

Aula 2

Introdução à linguagem

This work © 2024 by Lucas Seiki Oshiro is licensed under CC BY-NC-SA 4.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

O que veremos hoje

Recursos do OpenSCAD como linguagem de programação:

- variáveis
- tipos de dados
- operações
- funções

Linguagens de programação

- Usadas para descrever o **funcionamento** de programas
- Podem ser divididas em:
 - Propósito geral (**GPL**): Python, C, Java, JS, etc
 - Domínio específico (**DSL**): R, MATLAB, PureData, **OpenSCAD**

Características da linguagem OpenSCAD

- Bastante **especializada**
 - Poucos tipos
 - Sintaxe bastante direcionada para a modelagem
- Pouco **poderosa**
 - No sentido **computacional**, para processamento
 - Mas atende **muito bem** seu propósito
- **Declarativa**
 - Você diz **o que** quer e não *como* quer
- **Funcional**
 - Não tão pura quanto Haskell (nem mesmo quanto LISP)
 - Mas não é **imperativa**, como C, Java, Python, etc

Não sei programar, entendi nada...

Tudo bem!

Falaremos de **conceitos básicos!**

Sei programar, posso pular?

Não!

Falaremos das **particularidades** do OpenSCAD!

Variáveis e tipos de dados

Variáveis

- Igual na matemática (nas equações e funções)!
- Guardam **valores**
- Os valores podem assumir os seguintes **tipos**:
 - Números
 - Booleanos
 - Strings
 - Listas
 - Intervalos
 - Indefinidos
- **Definição**:
 - `variavel = valor;`
- Não podem ter **acentos**, **espaços** ou **caracteres especiais** (`_` é uma exceção)

Números

- Números **reais** = ponto flutuante (**float**) = números com vírgula
- Aceita operações **aritméticas**:
 - +, -, * (multiplicação), /
 - % (módulo = resto da divisão), ^ (potência)
- Aceitam expressões:
 - As **precedências** também funcionam aqui (ex: multiplicações feita antes das somas)
 - Se precisar mudar a precedência, **parênteses** também funcionam!

Booleanos

- Valores **lógicos** = **true** (verdadeiro) e **false** (falso)
- Aceitam operações **lógicas**:
 - and (e): &&
 - or (ou): ||
 - not (não): !
 - Precedência (menor para maior): ||, &&, !
- Operações de **comparação** geram booleanos:
 - <, >, <=, >=, ==, !=
- Booleanos podem decidir valores usando **operador ternário**:
 - valor = booleano ? verdadeiro : falso

Strings

- Textos
- Declaradas entre **aspas duplas**:
 - `minha_string = "oi tudo bem?"`
- Outros tipos podem ser **convertidos** para string
 - `string_1 = str(1);`
- Concatenação
 - `oi_tudo_bem = str("oi", " ", "tudo", " ", "bem");`
- Tamanho da string:
 - `tamanho = len("o rato roeu a roupa do rei de roma");`

Vetores

- **Lista** de valores
- Começa em 0
- Vetores de vetores = **matrizes**
- **Sintaxe:**
 - `vetor_de_numeros = [1, 2, 3, 4]`
 - `vetor_de_strings = ["um", "dois", "tres"]`
 - `matrix_2d = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]`
- **Indexação**
 - `vetor_de_numeros[0]`
 - `matriz_2d[1][2]`
- **Notação de ponto**
 - `vetor_de_numeros.x`
 - `vetor_de_numeros.y`
 - `vetor_de_numeros.z`

Vetores

- **Concatenação**
 - concatenado = concat(vetor1, vetor2, vetor3);
- **Tamanho**
 - tamanho = len(vetor);
- **Operações **vetoriais****
 - produto por escalar
 - $5 * [1, 2, 3]$
 - produto interno
 - $[1, 2, 3] * [4, 5, 6]$
- **Operações **matriciais****
 - matriz1 + matriz2
 - matriz1 * matriz2

Outros tipos

- Intervalos (**range**): usados em construções mais avançadas:
 - list comprehensions: [for (i = 0:2:10) i]
 - for (veremos mais pra frente)
- **undef**: equivalente ao **null** de outras linguagens

Funções

Funções

- Parecido com o conceito de **funções** na matemática
- Permite **reaproveitar** código
- **Facilitam** o desenvolvimento, quebrando o problema em problemas menores
- No OpenSCAD, as funções não têm **efeito colateral**

Funções trigonométricas

- **sin**: seno, **cos**: cosseno, **tan**: tangente
- **asin**: arco-seno, **acos**: arco-cosseno, **atan**: arco-tangente
- Todos os ângulos são expressos em **graus**, e não em radianos!
- Exemplos
 - $\sin(30)$:
 - $\text{asin}(0.5)$;
- Obs: **PI** é uma constante
 - $\text{asin}(0.5) * \text{PI} / 180$

Outras funções matemáticas

- **abs**: valor absoluto
- **ceil**: teto (arredondar para cima), **floor**: piso (arredondar para baixo)
- **round**: arredondar
- **ln**: logaritmo natural (base e), **log**: logaritmo na base 10
- **sqrt**: raiz quadrada
- **max**: máximo, **min**: mínimo

Definição de funções

Você pode **criar** as suas próprias funções!

Sintaxe:

```
function nome_da_funcao(parametro1, parametro2) = resultado;
```