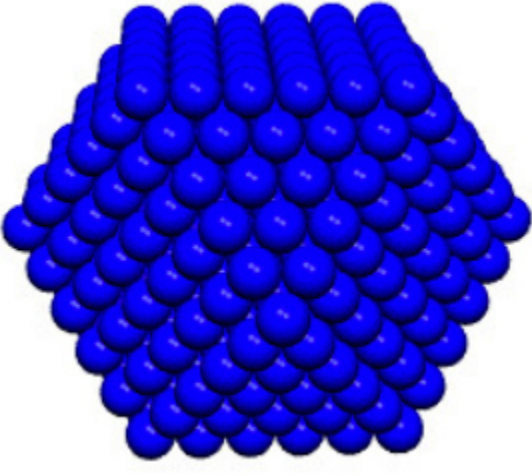


Sphere Packing with Julia

by Kirby Urner



My pedagogy around teaching any computer language is to have the target language e.g. Python on the front burner but then compare and contrast with one or more back burner languages e.g. Julia, Clojure...

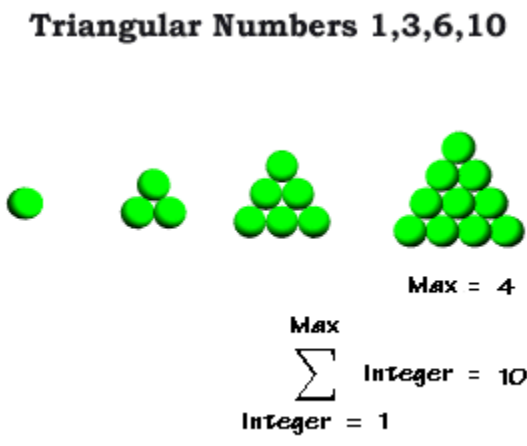
[Replit makes it easy](#) to walk my talk in this respect.

I'm eager to demonstrate starting a new Jupyter Notebook and having a choice among languages: Ju for Julia, Pyt for Python, Er for R.

The focus is ball packing: triangular and square numbers (figurate on a plain (plane)); tetrahedral and [cuboctahedral](#) (= icosahedral) in space.

A student is co-developing a lexical skill (coding language) with a graphical skill (visualizing balls packing in space).

Triangular Numbers



```
In [1]: # Triangular Numbers
# https://oeis.org/A000217
tri = n -> ((n == 0) ? 0 : n + tri(n - 1)) # recursive
println("A000217:", [tri(x) for x in 0:10])
```

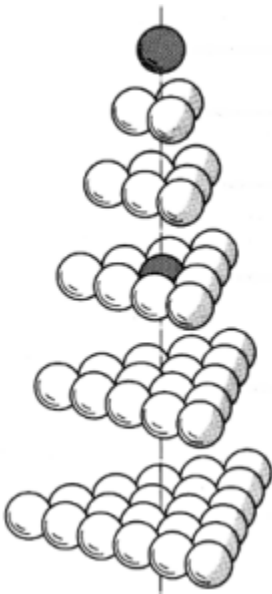
A000217:[0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55]

Alternative function syntax, as well as a different algorithm:

```
In [2]: function tri2(n)
        Int(n * (n + 1) / 2)
end
println("A000217:", [tri2(x) for x in 0:10])
```

A000217:[0, 1, 3, 6, 10, 15, 21, 28, 36, 45, 55]

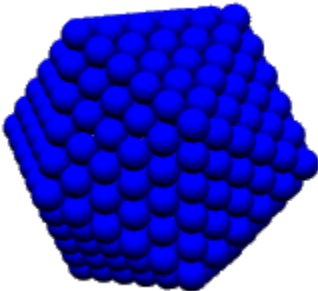
Tetrahedral Numbers



```
In [3]: # https://en.wikipedia.org/wiki/Tetrahedral_number
# https://oeis.org/A000292
function tetra(n)
    Int((n * (n + 1) * (n + 2)) / 6)
end
println("A000292:", [tetra(x) for x in 1:10])
```

A000292:[1, 4, 10, 20, 35, 56, 84, 120, 165, 220]

Cuboctahedral Numbers (= Icosahedral Numbers)



```
In [4]: # https://oeis.org/A005901
cubocta = n -> ((n == 0) ? 1 : 10 * n * n + 2)
println("A005901:", [cubocta(x) for x in 0:10])
```

A005901:[1, 12, 42, 92, 162, 252, 362, 492, 642, 812, 1002]

Just as tetrahedral numbers may be viewed as cumulative triangular numbers, so may we accumulate successive layers (1, 12, 42, 92...) in the cuboctahedral packing (1, 13, 55, 147...).

```
In [5]: # https://oeis.org/A005902
accum_cubocta(n) = (
    (n == 0) ? 1 :
    cubocta(n) + accum_cubocta(n-1)) # recursive
println("A005902:", [accum_cubocta(x) for x in 0:10])
```

A005902:[1, 13, 55, 147, 309, 561, 923, 1415, 2057, 2869, 3871]

```
In [6]: # http://www.4dsolutions.net/ocn/sphpack2.html
# http://www.4dsolutions.net/synergetica/synergetica2.html
function accum_cubocta2(n)
    Int(round((10/3)*n^3 + 5*n^2 + (11/3)*n + 1, digits=0))
end
println("A005902:", [accum_cubocta2(x) for x in 0:10])
```

A005902:[1, 13, 55, 147, 309, 561, 923, 1415, 2057, 2869, 3871]



Tomb of Ayatollah Khomeini (سید روح‌الله موسوی خمینی) with an [octet truss](#) as a part of its architectural design.

Relevant Readings:

- *Gnomon* by Midhat Gazale
- *The Book of Numbers* by Conway and Guy
- *Synergetics* by Buckminster Fuller