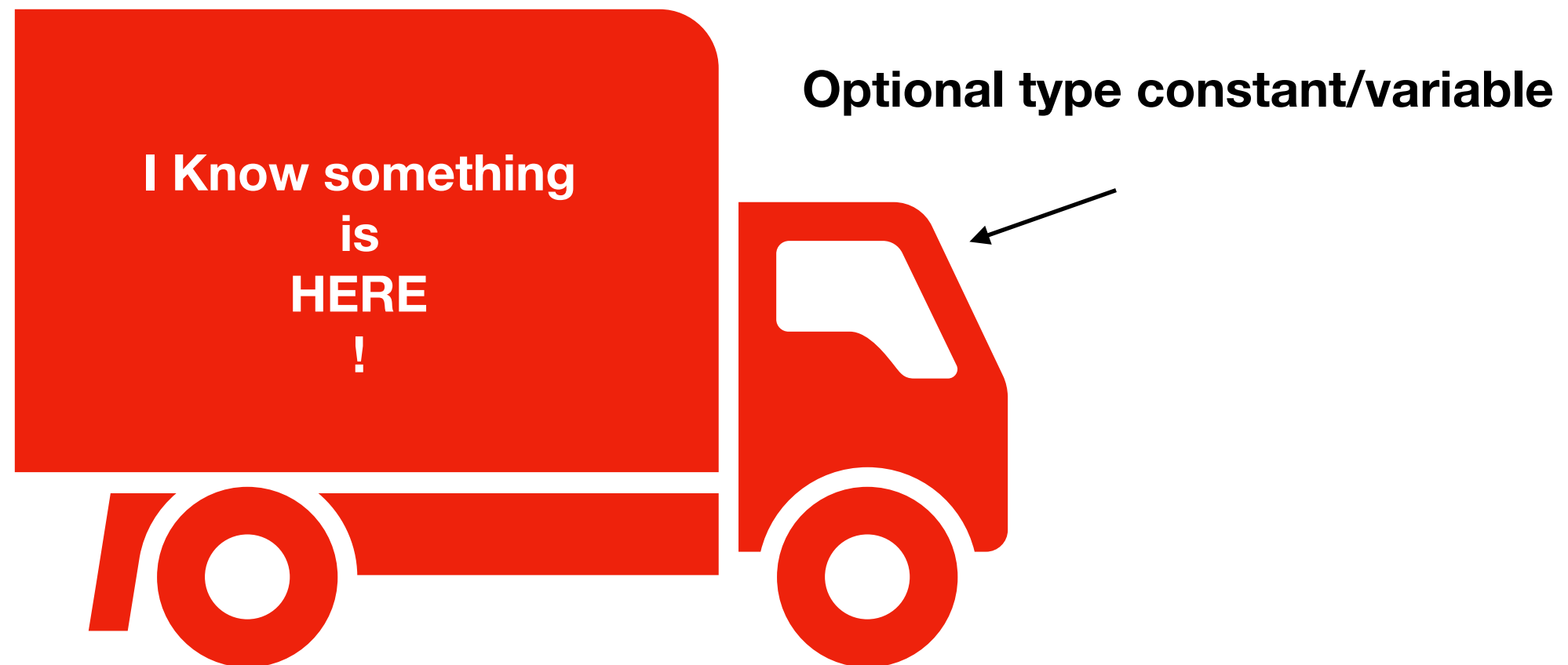
Ser o no ser.

- Se utilizan cuando un valor puede estar ausente.



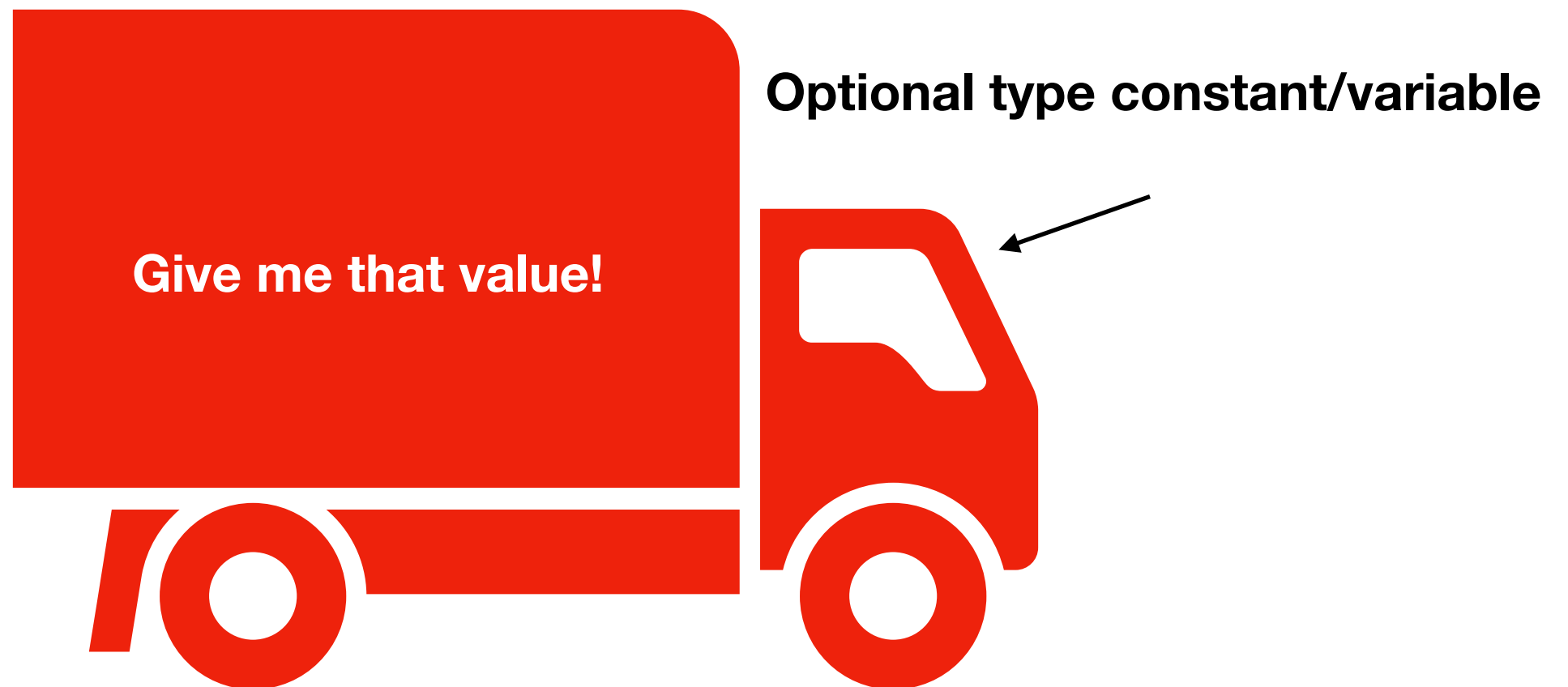
Force Unwrapping

- Se utilizan cuando se quiere acceder al valor.



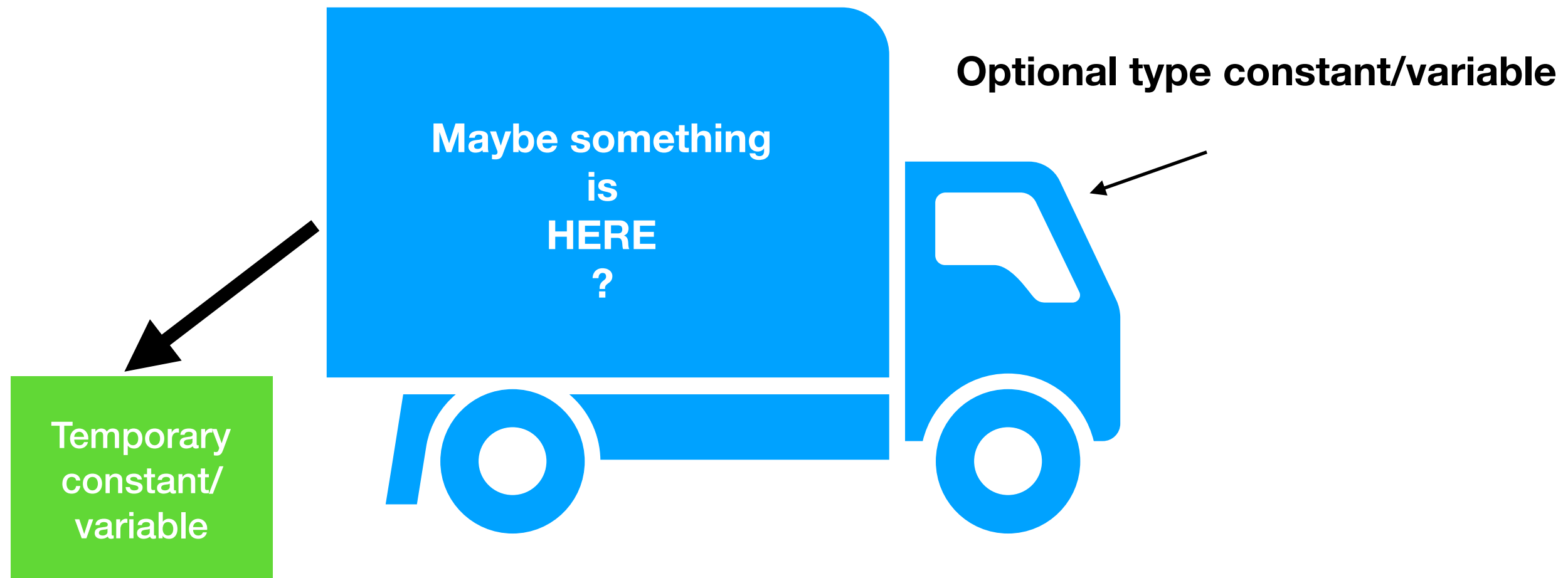
Implicitly Unwrapped

- Se utilizan cuando se quiere acceder al valor, siempre que la variable/constante sea requerida.



Optional Binding

- Se utilizan cuando se quiere acceder al valor de una manera más segura (sin accidentes).



Operadores Básicos

C con esteroides

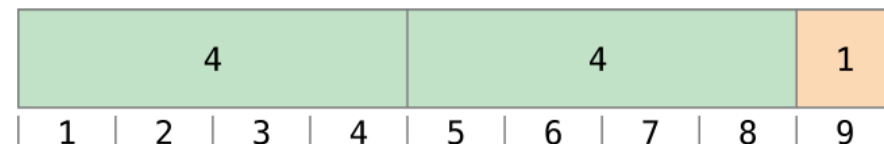
- Elimina errores comunes. (Overflow, = / ==).
- Operadores aritméticos y de comparación (+, - , *, / ...)
- Operadores logicos. (&&, ==, ||)
- a..< b

Terminología.

- Operandos = valores que los operadores afectan.
- Operadores Unitarios. (pre - post)fijo
- Operadores Binarios. (in)fijo
- Operadores Ternarios. (Solo 1).

Básicos

- **Operador de asignación.** Actualiza o inicializa una variable/constante.
- **Operadores de aritmética.** Adición, substracción , multiplicación y division.



- **Operador “Módulo”.**
- **Operador unitario Menos/Más.** (-) El signo de un valor numérico es modificado con este operador prefijo, (+) Regresa el valor tal cual del operando en el que actúa.

Operador de asignación compuesta

- Combinan el operador de asignación con algún otro operador.
- No regresan un valor, solo actual sobre la variable.

Operadores de comparación

- Regresan valor del tipo **Bool** para indicar si la comparación/enunciado es verdadero o falso.
- Por lo regular se utilizan en declaraciones condicionales (conditional statements).
- Bajo ciertas condiciones pueden utilizarse en Tuples.

- Equal to ($a == b$)
- Not equal to ($a != b$)
- Greater than ($a > b$)
- Less than ($a < b$)
- Greater than or equal to ($a >= b$)
- Less than or equal to ($a <= b$)

Operadores de rango

- Expresan limites de valores
- $a...b$ = valores que van desde **a** como limite inferior a **b** como limite superior.
- $a...<b$ = valores que van desde **a** como limite inferior a hasta un valor inferior a **b** como limite superior. (No se incluye **b**)
- $...b || a...$ = valores que solo tienen un limite, ya sea inferior o superior.

Operadores lógicos

- Combinan o modifican valores de tipo **Bool**.

- Logical NOT (! a)
- Logical AND (a && b)
- Logical OR (a || b)

Operador ternario

- *pregunta ? respuesta 1 : respuesta 2*

$a \neq \text{nil} \ ? \ a! : b$

- Nil - Coalescing
- (a: Int?) ?? (b: Int)
- El operador actúa en un optional.
- Se da un unwrapping del optional en caso de que no sea nil y se toma el valor.

Collection Types

3 tipos principales en swift

- Almacenan varios valores en una sola variable/constante.
- **Array**. Una colección ordenada.
- **Set**. Colección sin ordenar dónde cada valor es único.
- **Dictionaries**. Una llave tiene asociado un valor.

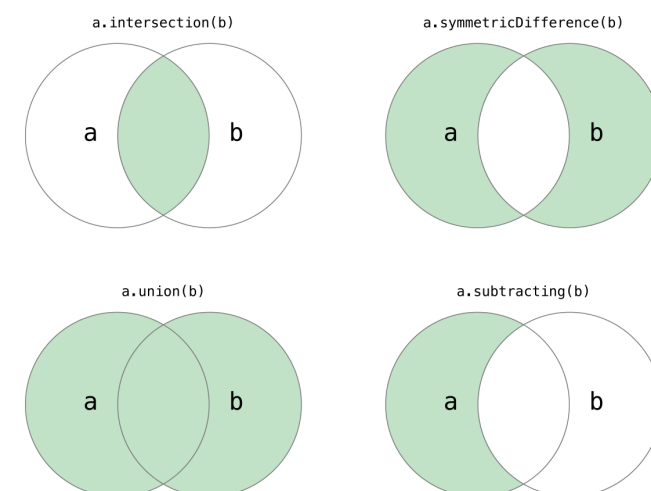
Array<Element>

- Regularmente se utiliza la notación [Element].
- Pueden ser combinados utilizando el símbolo de +.
- Puedes agregar más elementos con +=.

```
[value 1, value 2, value 3]
```

Set<Element>

- Regularmente se utiliza cuando el orden no es importante y cuando se necesita que un valor sea único dentro de la colección.
- Para ello el tipo del valor almacenado debe de apegarse al protocolo **Hashable**.



Dictionary<Key, Value>

- Contiene colecciones de relaciones entre KEYS (del mismo tipo) y Valores (del mismo tipo) sin un orden definido.
- Para ello el tipo del valor Key debe de apegarse al protocolo **Hashable**.

Control flow

4 tipos principales en swift

- **While Loops:** Para realizar actividades que se repiten.
- **If; guard.** Ejecutar código adecuado.
- **For-in.** Que facilita la iteración de una secuencia.
- **Switch.** Como el switch de todos los lenguajes con linaje C, pero con esteroides.

For-in

- Arrays
- Dictionaries
- Cualquier tipo de secuencias.

While loops

- Repite la ejecución de código hasta que la condición sea falsa.
- Son recomendados cuando el numero total de iteraciones no se sabe de antemano.
- El orden de la revisión de la condición puede ser al inicio (while) o al final (repeat-while)

```
while condition {  
    statements  
}
```

```
repeat {  
    statements  
} while condition
```

If | guard

- Ejecuta el código siempre y cuando la condición sea verdadera.
- Puedes utilizar la cláusula **else para ejecutar código** cuando la condición no se satisfaga.
- Puedes combinar varios ifs con una cláusula else + un if.\
- **Guard:** misma función que un if pero siempre viene acompañado de un else.

Switch

- Evita que se utilicen varios if-else.
- Debe de ser exhaustivo.
- No tiene un fallthrough implícito.
- Cada opción(case) debe de tener código.

```
switch (some value to consider) {  
  case value 1:  
    respond to value 1  
  
  case value 2,  
        value 3:  
    respond to value 2 or 3  
  
  default:  
    otherwise, do something else  
}
```

Intro to Swift 2

- Funciones.
- Classes, Structures & Enums
- Properties, Methods & Subscripts.
- Inherance

**Gracias por su
atención**

