

ComProg Cheat Sheet (Quiz 1)

คำสั่งพื้นฐาน	string s
<ul style="list-style-type: none"> • <code>print(_args_[,sep=?,end=?])</code> แสดงผล ออกทางหน้าจอโดยคั่นแต่ละ <code>_args_</code> ด้วย <code>sep</code> และจบด้วย <code>end</code> (หากไม่กำหนด <code>sep=" "</code> และ <code>end="\n"</code>) • <code>int(x)</code>, <code>float(x)</code>, <code>str(x)</code> <u>คืนค่า</u> <code>x</code> ที่เป็น จำนวนเต็ม จำนวนจริง และสตริงตามลำดับ • <code>abs(x)</code> <u>คืนค่า</u> สัมบูรณ์ของ <code>x</code> • <code>round(x[,d])</code> <u>คืนค่า</u> จำนวนที่ได้จากการปัดเศษ ของ <code>x</code> ให้มีจำนวนหลังจุดทศนิยม <code>d</code> จำนวน (0 จำนวนหากไม่กำหนด) • <code>min(_args_)</code>, <code>max(_args_)</code> <u>คืนค่า</u> มากสุด และน้อยสุดใน <code>_args_</code> • <code>range(start, stop[, step])</code> หรือ <code>range(stop)</code> <u>คืนค่า</u> ช่วงตั้งแต่ <code>start</code> ถึง <code>stop</code> โดยเพิ่มขึ้นทีละ <code>step</code> (หากไม่กำหนด <code>start = 0</code> และ <code>step = 1</code>) 	<ul style="list-style-type: none"> • <code>len(s)</code> <u>คืนค่า</u> ความยาวของสตริง <code>s</code> • <code>s.strip()</code> <u>คืนค่า</u> สตริง <code>s</code> ที่ตัด <code>space</code> หัวท้าย ออก • <code>s.split(sep)</code> <u>คืนค่า</u> ลิสต์ของสตริงที่แยก <code>s</code> ด้วย <code>sep</code> (ช่องว่างหากไม่กำหนด) • <code>s.lower()</code> <u>คืนค่า</u> <code>s</code> ที่เป็นตัวพิมพ์เล็กทั้งหมด • <code>s.upper()</code> <u>คืนค่า</u> <code>s</code> ที่เป็นตัวพิมพ์ใหญ่ทั้งหมด • <code>s.count(x)</code> <u>คืนค่า</u> จำนวนครั้งที่เจอ <code>x</code> ใน <code>s</code> • <code>s.find(x[,l[,r]])</code> <u>คืนค่า</u> ตำแหน่งแรกที่เจอ <code>x</code> ใน <code>s[l:r]</code> (หากไม่กำหนด <code>l=0</code> และ <code>r=len(s)</code>) หากไม่เจอจะ <u>คืนค่า</u> -1 • <code>s.rfind(x[,l[,r]])</code> <u>คืนค่า</u> ตำแหน่งสุดท้าย ที่เจอ <code>x</code> ใน <code>s[l:r]</code> (หากไม่กำหนด <code>l=0</code> และ <code>r=len(s)</code>) หากไม่เจอจะ <u>คืนค่า</u> -1 • <code>s.format(_args_)</code> <u>คืนค่า</u> สตริงที่แทนค่าแต่ ละตัวใน <code>_args_</code> ลงใน <code>s</code> • <code>s.replace(old, new[, cnt])</code> <u>คืนค่า</u> สตริง ที่แทนสตริง <code>old</code> ด้วยสตริง <code>new</code> ลงใน <code>s</code> จำนวน <code>cnt</code> ครั้ง (ถ้าไม่กำหนดจะแทนจนหมดสตริง) • <code>s.join(ls)</code> <u>คืนค่า</u> สตริงที่เชื่อม <code>ls</code> ด้วย <code>s</code> (<code>ls</code> เป็นลิสต์ของสตริง) • <code>s.isalpha()</code> <u>คืนค่า</u> <code>True</code> ถ้าสตริง <code>s</code> ประกอบไปด้วยตัวอักษรเท่านั้น และ <code>s</code> ไม่เป็น สตริงว่าง ไม่เช่นนั้นจะ <u>คืนค่า</u> <code>False</code> • <code>s.isdigit()</code> <u>คืนค่า</u> <code>True</code> ถ้าสตริง <code>s</code> ประกอบไปด้วยตัวเลขเท่านั้น และ <code>s</code> ไม่เป็นสตริง ว่าง ไม่เช่นนั้นจะ <u>คืนค่า</u> <code>False</code> • Escape characters : <code>\n</code>, <code>\r</code>, <code>\'</code>, <code>\"</code>, <code>\t</code>, <code>\\</code>
import math <ul style="list-style-type: none"> • <code>math.pi</code>, <code>math.e</code> <u>คืนค่า</u> คงที่ π และ <code>e</code> • <code>math.ceil(x)</code> <u>คืนค่า</u> <code>x</code> ที่ปัดเศษขึ้น • <code>math.floor(x)</code> <u>คืนค่า</u> <code>x</code> ที่ปัดเศษทิ้ง • <code>math.sqrt(x)</code> <u>คืนค่า</u> \sqrt{x} • <code>math.pow(x, y)</code> <u>คืนค่า</u> x^y • <code>math.log(x, b)</code> <u>คืนค่า</u> $\log_b x$ • <code>math.sin(x)</code> <u>คืนค่า</u> $\sin(x)$ (<code>x</code> หน่วยเรเดียน) • <code>math.cos(x)</code> <u>คืนค่า</u> $\cos(x)$ (<code>x</code> หน่วยเรเดียน) • <code>math.tan(x)</code> <u>คืนค่า</u> $\tan(x)$ (<code>x</code> หน่วยเรเดียน) • <code>math.degrees(x)</code> <u>คืนค่า</u> <code>x</code> จากเรเดียนเป็นองศา • <code>math.radians(x)</code> <u>คืนค่า</u> <code>x</code> จากองศาเป็นเรเดียน 	

list ls	Tips (ไม่เข้าใจไม่ต้องใช้ก็ได้)
<ul style="list-style-type: none"> list([x]) <u>คืนค่า</u>ลิสต์ของ x (หากไม่กำหนดจะสร้างลิสต์ว่าง) len(ls) <u>คืนค่า</u>ความยาวของลิสต์ ls ls[idx] <u>คืนค่า</u>ข้อมูลตำแหน่งที่ idx ของ ls ls[l:r:step] <u>คืนค่า</u>ลิสต์ย่อยของ ls ตั้งแต่ตำแหน่งที่ l จนถึง r-1 ไหลทีละ step ตัว ls.append(x) <u>แก้ค่า</u> ls โดยนำ x ไปใส่ท้ายลิสต์ ls.extend(L) <u>แก้ค่า</u> ls โดยนำลิสต์ L ไปต่อท้ายลิสต์ ls.insert(idx, x) <u>แก้ค่า</u> ls โดยแทรก x ที่ตำแหน่ง idx ls.pop([idx]) ลบข้อมูลตำแหน่งที่ idx (ตัวสุดท้ายหากไม่กำหนด) และ<u>คืนค่า</u>จำนวนครั้งที่เจอ x ใน ls ls.index(x[,l[,r]]) <u>คืนค่า</u>ตำแหน่งแรกที่เจอ x ใน s[l:r] (หากไม่กำหนด l=0 และ r=len(s)) หากไม่เจอจะ error ls.count(x) <u>คืนค่า</u>จำนวนครั้งที่เจอ x ใน ls ls.sort() <u>แก้ค่า</u>โดยเรียงลำดับข้อมูลจากน้อยไปมาก) sorted(ls) <u>คืนค่า</u>ลิสต์ ls ที่เรียงลำดับข้อมูลจากน้อยไปมาก) 	<ul style="list-style-type: none"> เราสามารถใส่คำสั่ง from math import <u>functions</u> ในกรณีที่เรารู้ฟังก์ชันที่จะใช้อยู่แล้ว (ไม่จำเป็นต้องใส่ math. อีกต่อไป) เราสามารถเขียน if-else ในบรรทัดเดียวกันได้ แต่ต้องมี else ด้วยเสมอ (ตัวอย่างไปหาเอาเอง) เราสามารถใช้ index() กับสตริงได้ด้วย ซึ่งมี ความแตกต่างกับ find() เพียงเล็กน้อย (มีอธิบายทั้งสองฟังก์ชันแล้ว) ควรมีชื่อตัวแปรที่ใช้เป็นประจำอยู่และสื่อความหมายเหมือนกัน เช่น n แทนจำนวนอินพุต s แทนสตริง และ ls แทนลิสต์ จำอันไหนไม่ได้มี help(ชื่อฟังก์ชัน) ช่วยได้
	ข้อควรระวัง (สำคัญ !!!) <ul style="list-style-type: none"> <u>ระวัง</u>เรื่องการตั้งชื่อตัวแปรซ้ำกับ reserved word ของ python (มีอะไรบ้างไปหาเอง) <u>อย่าลืม</u> strip() ทุกครั้งที่รับอินพุตเป็นสตริง (ถ้าโจทย์ไม่บอกว่าไม่ต้องใช้) <u>ระวัง</u>การ print ข้อความ <u>อย่า</u>พิมพ์ผิด เช่น card เป็น cards จะทำให้เสียคะแนนแบบน่าเสียดาย (ก็อปวางเลยชัวร์สุด) สตริงแก้ค่า<u>ไม่ได้</u> แต่กำหนดค่าใหม่ได้ ดูดีๆ ว่าฟังก์ชันไหน<u>คืนค่า</u> ฟังก์ชันไหน<u>แก้ค่า</u> <u>อย่าลืม</u>เอา debug ออก สำคัญที่สุด <u>อ่านโจทย์ดีๆ</u>

สงสัยอะไรลองอ่านใน <https://github.com/UtopiaBeam/ComProg> หรือถามใน FB : Natchapol Srisang