



Recursión

Agosto 2017

Recursión

- La recursión es un método que permite resolver un problema basado en casos simples
- Posibilita una forma de control repetitiva distinta a las tradicionales (for, while)
- Normalmente en computación, el uso del término recursión hace referencia a los métodos o funciones recursivas:
 - Las cuales se basan en la invocación a sí mismo(a) para crear esa estructura repetitiva

Ejemplo

- **Factorial**
- Se define en principio como el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta n
- Por ejemplo:

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

$$n! = 1 \times 2 \times 3 \times 4 \times \dots \times (n - 1) \times n$$

$$n! = n \cdot (n - 1)!$$

n	$n!$
0	1
1	1
2	2
3	6
4	24
5	120
6	720
7	5040
8	40 320
9	362 880
10	3 628 800
15	1 307 674 368 000
20	2 432 902 008 176 640 000
25	15 511 210 043 330 985 984 000 000
50	$30\,414\,093\,201\,713\,378\,043 \times 10^{45}$
70	$1,19785717... \times 10^{100}$
450	$1,73336873... \times 10^{1000}$
3.249	$6,41233768... \times 10^{10\,000}$
25.206	$1,205703438... \times 10^{100\,000}$
100.000	$2,8242294079... \times 10^{456\,573}$

Funciones Recursivas

- Es necesario que una función tenga al menos un caso base, el cual puede ser un caso de finalización
- En el ejemplo anterior el caso base es $(0)! = 1$
- Cuando no se tiene un caso base el método o función se vuelve infinito

Diseño de una solución recursiva

- ▀ Identificar el caso base, considerando que puede existir más de uno
- ▀ Identificar las variables que se necesitan, considerando cuales tendrían que ser utilizadas como argumentos de un método
 - ▀ Estos argumentos permiten el control
- ▀ Analizar los otros casos además del base