Arrays, tuplas y sets

Arrays

> Son colecciones de valores agrupados como un solo valor.

Por ejemplo, azul, verde, rojo y amarillo son colores separados, pero los agrupamos en un `array` con el nombre de `colores`.

> Los arrays son mutables, es decir, podemos modificarlos constantemente.

```
var colors = ["Red", "Green", "Yellow", "Blue"]

//Otra forma de hacerlo es

let blue = "Azul"
let green = "Verde"
let red = "Rojo"
let yellow = "Amarillo"

var colorsInSpanish = [blue, green, red, yellow]
```

Estructura

Declaramos la variable y luego entre dos corchetes agregamos los valores, que van separados por comas.

Declaremos uno vacío:

```
var emptyArray:[String] = []

//Aca estamos creando un array que unicamente va a contener strings
```

Leer los valores de un array

Podemos ver un valor especifico de un array `indicando su posición`. Empezando desde cero, llamamos al array y entre corchetes, indicamos la posición del elemento a conocer.

Si leemos un item en una posición que no existe, rompe el programa.

```
var colores = ["azul", "rojo", "verde", "amarillo"]

print(colores[0]) //azul
print(colores[9]) //rompe el programa
```

Sets

💡 Son colecciones de valores también, pero con dos diferencias claves:
- No poseen un orden.
- Cada elemento es único.

Estructura

Se declaran con la palabra reservada `Set` y los contenidos van entre dos paréntesis y corchetes, de esta manera:

```
var legajo : Set = [112, 114, 115, 118]
```

Declaremos un set vacío:

```
var word = Set<String>()
//Generemos un set vacio, indicando que va a ser un set que contiene unicamente strings.
```

Más sobre sets

1. No importa el orden con el que ingresemos los datos, dado que esta estructura no cuenta con un orden.
2. Si agregamos elementos repetidos, cuando printeamos el set vamos a ver que cada elemento va a aparecer una única vez.

```
var colores : Set = ["Yellow", "Green", "Blue", "Red", "Red", "Blue"]

//Cuando lo printeamos van a aparecer los elementos en otro orden y una vez cada uno. Ej: ["Green", "Red", "Yellow", "Blue"]
```

Tuplas

💡 Tambien son colecciones de valores, pero con la siguientes diferencias:
- No se pueden agregar ni borrar datos de un tupla una vez creada.
- No se pueden cambiar los tipos de datos una vez creada.
- Las tuplas pueden contener múltiples tipos de datos

Estructura

> Las declaramos entre paréntesis y cada elementos tiene un nombre y un valor. Por ejemplo:

```
let email = (dominio: "usuario", direccion: "yahoo")
```

💡 Si intentamos agregar un valor a esa tupla, como puede ser "país", no nos va a dejar, porque **esa tupla fue creada como una colección de solo dos valores**: dominio y dirección.

Otro ejemplo de tupla:

Combinamos varios tipos de datos!

```
let dataUser = (pais: "argentina", edad: 26, esMayorDeEdad: true)
```

> Para acceder a los elementos de un tupla, podemos hacerlo con su posición o con su nombre.

```
let dataUser = (pais: "argentina", edad: 26, esMayorDeEdad: true)

print(dataUser.pais)
print(dataUser.0)
```