

# Prática 1

---

Conteúdos que serão praticados

- **Definição** de métodos `static`
  - **Chamada** de métodos `static`
  - Entrada de dados com `Scanner`
  - Variáveis **locais**
  - Passagem de parâmetros na chamada de métodos
  - Retorno de valores após chamada de métodos
- 

## Atividade 1

1. Crie um projeto Java Padrão e dê o nome de Atividade1
2. Na classe principal, defina (crie) um método `static` que apresente a mensagem "Atividade 1" na tela
3. Chame esse método na `main` de tal forma que, ao rodar o programa, a mensagem "Atividade 1" apareça na tela

OBSERVAÇÕES:

- NÃO use nenhuma variável
  - O `System.out.print` deve ser usando no método definido no item 2 (para apresentar o resultado)
  - O `System.out.print` NÃO DEVE ser utilizado **na** `main`
- 

## Atividade 2

1. Crie um projeto Java Padrão e dê o nome de Atividade2
2. Na classe principal, defina um método `static` que receba um valor do tipo `String` como parâmetro e mostre esse valor `String` na tela (`System.out.println` deve estar nesse método).
3. Chame esse método na `main` de tal forma que, ao rodar o programa, a mensagem passada por parâmetro apareça na tela

OBS.:

- NÃO use nenhuma variável

- O `System.out.print` deve estar no método do item 2 (para mostrar os resultados)
  - NÃO use `System.out.print` na `main` (para mostrar os resultados)
- 

## Atividade 3

1. Crie um projeto Java Padrão e dê o nome de Atividade3
2. Na classe principal, defina um método `static` que receba um valor do tipo `String` como parâmetro e mostre esse valor `String` na tela (`System.out.println` deve estar nesse método).
3. Na `main`, peça para o usuário digitar uma mensagem qualquer e armazene essa mensagem em uma variável local do tipo `String` (`Scanner` deve estar na `main`)
4. Chame o método definido no item 2 passando como parâmetro a mensagem que o usuário inseriu

### OBSERVAÇÃO:

- O `Scanner` deve ser usado na `main`
  - O `System.out.print` deve ser usado no método definido no item 2
  - NÃO use o `System.out.print` na `main` (para mostrar os resultados)
- 

## Atividade 4

1. Crie um projeto Java Padrão e dê o nome de Atividade4
2. Na classe principal, defina um método `static` (sem parâmetros) que retorne um valor do tipo `String`; esse método deve pedir para o usuário digitar uma mensagem, armazenar a mensagem em uma variável local do tipo `String` e retornar seu valor (`Scanner` deve ser usado nesse método)
3. Na `main`, crie uma variável local e armazene nela o retorno da chamada do método do item 2
4. Apresente, na `main`, o conteúdo dessa variável local na tela para o usuário (`System.out.print` deve ser usado na `main`)

### OBSERVAÇÃO:

- O `System.out.print` deve ser usado na `main`
  - NÃO USE o `System.out.print` no método do item 2
  - O `Scanner` deve ser usado no método do item 2
  - NÃO USE o `Scanner` na `main`
-

## Atividade 5

1. Crie um projeto Java Padrão e dê o nome de Atividade5
2. Na Classe Principal, defina os seguintes métodos `static`
  - o `soma`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
  - o `sub`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
  - o `mult`: recebe dois parâmetros do tipo `int` e retorna um valor do tipo `int`
  - o `div`: recebe dois parâmetros do tipo `float` e retorna um valor do tipo `float`
3. Na `main`, peça para o usuário digitar valores do tipo `int` e armazene esses valores nas variáveis locais `x` e `y` (`Scanner` deve ser usado na `main`)
4. Na `main`, faça a chamada para cada um dos métodos definidos no item 2 passando os valores de `x` e `y` como parâmetro e apresente os valores retornados por cada um dos métodos na tela (`System.out.print` deve ser usado na `main`)

### OBSERVAÇÃO:

- O `Scanner` deve ser usado na `main`
- NÃO USE o `Scanner` em nenhum dos métodos do item 2
- O `System.out.print` deve estar na `main` (para apresentar os resultados)
- NÃO USE o `System.out.print` nos métodos `static` do item 2 (para apresentar os resultados)