

Genel Kurallar:

*Kod pratiđinizi Google Colab üzerinden yapabilirsiniz. Google arama butonunda yazdığınızda direkt karşınıza çıkacaktır. Gmail hesabınız giriş yaparak ile kendi Colab hesabınızı oluşturabilirsiniz.

*Aşağıda belirttiđimiz açıklamalarda, # işareti ile başlayan kısımlar yazmanız gereken kodu açıklamaktadır.

3.Gün: Booleans(Mantıksal Veri Tipleri) ve Şartlı İfadeler

1. Sayısal bağımsız değışkeni negatifse -1, pozitifse 1 ve 0 ise 0 döndüren “sign” adında bir fonksiyon tanımlayın.

#

2. Yağmurlu bir havaya hazırlıklı olmak için;

- Şemsiyem olmalı

- Yağmur çok şiddetli değilse kapüşonum da olmalı

Aksi durumda yağmur yağmadığı ve iş günü de olmadığı sürece bir sorun yaşamayacağım.

Aşağıdaki fonksiyon, bu mantık ile bir Python ifadesine dönüştürülmüştür. Fakat bu kodda bir hata var. Bu hata nedir?

```
def hava_icin_hazirlik(semseye_al, yagmur_seviyesi, baslik_al, is_gunu):
```

```
    # Bu kodu değıştirmeyin, hedefimiz hatayı bulmak, kodu yeniden yazmak değil!
```

```
    return semseye_al or yagmur_seviyesi < 5 and baslik_al or not yagmur_seviyesi > 0 and is_gunu
```

Bu girdilerin değerlerini, hava_icin_hazirlik fonksiyonunun yanlış yanıtı döndürdüğü bir durumu temsil edecek şekilde değıştirin.

```
semseye_al = True
```

```
yagmur_seviyesi = 0.0
```

```
baslik_al = True
```

```
is_gunu = True
```

Yukarıdaki değişkenlerin geçerli değerleri göz önüne alındığında işlevin ne döndürdüğünü kontrol edin.

```
guncel_deger = hava_icin_hazirlik(semsiye_al, yagmur_seviyesi, baslik_al, is_gunu)
print(guncel_deger)
```

3. Aşağıdaki `is_negative` işlevi doğru şekilde uygulanıyor; verilen sayı negatifse `True`, aksi halde `False` değerini döndürüyor. Ancak, kod olması gerekenden daha ayrıntılı. Bu işlevdeki kod satırı sayısını 75% oranında azaltabiliriz. Sadece tek bir satır kod kullanarak eşdeğer bir kod bulup bulamayacağınıza bakın ve `concise_is_negative` fonksiyonuna yazın.

```
def is_negative(number):
```

```
    if number < 0:
```

```
        return True
```

```
    else:
```

```
        return False
```

```
def concise_is_negative(number):
```

```
    pass # Kodunuzu buraya yazacaksınız, yukarıdaki işlemi buraya tek satırda belirtin.
```

4. Boolean değişkenleri 'ketcap', 'hardal' ve 'sogan', bir müşterinin sosisli sandviç içinde isteyip istemediğini yiyecekleri temsil eder. Müşterinin siparişiyle ilgili bazı evet veya hayır sorularına karşılık gelen bir dizi boolean işlevi uygulamak istiyoruz. Örneğin;

```
def sogansız(ketcap, hardal, sogan):
```

```
    #soğansız bir sandviç döndürecek.
```

```
    return not sogan
```

Aşağıdaki her bir kodu, istenen şekilde tamamlayın.

```
def hersey_olsun (ketcap, hardal, sogan):
```

```
    #içinde herşey olan bir sandviç çıktısı olacak şekilde tamamlayın.
```

```
    pass
```

```
def sadece_sosisli (ketcap, hardal, sogan):
```

```
#içinde sosis dışında hiçbir şey olmayan bir sandviç çıktısı döndürün.
```

```
    pass
```

```
def yalnızca_bir_sos_olsun (ketcap, hardal, sogan):
```

```
#içinde sadece ketçap ya da hardal olan ama ikisi birlikte olmayan bir sandviç çıktısı döndürün.
```

```
    pass
```