

# Tarea 12 - Programación Orientada a Objetos en Python

## Curso de Python

### Ejercicio 1

A lo largo de toda esta tarea vas a construir la clase `Date`. Empieza con el constructor, que recibe por parámetros el día (`day`), mes (`month`) y año (`year`). Los 3 parámetros son de tipo `int` y por defecto todos valen 1.

### Ejercicio 2

Configura el método `__str__()` para que muestre la fecha en formato `day / month / year`. Si el valor del día o el mes son menores a 10, mostrar el valor con un 0 delante. Por ejemplo, si `day = 8`, `month = 7` y `year = 1998`, entonces se debería mostrar `08 / 07 / 1998`.

En el caso del año, si el año es menor a 1000, mostrar con un cero delante; si es menor a 100, mostrar con 2 ceros delante; y si es menor a 10, mostrar con 3 ceros delante.

PISTA: Puedes crear una función que dado un número entero y el número de cifras que debe tener, rellene con ceros a la izquierda hasta completar el número de cifras indicado.

### Ejercicio 3

Implementa el método de instancia `.isLeap()` que diga si el año es bisiesto o no.

### Ejercicio 4

- Implementa un método de instancia `.totalMonthDays()` que diga el número de días del mes. Ten en cuenta que en los años bisiestos, Febrero tiene 29 días.
- Implementa el método de instancia `.validDate()` que determine si una fecha es válida. Modifica el constructor para que si la fecha introducida no es válida, devuelva un mensaje indicando “!!!La fecha introducida no es una fecha válida!!!”.

### Ejercicio 5

Implementa la propiedad `.monthName` que devuelva el nombre del mes en inglés. Por ejemplo, si nuestra fecha es `day = 8`, `month = 7` y `year = 1998`, la propiedad debe devolver `July`.

### Ejercicio 6

- Implementa el método estático `.areEqual()`, que dadas dos fechas diga si son iguales.
- Implementa el método estático `.isLater()`, que dadas dos fechas diga si la primera es posterior a la segunda.
- Implementa el método estático `.isPrevious()`, que dadas dos fechas diga si la primera es anterior a la segunda.

## Ejercicio 7

- Implementa el método de clase `.firstDayOfTheYear()` que dado un año cree un objeto `Date` con la fecha correspondiente al primer día del año indicado.
- Implementa el método de clase `.lastDayOfTheYear()` que dado un año cree un objeto `Date` con la fecha correspondiente al último día del año indicado.
- Implementa el método de instancia `.plusDay()` que incremente un día la fecha. Ten en cuenta que si estamos en el último día del mes y añadimos un día, tendremos que cambiar de mes (pasar al siguiente). Y lo mismo si estamos en el último día del año (tendremos que pasar al siguiente año).
- Implementa el método de instancia `.minusDay()` que decremente un día la fecha. Ten en cuenta que si estamos en el primer día del mes y restamos un día, tendremos que cambiar de mes (pasar al anterior). Y lo mismo si estamos en el primer día del año (tendremos que pasar al año anterior).

## Ejercicio 8

- Implementa el método de clase `.copy()` que dado un objeto `Date`, devuelva otro objeto `Date` con los mismos atributos.
- Implementa el método estático `.difference()` que dadas dos fechas devuelva el número de días que hay entre ellas.

## Ejercicio 9

Implementa el método de clase `.randomDate()` que cree una fecha aleatoria válida.

## Ejercicio 10

Implementa el método de clase `.toDate()` que dado un string con el formato "01 January 0001" devuelva el objeto fecha correspondiente a `day = 1`, `month = 1`, `year = 1`. Por ejemplo, si se pasa por parámetro 05 July 1985, el método debería devolver el objeto `Date` con los atributos `day = 5`, `month = 7`, `year = 1985`.