



# Sejam bem-vindos!



Utilize a nossa redes de wi-fi:

**#ALDEIA**

utilizando a senha

n2opte

# A Aldeia é muito mais que espaço

---

Somos um movimento de desenvolvimento de realizadores.

Temos tudo que realizadores precisam para fazer uma ideia dar certo.

<http://aldeia.cc>

Cursos

Confrarias

Coworking

Offices

Networking

Eventos

Acelerações



# Não passe perrengue

---

Tem água e café à vontade, e um doce e um salgado para você pegar na hora que quiser.

Temos banheiros nos dois andares da **Cândido**:

- Primeiro andar: atrás da recepção
- Segundo andar: ao lado da escada

E atrás da recepção na unidade **Estação**.

**Se algo não estiver certo, fale com a nossa equipe**

# Faça parte da nossa Tribo

---

Receba os **materiais do curso** e seu **certificado** de participação por meio da nossa comunidade virtual.

Acesse <https://aldeia.cc/chamado> e faça sua solicitação para fazer parte da plataforma, utilizando o e-mail da compra do curso para se identificar.



Tire uma foto deste QR code e vá direto para a página da Tribo



# Curso de Data Science

Charles Adriano dos Santos  
Rafael Roberto Dias



# Manhã

---

## Horário Assunto

09:30	Nos episódios anteriores...
10:00	Homework ETL
10:30	Namorando os Dados (Queries SQL) – Continuação
12:00	Almoço

# Tarde

---

## Horário Assunto

- 13:00 Aprendendo Linguagem R no RStudio
- 15:00 Welcome to Python! (Básico + Numpy + Pandas + Banco)
- 17:00 Analisando Qualidade dos Dados

# Nos Episódios Anteriores...



Profissão Data Science

Estatística & Ciência da Computação

Desafio Agro XP

- ETL
- Modelagem de Dados
- Banco de Dados
- Queries SQL



**1 – Homework ETL**

**2 – Namorando Dados (Queries SQL)**

## 1 – Homework ETL

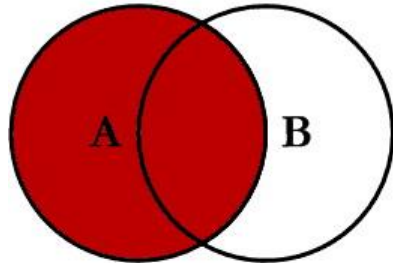
## 2 – Namorando Dados (Queries SQL)

1 – Homework ETL

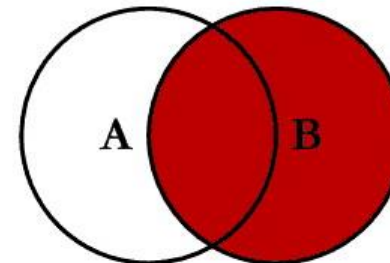
2 – Namorando Dados (Queries SQL)

# Namorando os Dados (Queries SQL)

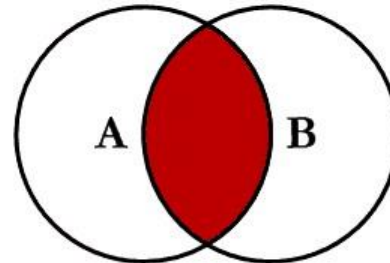
## SQL JOINS



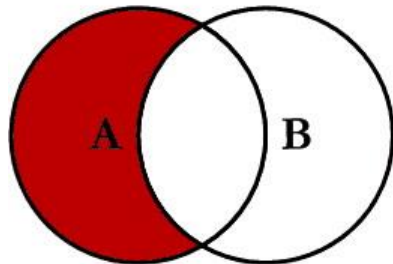
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



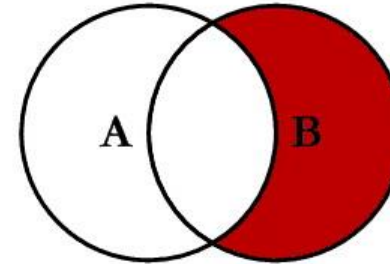
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



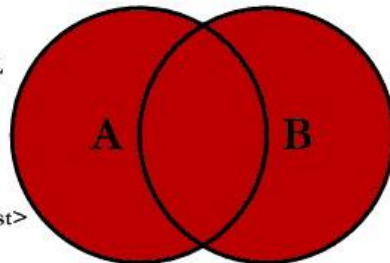
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



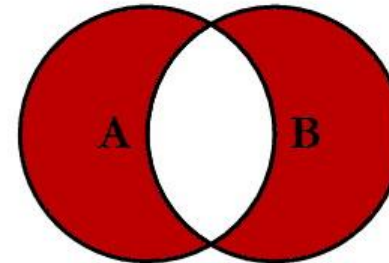
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

**1 – Aprendendo Linguagem R no RStudio**

**2 – Welcome to Python**

**3 – Analisando Qualidade dos Dados**

**1 – Aprendendo Linguagem R no RStudio**

**2 – Welcome to Python**

**3 – Analisando Qualidade dos Dados**



# Quais são os principais softwares Estatísticos?



- **MiniTab** - Software Matemático e Estatístico
- **SAS** - Statistical Analysis System
- **SPSS** - Statistical Package for the Social Sciences
- **S-PLUS** - Versão paga do R
- **Python** - Linguagem Interpretada
- **R** - (Ross e Robert)



- **Linguagem Alto Nível** - Longe do código de máquina e mais próximo à linguagem humana
- **Interpretada** - O programa resultante não é executado diretamente pelo sistema operacional ou processador
- **Script** - Programas escritos para um sistema de tempo que automatiza a execução de tarefas
- **Orientada a objetos** - Abstração, Encapsulamento, Herança e Polimorfismo

O R disponibiliza uma ampla variedade de



- Técnicas estatísticas
- Gráficos
- Modelos Lineares
- Modelos não Lineares
- Testes estatísticos clássicos
- Análises de Séries Temporais
- Classificação
- Agrupamento
- Machine Learning
- Artificial Intelligence

# Detalhes Software R



- O R é utilizado através de um Interpretador de comandos
- Ao escrever  $4 + 4$  na linha de comando, obtém-se o seguinte resultado:

```
> 4 + 4  
[1] 8  
> |
```

- A linguagem R suporta matrizes aritméticas, escalares, vetores, matrizes, quadros de dados (similares a tabelas numa base de dados relacional) e listas

# Detalhes Software RStudio



- RStudio é um software livre de ambiente de desenvolvimento, e que possui uma interface gráfica amigável
- O R Studio é uma interface para o R, com diversas utilidades diferentes que a tornam uma ferramenta mais simples em comparação ao R
- Ele possui duas versões: RStudio Desktop, que roda localmente em desktop e RStudio Server, que permite acessá-lo usando um navegador web enquanto ele roda em um servidor GNU/Linux remoto



**Bora  
Praticar?**





1 – Aprendendo Linguagem R no RStudio

2 – Welcome to Python

3 – Analisando Qualidade dos Dados

# Python



Mar 2019	Mar 2018	Change	Programming Language	Ratings	Change
1	1		Java	14.880%	-0.06%
2	2		C	13.305%	+0.55%
3	4	▲	Python	8.262%	+2.39%
4	3	▼	C++	8.126%	+1.67%
5	6	▲	Visual Basic .NET	6.429%	+2.34%
6	5	▼	C#	3.267%	-1.80%
7	8	▲	JavaScript	2.426%	-1.49%
8	7	▼	PHP	2.420%	-1.59%
9	10	▲	SQL	1.926%	-0.76%
10	14	▲▲	Objective-C	1.681%	-0.09%
11	18	▲▲	MATLAB	1.469%	+0.06%
12	16	▲▲	Assembly language	1.413%	-0.29%
13	11	▼	Perl	1.302%	-0.93%
14	20	▲▲	R	1.278%	+0.15%
15	9	▼▼	Ruby	1.202%	-1.54%
16	60	▲▲	Groovy	1.178%	+1.04%
17	12	▼▼	Swift	1.158%	-0.99%
18	17	▼	Go	1.016%	-0.43%
19	13	▼▼	Delphi/Object Pascal	1.012%	-0.78%
20	15	▼▼	Visual Basic	0.954%	-0.79%

Fonte: <https://www.tiobe.com/tiobe-index/>



# Python – Me Dê Motivos

**Linguagem em forte ascensão** ([3ª linguagem mais amada](#) Stack Overflow)

**Curva de Aprendizado Baixa**

**Free** (Licença GLP)



**Estável** (1ª versão 1991)

**Multiplataforma** (Windows, Linux, MacOS e etc.)

**Comunidade**

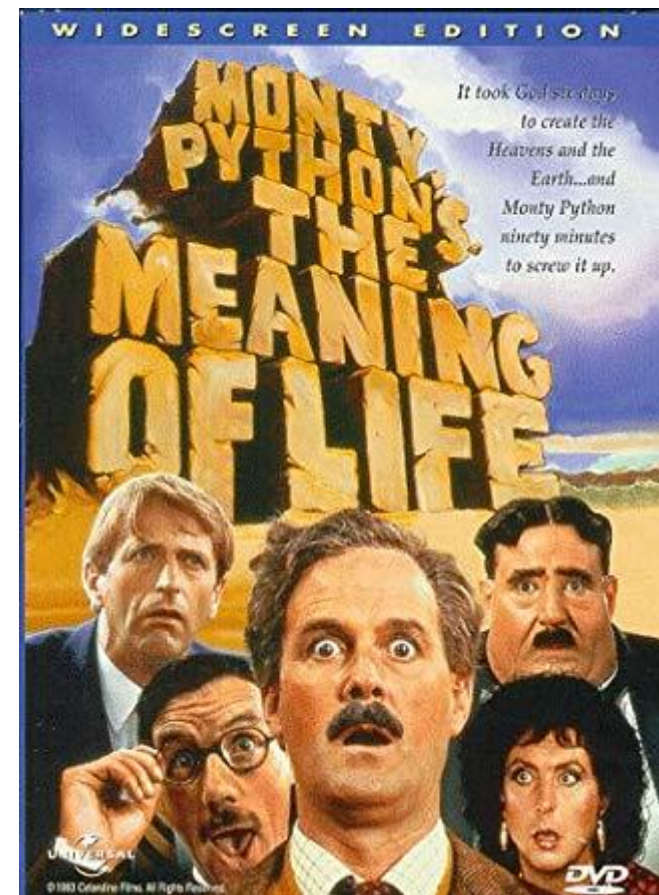
**Data Science → Ótimos pacotes**

# Python – História

Pai do Python →  
[Guido van Rossum](#)



A inspiração do nome →



## Versão 2 (2.7) x Versão 3 (3.5)

3/4 Paradigmas de Programação:

- **Programação Imperativa** → Ações/Comandos de um programa
- **Programação Orientada o Objeto** → Abstração, Encapsulamento, Herança e Polimorfismo
- **Programação Funcional** → Soluções como problemas de funções



Interpretada

# Python – Hands-on





# Python – Versão 2 x Versão 3



Python 2.X	Python 3.X
There's ASCII <code>str</code> type and <code>unicode</code> type, but no separate type to handle bytes of data	All strings ( <code>str</code> ) are Unicode strings; two byte classes are introduced: <code>bytes</code> and <code>bytearray</code>
Two types of integers: C-based integers ( <code>int</code> ) and Python long integer ( <code>long</code> )	All integers are long but referred to by the <code>int</code> type
Return type of division is <code>int</code> if operands are integers: <code>5 / 4</code> gives 1; <code>4 / 2</code> gives 2	Return type of division is <code>float</code> even if operands or result are integers: <code>5 / 4</code> gives 1.25; <code>4 / 2</code> gives 2.0
<code>round(16.5)</code> returns a float of value 16.0	<code>round(16.5)</code> returns an int of value 16
Unorderable types can be compared	Comparison of unorderable types raises a <code>TypeError</code>
<code>print</code> is a statement: <code>print "Hello World!"</code>	<code>print()</code> is a built-in function: <code>print("Hello World!")</code>
<code>range()</code> returns a list of numbers while <code>xrange()</code> returns an object for lazy evaluation	<code>range()</code> returns an object for lazy evaluation similar to Python 2 <code>xrange()</code> ; and <code>range()</code> method <code>__contains__</code> speeds up lookups
Functions/methods <code>map()</code> , <code>filter()</code> , <code>zip()</code> , <code>dict.items()</code> , <code>dict.keys()</code> , <code>dict.values()</code> return lists	These function/methods return objects for lazy evaluation
<code>raw_input()</code> returns input as <code>str</code> and <code>input()</code> evaluates the input as a Python expression	<code>input()</code> will return a string similar to Python 2 <code>raw_input()</code>
Raising exceptions: <code>raise IOError("file error")</code> or <code>raise IOError, "file error"</code>	Raising exceptions: <code>raise IOError("file error")</code>
Handling exceptions: <code>except NameError, err:</code> or <code>except (TypeError, NameError), err:</code>	Handling exceptions: <code>except NameError as err</code> or <code>except (TypeError, NameError) as err</code>
On generators, a method or function call: <code>g.next()</code> or <code>next(g)</code>	On generators, only a function call: <code>next(g)</code>
Loop variables in a comprehension leak to global namespace	Loop variables are limited in scope to the comprehension

Fonte: <https://devopedia.org/python-2-vs-3>

**1 – Aprendendo Linguagem R no RStudio**

**2 – Welcome to Python**

**3 – Analisando Qualidade dos Dados**

# Analizando a Qualidade dos Dados

- Objetivo nesta etapa do estudo é verificar a qualidade dos dados para entender quais tem potencial de fazer parte do estudo
- Foco maior em verificar se existem dados faltantes ou nulos que podem interferir no estudo
- Também aqui começa o entendimento de como cada variável ajuda a explicar o evento em estudo
- Aqui começam as **descobertas** do Cientista de Dados

# Obrigado!

📍 Charles Adriano dos Santos

✉️ [charles.a.santos@caelis.it](mailto:charles.a.santos@caelis.it)

🌐 chadri

☎️ 41 99144 6663

📍 Rafael Roberto Dias

✉️ [rafael.dias@madeiramadeira.com.br](mailto:rafael.dias@madeiramadeira.com.br)

🌐 [rafael-roberto-dias-00b39123](#)

☎️ 41 99672 7170