|  |
| --- |
| 教師：曾智義 課程名稱：基礎程式設計 學期：112-1 |
| 一、第06週課堂練習 |
| 綜合運用至目前所學(基本運算、變數、字串、List、Dict、Tuple、Set） |
| 二、個人/成員 |
| 林彧頎\_資工系\_A1105505 |
| 三、議題規劃 |
| 在如今時代發展得愈快速，生活腳步愈來愈快，當人愈在意自身的工作時，往往會不小心忽略了自己的身體，人的健康所需除了六大營養以外，在衛生福利部國民健康署的建議中，其中有兩項分別為「我的餐盤」與「每日飲食建議量(含每日熱量建議)」，以上數據可以幫助人，對於每日的飲食需求做每日的審視，並且依照自己每日的身體數據，可以進行相對應的調整，因此我將設計出一個每日列表，記錄我的每日的餐盤建議與每日飲食建議。  我將透過結合W3所得出的熱量計算，搭配食物分配，幫助人由於想每日要吃什麼而煩惱時，可以有更營養且符合自己熱量所需的選擇。  在W6中我將結合前面所做的成果，針對每日飲食進行多樣化的改善，我將新增我的飲食列表，並建立一個三餐的飲食編排，這將會使我每日對於飲食的多樣性會有更不一樣的改變，並且又能同時兼顧飲食健康。  (我也將在設計中模擬自身的數據)       |  |  | | --- | --- | | 每天早晚一杯奶 | 每天早上和晚上要各喝一杯牛奶（約240毫升），來補充蛋白質和國人十分缺乏的鈣質。自助餐店比較少在賣鮮奶，所以記得一整天要喝兩杯牛奶，才會更均衡！ | | 每餐水果拳頭大 | 每一餐都要吃到一個拳頭大的水果。自助餐店若沒有提供水果的話，別忘了在餐間或餐後再吃些水果！選擇各類多種顏色的水果，以攝取豐富的膳食纖維、維生素C和有抗氧化功能的植化素。 | | 菜比水果多一點 | 每餐吃的蔬菜量要比一個拳頭多一些。在自助餐店可以選擇多種顏色不同的蔬菜（如炒空心菜、玉米筍、滷紅蘿蔔、海帶等），也可以夾一些和蔬菜一起烹調的主菜（如洋蔥豬柳、匏瓜炒肉片等）來達到所需的蔬菜量，並攝取到各種不同顏色蔬菜所含的植化素。 | | 飯跟蔬菜一樣多 | 飯和蔬菜吃的份量差不多。在自助餐多選擇糙米、紫米或五穀飯等全穀雜糧，可以補充到更多的膳食纖維與其他營養素。 | | 豆魚蛋肉一掌心 | 每餐的蛋白質食物，大約要吃到一個掌心的大小。在自助餐店可以夾一些烹調方式較簡單清淡的豆類（如醬燒豆干、嫩豆腐、炒毛豆仁等）、海鮮類（如炒花枝、烤鯖魚等）、蛋類（如蒸蛋、番茄炒蛋等）或肉類（如煎豬里肌、白斬雞等）。記得盡量別選擇油炸的主菜哦！ | | 堅果種子一茶匙 | 每一餐的堅果種子類，約是一茶匙或一個拇指指節的大小；也可以一次吃掉一整天的建議量，約是一湯匙或一個拇指的大小。可以自己準備無調味的堅果在餐間肚子餓時吃，營養會更加均衡！ | |
| 四、問題定義  (自行設計問題，問題可列舉或敘述方式，自由模擬，以下為例子) |
| 目標一   1. 需要透過撰寫熱量對應的卡路里以方便計算，此可以幫助未來做其他的熱量數據進行操作(使用dict結合list的方法) 2. 需要計算我自身的熱量 3. 並依照我的熱量，寫出相對應的每日飲食建議與每日餐盤建議 4. 並計算我一天建議總共所食用的食物總數量 5. 列出我可能的飲食規劃表，並對應是否有符合上述計算出的選項 6. 顯示結果   目標二   1. 製作三餐飲食規劃 2. 使用tuple與set 3. 使三餐的飲食不僅僅是符合熱量需求的，並且可以隨每一次重新啟動程式，皆有不同的三餐飲食組合 |
| 五、程式碼設計  (將執行的程式碼貼入並註解，綜合運用所學，以下為例子) |
| 1. import math  2. #工作量、體重(字典定義)  3. #將key改為tuple(m(中度),l(輕度),h(重度))  4. #體重(l),工作(l)  5. ll=('體重過輕者','輕度工作')  6. ml=('體重正常者','輕度工作')  7. hl=('體重過重者','輕度工作')  8.  9. lm=('體重過輕者','中度工作')  10. mm=('體重正常者','中度工作')  11. hm=('體重過重者','中度工作')  12.  13. lh=('體重過輕者','重度工作')  14. mh=('體重正常者','重度工作')  15. hh=('體重過重者','重度工作')  16.  17. dict\_kcal={ll:35,  18. ml:30,  19. hl:25,  20.  21. lm:40,  22. mm:35,  23. hm:30,  24.  25. lh:45,  26. mh:40,  27. hh:35}  28.  29. #每日熱量  30. print('每日熱量建議')  31. #字典輸出  32. #體重過輕者  33. print('-----------------')  34. print('體重過輕者所需熱量:')  35. print('輕度工作: '+str(dict\_kcal[ll])+' kcal')  36. print('中度工作: '+str(dict\_kcal[lm])+' kcal')  37. print('重度工作: '+str(dict\_kcal[lh])+' kcal')  38. #體重正常者  39. print('-----------------')  40. print('體重正常者所需熱量:')  41. print('輕度工作: '+str(dict\_kcal[ml])+' kcal')  42. print('中度工作: '+str(dict\_kcal[mm])+' kcal')  43. print('重度工作: '+str(dict\_kcal[mh])+' kcal')  44. #體重過重者  45. print('-----------------')  46. print('體重過重者所需熱量:')  47. print('輕度工作: '+str(dict\_kcal[hl])+' kcal')  48. print('中度工作: '+str(dict\_kcal[hm])+' kcal')  49. print('重度工作: '+str(dict\_kcal[hh])+' kcal')  50. print('-----------------')  51. #顯示我的熱量計算  52. print('我的體重:49kg')  53. w=49  54. print('我的身高:158cm')  55. h=158  56. print('我的工作量:正常')  57. print('我的bmi:正常')  58. print('我所需的熱量:'+str(dict\_kcal[mm]\*w))  59. print('-----------------')  60. #顯示我的每日飲食建議量  61. print('請參考以下\n每日飲食建議量')  62. #使用變數，可幫助之後有其他大卡的數據可以直接修正  63. a=3  64. b=5  65. c=1.5  66. d=3  67. e=2  68. f=5  69. print('由於我是1715大卡\n(因此我大約需要1800大卡):')  70. print('全榖雜糧類:'+str(a)+'份')  71. print('豆魚蛋肉類:'+str(b)+'份')  72. print('乳品類 :'+str(c)+'份')  73. print('蔬菜類 :'+str(d)+'份')  74. print('水果類 :'+str(e)+'份')  75. print('油脂類 :'+str(f)+'份')  76. print('total :'+str(a+b+c+d+e+f)+'份')  77. print('-----------------')  78. print('以下為每日餐盤建議')  79. print('每天早晚一杯奶\n')  80. print('每餐水果拳頭大\n')  81. print('菜比水果多一點\n')  82. print('飯跟蔬菜一樣多\n')  83. print('豆魚蛋肉一掌心\n')  84. print('堅果種子一茶匙\n')  85. print('-----------------')  86. print('以下為我的食物組合:')  87. print('請選擇最適合的~')  88. #食物組合與熱量(使用字典與list)  89. #list(名稱,份,大卡)  90. 全榖雜糧類1=['飯',3,840]  91. 全榖雜糧類2=['飯',2,560]  92.  93. 豆魚蛋肉類1=['肉',4,160]  94. 豆魚蛋肉類2=['肉',5,200]  95.  96. 乳品類1=['牛奶',1.5,225]  97. 乳品類2=['牛奶',2,300]  98.  99. 蔬菜類1=['菜',1,25]  100. 蔬菜類2=['菜',3,75]  101.  102. 水果類1=['水果',1,60]  103. 水果類2=['水果',2,120]  104.  105. 油脂類1=['油',4,240]  106. 油脂類2=['油',5,300]  107.  108. food\_set1={'全榖雜糧類':全榖雜糧類1,'豆魚蛋肉類':豆魚蛋肉類2,'乳品類':乳品類1,  109. '蔬菜類':蔬菜類2,'水果類':水果類2,'油脂類':油脂類1}  110. food\_set2={'全榖雜糧類':全榖雜糧類2,'豆魚蛋肉類':豆魚蛋肉類1,'乳品類':乳品類2,  111. '蔬菜類':蔬菜類1,'水果類':水果類1,'油脂類':油脂類2}  112. #印出結果並自行比較與選擇  113. print('食物組合1\n(名稱、份數、熱量(大卡))')  114. print(food\_set1.values())  115. print('總熱量:'+str(food\_set1['全榖雜糧類'][2]+food\_set1['豆魚蛋肉類'][2]+  116. food\_set1['蔬菜類'][2]+food\_set1['乳品類'][2]+  117. food\_set1['水果類'][2]+food\_set1['油脂類'][2])+'kcal')  118.  119. print('食物組合2\n(名稱、份數、熱量(大卡))')  120. print(food\_set2.values())  121. print('總熱量:'+str(food\_set2['全榖雜糧類'][2]+food\_set2['豆魚蛋肉類'][2]+  122. food\_set2['蔬菜類'][2]+food\_set2['乳品類'][2]+  123. food\_set2['水果類'][2]+food\_set2['油脂類'][2])+'kcal')  124. print('-----------------')  125. print('3餐食物隨機組合:')  126. #list與set的使用(並同時做早餐的重新排序)  127. list\_rice=['白飯','麵線']  128. set\_rice=set(list\_rice)  129. #測試update  130. set\_rice.update(['白飯'])  131. set\_rice.update(['紫米飯'])  132. set\_rice.update(['五穀雜糧飯'])  133. list\_rice=list(set\_rice)  134. #print(list\_rice) #測試是否隨機產生(是)  135.  136. list\_meat=['炒豬肉絲','煎魚','雞肉丁','丁香魚','肉丸','扁食']  137. set\_meat=set(list\_meat)  138. list\_meat=list(set\_meat)  139.  140. list\_milk=['牛奶','羊奶','起司','優格','乳酪蛋糕']  141. set\_milk=set(list\_milk)  142. list\_milk=list(set\_milk)  143.  144. list\_veg=['炒高麗菜','燙地瓜葉','苦瓜鹹蛋','炒A菜','炒筍子']  145. set\_veg=set(list\_veg)  146. list\_veg=list(set\_veg)  147.  148. list\_fruit=['蘋果','西瓜','奇異果','香蕉','蓮霧','水蜜桃']  149. set\_fruit=set(list\_fruit)  150. list\_fruit=list(set\_fruit)  151.  152. list\_oil=['酪梨','堅果','橄欖油','玄米油']  153. set\_oil=set(list\_oil)  154. list\_oil=list(set\_oil)  155.  156.  157. tuple\_breakfast=('早餐','400卡')  158. tuple\_lunch=('中餐','400卡')  159. tuple\_dinner=('晚餐','400卡')  160.  161. dict\_meals={}  162.  163. dict\_meals[tuple\_breakfast]=(list\_rice[0],list\_meat[1],list\_milk[2],list\_veg[0],list\_fruit[1],list\_oil[2])  164. print('早餐:'+str(dict\_meals[tuple\_breakfast]))  165. setb={list\_rice[0],list\_meat[1],list\_milk[2],list\_veg[0],list\_fruit[1],list\_oil[2]}  166.  167. #再次重新排序(中餐)  168. set\_rice=set(list\_rice)  169. list\_rice=list(set\_rice)  170.  171. set\_meat=set(list\_meat)  172. list\_meat=list(set\_meat)  173.  174. set\_milk=set(list\_milk)  175. list\_milk=list(set\_milk)  176.  177. set\_veg=set(list\_veg)  178. list\_veg=list(set\_veg)  179.  180. set\_fruit=set(list\_fruit)  181. list\_fruit=list(set\_fruit)  182.  183. set\_oil=set(list\_oil)  184. list\_oil=list(set\_oil)  185.  186. dict\_meals[tuple\_lunch]=(list\_rice[1],list\_meat[2],list\_milk[0],list\_veg[1],list\_fruit[2],list\_oil[0])  187. print('中餐:'+str(dict\_meals[tuple\_lunch]))  188. setl={list\_rice[1],list\_meat[2],list\_milk[0],list\_veg[1],list\_fruit[2],list\_oil[0]}  189.  190. #再次重新排序(晚餐)  191. set\_rice=set(list\_rice)  192. list\_rice=list(set\_rice)  193.  194. set\_meat=set(list\_meat)  195. list\_meat=list(set\_meat)  196.  197. set\_milk=set(list\_milk)  198. list\_milk=list(set\_milk)  199.  200. set\_veg=set(list\_veg)  201. list\_veg=list(set\_veg)  202.  203. set\_fruit=set(list\_fruit)  204. list\_fruit=list(set\_fruit)  205.  206. set\_oil=set(list\_oil)  207. list\_oil=list(set\_oil)  208.  209. dict\_meals[tuple\_dinner]=(list\_rice[2],list\_meat[1],list\_milk[0],list\_veg[2],list\_fruit[1],list\_oil[0])  210. setd={list\_rice[2],list\_meat[1],list\_milk[0],list\_veg[2],list\_fruit[1],list\_oil[0]}  211. #判斷set集合比較運算子  212. print('晚餐:'+str(dict\_meals[tuple\_dinner]))  213. print('早餐與中餐有重複嗎?',setb>=setl)  214. print('中餐與晚餐有重複嗎?',setd>=setl)  215. print('早餐與晚餐有重複嗎?',setb>=setd)  216.  217. |
| 六、執行結果  (將正確執行的結果貼入，文字為主截圖為輔，以下為例子) |
| |  |  |  | | --- | --- | --- | | (結果1)    (結果2) |  |  | | 由以上的結果可以得出，我們的熱量計算是中度工作、正常體重，可以得出  所需的熱量為1715大卡，而在參考每日餐盤與飲食建議後，我們的食物組合有兩種，  分別對應的是:食物組合1為1760大卡，食物組合2為1345大卡，由此可知最適合  我的食物組合為食物組合1(1760大卡(近1715大卡))。  並且於3餐的隨機組合中，我刻意讓程式碼跑兩次，  以展現每次程式碼執行時皆會有隨機的3餐組合，這是使用set與list的結合。  並且在本次的作業中，有將之前dictionary的值組key更正為tuple的型態。  最後我將它們以set分析集合的方法來判斷3餐是否有重複  (下一頁繼續) |  |  | |
| 七、學習意涵詮釋  (可分為兩個部分：一、學到什麼程式語法 二、學到解決什麼問題，以下為例子) |
| 1.直接將整數變數放前+字串變數印會抱錯，要加上str(int變數)  2. Tuple Type: 多個元素組成一組  3. Tuple Type與List Type的差異: List Type:宣告後可以變更存放的資料->[]/ Tuple:宣告後不行更改->()  4. Tuple可以用來當Dictionary type 的key(因為Tuple值不能再被異動->Key為不能異動的值)(但其實可以透過將tuple轉list做修改再轉回tuple就可以修改了)  5. dict中list當value,而tuple當key  6.set是集合型別(candy={'a','b'}#(會打亂順序ex'a','b'變'b','a') )  7. set():轉set/ list():轉list/ tuple():轉tuple  8. set可以將list轉為set:1.字元不會打散(因為list)2.於update()裡用[]可以增加Set的資料  ->因此可以使用list來新增資料，再用set()來轉成Set，再以update([])來新增元素  9.set也可以做到刪除重複元素(set可以幫助list重複資料刪除)  10. 集合型別的運算(交集&聯集...)  11.我透過這次的學習，發現了python中可以用到許多種的 料處哩，並且這些功能之間還可以達到1+1大於2的結合效果，我不僅僅將每一種的使用方法，從過去的作業中不斷地延伸，也同時學習到更好的融會貫通他們之間的方法，聖致這項時做，讓我認為，飲食真的是人生活中不可或卻的一部份，我們需要透過每日飲食的攝取，才可以有更好的生活與健康! |
| 八、參考說明  (列出所參考資料並說明如何應用或修改其內容，修改ChatGPT亦可，以下舉例) |
| 每日餐盤-衛生福利部國民健康署  <https://www.hpa.gov.tw/Pages/Detail.aspx?nodeid=4687&pid=11680>  每日飲食建議與熱量建議-衛生福利部國民健康署  <https://www.hpa.gov.tw/Pages/Detail.aspx?nodeid=544&pid=728> |