

Q2P2: HEX to Color

ข้อกำหนด

- การเรียกใช้ฟังก์ชันเพื่อการทดสอบ ต้องอยู่ภายใต้เงื่อนไข `if __name__ == '__main__':` เพื่อให้สามารถ import ไปเรียกใช้งานจาก Script อื่น ๆ ได้อย่างถูกต้อง
- สามารถใช้ทุกเรื่องจากบทเรียนจนถึงสัปดาห์ปัจจุบันในการแก้ปัญหา
- เขียนรายละเอียดที่หัวไฟล์ดังนี้

```
#!/usr/bin/env python3
# ชื่อ (ไม่ต้องใส่นามสกุล)
# รหัสสนศ
# Sec00x
```

2) **100 คะแนน (Q2P2_6XXXXXXXXX.py) [Attachment]** การอ้างอิงสีที่ใช้ในเว็บแบบหนึ่งที่ใช้กันอย่างแพร่หลายคือ อ้างอิงตามเลขฐาน 16 ในรูปแบบ RGB รหัสผสมสีของสีแดง สีเขียว และสีน้ำเงิน ในแต่ละหลักแสดงถึงส่วนประกอบของแม่สีแสงได้แก่ สีแดง (r) สีเขียว (g) และสีน้ำเงิน (b) ซึ่งทุกสีสามารถประกอบขึ้นได้จากปริมาณที่แตกต่างกันของแม่สีแสงทั้งสาม ซึ่งมีการแสดงค่าในสองรูปแบบคือ

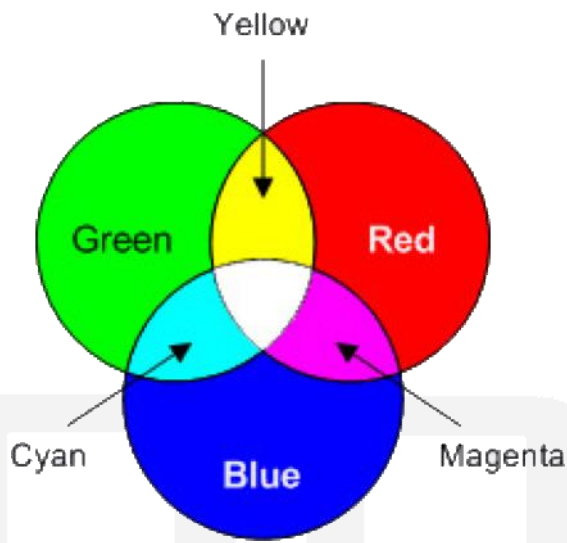
- เลขฐาน 16 3 หลัก ที่นำหน้าด้วยเครื่องหมาย pound sign '#' แต่ละหลักมีค่า 0 ถึง F (15)
- ของเลขฐาน 16 6 หลักที่นำหน้าด้วยเครื่องหมาย pound sign '#' ทุก ๆ 2 หลักจะมีค่าตั้งแต่ 00 ถึง FF (255)

ยกตัวอย่างกรณี 6 หลักเช่น กำหนดให้ สีแดง=36 สีเขียว=104 และสีน้ำเงิน=160 จะใช้เลขฐาน 16 6 หลัก มีค่าเท่ากับ 24, 68, A0 ตามลำดับ (อักษรตัวเล็กตัวใหญ่ไม่มีผล) โดยนำมาจัดเรียงติดกันโดยไม่เว้นวรรค แล้วนำหน้าด้วยเครื่องหมาย pound sign รวมเป็น #2468A0 ดังนั้น ค่าที่ต่ำที่สุดที่เป็นไปได้คือ #000 (3 หลัก) หรือ #000000 (6 หลัก) จะแทนสีดำ และค่าที่สูงที่สุดที่เป็นไปได้คือ #FFF (3 หลัก) หรือ #FFFFFF (6 หลัก) จะแทนสีขาว

หน้าที่ของคุณคือ ให้เขียนฟังก์ชัน `hex2color(hex_rgb)` เพื่อคืนค่าสตริงแทนสีที่สูงที่สุดจาก `hex_rgb` เช่น หากสีเขียว (green) มีค่าสูงที่สุด ให้แสดงค่า green ตามด้วย underscore '_' และค่าของสีที่แปลงเป็นเลขฐาน 10 แล้ว เช่น "#22AA00" มีค่าสีเขียวสูงสุด ดังนั้นจะคืนค่า "green_170" เนื่องจาก AA มีค่า 170 ในฐาน 10 ทั้งนี้ `hex_rgb` จะประกอบด้วยเครื่องหมาย # ตัวเลข 0-9, ตัวอักษร A-Z และตัวอักษร a-z จำนวน 3 หรือ 6 หลักเท่านั้น และจะไม่มี input ที่อยู่ในรูปแบบที่ไม่ถูกต้อง (ไม่จำเป็นต้องตรวจสอบ) และสามารถมีตัวพิมพ์เล็กและพิมพ์ใหญ่ปนกันได้ โดยให้คืนค่าชื่อสีตามรายละเอียดดังนี้

- red หาก ค่า r เป็นค่าที่สูงที่สุดจากทั้งสามค่า
- green หาก ค่า g เป็นค่าที่สูงที่สุดจากทั้งสามค่า
- blue หาก ค่า b เป็นค่าที่สูงที่สุดจากทั้งสามค่า
- cyan หากค่า g และ b มีค่าเท่ากันและเป็นค่าที่สูงที่สุด
- magenta หากค่า r และ b มีค่าเท่ากันและเป็นค่าที่สูงที่สุด
- yellow หากค่า r และ g มีค่าเท่ากันและเป็นค่าที่สูงที่สุด
- white หากทั้ง r และ g และ b มีค่าเท่ากับค่าสูงที่สุดที่เป็นไปได้คือ #FFF (3 หลัก) หรือ #FFFFFF (6 หลัก)
- black หากทั้ง r และ g และ b มีค่าเท่ากับค่าต่ำที่สุดที่เป็นไปได้คือ #000 (3 หลัก) หรือ #000000 (6 หลัก)

- grey หาก r และ g และ b มีค่าเท่ากันทั้งหมด แต่ไม่เท่ากับค่าสูงที่สุดที่เป็นไปได้ (white) และไม่เท่ากับค่าต่ำที่สุดที่เป็นไปได้ (black)



ข้อกำหนด

- ไม่อนุญาตให้ใช้ ฟังก์ชัน `int()` ในการแก้ปัญหา
- ไม่อนุญาตให้ `import module` เพิ่มเติมนอกจาก module `string`
- สามารถเรียกใช้ฟังก์ชัน `single_hex2dec(s)` เพื่อแปลงจากเลขฐาน 16 หนึ่งหลัก เป็นเลขฐาน 10 เมื่อ `s` คือตัวอักษรหนึ่งตัวในช่วง 0-9, A-F และ a-f
- สามารถใช้ ASCII table ด้านล่างในการประกอบการตัดสินใจ

0	NUL	16	DLE	32	SP	48	0	64	@	80	P	96	,	112	p
1	SOH	17	DC1	33	!	49	1	65	A	81	Q	97	a	113	q
2	STX	18	DC2	34	"	50	2	66	B	82	R	98	b	114	r
3	ETX	19	DC3	35	#	51	3	67	C	83	S	99	c	115	s
4	EOT	20	DC4	36	\$	52	4	68	D	84	T	100	d	116	t
5	ENQ	21	NAK	37	%	53	5	69	E	85	U	101	e	117	u
6	ACK	22	SYN	38	&	54	6	70	F	86	V	102	f	118	v
7	BEL	23	ETB	39	'	55	7	71	G	87	W	103	g	119	w
8	BS	24	CAN	40	(56	8	72	H	88	X	104	h	120	x
9	HT	25	EM	41)	57	9	73	I	89	Y	105	i	121	y
10	LF	26	SUB	42	*	58	:	74	J	90	Z	106	j	122	z
11	VT	27	ESC	43	+	59	;	75	K	91	[107	k	123	{
12	FF	28	FS	44	,	60	<	76	L	92	\	108	l	124	
13	CR	29	GS	45	-	61	=	77	M	93]	109	m	125	}
14	SO	30	RS	46	.	62	>	78	N	94	^	110	n	126	~
15	SI	31	US	47	/	63	?	79	O	95	_	111	o	127	DEL

Input

Output

#FF0000	red_255
#11f	blue_15
#22AA00	green_170
#e253E2	magenta_226

คำอธิบาย Test Case:

- Case 1: เหมือนตัวอย่าง ตรวจสอบแค่ชื่อสี
- Case 2-5: 3 digits ตรวจสอบแค่ชื่อสี
- Case 6-10: 3 & 6 digits ตรวจสอบแค่ชื่อสี
- Case 11: เหมือนตัวอย่าง ตรวจสอบทั้งชื่อสีและตัวเลข
- Case 12-15: 3 digits ตรวจสอบทั้งชื่อสีและตัวเลข
- Case 16-20: ทุกกรณีที่โจทย์ระบุ

Python Tutor Visualizer: <http://10.10.10.11/visualize.html>

Grader: <http://10.10.10.10>