functions

9/10

April 29, 2024

[ ]:
```
#even or odd
def evenorodd(number):
    if number%2==0:
      return 'even'
    else:
      return 'odd'
evenorodd(5)
```

[ ]: 'odd'

[ ]:
```
#maxOfTwo
def maxOfTwo(num1,num2):
    if num1>num2:
        return num1
    else:
        return num2
maxOfTwo(5,6)
```

[ ]: 6

[ ]:
```
#Leap year checker
def isLeapYear(year):
    if (year%4==0 and year%100!=0 or year%400==0):
        return 'leap year'
    else:
        return 'Not leap year'
isLeapYear(1900)
```

[ ]: 'Not leap year'

[ ]:
```
#factorial Calculator
def factorialCalculator(num):
    if num==1:
        return num
    else:
        return num*(factorialCalculator(num-1))
factorialCalculator(3)
```

[ ]: 6

[ ]:
```
def gcd(a,b):
    while b!=0:
        a,b = b,a%b
        return a
gcd(3,6)
```

[ ]: 6

[ ]:
```
#absolute value
def absoluteValue(num):
    if num<0:
        return -num
    else:
        return num
absoluteValue(-4)
```

[ ]: 4

[ ]:
```
#temperature converter
def convertTemp(celcius):
    Farenheit = (celcius*(9/5))+32
    return Farenheit

convertTemp(30)
```

[ ]: 86.0

[ ]:
```
#Grade calculator
def calculateGrade(score):
    if score >= 90:
        return 'A'
    elif score >= 80:
        return 'B'
    elif score >=70:
        return 'C'
    elif score >= 60:
        return 'D'
    elif score <= 59:
        return 'F'
calculateGrade(89)
```

[ ]: 'B'

[ ]:
```
# calcculate sum of squares
def sumOfSquares(n):
    sum = 0
    for number in range(0,n+1):
        sum += number*number
    return sum
sumOfSquares(4)
```

[ ]: 30

[ ]:
```
#Quadratic equation solve
def solveQuadratic(a,b,c):
        discriminant = b**2-4*a*c
        if discriminant > 0:
            root1 = (-b+(discriminant)**0.5/(2*a))
            root2 = (-b-(discriminant)**0.5/(2*a))
            return (root1,root2)
        elif discriminant == 0:
            root = -b/2*a
            return (root,root)
        elif discriminant < 0:
            return 'None'
solveQuadratic(1,4,0)
```

a=1, b=4, c=0 should return 4, 0

[ ]: (-2.0, -6.0)

[ ]:

20