

## ComProg Cheat Sheet (Quiz 1)

คำสั่งพื้นฐาน	string s
<ul style="list-style-type: none"> <li>• <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>)</li> <li>• <code>int(x)</code>, <code>float(x)</code>, <code>str(x)</code> <u>คืนค่า</u> <code>x</code> ที่เป็น จำนวนเต็ม จำนวนจริง และสตริงตามลำดับ</li> <li>• <code>abs(x)</code> <u>คืนค่า</u> สัมบูรณ์ของ <code>x</code></li> <li>• <code>round(x[,d])</code> <u>คืนค่า</u> จำนวนที่ได้จากการปัดเศษ ของ <code>x</code> ให้มีจำนวนหลังจุดทศนิยม <code>d</code> จำนวน (0 จำนวนหากไม่กำหนด)</li> <li>• <code>min(_args_)</code>, <code>max(_args_)</code> <u>คืนค่า</u> มากสุด และน้อยสุดใน <code>_args_</code></li> <li>• <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>)</li> </ul>	<ul style="list-style-type: none"> <li>• <code>len(s)</code> <u>คืนค่า</u> ความยาวของสตริง <code>s</code></li> <li>• <code>s.strip()</code> <u>คืนค่า</u> สตริง <code>s</code> ที่ตัด <code>space</code> หัวท้าย ออก</li> <li>• <code>s.split(sep)</code> <u>คืนค่า</u> ลิสต์ของสตริงที่แยก <code>s</code> ด้วย <code>sep</code> (ช่องว่างหากไม่กำหนด)</li> <li>• <code>s.lower()</code> <u>คืนค่า</u> <code>s</code> ที่เป็นตัวพิมพ์เล็กทั้งหมด</li> <li>• <code>s.upper()</code> <u>คืนค่า</u> <code>s</code> ที่เป็นตัวพิมพ์ใหญ่ทั้งหมด</li> <li>• <code>s.count(x)</code> <u>คืนค่า</u> จำนวนครั้งที่เจอ <code>x</code> ใน <code>s</code></li> <li>• <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</li> <li>• <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</li> <li>• <code>s.format(_args_)</code> <u>คืนค่า</u> สตริงที่แทนค่าแต่ ละตัวใน <code>_args_</code> ลงใน <code>s</code></li> <li>• <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)</li> <li>• <code>s.join(ls)</code> <u>คืนค่า</u> สตริงที่เชื่อม <code>ls</code> ด้วย <code>s</code> (<code>ls</code> เป็นลิสต์ของสตริง)</li> <li>• <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></li> <li>• <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></li> <li>• Escape characters : <code>\n</code>, <code>\r</code>, <code>\'</code>, <code>\"</code>, <code>\t</code>, <code>\\</code></li> </ul>
<b>import math</b> <ul style="list-style-type: none"> <li>• <code>math.pi</code>, <code>math.e</code> <u>คืนค่า</u> คงที่ <math>\pi</math> และ <code>e</code></li> <li>• <code>math.ceil(x)</code> <u>คืนค่า</u> <code>x</code> ที่ปัดเศษขึ้น</li> <li>• <code>math.floor(x)</code> <u>คืนค่า</u> <code>x</code> ที่ปัดเศษทิ้ง</li> <li>• <code>math.sqrt(x)</code> <u>คืนค่า</u> <math>\sqrt{x}</math></li> <li>• <code>math.pow(x, y)</code> <u>คืนค่า</u> <math>x^y</math></li> <li>• <code>math.log(x, b)</code> <u>คืนค่า</u> <math>\log_b x</math></li> <li>• <code>math.sin(x)</code> <u>คืนค่า</u> <math>\sin(x)</math> (<code>x</code> หน่วยเรเดียน)</li> <li>• <code>math.cos(x)</code> <u>คืนค่า</u> <math>\cos(x)</math> (<code>x</code> หน่วยเรเดียน)</li> <li>• <code>math.tan(x)</code> <u>คืนค่า</u> <math>\tan(x)</math> (<code>x</code> หน่วยเรเดียน)</li> <li>• <code>math.degrees(x)</code> <u>คืนค่า</u> <code>x</code> จากเรเดียนเป็นองศา</li> <li>• <code>math.radians(x)</code> <u>คืนค่า</u> <code>x</code> จากองศาเป็นเรเดียน</li> </ul>	

list ls	Tips (ไม่เข้าใจไม่ต้องใช้ก็ได้)
<ul style="list-style-type: none"> <li>• <code>list([x])</code> <u>คืนค่า</u>ลิสต์ของ x (หากไม่กำหนดจะสร้างลิสต์ว่าง)</li> <li>• <code>len(ls)</code> <u>คืนค่า</u>ความยาวของลิสต์ ls</li> <li>• <code>ls[idx]</code> <u>คืนค่า</u>ข้อมูลตำแหน่งที่ idx ของ ls</li> <li>• <code>ls[l:r:step]</code> <u>คืนค่า</u>ลิสต์ย่อยของ ls ตั้งแต่ตำแหน่งที่ l จนถึง r-1 ไล่ทีละ step ตัว</li> <li>• <code>ls.append(x)</code> <u>แก้ค่า</u> ls โดยนำ x ไปใส่ท้ายลิสต์</li> <li>• <code>ls.extend(L)</code> <u>แก้ค่า</u> ls โดยนำลิสต์ L ไปต่อท้ายลิสต์</li> <li>• <code>ls.insert(idx, x)</code> <u>แก้ค่า</u> ls โดยแทรก x ที่ตำแหน่ง idx</li> <li>• <code>ls.pop([idx])</code> <u>แก้ไข</u>ลิสต์โดยลบข้อมูลตำแหน่งที่ idx (ตัวสุดท้ายหากไม่กำหนด) และ<u>คืนค่า</u>จำนวนที่ลบไป</li> <li>• <code>ls.index(x[,l[,r]])</code> <u>คืนค่า</u>ตำแหน่งแรกที่เจอ x ใน <code>s[l:r]</code> (หากไม่กำหนด <code>l=0</code> และ <code>r=len(s)</code>) หากไม่เจอจะ error</li> <li>• <code>ls.count(x)</code> <u>คืนค่า</u>จำนวนครั้งที่เจอ x ใน ls</li> <li>• <code>ls.sort()</code> <u>แก้ค่า</u>โดยเรียงลำดับข้อมูลจากน้อยไปมาก)</li> <li>• <code>sorted(ls)</code> <u>คืนค่า</u>ลิสต์ ls ที่เรียงลำดับข้อมูลจากน้อยไปมาก)</li> </ul>	<ul style="list-style-type: none"> <li>• เราสามารถใช้คำสั่ง <code>from math import _functions_</code> ในกรณีที่เรารู้ฟังก์ชันที่จะใช้อยู่แล้ว (ไม่จำเป็นต้องใส่ <code>math.</code> อีกต่อไป)</li> <li>• เราสามารถเขียน <code>if-else</code> ในบรรทัดเดียวกันได้ แต่ต้องมี <code>else</code> ด้วยเสมอ (ตัวอย่างไปหาเอาเอง)</li> <li>• เราสามารถใช้ <code>index()</code> กับสตริงได้ด้วย ซึ่งมี ความแตกต่างกับ <code>find()</code> เพียงเล็กน้อย (มีอธิบายทั้งสองฟังก์ชันแล้ว)</li> <li>• จริงๆ แล้ว <code>False = 0</code> และ <code>True = 1</code> (ใช้เหมือนตัวเลขได้เลย)</li> <li>• จำอันไหนไม่ได้มี <code>help(ชื่อฟังก์ชัน)</code> ช่วยได้</li> </ul>
	ข้อควรระวัง (สำคัญ !!!)
	<ul style="list-style-type: none"> <li>• <u>ระวัง</u>เรื่องการตั้งชื่อตัวแปรซ้ำกับ reserved word ของ python (มีอะไรบ้างไปหาเอง)</li> <li>• <u>อย่าลืม</u> <code>strip()</code> ทุกครั้งที่รับอินพุตเป็นสตริง (ถ้าโจทย์ไม่บอกว่าไม่ต้องใช้)</li> <li>• <u>อย่า</u>พิมพ์ผิด เช่น <code>file</code> เป็น <code>flie</code></li> <li>• สตริงแก้ค่า<u>ไม่ได้</u> แต่กำหนดค่าใหม่ได้</li> <li>• ดูดีๆ ว่าฟังก์ชันไหน<u>คืนค่า</u> ฟังก์ชันไหน<u>แก้ค่า</u></li> <li>• <u>อย่าลืม</u>เอา debug ออก</li> <li>• สำคัญที่สุด <u>อ่านโจทย์ดีๆ</u></li> </ul>

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