

# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**

# Estruturas de seleção

## Atividades práticas – Criação de programas

### Aula 4

Código da aula: [SIS]ANO1C1B2S9A4

## Exposição



### Objetivos da aula

- Compreender a aplicação prática das estruturas de seleção na lógica de programação;
- Conhecer exemplos de aplicações no do dia a dia, utilizando o *Python*.



### Competências da unidade (técnicas e socioemocionais)

- Desenvolver sistemas computacionais utilizando um ambiente de desenvolvimento;
- Explorar a criatividade na resolução de problemas computacionais.



### Recursos didáticos

- Recursos audiovisuais para exibição de imagens;
- Caderno, canetas e lápis;
- Computador com acesso à Internet.



### Duração da aula

50 minutos.

**Exposição**

## **Desenvolvimento da aula**

- ✓ Aplicação prática de algoritmos;
- ✓ Programação aplicada à resolução de problemas;
- ✓ Utilização de linguagens de programação para o aprendizado.



# Desenvolvimento da aula

Vamos recapitular o que são estruturas de seleção e sua importância ao longo do desenvolvimento de um código?

As estruturas de seleção são uma **parte fundamental da programação**. Elas permitem que um programa **execute diferentes ações dependendo de certas condições**. Em *Python*, as estruturas de seleção **mais usadas são *if*, *elif* e *else***.

*if* – É a estrutura básica de seleção. Ela executa um bloco de código se uma condição específica for verdadeira. Por exemplo:

**if condição: # código a executa se a condição for verdadeira.**

## Desenvolvimento da aula

***elif*** – Abreviação de "else if". É usada para verificar múltiplas condições. Se a condição do *if* não for verdadeira, ele verifica a condição do *elif*. Pode haver várias declarações *elif*, Por exemplo:

***if* condição1: # código a executa se a condição1 for verdadeira.**

***elif* condição2: # código a executa se a condição2 for verdadeira.**

***else*** – É usado após *if* e *elif*. Esse bloco de código é executado se todas as condições anteriores forem falsas. Por exemplo:

***if* condição: # código a executa se a condição for verdadeira.**

***else*: # código a executa se a condição for falsa.**





# Diferenças entre versões do *Python*?

Ao começarmos a trabalhar com *Python*, é muito importante sabermos que existem diferenças entre suas versões.

Exposição



ALURA. *Python: começando com a linguagem*. Diferenças entre o Python 2 e o Python 3. Disponível em: <https://cursos.alura.com.br/course/python-introducao-a-linguagem/task/22809> Acesso em: 13 fev. 2024. Acesso para link externo.

# Vamos programar

1. Vamos utilizar essas estruturas para **criar um jogo de adivinhação** e **um classificador de idades**. Ambos os programas dependem de tomar decisões com base na entrada do usuário. **No jogo, o programa decide se o palpite é alto ou baixo em comparação a um número secreto.** No **classificador de idades**, o programa decide em qual categoria de idade o usuário se encaixa.



### Atenção!

Desafie-se a fazer mais: que tal treinar o jogo de adivinhação com mais números?



Vamos  
fazer uma  
**atividade**

## Exercícios práticos – Criação de programas



### Materiais necessários

- Recursos audiovisuais para exibição de vídeos e imagens;
- Acesso ao laboratório de informática e à internet.



### Passo a passo

1. Abra o **programa de desenvolvimento em *Python***;
2. Faça a leitura do enunciado dos exercícios e atente-se ao que está sendo solicitado;
3. Resolva os exercícios propostos, não se esquecendo de realizar os testes dos códigos produzidos.

Construir um programa em *Python* que permita fazer a leitura do enunciado do exercício.



**30 minutos**



Faça download da explicação da atividade



O que nós  
**aprendemos  
hoje?**

## Hoje desenvolvemos:

- 1 Capacidade de usar a estrutura condicional *if*, *elif* e *else* para fazer seleções em nosso código, permitindo executar diferentes blocos de código com base em condições específicas;
- 2 Conhecimento sobre as diferentes versões da linguagem *Python*, além de identificarmos a versão utilizada dentro do nosso contexto de desenvolvimento;
- 3 Aplicação de forma prática e inserção de módulos dentro da linguagem *Python*, explorando detalhes sobre o funcionamento da solução randômica e criando modelos de jogos a partir da linguagem.

© Getty Images



# Saiba mais

Agora que **começamos a conhecer mais sobre as possibilidades do desenvolvimento em *Python*** para jogos simples, que tal o **aprender sobre o jogo *Snake* utilizando essa linguagem?**

HASHTAG PROGRAMAÇÃO. *Jogo Snake em Python*  
– Criando o jogo da cobrinha com PyGame  
[Tutorial completo]. Disponível em:  
<https://www.youtube.com/watch?v=bgsM YO m-W80> Acesso em: 13 fev. 2024.



# Referências da aula

MACHADO, V. *Estruturas de seleção*. Rio de Janeiro: IBMEC, [s.d.]. Disponível em: [https://victor0machado.github.io/prog/notas\\_aula/006-estruturas-selecao.html](https://victor0machado.github.io/prog/notas_aula/006-estruturas-selecao.html) Acesso em: 13 fev. 2024.

HASHTAG PROGRAMAÇÃO. *Jogo Snake em Python* – Criando o jogo da cobrinha com PyGame [Tutorial completo]. Disponível em: <https://www.youtube.com/watch?v=bgsM YO m - W80> Acesso em: 13 fev. 2024.

Identidade visual: Imagens © Getty Images

# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**