



Hexadecimal₁₆

Rodrigo M. Cucick - 2018



Introdução ao Sistema Numérico 0x

- Notação hexadecimal (base 16)
 - Símbolos 0 .. 9 e A .. F, por exemplo; 10AF, DE9A, FFFF.
 - onde A = 10, B = 11, C = 12...



Aplicação

- Comumente utilizado em computação, para representar números binários de uma forma mais legível por nós, humanos.
 - $11011100\ 01101111_2 = DC6F_{16}$



Aplicação (cont.)

- Endereçamento de memória, ex;
 - Uma CPU com um *address bus* de 16-bit é capaz de acessar 65536 (64kB) endereços de memória distintos, pois $2^{16} = 65536$, ou seja, é possível realizar 65536 combinações diferentes usando 1 par de 8 bits (2 bytes),
 $11111111\ 11111111_2 = \text{FFFF}_{16} = 65535_{10} (0 \dots 65535)$



Aplicação (cont.)

- Um único dígito hexadecimal, por exemplo, E_{16} , é capaz de expressar um agregado de 4 bits, chamado de *nibble*, no entanto, na maioria das vezes, vocês encontrarão valores hexadecimais contendo pelo menos dois dígitos, para representarem 1 byte:
 - $E_{16} = 1110_2$ (nibble)
 - $9A_{16} = 1001\ 1010_2$ (byte)



Conversão de hexadecimal para decimal

Primeiramente, aprenderemos a conversão hexadecimal -> decimal.

$$8_{16^3} \ A_{16^2} \ 0_{16^1} \ E_{16^0}$$

$$(8 * (16 ^ 3)) + (10 * (16 ^ 2)) + (0 * (16 ^ 1)) + (14 * (16 ^ 0))$$

$$32768 + 2560 + 0 + 14$$

$$35342_{10}$$

Basta multiplicarmos cada dígito hexadecimal pela potência de 16 correspondente à sua posição no número e somarmos os resultados de todas as multiplicações.



Um outro exemplo de conversão hex -> dec

$$2_{16^3} \text{ C}_{16^2} \text{ F}_{16^1} 9_{16^0}$$

$$(2 * (16 ^ 3)) + (12 * (16 ^ 2)) + (15 * (16 ^ 1)) + (9 * (16 ^ 0))$$

$$8192 + 3072 + 240 + 9$$

$$11513_{10}$$



Módulo hex -> dec (Py)

- Ideia: Receber uma cadeia de caracteres (string) representando um número hexadecimal e transformá-la em decimal (int).
 - Ex; receber o *input* “0xFFFF” e retornar 65535.
 - Deverá haver validação e sanitização de *input*.
 - Código modular e conciso. Documentado.
 - *Python Enhancement Proposals* (PEP).
 - Usaremos Python 3.7.1 (implementação CPython).



Programando...

Estilo *live coding*, produto final será disponibilizado em um repositório específico.



Links

Todo e qualquer material utilizado e/ou gerado durante as aulas, estará disponível em: <https://github.com/rodrigoCucick/aulas>