



Engenharia de Software

Laboratório de Programação com interfaces com o usuário

Prof. M.Sc. André Saraiva

2022

Introdução

- **Carga Horária: 60 horas**
 - **Fevereiro: Revisão de conceitos Python (2 aulas)**
 - **Avaliação:**
 - **Prova (80%)**
 - **Atividade (20%)**

Introdução

- **Java x Python**
 - **Python é mais dinâmica;**
 - **Python possui o melhor custo benefício;**
 - **Java segue regras sintáticas rígidas;**
- Salário do Desenvolvedor Python pode chegar a R\$ 7.180,00
- Salário do Desenvolvedor Java pode chegar a R\$ 6.603,00

Bibliografia

**SUMMERFIELD, Mark. Programação em Python
3: uma introdução completa à linguagem
Python. Rio de Janeiro, 2012.**

Bibliografia complementar

BARRY, Paul. **Use a cabeça! Python**. Alta Books, 2019.

Bibliografia complementar



<https://docs.python.org/3/library/>

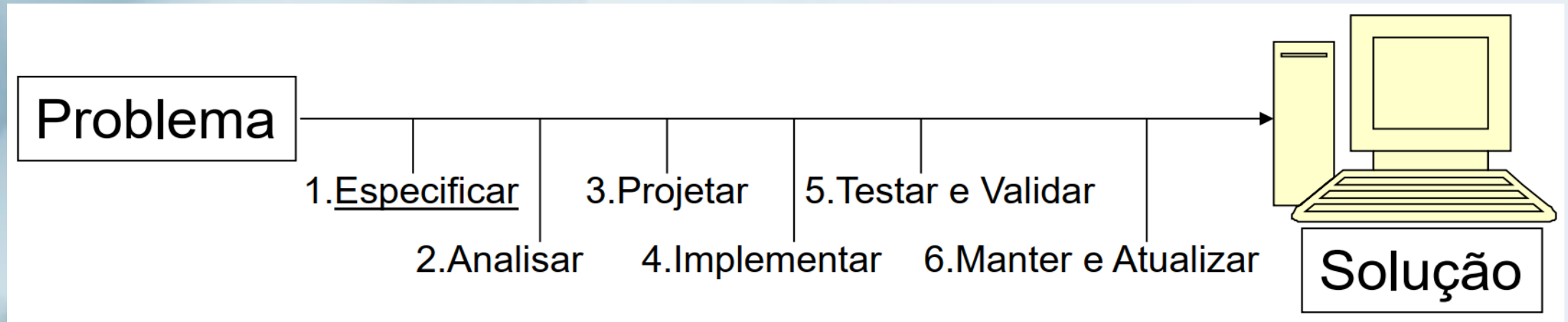
Revisão

- **Estrutura de Dados** é um modo particular de armazenamento e organização de dados de modo que possam ser usados eficientemente.
- **Algoritmo** é a descrição de um padrão de comportamento, especificado em termos de um conjunto BEM definido e finito de ações que PODEM ser executadas.

PROGRAMA = ALGORITMO + ESTRUTURA DE DADOS

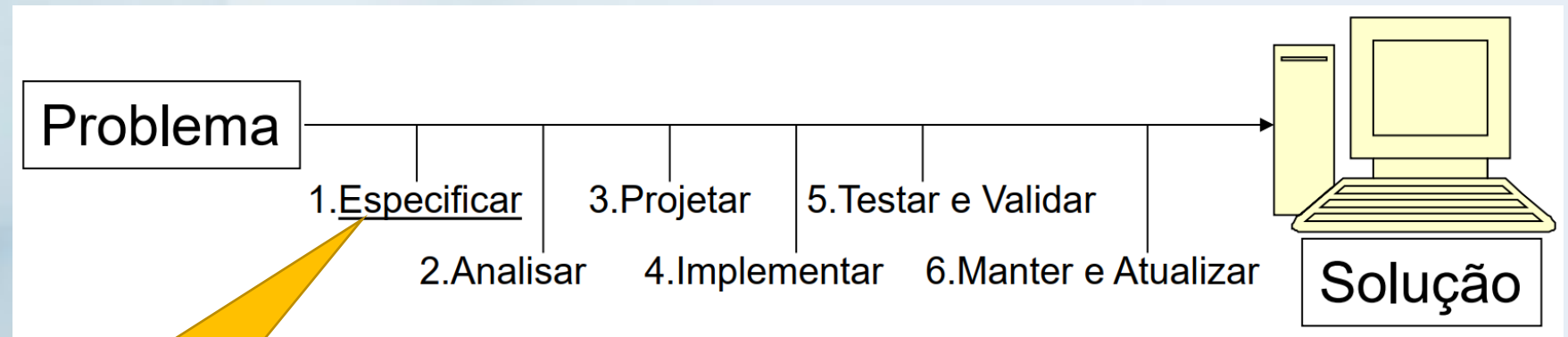
Revisão

- **Ciclo de Vida do Software**



Revisão

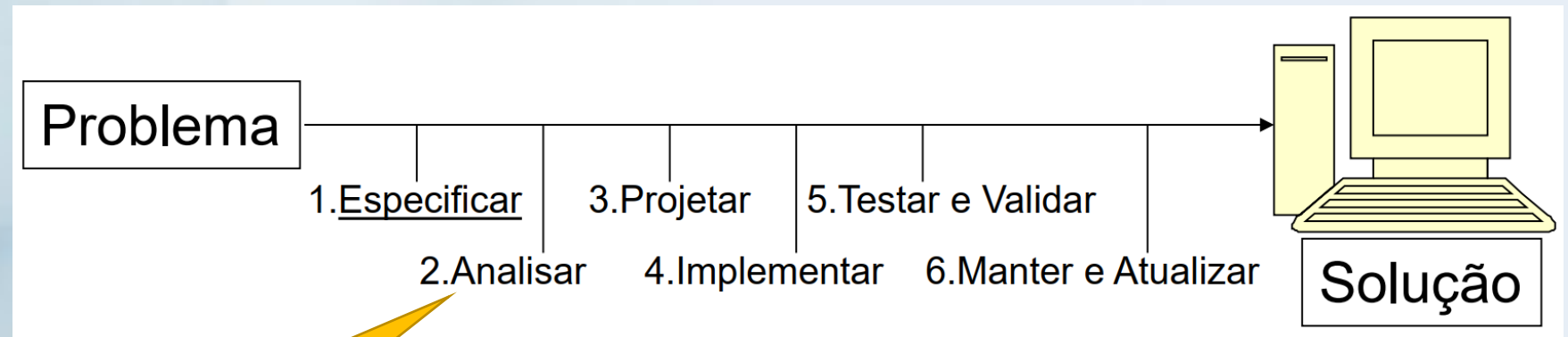
- **Ciclo de Vida do Software**



Especificar os requisitos
do problema

Revisão

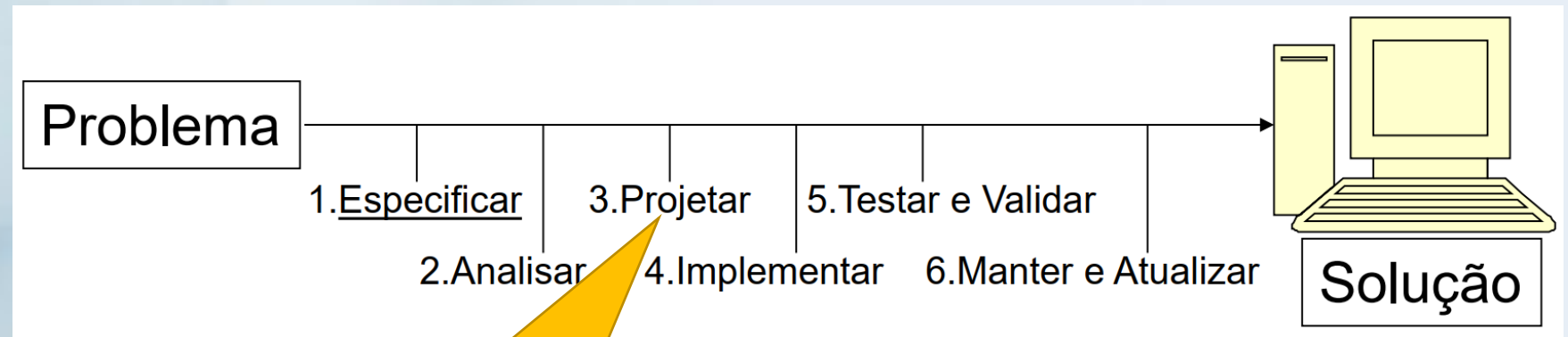
- **Ciclo de Vida do Software**



Analisar o problema

Revisão

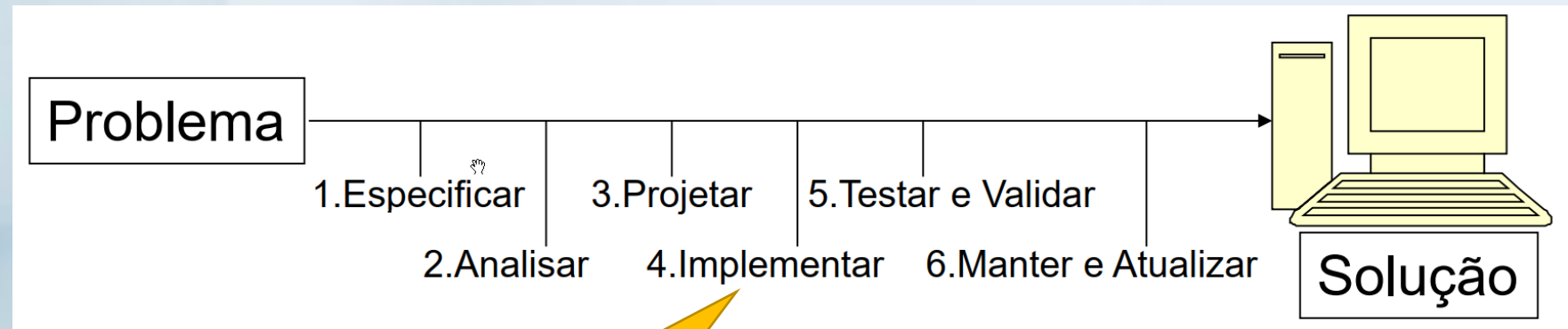
- **Ciclo de Vida do Software**



Projetar o Programa para
solucionar o problema

Revisão

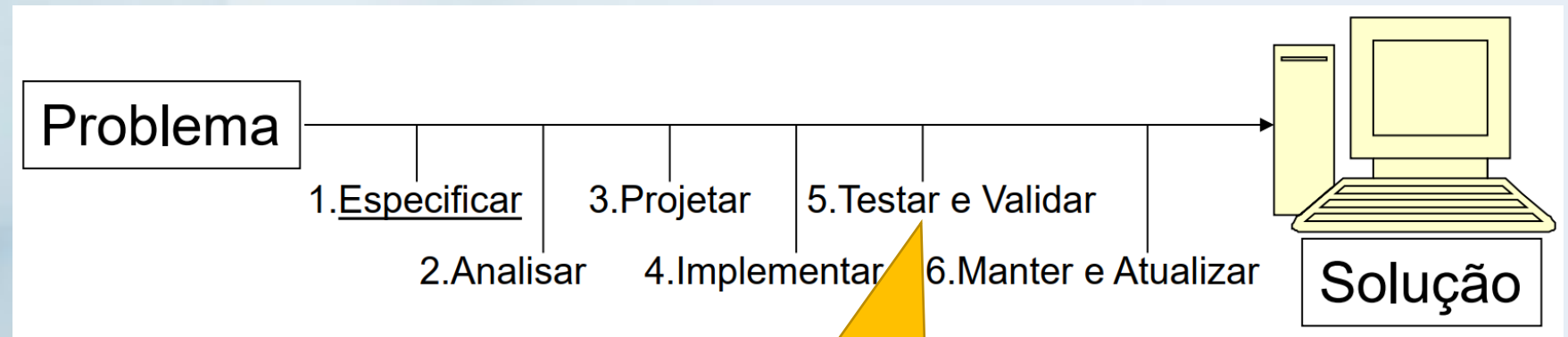
- **Ciclo de Vida do Software**



Implementar o projeto

Revisão

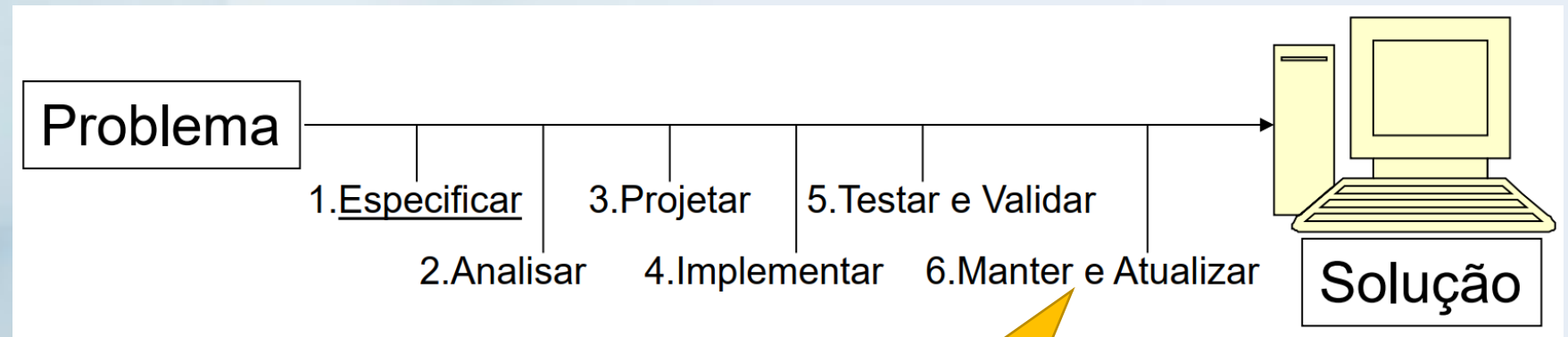
- Ciclo de Vida do Software



Testar a implementação
e validá-la

Revisão

- Ciclo de Vida do Software



Pós resolução do
problema

Revisão

- **PYTHON**

- Multiparadigma (estruturado, orientado a objetos, funcional e etc.)
- Multiplataforma (Windows, Linux, iOS, etc.)
- Interpretada
- Tipagem dinâmica
- Gerência de memória automatizada (coletor de lixo)

Revisão

- **O ambiente de desenvolvimento Python inclui:**
 - Um amplo conjunto de Interfaces de Programação de Aplicações (APIs, do inglês “*Application Programming Interfaces*”)
 - Ferramentas de compilação e depuração de código
- **Python não é uma IDE:**
 - Um amplo conjunto de Interfaces de Programação de Aplicações (APIs, do inglês “*Application Programming Interfaces*”)
 - Ferramentas de compilação e depuração de código

Revisão

- **Variáveis:**

- Área de memória que mantém um valor, que pode ser mudado
- O identificador da variável é o nome da variável, representada por uma sequência de caracteres, iniciada por uma letra minúscula. Exemplos: nota, saldo, deposito, saque, casa3Marica,...
- **Nossa Padronização:** não deve ter acento e deve possuir apenas caracteres alfa-numéricos. Usar notação de camelo (minhaNota, meuSaldo, saqueDiario,...)

Revisão - Tipos

- **Inteiros (int):**
 - Pode ter centenas de dígitos, limitado apenas pela memória do computador;
 - O padrão é decimal, mas pode-se usar outras bases como binária (iniciada com **0b**), octal (iniciada com **0o**) ou hexadecimal (iniciada com **0x**).

Revisão - Tipos

- **Lógicos (bool):**
 - Só possui dois dígitos, 0 e 1;
 - 0 é **False** (falso) e 1 é **True** (verdadeiro)

Revisão - Tipos

- **Ponto-Flutuante (float):**
 - Também pode ter centenas de dígitos, limitado apenas pela memória do computador
 - Representado por um par de números de ponto-flutuante.
Exemplo: 3.14

Revisão - Tipos

- **String (str):**
 - Representada por uma sequência de caracteres Unicode
 - Sempre iniciada e terminada por aspas simples ou duplas.
Exemplo: “Hello World” ou ‘Olá Mundo’

Revisão – Conversão de Tipos

- String, booleano ou ponto-flutuante para inteiro: **int**
int(false) = 0
- String, booleano ou inteiro para ponto-flutuante: **float**
float(1) = 1.00
- String, inteiro ou ponto-flutuante para booleano: **bool**
bool(1) = True
- Booleano, inteiro ou ponto-flutuante para String: **str**
str(123) = "123"

Revisão – Conversão de Tipos

- String, booleano ou ponto-flutuante para inteiro: **int**

int(False) = 0

**int(True) + int("456") =
457**

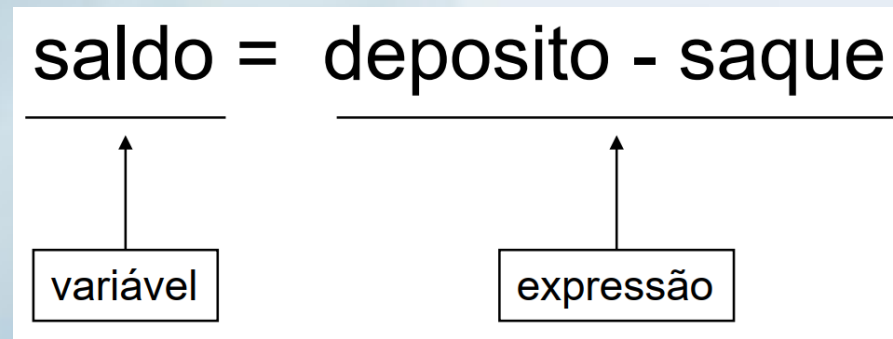
- Booleano, inteiro ou ponto-flutuante para String: **str**

str(123)="123"

**str("123") + str("456") =
123456**

Revisão – Comando de Atribuição

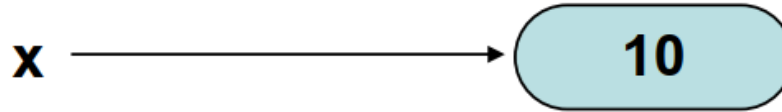
- Comando mais importante de uma linguagem imperativa.
- Em um comando de atribuição, uma variável recebe algo ou o resultado da avaliação de uma expressão.



Revisão – Comando de Atribuição

- `x = 10`

x



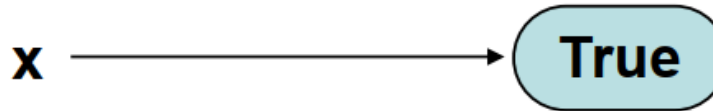
- `x = "Maria"`

x



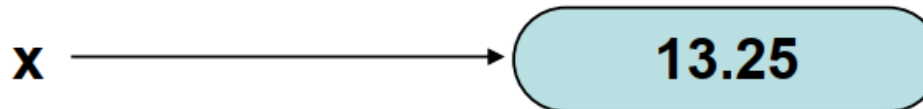
- `x = True`

x



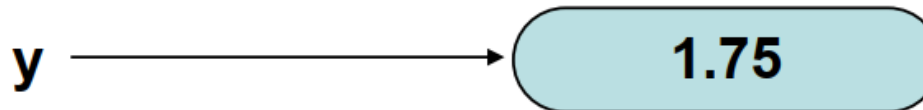
- `x = 13.25`

x



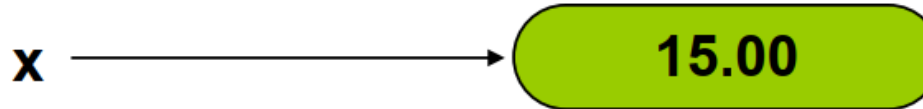
- `y = 1.75`

y



- `x = x+y`

x



Revisão – Operadores Artimétricos

- Soma (+)
 - Subtração (-)
- } operadores aditivos
- Produto (*)
 - Divisão de ponto flutuante (/)
 - Divisão inteira (//)
 - Resto da divisão inteira (%)
- } operadores multiplicativos
- Potenciação (**)

$$(5.75 + (a \% b) - 7) / 8.1$$

Revisão – Operadores Lógicos

- Disjunção lógica ou soma lógica (**or**)
- Conjunção lógica ou produto lógico (**and**)

$(p \text{ or } q) \text{ and } r$

Revisão – Operadores Lógicos

- Disjunção lógica ou soma lógica (**or**)
- Conjunção lógica ou produto lógico (**and**)

$(p \text{ or } q) \text{ and } r$

Revisão – Operadores Lógicos

- Tabela-verdade das conjunções “e” (\wedge), “ou” (\vee) e “não” (\neg)

A	B	$A \wedge B$	$A \vee B$	$\neg A$
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

Revisão – Operadores Binários

- Igual a (==)
- Diferente de (!=)
- Maior que (>)
- Menor que (<)
- Maior ou igual a (>=)
- Menor ou igual a (<=)

Revisão – Operadores Binários

- Igual a (==)
- Diferente de (!=)
- Maior que (>)
- Menor que (<)
- Maior ou igual a (>=)
- Menor ou igual a (<=)

$(2 + 3) == 5$ True

$(2 + 3) != 6$ True

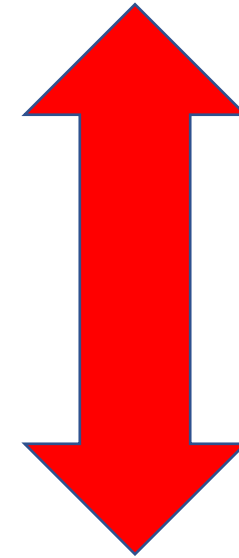
$(2 + 3) <= 5$ True

$(2 + 3) >= 5$ True

Revisão – Precedência

1. Expressões entre Parênteses ()
2. Potenciação (**);
3. Unários (+, -);
4. Binários Multiplicativos (*, /, %, //);
5. Binários Aditivos (+, -);
6. Relacionais (==, !=, <, >, <=, >=);
7. Lógico **not**;
8. Lógico **and**;
9. Lógico **or**

Maior prioridade



Menor prioridade

Revisão – Print

- **print(*expressão*, end = *término*)**
 - Escreve na saída padrão (vídeo) o resultado da avaliação da *expressão*;
 - Ao final, escreve a String de *término*.
- **print(*exp1*, *exp2*, ..., *expN*, end = *término*)**
 - Escreve na saída padrão (vídeo) o resultado da avaliação de cada expressão *expNum*;
 - Um espaço em branco é escrito entre cada par de *expNum*;
 - Ao final, escreve a String de *término*.

Revisão – Print

- Sintaxe da expressão formatada:

texto = "...%formato1 ... %formatoN..." %(exp1, ..., expN)

- Formatos mais comuns:

d (inteiro), **f** (número com ponto flutuante) e **s** (String)

- `variavel = "A média dos números é %4.2f" % (82432.923421)`
`print(variavel)`
 - Escreve "A média dos números é 82432.92" e pula de linha
- `print("%f + %f = %4.1f" % (15,7.8313, 15+7.8313))`
 - Escreve "15.000000 + 7.831300 = 22.8" e pula de linha
- `print("%d + %d = %d" % (5.89, 7.83, 5.89+7.83), end = "!!!")`
 - Escreve "5 + 7 = 13!!!" (%d = parte inteira) e end = "!!!" não pula de linha

Revisão – Input

- Comando que aguarda o usuário fornecer, pela entrada padrão (teclado), um valor expresso por uma sequência de caracteres, e o retorna
- Este comando tem o efeito de suspender a execução do programa até que o usuário escreva sua entrada e pressione a tecla <enter>.
- Normalmente é seguido de uma mensagem na saída padrão:
`primeiroNumero = input("Digite o primeiro Número")`

Atividade

- Criar uma calculadora no terminal para as operações básicas (soma, subtração, multiplicação e divisão). Seu software deve:
 - Permitir a entrada de um número;
 - Permitir a entrada de uma das operações citadas;
 - Soma(+), subtração(-), multiplicação(*) e divisão(/ ou //).
 - Permitir a entrada do segundo número;
 - Apresentar na saída padrão (vídeo) o resultado.
- Neste momento não iremos fazer nenhum tratamento ou condicionais

Próxima Aula

- **Nas próximas aulas abordaremos:**
 - **Estruturas de Controle:**
 - **If / Else;**
 - **For;**
 - **While.**

Fim

Obrigado!!!

Contato:

andresaraiva@id.uff.br