

FUN WITH CALENDAR: 10

Alright guys in the previous videos you had seen how we can get the input and check if it is a valid input and then once the input is valid will proceed with how to check for the weak day that is the day of the week on which that day particular falls. So we had inputted the year anything from and after nineteen seventy is allowed and we had inputted the month and allowed range of values is from one to twelve and we had inputted the date. Valid date factor varies based on if it is a leap year the month the date, date month year based on all these factors your fact that the date is valid or not varies. We had seen how the variation is all these we had seen in the previous videos also i had explained in the slides as well that was we had coded from the slides you can try coding it in your own and you can see how the similarities and differences are between your approach and my approach exist and you can discuss all this in discussion form just that given that there are so many constrains it may appear that its very complex or something but nothing it's just that my way of wiring the conditions is like this there can be multiple ways you try wiring by yourself you will understand that is very easy alright so by now you have got the input of by year month and date that is needed for our built in function that built in function is found in the library calendar so let me import that first. So that i can use it that is my ultimate purpose of this exercise, i will import calendar right so i had imported calendar so now i will be using it so this functionality week day presenting calendar is what will return as what is the day of the week given a date right? so we had seen how the signature of it we had seen how to use in a pre requisites videos will revisit it now so one thing that we have noticed it, it returns the index it says zero for Monday one for Tuesday and so on till six for Sunday this is how it returns your input it returns the output in this fashion. Zero for Monday to six for Sunday, zero for Monday, one for Tuesday, two for Wednesday and so on till six for Sunday This is how the calendar dot week day the week day functionality that is present in the calendar library works but that is difficult for humans to interpret so we need to convert that as well so first thing what we have to do is we have to get that index whatever it is returning so let me say day index day index is return by this thing calendar dot week day calendar dot week day of see the arguments of year comma month comma day, day is nothing but the date year comma month comma date ok so we had inputted it we had send the inputs for this functionality to work so it will work it will return an index which will say which day it is zero is for Monday one is for Tuesday so on till six is for Sunday but that is not easy for humans right so we will have to get the day from this day index so let me say day i have to get it from day index i will say get day i have to get day form day index day index right see there is a warning symbol which says that undefined that just said the name i haven't defined what this has to do so let us define it. So let us just define here its fine define get day let me define it so what is the list of days let me say list of days the encoding is its starts from Monday zero is for Monday right so let me say Monday Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday this is the order in which encodes right Monday is encoded as zero, Tuesday is encoded as one and Wednesday is encoded as two and so on till Sunday is encoded as six. So i have created a list of days basically so i guess you would understand you could figure out what will be my next step, i will just return ok for this i need to have the day index right so day index i need to have the day index so i will just return the corresponding value from this list given by day index. List

of days of day index this is the most simplest definition i guess you are able to understand it the encoding is zero is for Monday one is for Tuesday and so on till six is for Sunday this i show the lists have their indexing mechanism so we are creating the list of days and whatever is the index that is being passed we will pass the corresponding element in that index as the output that is if the if the passed index is three then zero one two three its zero one two three its Thursday, Thursday will be taken as the output so we will get the name of the day here so that is obtained now we have to print the answer, let me print i will print a very good message i will say date this is how humans do it right by slash will separate it date month then we have another slash and year this is our human friendly format and i am printing in the human friendly format only i will say it falls on this particular day that is captured in the variable day so this falls on this particular day this is how the working is simple will just give a very quick summary and then will go ahead with running of this code see i am getting an input of year i am getting an year i ma inputting year if it is after or from nineteen seventy i agree that input and i am inputting month that should be between the range one to twelve and i am inputting date and based on the factors if it is a leap year what is the month that is being enter? I determine if the entered date is a valid date or not then based on this i have i am inputting some date month year right so based on this things i am now passing it to the built in functionality calendar dot week day in calendar library you have the functionality week day we are passing this and it is returning a index it's not returning the name of the day its returning the index so for getting the name of the day i am using the list mechanism and i am getting day and after getting the day i am printing the this particular date month year falls on this day. This is how the working is so let me save it and let me run ok so its asking me to enter a year so let me say two thousand let me enter a month let me say four april let me enter a date let me say twenty five ok twenty five four two thousand falls on Tuesday i got an output. So like this you can check for any dates once your programs is ready running is see as yo could see maximum it would have taken half a second that's it its all the delay by which you are giving your input that's it. If you hardcore the input if it is to be read from the code itself it will be really within in the fractions of seconds so this is really really very fast ok by which you can know the day of the week given a date alright guys so we had seen one task will see another task that we can do using these cool python libraries in the next part of the video.