De Daniel San a Sr. Miyagi en Python

aprendiendo mediante katas

Irene Pérez Encinar

Head Of Technology @ Ringo Technologies

Madre de Águeda

https://github.com/IrenePEncinar

@irenuchi

INTRODUCCIÓN

¿Pero vamos a programar o a hacer kárate?

- Las "katas" son retos de programación
- Término acuñado en 1999 por Dave Thomas (The Pragmatic Programmer)
- Mejorar habilidades de programación mediante práctica y entrenamiento
- Medir tu nivel o aprender un lenguaje
- Prepararte para entrevistas técnicas / concursos

Codewars

- Entrenar distintos lenguajes de programación mediante katas
- Todas las katas incluyen tests
- Sistema de niveles: Kyu (8-1) y Dan (1-8)
- Comunidad de desarrolladores
- Comparar tu solución con las de otros
- Proponer katas una vez que tengas 75+ honor

Cómo lo uso yo

Modo 1

A ver qué echan hoy

Weekly Coding Challenges

Cómo lo uso yo

Modo 1

A ver qué echan hoy

Weekly Coding Challenges

Modo 2

Entrenamiento lenguaje

Ej: módulos, librerías

Cómo lo uso yo

Modo 1

A ver qué echan hoy

Weekly Coding Challenges

Modo 2

Entrenamiento lenguaje

Ej: módulos, librerías

Modo 3

Entrenamiento algoritmia

Tipo / naturaleza problema

Cómo lo uso yo

Modo 1

A ver qué echan hoy

Weekly Coding Challenges

Modo 2

Entrenamiento lenguaje

Ej: módulos, librerías

Modo 3

Entrenamiento algoritmia

Tipo / naturaleza problema

Modo 4

Subir de nivel

Juego / entretenimiento

Cómo lo uso yo

Modo 1
A ver qué echan hoy

Weekly Coding Challenges

Modo 2
Entrenamiento lenguaje

Ej: módulos, librerías

Modo 3
Entrenamiento algoritmia

Tipo / naturaleza problema

Modo 4
Subir de nivel

Juego / entretenimiento

Resolver problemas de forma creativa

Investigar

Documentación

Cómo lo uso yo

Modo 1
A ver qué echan hoy

Weekly Coding Challenges

Modo 2
Entrenamiento lenguaje

Ej: módulos, librerías

Modo 3
Entrenamiento algoritmia

Tipo / naturaleza problema

Modo 4
Subir de nivel

Juego / entretenimiento

Deliberate practice

Resolver problemas de forma creativa

Investigar

Documentación

TRES EJEMPLOS

Primer ejemplo - algoritmos

Thinkful - Logic Drills: Traffic light

- Controlar las luces del semáforo
- Definir una función que reciba un string con un color y devuelva el siguiente
- Por ejemplo:
    - update_light('green') # devuelve 'yellow'

Primer ejemplo - algoritmos

```
def update_light (current):
    next_light= {
        'green': 'yellow',
        'yellow': 'red',
        'red': 'green'
    }
    return next_light [current]
```

Primer ejemplo - algoritmos

```
def update_light (current):
    next_light= {
        'green': 'yellow',
        'yellow': 'red',
        'red': 'green'
    }
    return next_light [current]
```

```
def update_light (current):
    lights = ['green', 'yellow', 'red']
    current_i = lights.index(current)
    next_i = (current_i + 1) % len(lights)
    return lights [next_i]
```

Primer ejemplo - algoritmos

¿mejor?

Segundo ejemplo - batteries included

Convert string to camel case

- Convertir frase con palabras delimitadas por guión / barra baja a camel case
- Primera letra en mayúscula sólo si lo está en la frase original (Pascal case)
- Por ejemplo:
  - to_camel_case("the-stealth-warrior") # devuelve "theStealthWarrior"
  - to_camel_case("The_Stealth_Warrior") # devuelve "TheStealthWarrior"

Segundo ejemplo - batteries included

```
import re


def to_camel_case (text):

   words = re.split(r'-|_', text)
```

Segundo ejemplo - batteries included

```
import re


def to_camel_case (text):

   words = re.split(r'-|_', text)  # ['the', 'stealth', 'warrior']
```

Segundo ejemplo - batteries included

```
import re


def to_camel_case (text):
    words = re.split(r'-|_', text)  # ['the', 'stealth', 'warrior']
    return ''.join([w.capitalize() if i != 0 else w for i, w in enumerate(words)])
```

Segundo ejemplo - batteries included

```
import re


def to_camel_case (text):
    words = re.split(r'-|_', text)  # ['one', 'stealth', 'warrior']
    return ''.join([w.capitalize() if i != 0 else w for i, w in enumerate(words)])
```


80sKid.com

Segundo ejemplo - batteries included

```
import re

def to_camel_case (text):

  return re.sub('[_-](.)', lambda x: x.group(1).upper(), text)
```

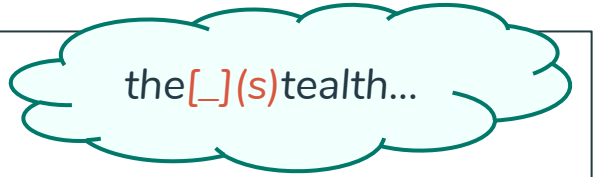
```
def to_camel_case (text):

  return text[0] + text.title().translate(str.maketrans(dict.fromkeys('-_')))[1:] if text else ''
```

Segundo ejemplo - batteries included

```
import re

def to_camel_case (text):
  return re.sub('[_-](.)', lambda x: x.group(1).upper(), text)
```




```
def to_camel_case (text):
  return text[0] + text.title().translate(str.maketrans(dict.fromkeys('-_')))[1:] if text else ''
```

Tercer ejemplo - Python mola

Kids and candles

- He invitado a los amigos de mi hija a su cumpleaños y les daré caramelos
- Todos los niños tienen que recibir el mismo número de caramelos sin que sobren
- No todos los niños invitados vendrán, pero sí es seguro que habrá 1 al menos

¿Cuál es el número mínimo de caramelos que tengo que comprar?

Tercer ejemplo - Python mola

Kids and candles

- He invitado a los amigos de mi hija a su cumpleaños y les daré caramelos
- Todos los niños tienen que recibir el mismo número de caramelos sin que sobren
- No todos los niños invitados vendrán, pero sí es seguro que habrá 1 al menos

¿Cuál es el número mínimo de caramelos que tengo que comprar?

Tercer ejemplo - Python mola

¿Y cómo calculo el mínimo común múltiplo de 1 a n?

- Descomponiendo en factores primos y multiplicando todos los factores comunes y no comunes elevados a la mayor potencia...
- A partir del máximo común divisor:

$$\text{mcm}(a,b) = \frac{a \cdot b}{\text{MCD}(a,b)}$$

Tercer ejemplo - Python mola

¿Y cómo calculo el mínimo común múltiplo de 1 a n?

- Descomponiendo en factores primos y multiplicando todos los factores comunes y no comunes elevados a la mayor potencia...
- A partir del máximo común divisor:

$$\text{mcm}(a, b) = \frac{a \cdot b}{\text{MCD}(a, b)}$$

math.**gcd**(a, b)

Tercer ejemplo - Python mola

```
from math import gcd
from functools import reduce


def candies_to_buy (kids_invited): amount_of_kids_invited):
    return reduce(lambda x, y: x*y / gcd(x,y), range(1, kids_invited+1))
```

Tercer ejemplo - Python mola

```
from math import gcd
from functools import reduce


def candies_to_buy (kids_invited): amount_of_kids_invited):
    return reduce(lambda x, y: x*y / gcd(x,y), range(1, kids_invited+1))
```

TypeError: 'float' object cannot be interpreted as an integer

Tercer ejemplo - Python mola

```
from math import gcd
from functools import reduce


def candies_to_buy (kids_invited): amount_of_kids_invited):
    return reduce(lambda x, y: int(x*y / gcd(x,y)), range(1, kids_invited+1))
```


#TeachersSeries
IT'S CALLED

▼ Basic Tests

- Test Passed
- Test Passed
- Test Passed
- Test Passed
- Test Passed
- Test Passed
- 329935398091387960548266698286672853021491 2 should equal 969071216477723170091280 0
- 201260592835746660267232823453701402490044416 should equal 591133442051411133755680800
- 312407341776354014834072243597373389403112974556867619398435671775959632070648 4 should equal 697203752297124771645338089353123035568 00
- 315531415194117565550531739741313269099016877422379147727113232937358224001214 3 should equal 704175789820096019361791470246654265923680 0

Tercer ejemplo - Python mola

```
from math import gcd
from functools import reduce


def candies_to_buy (kids_invited): amount_of_kids_invited):
    return reduce(lambda x, y: x*y // gcd(x,y), range(1, kids_invited+1))
```

https://docs.python.org/3/tutorial/floatingpoint.html

OTROS RECURSOS

Otros sitios

- Advent of Code - 1 reto por día de adviento ;)
- Project Euler - problemas de programación y matemáticas
- Edabit - retos en varios lenguajes y niveles
- CodeAbbey - problemas clasificados por categorías
- Code Jam - competición anual de programación de Google

Otros sitios

- [cyber-dojo] - resolver problemas + crear tests
- [Github karan Projects] - listado de problemas y soluciones
- [CodeChef] - concursos de programación mensuales
- [Sphere Online Judge] - problemas para mejorar en algoritmia

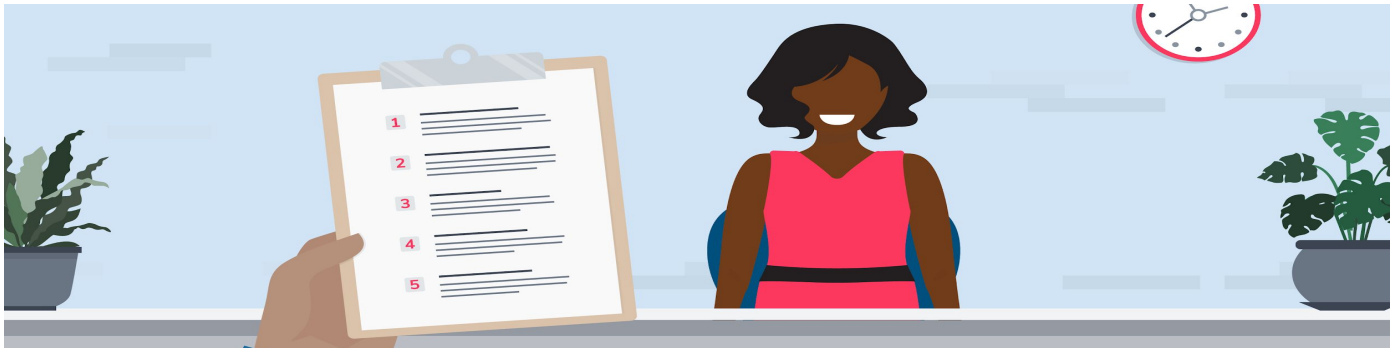
Otros sitios

- Python Challenge y su wiki
- Practice Python
- Python Exercises @ w3resource
- Python Basics
- Pynative

- Python koans - aprendiendo mediante TDD
- Python programming exercises
- Practice Python @ blogspot

Otros sitios - orientados a hiring

- Hackerrank
- CodeSignal
- Coderbyte
- Data Flair
- Testdome
- Tests4Geeks



Bonus: aprender + contribuir

¡GRACIAS!

¿preguntas?