

```
name = 'Jason'  
if name == 'Jason':  
    print("Hello Jason, Welcome")  
else:  
    print("Sorry, I don't know you")
```

Reading Python Programming

Dados e Condições Lógicas

4 de julho de 2024

Agenda da Sessão

01. Dados *Simples*

02. Operadores Aritméticos, Lógicos e de Comparação

03. Expressões Condicionais

01

Dados *Simples*

Tipos de Dados

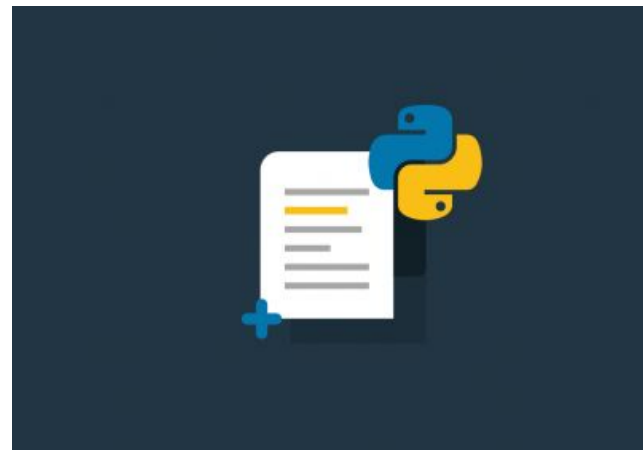
Contextualização

Geral

- Dados servem para serem armazenados e manipulados
- Compreensão e utilização de dados correta permite escalabilidade, eficiência e robustez do projeto

Python

- Existem 3 categorias fundamentais – numéricos, textuais e booleanos
- Existem ainda outros tipos de dados como listas, tuplos e dicionários e ainda objetos criados no âmbito do paradigma de programação orientada a objetos



Numéricos

int (Inteiros)

- Representa números inteiros

```
age = 28
```

float (Decimais):

- Representa números com casas decimais

```
height = 1.75
```

complex (Complexos):

- Representa números complexos, ou seja, números compostos por parte real e parte imaginária - representado por uma letra

```
complex = 3 + 2j
```

Textuais

str (Strings)

- Representa sequências de caracteres
- As podem ser manipuladas de várias maneiras, incluindo concatenação ou indexação

```
name = "Anna"
```

Booleanos

bool (Booleano)

- Representa valores verdadeiro (True) ou falso (False).
- Os booleanos são frequentemente usados em expressões condicionais e lógicas, como instruções if e while.

```
is_open = True
```

Conversão de Dados

Funções de Conversão

int()

- Converte um valor para um inteiro.

```
int(10.1) # retorna 10
```

float():

- Converte um valor para um número de ponto flutuante.

```
float("3.14") # retorna 3.14
```

str():

- Converte um valor para uma string.

```
str(42) # retorna "42"
```


Cocktail 1+1

Um mix de variáveis

Objetivo

Imprimir no terminal/consola a palavra “1+um=2.0”

Como?

1. Abrir ficheiro cocktail.py
2. Converter as variáveis para o tipo numérico correto
3. Converter as mesmas para string
4. Concatenar e imprimir no terminal



02

Operadores Aritméticos, Lógicos e de Comparação

Operadores Aritméticos

Adição (+)

```
total = product_price + tax
```

```
hello_person_text = "Hello, " + name + "!"
```

Subtração (-)

```
discounted_price = total - discount
```

Multiplicação (*)

```
total = product_price * quantity
```

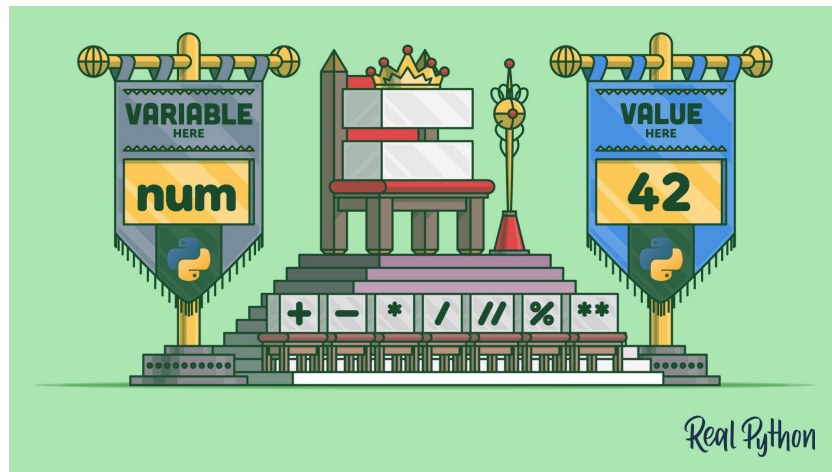
```
hello_world_repetitions = "Hello, World!" * 3
```

Divisão (/)

```
split_receipt = total / persons
```

Módulo (%)

```
remaining_items = total_items % persons
```



Operadores Aritméticos

Exponencial (**)

```
total_area_square = length ** 2
```

Divisão Inteira (//)

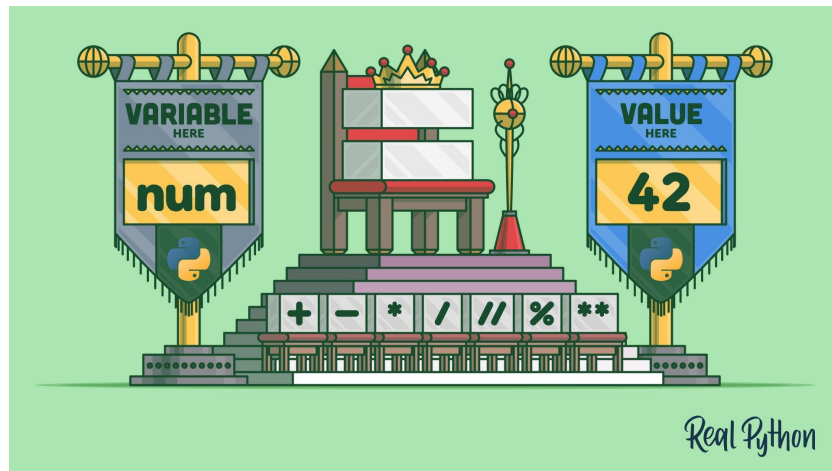
```
number_of_slices_per_person = total_slices // persons
```

Simplificação de fórmulas

```
price -= discount
```

é a simplificação de

```
price = price - discount
```



Operadores de Comparação

Operadores exclusivamente numéricos

Maior que (>)

```
driver_age > 17
```

Menor que (<)

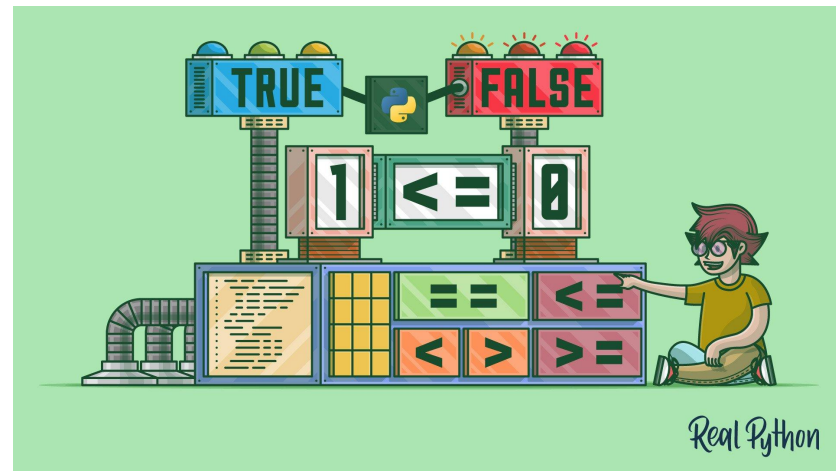
```
worker_age < 67
```

Maior ou igual que (>=)

```
driver_age >= 18
```

Menor ou igual que (<=)

```
worker_age <= 66
```



Operadores de Comparação

Operadores “universais”

Igual a (==)

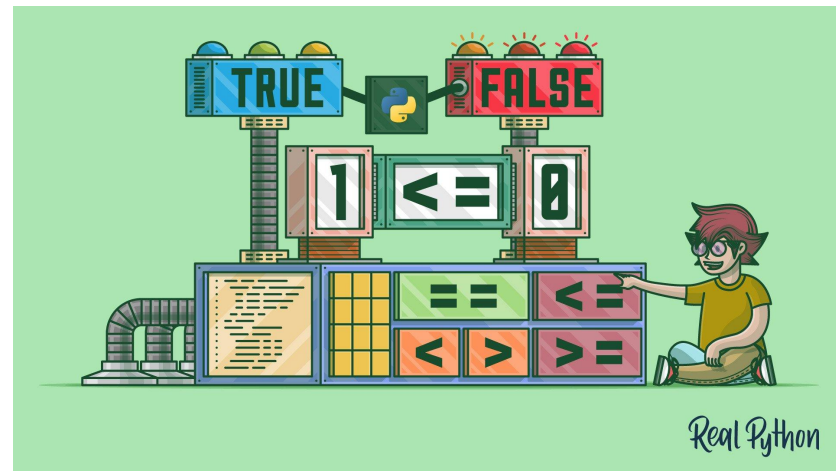
```
familiar_pizza_slices == 8
```

```
favourite_food == "Francesinha"
```

Diferente de (!=)

```
pizza_slices != 0
```

```
great_name != "Aires"
```



Operadores Lógicos

E (and)

- Para retornar verdadeiro (True) todas as comparações tem que ser verdadeiras

```
worker_age > 17 and worker_age < 67
```

Ou (or)

- Para retornar verdadeiro (True) basta uma das comparações ser verdadeira

```
payment_method == "Visa" or payment_method == "Mastercard"
```

Não é (not)

- Contraria o valor do booleano associado

```
not has_driving_license
```

Python - Logical Operators

- not

x	not x
False	True
True	False

- and

x	y	x and y
False	False	False
False	True	False
True	False	False
True	True	True

- or

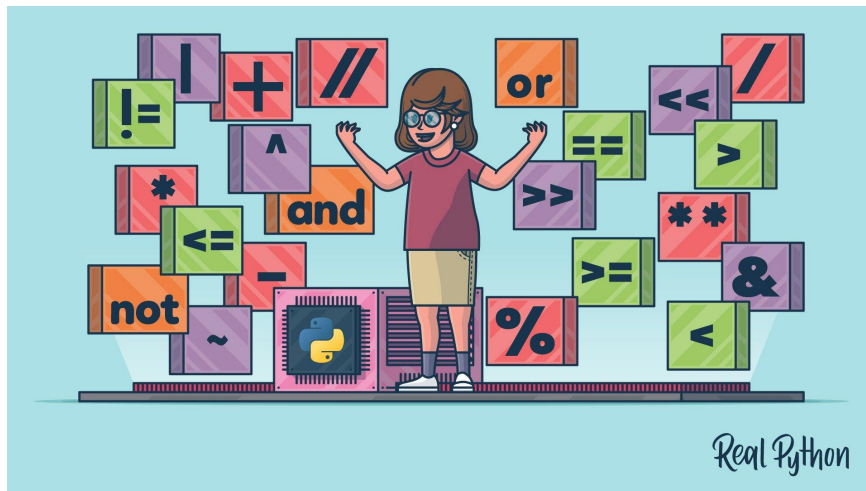
x	y	x or y
False	False	False
False	True	True
True	False	True
True	True	True

Operadores

Playground

Objetivo

Usar o ficheiro `operators_playground.py` para experimentar/criar exemplos de cada um dos operadores anteriores.



03

Expressões Condicionais

Fluxo Condicional

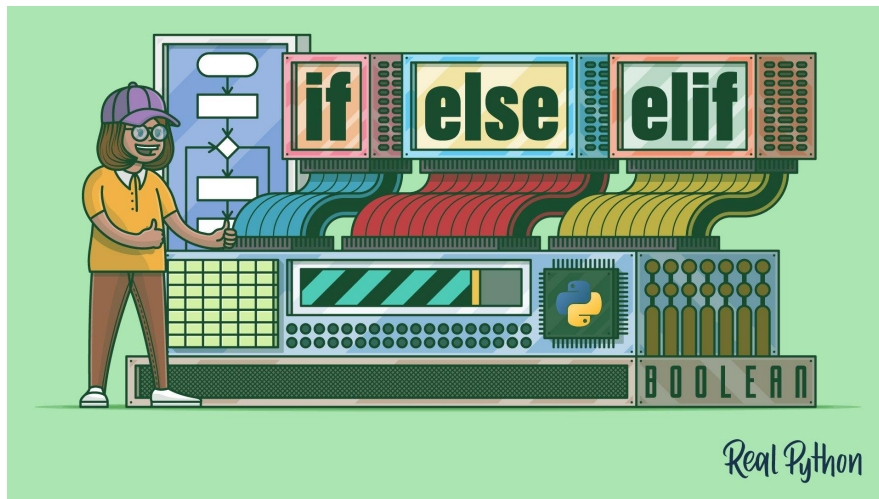
Contextualização

Geral

- As instruções condicionais permitem que o código seja flexível e consiga executar os fluxos definidos para cada contexto.
- A partir da veracidade de condições é possível definir diferentes decisões na execução do código

Python - Keywords

- If
- Else
- Elif



If

Contexto

- Permite execução de código em conformidade com a veracidade expectável na condição proposta
- Pode ser usado isoladamente

Implementação

- Inicia-se com if e após a condição é obrigatório colocar : (dois pontos)
- O código a executar tem que ser indentado corretamente (com 4 espaços ou um tab)

```
if age >= 65 :  
    print ("Senior Discount")
```

Else

Contexto

- Permite execução de código quando a veracidade na condição proposta não é a verdadeira
- Tem que ser utilizado na sequência do if
- Para cada if existe apenas um Else
- Não tem qualquer condição após a keyword dedicada;

Implementação

- Indentado ao mesmo nível do if
- O código a executar tem que ser indentado corretamente (com 4 espaços ou um tab)

```
if age >= 65 :  
    print ("Senior Discount")  
else :  
    print ("No Discount")
```

Elif

Contexto

- Permite execução de código quando a veracidade na condição inicial não é a verdadeira
- Tem que ser utilizado na sequência do if
- Cada if pode ter inúmeros elif
- Cada elif tem a sua condição própria
- Após se atingir a veracidade esperada num if/elif nenhum dos subsequentes corre
- Pode (e deve) ser finalizado com o Else

Implementação

- Indentado ao mesmo nível do if
- Inicia-se com elif e após a condição é obrigatório colocar : (dois pontos)
- O código a executar tem que ser indentado corretamente (com 4 espaços ou um tab)

```
if age >= 65 :  
    print ("Senior Discout")  
elif age < 3 :  
    print ("Free")  
elif age < 18 :  
    print ("Junior Discount")  
else :  
    print ("No Discount")
```

Satisfaz Bastante

Avaliação de alunos

Objetivo

Atualizando o ficheiro grade.py, retornar a menção qualitativa consoante a nota do aluno.

Extra

Consoante a menção qualitativa, retornar se o aluno passa ou não de ano

Percentagem	Menção Qualitativa
0 a 19	Fraco
20 a 49	Não Satisfaz
50 a 54	Satisfaz Pouco
55 a 64	Satisfaz
65 a 69	Satisfaz Bastante
70 a 74	Bom
75 a 89	Muito Bom
90 a 100	Excelente

Formação e
Certificação
Técnica que
potenciam
Profissionais e
Organizações

Lisboa

Edifício Mirage – Entrecampos
Rua Dr. Eduardo Neves, 3
1050-077 Lisboa

[ver google maps](#) ↗

Tel +351 217 824 100
Email info@training.rumos.pt

Siga-nos



Porto

Edifício Mirage – Entrecampos
Rua Dr. Eduardo Neves, 3
1050-077 Lisboa

[ver google maps](#) ↗

Tel +351 222 006 551
Email info@training.rumos.pt