



escola  
britânica de  
artes criativas  
& tecnologia

# Profissão: Analista de dados



# VARIÁVEIS E TIPOS DE DADOS



# GUIA DA AULA 5



# Aprenda booleanos



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula

● **Motivação**

● **Definição**

● **Operações**

● **Conversão**

● **Revisitando a  
motivação**



## 1. Motivação

Em *websites* (redes sociais, *e-commerce*, corporativos, etc.) é comum o uso de sistemas de controle de acesso, o famoso *login*. Em geral, nestes sistemas um usuário fornece dois dados: `usuário` e `senha`.

In [ ]:

```

usuario = 'andre.perez'
senha = 'andre123'

```



## 1. Motivação

Do lado do servidor, o *backend* do *website* tem armazenado os dados de usuário e senha fornecidas pelo usuário no momento do cadastro:

`usuario_cadastro` e `senha_cadastro`.

```
In [ ]: usuario_cadastro = 'andre.perez'
        senha_cadastro = 'andre321'
```

Como comparamos se as *strings* (`usuario`, `usuario_cadastro`) e (`senha`, `senha_cadastro`) são iguais para conceder ou bloquear o acesso do usuário?



## 2. Definição

Armazenam **valores lógicos**:

- `True` (verdadeiro);
- `False` (falso).

```
In [ ]: verdadeiro = True
        print(verdadeiro)
```

```
In [ ]: falso = False
        print(falso)
```

São do tipo `bool`

```
In [ ]: print(type(True))
```



## 2. Definição

São resultados de comparações lógicas. Os operadores de comparação lógica são:

- > (maior);
- < (menor);
- == (igual);
- >= (maior ou igual);
- <= (menor ou igual);
- != (diferente).



## 2. Definição

### Exemplo 1: Caixa eletrônico

```

In [ ]:
saldo_em_conta = 200
valor_do_saque = 100

pode_executar_saque = valor_do_saque <= saldo_em_conta
print(pode_executar_saque)
  
```

### Exemplo 2: Cartão de crédito

```

In [ ]:
codigo_de_seguranca = '852'
codigo_de_seguranca_cadastro = '010'

pode_efetuar_pagamento = codigo_de_seguranca == codigo_de_seguranca_cadastro
print(pode_efetuar_pagamento)
  
```





### 3. Operações

As operações de variáveis booleanas são:

| (operador ou);  
 & (operador e);  
 not (operador não).

O conjunto de resultados de operações lógicas geralmente é resumido em uma tabela chamada "tabela da verdade":

A	B	A OR B	A AND B	NOT A
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE



### 3. Operações

**Exemplo:** Tabela da verdade do operador | (operador ou);

In [ ]:

```

print(True | True)
print(True | False)
print(False | False)
print(False | True)

```

**Exemplo:** Tabela da verdade do operador & (operador e);

In [ ]:

```

print(True & True)
print(True & False)
print(False & False)
print(False & True)

```



### 3. Operações

**Exemplo:** Tabela da verdade do operador not (operador não);

In [ ]:

```
print(not True)
print(not False)
```



## 4. Conversão

Podemos converter tipos numéricos e *strings* para booleanos através do método nativo `bool`

In [ ]:

```

idade = 19
tipo_sangue = 'O-'
filhos = 0
telefone_fixo = None
telefone_fixo = ''

print(bool(idade))
print(bool(tipo_sangue))
print(bool(filhos))
print(bool(telefone_fixo))
print(bool(telefone_fixo))

```



## 5. Revisitando a motivação

Compara se os dados fornecidos pelo usuário são iguais aos dados do cadastro:

In [ ]:

```

usuario_igual = usuario == usuario_cadastro
senha_igual = senha == senha_cadastro

print(usuario_igual)
print(senha_igual)

```

Decide se concede o acesso:

In [ ]:

```

conceder_acesso = usuario_igual & senha_igual
print(conceder_acesso)

```

