

@author: Shada Al-Wakkal, Git:Shada12354

Sammanfattning

Färgkod: variabel, funktion, loop & modul.

Denna Python-kod består av olika funktioner som utför olika matematiska operationer och beräkningar. Koden täcker ämnen som binär konvertering, kombinationer, procentandelar, Fibonacci-sekvens, slantsingling och flera andra matematiska beräkningar. Dessutom importerar den modulerna `math` och `random` för användning i vissa funktioner.

Kod

Moduler

Skriptet importerar modulerna `math` för de matematiska operationerna, och `random` för att generera slumpmässiga nummer.

BINARY CONVERSION

- Denna del visar exempel på binär konvertering med hjälp av Pythons inbyggda funktioner `bin()` och `int()`. Med `bin()` konverteras ett heltal till ett binärtal, och motsatt funktion med `int()`.

N CHOOSE K COMBINATIONS

- Definierar en funktion `combination(n, k)` som beräknar kombinationer (n choose k) med hjälp av faktoriell beräkning.
- Använder `math.factorial()` för att beräkna faktorialer av olika tal och returnerar det totala antalet kombinationer.

PERCENTAGE

- Definierar funktionen `percentCal(n, part, roundOff)` för att beräkna procentandelen av en del jämfört med en helhet.
- Ger alternativ att avrunda resultatet nedåt, avrunda uppåt eller lämna det oskuret beroende på användarens val.

FIBONACCI SEQUENCE

- Definierar funktionen `fib(n)` som genererar Fibonacci-serien upp till gränsen `n`.
- Använder en `'while'`-loop för att generera serien genom att upprepa beräkningen av de närmaste talen i sekvensen.

COINFLIP

- Skriptet välkomnar användaren till ett enkelt myntkastningsspel.
- Användaren uppmanas att välja "heads" eller "tails".
- Skriptet genererar ett slumpmässigt nummer, antingen 1 eller 2, vilket representerar resultatet av myntkastningen.
- Om det slumpmässiga numret är 1 betraktas resultatet som "heads"; om det är 2 betraktas resultatet som "tails".
- Användarens val jämförs sedan med det slumpmässiga resultatet för att avgöra om användaren gissade rätt eller inte.
- Ett meddelande skrivs ut för att ange om användaren vann eller förlorade spelet.

BINARY CONVERSION

- Funktionen `bin2dec(bin)` tar en binär sträng som input och omvandlar den till ett decimaltal genom att iterera över varje bit i strängen. Den använder positionsvärdena och "bitwise" operationer för att beräkna det decimala värdet.
- Funktionen `dec2bin(dec)` omvandlar ett decimaltal till binär representation genom att utföra upprepade divisioner med basen 2 och samtidigt konstruera den binära strängen.

BINTABLE

- Funktionen tar en parameter `dec`, som är det övre gränsvärdet för tabellen, det vill säga det högsta decimala talet som ska inkluderas i tabellen.
- Inuti funktionen används en `'for'`-loop som itererar över decimala tal från 1 till och med det givna värdet `dec`.
- Funktionen `dec2bin(n)` används för att konvertera det aktuella decimala talet till dess binära motsvarighet. Den binära representationen lagras i variabeln `bn`.
- Sedan skrivs det aktuella decimala talet och dess binära representation ut i tabellformat med hjälp av `print()`-funktionen.

MULTITABLE

- Funktionen `multiTable(n)` genererar en multiplikationstabell genom att använda två loopar för att multiplicera alla tal från 1 till n med varandra och skriva ut resultaten i tabellform.

PYTHAGORAS SATS

- Funktionen `hypotenuse(a, b)` beräknar hypotenusan av en rätvinklig triangel med sidorna a och b genom att använda Pythagoras sats.
- Funktionen `pythagoreans(mx)` genererar Pythagoras triplar genom att använda tre loopar för att hitta kombinationer av a, b och c som uppfyller Pythagoras sats.

NEWTON

- Funktionen `newton(r, n)` använder Newtons metod för att approximera roten av ett tal genom upprepade iterationer. Den beräknar en ny approximation genom att använda formeln för en tangentlinje till kurvan vid den föregående approximationen.