

Problema 1:

#Teste 1: se capital passado for menor ou igual a 0, deve apresentar o erro 'Capital precisa ser maior que zero.'

#Teste 2: a taxa não pode ser maior do que 0 e nem menor do que 1, caso seja deverá apresentar o erro 'Taxa precisa estar entre 0 e 1.'

#Teste 3: o número de períodos precisa ser maior do que ser, caso contrário deverá apresentar o erro 'Período precisa ser maior que zero.'

#Teste 4: caso for passado uma string em qualquer um dos parâmetros deverá apresentar o type error.

#Teste 5: passando os valores 1,1,2 respectivamente temos o resultado de 3

#Melhorias: Existe a falta de verificações de erros para tipos não numéricos e a verificação da taxa poderia ser apenas maior que zero, poderia ser utilizado if, elif e else ao invés de uma repetição de 3 ifs.

Problema 2:

#Teste 1: se os valores passado forem 2,3,5 e 6 deverá retornar 16

#Teste 2: caso for passado um número e uma string, deverá apresentar o erro 'Incompatibilidade de tipos. Verificar parâmetros'

#Teste 3: caso for passado um número e uma lista como parâmetro, deverá apresentar o erro 'Incompatibilidade de tipos. Verificar parâmetros'

#Teste 4: caso for passado dois números em formato de string deverá apresentar o erro 'Incompatibilidade de tipos. Verificar parâmetros'

#Teste 5: caso for passado um número inteiro e um número float, deverá retornar o valor correto.

#Melhorias: a soma dentro do try poderia ser melhorado utilizando +=

Problema 3:

#Teste 1: se nenhum valor for passado a função irá retornar um dicionário vazio.

#Teste 2: caso for passado uma chave duplicada, deverá apresentar erro de keyword repeated.

#Teste 3: caso for passado uma string como chave e uma string como valor, deverá apresentar o erro 'expression cannot contain assignment, perhaps you meant "=="'

#Teste 4: caso for passado um número como chave e uma string como valor, deverá apresentar o erro 'expression cannot contain assignment, perhaps you meant "=="'

#Teste 5: caso for passado uma variável como chave e uma string ou número como valor deverá retornar o resultado.

#Melhorias: poderia ser corrigido problemas de UTF – 8, já que caracteres especiais não estão sendo exibidos, poderia também ser criado um arquivo json.

Problema 4:

#Teste 1: se ausência for menor do que zero ou maior que 80, deverá apresentar erro de 'Ausências entre 0 e 80'.

#Teste 2: se a nota final for menor do que zero ou maior que 10, deverá apresentar o erro de 'Nota Final entre 0 e 10'

#Teste 3: se a ausência for maior que 40 deverá apresentar 'Reprovado por falta'

#Teste 4: se a nota final for menor que 5 deverá apresentar 'Reprovado por nota'

#Teste 5: se a nota final for menor que 7 deverá apresentar 'Reprovado, em regime de recuperação'

#Teste 6: caso contrário se passar a nota 8 deverá apresentar 'Aprovado'

#Melhorias: poderia ser melhorado as verificações das condicionais no geral

Problema 5:

#Teste 1: se for passado a profissão médico e idade qualquer, deverá retornar autorizado vacinação.

#Teste 2: caso for passado uma profissão que não esteja na lista de prioridade, mas a idade é maior que 69 deverá retornar autorizado a vacinação.

#Teste 3: se a profissão for professor e a idade maior que 47, deverá retornar autorizado a vacinação.

#Teste 4: caso nenhuma das exigências forem atendidas a função deverá retornar não autorizado por enquanto.

#Melhorias: poderia ser feito apenas uma lista ou dicionário com a chave profissionais da saúde e os valores como profissões

Problema 6:

#Teste 1: se a data passada não for no formato especificado, deverá retornar erro.

#Teste 2: se a data for passada sem ser como string, deverá retornar erro 'leading zeros in decimal integer literals are not permitted'.

#Teste 3: se o mês passado como parâmetro for 0 ou maior que 13 deverá apresentar o erro 'Mês inválido'.

#Melhorias: faltam inúmeras verificações de filtro da data recebida e também poderia ser utilizado recursos do próprio datetime para mostrar o mês em extenso.

Problema 7:

#Teste 1: se for passado valores numéricos para o args, deverá retornar o valor correto.

#Teste 2: caso for passado uma string como valor para o args, deverá apresentar type error.

#Teste 3: se o valor passado para o kwargs for uma chave valor com valor numérico, deverá retornar o valor correto.

#Teste 4: caso for passado uma string como valor para o kwargs, deverá apresentar type error.

#Teste 5: caso for passado valores tanto para o args como para kwargs, somente será retornado valores do args.

#Melhorias: poderia ser feita verificações de filtro onde só aceitassem tipos numéricos

Problema 8:

#Justificativa: Em debate com a equipe, achamos que a melhor função para ser testada automaticamente seja a primeira, uma vez que ela já possui os inputs e as variáveis internamente o que facilita a verificação dos valores dessas variáveis e o filtro do retorno da função.

Equipe:

Caio Santoro

Evandro Dametto

Vagner Bohm

