

MAC 113 — Introdução à Ciência da Computação

Aula 1

Nelson Lago

1º/2025



- **Informações sobre a disciplina no eDisciplinas**

- ▶ Datas de provas e critérios de avaliação
- ▶ Slides das aulas (a cada aula)
- ▶ Exercícios para praticar
- ▶ Entrega dos exercícios que valem nota
- ▶ Links para muitos materiais adicionais interessantes
- ▶ Fórum de discussões e informações sobre a monitoria
- ▶ O eDisciplinas pode enviar avisos por email; não deixe de checar sua caixa postal regularmente

Fazer muitos exercícios é fundamental



MAC113 – Intro à Computação para C. Humanas
(adm noturno 2025)

edisciplinas.usp.br/course/view.php?id=127660

Plágio/cola/etc. não dá, né?

**Leia o item sobre “Plágio++”
no eDisciplinas**

Computação:

Resolução de problemas do mundo real com os recursos que o computador oferece

Objetivos

- **Desenvolver o raciocínio aplicado na formulação e resolução de problemas computacionais**
 - ▶ Aprender a escrever programas de computador (na linguagem **R**)
 - ▶ Com exemplos usando ciência de dados
- **Mas não é um “curso de R”!**
 - ▶ O ensino de uma linguagem de programação é uma ferramenta didática e um bônus concreto de aprendizagem
- **Não é um curso de ciência de dados!**
 - ▶ A ciência de dados também é uma ferramenta didática e um bônus concreto da aprendizagem

Resolução de problemas do mundo real com os recursos que o computador oferece

Programando

Programar envolve

1 Compreender um problema em termos computacionais

- » Cálculos?
- » Processamento de multimídia?
- » Armazenamento e recuperação de dados?
- » ...

2 Definir como esse problema pode ser solucionado (*algoritmo*)

- ▶ Sequência **finita** de passos **bem definidos** baseados em um conjunto limitado (**vocabulário**) de operações possíveis
 - » O algoritmo é abstrato (como a planta de um prédio ou uma receita de bolo)
 - » E não tem vontade própria! (“o sistema não permite...”)

3 Implementar o algoritmo em uma linguagem de programação

- ▶ Gerando um *programa* que pode ser *executado* para solucionar o problema
 - » Uma receita de bolo em alemão é útil? Para quem?

4 Testar o programa

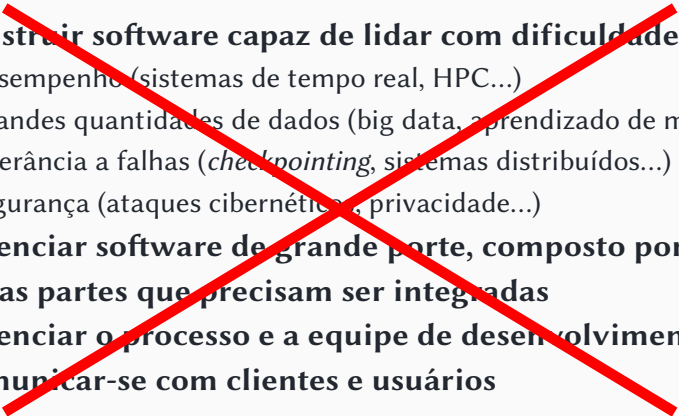
Um exemplo de algoritmo

Se eu sou capaz de multiplicar e somar dois dígitos,
posso multiplicar quaisquer inteiros:

$$\begin{array}{r} 123 \\ \times 45 \\ \hline 615 \\ 492+ \\ \hline 5535 \end{array}$$

Programando

Programar também envolve

- 
- ❶ **Construir software capaz de lidar com dificuldades especiais**
 - ▶ Desempenho (sistemas de tempo real, HPC...)
 - ▶ Grandes quantidades de dados (big data, aprendizado de máquina...)
 - ▶ Tolerância a falhas (*checkpointing*, sistemas distribuídos...)
 - ▶ Segurança (ataques cibernéticos, privacidade...)
 - ❷ **Gerenciar software de grande porte, composto por várias partes que precisam ser integradas**
 - ❸ **Gerenciar o processo e a equipe de desenvolvimento**
 - ❹ **Comunicar-se com clientes e usuários**

Para ser útil, um programa geralmente

❶ **Obtém dados**

❷ **“Faz alguma coisa” com esses dados**

- ▶ Gerando um resultado

❸ **“Faz alguma coisa” com esse resultado**

- ▶ Mostra para o usuário

- ▶ **Utiliza como dado para fazer outra coisa**

Lembre-se:

O computador só faz o que você manda!

(não o que você quer)

Se você esquecer um passo, ele obedece!

Linguagens de programação

- **Um programa de computador é uma descrição detalhada dos algoritmos usados para resolver um problema**
 - ▶ Essa descrição é feita usando uma **linguagem de programação**
 - » *Criada para facilitar a “comunicação” entre o programador e o computador*
 - » *Ou melhor, para facilitar a expressão de soluções algorítmicas usando o computador*
- **Existem centenas de linguagens de programação**
 - ▶ Vantagens e desvantagens ↔ tipo de problema
 - ▶ Conhecimento prévio
 - ▶ Gosto pessoal
 - ▶ ...
- **R (r-project.org) é uma linguagem amplamente usada em estatística e economia**
 - ▶ Coincidentemente, é a linguagem que vamos usar neste curso 😊

- Em geral, um programa é um **arquivo de texto**
 - ▶ Arquivos de texto podem ser escritos em português, chinês, alemão...
 - ▶ Ou em uma linguagem de programação
- É possível usar qualquer editor de texto
- **MAS!**
- É muito mais confortável usar editores especializados
 - ▶ Visual Studio Code + plugin REditorSupport
 - ▶ StatET for R (baseado no Eclipse)
 - ▶ ...
 - ▶ O mais popular é o **R Studio**
- Durante as aulas, usaremos o **google colab**
 - ▶ colab.research.google.com
- Mas você pode usar qualquer outro

Configurando o ambiente

The screenshot shows the Google Colab web interface in a browser. The address bar displays `https://colab.research.google.com`. The page title is "Olá, este é o Colaboratory". The left sidebar contains a navigation menu with options like "Índice", "Vamos começar", "Ciência de dados", "Machine learning", and "Mais recursos". The main content area is titled "Conheça o Colab" and includes a section "(Novidade) Teste a API Gemini" with a list of links: "Generate a Gemini API key", "Talk to Gemini with the Speech-to-Text API", "Gemini API: Quickstart with Python", "Gemini API code sample", "Compare Gemini with ChatGPT", and "More notebooks". Below this, there is a paragraph about trying Colab and a video thumbnail titled "3 Cool Google Colab Features". At the bottom, there is a code input field with the text "[] Conecte a programar ou gere código com IA." and a section titled "O que é o Colab?" with a brief description.

← → ↺ `https://colab.research.google.com` gmail ime

Introdução Most Visited Smart Bookmarks Getting Started Slashdot Getting Started Most Visited Outros favoritos

Olá, este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Compartilhar

Índice

Vamos começar

Ciência de dados

Machine learning

Mais recursos

Exemplos em destaque

+ Seção

Conheça o Colab

(Novidade) Teste a API Gemini

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

Se você já conhece bem o Colab, confira este vídeo para saber mais sobre as tabelas interativas, a visualização do histórico de código executado e o Palette de comandos.

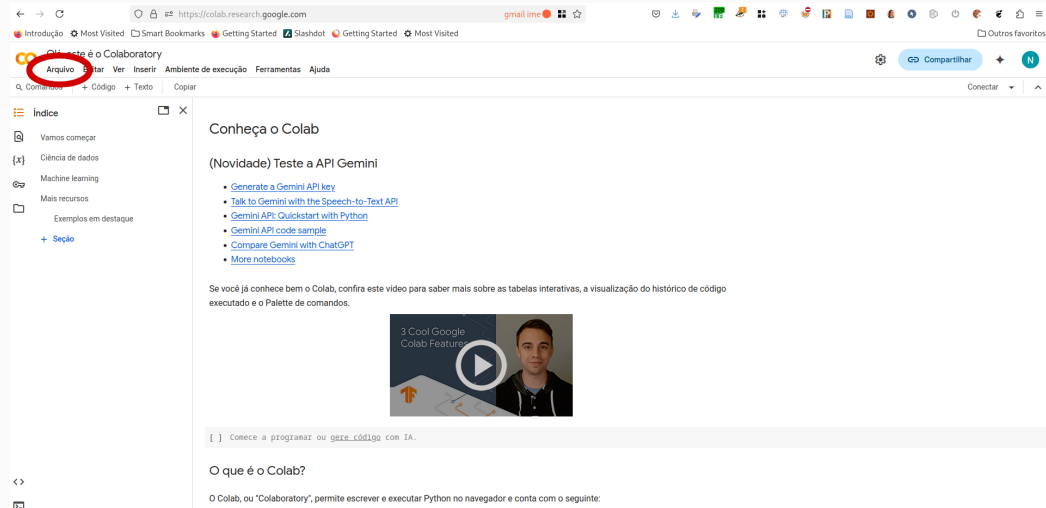
3 Cool Google Colab Features

[] Conecte a programar ou gere código com IA.

O que é o Colab?

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador e conta com o seguinte:

Configurando o ambiente



← → ↻ https://colab.research.google.com gmail ime

Introdução Most Visited Smart Bookmarks Getting Started Slashdot Getting Started Most Visited

Colab este é o Colaboratory

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Compartilhar + Código + Texto Copiar

Conectar

Conheça o Colab

(Novidade) Teste a API Gemini

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Gemini API: Quickstart with Python](#)
- [Gemini API code sample](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

Se você já conhece bem o Colab, confira este vídeo para saber mais sobre as tabelas interativas, a visualização do histórico de código executado e o Palette de comandos.

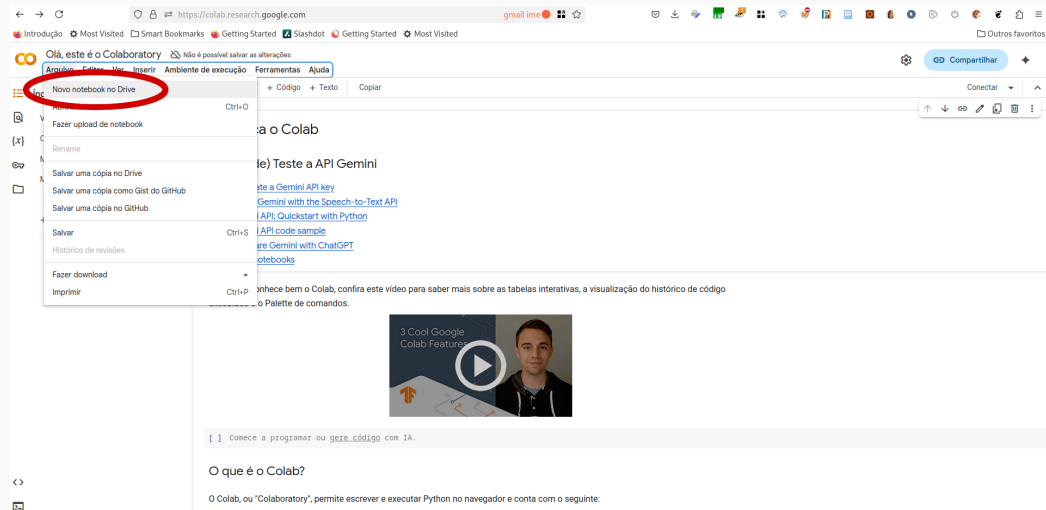


[] Comece a programar ou [gere código](#) com IA.

O que é o Colab?

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador e conta com o seguinte:

Configurando o ambiente



The screenshot shows the Google Colab web interface. The browser address bar displays `https://colab.research.google.com`. The top navigation bar includes links for 'Introdução', 'Most Visited', 'Smart Bookmarks', 'Getting Started', 'Slashdot', and 'Most Visited'. The main header area shows 'Olá, este é o Colaboratory' and a message 'Não é possível salvar as alterações'. The 'File' menu is open, with 'Novo notebook no Drive' highlighted by a red circle. Other menu items include 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. The main content area displays a video player with the title '3 Cool Google Colab Features' and a play button. Below the video, there is a text area with the heading 'O que é o Colab?' and a paragraph explaining that Colab allows writing and executing Python code in a browser.

https://colab.research.google.com

Introdução Most Visited Smart Bookmarks Getting Started Slashdot Getting Started Most Visited

Olá, este é o Colaboratory Não é possível salvar as alterações

Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda

Novo notebook no Drive

Ctrl+O

Fazer upload de notebook

Rename

Salvar uma cópia no Drive

Salvar uma cópia como Gist do GitHub

Salvar uma cópia no GitHub

Salvar Ctrl+S

Histórico de revisões

Fazer download

Imprimir Ctrl+P

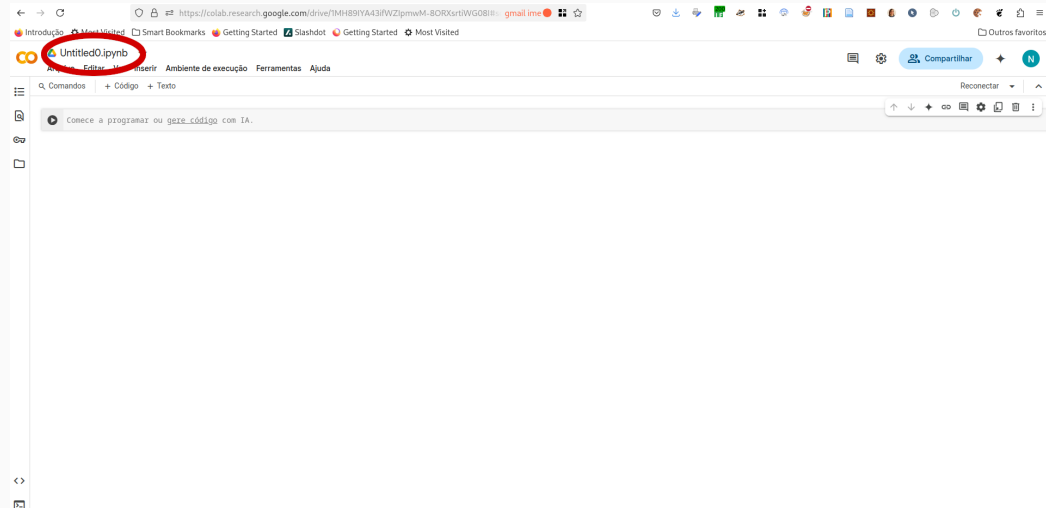
Conectar

3 Cool Google Colab Features

O que é o Colab?

O Colab, ou "Colaboratory", permite escrever e executar Python no navegador e conta com o seguinte:

Configurando o ambiente



Configurando o ambiente

The image shows a web browser window displaying the Google Colab interface. The address bar shows the URL: <https://colab.research.google.com/drive/1MH891YA43iFWZlpmwM-8ORXsrtiWG08l>. The page title is "Untitled0.ipynb". The menu bar includes "Arquivo", "Editar", "Ver", "Inserir", "Ambiente de execução", "Ferramentas", and "Ajuda". The "Ambiente de execução" menu is open, showing a list of options with keyboard shortcuts. The option "Alterar o tipo de ambiente de execução" is highlighted with a red circle. Other options include "Executar tudo" (Ctrl+F9), "Executar antes" (Ctrl+F8), "Executar a célula em foco" (Ctrl+Enter), "Executar seleção" (Ctrl+Shift+Enter), "Executar célula e abaixo" (Ctrl+F10), "Interromper execução" (Ctrl+M I), "Reiniciar sessão" (Ctrl+M .), "Reiniciar sessão e executar tudo", "Desconectar e atualizar ambiente de execução", "Gerenciar...", "Ver recursos", and "Ver registros do ambiente de execução". The background shows a code editor with a prompt "Comece a programar ou" and a button "Analisar arquivos com o Gemini".

Google Colab interface showing the "Ambiente de execução" (Execution Environment) menu. The option "Alterar o tipo de ambiente de execução" (Change execution environment type) is highlighted with a red circle.

Menu items and shortcuts visible:

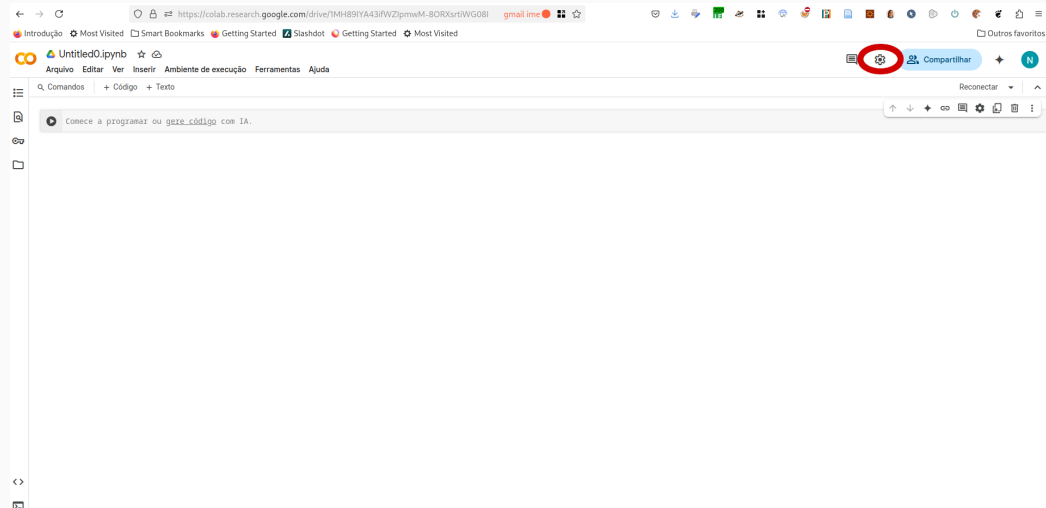
- Executar tudo (Ctrl+F9)
- Executar antes (Ctrl+F8)
- Executar a célula em foco (Ctrl+Enter)
- Executar seleção (Ctrl+Shift+Enter)
- Executar célula e abaixo (Ctrl+F10)
- Interromper execução (Ctrl+M I)
- Reiniciar sessão (Ctrl+M .)
- Reiniciar sessão e executar tudo
- Desconectar e atualizar ambiente de execução
- Alterar o tipo de ambiente de execução**
- Gerenciar...
- Ver recursos
- Ver registros do ambiente de execução

Configurando o ambiente

The screenshot shows the Google Colab web interface. At the top, the browser address bar displays the URL: <https://colab.research.google.com/drive/1MH891YA43iFWZlpmwM-8ORXsrtiWG08l>. The Colab toolbar includes options like 'Introdução', 'Most Visited', 'Smart Bookmarks', 'Getting Started', and 'Slashdot'. The main editor area shows a file named 'Untitled0.ipynb' with a 'Salvando...' status. The menu bar includes 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. Below the menu, there are tabs for 'Comandos', 'Código', and 'Texto'. The main code area contains the text: 'Comece a programar ou gere código com IA.'.

A dialog box titled 'Alterar o tipo de ambiente de execução' (Change the type of execution environment) is open in the center. It shows the current environment as 'R'. A dropdown menu is open, showing options: 'Python 3' (selected), 'R' (highlighted with a red circle), and 'Julia'. To the right of the dropdown, there are radio buttons for hardware acceleration: 'GPUs: T4' (selected), 'GPUs: A100', 'GPUs: L4', and 'TPU v5e-1'. At the bottom of the dialog, there is a message: 'Quer acesso a GPUs premium? [Compre mais unidades de computação](#). Sua organização pode comprar licenças do Colab Pro e do Colab Pro+ em massa. [Saiba mais](#)'. The dialog has 'Cancelar' and 'Salvar' buttons at the bottom right.

Configurando o ambiente



Configurando o ambiente

The image shows the Google Colab web interface. The browser address bar displays the URL: <https://colab.research.google.com/drive/1MH891YA43iFWZlpmwM-8ORXsrtiVWG08l>. The Colab environment is titled 'Untitled0.ipynb'. The 'Configurações' (Settings) dialog box is open, showing the 'Editor' tab, which is highlighted with a red circle. The settings for the Editor include:

- Site: Vinculações de teclas do editor default
- Assistência de IA: Tamanho da fonte (px) 14
- Colab Pro: Família de fontes usada para renderizar código monospace
- GitHub: Largura do recuo em espaços 2
- Diversos: Coluna com régua vertical 80

Under the 'Diversos' (Miscellaneous) section, the checkbox for 'Mostrar sugestões de preenchimento de código com base no contexto' (Show code completion suggestions based on context) is checked and circled in red. Other options include 'Mostrar números de linha' and 'Mostrar guias de recuo'. The dialog box has 'Cancelar' (Cancel) and 'Salvar' (Save) buttons at the bottom right.

Configurando o ambiente

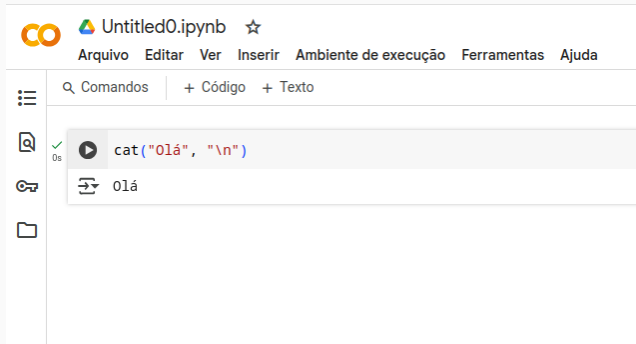
The screenshot shows the Google Colab web interface. A 'Configurações' (Settings) dialog box is open, displaying various configuration options. The 'Assistência de IA' (AI Assistance) section is highlighted with a red circle. Within this section, the checkbox 'Mostrar sugestões inline geradas com IA' (Show inline suggestions generated with AI) is also highlighted with a red circle. Other settings visible include 'Editor', 'Colab Pro', 'GitHub', and 'Diversos'. A warning message at the bottom of the dialog states: 'A IA generativa no Colab é uma tecnologia experimental e pode, às vezes, mostrar informações imprecisas ou ofensivas que não representam as opiniões do Google. Portanto, [use o código com cuidado](#).' (Generative AI in Colab is an experimental technology and may, at times, show inaccurate or offensive information that does not represent Google's opinions. Therefore, [use the code with care](#).)

Configurações

- Site > ☐ Mostrar sugestões inline geradas com IA
- Editor > ☐ Consentimento para usar recursos de IA generativa
- Assistência de IA** > ☐ Ocultar recursos de IA generativa
- Colab Pro >
- GitHub >
- Diversos >

A IA generativa no Colab é uma tecnologia experimental e pode, às vezes, mostrar informações imprecisas ou ofensivas que não representam as opiniões do Google. Portanto, [use o código com cuidado](#).

Fechar



cat()

```
cat("Olá!", "\n")
```

Olá

Blocos de código no colab

The image shows a web browser window displaying the Google Colab interface. The address bar shows the URL: <https://colab.research.google.com/drive/1MH891YA43iIWZlpmwM-8ORXsrtIWG08I#>. The browser's bookmark bar includes links like 'Introdução', 'Most Visited', 'Smart Bookmarks', 'Getting Started', 'Slashdot', and 'Outros favoritos'. The Colab interface itself has a top menu bar with 'Arquivo', 'Editar', 'Ver', 'Inserir', 'Ambiente de execução', 'Ferramentas', and 'Ajuda'. A 'Compartilhar' button is visible on the right. Below the menu, there's a search bar for 'Comandos' and tabs for '+ Código' and '+ Texto'. The main area contains a code block with the text: '[] Comece a programar ou gere código com IA.' The block has a play button icon on the left and a toolbar on the right with icons for undo, redo, copy, paste, settings, and other actions.

Expressões em R

- **A maioria das coisas em R são *expressões***
 - ▶ (expressões são coisas que têm um *valor*)
 - ▶ Exemplo: 47
 - ▶ Exemplo: 2 + 3
 - ▶ Exemplo: "Oi galera!"
- **Expressões podem ser combinadas ou utilizadas como partes de outras expressões**
 - ▶ Exemplo: 2 + 3 + 7
 - ▶ Exemplo: $\frac{2+3+7}{6}$



Expressões aritméticas

$$2 + 3 + 7$$

```
cat(2 + 3 + 7, "\n")
```

12

Expressões aritméticas

$$\frac{2+3+7}{6}$$

```
cat((2 + 3 + 7) / 6, "\n")
```

2

Precedência de operadores

operador	descrição	associatividade
()	parênteses	da esquerda para a direita
^	exponenciação	da direita para a esquerda
+ -	positivo e negativo unário	da esquerda para a direita
%% %/%	resto e divisão inteira	da esquerda para a direita
* /	multiplicação e divisão	da esquerda para a direita
+ -	soma e subtração	da esquerda para a direita

Brincando com dígitos

- O operador `%/%` faz uma divisão inteira
 - ▶ $13 \% \% 5 = 2$
- Se o divisor é 10, o resultado é que eliminamos o último dígito:
 - ▶ $12345 \% \% 10 = 1234$
- O operador `%%` fornece o resto da divisão
 - ▶ $13 \% \% 5 = 3$
 - » *(Por definição, o resultado sempre é menor que o divisor)*
- Se o divisor é 10, o resultado é igual ao último dígito:
 - ▶ $12345 \% \% 10 = 5$
 - » *Qualquer número ...xyz pode ser escrito como $\dots xy * 10 + z$*
 - » *$\dots xy * 10$ é múltiplo de 10, então o resto da divisão de ...xyz por 10 é z*
 - » *(e, como visto acima, z sempre é menor que 10)*

(Não é exatamente “divisão inteira” nem “resto da divisão”, mas ok)

Brincando com dígitos

Encontre o último dígito de um número:

```
cat(2407 %% 10, "\n")
```

```
cat(110 %% 10, "\n")
```

```
cat(3 %% 10, "\n")
```

```
cat(0 %% 10, "\n")
```

7

0

3

0

Brincando com dígitos

Dado um número n , transforme-o no formato $a + b$, onde $a + b = n$ e a é o maior múltiplo de 10 tal que $a \leq n$

```
cat("b:", n %% 10, "\n")  
cat("a:", n - b, "\n")
```

Tipos

```
cat(2 + 3, "\n")  
cat("2 + 3", "\n")  
cat("2" + "3", "\n")  
cat(1/3, "\n")  
options(digits=15)  
cat(2 + 3, "\n")  
cat(1/3, "\n")
```

```
5  
2 + 3  
[Erro]  
0.3333333  
5  
0.333333333333333
```



- **Obviamente, $2 + 3$ não é a mesma coisa que "2 + 3"**
 - ▶ Um é uma operação, o outro é um texto
 - » $2 + 3$ e $2.4 + 2.6$ também não são a mesma operação (somar números inteiros e somar números não-inteiros são operações diferentes, mesmo que chamemos ambas de "+")
 - » Imprimir um número como $\frac{1}{3}$ também não é a mesma operação que imprimir um número inteiro (é preciso "decidir" o que fazer com o arredondamento)
- **Se solicitarmos ao computador que calcule $\frac{2}{3}$, quanto de memória vai ser necessário para armazenar o resultado?**
- **E $\sqrt{2}$?**

Tipos

- Existem *tipos de dados* diferentes em R

- ▶ Números inteiros, como 2 ou -437

- » Não são comumente usados em R

- » Números entre ~ -2 bilhões e ~ 2 bilhões

- ▶ Números (potencialmente) não-inteiros (“números de ponto flutuante”), como 2.0 ou 6.166666666666667

- » O tipo mais comum em R

- » Às vezes o resultado de cálculos pode ser aproximado

- » `options(digits=18)`

- `cat(5^0.5 * 5^0.5, "\n")`

- 5.00000000000000000089 *Ok, números irracionais...*

- `cat(0.1 + 0.1 + 0.1, "\n")`

- 0.30000000000000000044 🤪

- ▶ Textos (“strings” ou “cadeias de caracteres”), como "Oi galera!"

- R “sabe” quais operações podem ser realizadas com cada tipo

Nomes (variáveis)

- Ao programar, preferimos pensar no problema a ser resolvido e não nas idiossincrasias do computador
- Linguagens de programação de alto nível procuram oferecer os recursos para isso
- Uma das coisas mais importantes para esse fim é utilizar *nomes*

Nomes (variáveis)

```
x ← 5 (atribuição)
```

Há um número finito de caracteres no teclado, então fazemos atribuição em R com “<-”:

```
x <- 5
```

Muitas linguagens utilizam = para fazer atribuição, e R aceita também:

```
x = x + 1
```

expressão (valor)

atribuição

nome

AAAAAHHHH!!!!!!

Nomes (variáveis)

```
cat(class(5), "\n")  
cat(class("Olar!"), "\n")  
cat(class(x), "\n")  
x <- 5  
cat(class(x), "\n")  
cat(class(5L), "\n")  
x <- "Olar!"  
cat(class(x), "\n")
```

numeric

character

Error in cat(class(x)) : objeto 'x' não encontrado

numeric

integer

character

Como o R faz a distinção
entre nomes e strings?

Aspas!

Nomes (variáveis)

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$-x^2 + 3x + 4 = 0$$

```
cat((-3 + sqrt(3^2 - 4 * -1 * 4)) / -2,  
      (-3 - sqrt(3^2 - 4 * -1 * 4)) / -2, "\n")
```

-1 4



Nomes (variáveis)

```
# Calcula raízes da equação de segundo grau
a <- -1
b <- 3
c <- 4
delta <- b^2 - 4 * a * c # Não pode ser menor que zero!
raiz1 <- (-1 * b + sqrt(delta)) / 2 * a
raiz2 <- (-1 * b - sqrt(delta)) / 2 * a
cat("As raízes são", raiz1, "e", raiz2, "\n")
```

As raízes são -1 e 4

Nomes (variáveis)

- **Por que o título destes slides é “Nomes (variáveis)”?**
 - ▶ Não há muita graça em escrever programas como os que vimos acima, em que os dados são sempre os mesmos
 - ▶ Um dos principais usos de nomes é representar valores que *variam* (basicamente, alguma informação “real” que está em algum lugar na memória do computador)
 - » *Como na matemática!*
 - ▶ Por isso, chamamos esses nomes de “variáveis”
 - ▶ Acima, definimos o valor dos nomes (variáveis) com comandos de atribuição (a <- -1) fixos. Como fazer se queremos valores que não sejam fixos?
 - ▶ `readline()`

Primeiro programa – bully

```
idade <- readline("Informe sua idade: ")
idade <- as.integer(idade)
cat("Você tem só", idade, "anos?!?!", "\n")
cat("Nossa, você aparenta ter", 2 * idade, "anos!", "\n")
```