

## 02 Skilaverkefni -15%

FORR2FA05BU – dictionary og föll

Setjið upp valmynd þar sem hægt er að velja úr verkefnum hér að neðan. Hafið líka hætta möguleika í valmyndinni. Ef valið er eitthvað annað en það sem er í valmynd koma skilaboðin „Ekki rétt val“ og valmyndin birtist aftur. Forritið hættir ekki keyrslu fyrr en notandi velur að hætta.

**Munið að merkja ykkur lausnina og kommenta kóðann vel. Einnig þarf að koma skýrt fram þegar þið eruð að prenta út, hvað þið eruð að prenta út hverju sinni. Allir liðir gilda jafnt og valmynd hefur einnig vægi.**

### 1 - Lítrar eða gallon

Búið til föll sem breyta gallonum í lítra og öfugt, `def gallon_i_litra(g)`, `litrar_i_gallon(l)`, föllin eiga að hafa skilagildi – return.

### 2 - Hraðatakmörk

Búið til fall sem tékkar aksturshraða bíla, `def hversuHratt(hradi)`.

Forritið prentar síðan út af eftirfarandi eftir hve hratt viðkomandi ók

- Ef hraðinn er lægri en 30 km/klst á að prentast út: *Ókídókí*
- Fyrir hverja byrjaða 5 km yfir hámarkshraðann (30 km/klst) á að gefa notanda refsistig og síðan prenta út hversu mörg refsistig viðkomandi fær t.d.:  
*Refsistig = 3*
- Ef notandi fær meira en 8 refsistig á einnig að prentast út: *Þú missir ökuskrteinið*

### 3 - Veldisreikningur

Hannið fallið `def Veldi(tala, veldi)` sem tekur inn tvær jákvæðar heiltölur sem færibreytur. Fallið skrifar svo á skjáinn veldarununa og reiknar töluna í veldinu og prentar á skjáinn. Dæmi um keyrslu fallsins: Ef `Veldi(2,3)` er keyrt mundi fallið skrifa á skjáinn

$2*2*2=8$

Annað dæmi um keyrslu fallsins: `Veldi(3,4)` mundi fallið skrifa á skjáinn

$3*3*3*3=81$

Fallið á skila réttri niðurstöðu þó að veldin 0 og 1 séu sett sem færibreytur.

### 4- Teningakast

Hannið forrit sem hefur fallið `kasta(oft)` sem tekur inn heiltölu sem segir til um hversu oft á að kasta. Fallið á að kasta tening eins oft og færibreytan segir til um, leggur saman öll köstin og skilar summunni – nota return.

## 5 - Listar

Hannið forrit sem hefur fallið `def eins(listi1, listi2)` sem tekur inn tvo lista. Finnið síðan allar tölur sem enda á 2 í öllum listunum og setjið í nýjan lista og skilið honum— nota `return`.

## 6 - Talnadictionary

Takið tvö dictionary og breytið því í eitt með heildafjölda hluta. Notið dictionary

```
dict1= {"stólar":10, "borð":20, "myndir":2}
dict2= {"stólar":11, "borð":70, "myndir":80}
```

sem eru notuð í þessu dæmi.

útkoman er þá:

```
dictsaman={'stólar': 21, 'borð': 90, 'myndir': 82}
```

## 7 – Einkunnir

Vinnið með `nemadict` hér að neðan þar sem lykilinn er nafn nemandans og gildið er dictionary af einkunnum. Gerið eitt fall `def medaleinkunnStæForr(dict)` sem reiknar meðaleinkunn hvers nemanda í stærðfræði og forritun og skilar `dict` með nafni og meðaleinkunn fyrir hvern og einn nemanda

```
nemadict ={"Jón Hallsson":{"STÆR":7, "ENSK":5, "FORR":10, "VERS":8},
           "Hallur Halldórsson":{"STÆR":2, "ENSK":8, "FORR":9, "VERS":9},
           "Dísa Jónsdóttir":{"STÆR":3, "ENSK":7, "FORR":5, "VERS":6},
           "Anna Eiríksdóttir":{"STÆR":10, "ENSK":6, "FORR":1, "VERS":7},
           "Kristín Vilhjálmisdóttir":{"STÆR":9, "ENSK":5, "FORR":6, "VERS":5},
           "Dáníel Sindrason":{"STÆR":7, "ENSK":7, "FORR":5, "VERS":10}, }
```

## 8. Spurningar

Svarið eftirfarandi spurningum með því að prenta út svarið með `print` undir þessum valmöguleika í valmyndinni ykkar.

- Hvenær er eðlilegra að nota tuple heldur en lista í forritun og hvers vegna. Hverjir eru kostir og gallar tuple og lista.
- Hver er grundvallar munur á gagnagrindunum list og dictionary í Python?