

# transforme ■ se



# Na aula de hoje

- Finalizar a atividade da última aula
- IDEs
- Pycharm
- Estrutura de Dados
- Sequências
- Conjuntos
- Dicionários

# Anteriormente...

## Jogo da adivinhação

#escrever uma saudação

ESCREVA("Bem Vindo ao Jogo da Adivinhação")

#definir um numero mágico

numero\_magico = 42

#receber um chute do usuário

ESCREVA("Tente advinhar o número mágico, faça seu chute:")

LEIA(chute)

#comparar o chute com número secreto

SE(chute == numero\_magico)

    ESCREVER("Acertou")

SENAO SE(chute > numero\_magico)

    ESCREVER("Errou, chutou alto!")

SENAO

    ESCREVER("Errou, chutou baixo!")

# Comando while

Operador de laço

```
x = 5
```

```
while(x > 1):
```

```
    print(x)
```

```
    x = x - 1
```

# Exercício:

## Modificando o Programa

Vamos fazer juntos!

- Acrescentar um limite de tentativas que o usuário tem para acertar o número secreto. O algoritmo deverá informar a quantidade de rodadas que faltam, bem como em que rodada está no momento.

# Exercício:

Modificando o Programa

Vamos fazer juntos!

Vamos passar o portugol para Python!

# Função range()

Define uma série de valores

O `range(1, 10)` vai gerar os números de 1 a 9.

Para definir um passo, um intervalo entre os elementos, passamos um terceiro valor, `range(1, 10, 2)`.

Por padrão, o passo sempre é 1.

# Comando for

## Operador de laço

Usando a função range():

```
for x in range(1,10):  
    print(x)
```

Podemos passar os valores da sequência manualmente:

```
for x in [1,2,3,4,5,6]:  
    print(x)
```



# Exercício:

## Utilizando o for no jogo e novas funcionalidades

Para casa. Entrega código no github até às 19h do dia 26/06.

Valendo nota de participação. Grupos de 1 a 6 pessoas, escolha livre.

- Substitua o comando while pelo comando for.
- Escolha um número secreto entre 0 e 100.
- Crie um nível de dificuldade para o jogo. Peça para o usuário escolher em qual nível ele deseja jogar. O nível é mensurado de acordo com as tentativas.
- Acrescente um sistema de pontuação. O jogador deve começar com 1000 pontos e cada chute errado deve ser subtraído do total de pontos um valor que corresponde a diferença entre o chute e o número secreto. (dica, pesquise sobre a função abs(), ela pode ser útil)

# Integrated Development Environments

Qual ferramenta vou utilizar para escrever código?

Uma IDE é um software que oferece um ambiente integrado para desenvolvimento de software. Ela combina recursos como editor de texto, depuração, gerenciamento de projetos e execução de código em um único lugar.

- Visual Studio Code: meu preferido, mas é mais considerado um editor de texto. Possui extensões que o transformam em uma excelente IDE para Python. Ele fornece recursos de depuração, formatação de código, gerenciamento de pacotes e integração com Git.
- PyCharm: Desenvolvido pela JetBrains, o PyCharm é uma IDE de Python poderosa e amplamente utilizada. Vamos conhecer mais a respeito hoje!
- Thonny é outra IDE de Python popular, especialmente recomendada para iniciantes e estudantes. Ele é projetado para ser simples e fácil de usar, com uma interface de usuário intuitiva.

# PYCHARM

Com o PYCHARM você pode programar em Python.

O Pycharm é multiplataforma com versões para Windows, MacOS e Linux. O PyCharm é desenvolvido pela empresa JetBrains e fornece análise de código, depurador gráfico, autocompletamento de código e capacidades de navegação que facilitam a escrita de código.

Vamos instalar juntos!

<https://www.jetbrains.com/pt-br/pycharm/>

# ESTRUTURA DE DADOS

Estruturas de dados permitem armazenar e manipular coleções de valores.

Aqui estão algumas das estruturas de dados mais comumente usadas em Python:

1. Listas (Lists)
2. Tuplas (Tuples)
3. Dicionários (Dictionaries)
4. Conjuntos (Sets)

# Listas

Estruturas de dados permitem armazenar e manipular coleções de valores.

As listas são sequências mutáveis e ordenadas de elementos, que podem ser de diferentes tipos. Elas são definidas usando colchetes `[]` e os elementos são separados por vírgulas. É possível adicionar, remover e modificar elementos em uma lista.

```
lista = [1, 2, 3, 'a', 'b', 'c']
```

# Tuplas

Estruturas de dados permitem armazenar e manipular coleções de valores.

As tuplas são sequências imutáveis e ordenadas de elementos. Elas são definidas usando parênteses () ou simplesmente separando os elementos por vírgulas. Ao contrário das listas, as tuplas não podem ser modificadas após a criação.

```
tupla = (1, 2, 3, 'a', 'b', 'c')
```

# Conjuntos

Estruturas de dados permitem armazenar e manipular coleções de valores.

Os conjuntos são coleções não ordenadas e mutáveis de elementos únicos. Eles são úteis para realizar operações como união, interseção e diferença entre conjuntos. Os conjuntos são definidos a função `set()`.

```
conjunto = {1, 2, 3, 4, 5}
```

# Dicionários

Estruturas de dados permitem armazenar e manipular coleções de valores.

Os dicionários são coleções mutáveis de pares chave-valor, onde cada valor é associado a uma chave única. Eles são definidos usando chaves {} e os pares chave-valor são separados por dois pontos (:), com cada par separado por vírgula. Os dicionários permitem acessar, adicionar, modificar e remover valores com base em suas chaves.

```
dicionario = {'nome': 'João', 'idade': 25, 'cidade': 'São Paulo'}
```



# Exercício em Sala

## Jogo da Forca

Vamos usar nosso conhecimento em estrutura de dados para fazer com que o jogo se lembre das letras acertadas pelo jogador.

# Atividade para casa

Entrega junto com o jogo da adivinhação

Dada a lista = [13, -3, 5, 9, 19, 46, 79, 37, -18, 3, 13, 7, 4, 4, -42], faça um programa que:

- Imprima o maior elemento
- Imprima o menor elemento
- imprima os números pares
- imprima o número de ocorrências do primeiro elemento da lista
- imprima a média dos elementos
- imprima a soma dos elementos de valor negativo

# transforme ■ se

O conhecimento é o poder  
de transformar o seu futuro.